Effective Supervised Discretization for Classification based on Correlation Maximization

Qiusha Zhu, Lin Lin, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
Email: {q.zhu2, l.lin2}@umiami.edu, shyu@miami.edu

Shu-Ching Chen
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
Email: chens@cs.fiu.edu

Abstract

In many real-world applications, there are features (or attributes) that are continuous or numerical in the data. However, many classification models only take nominal features as the inputs. Therefore, it is necessary to apply discretization as a pre-processing step to transform numerical data into nominal data for such models. Well-discretized data should not only characterize the original data to produce a concise summarization, but also improve the classification performance. In this paper, a novel and effective supervised discretization algorithm based on correlation maximization (CM) is proposed by using multiple correspondence analysis (MCA) which is a technique to capture the correlations between multiple variables. For each numeric feature, the correlation information generated from MCA is used to build the discretization algorithm that maximizes the correlations between feature intervals/items and classes. Empirical comparisons with four other commonly used discretization algorithms are conducted using six well-known classifiers. Results on five UCI datasets and five TRECVID datasets demonstrate that our proposed discretization algorithm can automatically generate a better set of features (feature intervals) by maximizing their correlations with the classes and thus improve the classification performance.

Keywords: Supervised Discretization, Multiple Correspondence Analysis (MCA), Correlation, Classification.

1 Introduction

Nowadays, manual organization of the data/information can be very expensive or simply not feasible when time is limited or when the amount of data is enormous. To extract information and discover useful knowledge from it, data mining has become a dominant approach in the research community [1][2][10][12]. Classification, one of the main data mining techniques, has shown its attractive performance in several areas such as document categorization, image tagging, and video retrieval. However, features (or attributes) extracted from raw data are often numerical (or continuous). Some classification algorithms can only take nominal data as inputs, such as associative classifier, and some discretize numeric data into nominal data during the learning process, such as decision tree and rule-based learning. Therefore, discretization is needed as a pre-processing step to partition each numeric feature into a finite set of adjacent distinct intervals/items. A good discretization algorithm should not only characterize the original data to produce a concise summarization, but also help the classification performance.

Discretization algorithms can be categorized into unsupervised and supervised based on whether the class label information is used. Equal Width and Equal Frequency are two representative unsupervised discretization algorithms. Compared to supervised discretization, previous research [6][9] has indicated that unsupervised discretization algorithms have less computational complexity, but may result in much worse classification performance. When classification performance is the main concern, supervised discretization should be adopted. Algorithms such as IEM (information entropy maximization) [3], IEM's variant (IEMV) [5], class-attribute interdependence maximization (CAIM) [7], and class-attribute contingency coefficient (CACC) [11] belong to supervised discretization category.

In this paper, a novel and effective supervised discretization algorithm based on correlation maximization (CM) is proposed by using multiple correspondence analysis (MCA). MCA has been shown to be an effective technique to capture the correlations between multiple variables [8]. For each numeric feature, the candidate cut-point that maximizes the correlation between feature intervals and classes

is selected as the first cut-point, then this strategy is carried out in the left and right intervals recursively to further partition the intervals. Empirical comparisons with IEM, IEMV, CAIM, and CACC supervised discretization algorithms are conducted using six well-known classifiers. Results on five UCI datasets and five concepts from TRECVID 2009 demonstrate that our proposed discretization algorithm can automatically generate a better set of feature intervals by maximizing their correlation with classes and thus improves the classification performance.

The rest of the paper is organized as follows. Related work is introduced in Section 2. Our proposed correlation-maximization-based discretization is presented in Section 3, followed by an analysis of the experimental results in Section 4. Lastly, we conclude and discuss the future work in Section 5.

2 Related Work

Two main questions need to be answered when developing a discretization algorithm: when to cut and how to cut. Many discretization algorithms are based on information entropy, such as maximum entropy which discretizes the numeric attributes using the criterion of minimum information loss. IEM [3] is a widely used one due to its efficiency and good performance in the classification stage. IEM selects the first cut-point that minimizes the entropy function over all possible candidate cut-points and recursively applies this strategy to both induced intervals. The Minimum Description Length (MDL) principle is employed to determine whether to accept a selected candidate cut-point or not, and thus stop the recursion if the cut-point does not satisfy a pre-defined condition.

Compared to [3], the discretization algorithm in [5] uses the same strategy to select the best cut point but a different criterion to decide when to stop the recursion. Thus, it is considered as a variant of IEM (called IEMV). The goal is to compress the feature during the partitioning. Empirical studies have shown that this criterion is always negative for irrelevant features, which means irrelevant features are non-compressive.

In addition to entropy maximization, another well-known discretization criterion is class-attribute inter-dependence redundancy (CAIR) which measures the inter-dependence between classes and each discretized attribute, though it may be overfitting. CAIM [7] is a representative algorithm that maximizes mutual class-attribute inter-dependence and generates possibly the smallest number of intervals for a given numeric feature. The larger the value of CAIM, the higher the inter-dependence between the class labels and the discrete intervals. Instead of using the recursive strategy, CAIM selects the first cut-points from all candidates and then selects the next one from the rest of the candidate cut-points. It keeps the one with the highest CAIM value, and stops until the CAIM value of the next selected cut-point being smaller than the current highest one.

However, as pointed out in [11], CAIM gives a high factor to the number of generated intervals, which is usually very close to the number of classes. Also, CAIM only considers the majority class and ignores the rest. A discretization algorithm that follows the same strategy to select cut-points but uses contingency coefficient to measure the strength of dependence between the variables was proposed in [11]. Experiments on both real and artificial datasets indicated that CACC can generate a higher CAIR value compared to CAIM and improve classification accuracy like the decision trees.

3 The Proposed Discretization ALGORITHM

Multiple correspondence analysis (MCA) is a technique used to measure the correlation between multiple variables [4]. In this paper, for a candidate cut-point, MCA is used to measure the correlation between intervals/items and classes. The one that gives the highest correlation with the classes is selected as a cut-point. The geometrical representation of MCA not only visualizes the correlation relationship between intervals/items and classes, but also presents an elegant way to decide the cut-points. In this paper, we start with discretizing the numeric features with two classes, but it can be extended to a dataset with more than two classes.

3.1 Correlation Information from MCA

MCA can be considered as an extension of the standard correspondence analysis (CA) to more than two variables. It first constructs an indicator matrix (a two-way frequency cross tabulation table) with instances as rows and intervals of variables as columns. Given a feature of M intervals, and total number of data instances is N, the size of the indicator matrix denoted by Z is $N \times (M + K)$, where K is the number of classes. MCA analyzes the inner product of the indicator matrix $Z^T Z$, called the Burt Matrix which is symmetric with the size of $(M + K) \times (M + K)$. $P = Z^T Z/N$ is called the correspondence matrix with each element denoted as p_{ij}. Let r and c be the row and column mass vectors of P, i.e., $r_i = \sum_j p_{ij}$ and $c_j = \sum_i p_{ij}$. The centering involves calculating the differences $(p_{ij} - r_i c_j)$ between the observed and expected relative frequencies, and normalization involves dividing these differences by $\sqrt{r_i c_j}$ and leading to a matrix of standardized residuals $s_{ij} = (p_{ij} - r_i c_j)/\sqrt{r_i c_j}$, as shown in Equation (1).

$$S = D_r^{-1/2}(P - rc^T)D_c^{-1/2}, where \qquad (1)$$

D_r and D_c are diagonal matrices with these masses on the respective diagonals. Singular Value Decomposition (SVD) is performed on S as $S = U\Sigma V^T$, where Σ is the diagonal matrix with singular values, and $\Lambda = \Sigma^2$ is the diagonal matrix of the eigenvalues, which are also called principal inertias. The summation of each principal inertia is the total inertia which is also the amount that quantifies the total variance of S. The objective of MCA is to represent the maximum possible variance in a map of a few dimensions. Usually, the first two dimensions could capture over 95% of the total variance.

The graphical representation of MCA, called the symmetric map, can visualize the intervals of a feature and the classes as points in a two dimensional map. Thus, the correlation between an interval and a class can be well represented by the cosine angle between these two vectors in the first two dimensions. The larger the cosine value of the angle is, the stronger the correlation between them is. Fig. 1 shows a feature F_i with two intervals F_{ij}^1 and F_{ij}^2 given the candidate cut-point is t_j and two classes C_1 (positive class) and C_2 (negative class). a_{ij}^1 is the angle between F_{ij}^1 and C_1, and a_{ij}^2 is the angle between F_{ij}^2 and C_1. Since there are two intervals and two classes, if one interval is correlated with one class, then the other interval is negatively correlated with this class to the same degree, which means the sum of these two angles is 180 degrees. Thus Equation (2) stands.

$$cos(a_{ij}^1) = -cos(a_{ij}^2). \quad (2)$$

As shown in Fig. 1, a_{ij}^1 is much smaller than 90 degrees, which indicates that there is a higher correlation between F_{ij}^1 and the positive class, while F_{ij}^2 and the positive class are negatively correlated to the same degree which is given by $|cos(a_{ij}^1)|$ or $|cos(a_{ij}^2)|$. This motivates us to use the correlation information calculated from MCA to measure the quality of intervals generated by a candidate cut-point. A discretization scheme should contain cut-points that maximize the correlation between the feature intervals and the classes, so the discretized feature could give the most information of the class labels when used for classification.

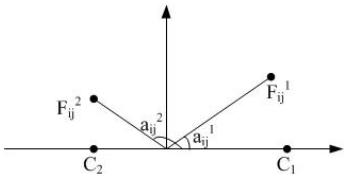

Figure 1. The symmetric map of the first two dimensions

3.2 Correlation Maximization (CM) Discretization

For a numeric feature F_i, all values of this feature are sorted to form a set of $n + 1$ distinct values. Candidate cut-points are the midpoints of all adjacent pairs in the set. For a candidate cut-point t_j, $|cos(a_{ij}^1)|$ or $|cos(a_{ij}^2)|$ is the value associated with t_j that is used to measure the correlation between the interval and the class, and thus represents the "discretization quality" of t_j. The one with the largest cosine value is selected as the first cut-point T_1. Then the same strategy can be carried out separately in the left and right intervals in a binary recursive way.

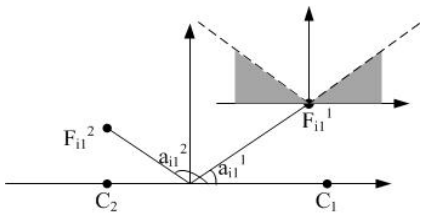

Figure 2. The symmetric map of interval partition

After selecting the cut-points, we need to decide when to stop the splitting recursion. In fact, the geometrical representation of MCA provides a clear way to define the stopping criterion. The idea is to terminate the recursion if the correlation between the current intervals and the classes is lower than the correlation between their predecessor and the classes. For feature F_i, suppose the first cut-point T_1 has been selected, the correlation associated with T_1 is $|cos(a_{i1}^1)|$, and the left interval F_{i1}^1 and the right interval F_{i1}^2 are displayed as Fig. 2. Take F_{i1}^1 as an example. The "discretization quality" of each candidate cut-point within this interval is calculated. If the selected cut-point produces a larger absolute cosine value than that of T_1, then the two generated subintervals within F_{i1}^1 will lie inside the dark regions below the dashed lines, as shown in Fig. 2. This indicates that the subintervals have higher correlations with the classes than their predecessor F_{i1}^1, and the overall correlation between feature F_i and the classes is improved because of F_{i1}^1's two subintervals. In contrast, if the subintervals of F_{i1}^1 fall into the region above the dashed lines, then the correlation between the subintervals and the classes is lower than F_{i1}^1 with the classes, and so the overall quality of F_i is decreased if such a cut-point is accepted. The same criterion is adopted to decide whether to further partition the right interval F_{i1}^2. The pseudo code of the CM Discretization Algorithm is as follows.

CM DISCRETIZATION ALGORITHM

```
1   for each feature F_i
2       set pre_correlation = 0
3       initialize an empty discretization scheme D_i
4       select the distinct values of F_i
5       sort the values in ascending order
6       calculate the midpoints of each adjacent pair
7       set max_correlation = 0
8       for each midpoint t_j in the current interval
9           calculate correlation between the interval
            and the classes by MCA
10          if correlation > max_correlation
11          set max_correlation = correlation
12      end
13      if max_correlation > pre_correlation
14          add the cut-point t_j into D_i
15          set pre_correlation = max_correlation
16      else
17          return the discretization scheme D_i
18      end
19      For the left interval, go to Line 7
20      For the right interval, go to Line 7
21  end
```

4 Experimental Results

To evaluate the proposed CM discretization algorithm, several experiments using the data from two benchmark sources: UCI datasets and TRECVID 2009 datasets were conducted. First, three-fold across validation is applied to split each dataset into three subsets with an approximately equal number of data instances and an equal P/N ratio (positive class to negative class ratio). Next, discretization is applied on the training dataset, and the same discretization scheme obtained from the training dataset is used to discretize the testing dataset. The final classification result is the average of these three folds. The performance of the proposed CM discretization algorithm is evaluated against four popular supervised discretization algorithms described in Section 2: IEM, IEMV, CAIM, and CACC. Six well-known classifiers in WEKA [12] are used to compare the final classification results. They are Adaptive Boosting (Ada), Decision Tree (DT), Rule based JRip (JRip), K Nearest Neighbor (KNN) where k=3, Native Bayes (NB), and Support Vector Machine (Sequential Minimal Optimization) (SVM). Precision (pre), recall (rec), and F1-score (F1) are adopted as our evaluation metrics for classification. The F1-score is the most important metric since it considers both precision and recall values. The classification results of the 5 UCI datasets are first analyzed. Then we discuss the results of 5 concepts from the TRECVID datasets which are more challenging for classification due to the highly imbalanced P/N ratio. Finally, some other criteria of evaluating a discretization algorithm are considered.

Table 1. UCI datasets

No.	data name	instances	features
1	pima_diabetes	768	8
2	haberman	306	3
3	hill_valley	606	100
4	ionosphere	351	34
5	breast_cancer	569	30

Table 2. TRECVID datasets

No.	concept name	P/N ratio
1	chair	0.07
2	traffic_intersection	0.01
3	person_playing_musical_instrument	0.04
4	person_playing_soccer	0.01
5	person_riding_bicycle	0.02

The major properties of the 5 UCI datasets and the 5 TRECVID datasets are described in Table 1 and Table 2, respectively. Compared to the UCI datasets, the 5 TRECVID datasets contain various P/N ratios and semantic meanings, and they have 12669 data instances and 48 numeric features. According to the P/N ratios, the TRECVID datasets are highly imbalanced with very few positive data instances.

For the UCI datasets, from the results shown in Table 3, IEM and IEMV produced very similar classification results (in pre, rec and F1) across all classifiers. CAIM is close to IEM and IEMV in general, and has about 2% or 3% less in F1-score, but sometimes outperforms them by a considerable margin. The results from CACC are comparable to CAIM but not as stable as CAIM. For example, in dataset No. 2, JRip and NB generate the best F1 value, while Ada and SMO give the worst. CM has the best performance in F1, achieving 4% to 5% higher than the other four methods (on average). In addition, as can be seen from Table 3, its performance is quite stable. As for the TRECVID datasets, as can be seen from Table 4, the performance of IEM and IEMV are very similar again, which is consistent with our observation on the UCI datasets. CAIM has the worst performance, especially in concepts No. 3, No. 4, and No. 5, followed by CACC. CM generates the best results, and outperforms IEM and IEMV by 6%, CACC by 10%, and CAIM by 15% in F1-score (on average).

Another important criterion of evaluating a discretization algorithm is the number of intervals, since a smaller number of intervals can speed up the classifier training process and also produce a more concise summarization of the original data. CAIM generates the lowest number of intervals, discretizes the features of all datasets into two intervals (which leads to less desirable classification results), and also confirms the conclusion from [11]: the number of intervals of

CAIM is very close to the number of classes. The number of intervals of CM is slightly higher than IEM and IEMV on average. For CM, each feature is discretized into at least two intervals. While for IEM and IEMV, about $1/5$ to $1/2$ of the features of TRECVID datasets have only one interval, which means these features are useless in the classification stage. Though this increases the efficiency of the classifiers, it results in worse classification performance compared to CM due to some mis-removed features. CACC generates the largest number of intervals, especially on TRECVID datasets. The concept "chair" (No. 1) has several features being discretized to a range from 400 to 700 intervals with many having 1 or few data instances. For other concepts, many intervals of features also contain few data instances, which is overfitting and clearly should be merged together or with other intervals.

Last, we compare the computational complexity of each algorithm. Due to the implementation issue, it is probably not fair by just looking at the running time, so we also analyze the time complexity. All these five algorithms need to sort the distinct values in a feature. Suppose there are n candidate cut-points, IEM, IEMV and CM use a binary recursive way to partition the intervals, so the time complexity is $O(nlog_2(n))$. While CAIM and CACC examine all the rest of the candidate cut-points at each round, so their time complexity is quadratic $O(n^2)$ to the number of instances. As can be seen from Table 3 and Table 4, checking every candidate cut-point does not generate a better discretization scheme for classification.

5 Conclusion and Future Work

Discretization is an important and necessary preprocessing step for many classification models. In this paper, a novel and effective discretization algorithm based on correlation maximization is proposed. MCA is utilized to measure the correlation between feature intervals and classes. The candidate cut-point that maximizes the correlation between feature intervals and classes is selected as a cut-point. This strategy is carried out in each interval recursively to further partition it. It stops when the correlation between the current intervals and the classes is lower than that of its predecessor. Experiments and analyses comparing our proposed discretization algorithm against other four discretization algorithms on six classifiers demonstrate that our proposed algorithm generates the best discretization scheme for classification, while containing a relatively small number of intervals and having a low computational complexity. Currently, CM focuses on discretizing a dataset with two classes and shows promising results. We will extend it to deal with a dataset with more than two classes in our future work. Furthermore, more datasets will be tested to evaluate the proposed CM discretization algorithm.

6 Acknowledgement

For Shu-Ching Chen, this material is based upon work supported by the U.S. Department of Homeland Security under grant Award Number 2010-ST-062-000039, by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0001, and by NSF HRD-0833093.

References

[1] S.-C. Chen, M.-L. Shyu, C. Zhang, and M. Chen. A multimodal data mining framework for soccer goal detection ba sed on decision tree logic. *International Journal of Computer Applications in Technology, S pecial Issue on Data Mining Applications*, 27(4):312–323, 2006.

[2] S.-C. Chen, M.-L. Shyu, C. Zhang, L. Luo, and M. Chen. Detection of soccer goal shots using joint multimedia features and classification rules. In *Proceedings of the Fourth International Workshop on Multime dia Data Mining*, pages 36–44, 2003.

[3] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1993.

[4] M. J. Greenacre and J. Blasius. *Multiple Correspondence Analysis and Related Methods*. Chapman and Hall/CRC, 2006.

[5] I. Kononenko. On biases in estimating multi-valued attributes. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1034–1040, 1995.

[6] S. Kotsiantis and D. Kanellopoulos. Discretization techniques: A recent survey. *GESTS International Transactions on Computer Science and Engineering*, 32(1):47–58, 2006.

[7] L. A. Kurgan and K. J. Cios. Caim discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):145–153, 2004.

[8] L. Lin, M.-L. Shyu, and S.-C. Chen. Correlation-based interestingness measure for video semantic concept detection. In *Proceedings of the 10th IEEE international conference on Information Reuse & Integration*, pages 120–125, 2009.

[9] M. Mizianty, L. Kurgan, and M. Ogiela. Comparative analysis of the impact of discretization on the classification with naive bayes and semi-naive bayes classifiers. In *Proceedings of the 2008 Seventh International Conference on Machine Learning and Applications*, pages 823–828, 2008.

[10] M.-L. Shyu, Z. Xie, M. Chen, and S.-C. Chen. Video semantic event/concept detection using a subspace-based m ultimedia data mining framework. *IEEE Transactions on Multimedia*, 10(2):252–259, 2008.

[11] C.-J. Tsai, C.-I. Lee, and W.-P. Yang. A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences*, 178(3):714–731, 2008.

[12] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, 2nd ed*. Morgan Kaufmann, 2005.

Table 3. Classification Results of UCI Datasets

No.	Methods	Ada			DT			JRip			KNN			NB			SMO		
		pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1
1	IEM	0.63	0.59	0.61	0.66	0.60	0.63	0.65	0.59	0.62	0.67	0.46	0.55	0.63	0.62	0.62	0.68	0.53	0.60
	IEMV	0.63	0.59	0.61	0.66	0.60	0.63	0.65	0.59	0.62	0.67	0.46	0.55	0.63	0.62	0.62	0.68	0.53	0.60
	CAIM	0.62	0.54	0.58	0.64	0.47	0.54	0.62	0.53	0.57	0.62	0.49	0.55	0.62	0.53	0.57	0.65	0.50	0.57
	CACC	0.63	0.59	0.61	0.63	0.58	0.60	0.63	0.59	0.61	0.65	0.53	0.58	0.61	0.62	0.61	0.64	0.52	0.57
	CM	0.62	0.61	0.61	0.67	0.63	0.65	0.65	0.62	0.63	0.66	0.53	0.59	0.62	0.67	0.64	0.64	0.60	0.62
2	IEM	0.47	0.44	0.45	0.14	0.16	0.15	0.14	0.16	0.15	0.29	0.30	0.29	0.30	0.20	0.24	0.14	0.16	0.15
	IEMV	0.47	0.44	0.45	0.14	0.16	0.15	0.14	0.16	0.15	0.29	0.30	0.29	0.30	0.20	0.24	0.14	0.16	0.15
	CAIM	0.52	0.35	0.42	0.34	0.20	0.25	0.33	0.21	0.26	0.37	0.19	0.25	0.51	0.22	0.31	0.34	0.20	0.25
	CACC	0.34	0.27	0.30	0.28	0.23	0.25	0.32	0.36	0.34	0.33	0.23	0.27	0.63	0.26	0.37	0.09	0.09	0.09
	CM	0.50	0.49	0.49	0.32	0.28	0.30	0.37	0.39	0.38	0.32	0.31	0.31	0.56	0.35	0.43	0.36	0.23	0.28
3	IEM	0.5	0.49	0.49	0.49	0.49	0.49	0.49	0.45	0.47	0.49	0.49	0.49	0.48	0.55	0.51	0.49	0.49	0.49
	IEMV	0.5	0.49	0.49	0.49	0.49	0.49	0.49	0.45	0.47	0.49	0.49	0.49	0.48	0.55	0.51	0.49	0.49	0.49
	CAIM	0.49	0.57	0.53	0.49	0.57	0.53	0.48	0.46	0.47	0.49	0.74	0.59	0.47	0.52	0.49	0.49	0.49	0.49
	CACC	0.48	0.48	0.48	0.48	0.48	0.48	0.51	0.49	0.50	0.48	0.47	0.47	0.50	0.48	0.49	0.51	0.49	0.50
	CM	0.52	0.59	0.55	0.51	0.64	0.57	0.51	0.49	0.50	0.54	0.69	0.61	0.51	0.57	0.54	0.56	0.51	0.53
4	IEM	0.9	0.94	0.92	0.9	0.96	0.93	0.90	0.95	0.92	0.86	0.99	0.92	0.89	0.93	0.91	0.92	0.95	0.93
	IEMV	0.91	0.93	0.92	0.88	0.94	0.91	0.92	0.89	0.90	0.87	0.99	0.93	0.89	0.93	0.91	0.91	0.94	0.92
	CAIM	0.91	0.93	0.92	0.9	0.94	0.92	0.89	0.93	0.91	0.83	0.99	0.90	0.92	0.90	0.91	0.91	0.95	0.93
	CACC	0.92	0.93	0.92	0.92	0.94	0.93	0.93	0.89	0.91	0.86	0.99	0.92	0.90	0.93	0.91	0.92	0.94	0.93
	CM	0.92	0.96	0.94	0.9	0.96	0.93	0.95	0.94	0.94	0.87	0.99	0.93	0.94	0.95	0.94	0.95	0.94	0.94
5	IEM	0.95	0.96	0.95	0.94	0.95	0.94	0.95	0.95	0.95	0.97	0.97	0.97	0.95	0.96	0.95	0.96	0.97	0.96
	IEMV	0.97	0.97	0.97	0.97	0.93	0.95	0.94	0.95	0.94	0.97	0.96	0.96	0.95	0.95	0.95	0.96	0.97	0.96
	CAIM	0.94	0.96	0.95	0.94	0.95	0.94	0.94	0.97	0.95	0.95	0.98	0.96	0.95	0.96	0.95	0.95	0.97	0.96
	CACC	0.92	0.94	0.93	0.93	0.97	0.95	0.95	0.93	0.94	0.96	0.97	0.96	0.96	0.96	0.96	0.96	0.97	0.96
	CM	0.97	0.97	0.97	0.95	0.98	0.96	0.97	0.95	0.96	0.98	0.97	0.97	0.95	0.98	0.96	0.96	0.97	0.96

Table 4. Classification Results of TRECVID Datasets

No.	Methods	Ada			DT			JRip			KNN			NB			SMO		
		pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1	pre	rec	F1
1	IEM	0.55	0.29	0.38	0.69	0.23	0.35	0.65	0.25	0.36	0.66	0.30	0.41	0.42	0.34	0.38	0.70	0.18	0.29
	IEMV	0.54	0.29	0.38	0.68	0.21	0.32	0.65	0.26	0.37	0.66	0.28	0.39	0.42	0.34	0.38	0.67	0.17	0.27
	CAIM	0.49	0.26	0.34	0.62	0.23	0.34	0.59	0.23	0.33	0.63	0.26	0.37	0.26	0.43	0.32	0	0	0
	CACC	0.24	0.14	0.18	0.67	0.23	0.34	0.64	0.25	0.36	0.71	0.28	0.40	0.42	0.28	0.34	0.34	0.13	0.19
	CM	0.65	0.34	0.45	0.65	0.31	0.42	0.58	0.32	0.41	0.64	0.37	0.45	0.40	0.55	0.46	0.45	0.26	0.33
2	IEM	0.47	0.24	0.32	0.92	0.17	0.29	0.79	0.20	0.31	0.91	0.21	0.34	0.12	0.35	0.18	0.91	0.17	0.29
	IEMV	0.73	0.26	0.38	0.97	0.18	0.30	0.90	0.16	0.27	0.94	0.20	0.33	0.13	0.34	0.19	0.91	0.15	0.26
	CAIM	0.48	0.21	0.29	0.74	0.17	0.28	0.83	0.23	0.33	0.94	0.17	0.29	0.14	0.29	0.19	0.90	0.16	0.27
	CACC	0.43	0.22	0.29	0.82	0.20	0.32	0.79	0.22	0.34	0.89	0.18	0.30	0.17	0.28	0.21	0.51	0.16	0.24
	CM	0.52	0.31	0.39	0.83	0.26	0.40	0.67	0.29	0.40	0.90	0.29	0.44	0.19	0.42	0.26	0.85	0.23	0.36
3	IEM	0.74	0.56	0.64	0.79	0.51	0.62	0.72	0.51	0.60	0.85	0.56	0.68	0.50	0.64	0.56	0.78	0.55	0.65
	IEMV	0.75	0.54	0.63	0.80	0.51	0.62	0.73	0.53	0.61	0.84	0.55	0.66	0.49	0.63	0.55	0.79	0.54	0.64
	CAIM	0.69	0.39	0.50	0.71	0.33	0.45	0.61	0.38	0.47	0.77	0.36	0.49	0.24	0.60	0.34	0.89	0.22	0.35
	CACC	0.80	0.48	0.60	0.87	0.41	0.56	0.69	0.50	0.58	0.85	0.46	0.60	0.42	0.58	0.49	0.83	0.43	0.57
	CM	0.75	0.59	0.66	0.81	0.53	0.64	0.71	0.61	0.66	0.78	0.64	0.70	0.51	0.68	0.58	0.83	0.68	0.75
4	IEM	0.60	0.46	0.52	0.82	0.24	0.37	0.62	0.40	0.48	0.86	0.49	0.62	0.29	0.83	0.43	0.76	0.46	0.57
	IEMV	0.61	0.45	0.52	0.77	0.22	0.34	0.58	0.41	0.48	0.92	0.41	0.57	0.28	0.81	0.42	0.82	0.42	0.56
	CAIM	0.67	0.36	0.47	0.65	0.25	0.36	0.70	0.44	0.54	0.90	0.22	0.35	0.27	0.61	0.37	0.76	0.32	0.45
	CACC	0.64	0.25	0.36	0.70	0.19	0.30	0.52	0.35	0.42	0.90	0.24	0.38	0.32	0.65	0.43	0.75	0.39	0.51
	CM	0.71	0.51	0.59	0.76	0.29	0.42	0.65	0.53	0.58	0.86	0.53	0.66	0.31	0.86	0.46	0.61	0.63	0.62
5	IEM	0.51	0.31	0.39	0.73	0.26	0.38	0.64	0.29	0.40	0.87	0.28	0.42	0.19	0.51	0.28	0.70	0.33	0.45
	IEMV	0.55	0.32	0.40	0.73	0.27	0.39	0.69	0.30	0.42	0.84	0.29	0.43	0.20	0.50	0.29	0.69	0.32	0.44
	CAIM	0.50	0.24	0.32	0.65	0.16	0.26	0.71	0.20	0.31	0.85	0.20	0.32	0.10	0.36	0.16	0.79	0.13	0.22
	CACC	0.60	0.31	0.41	0.68	0.24	0.35	0.59	0.34	0.43	0.84	0.27	0.41	0.19	0.49	0.27	0.64	0.30	0.41
	CM	0.51	0.38	0.44	0.71	0.34	0.46	0.61	0.34	0.44	0.81	0.33	0.47	0.22	0.55	0.31	0.64	0.41	0.50