

# Ranking Normalization Methods for Improving the Accuracy of SVM Algorithm by DEA Method

Maysam Eftekhary<sup>1</sup>, Peyman Gholami<sup>1</sup>, Saeed Safari<sup>1</sup> & Mohammad Shojaee<sup>2</sup>

<sup>1</sup> Department of Industrial Engineering, Arak Branch, Islamic Azad University, Arak, Iran

<sup>2</sup> Governmental Management, Arak, Iran

Correspondence: Peyman Gholami, Department of Industrial Engineering, Arak Branch, Islamic Azad University, Arak, Iran. Tel: 98-937-138-9131. E-mail: peyman711@gmail.com

Received: December 19, 2011

Accepted: September 20, 2012

Online Published: September 26, 2012

doi:10.5539/mas.v6n10p26

URL: <http://dx.doi.org/10.5539/mas.v6n10p26>

## Abstract

Data mining techniques, extracting patterns from large databases have become widespread in all life's aspect. One of the most important data mining tasks is classification. Classification is an important and widely studied topic in many disciplines, including statistics, artificial intelligent, operations research, computer science and data mining and knowledge discovery. One of the important things that should be done before using classification algorithms is preprocessing operations which cause to improve the accuracy of classification algorithms. Preprocessing operations include various methods that one of them is normalization. In this paper, we selected five applicable normalization methods and then we normalized selected data sets afterward we calculated the accuracy of classification algorithm before and after normalization. In this study the SVM algorithm was used in classification because this algorithm works based on n-dimension space and if the data sets become normalized the improvement of results will be expected. Eventually Data Envelopment Analysis (DEA) is used for ranking normalization methods. We have used four data sets in order to rank the normalization methods due to increase the accuracy then using DEA and AP-model outrank these methods.

**Keywords:** data mining, SVM, DEA, normalization methods, ranking, classification algorithm

## 1. Introduction

Data mining and knowledge discovery (DMKD) has made predominant progress during the past two decades (Peng et al., 2008). It utilizes methods, algorithms, and techniques from many disciplines, including statistics, datasets, machine learning, pattern recognition, artificial intelligence, data visualization, and optimization (Fayyad, 1996).

In recent years, the field of data mining has seen an explosion of interest from both academic and industry (e.g., Olafson, Li, & Wu, 2008). Increasing volume of data, increasing awareness of inadequacy of human brain to process data and increasing affordability of machine learning are reasons of growing popularity of data mining (e.g., Marakas, 2004). Data mining (DM) is the process for automatic discovery of high level knowledge by obtaining information from real data.

One of the major tasks in DMKD is classification. Researchers in a variety of fields have created a large number of classification algorithms, such as decision tree, neural networks, Bayesian network, linear logistic regression, Naive Bayes, and K-nearest-neighbor.

Learning algorithms are now used in many domains, and different performance metrics are suited for each domain. For example Precision/Recall measures are used in information retrieval; medicine prefers ROC area; Lift is appropriate for some marketing tasks, etc.

The different performance metrics measure different trade of in the predictions made by a classification and it is possible for learning methods to perform well on one metric, but be suboptimal on other metrics. Because of this it is to calculate algorithms on a broad set of performance metrics (Caruana & Niculescu-Mizil, 2006).

Classification is an important and widely studied topic in many disciplines, including statistics, artificial intelligent, operations research, computer science and data mining and knowledge discovery (Chen, Xu, & Chi, 1999).

Based on the number of predefined groups, classification can be divided into binary and multiclass classification. Binary classification assigns data objects into one of the two groups and multiclass classification involves three or more groups. Compare with binary classification, multiclass classification problem is more complex.

With increased use in real-world applications, such as disease diagnosis, text categorization, credit analysis, software risk management and network intrusion detection, a variety of methods and algorithms have been developed for multiclass classification in recent years.

Chen et al. scrutinized the reason why the mixture of experts (ME) performance poorly in multiclass classification and proposed an approximation for the Newton-Raphson algorithm to improve the performance of the ME architecture in multiclass classification.

Platt et al. (1999) presented the decision directed acyclic graph architecture and a learning algorithm for multiclass classification. Allwein et al. (2000) proposed a unifying framework for multiclass classification using a margin-based binary learning algorithm.

Crammer and Singer (2001) described the algorithmic implementation of multiclass kernel-based vector machines and conducted experiments to compare the presented approach to previously studied kernel-based methods. Loucopoulos proposed a mixed-integer programming model for the minimization of misclassification costs in the three-group problem.

Thorsten Joachims (1997) published results on a set of binary text classification experiments using the Support Vector Machine. The SVM concluded lower error than many other classification techniques.

Yang and Liu (1995) followed two years later with experiments of their own on the same data set. They used improved versions of Naive Bayes (NB) and k-nearest neighbors (KNN) but still found that the SVM performed at least as well as all other classifiers they tried.

Nowadays normalization methods are very applicable and appropriate for solving the Multiple Criteria Decision Making (MCDM) problems. There are different methods for normalization and these methods used to obtain concise answers. The normalization methods extracted from (Hwang & Yoon, 1981; Milani, Shanian, & Madoliat, 2005; Yoon & Hwang, 1995) and are used in the experimental study.

Ranking of normalization methods normally need to examine several criteria, such as accuracy, computational time, and misclassification rate. Therefore algorithm selection can be modeled as multiple criteria decision making (MCDM) problems (Peng et al., 2009; Rokach & Ensemble, 2009).

As mentioned heretofore, algorithm ranking is a useful strategy for selecting the appropriate classifier and the preferences of users are important in algorithm ranking (Berrer, Paterson, & Keller, 2000). Some existing MCDM methods are able to rank classifiers based on multiple performance measures and take the preferences of users into the ranking process.

DEA is a non-parametric linear programming based technique for measuring the relative efficiency of a set of similar units, usually referred to as decision making units (DMUs). Because of its successful application, DEA has gained too much attention and vast use by business and academic researchers.

Evaluation of data warehouse operations (e.g., Mannino, Hong, & Choi, 2008), selection of flexible manufacturing system (e.g., Camanho & Dyson, 2005) assessment of bank branch performance (e.g., Chen, Skully, & Brown, 2005) examining bank efficiency (e.g., Edirisinghe & Zhang, 2007), analyzing firm's financial statements (e.g., Johnes, 2006) measuring the efficiency of higher education (by solving only one LP) institutions (e.g., Ertay, Ruan, & Tuzkaya, 2006), solving facility layout design (FLD) problem (e.g., Shafer & Byrd, 2006) and measuring the efficiency of organizational investments in information technology (e.g., Vladimir Vapnik, 1995) are examples of using DEA in different areas.

The rest of this paper is organized as follows. In Section II normalization, SVM and DEA are described. The design of the experimental study is provided in Section III. Section IV presents and analyzes the experimental results. The Section V summarizes the findings and discusses future research directions.

## 2. Preliminaries

### 2.1 Normalization

The normalization methods and distance measures are also taken into consideration as well. Some of them are very useable that we selected 5 popular methods with study literature that such as:

- 1) Vector normalization:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (1)$$

2) Linear normalization (I):

$$r_{ij} = \frac{x_{ij}}{x_j^*} \quad (2)$$

$$i = 1, \dots, m; j = 1, \dots, n; x_j^* = \max\{x_{ij}\}$$

3) Linear normalization (II):

$$r_{ij} = \frac{x_{ij} - x_j^-}{x_j^* - x_j^-} \quad (3)$$

$$i = 1, \dots, m; j = 1, \dots, n; x_j^- = \min\{x_{ij}\}$$

4) Linear normalization (III):

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^m x_{ij}} \quad (4)$$

$$i = 1, \dots, m; j = 1, \dots, n$$

5) Non-monotonic normalization:

$$e^{-\frac{z^2}{2}}, z = \frac{(x_{ij} - x_j^0)}{\sigma_j} \quad (5)$$

$x_j$  is the most favorable value and  $\sigma_j$  is the standard deviation of alternative ratings with respect to the  $j$ th attribute (Hwang & Yoon, 1981; Milani, Shanian, & Madoliat, 2005; Yoon & Hwang, 1995).

## 2.2 Support Vector Machine (SVM)

The Support Vector Machine is a classifier, originally proposed by Vapnik that finds a maximal margin separating hyper plane between two classes of data (e.g., Christopher, 1998). An SVM is trained via the following optimization problem:

$$\text{Argmin}_{\omega} \frac{1}{2} \omega^2 + \mathcal{C} \sum_i \varepsilon_i \quad (6)$$

with constraints:

$$y_i (X_i \cdot \omega + b) \geq 1 - \varepsilon_i \forall i \quad (7)$$

For more information, see Burges' tutorial and Cristianini and Shawe-Taylor's book (e.g., Nello & John, 2000; Yang & Liu, 1999). There are non-linear extensions to the SVM, but Yang and Liu found the linear kernel to outperform non-linear kernels in text classification (e.g., Ryan, 2000).

In informal experiments, we also found that linear performs at least as well as non-linear kernels. Hence, we only present linear SVM results. We use the SMART ltc<sup>2</sup> transform; the SvmFu package is used for running experiments (e.g., Charnes, Cooper, & Rhodes, 1978).

The SVM must read in the training set and then perform a quadratic optimization. This can be done quickly when the number of training examples is small (e.g., <10000 documents), but can be a bottleneck on larger training sets. We realize speed improvements with chunking and by caching kernel values between the training of binary classifiers.

### 2.3 Data Envelopment Analysis (DEA)

Charnes, Cooper, and Rhodes developed data envelopment analysis (DEA) to evaluate the efficiency of decision making units (DMUs) through identifying the efficiency frontier and comparing each DMU with the frontier (Koksalan & Tuncer, 2009). Since DEA is able to estimate efficiency with minimal prior assumptions (Li & Ma, 2008; Cherchye & Post, 2003), it has a comparative advantage to approaches that require a priori assumptions, such as standard forms of statistical regression analysis (Cooper, 2004).

During the past thirty years, various DEA extensions and models have been developed and established themselves as powerful analytical tools (Banker, Charnes, & Cooper, 1984).

The original DEA model presented by Charnes, Cooper, and Rhodes (Koksalan & Tuncer, 2009) is called "CCR ratio model", which uses the ratio of outputs to inputs to measure the efficiency of DMUs. Assume that there are  $n$  DMUs with  $m$  inputs to produce  $s$  outputs.  $X_{ij}$  and  $y_{rj}$  represent the amount of input  $i$  and output  $r$  for  $DMU_j$  ( $j=1,2,\dots,n$ ), respectively. Then the ratio-form of DEA can be represented as:

$$\begin{aligned} \max h_o(u,v) &= \frac{\sum_r u_r y_{ro}}{\sum_i v_i x_{io}} \\ \text{subject to: } & \sum_r u_r y_{ro} / \sum_i v_i x_{io} \leq 1 \text{ for } j = 1, \dots, n \\ & u_r, v_i \geq 0 \text{ for all } i \text{ and } r \end{aligned} \quad (8)$$

where the  $u_r$ 's and the  $v_i$ 's are the variables and the  $y_{ro}$ 's and  $x_{io}$ 's are the observed output and input values of the DMU to be evaluated (i.e.,  $DMU_o$ ), respectively. The equivalent linear programming problem using the Charnes-Cooper transformation is (Banker, Charnes, & Cooper, 1984):

$$\begin{aligned} \max z &= \sum_{r=1}^s u_r y_{ro} \\ \text{subject to: } & \sum_{r=1}^s u_r y_{rj} - \sum_{i=1}^m v_i x_{ij} \leq 0 \\ & \sum_{i=1}^m v_i x_{ij} = 1 \\ & u_r, v_i \geq 0 \end{aligned} \quad (9)$$

Banker, Charnes, and Cooper introduced the BCC model by adding a constraint  $\sum \lambda_j = 1$  to the CCR model (Nakhaeizadeh & Schnabl, 1997). These models can be solved using the simplex method for each DMUs. DMUs with value of 1 are efficient and others are inefficient. Nakhaeizadeh and Schnabl proposed to use DEA approach in data mining algorithms selection (Fayyad, Piatetsky-Shapiro, & Smyth, 1996).

They argued that in order to make an objective evaluation of data mining algorithms that all the available positive and negative properties of algorithms are important and DEA models are able to take both aspects into consideration. Positive and negative properties of data mining algorithms can be considered as output and input components in DEA, respectively.

For example, the overall accuracy rate of a classification algorithm is an output component and the computation time of an algorithm is an input component. Using existing DEA models, it is possible to give a comprehensive evaluation of data mining algorithms.

In the empirical study, input components are 1. Output components include the accuracy SVM algorithm with use each normalization method and the DMU (Decision Making Unit) are the normalization method.

After solving this LP and determining the weights, the algorithms with  $h_r = 1$  (100%) are efficient algorithms and form the efficiency frontier or envelope.

The other algorithms do not belong to the efficiency frontier and remain outside of it. As already mentioned, the definition of efficiency is more general than interestingness as suggested by Fayyad et al. (1993).

One can connect also both concepts in this form that more efficient algorithms are more interesting. For ranking the algorithms, one can use the approach suggested by Andersen and Petersen (2008). (AP-model). They use a criterion that we call it the AP-value.

In input-oriented models the AP-value measures how much an efficient algorithm can radically enlarge its input-levels while remaining still efficient (output-oriented is analogous). For example, for an input-oriented method an AP-value equal to 1.5 means that the algorithm remains still efficient when its input values are all enlarged by 50%. If the algorithm is inefficient then the AP-value is equal to the efficiency value. In this paper, the CCR and BCC models are utilized to rank a normalization method.

### 3. Experimental Study

The experiment is designed to rank normalization methods with the SVM classification algorithm using the DEA model described in the previous section. The following subsections describe the performance measures, data sources and experimental design.

#### 3.1 Performance Measures

There are an extensive number of performance measures for classification. Commonly used performance measures in software defect classification are accuracy, precision, recall, F-measure, the area under receiver operating characteristic (AUC), and mean absolute error (Elish & Elish, 2008; Lessmann et al., 2008; Mair et al., 2000; Han & Kamber, 2006).

Besides these popular measures, this work includes seven other classification measures. The following paragraphs briefly describe these measures.

Overall accuracy: Accuracy is the percentage of correctly classified modules (Baeza-Yates & Ribeiro-Neto, 1999). It is one the most widely used classification performance metrics.

$$\text{Overall accuracy} = \frac{TN + TP}{TP + FP + FN + TN} \quad (10)$$

True positive (TP): TP is the number of correctly classified fault-prone modules. TP rate measures how well a classifier can recognize fault-prone modules. It is also called sensitivity measure.

$$\frac{\text{True positive rate}}{\text{sensitivity}} = \frac{TP}{TP + FN} \quad (11)$$

False positive (FP): FP is the number of non-fault-prone modules that is misclassified as fault-prone class. FP rate measures the percentage of non-fault-prone modules that were incorrectly classified.

$$\text{False positive rate} = \frac{FP}{FP + TN} \quad (12)$$

True negative (TN): TN is the number of correctly classified non-fault-prone modules. TN rate measures how well a classifier can recognize non-fault-prone modules. It is also called specificity measure.

$$\frac{\text{True negative rate}}{\text{specificity}} = \frac{TN}{TN + FP} \quad (13)$$

False negative (FN): FN is the number of fault-prone modules that is misclassified as non-fault-prone class. FN rate measures the percentage of fault-prone modules that were incorrectly classified.

$$\text{False negative rate} = \frac{FN}{FN + TP} \quad (14)$$

Precision: This is the number of classified fault-prone modules that actually are fault-prone modules.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (15)$$

Recall: This is the percentage of fault-prone modules that are correctly classified.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (16)$$

F-measure: It is the harmonic mean of precision and recall. F-measure has been widely used in information retrieval (Ferri, Hernandezorrallo, & Modroiu, 2009).

$$F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (17)$$

AUC: ROC stands for Receiver Operating Characteristic, which shows the tradeoff between TP rate and FP rate

(Ferri, Hernandezorrallo, & Modroiu, 2009). AUC represents the accuracy of a classifier. The larger is the area, the better is the classifier.

Kappa statistic (KapS): This is a classifier performance measure that estimates the similarity between the members of an ensemble in multi-classifiers systems (Witten & Frank, 2005).

$$KapS = \frac{P(A) - P(E)}{1 - P(E)} \quad (18)$$

P(A) is the accuracy of the classifier and P(E) is the probability that agreement among classifiers is due to chance.

$$P(E) = \frac{\sum_{k=1}^c \left( \left[ \sum_{j=1}^c \sum_{i=1}^m f(i,k) C(i,j) \right] \left[ \sum_{j=1}^c \sum_{i=1}^m f(i,j) C(i,k) \right] \right)}{m^c} \quad (19)$$

$m$  is the number of modules and  $c$  is the number of classes.  $f(i, j)$  is the actual probability of  $i$  module to be of class  $j$ .  $\sum_{i=1}^m f(i, j)$  is the number of modules of class  $j$ . Given threshold  $\theta$ ,  $C_\theta(i, j)$  is 1 if  $j$  is the predicted class for  $i$  obtained from  $P(i, j)$ ; otherwise it is [0, 1] (Triantaphyllou & Baig, 2005).

Mean absolute error (MAE): This measures how much the predictions deviate from the true probability.  $P(i, j)$  is the estimated probability of  $i$  module to be of class  $j$  taking values in [0,1] (Triantaphyllou & Baig, 2005).

$$MAE = \frac{\sum_{j=1}^c \sum_{i=1}^m |f(i, j) - p(i, j)|}{m \cdot c} \quad (20)$$

As the name suggests, the mean absolute error is an average of the absolute errors  $e_i = f_i - y_i$ , where  $f_i$  is the prediction and  $y_i$  the true value. Note that alternative formulations may include relative frequencies as weight factors. The mean absolute error is one of a number of ways of comparing forecasts with their eventual outcomes.

Well-established alternatives are the mean absolute scaled error (MASE) and the mean squared error. These all summarize performance in ways that disregard the direction of over- or under- prediction; a measure that does place emphasis on this is the mean signed difference.

Where a prediction model is to be fitted using a selected performance measure, in the sense that the least squares approach is related to the mean squared error, the equivalent for mean absolute error is least absolute deviations.

- Training time: the time needed to train a classification algorithm or ensemble method.
- Test time: the time needed to test a classification algorithm or ensemble method.

### 3.2 Data Sources

The data used in this study are 6 public-domain data sets from four application domains including Iris, Glass and Breast Cancer and Breast Cancer Wisconsin are provided by the UCI machine learning repository (<http://archive.ics.uci.edu/ml/>).

The Iris is a three-class (Iris Setosa, Iris Versicolour, Iris Virginica) data set that has 150 instances with 4 continuous predictor variables and 1 class variable. The predictor variables describe the Sepal length, Sepal width, Petal length and Petal width of Iris plants and the class variable indicates the type of iris plant.

The Breast Cancer is a two-class (no-recurrence-events, recurrence-events) dataset that has 699 instances with 9 continuous predictor variables and 1 class variable. The predictor variables describe the Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, mitoses.

The Breast Cancer Wisconsin is a two-class (R = recur, N = nonrecur) dataset that has 198 instances with 10 continuous predictor variables and 1 class variable. The predictor variables describe the radius (mean of distances from center to points on the perimeter), texture (standard deviation of gray-scale values), perimeter, area, smoothness (local variation in radius lengths), compactness (perimeter<sup>2</sup>/area - 1.0), concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, fractal dimension ("coastline approximation" - 1).

The Cardiocography is a three-class (N=normal; S=suspect; P=pathologic) dataset that has 15888 instances

with 22 continuous predictor variables and 1 class variable. The predictor variables describe the LB - FHR baseline (beats per minute), AC-#of accelerations per second, FM - # of fetal movements per second, UC - # of uterine contractions per second, DL - # of light decelerations per second, DS - # of severe decelerations per second, DP - # of prolonged decelerations per second ASTV - percentage of time with abnormal short term variability, MSTV - mean value of short term variability ALTV - percentage of time with abnormal long term variability, MLTV - mean value of long term variability, Width-width of FHR histogram, Min-minimum of FHR histogram, Max - Maximum of FHR histogram, Nmax - # of histogram peaks, Nzeros - # of histogram zeros, Mode - histogram mode, Mean - histogram mean, Median - histogram median, Variance - histogram variance, Tendency - histogram tendency, CLASS - FHR pattern class code (1 to 10).

The Wall-Following Robot Navigation is a four-class (Move-Forward, Slight-Right-Turn, Sharp-Right-Turn, Slight-Left-Turn) data set that has 5456 instances with 24 continuous predictor variables and 1 class variable. The predictor variables describe US1: ultrasound sensor at the front of the robot (reference angle: 180°), US2: ultrasound reading (reference angle: -165°), US3: ultrasound reading (reference angle: -150°), US4: ultrasound reading (reference angle: -135°), US5: ultrasound reading (reference angle: -120°), US6: ultrasound reading (reference angle: -105°), US7: ultrasound reading (reference angle: -90°), US8: ultrasound reading (reference angle: -75°), US9: ultrasound reading (reference angle: -60°), US10: ultrasound reading (reference angle: -45°), US11: ultrasound reading (reference angle: -30°), US12: ultrasound reading (reference angle: -15°), US13: reading of ultrasound sensor situated at the back of the robot (reference angle: 0°), US14: ultrasound reading (reference angle: 15°), US15: ultrasound reading (reference angle: 30°), US16: ultrasound reading (reference angle: 45°), US17: ultrasound reading (reference angle: 60°), US18: ultrasound reading (reference angle: 75°), US19: ultrasound reading (reference angle: 90°), US20: ultrasound reading (reference angle: 105°), US21: ultrasound reading (reference angle: 120°), US22: ultrasound reading (reference angle: 135°), US23: ultrasound reading (reference angle: 150°), US24: ultrasound reading (reference angle: 165°).

### 3.3 Experimental Design

The experiment was carried out according to the following process:

Input: the four selected datasets.

Output: Ranking of normalization techniques.

Step 1: Prepare target datasets: select and transform relevant features; data cleaning; data integration.

Step 2: Train and test the SVM classification model on a randomly sampled partitions (i.e., 10-fold cross-validation) using WEKA 3.7 (Shih, Huan-JyhShyur, & Lee, 2007).

Step 3: Normalized the data set by five methods.

Step 4: Train and test the SVM classification model on a randomly sampled partitions after normalization the data sets (i.e., 10-fold cross-validation) using WEKA 3.7 (Shih, Huan-JyhShyur, & Lee, 2007).

Step 5: Ranking the Normalization methods with attention accuracies get from SVM classification algorithm by DEA method.

Step 6: If gained efficiency from DEA model of two or more algorithms equal 1 then ranking Normalization methods with using A-P model of DEA.

END

Normalization methods selection problem involves benefit criteria. A criterion is called the benefit because the higher a normalization methods scores in terms of the corresponding criterion, the better the algorithm is (Abbas, Morteza, & Karimi, 2011).

## 4. Discussion of Results

Table 1 shows the classification results of 4 data sets using SVM classification algorithm before normalization methods.

Table 1. Classification results before normalization methods

Accuracy(SVM)	Data sets
Iris	96
Glass	80.84
Breast Cancer	96.71
Breast Cancer Wisconsin	96.71
Cardiotocography	60.35
Wall-Following Robot Navigation	74.8

Table 2 shows the classification results of 4 data sets using SVM classification algorithm after normalization methods.

Table 2. Classification results after normalization methods

Data set	Vector normalization	Accuracy of normalization Methods			
		Linear normalization (I)	Linear normalization (II)	Linear normalization (III)	Non-monotonic normalization
Iris	96.71	96.27	96	96	97.07
Glass	81	81	80.84	80.84	81.97
Breast Cancer	97.14	96.71	96.6	96.6	97.02
Breast Cancer Wisconsin	97.71	97.05	97.05	98.27	98.82
Cardiotocography	63.06	61	64.2	62.09	68.05
Wall-Following Robot Navigation	77.52	76.02	74.92	75.81	77.3

Table 3 represents the ranking results generated by DEA. That the efficiency of all normalization methods is 1 it means all normalization methods are efficient and for ranking the normalization methods we should rank them by A.P model.

Table 3. Ranking results generated by DEA

Normalization methods	Efficiency
Vector normalization	1
Linear normalization (I)	1
Linear normalization (II)	1
Linear normalization (III)	1
Non-monotonic normalization	1

Table 4 represents the ranking results generated by A.P model and ultimate evaluation of them.

Table 4. Represents the ranking results generated by A.P

Normalization methods	Overall efficiency by A.P Model	Final Ranking
Vector normalization	6.71	2
Linear normalization (I)	4.26	4
Linear normalization (II)	3.11	5
Linear normalization (III)	5.10	3
Non-monotonic normalization	8.42	1

## 5. Conclusions and Future Work

In this paper we implemented different normalization methods as a preprocessing to improve SVM algorithm. After testing the accuracy of SVM method on various Data Sets we concluded that these simple methods have low amount of processing and also increase the accuracy of SVM algorithm. For ranking these methods we used DEA which is the most practical decision making techniques.

We suggest to the future researchers to apply normalization methods on more data sets and also use other decision making techniques like TOPSIS and VIKOR (Peng et al., 1999) for ranking them. In addition testing these methods on other classification algorithms especially those based on n-dimension space like SMO to see how their accuracy improves is also suggested. And also they considerate time complexity in their researches.

## References

- Allwein, E. L., Schapire, R. E., Singer, Y., & Kaelbling, P. (2000). Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1, 113-124.
- Andersen, P., & Petersen, N. C. (1993). A Procedure for Ranking Efficient Units in Data Envelopment Analysis. *Management Science*, 39(10), 1261-1264. <http://dx.doi.org/10.1287/mnsc.39.10.1261>
- Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison, Wesley. New York, NY: ACM Press.
- Banker, R., Charnes, A., & Cooper. W. W. (1984). Some models for estimating technical and scale inefficiencies in data envelopment analysis. *Management Science*, 30, 1078-1092. <http://dx.doi.org/10.1287/mnsc.30.9.1078>
- Bazzazi, A. A., Osanloo, M., & Karimi, B. (2011). Deriving preference order of open pit mines equipment through MADM methods: Application of modified VIKORE method. *Expert Systems with Applications*, 38, 2550-2556. <http://dx.doi.org/10.1016/j.eswa.2010.08.043>
- Berrer, H., Paterson, I., & Keller, J. (2000). Evaluation of machine-learning algorithm ranking advisors. in: P. Brazdil, A. Jorge (Eds.), *Proceedings of the PKDD2000Workshop on Data Mining, Decision Support, Meta-Learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions*, 1-13.
- Camanho, A. S., & Dyson, R. G. (2005). Cost efficiency measurement with price uncertainty: A DEA application to bank branch assessments. *European Journal of Operational Research*, 161, 432-446. <http://dx.doi.org/10.1016/j.ejor.2003.07.018>
- Caruana, R., & Niculescu-Mizil, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. Appearing in *Proceedings of the 23 rd International Conference on Machine Learning*, Pittsburgh, PA.
- Challagulla, V. U. B., Bastani, F. B., Raymond, I. Y., & Paul, A. (2008). Empirical assessment of machine learning based software defect prediction techniques. *International Journal on Artificial Intelligence Tools*, 17(2), 389-400. <http://dx.doi.org/10.1142/S0218213008003947>
- Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6), 429-444. [http://dx.doi.org/10.1016/0377-2217\(78\)90138-8](http://dx.doi.org/10.1016/0377-2217(78)90138-8)
- Chen, K., Xu, L., & Chi, H. (1999). Improved learning algorithms for mixture of experts in multiclass classification. *Neural Networks*, 12, 1229-1252. [http://dx.doi.org/10.1016/S0893-6080\(99\)00043-X](http://dx.doi.org/10.1016/S0893-6080(99)00043-X)
- Chen, X., Skully, M., & Brown, K. (2005). Banking efficiency in China: Application of DEA to pre- and post-deregulation eras: 1993-2000. *China Economic Review*, 16, 229-245. <http://dx.doi.org/10.1016/j.chieco.2005.02.001>

- Cherchye, L., & Post, T. (2003). Methodological advances in DEA: a survey and an application for the Dutch electricity sector. *Statistica Neerlandica*, 57(4), 410-438. <http://dx.doi.org/10.1111/1467-9574.00238>
- Christopher, J. C. B. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 121-167. <http://dx.doi.org/10.1023/A:1009715923555>
- Cooper, W. W., Seiford, L. M., & Zhu, J. (2004). Data envelopment analysis: history, models and interpretations. *Handbook on Data Envelopment Analysis*. Boston: Kluwer Academic Publisher, Chapter 1, 1-39.
- Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel based vector machines. *Journal of Machine Learning Research*, 2, 265-292.
- Edirisinghe, N. C. P., & Zhang, X. (2007). Generalized DEA model of fundamental analysis and its application to portfolio optimization. *Journal of Banking & Finance*, 31, 3311-3335. <http://dx.doi.org/10.1016/j.jbankfin.2007.04.008>
- Elish, K. O. (2008). Predicting defect-prone software modules using support vector machines. *Journal of Systems and Software*, 81(5), 649-660. <http://dx.doi.org/10.1016/j.jss.2007.07.040>
- Ertay, T., Ruan, D., & Tuzkaya, U. R. (2006). Integrating data envelopment analysis and analytic hierarchy for the facility layout design in manufacturing systems. *Information Sciences*, 176, 237-262. <http://dx.doi.org/10.1016/j.ins.2004.12.001>
- Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: an overview. *Advances in Knowledge Discovery and Data Mining*, 1-34.
- Ferri, C., Hernandezorrallo, J., & Modroi, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 27-38. <http://dx.doi.org/10.1016/j.patrec.2008.08.010>
- Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques (2<sup>nd</sup> ed.)*. San Francisco: Morgan Kaufman.
- Hsu-Shih, S., Huan-Jyh, S., & Lee, E. S. (2007). An extension of TOPSIS for group decision making. *Mathematical and Computer Modeling*, 45, 801-813. <http://dx.doi.org/10.1016/j.mcm.2006.03.023>
- Hwang, C. L., & Yoon, K. (1981). *Multiple Attribute Decision Making*. Berlin: Springer-Verlag. <http://dx.doi.org/10.1007/978-3-642-48318-9>
- Johnes, J. (2006). Measuring teaching efficiency in higher education: An application of data envelopment analysis to economics graduates from UK Universities. *European Journal of Operational Research*, 174, 443-456. <http://dx.doi.org/10.1016/j.ejor.2005.02.044>
- Koksalan, M., & Tuncer, C. (2009). A DEA-based approach to ranking multi-criteria alternatives. *International Journal of Information Technology and Decision Making*, 8(1), 29-54. <http://dx.doi.org/10.1142/S0219622009003259>
- Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley. <http://dx.doi.org/10.1002/0471660264>
- Lessmann, S., Baesens, B., Mues, C., & Pietsch, S. (2008). Benchmarking classification models for software defect prediction: a proposed framework and novel findings. *IEEE Transactions on Software Engineering*, 34(4), 485-496. <http://dx.doi.org/10.1109/TSE.2008.35>
- Li, H., & Ma, L. (2008). Ranking decision alternatives by integrated DEA, AHP and gower plot techniques. *International Journal of Information Technology & Decision Making*, 7(2), 241-258. <http://dx.doi.org/10.1142/S0219622008002922>
- Liu, S. T. (2008). A fuzzy DEA/AR approach to the selection of flexible manufacturing systems. *Computers & Industrial Engineering*, 54, 66-76. <http://dx.doi.org/10.1016/j.dss.2007.10.011>
- Loucopoulos, C. (2001). Three-group classification with unequal misclassification costs: a mathematical programming approach. *Omega*, 29(3), 291-297. [http://dx.doi.org/10.1016/S0305-0483\(01\)00023-8](http://dx.doi.org/10.1016/S0305-0483(01)00023-8)
- Mair, C., Kadoda, G., Leflel, M., Phapl, L., Schofield, K., Shepperd, M., & Webster, S. (2000). An investigation of machine learning based prediction systems. *Journal of Systems Software*, 53(1), 23-29. [http://dx.doi.org/10.1016/S0164-1212\(00\)00005-4](http://dx.doi.org/10.1016/S0164-1212(00)00005-4)
- Mannino, M., Hong, S. N., & Choi, I. J. (2008). Efficiency evaluation of data warehouse operations. *Decision Support Systems*, 44, 883-898. <http://dx.doi.org/10.1016/j.dss.2007.10.011>

- Marakas, G. M. (2004). *Decision support system in the 21<sup>st</sup> Century (2<sup>nd</sup> ed.)*. Prentice-Hall of India.
- Milani, A. S., Shanian, A., & Madoliat, R. (2005). The effect of normalization norms in multiple attribute decision making model: A case study in gear material selection. *Structure Multiciplinary Optimization*, 29(4), 312-318.
- Nakhaeizadeh, G., & Schnabl, A. (1997). Development of multi-criteria metrics for evaluation of data mining algorithms. In *Proceeding of the Third International Conference on Knowledge Discovery and Data Mining (KDD'97)*, Newport Beach, California, August 14-17, 37-42.
- Nello, C., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge University Press.
- Olafson, S., Li, X., & Wu, S. (2008). Operations research and data mining. *European Journal of Operational Research*, 187, 1429-1448. <http://dx.doi.org/10.1016/j.ejor.2006.09.023>
- Peng, Y., Kou, G., Shi, Y., & Chen, Z. X. (2008). A descriptive framework for the field of data mining and knowledge discovery. *International Journal of Information Technology and Decision Making*, 7(4), 639-682. <http://dx.doi.org/10.1142/S0219622008003204>
- Peng, Y., Kou, G., Wang, G., Wang, H., & Ko, F. (2009). Empirical evaluation of classifiers for software risk management. *International Journal of Information Technology and Decision Making*, 8(4), 749-768. <http://dx.doi.org/10.1142/S0219622009003715>
- Platt, J. C., Cristianini, N., Shawe-taylor, J. (1999). Large margin DAGs for multiclass classification. In *Proceedings of neural information processing systems, NIPS'99*, MIT Press. 547-553.
- Rifkin, R. (2000). Svmfu. Retrieved from <http://five-percent-nation.mit.edu/SvmFu/>
- Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2), 1-39. <http://dx.doi.org/10.1007/s10462-009-9124-7>
- Shafer, S. M., & Byrd, T. A. (2006). A framework for measuring the efficiency of organizational investments in information technology using data envelopment analysis. *Omega*, 28, 125-141. [http://dx.doi.org/10.1016/S0305-0483\(99\)00039-0](http://dx.doi.org/10.1016/S0305-0483(99)00039-0)
- Thorsten, J. (1997). *Text categorization with support vector machines: Learning with many relevant features*. Technical report, University of Dortmund, Computer Science Department.
- Triantaphyllou, E., & Baig, K. (2005). The impact of aggregating benefit and cost criteria in four MCDA methods. *IEEE Transactions on Engineering Management*, 52, 213-262. <http://dx.doi.org/10.1109/TEM.2005.845221>
- Vladimir, V. (1995). *The Nature of Statistical Learning Theory (2<sup>nd</sup> ed.)*. New York, NY: Springer-Verlag.
- Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques (2<sup>nd</sup> ed.)*. San Francisco: Morgan Kaufman.
- Yang, Y. M., & Liu, X. (1995). *A re-examination of text categorization methods*. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Yang, Y. M., & Liu, X. (1999). *A re-examination of text categorization methods*. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Yoon, K. P., & Hwang, C. L. (1995). *Multiple Attribute Decision Making: An Introduction*. London: Sage Pub.