ARG WI2 No.15, 2019
WI2-2019-18

A proposal to improve the performance of feature selection methods with low-sample-size data

Wanwan Zheng[1] Mingzhe Jin[2]

[1] [2] Graduate School of Culture and Information Science, Doshisha University

teiwanwan@gmail.com

Abstract Feature selection refers to a critical preprocessing of machine learning to remove irrelevant and redundant data. According to feature selection methods, sufficient samples are usually required to select a reliable feature subset, especially considering the presence of outliers. However, sufficient samples cannot always be ensured in several real-world applications (e.g. neuroimaging, bioinformatics, psychology, as well as sport sciences). In this study, a method to improve the performance of feature selection methods with low-sample-size data was proposed, which is named Feature Selection Based on Data Quality and Variable Training Samples (QVT). Given that none of the considered feature selection methods perform optimally in all scenarios, QVT is primarily characterized by its versatility, because it can be implemented in any feature selection method. An experiment was performed using 20 benchmark datasets, three feature selection methods and three classifiers to verify the feasibility of QVT; the results suggested that QVT was applicable to different feature selection methods and significantly improved predictive performance of different classifiers.

Keywords Feature selection, low-sample-size data, predictive performance, variable training samples

1 Introduction

According to feature selection methods, sufficient samples are usually required to select a reliable feature subset. In a dataset with a considerable number of samples, the effects of outliers will be limited, and the training data will represent the population at large. However, with low-sample-size data, the values of few outliers can significantly convert the set of selected features into a new set of potential noisy features that may not fully reflect or capture class-specific differences (Golugula and Lee, 2011). Furthermore, though conventional feature selection adopts random sampling to improve the performance, low-sample-size datasets are typically too small to be processed using this method. Datasets are characterized by a small number of samples that are common in plenty of areas (e.g. studies of rare diseases or extraordinary athletes).

The question that whether the available sample has only a couple of dozens or even less has been raised by Raudys and Jain in 1990. The authors discussed the effects of sample size on the feature selection and error estimation for several types of classifiers. It was highlighted that a small sample size data can cause many problems in designing a pattern recognition system, and considerable training samples are needed if a complex classification rule with many features is being adopted. Yu and Liu (2003) reported that both size and dimensionality pose severe challenges to feature selection algorithms, and the feature selection study has emphasized handling numerous samples to address these challenges. Similar opinion was raised by Liu et al. (2002). Zhu et al. (2013) developed a novel Self-taught Dimensionality Reduction (STDR) approach to transfer external knowledge (or information) from freely available external data to the small-sized data. According to the experimental results at five datasets, the STDR outperforms the existing algorithms in terms of *k*-means clustering performance. However, *k*-means clustering, an unsupervised machine learning algorithm, makes inferences from datasets using only input vectors without label information, i.e., even a slight change of dataset may significantly affect the predictive performance of *k*-means clustering. Accordingly, it is questioned that how the feasibility of STDR will be in superior supervised algorithms (e.g. Random Forests and Support Vector Machines). Likewise, Kuncheva and Rodríguez (2018) stated that to obtain a more stable feature subset, the sample size should be up-regulated.

This study aimed to propose a novel approach (Feature Selection Based on Data Quality and Variable Training

Samples, QVT) that can fit wide feature selection methods with low-sample-size data. Besides, the following two questions were also discussed.

1. Does QVT manage to improve the performance of different feature selection methods?

2. Does QVT have the same impact over different classifiers?

To verify the feasibility of QVT, the experiment was performed using 20 benchmark datasets in different fields, three feature selection methods (Information Gain, IG; Boruta; Least Absolute Shrinkage and Selection Operator, Lasso) as well as three classifier models (Support Vector Machine, SVM; Naïve Bayes, NB; Logistic Regression, LR).

The rest of this article is organized as follows. In Section 2 the methodology is explained. In Section 3, the experimental results are reported and discussed. Lastly, in Section 4, the conclusions are drawn.

2 Methodology

The QVT refers to a two-phase hybrid approach. Because the performance of feature selection methods is affected by the quality of data and the number of samples, to improve the performance of feature selection with limited data, the first phase is to define the most typical samples of each class. In the second phase, feature selection starts from using the most typical samples, which was repeated with a steady increase in sample size until all of samples are used. In this process, the list of selected features would be kept updated.

In this study, Mutual Information (MI) was used as a measure of the amount of information that one random variable has about another variable that has two main properties. Consider $y = (y_1, ..., y_n)$ representing some *n* samples in class *Y*. In the first phase, MI between y_i and the other samples is computed, and the final value of y_i is defined as:

$$S(y_i) = \sum_{j=1, j\neq i}^{n} \mathrm{MI}(y_i, y_j)$$

Subsequently, y_i is ranked. The sample achieving the highest final value is on the top of the ranking, which is considered a plausible *Y*. However, the sample achieving the lowest final value holds the lowest rank position and it might be an outlier.

The second phase is a repeated process. Hypothesize the numbers of samples of each class are the same. In the first round, the top *k* samples of each class are used to perform the first-time feature selection ($k = 2n/3$), and the list of selected features with importance score is defined as F_1. Next, the top *k*+2 samples of each class are used to perform the second-time feature selection, and the list of selected features with importance score is F_2. F_1 and F_2 are combined by revaluing features involved in both F_1 and F_2 with the average score. Features appear in either F_1 or F_2 will be recorded as they are. The combined feature list is defined as F_c. The third-time feature selection is performed with the top *k*+4 samples, and the list of selected features F_3 will be combined with F_c to obtain a new feature list. In such a way, two samples of each class are added each time to update both the list of selected features and their importance scores until all of samples are used. After the final feature list has been decided, features will be sorted by scores and the feature with the highest score ranks the top. The algorithm for QVT is shown in Table 1.

Table 1 The algorithm for QVT

1: Input data $D = \{(I_i, V_i)\}_{i=1}^{m}$, I_i is a sample in D, V_i is the set of features of I_i
2: For $i = 1$ to n do:
3: Compute $S(y_i) = \sum_{j=1, j\neq i}^{n} MI(y_i, y_j)$, n is the number of samples in class Y
4: Rank all of samples in descending order of S
5: Input subset s with the top k samples in rank of each class
6: Perform feature selection for s using method M, and the selected feature list is F_1
7: Let $size$= the number of samples in s
8: If $size \leq 2 \times (n - k)$:
9: if $k + 2 < n$
10: $k = k + 2$
11: else
12: $k = k + 1$
13: Update s with the top k samples in rank of each class
14: Perform feature selection for s, and the selected feature list is F_2
15: Let $F_c = \emptyset$
16: For each feature appears in both F_1 and F_2
17: Compute the average importance score. Record the feature and the average score in F_c
18: For each feature appears in either F_1 or F_2
19: Record the importance score as it is. Put the feature and its score in F_c
20: $F_1 \leftarrow F_c$
21: $size \leftarrow$ the number of samples in s
22: Go step 8
23: Endif
24: Return the final list of selected features, which is sorted by improtance score

3 Experiments study

3.1 Data

The characteristics of the 20 datasets used here are listed in Table 2, which were taken from the UCI Machine Learning Repository[1] and openML[2].

To make low-sample-size data, first, 20, 30, 40, 50 samples were extracted from each class randomly. Second, 2/3 samples were further extracted randomly to work as training data, and the rest samples were adopted as test data, which was repeated ten times. In such a way, after each random sampling the numbers of training data and test data of each class are 13, 20, 27, 33 and 7, 10, 13, 17, respectively.

[1] https://archive.ics.uci.edu/
[2] https://www.openml.org/

Table 2 Characteristics of 20 benchmark datasets. #Features is the number of features; #Samples is the number of samples; #Class 1 and #Class 2 are the numbers of samples in group 1 and group 2, respectively.

Dataset	#Features	#Samples	#Class1	#Class2
Bioreponse	1,776	3,751	2,034	1,717
Calibri	400	19,068	9,532	9,536
Christine	1,636	5,418	2,709	2,709
Eating	2,000	280	140	140
Fashion_minst	784	14,000	7,000	7,000
Gina_diagnostic	970	3,468	1,763	1,705
Halloffame	14	1,283	1,215	68
Har	561	3,266	1,722	1,544
Lonosphere	34	351	126	225
Kc1	21	2,109	1,783	326
Madelon	500	2,000	1,000	1,000
Musk	166	6,598	5,581	1,017
Ozone	72	2,534	2,374	160
Qsar	41	1,055	699	356
Semeion	256	319	161	158
Software	21	1,109	1,032	77
Spambase	57	4,601	2,788	1,813
Speech	400	3,686	61	3,625
Steel plats fault	33	1,941	1,268	673
Waveform	40	3,345	1,692	1,653

3.2 Feature selection methods, classifiers and performance metrics

In this study, three feature selection methods (IG, Boruta and Lasso) and three classifiers (SVM, NB and LR) were adopted to verify the feasibility of QVT.

First, the selected features were learned by classifiers. Subsequently, the performance of classifiers was evaluated by two metrics: macro-F (macro-averaged F-measure) and AUC (Area Under the ROC curve).

3.3 Experiments

Five steps were performed.

1. 2/3 samples were extracted from each class randomly to work as training data, and the rest samples were used as test data.

2. Boruta, QVT(B), IG, QVT(I), Lasso and QVT(L) were adopted to perform feature selection for training data, respectively. Subsequently, the selected features were sorted according to the importance score. The most useful feature for classification was presented on the top of rank.

3. Increasing one by one from two features to train classifiers (SVM, NB and LR). Then, test data was predicted, and macro-F and AUC were computed each time.

4. Six measures were adopted to evaluate the validity of feature selection methods, which are explained as follows.

- The lowest value of performance metric (Min.) and the greatest value of performance metric (Max.).
- Because after implementing QVT, the number of selected features might be different, the other two measures are the performance metric using the same number of selected features (Ave.1) and the performance metric using all selected features (Ave.2).
- The macro-F and AUC of X and QVT(X) were compared after features were increased each time. Lastly, the win times of X and QVT(X) were counted (#Win). Furthermore, the average number of used top features (mRank) when X or QVT(X) wins was computed.

5. After step1~step4 were performed ten times, the average of Min., Max., Ave.1, Ave.2, #Win and mRank were taken and considered as the final evaluation measure of X and QVT(X).

3.4 Experiment results

In this section, the results are presented separately to answer the two mentioned questions.

3.4.1 Does QVT manage to improve the performance of different feature selection methods?

The macro-F of SVM with 13 training samples using features selected by Boruta and QVT(B) is listed in Table 3. The average value was used to help explain results of this study, and the standard deviation (SD) is shown to present further detailed results. For Boruta and QVT(B) on average, the Min. were 0.61 and 0.67, the Max. 0.77 and 0.87, the Ave.1 0.70 and 0.76, the Ave.2 0.70 and 0.83, the #Win 1.15 and 3, and the mRank 11.99 and 6.63. Furthermore, according to the average values, the times of Win/Loss/Tie (W/L/T) of QVT(B) were 6/0/0 with six voters (evaluation measures) in total. Besides, the average times of W/L/T of QVT(B) across 20 datasets were 4.38/0.62/1. In the case of IG and Lasso, the results were similar to those achieved using Boruta.

The average of results of 20 datasets when 13 training samples were used is listed in Table 4. For QVT(B) judged using six evaluation measures, the times of win were 6, 6, 4, 6, 6 and 4, while those of loss were 0, 0, 2, 0, 0 and 2, and no ties happened. In the case of QVT(I), the times of win were 5, 6, 4, 6, 5 and 4, while those of loss were 1, 0, 2, 0, 1 and 2, and no ties happened; In the case of QVT(L), the times of win were 6, 5, 5, 6, 5 and 5, while those of loss were 0, 1, 1, 0, 1 and 1, and no ties happened. Furthermore, considering average times of W/L/T, the number of win was significantly higher than that of Loss. Similar results were obtained when 20, 27 and 33 training samples were used.

Because the performance becomes greater after QVT implemented than explicit feature selection method, as a

conclusion, QVT can enhance the performance of different feature selection methods.

3.4.2 Does QVT have the same impact over different classifier models?

Because the limitation of pages, the detail results are not shown in this article. According to the results, the performance of all classifiers was improved after implementing QVT compared with that using explicit feature selection method. In conclusion, QVT has the similar effect on different classifiers.

Furthermore, statistical analysis was based on welch's *t*-test. All Max. of QVT(X) were greater than X and there was significant difference, suggesting QVT(X) can always select more effective feature subset than X.

4 Conclusions

In this study, QVT was proposed to improve the performance of feature selection methods with low-sample-size data. According to the experiment results: (1) the feature selection methods fit a classification problem with less than 33 training samples; (2) a smaller number of training samples led to a more significant difference between QVT and the explicit feature selection method, and QVT was verified as the better one; (3) QVT fits different feature selection methods, and it can significantly improve the predictive performance of different classifiers.

The reason why QVT works is associated with the consideration of both the quality of data and the size of data. First, several typical samples of each class were extracted to lay a relatively reliable base. Subsequently, the number of training samples gradually increased, and the list of selected features was updated. In such a way, the same effect as repeated learning with different training data was obtained.

Reference

[1] Kuncheva, L. and Rodríguez, J.: Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning, Neurocomputing, 315, pp. 128-134, 2018.

Table 3 Macro-averaged F-measure of SVM with 13 training samples using features selected by Boruta and QVT(B).

	SVM														
	Min.		Max.		Ave.1		Ave.2		#Win		mRank				
	Boruta	QVT(B)	Boruta	QVT(B)	Boruta	QVT(B)	Boruta	QVT(B)	Boruta	QVT(B)	Boruta	QVT(B)	Win	Loss	Tie
Bioreponse	0.3636	0.4000	0.4000	0.6050	0.3879	0.4818	0.3879	0.5982	0	2	NA	2.5	6	0	0
Calibri	0.1667	0.2000	0.6667	0.7692	0.3086	0.5093	0.3086	0.5167	1	8	10	6.5	6	0	0
Christine	0.3636	0.3636	0.3636	0.8571	0.3636	0.3818	0.3636	0.6499	0	1	NA	3	5	0	1
Eating	0.2000	0.4615	0.6154	0.8571	0.5188	0.6809	0.5188	0.7153	1	6	7	5.83	6	0	0
Fashion_minst	0.8000	0.8000	0.9231	0.9231	0.8429	0.8428	0.8429	0.8479	8	9	47.25	40.33	3	1	2
Gina_diagnostic	0.5714	0.7692	0.7692	1.0000	0.6965	0.8339	0.6965	0.8416	0	7	NA	5	6	0	0
Halloffame	0.7500	0.7500	0.8571	0.9333	0.8223	0.8185	0.8173	0.8020	2	1	6.5	2	2	3	1
Har	0.6667	0.8000	1.0000	1.0000	0.8916	0.9050	0.8916	0.9208	2	4	13	11.5	5	0	1
Lonosphere	0.6667	0.8235	0.8571	0.8571	0.8142	0.8403	0.8142	0.8471	0	4	NA	4	5	0	1
Kc1	0.6154	0.7500	0.7500	0.8000	0.6827	0.7500	0.6827	0.7733	0	1	NA	3	6	0	0
Madelon	0.4706	0.5556	0.4706	0.6667	0.4706	0.6667	0.4706	0.6203	0	1	NA	2	6	0	0
Musk	0.7143	0.7143	0.8750	0.9333	0.8301	0.8433	0.8301	0.8706	0	2	NA	3.5	5	0	1
Ozone	0.6667	0.7273	0.8333	0.9231	0.7448	0.8136	0.7448	0.8169	0	4	NA	3.5	6	0	0
Qsar	0.6250	0.6250	0.7500	0.8000	0.7020	0.6962	0.7020	0.7294	2	1	7.5	5	3	2	1
Semeion	0.7692	0.8333	1.0000	1.0000	0.9212	0.9218	0.9212	0.9327	3	2	15.67	6.5	4	1	1
Software	0.8235	0.8235	0.8235	0.8235	0.8235	0.8235	0.8235	0.8235	0	0	NA	NA	0	0	6
Spambase	0.7500	0.8235	0.9333	1.0000	0.8879	0.9257	0.8879	0.9527	2	3	6	4.33	6	0	0
Speech	0.7143	0.7143	0.8571	0.9333	0.7648	0.8162	0.7648	0.8181	1	4	3	4.25	4	1	1
Steel plats fault	0.6667	0.6667	0.7143	0.8235	0.6825	0.6667	0.6825	0.7007	1	0	4	NA	2	3	1
Waveform	0.9333	0.8571	0.9333	0.9333	0.9333	0.9333	0.9333	0.9181	0	0	NA	NA	0	2	4
Average	**0.6149**	**0.6729**	**0.7696**	**0.8719**	**0.7045**	**0.7576**	**0.7043**	**0.8348**	**1.1500**	**3**	**11.9917**	**6.6324**	**6**	**0**	**0**
SD	**0.1990**	**0.1792**	**0.1808**	**0.1062**	**0.1886**	**0.1508**	**0.1885**	**0.2065**	**1.8241**	**2.6458**	**12.3187**	**8.7088**	**1.9519**	**1.0137**	**1.4654**
Average													**4.3810**	**0.6190**	**1**

Table 4 The average of results of 20 datasets when 13 training samples were used.

			Min.		Max. ***		Ave.1*		Ave.2**		#Win†		mRank		Average times of W/L/T					
			X	QVT(X)	X	QVT(X)	X	QVT(X)	X	QVT(X)	X	QVT(X)	X	QVT(X)	Win	Loss	Tie	Win	Loss	Tie
Boruta	macro-F	SVM	0.61	**0.67**	0.77	**0.87**	0.70	**0.76**	0.70	**0.83**	1.15	**3.00**	11.99	**6.63**	6	0	0	4.38	0.62	1.00
		NB	0.63	**0.64**	0.77	**0.83**	0.70	**0.72**	0.70	**0.74**	1.30	**2.00**	5.88	**5.52**	6	0	0	3.00	1.71	1.29
		LR	0.58	0.58	0.78	**0.82**	0.67	**0.70**	0.67	**0.70**	1.45	**6.30**	**6.24**	8.47	4	2	0	3.24	1.67	1.10
	AUC	SVM	0.71	**0.73**	0.82	**0.90**	0.78	**0.82**	0.78	**0.82**	1.35	**2.95**	8.75	**6.23**	6	0	0	3.52	1.38	1.10
		NB	0.75	**0.78**	0.83	**0.90**	0.79	**0.83**	0.79	**0.84**	1.00	**3.15**	7.33	**7.26**	6	0	0	3.86	1.05	1.10
		LR	**0.67**	0.63	0.84	**0.87**	0.75	**0.77**	0.75	**0.76**	2.20	**6.05**	**6.85**	7.81	4	2	0	2.81	1.95	1.24
IG	macro-F	SVM	0.61	**0.63**	0.73	**0.86**	0.69	**0.72**	0.69	**0.75**	**16.80**	10.55	30.50	**20.70**	5	1	0	3.43	1.48	1.10
		NB	0.61	**0.62**	0.70	**0.82**	0.66	**0.70**	0.66	**0.72**	9.00	**19.05**	27.92	**22.62**	6	0	0	4.05	0.86	1.10
		LR	**0.58**	0.56	0.76	**0.82**	0.65	**0.67**	0.65	**0.69**	3.10	**30.85**	**7.53**	30.67	4	2	0	3.10	1.86	1.05
	AUC	SVM	0.71	**0.73**	0.77	**0.87**	0.75	**0.78**	0.75	**0.81**	6.15	**7.10**	35.69	**10.12**	6	0	0	4.29	0.76	0.95
		NB	**0.71**	0.67	0.78	**0.84**	0.75	**0.76**	0.75	**0.77**	10.35	**11.20**	31.62	**17.22**	5	1	0	3.62	1.52	0.86
		LR	**0.60**	0.58	0.77	**0.85**	0.66	**0.68**	0.66	**0.69**	2.40	**31.35**	**8.68**	26.66	4	2	0	3.38	1.57	1.05
Lasso	macro-F	SVM	0.58	**0.63**	0.70	**0.80**	0.64	**0.72**	0.64	**0.72**	0.55	**2.75**	5.80	**3.74**	6	0	0	4.81	0.76	0.43
		NB	0.60	0.60	0.70	**0.78**	0.66	**0.70**	0.66	**0.70**	0.90	**1.70**	4.98	**3.81**	5	1	0	3.52	1.38	1.10
		LR	0.54	0.54	0.66	**0.79**	0.60	**0.67**	0.60	**0.66**	0.75	**2.40**	4.53	**3.97**	5	1	0	4.05	1.24	0.71
	AUC	SVM	0.56	**0.64**	0.70	**0.85**	0.64	**0.75**	0.64	**0.75**	0.45	**2.80**	4.14	**3.87**	6	0	0	4.67	0.95	0.38
		NB	0.63	**0.65**	0.71	**0.82**	0.68	**0.73**	0.68	**0.74**	0.60	**2.45**	**3.85**	3.97	5	1	0	4.00	1.00	1.00
		LR	**0.58**	0.57	0.69	**0.82**	0.64	**0.73**	0.64	**0.70**	0.65	**2.55**	4.14	**4.02**	5	1	0	4.43	1.10	0.48

*** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$, † $p < 0.1$