Data Reduction: Using Principle Component Analysis Approach

Xue Yang
West Virginia University
xuyang@mix.wvu.edu

ABSTRACT

This paper explores an effective tool to deal with large and complex data sets and also achieve users engagement in the data analysis process that they can get insights of big raw data through a representative summarization of them.

Specifically, in order to to simplify the training data the approach mainly combines two data reduction techniques: feature selection and instance reduction. Initially, the algorithm removes specific percentage of less significant attributes measured by information gain. Then Principle Component Analysis(PCA) method is applied to reduce data dimensionality. And through the grid-clustering process, the instances can be replaced by a small number of clusters' centroids. In addition, a modified K Near Neighbour(KNN) approach extrapolates between centroids to predict the class of testing cases.

This data reduction approach has been tested on 10 data sets from PROMISE repository[1] for "bug-defective" prediction. The results reveal that via data reduction procedure about 93% of instances and 98% of cells for each data set can be removed with no information loss. To measure its predicative capability, it has been compared with several widely used data mining methods such as Naive Bayes (NB) and KNN. In 7 of 10 given data sets, this tool can perform as well as other learners.

Keywords

Data MiningData Reduction, PCA Projection, Feature Selection

1. INTRODUCTION

Data mining is applied to automatically discover valuable information from raw data. Nowadays, with the significant improvement of computational capability, data has become increasingly larger not only in rows (i.e. number of instances) but also in columns (i.e. number of features). According to previous reach, data with high dimensionality brings great difficulty in both machine learning and data mining procedures[11].Therefore, the main challenge is to deal with increasing big and complex data set by designing some "clever" algorithm to simplify raw data.

Moreover, most of the learning algorithms handle input data as "black-box" that make predictions according to some input, without providing concrete descriptions about how they reach the conclusions. Furthermore, most of time the content and structure of the output from the learning algorithms are difficult to understand or implement by the users. Therefore, how to generate a transparent predictive model is a key factor to help learners get full use of the model.

This paper explores an algorithm that simplifies raw training data by employing a combination of feature selection and instance reduction and present a representative summary of the training data that help users get insights from them. Furthermore, the classifier consisted of projected centroid table and modified KNN can be used to infer predictions from the condensed training data.

In general, this data reduction algorithm is composed of three main steps to generate a representative summary of the raw training data:

- *Feature Selection via Information Gain.* The procedure is to conduct column reduction by keeping a specific percentage of given attributes with the highest information gain.
- *Projection via Principle Component Analysis (PCA).* The learner projects the given instance space onto the two dimensions with the greatest variability using PCA method.
- *Grid-Clustering.* After the projection step, a new pair of (x,y) coordinate is generated for each instance. In order to achieve row reduction, grid-clustering is employed, which is to recursively partition the new coordinates by the median value of each projected dimension and stop when the cluster size falls below a specific pre-defined value. The results of grid-clustering is a condensed table that contains the centroids for all the clusters and represents all the input instances.

In order to test this data reduction approach, 10 Software

[1]https://code.google.com/p/promisedata/

Engineering data from PROMISE repository is employed for defect prediction. The number of instances contained within these collected data sets range from 100 to 800. And with the help of our designed data reduction approach, over 94% of instances and 75% (this percentage can be adjusted by users) of features can be pruned without losing significant information about data. Moreover, our experimental results illustrate this data reduction method can perform as well as several widely used learners such as Naive Bayes and K-Near Neighbors.

2. RELATED WORK

Data Reduction is an effective tool addressed in data mining research, which is often considered as an improvement in data preprocessing technique. It will leads to several potential advantages: reducing data storage, shortening training and utilizing times[4], simplifying classifiers and improving prediction performance in terms of speed as well as its accuracy[6]. A lot of data reduction methods were proposed. However, because of the complexity of data in real-world application, it seems difficult to build a general data reduction method and is till attracting a lot of attention from data mining area. Data reduction can be mainly composed of two aspects: removing a number of features and reducing the amount of instances. The first component is implemented via data dimensionality reduction techniques and the second one is done by resampling instances and keeping most of the information.

2.1 Dimensionality Reduction

Specifically, there are two major categories of data dimensionality reduction. *feature transform* methods refers to constructing new attributes with a linear or nonlinear transformation out of the original input feature space[23], while *feature selection* is to find several informative features from given attributes and discard irrelevant ones. In precious research about feature transform, principle component analysis (PCA)[25], independent component analysis (ICA)[15] and linear discriminant analysis (LDA)[26] are used to compose a nonlinear mapping from input attribute space to a reduced dimensionality. However, there exists a major drawback that the constructed features do not have true meaning and complex computation might be needed[24].

Particularly, finding an appropriate and effective evaluation function for features is a critical component in the feature selection procedure that aims to extract a specific number of the most informative attributes,. Generally speaking, distance measures[19][17], information measures[8][18], correlation coefficient measures[10][9] and consistency measures[5][22] can be employed as an effective tool to assess given attributes. In this experiment, feature selection via information gain assessment is contained in our designed data reduction procedure.

2.2 Instance Reduction

In terms of instance reduction, related work has shown that collected data sets are built up with a lot of irrelevant and redundant instances and most of the information in the training data is contained in a small subset of the data. Subtracting these less informative parts of data may effective improve the performance of the learned model. A variety of useful instance reduction methods have been proposed. Datta and Kibler introduced the learner that finds the representative instances in each partition of dividing data and proposes a symbolic nearest mean classifier using k-means clustering to group instances of the same class[7].Wai Lam's Prototype Generation Filtering (PGF) algorithm operates by combining nearest instances and calculating the distance of each instance[13]. J.S. Sanchez's reduction learner conducts space partitioning (RSP) algorithms diving the training set into several subsets based on its diameter which is defined as the distance between its two farthest instances[21]. However, in literature no single instance-reduction approach is guaranteed superior to others and to output satisfactory results.

The data reduction proposed in this paper follows the similar procedures of PEEKING2 learner introduced by Papakroni[16]. In PEEKING2 learning algorithm, the instance space is initially projected in the directions of the greatest variability via FASTMAP heuristic, which is a linear approximation of PCA that projects given data on two new dimensions with the greatest variability. Next, the clustering algorithm recursively partitions the instance space by the median values of the dimensions found by FASTMAP until the density of the resulted clusters fall below a pre-defined minimum cluster size.

In this paper, a full version of PCA data project approach is implemented instead of using its linear approximation, FASTMAP projection.

3. HYPOTHESES

In this experiment, in order to illustrate the designed data reduction leaner's superiority on large Software Engineering (SE) data sets, several critical hypotheses listed as following are mainly addressed:

- *Conducting feature selection procedure will case no information loss.*
- *Data reduction algorithm will significantly reduce the size of training data set.*
- *The designed classifier with data reduction will perform better or at least as well as other elaborate learners.*

4. EXPERIMENT DESCRIPTION

This section presents the implementation and application of the designed data reduction algorithms on Software Engineering for defect prediction. The whole procedures about this experiment is displayed in Figure 4.1. More details about this experiment are explained below.

4.1 Data Sets

This data reduction learner presented in this paper has been applied on 10 data sets collected from PROMISE repository. All the data sets are composed of the information extracted from the specific modules of several development projects and for each set there are 20 different metrics corresponding to a specific software. In addition, a binary class indicates whether this module has been defective in any bug-tracking system. Therefore, the data can be used to train the learner in order to make automatic defect prediction for any incoming testing instances. Table 1 illustrates a detailed description about the 10 data sets considered in this study.

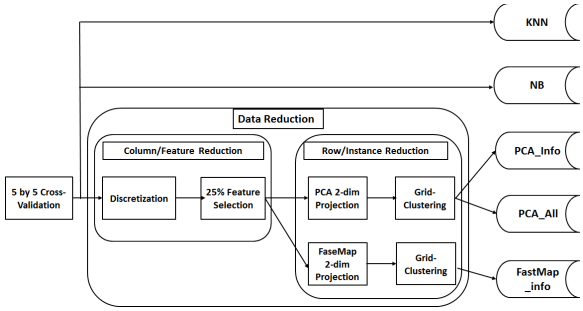

Figure 1: Major procedures in this experiment

Table 1: Details about 10 SE data sets

Data Set Name	# Instances	# Defects	% Defects
ant-1.7	745	166	22
ivy-1.1	111	63	57
jedit-4.1	312	79	25
log4j-1.1	109	37	34
lucenne-2.4	340	203	60
poi-3.0	442	281	64
synapse-1.2	256	86	34
velocity-1.6	229	78	34
xalan-2.6	885	411	46
xerces-1.4	588	437	74

In order to compare the designed data reduction algorithm with PEEKING2 results, I employed the same data sets as those used in PEEKING2 experiment[16].

4.2 Data Reduction

The learning algorithm presented in this paper is designed to simplify large and complex data sets. The designed learner applied feature selection/column reduction and data projection techniques to remove redundant information and generate a small representative summary of data. There are four major steps executed in this data reduction approach: (1) *Discretization*; (2) *Feature Selection via Information Gain*; (3) *Projection via PCA*; and (4) *Grid-Clustering*. The rest of this section explains these four operators in details.

4.2.1 Discretization

Since all the features in the data sets are numeric types and in the next feature selection step the computation of entropy is required, the manipulation of discretization is quite necessary. According to the range and intervals of numbers from each column, the unsupervised discretization approach, the Gaussian Chops, is implemented.

The Gaussian chops is basically assumed that each column of numbers comes from a Gaussian distribution. And all the number from same column need to be normalized by subtracting the mean and dividing by the standard deviation, and the two major factors can be calculated from each column. Finally, all cells need to divide the specific values according to the desired number of bins using breaks take from a referenced table. In this study, a fixed bin size of 10 is used.

4.2.2 Feature Selection via Information Gain

The selected attributes play a dominant role in any data mining problems. And instance-based learning algorithms are particularly sensitive to irrelevant features. For instance, the K Nearest Neighbour method computes the sum of absolute differences between all features of two compared instances, which means that the irrelevant and relevant attributes make the same contribution to the classification process. Therefore, in this data reduction study, feature selection is crucial to the success of the experiment.

This data reduction algorithm assess features based on their relevance to the class variable using *Information Gain (IG)*, which is widely used in several learners such as decision tree method. *Entropy Reduction* for a given attribute can define IG more specific, which is to measure the change of entropy for the class feature by removing each attribute. The equations of calculating entropy and IG for a given feature are listed below:

$$H(D) = \sum_{n \epsilon Class} f_c \log f_c \tag{1}$$

$$InfoGain(D, A) = H(D) - \sum_{v \epsilon A} \frac{|D_{A=v}|}{D} H(D_{A=v}) \tag{2}$$

where *H(D)* is the entropy computation, D is the set of training set, f_c is the computed frequency of input data with class label equal to c, *Class* is the list containing all potential class values, $D_{A=v}$ is the subset of training instances with attribute A equals to specific value v, and A is a list of all possible feature values for a given independent attribute.

After ranking all features in terms of their corresponding IG, a specific percentage of features are selected with the *highest Information Gain*. And the percentage can be adjusted by users. In our implement, 25% of attributes are selected (the number 25% is an Engineering instinct). Consequently, from 20 input features given in the data sets, 5 of them containing most information are kept. Table 2 presents an example of resulted 5 feature via using feature selection on POI-3.0 data set.

Table 2: Example of Feature Selection applied on POI-3.0 data set. The listed attributes are the 25% of the given features ranked in decreasing order of their information gain. The total entropy of the entire data set is about 0.95.

Attribute Names	Entropy	IG
lcom3	0.687	0.263
cbm	0.711	0.239
rfc	0.728	0.222
max_cc	0.732	0.218
wmc	0.733	0.217

4.2.3 Feature Projection via PCA

Principal component analysis (PCA) approach is to find a subspace with basis vectors correspond to the maximum-variance directions in the original space. Figure 2, generally describes how PCA projects given data on two new dimensions[2], which is the same number of dimensionality used in PEEKING2 experiment.

[2]http://ufldl.stanford.edu/wiki/index.php/PCA

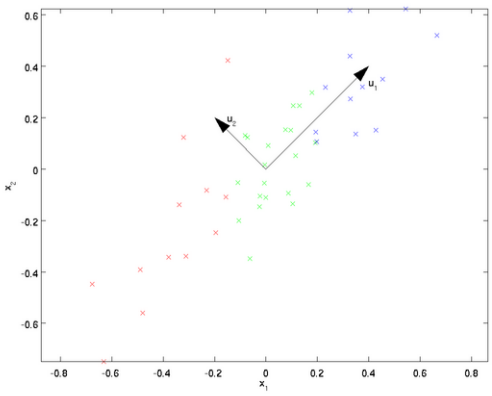

Figure 2: Projection of the instance space via PCA.

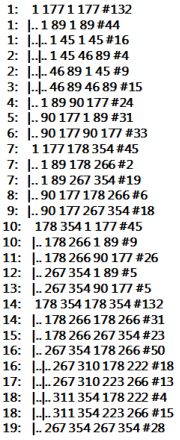

Figure 3: Example of a nested clusters tree generated from applying grid-clustering on POI-3.0 data set.

The algorithm initially generates all the eigenvectors that represent the given instance space and selects two of them with the highest absolute eigenvalues. Then consists the transformation matrix using these two eigenvectors that maps the original data set with 20 features on 2 dimensions, the generated new coordinates *(x,y)* for each instance. And the following equations illustrate the PCA computation procedures.

$$Q = XX^T \tag{3}$$

$$\lambda_i e_i = Qe_i \tag{4}$$

$$Y = XW \tag{5}$$

where X is the training instance matrix, Q is covariance matrix, λ_i and e_i are the resulted eigenvalue and its corresponding eigenvector, W is the transformation matrix consisted of t selected eigenvectors, t is the number of target projected dimensionality.

Figure 4 illustrate the application of PCA projection on POI-3.0 data set. The pictures on top row display the PCA projection applied on both the original 20-dimensional data (left) and the resulted 5-dimensional data after conducting 25% feature selection (right). Each point is a representation for a particular software module.

4.2.4 Grid-Clustering

After PCA projection, the new coordinate is generated for each instance. Then a grid-clustering algorithm is applied on these new (x, y) pairs to partition the instance space into a set of disjoint quadrants. Generally speaking, grid-clustering is a hierarchical clustering algorithm that recursively splits large clusters into smaller ones until the size of clusters are under some pre-defined value. Consequently, each constructed cluster is replaced by its centroids, which is composed of the mean feature values of all cases included in that cluster, and the resulted centroid-table is the representative summary of all the training data and can be used for future prediction procedure. Finally, a continuous class named *defect_rate* is assigned to each centroid instead of using previous binary class label. This class is defined as the average rate of defective modules in the corresponding cluster, which ranges from 0 to 1.

The output of hierarchical clustering method is a tree of nested clusters shown in Figure 3. Moreover, Figure 4 presents the visualization of the data reduction output on POI-3.0 data set.

The pre-defined minimum allowed cluster size controls the level of instance reduction procedure applied on the training data. In general, with the decrement of minimum cluster size, more clusters will be generated, then less data will be removed. In this experiment, only one level of instance reduction is tested with the minimum cluster size set to $2*\sqrt{n}$, where n is the number of cases in that data set.

4.3 Prediction Method

In order to assist users analyse the condensed data and compare the effect of the data reduction method, instance-based learning approach is employed.

- *K Nearest Neighbor Classifier:* The modified KNN extrapolates between the k nearest centroids of a given centroid-table generated from applying the grid-clustering on a given projected data set in order to predict the module as "defective" or "non-defective".

4.3.1 K Nearest Neighbor Classifier

The modified KNN classifier classifies new software modules based on their similarity to the cluster centroids. Given a new instance, the learner initially finds its k closest neighbors with the smallest difference measured by *"Eclidean distance"* from the condensed centroid-table. Then the classifier estimates the class of the test case according to its *average defect_rate* of its k nearest centroids, which is defined by the Equation (6). The weights are designed that the nearest centroids will affect more in the prediction.

$$avg\_defect\_rate = \sum_{i=1}^{k} \frac{1/d_i}{\sum_j 1/d_i} r_i \tag{6}$$

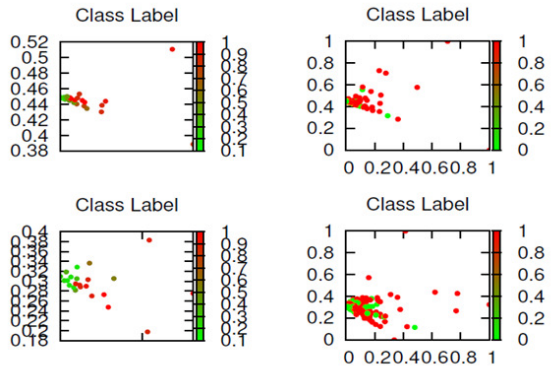

Figure 4: Example of feature selection, PCA projection, grid-clustering and centroids estimation applied on POI-3.0 data set that includes 442 instances. *Top Row:* PCA projection worked on the entire data set with all 20 features. *Bottom Row:* PCA projection applied on the entire data set with the selected 25% of high IG features. *Left Column:* Raw training data projected on 2 dimensions with the highest eigenvalue found by PCA. *Right Column:* Centroids mapped on 2 dimensions with the averaged projected coordinates from the instances contained in that cluster.

where d_i is the *Euclideandistance* of the i^th centroid from the test instances, and r_i is the defect rate of the i^th centroid. In this experiment, the classifier only analysed k=2 Nearest Neighbor.

The generated avg_defct_rate ranges from 0 to 1. And users can design a threshold to help classify the new instances. If the avg_defct_rate is greater than the specific threshold, this new case will be classified as "defective", otherwise it will be labelled as "non-defective". In this study, a threshold of 0.5 is applied.

5. RESULTS AND CONCLUSIONS

In this experiment, 5 by 5 cross-validation is applied and all the 5 learning algorithms work on the same training data set for each run time. This section describes the experimental results of comparing and conclusions.

5.1 Learning Algorithms

Moreover, in order to compare the accuracy of the designed data reduction classifier, several elaborate learning algorithms are also tested on the same training data.

- *PCA(info)*: Modified KNN (k=2) algorithm applies algorithm on entire data set with 25% selected features using InfoGain and projects via PCA.
- *PCA(all)*: Modified KNN (k=2) algorithm applies on data set with all features (k=2) and projects via PCA.
- *FASTMAP(info)*: Modified KNN (k=2) algorithm applies on data set with 25% selected features using InfoGain and projects via FASTMAP.
- *K Nearest Neighbor (k=5).*
- *Naive Bayes (NB).*

5.2 Performance Measurement

The following equations provide a list of measures from confusion matrix to assess the predictive performance of the learners. And in order to take into consideration both the recall and precision of the predictor, F-Measure is the key estimator in the analysis.

$$precision = \frac{TP}{TP + FP} \tag{7}$$

$$recall(pd) = \frac{TP}{TP + FN} \tag{8}$$

$$F = \frac{2 * pd * prec}{pd + prec} \tag{9}$$

where *TP, TN, FP, FN* are from the confusion matrix in Table 3.

Table 3: Confusion Matrix

		Actual	
		non-defective	defective
Predicted	non-defective	TN	FN
	defective	FP	TP

5.3 Data Reduction Results

Table 4 and Table 5 report the amount of data reduction applied PCA projection on both the data with all features and the data with 25% selected features. In terms of the median results, both versions reduced the original number of rows by over 93%. However, PCA(info) discards a larger amount of data due to the feature selection operator. In fact, PCA(inf) has reduced 98% of the data/cell for each data set. On the other hand, the median reduction of PCA(all) is 93%.

Table 4: PCA (all) Data Reduction Results for 10 SE Data Sets

Data Set	#Instances	%Centroids	%Row Reduction
ant-1.7	596	31	95
ivy-1.1	89	8	91
jedit-4.1	250	18	93
log4j-1.1	87	8	91
lucenne-2.4	272	19	93
poi-3.0	354	25	93
synapse-1.2	205	18	91
velocity-1.6	183	16	91
xalan-2.6	708	31	96
xerces-1.4	470	26	94
median	321	20	93

5.4 Predictive Performance

In terms of the classification performance, Table 6 reports the median F-Measure observed on 25 experimental run (5 times by 5 folders) for each leaning algorithm and data set. All the learners are ranked based on their F value.

The results are grouped according to the predictive performance of the PCA projection learner i comparison to the other learners:

Table 5: PCA (info) Data Reduction Results for 10 SE Data Sets

Data Set	#Instances	#Centroids	%Row Reduction	%Cell Reduction
ant-1.7	596	29	95	99
ivy-1.1	89	9	90	97
jedit-4.1	250	215	94	98
log4j-1.1	87	7	92	98
lucenne-2.4	93	16	94	99
poi-3.0	354	21	94	99
synapse-1.2	205	13	94	98
velocity-1.6	183	13	93	98
xalan-2.6	708	25	96	99
xerces-1.4	470	23	95	99
median	321	17	94	98

Table 6: Ranked predictive performance for each Leaner and data set.

Group1: Both versions of PCA projection perform well.			
POI-3.0	Learner	F	Rank
	KNN	0.83	1
	PCA(all)	0.82	1
	PCA(info)	0.79	1
	Fastmap(info)	0.77	1
	NB	0.49	2
IVY-1.1	Learner	F	Rank
	KNN	0.72	1
	Fastmap(info)	0.72	1
	PCA(info)	0.71	1
	PCA(all)	0.67	1
	NB	0.55	2

Group2: Both versions of PCA projection perform not well.			
VELOCITY-1.6	Learner	F	Rank
	KNN	0.58	1
	Fastmap(info)	0.50	2
	PCA(all)	0.41	3
	PCA(info)	0.40	3
	NB	0.35	3
LOG4J-1.1	Learner	F	Rank
	NB	0.70	1
	KNN	0.66	1
	PCA(info)	0.64	1
	PCA(all)	0.58	2
	Fastmap(info)	0.49	3
ANT-1.7	Learner	F	Rank
	NB	0.56	1
	KNN	0.52	1
	PCA(info)	0.46	2
	PCA(all)	0.43	3
	Fastmap(info)	0.36	4

Group3: PCA(info) performs better then PCA(all)			
xerces-1.4	Learner	F	Rank
	PCA(info)	0.94	1
	KNN	0.91	1
	Fastmap(info)	0.85	2
	PCA(all)	0.84	1
	NB	0.72	3
JEDIT-4.1	Learner	F	Rank
	KNN	0.55	1
	PCA(info)	0.54	1
	NB	0.53	1
	PCA(all)	0.51	1
	Fastmap(info)	0.40	2

Group4: PCA(all) performs better then PCA(info)			
LUCENE-2.4	Learner	F	Rank
	KNN	0.76	1
	Fastmap(info)	0.75	1
	PCA(all)	0.72	1
	PCA(info)	0.70	1
	NB	0.53	2
XALAN-2.6	Learner	F	Rank
	KNN	0.70	1
	PCA(all)	0.65	2
	Fastmap(info)	0.63	2
	PCA(info)	0.63	2
	NB	0.60	3
SYNAPSE-1.2	Learner	F	Rank
	KNN	0.60	1
	PCA(all)	0.57	1
	NB	0.57	1
	Fastmap(info)	0.50	2
	PCA(info)	0.47	3

- *Group1:* Both versions of PCA projection classifiers perform as well or better as NB and KNN.
- *Group2:* Both versions of PCA projection classifiers perform worse than NB and KNN.
- *Group1:* PCA(info) projection classifiers perform close to NB and KNN.
- *Group1:* PCA(all) projection classifiers perform close to NB and KNN.

Here the definition that an algorithm performs as "well" as another indicates that its median F-measure among 25 run times is at most 0.05 lower than that of other learners.

From the results we can see that in 7 of 10 data sets, at least one version of PCA projection can perform as well or even better than other learners and in 2 data sets both versions of PCA projection classifiers work well. However, in 3 data sets both PCA(info) and PCA(all) are not the optimal learning algorithms. Moreover, in most of the cases, the performance of FASTMAP(info) is quite similar to that of PCA(info), which means that using a linear approximation of PCA projection leads to no significant difference in the classification results.

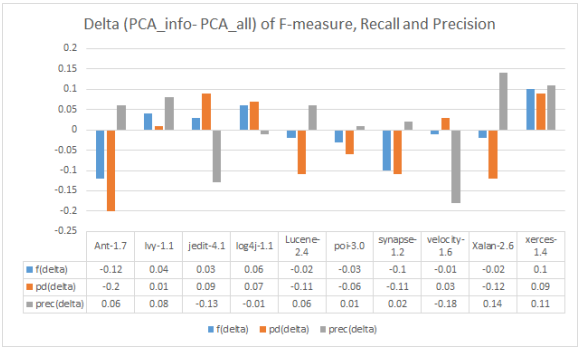

Figure 5: Delta values of F-Measure, recall and precision between PCA(info) and PCA(all).

5.4.1 PCA(info) vs. PCA(all)

The blocks of *F-measure* in Figure 5 shows little difference in the classification performance of two versions of PCA projection learner. In fact, feature selection via Information Gain shows no significant impact in the performance of the algorithm. Furthermore, it can be concluded that there is little information loss caused by data reduction.

And comparing the *recall* and *precision* scores we can figure out that the predictor increases one of these two measures at the cost of decreasing the other value. It also proves that F-measure is the most important measure of the analysis with both recall and precision taking into the computation.

In addition, the experimental results illustrate that PCA(info) can improve the *precision* value (in 7 of 10 data sets).

6. THREATS TO VALIDITY

This section discusses some of the potential validity threats of our experiment and final conclusions:

1. *Internal Validity:*

   In order to measure the data reduction results, the experiment compares the designed PCA projection classifier with K Nearest Neighbor and Naive Bayes. The selection of these two learning algorithms are based on their widely usage in constructing predictive model on SE data. However, this choice of classifiers may cause internal validity that other learners may lead to significantly different performance in the experiment. Also, the test is only applied on 10 SE data sets, which may be insufficient data provided to estimate the learning algorithm.

2. *External Validity:*

   The large external validity may exist with the selection of 10 SE data sets from PROMISE repository. It will affect the generalization ability of empirical studies for applying other types of raw data sets.

7. FUTURE WORK

The paper presents an effective data reduction approach to simplify the raw data without casing information loss and the designed classifier can perform as well as other widely used learning algorithm, such K Nearest Neighobr and Naive Bayes. However, there are still several research questions left to be investigate in the future:

- *Employ different dirscrtization approaches and corresponding bin sizes.*
- *How to decide the percentage of features selected from the raw features?*
- *How to decide the minimum allowed cluster size used in the grid-clustering procedure?*
- *Try different number of dimensionality in the projection step.*

8. REFERENCES

[1] M. Bowman, S. K. Debray, and L. L. Peterson. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.

[2] J. Braams. Babel, a multilingual style-option system for use with latex's standard document styles. *TUGboat*, 12(2):291–301, June 1991.

[3] M. Clark. Post congress tristesse. In *TeX90 Conference Proceedings*, pages 84–89. TeX Users Group, March 1991.

[4] I. Czarnowski and P. Jedrzejowicz. Instance reduction approach to machine learning and multi-database mining. In *Proceedings of the Scientific Session organized during XXI Fall Meeting of the Polish Information Processing Society, Informatica, ANNALES Universitatis Mariae Curie-Skłodowska, Lublin*, pages 60–71. Citeseer, 2006.

[5] M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial intelligence*, 151(1):155–176, 2003.

[6] M. Dash, H. Liu, and H. Motoda. Consistency based feature selection.

[7] P. Datta and D. Kibler. Symbolic nearest mean classifiers. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 82–87. Morgan Kaufmann Publishers, 1997.

[8] J. Grande, M. del Rosario Suárez, and J. R. Villar. A feature selection method using a fuzzy mutual information measure. In *Innovations in Hybrid Intelligent Systems*, pages 56–63. Springer, 2007.

[9] M. Haindl, P. Somol, D. Ververidis, and C. Kotropoulos. Feature selection based on mutual correlation. In *Progress in Pattern Recognition, Image Analysis and Applications*, pages 569–577. Springer, 2006.

[10] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[11] D. J. Hand, P. Smyth, and H. Mannila. *Principles of Data Mining*. MIT Press, Cambridge, MA, USA, 2001.

[12] M. Herlihy. A methodology for implementing highly concurrent data objects. *ACM Trans. Program. Lang. Syst.*, 15(5):745–770, November 1993.

[13] W. Lam, C.-K. Keung, and C. X. Ling. Learning good prototypes for classification using filtering and abstraction of instances. *Pattern Recognition*, 35(7):1491–1506, 2002.

[14] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[15] T.-W. Lee. *Independent component analysis*. Springer, 1998.

[16] V. Papakroni. *Data Carving: Identifying and Removing Irrelevancies in the Data*. PhD thesis, West Virginia University, 2013.

[17] N. Parthalain, Q. Shen, and R. Jensen. A distance measure approach to exploring the rough set boundary region for attribute reduction. *Knowledge and Data Engineering, IEEE Transactions on*, 22(3):305–317, 2010.

[18] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, 2005.

[19] S. Piramuthu. The hausdorff distance measure for feature selection in learning applications. In *Systems Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, volume Track6, pages 6 pp.–, 1999.

[20] S. Salas and E. Hille. *Calculus: One and Several Variable*. John Wiley and Sons, New York, 1978.

[21] J. S. Sánchez. High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognition*, 37(7):1561–1564, 2004.

[22] K. Shin and X. M. Xu. Consistency-based feature selection. In *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 342–350. Springer, 2009.

[23] K. Torkkola. Feature extraction by non parametric mutual information maximization. *J. Mach. Learn. Res.*, 3:1415–1438, Mar. 2003.

[24] E. C. Tsang, D. S. Yeung, and X. Wang. Offss: optimal fuzzy-valued feature subset selection. *Fuzzy Systems, IEEE Transactions on*, 11(2):202–213, 2003.

[25] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1âĂŞ3):37 – 52, 1987. <ce:title>Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists</ce:title>.

[26] J. Ye, R. Janardan, and Q. Li. Two-dimensional linear discriminant analysis. In *Advances in Neural Information Processing Systems*, pages 1569–1576, 2004.