Rank Aggregation based Text Feature Selection

Ou Wu, Haiqiang Zuo, Mingliang Zhu,
Weiming Hu and Jun Gao
NLPR, Institute of Automation, CAS
*{wuou, hqzuo, mlzhu,
wmhu,jgao}@nlpr.ia.ac.cn*

Hanzi Wang
*Computer Science Department, the University
of Adelaide,*
Hanzi.Wang@ieee.org

Abstract

Filtering feature selection method (filtering method, for short) is a well-known feature selection strategy in pattern recognition and data mining. Filtering method outperforms other feature selection methods in many cases when the dimension of features is large. There are so many filtering methods proposed in previous work leading to the "selection trouble" that how to select an appropriate filtering method for a given text data set. Since to find the best filtering method is usually intractable in real application, this paper takes an alternative path. We propose a feature selection framework that fuses the results obtained by different filtering methods. In fact, deriving a better rank list from different rank lists, known as rank aggregation, is a hot topic studied in many disciplines. Based on the proposed framework and Markov chains rank aggregation techniques, in this paper, we present two new feature selection methods: FR-MC1 and FR-MC4. We also introduce a perturbation algorithm to alleviate the drawbacks of Markov chains rank aggregation techniques. Empirical evaluation on two public text data sets shows that the two new feature selection methods achieve better or comparable results than classical filtering methods, which also demonstrate the effectiveness of our framework.

1. Introduction

Feature selection is a key issue in data mining and pattern recognition, in which a subset of the available features is selected to represent the samples. It has been widely accepted that an appropriate feature subset can avoid both the curse of dimensionality and over fitting effectively. In addition, feature selection can reduce the computational complexity.

Current feature selection methods proposed in the literature can be roughly classified into four categories [1, 2, 3, 4, 5, 6]: filtering, wrapper, embedding and hybrid methods. This paper focus on filtering strategy for its efficiency and effectiveness in handling the data sets with large size and high dimensions [1, 7]. There are numerous classical filtering methods and their variants proposed in previous literature. Naturally, one problem emerges: given a specific feature selection task, how to choose the best filtering method? We call this problem as "selection trouble". To the best of our knowledge, this problem is still open in theory. Many researchers choose to empirical evaluate existing filtering methods and then establish heuristic guidance. Due to the complexity of data properties, guidance which is useful for one data set usually does not work on another data set. As a matter of fact, many other areas also counter the similar problem. Take data classification as an example. There are also a great number of classifier models, how to choose a satisfying model is also difficult. We find that these applications usually take a fusion strategy to handle the "selection trouble". Since fusion strategy is prevails, we also try to take it to handle the "selection trouble" when using filtering methods.

With respect to filtering method fusion, there are two strategies: 1) combining different filtering criterions into a new one and 2) combining the results of different filtering method into a new result. We note that each filtering method can be treated as a voter which utilizes a criterion to order the original features. This is to say, different filtering methods merely differ in their produced feature ranks. Heuristically, we take the second strategy that fuses the rank lists of different methods into a new one. Deriving a better rank list from different rank lists, known as rank aggregation, has been studies in many research areas [8]. We combine filtering methods and rank aggregation techniques to generate a feature fusion (selection) framework. Two new feature selection methods are proposed based on the framework. The experimental results show the fusion framework is able to find robust and effective features.

The remainder of this paper is organized as follows. Section 2 briefly reviews filtering methods and the rank aggregation techniques. Section 3 describes our feature fusion framework and discusses rank aggregation algorithms. The section also proposes two concrete feature selection methods. Section 4 gives experimental evaluation on four public data sets and some discussions. Conclusions and future work are given in Section 5.

2. Related work

This section briefly introduces filtering feature selection methods and rank aggregation theory. We define some symbols which are used in the paper.

Let T= $\{t_1, t_2\ldots t_n\}$ be a set of alternatives or elements. τ_i is an ordered list defined on a subset of T, which is denoted by Z_i. τ_i can be written as $[t_i^1 \geq t_i^2 \geq \ldots \geq t_i^{|Z_i|}]$ where $|Z_i|$ denotes the number of elements in S_i. Let r_i^j be the position or ranks of the j-th element in τ_{i}. If $Z_i = T$, then τ_i is said to be a full list; otherwise, it is said to be a partial list. Let $S^\tau(i)$ represent a real-valued score assigned to the element i in τ. In most cases, a better rank is accompanied with a higher score. We can define a monotony decrease function to calculate scores for the elements. Suppose $\tau_1, \tau_2, \ldots, \tau_L$ are the input ordered lists. In most practical applications, they are not well ordered. Finding a better rank list based on aggregation the input lists can usually yield a more reasonable decision [8].

2.1. Filter feature selection methods

Let $X = \{x_i \mid 1 \leq i \leq L\}$ be the original feature sets. Theoretically, feature selection is to find a subset of X with the best discrimination ability compared with all other subsets of X. Note that the number of candidates is $2^{|X|}$. It is impractical to conduct an exhaustive test for each candidate. Considering the complexity of evaluating features subsets, many studies, i.e. filtering methods, focus on evaluating a single feature instead of a subset of features. They define a criterion $J(\cdot)$ to evaluate the ability to distinguish different classes of each candidate feature. Top k features are selected to represent samples (k is chosen manually or decided by the test data). Based on the previous work [14-18], five main criterions are chosen: Mutual Information (MI), Information Gain (IG), Chi-Squared (CHI), Bi-Normal Separation (BNS) and weighted Log Likelihood Ratio (WLLR). Since MI, IG and CHI are widely used in previous work, we only introduce BNS and WLLR briefly.

Bi-Normal Separation (BNS): BNS [14] was proposed for text classification. For a 2-category classification task, the criteria of BNS can be obtained as follows (t represents a feature):

$$J_{BNS}(t) = | F^{-1}(tpr) - F^{-1}(fpr) | \tag{1}$$

where F is the Normal condition density function; tpr is the rate of samples containing the candidate feature in one category and fpr is the rate of samples containing the candidate feature in the other category.

Weighted Log Likelihood Ratio (WLLR): WLLR [17] is also used to measure the information quantity of a feature. It can be written as:

$$J_{WLLR}(t) = \sum_{i=1}^{m} p(c_i) p(t \mid c_i) \log \frac{p(t \mid c_i)}{p(t \mid \neg c_i)} \tag{2}$$

where c_{i} denotes the i-th category.

2.2. Rank aggregation

Rank aggregation is a fundamental and classical optimization problem addressed in various areas such as economics, statistics, information retrieval, etc [8]. It combines many different rank orderings on the same set of members to obtain a “better” ordering. A great deal of techniques from different research areas have been proposed to address this rank aggregation problem [9, 10, 12]. Let τ represent an ordering list and $\tau\,(\cdot)$ represent the position of a given element in a rank list. Formally, rank aggregation is to find an optimal rank list τ^* with the following objective function:

$$\tau^* = \min_{\tau} \sum_i d(\tau,\ \tau_i)$$

where $d(\cdot,\cdot)$ is the distance function between the lists τ^* and τ_i. Different distance functions lead to different objective function and thus different optimization approaches. A widely accepted distance function for two fully ordered lists is shown below.

Kendall tau distance:

$$K(\tau_i, \tau_j) = | \{(x, y) \mid \tau_i(x) < \tau_i(y),\ \text{but}\ \tau_j(x) > \tau_j(y)\} | \tag{3}$$

When there are two or more input rank lists ($\tau_1, \tau_2, \ldots, \tau_H$), the *Kendall tau* distance between a list τ and ($\tau_1, \tau_2, \ldots, \tau_H$) is as follows:

$$K(\tau,\ \tau_1, \tau_2, \cdots, \tau_H) = \frac{1}{H} \sum_{i=1}^{H} K(\tau,\ \tau_i) \tag{4}$$

The aggregation obtained by optimizing *Kendall tau* distance is called as *Kemeny optimal aggregation. Kemeny optimal aggregation* is very popular for the reason that it simultaneously satisfies several important properties in social choice discipline [9, 13].

There are numerous rank aggregation techniques. One well-known yet simple method is Borda's count (BC). This method assigns a weight to each element. For the *k*-th element, its weight is defined as:

$$w(k) = \sum_{i=1}^{n} \tau_i(k) / n \tag{5}$$

Elements can be ranked according to their weights in an increasing order. Some other methods calculate the weights by combining the scores of elements in a linear way. One linear combination method (LCM) proposed in [12] is to calculate the weights according to the elements' scores, that is,

$$w(k) = \sum_{i=\mathbf{1}}^{n} s_{ik} / n \tag{6}$$

where s_{ik} represents the score of the *k*-th element in the *i*-th list. Then elements are ordered by the weights. Both BC and LCM methods can be seen as direct aggregation approach. There are some other methods utilizing optimization approach. Dwork [9] proposed Markov chains method to find the *Kemeny* optimal aggregation. Markov chains method is a series of algorithms differed in the strategy of constructing the state transmission matrix. This method views each element to rank is a state of a supposed stochastic system and constructs a state transition matrix denoted as *M* for all the elements (states) from the input rank lists. Based on the property of Markov chains, the left principal eigenvector (denoted as *v*) of *M* represents the stationary distribution of the system's states. The elements (states) can be ordered according to their values in *v*. Here we choose two state transition matrix construction algorithms namely MC1 and MC4 proposed in [9] due to their satisfactory performances:

MC1: If the current state is *a*, the next state *b* is chosen uniformly from the multiset of all elements ranked higher than (or equal to) *a* in the input rank lists, that is, from the multiset $\cup_m\{\, b \mid \tau_m(b) \le \tau_m(a)\}$.

MC4: If the current state is *a*, the next state is chosen as follows: first an element *b* is picked up uniformly from the union of all elements ranked by the input rank lists. If $\tau(b) < \tau(a)$ holds for a majority of the lists τ that contains both *a* and *b*, the state is changed to *b*; otherwise, the state remains unchanged.

3. Rank aggregation based feature selection

This section introduces the proposed feature selection framework and practical algorithms. In the practical algorithms, we apply Markov chains method as the rank aggregation techniques. Markov chains aggregation methods have some drawbacks. We will discuss its properties and give our modifications.

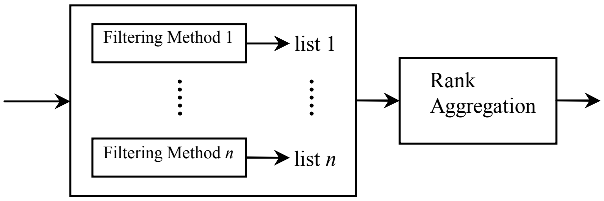

Figure 1. The outline of the proposed framework

3.1. The main framework

Each filtering criterion *J*(·) in last section is virtually a ranking function. There is no criterion which is able to outperform others in all the cases. That is, each existing criterion can be seen just revealing one aspect of the underlying discrimination abilities of features other than the whole. Thus existing criterions, seen as voters, can be combined to decide the feature orderings. Although the final orderings may not be the optimal orderings, it is believable that fusion (voting) strategy usually can provide robust and appropriate results. As a consequence, the "selective trouble" can be alleviated to some extent. However, directly combining different kinds of filtering criterion is also difficult for the reason that it is unclear whether their relationships are linear. We choose to combine the results of different kinds of filtering criterion. Our strategy can be written as:

Finding a better rank list τ, given {τ_1, ..., τ_n}

In essence, such strategy is just a rank aggregation approach. We then get a filtering fusion feature selection framework as shown in Figure 1. The whole approach contains two main steps: 1) applying different filtering methods to achieve feature rank lists and 2) utilizing rank aggregation technique to produce a new rank list.

Note that there are two main kinds of rank aggregation approach: score based such as LCM and rank based such as BC and Markov chains method. Studies in [12] shows that the rank-based aggregation outperforms the score-based aggregation if the lists in the aggregation have different rank-score curves. We investigate the rank-score curves of text features which are shown in Fig. 2. The curves suggest that the rank-based approach is more appropriate in aggregating text feature ranks. As a consequence, this study utilizes rank-based method.

Once the main issue of the framework is fixed, the next step is to choose rank aggregation technique to

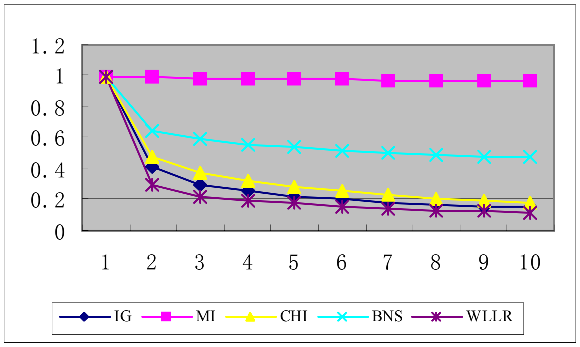

Figure 2. Rank-score curves of text features

generate concrete feature selection methods. Due to the satisfactory performance of Markov chains method [9], we will apply it into the proposed fusion approach. However, when using the Markov chains method to construct the state transition matrix, some groups become absorbing states leading that the ranks of other groups (non-absorbing states) cannot be achieved from the left principle vector. See the following case:

Case 1:

Elements	τ_1	τ_2	τ_3
a	a	a	a
b	c	b	b
c	b	c	d
d	d	d	c

The state transition matrixes obtained by MC1 and MC4 are as follows:

$$\begin{array}{cc} \text{MC1} & \text{MC4} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3/7 & 3/7 & 1/7 & 0 \\ 3/9 & 2/9 & 3/9 & 1/9 \\ 3/11 & 3/11 & 2/11 & 3/11 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/4 & 3/4 & 0 & 0 \\ 1/4 & 1/4 & 2/4 & 0 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix} \end{array} \tag{7}$$

Both their stationary distribution vectors (left principle eigenvector) are [1, 0, 0, 0], which means a>b=c=d, while the relationships among b, c and d are unknown (or say their orderings can not be reflected). The above cases also exist for other MC methods.

In the following subsection, we will discuss the underlying reasons and give a modification algorithm.

3.2. Rank aggregation discussion

Let M represent the state transition matrix to be determined. Once M is obtained, an ordered list can be derived according to the stationary distribution vector of M. However, in some cases, the ranks can not be obtained. To explain the underlying reason, we first introduce two element partitions:

ECC partition: An extended *Condorcet* criterion (ECC) [19] partition is defined as a partition $(C, \overline{C})$ of the element set T such that for any $i \in C$ and $j \in \overline{C}$ the majority prefers i to j.

Winner partition: A winner partition is defined as a partition $(C, \overline{C})$ of the element set T such that for any $i \in C$ and $j \in \overline{C}$ all prefers i to j.

Take Case 1 as an example, (a b | c d) is an ECC partition and (a | b c d) is a winner partition. It can be seen that a winner partition is also an ECC partition. With above definitions, we have the following theorem.

Theorem 1: *If there exists an ECC partition* $(C, \overline{C})$ *of element set A (or T), the orders of elements in* $\overline{C}$ *can not be obtained by the MC4 method.*

Proof: Without loss of generality, we can exchange the labels of elements such that the elements in C can be reordered to the front of the list. The state transition matrix can be written as:

$$M = \begin{bmatrix} M'_{|C|\times|C|} & \mathbf{0} \\ M''_{|C|\times|\overline{C}|} & M'''_{|\overline{C}|\times|C|} \end{bmatrix} \tag{8}$$

where for any $m_{ij} \in M''_{|c|\times|\overline{c}|}$, $m_{ij} > 0$. Then, we have:

$$\rho(M'''_{|\overline{C}|\times|\overline{C}|}) \le \left\| M'''_{|\overline{C}|\times|\overline{C}|} \right\|_{\infty} = \max_{i'} \sum_{j'}^{|C|} | m_{i'j'} | < 1 \tag{9}$$

From the properties of Markov Chains, the eigenvalue is 1. Let ($v^{a}_{|C|}$, $v^{b}_{|\overline{C}|}$) be the stationary distribution vector of M. We have:

$$[v^a_{|C|} \ \ v^b_{|\overline{C}|}] \begin{bmatrix} M'_{|C|\times|C|} & \mathbf{0} \\ M''_{|C|\times|\overline{C}|} & M'''_{|\overline{C}|\times|\overline{C}|} \end{bmatrix}$$

$$= \left[v^a_{|C|} M'_{|C|\times|C|} + v^b_{|C|} M''_{|C|\times|\overline{C}|} \quad v^a_{|C|}\mathbf{0} + v^b_{|C|} M'''_{|\overline{C}|\times|\overline{C}|} \right] \tag{10}$$

$$= \left[v^a_{|C|} M'_{|C|\times|C|} + v^b_{|\overline{C}|} M''_{|C|\times|\overline{C}|} \quad v^b_{|C|} M'''_{|\overline{C}|\times|\overline{C}|} \right]$$

It means:

$$v^b_{|\overline{C}|} M'''_{|\overline{C}|\times|\overline{C}|} = v^b_{|\overline{C}|} \tag{11}$$

If $v^b_{|\overline{C}|}$ is not zero vector, then

$$\rho(M'''_{|\overline{C}|\times|\overline{C}|}) \ge 1 \tag{12}$$

This is contradictory to (9). Consequently, the general

Table 1. Steps of the proposed feature selection algorithm

Input: Origimal feature set T

Output: Feature subset S

Steps:

a) Rank T using different filtering methods.

b) Apply MC1 (MC4) algorithm to construct the state transition matrix *M*.

c) Perturb *M* to *P* using Eq. (13) and calculate its left principle eigenvector *v*.

d) Order T according to *v*.

e) Select top *N* features as the feature subset S where *N* is heuristically defined or fixed on the validate set.

f) Output S.

form of the stationary distribution vector of M is ($v^{a}_{|C|}$, $\mathbf{0}$), from which the orders of the elements in $\overline{C}$ can not be inferred. With similar steps, we obtain the following theorem:

Theorem 2: *If there exists a winner partition* $(C, \overline{C})$ *of A (or T), the orders of elements in* $\overline{C}$ *can not be obtained by the MC1 method.*

Proof: Omitted.

To alleviate this drawback and motivated by the *Perron-Frobenius* theory [19] that a nonnegative irreducible matrix has a positive left principle eigenvector, this study proposes a heuristic strategy that perturbs the state transition matrix M to a nonnegative irreducible matrix. The perturbation matrix is chosen as follows:

$$E = -\varepsilon I + (\varepsilon / r(M))1 * 1^{T}$$

where $r(M)$ is the rank of M. Then the new state transition matrix is as follows

$$P = M + E \qquad (13)$$

After perturbation, each element of the new matrix is positive. Consequently, the new matrix is a nonnegative irreducible matrix and we can achieve all the ranks of each group (we have proved that the proposed perturbation does not change the order between absorbing states and non-absorbing states. See the appendix of the paper).

Using Eq. (13), the matrixes in (7) can be modified when ε is set to 0.0025. The new left principle eigenvector from MC1 is [0.9999995, 0.0007683, 0.0005792, 0.0003750]. The new left principle eigenvector from MC4 is [0.99997, 0.00663, 0.00224, 0.00113]. Both vectors show $a \succ b \succ c \succ d$.

3.3 Algorithms

If the rank aggregation in our framework shown in Figure 1 is implemented by Markov chains method, the detailed steps can be summarized as shown in Table 1.

Since the whole framework combines the filtering feature selection method and rank aggregation technique, the new algorithms we propose are called as FR-MC1 when MC1 is applied and FR-MC4 when MC4 is applied.

4. Evaluation

We evaluate the performances of the proposed filtering rank aggregation algorithms (FR-MC1 and FR-MC4) on text classification to compare the proposed methods with the five classical feature selection methods introduced in Subsection 2.1. The free software Libsvm-2.6 [25] is used to train and predict our data sets and RBF-SVM (Radial basis function support vector machine) is employed as our classifier model due to its satisfactory performance reported in previous literature [20].

We use the classification accuracy to measure the performance of a feature rank list:

$$accuracy = \frac{\text{the number of corrected classified samples}}{\text{the number of the whole test samples}}$$

In the perturbation of state transition matrix, ε is set to 0.01/rank(M).

4.2. Data sets and preprocessing

There are three public Chinese text sets toward classification: (a) Sogou set (Sogou-T V1.0 reduced version) which was collected by Sogou R&D center [21], (b) TanCorp V1.0 which is another Chinese text set [22] and (c) Fudan set [23] which consists of 20 subcategories involving 19737 documents. We compiled two new data sets: CS1 and CS2. CS1 contains 4306 documents while CS2 contains 10699. We also compiled two new data sets (ES1 and ES2) for English texts from the Reuters-21578 data collection [26]. ES1 contains 5896 documents consisting of two topics while ES2 contains 1140.

For Chinese texts, word segmentation needs to be done firstly. We use the free version of ICT-CLAS to segment the Chinese texts [24]. For example, if the string "知识发现与数据挖掘" is input, the segmentation performed by ICT-CLAS is "知识/发现/与/数据/挖掘".

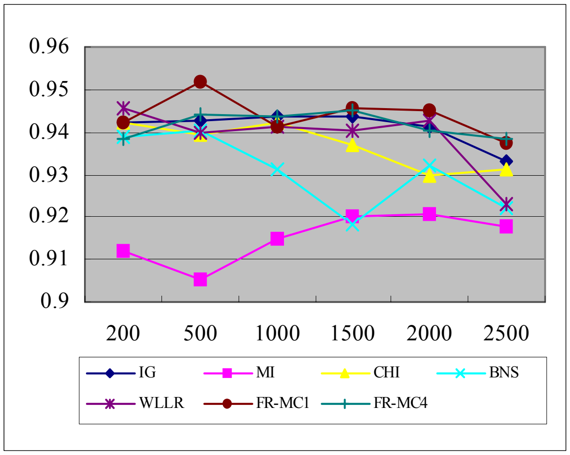

Figure 3. Classification results on CS1 set

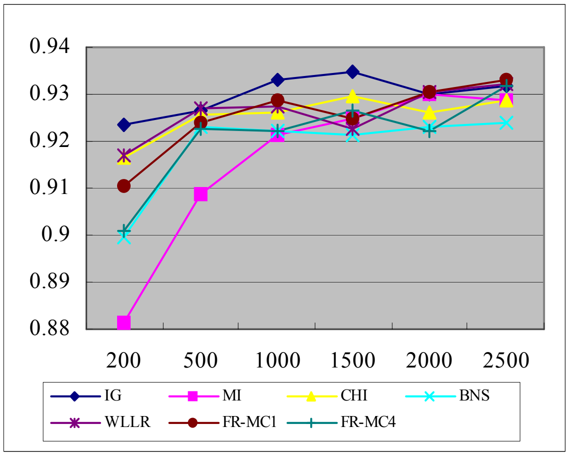

Figure 4. Classification results on CS2 set

4.2. Results

We evaluate the feature selection methods on the three data sets. For each data set, five different feature lists are obtained by using the five filtering methods respectively. When the construction of transition matrix, top 5000 (2500 * 2) of each list are selected as the input list. Once the new feature lists are obtained, top 200, 500, 1000, 1500, 2000, 2500 elements of each list are chosen to represent samples respectively. Each text set is divided into two equal parts: one is for training and the other is for testing.

CS1 set Fig.3 shows the performances of the five feature selection methods and the two new methods on the test part of CS1. It can be observed from Fig.3 that the two new methods (FR-MC1 and FR-MC4) achieve similar or better performances than all other methods. Between FR-MC1 and FR-MC4, FR-MC1 achieves the better overall performance. When the number of the features is 500, FR-MC1 yields the highest accuracy (0.9518) while all the values achieved by other methods are below 0.95. MI and BNS behave poorly on this data set.

CS2 set Fig.4 shows the performances of the five feature selection methods and the five aggregation methods on the test part of CS2. From Fig.4, the best two methods are IG and FR-MC1. The overall performances of MI and BNS are inferior to others. Although the overall performance of FR-MC4 is worse than CHI and WLLR, the highest accuracy FR-MC4 achieves equals to that of WLLR and higher than that of CHI.

ES1 set Fig.5 shows the performances of the five feature selection methods and the two filtering rank aggregation methods on the test part of ES.1 From Fig.5, one can observe that FR-MC4 and BNS achieves similar classification results and outperform others. FR-MC1 performs inferior to them but also better than the left methods: IG, WLLR, CHI and MI.

ES2 set Fig.6 shows the performances of the five feature selection methods and the two filtering rank aggregation methods on the test part of ES2. It can be observed that MI achieves the best results though it is the worst in other three data sets, which indicates that each filtering method has its own merit. FR-MC4 outperforms others except MI and BNS. FR-MC1 obtains similar results with IG. We can get that the performances of both FR-MC1 and FR-MC4 are not below the average.

4.3. Discussion

We can compare each classical method with the proposed new methods (FR-MC1 and FR-MC4). For FR-MC1 and IG, FR-MC1 outperforms IG on CS1 and ES1; RF-MCI performs inferior to IG only on CS2 and they achieve similar results on ES2. For FR-MC4 and CHI, we can obtain the same comparing conclusion with that of FR-MC1 and IG. In all, both FR-MC1 and FR-MC4 behave robust on all the data sets. They shows better or comparable performance then the other methods including IG and CHI while in text classification, IG and CHI are reported to be the two most effective methods [16].

Note that we aim to find a feature selection method which can achieve appropriate performance in most cases instead a super method which is able to achieve best results in all cases. We can get that the proposed framework with its two concrete methods provides a

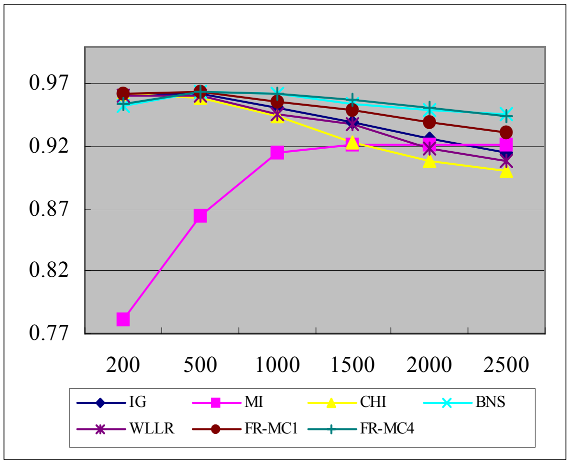

Figure 5. Classification results on ES1 set

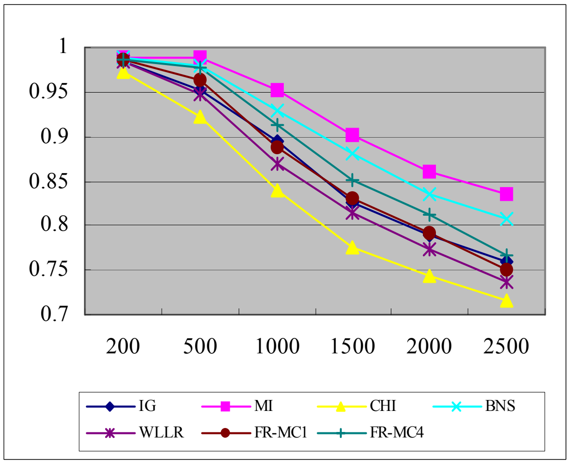

Figure 6. Classification results on ES2 set

feasible way with the probability of choosing an inappropriate features is reduced. That is, though the fusion framework can not ensure its results are the best, it is able to ensure that the results are above the average most probably and some times the best. Consequently, the "selection trouble" is alleviated.

5. Conclusions

In this paper, we propose a fusion framework for text feature selection. Our framework applies traditional filtering feature selection methods to produce several feature rank lists and then utilizes rank aggregation technique to fuse the rank lists. This framework provides a feasible way to combine different feature rank criterions, which is able to alleviate the "selection trouble" when there are so many available filtering methods (criterion). Based on the framework and Markov chains method, two new feature selection algorithms are introduced: FR-MC1 and FR-MC4. To alleviate the drawbacks of Markov chains methods, we study the properties of the state transition matrix and propose a perturbation algorithm. Four experiments are conducted on four public text data sets. The results suggested that the proposed new feature selection algorithms are able to yields promising performance and comparable to other classical filtering methods.

Evaluating a single feature is also essential in hybrid feature selection methods. It is possible that our study can improve the studies of hybrid feature selection.

6. Acknowledgement

This work is partly supported by NSFC (Grant No.60672040 and 60672040) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453).

7. References

[1] I. Guyon et al., "An Introduction to Variable and Feature Selection", *Journal of Machine Learning Research*, 3 (2003) pp.1157-1182.

[2] Z. Zhu et al., "Wrapper-filter feature selection algorithm using a memetic framework", *IEEE Trans. System Man and Cybernetics. Part B*, 2007, Vol.37, No.1, pp.70-76.

[3] L. Molina et al., "Feature Selection Algorithms: A Survey and Experimental Evaluation", *Pro. of International Conference on Data Mining* , 2002, pp.306-313.

[4] X. Geng et al., "Feature Selection for Ranking", *Pro. of ACM SIGIR Conference*, 2007, pp.407-414.

[5] A. Dasgupta et al., "Feature Selection Methods for Text Classification", *Pro. of ACM KDD Conference*, 2007, pp.230-239.

[6] L. Breiman et al., "Classification and regression trees", *Wadsworth and Brooks*,1984.

[7] E. Cant´u-Paz et al., "Feature Selection in Scientific Applications", *Pro. of ACM KDD Conference*, pp.788-793, 2004.

[8] C. Dwork, "Rank aggregation revisited", Manuscript, http://www.eecs.harvard.edu/~michaelm/CS222/rank2.pdf , 2001.

[9] C. Dwork et al., "Rank aggregation methods for the Web", *Pro. of World Wide Web Conference* , 2001, pp. 613-622.

[10]M. Farah et al., "An Outranking Approach for Rank Aggregation in Information Retrieval", *Pro. of ACM SIGIR Conference*, 2007, pp.591-598.

[11] M. Renda et al. "Web Metasearch: Rank vs. score based rank aggregation methods", *Pro. Of ACM SOAC Conference*, 2003, pp.841-846.

[12]J. Lee et al., "Analyses of multiple evidence combination", *Pro. of ACM SIGIR Conference*, 1997, pp. 267-276.

[13] H. Young et al., “A consistent extension of Condorcet's election principle”, *SIAM Journal on Applied Math*, 1978, 35(2):285-300.

[14] G. Forman, “An extensive empirical study of feature selection metrics for text classification”, *Journal of Machine Learning Research*, 2003,3(1):1533-7928.

[15]T.Liu et al, “An evaluation on feature selection for text clustering”, *Proc. of International Conference on Machine Learning*, 2003, pp.488-495.

[16]Y.Yang et al., “A Comparative Study on Feature Selection in Text Categorization”, *Proc. of International Conference on Machine Learning*,, 1997 pp.412-420.

[17]V. Ng et al., “Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Review”, *Proc. of the COLING/ACL Main Conference Poster Sessions*, 2006, pp.611–618.

[18]K.Nigam et al., “Text classification from labeled and unlabeled documents using EM”, *Machine Learning*, 2000, 39(2):103-134.

[19]S. Householder, “The Theory of Matrices in Numerical Analysis”, *Blaisdell Publishing Company*, 1964.

[20]T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, *Proc. the 10th ECML*, pp.137-142, 1998.

[21]http://www.sogou.com/labs/dl/t.html.

[22]S. Tan et al., “A Novel Refinement Approach for Text Categorization”, *Pro. of ACM CIKM Conference*, pp.469-476, 2005.

[23]http://www.nlp.org.cn/docs/download.php?doc_id=294

[24] http://sewm.pku.edu.cn/QA/

[25]http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[26]http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

Appendix:

Let C denote the set of absorbing states and $\overline{C}$ be the set of non-absorbing states. If using the original state transition matrix constructed by MC4, any state in C will rank better than all states in $\overline{C}$. This correctly reflects the state orderings. We concern about whether this relationship can hold after perturbation. Without loss of generality, we can exchange the labels of states such that the states in C can be arranged to the front of the list. Then the state transition matrix constructed by MC4 can be written as:

$$P = \begin{bmatrix} P^{'}_{|C|\times|C|} & \mathbf{0} \\ P^{''}_{|C|\times|\overline{C}|} & P^{'''}_{|\overline{C}|\times|C|} \end{bmatrix} \qquad \text{(A-1)}$$

We have the following lemma:

Lemma A.1: *Let (A-1) be the state transition matrix obtained by MC4. If M is modified to P by Eq. (14), states in C are still ordered better or at least no worse than all states in $\overline{C}$ by P.*

Proof: Let l be an arbitrary state in C and k be an arbitrary element $\overline{C}$, v be the left principal eigenvector. The corresponding values of l and k in v are v_l and v_k respectively. Note that the left principal value of the state transition matrix equals to 1. We denote P as $[P_1,\cdots,P_l,\cdots]$. We have:

$$v\cdot P_l = v_l \quad \text{and} \quad v\cdot P_k = v_k$$

or

$$\sum_{i\in c,i\neq l} p_{il}v_i + \sum_{j\in\overline{c},j\neq k} p_{jl}v_j + p_{ll}v_l + p_{kl}v_k = v_l \qquad \text{(A-2)}$$

$$\sum_{i\in c,i\neq l} p_{ik}v_i + \sum_{j\in\overline{c},j\neq k} p_{jk}v_j + p_{kk}v_k + p_{lk}v_l = v_k \qquad \text{(A-3)}$$

According to *MC4* and Eq. (14), we can obtain:

$$\begin{aligned} p_{il} &= m_{il} + \varepsilon / N \\ p_{jl} &= (1+\varepsilon)/N \\ p_{ll} &= m_{ll} - \varepsilon(N-1)/N \\ p_{ik} &= \varepsilon / N \\ p_{jk} &= m_{jk} + \varepsilon / N \\ p_{kk} &= m_{kk} - \varepsilon(N-1)/N \end{aligned} \qquad \text{(A-4)}$$

Plugging Eq. (A-4) into (A-2) and (A-3) and Using Eq. (A-2) subtracts (A-3), we have:

$$v_l - v_k = (1+\varepsilon-m_{kk})^{-1}(\sum_{i\in c,i\neq l} m_{il}v_i + \sum_{j\in\overline{c},j\neq k}(1/N-m_{jk})v_j + (m_{ll}-m_{kk})v_l + v_k/N) \qquad \text{(A-5)}$$

Note that we have the following constraints:

$$\begin{aligned} &\varepsilon > 0 \\ &m_{jk} \leq 1/N \\ &m_{ll} >= 1-|C|/N \\ &m_{kk} <= 1-|C|/N \end{aligned} \qquad \text{(A-6)}$$

Then we can deduce that every part in the right part of Eq. (A-5) is nonnegative. Then we obtain:

$$v_l - v_k \geq 0$$

It can be concluded that l is still ordered better or at least no worse than k by P.