SVM-RCE-R-OPT: Optimization of Scoring Function for SVM-RCE-R

Malik Yousef[1](✉), Amhar Jabeer[2], and Burcu Bakir-Gungor[2]

[1] Department of Information Systems, Zefat Academic College, 13206 Zefat, Israel
[2] Department of Computer Engineering, Faculty of Engineering, Abdullah Gul University, Kayseri, Turkey
{amhar.jabeer,burcu.gungor}@agu.edu.tr

Abstract. Gene expression data classification provides a challenge in classification due to it having high dimensionality and a relatively small sample size. Different feature selection approaches have been used to overcome this issue and SVM-RCE being one of the more successful approach. This study is a continuation of two previous research studies SVM-RCE and SVM-RCE-R. SVM-RCE-R suggests a new approach in the scoring function for the clusters, showing that for some different combination of weights the performance was improved. The aim of this study is to find the optimal weights for the scoring function suggested in the study of SVM-RCE-R using optimization approaches. We have discovered that finding the optimal weights for the scoring function would improve the performance of the SVM-RCE- in most cases. We have shown that in some cases the performance is increased dramatically by 10% in terms of accuracy and AUC. By increasing the performance of the algorithm, it is more likely that we can extract subset genes relating to the class association of a microarray sample.

Keywords: Optimization · Gene expression classification · Machine learning

1 Introduction

Gene expression research is one of the major research areas in the field of bioinformatics. There is exponential growth in the biological data produced by DNA microarray technology [1–3]. This approach is high throughput, allowing scientists to measures multitudes of genes at the same time. Through this method, researchers can study and analyze numerous genes at the same time. DNA microarray technologies is providing great insight in genomic data and is changing the field of bioinformatics. Drug discovery, prevention of disease as well as cures, biological interactions, plant and animal metabolisms are underlying issues addressed by gene expression levels [4]. Additionally, there is widespread research in cancer studies to find potential biomarkers based on gene expression levels in order to find potential biomarkers [5–7]. The focus of research is on a small subset of genes that are relevant to the phenomenon under study among the different genes also known as the feature subset problem. DNA microarray technology are essentially the measurement of different genes at the stages of translation and transcription. There are two major methods of obtaining DNA microarray data: hybridization of

G. Kotsis et al. (Eds.): DEXA 2021 Workshops, CCIS 1479, pp. 215–224, 2021.
https://doi.org/10.1007/978-3-030-87101-7_21

sample to cDNA and high-density oligonucleotide chips [8]. Nevertheless, the data produced by these methods suffer from being highly redundant, large scale and the curse of dimensionality [9]. In order to solve the problems and dispel the curse, feature selection is the approach widely regarded by the bioinformatics community [10–12].

In general, feature selection can be categorized into three groups: filter approach, wrapper approach, embedded approach (combination of the previous methods) [13]. Filter methods focuses on the intrinsic characteristics of the genes in terms of their relevance or in their discriminative properties. The genes are ranked according to the filter method and the highest ranked genes are used are the remaining are eliminated. This methodology does not rely on any machine learning algorithm therefore the time complexity is quite low and can be used for large datasets. Moreover, the results are simplified and can be easily be verified in wet labs by biological domain experts. Thus, univariate filter approaches have widely leveraged to analyze and study gene expression levels [14]. Among the different filter approaches, Xing et al. [15] reports that IG (Information Gain) to be the best approach. However, this approach does not perform well for heterogeneous datasets whereas Bayesian Networks show their strength in this regard [16]. Therefore, different filter techniques outperform each other depending on the dataset. In wrapper approach, the genes are searched then judged based on the estimated accuracy of a classifier. The extracted genes are then used to train the classifier. Zhang et al. [17] asserted that wrapper methods outperform filter methods in terms of predictive accuracy of the classifier. Moreover, this approach also integrates the interaction of the gene selection with the classification that is independent in the filter approach. Nevertheless, this approach has cost of being computationally intensive and in some cases cause overfitting of the classifier [18]. Finally, we have embedded approaches wherein the search algorithm is rooted in the classification algorithm. Therefore, it has the advantage of the interaction of search algorithm with the classifier while being far less computationally intensive [19]. One of the more successful approaches is to use SVM (Support Vector Machines) with an embedded feature selection algorithm [20].

SVM-RFE (Recursive feature elimination) [21] was introduced where the authors achieved very high accuracy with their classifier in comparison to other discriminant methods using SVM. In this method, the genes are ranked as features and the lowest ranked features are removed. Yousef et al. [22] introduced SVM-RCE (Recursive cluster elimination); moreover, it was reported to outperform SVM-RFE. SVM-RCE uses KNN to cluster the genes and then uses SVM to rank the clusters with their respective scores while eliminating the lower ranked clusters. Based on its widespread interest, Luo et al. [23] recently improved the computation time by applying an infinite norm of weight coefficient vector to each cluster to score them. They removed the lowest performing genes instead clusters when the number of clusters are small. Additionally, we wanted to empower SVM-RCE and we introduced SVM-RCE-R (Rank) [24] that extended SVM-RCE with a user specific ranking function. Here the user can choose which clusters should be ranked higher based on different metrics (accuracy, sensitivity, f-measure, area under the curve and precision), thereby allowing scientists to explore the biological data in depth to their needs.

Based on improving this method on a greater scale, we are now introducing SVM-RCE-R-OPT which searches for the optimal set of weights resulting in an improvement

in our classification results. We use Bayesian optimization to find the parameters for our six different weights. We compare SVM-RCE and SVM-RCE-OPT across 15 datasets to validate the findings that this approach does improve the classification results.

2 Methods and Implementation

We optimize SVM-RCE-R which is based on an early study SVM-RCE which lead us to the present approach of SVM-RCE-R-Opt. The methods and approaches used are described in the upcoming sections for SVM-RCE and SVM-RCE-R. We then describe in detail how we optimized SVM-RCE-R algorithm and the platform we used to implement SVM-RCE-R-Opt.

2.1 SVM-RCE

SVM-RCE is the first algorithm that suggests clustering genes using K-Means into clusters arranged according to correlation metric, in order to perform feature selection procedure by considering each cluster of genes as one unit. Then one needs to score each cluster of genes in terms of the classification of the training set that consist of two-classes. For that purpose, the training data was transformed to be represented based on the genes that belong to a specific cluster with the original class of the training set. Then an internal cross-validation is performed in order to compute the score. The score is the average of the accuracy performance of the cross-validation step. This step is applied for each cluster detected by k-mean. The next step is to rank all the clusters according to its score. The SVM-RCE removes the cluster with the lowest score or it can set to remove percentage of the lowest scored clusters. Thus, the results obtained is without the genes that are associated with the removed clusters as they do not contribute much to the prediction capabilities of the classifier.

2.2 SVM-RCE-R

Based on the interests of SVM-RCE in biological research, we decided to empower this algorithm with a user specific ranking function in SVM-RCE-R. In SVM-RCE, clusters were scored according to their accuracies. However, in data analysis of high dimensionality data with small sample size other metrics are preferred in the understanding of the features that contribute most to classification. This lead to the implementation of a user specific scoring function, in which researchers can choose the scoring function according to their needs. Therefore, we used the commonly used scoring metrics as described in Eq. 1 as our scoring function.

$$S(w1,w2,w3,w4,w5,w6) = w1 \times acc + w2 \times sen + w3 \times spe + w4 \times fm + w5 \times auc + w6 \times prec \quad (1)$$

where acc, sen, spe, fm, auc and prec refer to accuracy, sensitivity, specificity, f-mean, precision and area under curve respectively.

The ranking is then computed as a sorted list of clusters based on the score S(). We noted that we could achieve significant improvements in our results in comparison to

SVM-RCE. We also did notice that some combinations of clusters with different weights lead to better accuracy score even when the focus was solely on accuracy (Accuracy had the highest weight and the remaining metrics are zero). Therefore, the most important aspect is how one can compute the optimal weights w1,w2,w3,w4,w5,w6 such that it improves the overall performance of the algorithm. Therefore, we decided to focus on an optimization approach to find the best combination of weights for our next step in this approach.

2.3 SVM-RCE-R Optimal

Our new proposed approach SVM-RCE-R-OPT is implemented in KNIME [25] due to its user friendliness similarly to SVM-RCE-R. We split the gene expression dataset into train and test sets with a ratio of 30:70 respectively. Moreover, we used stratified sampling to make sure the training data and test data have the same ratio of negative to positive samples. The parameter optimization node in KNIME is used to find the optimal weights for our six different ranks which was used in SVM-RCE-R (acc, sen, spec, etc.). This node uses Bayesian optimization [26] as the search strategy to find the optimal weights for our six different ranks. The algorithm that is used for the maximization for the objective function is illustrated in Eq. 2 based on the original paper.

$$EI_{y*} = \int_{-\infty}^{\infty} \max(y * -y, 0)\ PM\,(y \mid x)dy \tag{2}$$

The search strategy works in two phases: warm up phase and then the Tree-structured Parzen Estimation (TPE) phase. During the warm up phase, random combinations of the weights are used, and then based on the objective values found, the TPE phase starts. Moreover, users can specify the step size of each weight as well as the number of iterations for each phase. Meanwhile, the search algorithm draws weights with replacement from the search space.

Our objective function is based on our scoring function which was stated in Eq. 1. The means the scoring function across all the cluster levels (number of clusters) is used as our objective function, which was set to be maximized. We specified the step size to be 1000, since prior testing showed that there was an improvement by using heavier weights. After the optimal set of weights have been identified in the search space from the Bayesian optimization, they are then used in a separate node that runs SVM-RCE-R with those weights with the training split of the data. Similarly, we also ran SVM-RCE-R with only the weight of accuracy (acc) set to the maximum and the remaining weights set to zero on the same training set. This provides us a reference to validate whether the weights found are an improvement over the original SVM-RCE algorithm, since the algorithm uses accuracy as the ranking function.

2.4 KNIME Workflow

There the overview of the workflow that we programmed in KNIME [25] as reflected in Fig. 1. The user has to set the list file node to the folder that contains the datasets. The workflow then generates a folder with all the relevant output files.

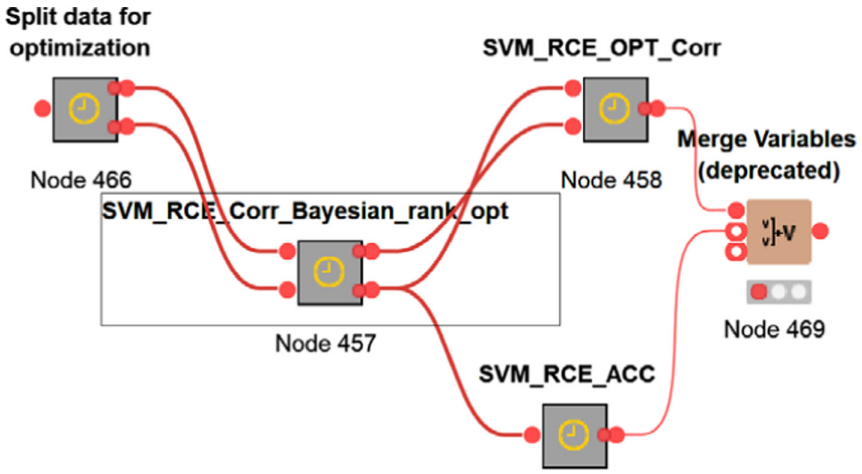

Fig. 1. KNIME workflow overview

The first step in the workflow is to split the data into two parts. One to calculate the optimal weights of the scoring function and the other for cross-validation of the standard approach (SVM_RCE_ACC meta node) and for the optimal approach (SVM_RCE_OPT_Corr meta node). The data that was used in computing the optimal weights is not used in the next steps to avoid overfitting.

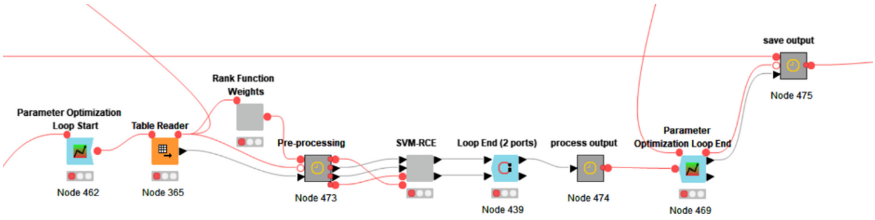

Fig. 2. Bayesian optimization of weights

The SVM-RCE-R is applied with different weights (Parameter Optimization Loop start node) on the training set (Fig. 2) where the results are collected at the loop end node. The maximum iterations used to find the optimum weights was limited to 15 iterations. The different weights are then computed according to the search algorithm mentioned in the previous section. When the optimal set of weights are discovered, then it is used for testing set of the data (Save output node).

3 Data

We have considered the same data used in the study of SVM-RCE-R and we have included two additional new datasets. The datasets represent a range of different

diseases, cancers and studies. In terms of diseases: Early-stage Parkinson's disease (GDS2519), Ulcerative colitis (GDS3268), celiac disease (GDS3646), diabetes in children (GDS3875), effects of tobacco smoke in foetal cells (GDS3929), HIV (GDS4228), asthma (GDS5037), acute dengue (GDS5093), pulmonary hypertensions (GDS5499), multi-omic analysis of COVID-19 (GSE157103). In terms of different types of cancer: glioma (GDS1962), prostate cancer (GDS2547), colorectal cancer, (GDS4516_4718), fear conditioning studies in mice (GDS3900). All of the datasets have at least 100 samples except for GDS5093 that has 56 samples. These 14 gene expression datasets are downloaded from Gene Expression Omnibus at NCBI (GEO) [27]. The format of datasets contains sample IDs as the column names and the gene name (gene symbols), according to their respective platform, as the row names with their relevant gene expression values (Fig. 3). Since most of the datasets produced are based on different chips or platforms, the number of genes vary and the exact amount can be found by their GEO accession numbers.

Table "default" - Rows: 180 Spec - Columns: 54614 Properties Flow Variables

Row ID	class	MIR4640	RFC2	HSPA6	PAX8	GUCA1A	MIR5193	THRA	PTPN21	CCL5	CYP2E1	EPHB3	ESRRA	CYP2A6	SCARB1
GSM97800	neg	4701.5	282.7	769.6	1616.3	232.7	357.7	245.1	33.2	30.7	224.6	107.5	738.3	314.8	1074.7
GSM97803	neg	4735	347.9	287.9	1527.2	204.8	336.5	186.2	22.9	57.1	133.7	270.8	564.2	355.8	1114.3
GSM97804	neg	2863.9	355	199	1793.8	119.3	328.7	349.3	30	17.8	270.1	300.8	510.1	371.6	1191.8
GSM97805	neg	5350.2	319.9	182.8	1880	180.2	304.7	325.4	47.6	30.7	186.4	163.2	542	336.2	1019.8
GSM97807	neg	4789.4	294.2	204.3	1012	156.7	190.1	132	18.8	11.8	218.5	221.4	601.2	216.4	1342.5
GSM97809	neg	5837.8	257.5	184.9	1024.4	155.1	253.3	182.6	28.9	13.4	165.8	355.3	698.2	268.5	1196.6
GSM97811	neg	4446.7	321	107.5	1133.8	236.2	342.5	184.5	15.8	17.9	249.4	212.6	599.1	287.6	1356.5
GSM97812	neg	4264.1	317.9	196.9	1295	235.9	284.1	214.2	22.5	29.1	275.7	228.9	572.8	230	1442.5
GSM97816	neg	11011.5	283.2	225.8	1500.6	65.9	371.2	170.4	48.2	34.1	225.3	132.3	616.5	317.9	1737.6
GSM97817	neg	3832.6	330.9	274.1	1736.6	194.3	521.3	244.6	33.6	30.6	183.2	303.4	716.9	365.6	1085.8
GSM97820	neg	5227.2	340.8	253.6	1504.2	282.2	322.8	192.4	24	16.8	177.2	260.6	562	357.7	1247
GSM97825	neg	2935.6	327	157.9	1651.9	243.1	354	234.7	29.9	24.1	171.7	365	573	337	1370.9
GSM97827	neg	3561.5	363.4	94.3	1410.2	271.1	310.4	215.6	78.8	56.7	138.1	460.2	791.9	309.4	1322
GSM97828	neg	10728.8	268.4	245.4	1423.4	75	424.6	162.8	27.6	45.7	196.7	200.3	616.6	371.9	1834.8
GSM97833	neg	4156.6	266.5	232.8	1420.9	321.1	275	146.5	102.9	12.5	168.9	283.6	557.5	313.9	1456.4
GSM97834	neg	4278.4	224.7	245.2	1544.2	264.9	267.7	283.8	41.7	22.3	161.5	372.1	646.4	297.7	1283.3
GSM97840	neg	5916.2	303.7	217	1442	137.6	443.4	111.5	92.1	192.7	142.9	293.3	704.3	306.8	1518.8
GSM97846	neg	3136.6	353.9	247.2	1662.2	217.2	325.8	326.5	53.6	7.1	227.8	195.8	632	314.8	1095.9
GSM97848	neg	4415.7	303	180.9	1345.5	288.9	321.3	212.2	18.6	17.9	123.1	197.3	539.2	299.8	1128.7
GSM97849	neg	3313.4	331.7	174.6	1378.2	261.5	357.3	172.5	17.5	32.3	193.1	141	627.9	293.7	1397.5
GSM97850	neg	6011.4	258.6	188.8	1632.2	331.9	413	257.1	88.6	15.1	238.2	290.8	687.1	304.8	1191.7
GSM97853	neg	4107.2	248.6	32.2	1427.9	316.6	277.6	238.5	21.6	4.5	255.5	241.6	610.5	286.1	1043.4
GSM97855	neg	7053.4	330.5	182.4	1258.5	172.3	219.5	139.1	18.2	12.5	158.5	219.9	618	220	1260.3
GSM97878	pos	8196.2	288.5	453.9	1589.3	6.4	598.7	281.5	143.1	1519.7	61.2	191.8	599.8	373.3	1803
GSM97913	pos	6006.6	412.5	3748.5	1779.1	13.8	421.5	250.3	42.8	76.6	115.4	348.1	666.6	279.5	1723.7
GSM97932	pos	2619.5	330.2	850	1705.6	118.9	875.5	256	77.2	2111	120.8	136.6	784.5	321.5	969.9
GSM97939	pos	11699.2	349.9	138.8	1388.6	88	486.5	213.6	163.1	10.3	251.4	899	919.2	311.8	2359.5
GSM97951	pos	7718.5	394.9	267.3	1325.3	190.6	387.7	200.6	80.2	14.2	196.1	372.3	645	248.7	1833.5
GSM97957	pos	12700.4	440.8	201.2	1209.2	31.8	489.9	152.8	66.5	13.4	249.6	224.9	701.5	258	1558.6
GSM97972	pos	8816.4	286.9	167.8	1398.6	54.9	461.3	158	25.6	26	303	428.1	891.3	281.5	1131.6
GSM97793	pos	10178.1	388.2	227.3	1665.4	90.7	469.9	459.1	20.4	63.8	293.2	140.4	783.6	292.2	852.4
GSM97795	pos	7826.6	352.4	306	1967	106.4	203.8	266.3	24.3	38.4	386.9	282.2	705.2	315.5	1247.3
GSM97802	pos	6857.8	735.4	261.6	1740.9	316.7	282.6	265.3	25.8	14.4	44.9	575.8	678.4	317	1427.6
GSM97810	pos	8255	373.8	149.5	972.1	99	439.8	196	130.1	13.8	111.7	276.5	610	238	1092.7
GSM97815	pos	8016.8	383.8	250.3	1668.4	134.4	483.1	282.9	32	20.9	164.9	501.5	791.3	295	1234.8

Fig. 3. Input table (dataset) format in KNIME

4 Results

We have applied 100-fold Monte Carlo cross-validation [27] for the original approach and for the optimal approach. In each fold, we compute different performance metrics such as accuracy, specificity and area under curve (AUC). The average of all the 100 folds is computed for each metric. Accuracy and specificity is computed in Eq. 3 where TP is true positives, TN is true negatives, FP is false positives and FN is false negatives. Meanwhile AUC is calculated based on the probability that a classifier will rank a randomly positive instance higher that a negative one.

$$Specificity = TN \ / \ (TN + FP)$$

$$Accuracy = (TP + TN) \ / \ (TP + TN + FP + FN) \tag{3}$$

To validate the optimal weights found from the training, we compute the difference in accuracy between cluster level 90 of the optimal solution and the base accuracy used in the original study SVM-RCE [22]. Using the optimized weights from the training part of the algorithm, we observed that we have an improvement for seven of the datasets, while two of the datasets (GDS3900, GDS4516_4718) showed similar accuracy as shown in Fig. 4. The figure shows that in some cases the improvement over the standard approach might even reach to 10%; in most cases, the improvement is around 5%.

Since we are dealing with high dimensional data, we need to look at other metrics more specifically AUC and specificity. In terms of the specificity, Fig. 5 illustrates that ten out of twelve datasets outperformed or had similar performance as the original SVM-RCE algorithm. This could imply that we are not overfitting in this approach and the robustness of the classifier is still preserved. Additionally, when comparing the AUC in Fig. 6, we note that the optimal approach generally shows an even greater improvement. We can see that seven of the datasets shows either similar or better performance. We can see about a 10% increase in AUC performance for four of the datasets (GDS2519, GDS3646, GDS3875, GDS5093) which is quite a significant improvement. From these results, we can conclude that there is an overall increase in the performance of the datasets. We believe if the number of iterations to search for the optimal set of weights is increased, we may see improvements across all the datasets. However, this could have the drawback of being computational expensive as well as time consuming.

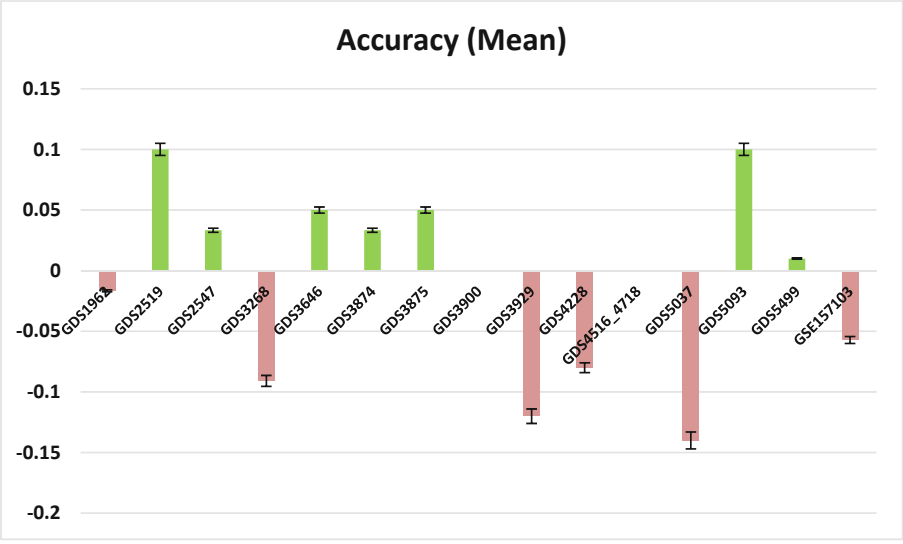

Fig. 4. The difference between the accuracy of the optimal weights to the base accuracy

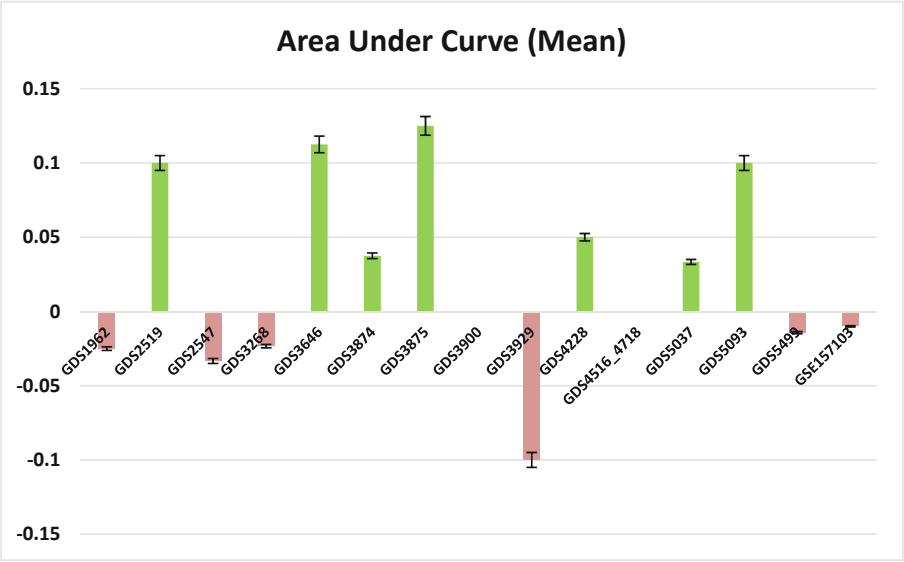

Fig. 5. The difference between the AUC of the optimal weights to the base AUC

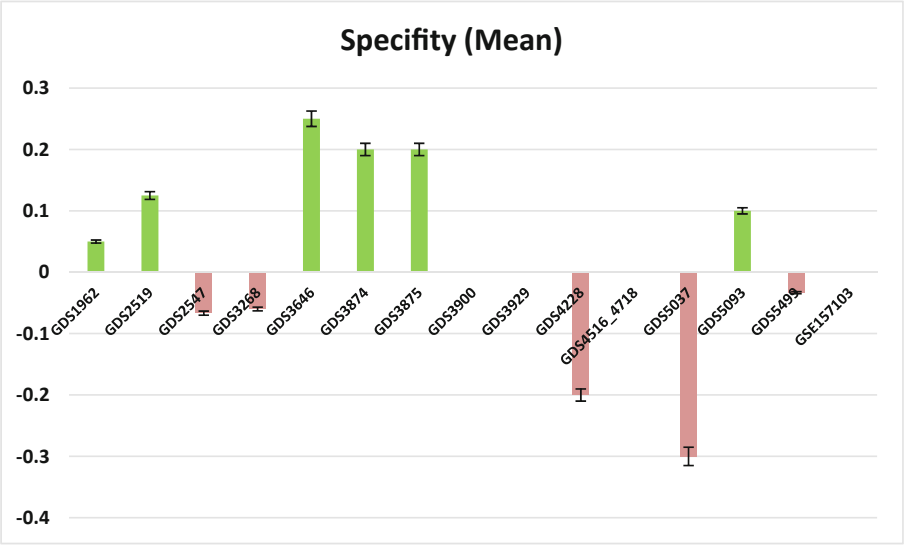

Fig. 6. The difference between the sensitivity of the optimal weights to the base sensitivity

5 Conclusions

In this study, we have proposed an optimization approach for computing the weights for the scoring function that would provide the optimal solution for ranking the clusters. We perform this by using Bayesian search optimization to find the optimal set of weights,

then we compare the results with the SVM-RCE. Since SVM-RCE operated using a different ranking function, we wanted to validate whether this approach provides an improvement. When comparing the results across 12 datasets the overall performance is improved in most cases in terms of the accuracy, sensitivity and AUC. However, we would like to note that this algorithm is time consuming since the optimization procedure requires a long time in order to find the suitable weights in the search space. Moreover, approach also allows researchers to understand underlying genes related to biological research since it is based on the SVM-RCE. Therefore, we can find the genes that most contribute to the certain disease or specific research topic (e.g. fear factor). This would help us to find genes that help in identifying diseases in terms of expression levels, under expressed and overexpressed. This could potentially help in medical diagnosis of diseases and understanding of the role genes play in biological processes.

References

1. Chandra, B., Gupta, M.: An efficient statistical feature selection approach for classification of gene expression data. J. Biomed. Inf. **44**, 529–535 (2011)
2. McConnell, P., Johnson, K., Lockhart, D.J.: An introduction to DNA microarrays. In: Methods of Microarray Data Analysis II. Proceedings of the Second Conference on Critical Assessment of Microarray Data Analysis, CAMDA 2001, pp. 9–21. Kluwer Academic Publishers, Dordrecht (2002)
3. Dopazo, J.: Microarray data processing and analysis. In: Methods of Microarray Data Analysis II, Proceedings of the Second Conference on Critical Assessment of Microarray Data Analysis, CAMDA 2001, pp. 43–63. Kluwer Academic Publishers, Dordrecht (2002)
4. Riva, A., Carpentier, A.S., Torresani, B., Henaut, A.: Comments on selected fundamental aspects of microarray analysis. Comput Biol Chem **29**, 319–336 (2005)
5. Veer, L., Da, H., Bijver, M., et al.: Gene expression profiling predicts clinical outcome of breast cancer. Nature **415**, 530–536 (2002)
6. Zajchowski, D., et al.: Identification of gene expression profiles that predict the aggressive behavior of breast cancer cells. Cancer Res. **61**, 5168–5178 (2001)
7. Veer, L., Jone, D.: The microarray way to tailored cancer treatment. Nat. Med. **8**, 13–14 (2002)
8. Allison, D.B., Cui, X., Page, G.P., Sabripour, M.: Microarray data analysis: from disarray to consolidation and consensus. Nat. Rev. Genet. **7**, 55–65 (2006)
9. Ying, L., Han, J.: Cancer classification using gene expression data. Inf. Syst. **28**, 243–268 (2003)
10. Lazar, C., et al.: A survey on filter techniques for feature selection in gene expression microarray analysis. IEEE/ACM Trans. Comput. Biol. Bioinf. **9**, 1106–1119 (2012)
11. Li, T., Zhang, C., Ogihara, M.: A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. Bioinformatics **20**, 2429–2437 (2004)
12. Ang, J.C., Mirzal, A., Haron, H., Hamed, H.N.A.: Supervised, unsupervised, and semi-supervised feature selection: a review on gene selection. IEEE/ACM Trans. Comput. Biol. Bioinf. **13**, 971–989 (2016)
13. Zhu, S., Wang, D., Yu, K., Li, T., Gong, Y.: Feature selection for gene expression using model-based entropy. IEEE/ACM Trans. Comput. Biol. Bioinf. **7**, 25–36 (2010)
14. Aris, V., Recce, M.A.: Method to improve detection of disease using selectively expressed genes in microarray data. In: Methods of Microarray Data Analysis, Proceedings of the First Conference on Critical Assessment of Microarray Data Analysis, CAMDA 2000, pp. 69–80. Kluwer Academic Publishers, Dordrecht (2002)

15. Xing, E.P., Jordan, M.I., Karp, R.M.: Feature selection for high-dimensional genomic microarray data. In: Proceeding of 18th International Conference on Machine Learning (2001)
16. Giallourakis, C., Henson, C., Reich, M., Xie, X., Mootha, V.K.: Disease gene discovery through integrative genomics. Annu. Rev. Genomics Hum. Genet. **6**, 381–406 (2005)
17. Zhang, H., Ho, T.B., Kawasaki, S.: Wrapper feature extraction for time series classification using singular value decomposition. Int. J. Knowl. Syst. Sci. **3**, 53–60 (2006)
18. Loughrey, J., Cunningham, P.: Overfitting in wrapper-based feature subset selection: the harder you try the worse it gets. In: Bramer, M., Coenen, F., Allen, T. (eds.) Research and Development in Intelligent Systems XXI, pp. 33–43. Springer London, London (2005). https://doi.org/10.1007/1-84628-102-4_3
19. George, V.S., Raj, C.: Review on feature selection techniques and the impact of svm for cancer classification using gene expression profile. Int. J. Comput. Sci. Eng. Surv. **2**, 16–27 (2011)
20. Li, F., Yang, Y.: Analysis of recursive gene selection approaches from microarray data. Bioinformatics **21**, 3741–3747 (2005)
21. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. **46**, 389–422 (2002)
22. Yousef, M., Jung, S., Showe, L.C., et al.: Recursive cluster elimination (RCE) for classification and feature selection from gene expression data. BMC Bioinformatics **8**, 144 (2007)
23. Luo, L., Huang, D., Ye, L., Zhou, Q., Shao, G., Peng, H.: Improving the computational efficiency of recursive cluster elimination for gene selection. IEEE/ACM Trans. Comput. Biol. Bioinf. **8**, 122–129 (2011)
24. Yousef, M., Bakir-Gungor, B., Jabeer, A., Goy, G., Qureshi, R., Showe, L.C.: Recursive cluster elimination based rank function (SVM-RCE-R) implemented in KNIME. F1000Research **9**, 1255 (2020). https://doi.org/10.12688/f1000research.26880.1
25. Berthold, M.R., et al.: KNIME - the Konstanz information miner. SIGKDD Explorations **11**, 26–31 (2009). https://doi.org/10.1145/1656274.1656280
26. Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Proceedings of the 24th International Conference on Neural Information Processing Systems, pp. 2546–2554. Curran Associates Inc., Red Hook, NY (2011)
27. Barrett, T., et al.: NCBI GEO: archive for functional genomics data sets—update. Nucleic Acids Res. **41** (2013). https://doi.org/10.1093/nar/gks1193
28. Xu, Q.-S., Liang, Y.-Z.: Monte Carlo cross validation. Chemom. Intell. Lab. Syst. **56**, 1–11 (2001). https://doi.org/10.1016/S0169-7439(00)00122-2