Multimedia Systems (2014) 20:389–413
DOI 10.1007/s00530-014-0360-6

REGULAR PAPER

RELIEF-MM: effective modality weighting for multimedia information retrieval

Turgay Yilmaz · Adnan Yazici · Masaru Kitsuregawa

Received: 31 May 2013 / Accepted: 17 January 2014 / Published online: 16 February 2014

Abstract Fusing multimodal information in multimedia data usually improves the retrieval performance. One of the major issues in multimodal fusion is how to determine the best modalities. To combine the modalities more effectively, we propose a RELIEF-based modality weighting approach, named as RELIEF-MM. The original RELIEF algorithm is extended for weaknesses in several major issues: class-specific feature selection, complexities with multi-labeled data and noise, handling unbalanced datasets, and using the algorithm with classifier predictions. RELIEF-MM employs an improved weight estimation function, which exploits the representation and reliability capabilities of modalities, as well as the discrimination capability, without any increase in the computational complexity. The comprehensive experiments conducted on TRECVID 2007, TRECVID 2008 and CCV datasets validate RELIEF-MM as an efficient, accurate and robust way of modality weighting for multimedia data.

Keywords RELIEF · Feature weighting · Multimodal fusion · Multimedia information retrieval

1 Introduction

Increase in the use of digital multimedia data in recent years has shown the need for multimedia retrieval systems. Retrieval of multimedia data is based on its semantic content. To handle the semantic content effectively, the nature of the multimedia data should be examined and information contained in multimedia data should be used completely. The multimedia data usually has a complex structure containing multimodal information (i.e., audial, visual and textual modalities). Regarding the noise in sensed data, non-universality of any single modality and the performance upper bound of each modality, relying on a single modality may not be applicable [37]. Furthermore, it has been observed that the sets of patterns misclassified by different modalities do not necessarily overlap, and complementary information provided by different modalities improves recognition capability [25]. Since each modality abstracts videos from a different aspect, different modalities in multimedia data complement each other [17]. Thus, combining multimodal information usually improves the retrieval performance. However, there exist two major issues that have not been adequately addressed yet and are still attractive research areas [31, 37, 52]: (1) How to determine the best modalities? (2) How best to fuse them? This paper focuses on the first problem and presents a

Communicated by L. Zhang.

This work is supported in part by a research grant from TÜBİTAK EEEAG (Grant Number 109E014).

T. Yilmaz (✉) · A. Yazici
Computer Engineering Department, Middle East Technical University, 06531 Ankara, Turkey
e-mail: turgay@ceng.metu.edu.tr

A. Yazici
e-mail: yazici@ceng.metu.edu.tr

T. Yilmaz · M. Kitsuregawa
Institute of Industrial Science, The University of Tokyo, Tokyo 153-8505, Japan
e-mail: kitsure@tkl.iis.u-tokyo.ac.jp

M. Kitsuregawa
National Institute of Informatics, Chiyoda-ku, Tokyo 101-8430, Japan

modality weighting approach to use the multiple modalities effectively.[1]

The modality selection is a combinatorial search problem that aims to find the best subset of available modalities giving the highest accuracy. Such a computational problem can be solved to some extent using a weighting strategy. Modality weighting is a generalization of the selection problem, where the modalities are ranked by assigning some weights in between [0, 1] to each modality, instead of a binary selection. The use of weights enables some well-established optimization techniques and efficient algorithmic implementations to be employed [46]. Furthermore, a weighting strategy is a practical solution since the most frequently utilized fusion approach is the Linear Weighted Fusion [9, 49, 53], in which the combined decision is calculated as a weighted sum of the available modalities.

The previous studies on using multiple modalities can be categorized into three groups: (1) using all features/modalities by averaging them (2) performing an empirical selection and (3) determining the effectiveness of each feature with a weighting algorithm. Despite their wide usage among fusion studies, the first two are simplistic approaches; the first one treats all features as equally likely although any of the features can be non-informative or redundant, whereas the second approach requires an empirical observation and manual selection based on the observation. On the other hand, the third direction requires design of an efficient feature weighting algorithm, which proposes a polynomial time heuristic for the combinatorial explosion problem while dealing with multiple features.

Regarding the third direction, we focus on some adaptable solutions from *feature weighting* studies in the machine learning literature. However, the feature weighting solutions are not easily applicable to the modality weighting problem, considering the issues of (1) the intrinsic multi-dimensionality of modalities and (2) the multivariate inputs of fusion systems. The former issue states that *feature weighting* methods give weights for each dimension of an input feature vector, whereas modality weighting methods assign weights to each modality, each of which is a multi-dimensional feature, by accepting each modality as a black-box. Besides, the latter is a more general issue in fusion systems. The inputs of a fusion system are not necessarily feature values. The prediction scores for different features/modalities are frequently combined in state-of-the-art fusion studies. An intuitive idea to discard these problems is to utilize a weighting approach that works in distance-based metric space, instead of using a feature space. Utilizing a distance space solves the intrinsic dimensionality problem of multiple modalities by converting multi-dimensional feature values of a modality to a uni-dimensional distance value. Furthermore, it enables handling of the prediction scores after converting them into applicable dissimilarity values with appropriate conversion functions.

Among the existing *feature weighting* algorithms, we focus on the RELIEF algorithm [23], which is considered one of the most successful weighting algorithms and in which the calculations are based on the distances between training samples. Furthermore, according to the best of our knowledge, there exists no usage of the RELIEF algorithm for multimodal feature selection[2] in multimedia retrieval. The key idea of RELIEF is to iteratively estimate feature weights according to their ability to discriminate between neighboring samples. Employing the RELIEF algorithm for multimodal feature selection on multimedia data enables to identify some weaknesses of the algorithm, which have not been addressed before. Our solution is based on RELIEF-F, which is the multi-class extension of the basic RELIEF algorithm. We extend RELIEF-F in the following aspects, considering the characteristics of multimedia data and multimedia retrieval systems:

1. *Class-specific selection* Multimedia retrieval is a multi-class problem with a high number of concepts/classes. One major drawback of RELIEF-F is that it generates weights in a class-common way, where the same feature weights are assigned for all concepts. However, each concept can be represented better with different features that are specific to that concept [50, 55]. Thus, it is important to use a class-specific modality weighting approach in the multimedia retrieval systems, to handle the high number of classes.
2. *Multi-labeled data* Multimedia data is usually multi-labeled. However, the RELIEF-F algorithm cannot perform well when the training samples are multi-labeled. RELIEF-F estimates the weights of the features according to their ability to discriminate between different classes. Having multi-labeled samples causes the algorithm not to discriminate between classes effectively, due to the ambiguity produced by the samples associated with multiple concept types.
3. *Noisy data* Multimedia data contains a vast amount of noise. However, the way RELIEF-F deals with noisy data is inadequate. Similar to the multi-label issue, noise in the samples hinders a correct discrimination between classes.
4. *Unbalanced data* The training samples provided in multimedia datasets are usually unbalanced between

[1] This paper is a revised and extended version of [54].

[2] The final goal of this study is to select the effective modalities by weighting the available modalities and each modality is a multi-dimensional feature. Thus, from now on, the phrases 'modality selection', 'modality weighting' and 'multimodal feature selection' are used interchangeably.

classes. Although RELIEF-F applies k nearest neighbor approach to deal with the outlier data, an unbalanced dataset prevents RELIEF-F from eliminating outlier data effectively. Assuming that each class has approximately the same amount of noisy samples (as a ratio), using the same k for all classes makes the algorithm include more noisy samples for the classes with smaller numbers of training samples. Thus, having different numbers of samples for each class affects the performance of the RELIEF-F algorithm negatively.

5. *Late fusion inputs* In regular use of RELIEF-based algorithms, the distances between instances are calculated using the feature values. However, the late fusion approaches usually rely on prediction scores and the feature values may not be available at the time of fusion. Thus, a procedure that enables using the prediction scores is necessary.

In this paper, we propose a new RELIEF extension for multimedia data (RELIEF for Multimedia data: RELIEF-MM) to handle the above given research issues. First, we restate the RELIEF-F algorithm in a class-specific way and show that the weights produced by the original RELIEF-F are equal to the average of all class-specific weights. Thus, generating class-specific weights does not have a negative effect on the computational complexity of the algorithm. Second, we deal with the multi-label and noise issues, and extend the weight estimation function by including the representation and reliability characteristics of the features in addition to the currently used discrimination capabilities. These characteristics of features are calculated based on the statistics of distances between the training instances, by complying with the distance–space criteria discussed before. The mean distances between the samples of each class are employed as the representative characteristics, and the correctness ratios of features for each class are used as the reliability characteristics. For the discriminative property, we calculate the distance between the means of classes, as in the original RELIEF-F. Third, we deal with the unbalanced data problem, and propose the use of dynamic k nearest neighbor selection. In dynamic k selection, a different k value is calculated for each class, instead of the same k value for all classes. The dynamic k value is used as a predefined ratio of the number of samples in each class. This modification makes the algorithm deal with approximately the same ratio of noisy instances for all classes and give more regularized weight assessments. Lastly, we enable RELIEF-F algorithm for use with classifier predictions by converting the prediction scores into distances between instances.

We evaluate the RELIEF-MM algorithm with the TRECVID 2007 [35], TRECVID 2008 [36] and Columbia Consumer Video (CCV) Database [19] datasets. For each of the issues discussed above, we perform comparative tests against the RELIEF-F algorithm. In addition, we compare the multimedia retrieval accuracies of the RELIEF-MM-based linear weighted fusion approach with single modalities, simple averaging and exhaustive search. As a general overview, we can state that the proposed RELIEF-MM algorithm generates better feature weights than the RELIEF-F algorithm and the computational complexity is still asymptotically the same as the original algorithm. It has been observed that the fusion methods empowered by RELIEF-MM guarantee higher accuracies than any single modality. RELIEF-MM also demonstrates much better performance than simple averaging and RELIEF-F-based methods. Moreover, RELIEF-MM gives nearly the same performance as the exhaustive search-based approach, yet it is computationally much more efficient than the exhaustive one.

The remainder of this paper is organized as follows: In Sect. 2, an overview of modality selection in information fusion, feature selection methods and a detailed description of the RELIEF algorithms are given. In Sect. 3, the RELIEF-MM algorithm is presented in detail. In this section, first of all, the RELIEF algorithm is restated in a class-specific way, then the extensions for multi-label, noisy and unbalanced data problems are described. After introducing the extensions in detail, the combined algorithm is presented, along with a computational complexity analysis. Lastly, the strategy for using the RELIEF-MM with late fusion inputs (i.e., prediction scores) is described. In Sect. 4, the empirical results and the evaluations of our proposed solutions are given. In the last section, some conclusions are drawn and some possible future studies are discussed.

2 Related work

In multimedia retrieval, the most popular strategies for combining multimodal information are early fusion and late fusion. Early fusion is the concatenation of all available modalities into a single feature vector, whereas late fusion is the linear combination of classifier outputs after processing each modality by a separate classifier [17]. The studies in the literature do not present a clear winner between these two approaches, in terms of accuracy. Yet, early fusion usually leads to the "curse of dimensionality problem" because of concatenation of the modalities. On the other hand, late fusion is simple in calculation and has a reasonable performance despite its simplicity. Thus, late fusion has attracted much more attention than early fusion in recent studies [17, 31]. However, the selection of modalities (i.e., assigning weighs for each modality) is an important issue in late fusion, and affects the retrieval accuracy in fusion results. In this study, we focus on efficiently determining the effectiveness

of modalities. Below, we first present recent studies on modality selection for multimedia data. Then, with a machine learning point of view, the modality selection problem is compared with the feature selection problem in machine learning literature, and the well-known approaches for feature selection are presented. Lastly, we discuss the family of the RELIEF algorithms.

2.1 Modality selection/weighting

In the multimedia domain, the majority of the fusion studies prefer simplistic solutions for combining all available modalities by performing an empirical weighting scheme or a simple averaging [15, 31, 45]. An empirical weighting method is based on empirical observations and manual selection of the features. Besides, a simple averaging approach assumes that all of the modalities are equally effective although any of the features can be non-informative or redundant. Some successful utilizations of simple averaging can be found in [19, 20], where they obtain higher retrieval accuracies than any single modality. Yet there are several studies that perform the selection/weighting by evaluating the effectiveness of each modality, and some of the recent ones are summarized below.

One popular approach for modality selection is the use of the accuracy values as the weight estimations. In [9], Fumera et al. provide a theoretical analysis of this idea. Some recent utilizations of this idea can be found in [14, 34, 39]. Another approach applied in the literature is to find the independent feature subsets, considering that the result of the fusion process is improved if complementary (independent) inputs are combined [27]. Towards this direction, Wu et al. [52] redefine 'modality' as an 'independent component' among the available features and find statistically independent modalities from raw features by employing principle component analysis (PCA), independent component analysis (ICA) and independent modality grouping (IMG) techniques. Kludas et al. [26] apply the independency idea and use correlation coefficients to measure the dependency between features. Besides these, Atrey et al. [1], Kankanhalli et al. [22] and Snidaro et al. [44] study the problem in another perspective, and try to combine multiple data streams (e.g. data obtained from several different sensors like video camera, microphone, etc.), where each data stream can be accepted as a different modality. Atrey et al. [1] use a dynamic programming approach to find the optimal subset of media streams based on several criteria which maximizes the information gain obtained. Kankanhalli et al. [22] propose an experiential sampling-based solution for selecting the most informative subset of data streams. Snidaro et al. [44] define a quality metric for the data streams and dynamically regulate the fusion process. Further recent studies on the topic is as follows: Kalamaras et al. [21] take the advantage of user feedback and learn the modality weights via an interactive user feedback scheme. Huang et al. [12] tailor the genetic algorithm to learn modality weights and apply it to alleviate the local minima problem during the process of finding an optimal solution. Moulin et al. [32] reformulate the modality weighting problem as a dimensionality reduction problem in a binary classification context and find the linear combination that best separate relevant and non-relevant documents for all queries using a Fisher Linear Discriminant Analysis-based approach. Chen et al. [5] calculate the modality weights by measuring the discriminative capability of each visual feature by a voting scheme, where the voting scheme is applied by processing all triples of the training samples (candidate, positive and negative) and assigning a vote for the candidate according to whether the candidate is closer to the positive or the negative. Wu et al. [51] consider the interactions among the multimodal classifier outputs and employ a fuzzy integral-based approach to find modality weights. The fuzzy integral approach provides an importance measure for each subset of available information sources [47].

However, each of these methods has their own limitations and drawbacks. First of all, they are either computationally complex or their weight estimation capabilities are limited. Furthermore, the selection process is usually class-common, which means, the same set of features are used for all classes. In addition, they usually evaluate the features individually, which may cause loss of the information that is obtained from the correlation between features. In this study, we propose a timely efficient and effective way for modality weighting, which exploits the class-specific information for modalities and enables the use of correlation between modalities.

2.2 Feature selection/weighting approaches

In addition to the above given methodologies, the *feature selection/weighting* studies in machine learning literature provide many different approaches for feature selection. Existing methods in the literature are categorized as filter or wrapper methods. Filter methods assess the relevance of features by looking only at the intrinsic properties of the data, whereas in wrapper methods the performance of a learning algorithm is used to evaluate the fitness of the feature subsets in the feature space. Filter methods are usually computationally much more efficient than wrapper methods; however, wrapper methods usually provide solutions closer to the optimal solution. Another weakness of the filter methods is that they usually evaluate the features individually. Thus, the quality of combined feature subsets is not analyzed and the correlation information between features cannot be exploited. Some well-known

filter methods are Information Gain [13], Gain Ratio [38], Correlation-based feature selection (CFS) [11], Chi-squared selection and RELIEF [23]. Some well-known wrapper methods are as follows: exhaustive search [16], sequential forward selection (SFS) [24], sequential backward elimination (SBE) [24], Plus q take-away r [8], simulated annealing and genetic algorithms. For more detailed discussions, interested readers can refer to [10, 16, 42] and the references therein.

With a machine learning point of view, the modality weighting problem is similar in nature to the feature weighting problem and, thus, efficient and effective feature weighting solutions can be applied for the modality weighting problem. However, it is not trivial to apply the available methods to the modality weighting problem due to several differences between the problems. The most crucial difference is the intrinsic dimensionality of modalities. In feature weighting, the input is a feature vector, which is a multi-dimensional vector of numerical/nominal values representing some pattern. Besides, in modality weighting, the input is multiple feature vectors. Feature weighting methods rank the dimensions of the input feature vector by assigning a weight for each dimension, whereas in modality weighting, the intrinsic dimensions of each modality are not the main concern. Modality weighting methods rank the modalities by assigning weights to each modality as a black-box. Still, an early combination (i.e., concatenation) of available modalities corresponds to a single multi-dimensional feature, which makes any feature weighting method applicable. However, the majority of the multimodal fusion studies employ late fusion approaches, in which each modality is processed separately. Thus, ranking the available modalities, instead of the intrinsic high-dimensional features, is still a crucial need for the multimodal information fusion. In addition, another concern may be performing some feature selection operations for each of the modalities. However, it can be assumed as a preprocessing step before modality selection/weighting. The second difference between feature and modality weighting is the values of the inputs. The inputs of a multimodal fusion system are not necessarily feature values; the most frequently utilized inputs in state-of-the-art fusion studies are the prediction scores. Thus, the modality selection approach should work under any of these inputs. One more issue related with the input values is that most of the frequently applied methods (e.g. Information Gain, Chi-squared) require the feature values to be binary or discretized. However, discretization of the modalities makes the process computationally complex, since each modality is represented by a multi-dimensional feature.

An applicable idea to deal with these problems is to work in a distance-based metric space, instead of in a feature space. Utilizing a distance space solves the intrinsic dimensionality problem of multiple modalities by converting the multi-dimensional feature values of a modality to a uni-dimensional distance value. Furthermore, it enables handling the scores, ranks and decisions after converting them into applicable dissimilarity values with appropriate conversion functions. Thus, we focus on a RELIEF-based algorithm, which generates the weights based on the distances between training samples. Being a filter approach, RELIEF avoids an exhaustive search and provides computationally a more efficient solution than the wrapper methods. Besides, it takes the context into account, exploits correlation information between features and thus usually performs better than the filter approaches. Details of the family of RELIEF algorithms are given below.

2.3 RELIEF algorithms

Among the available feature selection and weighting methods, the RELIEF algorithm [23] is among the most successful. It is a simple and effective way for feature selection [6]. In addition, RELIEF does not make a conditional independence assumption for features, as many other feature selection methods do, and can correctly estimate the quality of features with dependencies [41]. The key idea of RELIEF is to estimate weights for each feature according to their ability to discriminate between neighboring training samples by iterating through randomly selected instances in the training space. In [46], Sun presents the discrimination-based approximation of RELIEF with a novel mathematical interpretation from the optimization perspective, and shows that RELIEF utilizes a margin-based nonlinear classifier for searching useful features.

Algorithm 1: Basic RELIEF

Input: list of features $\mathcal{F} = \langle f_i \rangle_{i=1}^{n}$,
number of iterations m,
set of training instances $\mathcal{D} = \{d_j\}_{j=1}^{t}$
Output: the weight vector W of estimations for the qualities of features

```
1  begin
2      for i ← 1 to n do                    //for each feature in F
3          W[i] ← 0;
4      end
5      for j ← 1 to m do
6          r ← randomInstance(D);
7          ⟨H, M⟩ ← findNearestHitMiss(r, D);
8          for i ← 1 to n do                //for each feature in F
9              W[i] ← W[i] − diff(f_i,r,H)/m + diff(f_i,r,M)/m;
10         end
11     end
12 end
```

The basic RELIEF algorithm is given in Algorithm 1. The weight estimation function in Line 9 exploits the discrimination capability. The algorithm selects a random

sample r, one Near-Hit H (nearest neighbor with the same class with the random sample) and one Near-Miss M (nearest neighbor with a different class with the random sample) and distances between them are calculated. In this calculation, the distance between instances in different classes indicates a discrimination between classes, so diff(f_i, r, M) increases the weight. Inversely, distance between instances with the same class inhibits discrimination, so diff(f_i, r, H) decreases the weight.

Considering several deficiencies of the basic RELIEF algorithm, Kononenko [29] proposes several extensions for RELIEF: RELIEF-A uses k nearest neighbors instead of one and averages the contribution of k nearest instances to eliminate the effect of noisy instances; RELIEF-B, RELIEF-C and RELIEF-D extend the use of *diff* function to handle incomplete datasets; RELIEF-E and RELIEF-F improve the weight update function for multi-class problems. Other well-known extensions for RELIEF are as follows: Sikonja et al. [40] propose RRELIEF-F for handling regression problems. In [43], Sikonja proposes using k-d trees for the selection of nearest neighbors to decrease the computation complexity of the RELIEF algorithm. In [46], Sun introduces Iterative RELIEF (I-RELIEF), which uses an Expectation Maximization algorithm to eliminate outlier data. Also, Liu et al. [30] try to eliminate outlier data and propose using selective sampling by means of a modified kd-tree instead of random sampling (at Line 6 in Algorithm 1).

Among the available extensions of the RELIEF algorithm, RELIEF-F is the most widely utilized. RELIEF-F enables working with multi-class problems, by selecting k nearest misses for each class. Thus, the RELIEF-F algorithm updates Line 7 of Algorithm 1 with the following;

$$\langle \mathcal{H}, \mathcal{M} \rangle \leftarrow \text{findNearestHitsMisses}(r, \mathcal{D}, k, \mathcal{C});$$

where k is the number of nearest neighbors, and $\mathcal{C} = \langle c_u \rangle_{u=1}^{s}$ is the list of classes. $\mathcal{H}$ is the k-sized list of hit instances, where $\mathcal{H}_v$ denotes the vth nearest hit instance. Besides, $\mathcal{M}$ is the $s \times k$ sized matrix, where $\mathcal{M}_v^u$ represents the vth miss instance for class $c_u \in \mathcal{C}$. In addition, the weight estimation function in Line 9 is also updated as given below

$$W[i] \leftarrow W[i] - \sum_{v=1}^{k} \frac{\text{diff}(f_i, r, \mathcal{H}_v)}{m \cdot k} + \sum_{\substack{u=1 \\ c_u \neq C(r)}}^{s} \left(\frac{P(c_u)}{1 - P(C(r))} \sum_{v=1}^{k} \frac{diff(f_i, r, \mathcal{M}_v^u)}{m \cdot k} \right) \quad (1)$$

where, $P(c_u)$ represents the prior probability of class c_u, and $C(r)$ indicates the class of sample r.

In this study, we utilize RELIEF-F for multimodal feature selection in multimedia retrieval, which has not been done before, to the best to our knowledge. Using the RELIEF-F algorithm for multimodal feature selection on multimedia data enables us to identify some weaknesses of the RELIEF-F algorithm. Thus, we extend the RELIEF-F algorithm due to the aspects discussed in Sect. 1.

2.4 Complexity analysis

The feature selection/weighting problem is known as NP-hard, in terms of the number of features $n = |\mathcal{F}|$. An exhaustive search for generating all possible subsets requires $O(p^n)$ actions, where p is the number of assignable weights ($p = 2$ for binary selection). Considering that an exhaustive search is a wrapper method, it requires an evaluation for each of these subsets. Assuming a simple evaluation similar to RELIEF, based on the similarities/distances between m randomly selected instances to all t training instances, the total complexity of the exhaustive search becomes $O(m \cdot t \cdot n \cdot p^n)$. Moreover, if a class-specific approach is applied, the total complexity becomes $O(m \cdot t \cdot s \cdot n \cdot p^n)$, where s is the number of classes ($s = |\mathcal{C}|$).

On the other hand, the RELIEF algorithms provide solutions in polynomial time. The complexity of the basic RELIEF algorithm is $O(m \cdot t \cdot n)$, considering that the most complex operation is the selection of the nearest hit and miss instances since the distances between r and the other training instances should be calculated for each feature, which requires $O(t \cdot n)$ comparisons. Different from the basic RELIEF algorithm, the complexity of RELIEF-F depends on the number of nearest neighbors (k). If we use a priority queue, which is implemented with a heap structure, for the selection of k nearest neighbors, where the construction of the heap is $O(t)$ and the retrieval of k neighbors from each class is $O(k \cdot s \cdot \log t)$; the total complexity of selecting k nearest hits/misses becomes $O(m \cdot t \cdot n + m \cdot k \cdot s \cdot \log t + m \cdot k \cdot s \cdot n)$. In this equation, the first term is for the distance calculation, the second is for selecting nearest instances from the heap and the last is for the weight calculation [Eq. (1)]. If the dataset is a balanced one and the value of k is considerably small with respect to t, then the computational complexity of the RELIEF-F algorithm becomes the same as the basic RELIEF algorithm ($O(m \cdot t \cdot n)$). A computationally better solution can be obtained by utilizing k-d trees for improving the nearest hit and miss selection process ($O(n \cdot t \cdot \log t)$).

3 RELIEF-MM: modality weighting approach for multimedia data

To benefit from the simplicity and effectiveness of RELIEF algorithms, we propose a RELIEF-based multimodal feature selection solution, by extending the RELIEF-F algorithm. Below, each of our extensions is presented in a separate subsection.

3.1 Class-specific feature weighting

Multimedia retrieval requires dealing with a high number of different queries, where each query usually denotes a different concept occurring in videos. Thus, multimedia retrieval is accepted as a multi-class classification problem with a high number of classes, where each class is a concept occuring in videos. In addition, the variety of such concepts is so wide that they can be associated with different sets of features/modalities. In other words, each concept can be represented better with different features specific to the concept [50, 55]. For instance, an *explosion* concept can be represented relatively more accurately by the audio modality, whereas it is better to utilize visual modality for detecting a *mountain* concept. Similarly, it can be easier to recognize a *meeting* concept using both the visual and the audio modalities. Hence, a class-specific modality weighting approach is inevitable to be used in the multimedia retrieval systems, to handle the high number of classes/concepts. However, the traditional feature selection methods, including the RELIEF-F algorithm, propose class-common solutions in which the selection is performed independently from the classes.

Based on the motivation above, we propose a substantial extension on RELIEF-F, which is converting it to a class-specific solution. Since the RELIEF-F algorithm iterates over available training samples to obtain the final value of the modality weights, grouping the training samples according to their classes and processing samples of each class separately can achieve a class-specific solution.

Assuming that we iterate over m training samples $\mathcal{R} = \{r_i\}_{i=1}^{m}$, which are randomly selected from the set of all training samples $\mathcal{D} = \{d_i\}_{i=1}^{t}$, the final weight of f_i can be formalized as;

$$W(f_i) = \sum_{j=1}^{m}\left[-\sum_{v=1}^{k}\frac{\text{diff}(f_i, r_i, \mathcal{H}_v)}{m\cdot k} + \sum_{\substack{u=1\\ c_u\neq C(r_j)}}^{s}\left(\frac{P(c_u)}{1-P(C(r_j))}\sum_{v=1}^{k}\frac{\text{diff}(f_i, r_j, \mathcal{M}_v^u)}{m\cdot k}\right)\right] \quad (2)$$

Here, we can rewrite Eq. (2) as in Eq. (4), by assigning the effect of one training sample r_j on the final weight calculation of modality f_i into ΔW_j^i (Eq. (3)).

$$\Delta W(f_i, r_j) = -\sum_{v=1}^{k}\frac{\text{diff}(f_i, r_j, \mathcal{H}_v)}{k} + \sum_{\substack{u=1\\ c_u\neq C(r_j)}}^{s}\left(\frac{P(c_u)}{1-P(C(r_j))}\sum_{v=1}^{k}\frac{\text{diff}(f_i, r_j, \mathcal{M}_v^u)}{k}\right) \quad (3)$$

$$W(f_i) = \frac{1}{m}\sum_{j=1}^{m}\Delta W(f_i, r_j) \quad (4)$$

If the samples in $\mathcal{R}$ are grouped according to the class they belong to, we can represent the final weight of f_i as in Eq. (5). Here, each group is represented by $\mathcal{R}_u = \{r \,|\, r \in \mathcal{R} \wedge C(r) = c_u\}$, where $\mathcal{R} = \bigcup\{\mathcal{R}_u\}_{u=1}^{s}$ and $C(r)$ represents the class of r.

$$W(f_i) = \sum_{u=1}^{s}\left(P(c_u)\frac{1}{|\mathcal{R}_u|}\sum_{r\in R_u}\Delta W(f_i, r)\right) \quad (5)$$

Here, we can define a class-specific weight $\omega(c_u, f_i)$ as in Eq. (6).

$$\omega(c_u, f_i) = \frac{1}{|R_u|}\sum_{r\in R_u}\Delta W(f_i, r) \quad (6)$$

The original class-common weight estimation function of RELIEF-F can also be rewritten as in Eq. (7), in terms of class-specific weights.

$$W(f_i) = \sum_{u=1}^{s}P(c_u)\omega(c_u, f_i) \quad (7)$$

As seen in Eq. (7), RELIEF-F estimates the weights of the features by taking a weighted average of all class-specific weights and, thus, cannot reflect the characteristics of each class separately. Instead, we here propose to use weight estimations of each class separately. Consequently, this class-specific adaptation of RELIEF-F is presented with Algorithm 2.

Algorithm 2: Class-Specific Adapt. of RELIEF-F

Input: list of features $\mathcal{F} = \langle f_i\rangle_{i=1}^{n}$,
number of iterations m,
set training instances $\mathcal{D} = \{d_j\}_{j=1}^{t}$,
list of classes $\mathcal{C} = \langle c_u\rangle_{u=1}^{s}$,
number of nearest neighbors k

Output: the weight matrix ω of estimations for the qualities of features

```
begin
    for u ← 1 to s do                          //for each class in C
        for i ← 1 to n do                      //for each feature in F
            ω[u][i] ← 0;
        end
    end
    for u ← 1 to s do                          //for each class in C
        D_u ← getClassInstances(D, c_u);
        m' ← m · P(c_u) ;      // P(c_u)=size(D_u)/size(D)
        for j ← 1 to m' do
            r ← randomInstance(D_u);
            ⟨H, M⟩ ← findNearestHitsMisses(r, D, k, C);
            for i ← 1 to n do                  //for each feat. in F
                ω[u][i] ← ω[u][i] − Σ_{v=1}^{k} diff(f_i, r, H_v)/(m'·k)
                    + Σ_{u'=1, u'≠u}^{s} ( P(c_u')/(1−P(c_u')) Σ_{v=1}^{k} diff(f_i, r, M_v^u')/(m'·k) );
            end
        end
    end
end
```

We should also note that converting the original RELIEF-F algorithm into a class-specific version does not change computational complexity, since k hit/miss selection procedures and the number of processed samples do not change. As a result of having the same computational complexity, the approach can be accepted as scalable in terms of the number of class, since the computational complexity of the algorithm is linearly proportional to the number of classes (as given in Sect. 2.4).

3.2 Multi-labeled/noisy datasets

In a typical multimedia retrieval task, each multimedia document (i.e., shot or video) is usually associated with a number of different semantic concepts. This situation reveals the problem of the multi-label feature selection, in which each sample is associated with multiple labels. In multimedia data, the multi-labeled characteristic of the data can be originated from either having more than one concept for each multimedia document in any single modality contained (e.g. having both an *airplane* and a *mountain* in a visual scene, as given in Fig. 1), or containing different concepts in different modalities of the same document (e.g. having an *explosion* sound in the audio-modality and *military*-related vehicles in the visual modality at the same moment of the video).

In multi-label datasets, the samples are not mutually exclusive in terms of assigned labels, thus the discrimination of the samples between class labels becomes complicated. The discrimination of the samples between the retrieval classes is crucial to an effective feature selection. However, state-of-the-art studies accept the problem as a structural one, deal with converting the multi-labeled dataset into a single-labeled one for use with traditional feature selection methods [7, 28], and leave aside the cognitive aspect of the problem, which is also an important part of the problem. Here, the 'structural' side of the problem refers to the impossibility of using traditional learning/selection methods with the multi-labeled dataset due to the structure of the dataset, whereas the 'cognitive' side denotes the loss of the discrimination capability for learning. In this study, we regard both issues depicted and propose a two-step solution.

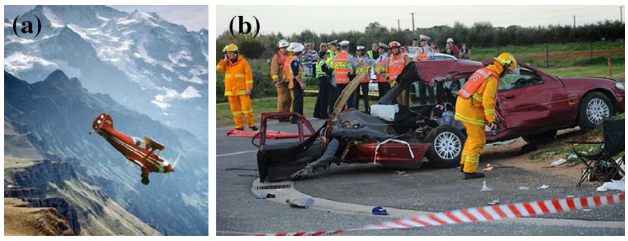

Fig. 1 Examples for multi-labeled shots. **a** *airplane* and *mountain*, **b** *car*, *accident*, *people* and *street*

As the first step, we consider that it is not possible to use the RELIEF-F algorithm directly for a multi-label dataset, since having multi-labeled samples makes the nearest hit/miss selection procedure ambiguous. For instance, we need a solution to select the nearest hits/misses of a random instance with two different class labels, or a nearest item is labeled with two different classes. Thus, we first look into the state-of-the-art transformation methods. The most popular transformation methods in the literature are random assignment (RA), binary relevance (BR), label power set (LP) and pruned problem transformation (PPT) [48]. In the RA approach, a multi-labeled sample is randomly assigned to one of its classes. In BR, the dataset is transformed into $|\mathcal{C}|$ single-label datasets, where $\mathcal{C} = \langle c_u \rangle_{u=1}^{s}$ is the list of available classes. In any $\mathcal{D}_u = \{d \,|\, d \in \mathcal{D} \wedge C(d) = c_u\}$ of these datasets, the samples are labeled in a binary form, depending on whether a sample d is associated with class c_u or not. In LP, the basic idea is to convert the set of classes $\mathcal{C}$ into $\mathcal{C}'$ such that $\mathcal{C}'$ is the power set of $\mathcal{C}$ ($\mathcal{C}' = \mathcal{P}(\mathcal{C})$). PPT is an improvement on LP, where unused subsets are removed from $\mathcal{C}'$. However, using any of these approaches causes loss of either the effectiveness or the efficiency of the algorithm. Using RA makes the process nondeterministic and also loses a large amount of valuable information due to the random class selection; thus results in an ineffective solution. On the other hand, although BR, LP and PPT are potentially good solutions to prevent information loss, the process becomes computationally complex. Hence we focus on an alternative solution that enables use of the RELIEF-F algorithm for multimedia data and does not increase the computational complexity.

Assuming that c_i, c_j and c_k are three classes different from each other, we decompose the multi-label problem for RELIEF-F into three cases:

- *Case-1* A random sample x is associated with both classes c_i and c_j. In this case, it is not clear which class will be accepted for hits and misses.
- *Case-2* Random sample x is labeled with c_i, and y is one of the nearest neighbors of x. If y is labeled with both c_i and c_j, it is unclear whether such a neighbor instance is a hit or a miss.
- *Case-3* Random sample x is labeled with c_i, and y is one of the nearest neighbors of x. The neighbor instance y is labeled with both c_j and c_k. In this case, it is clear that y is a miss. However, it is not clear which class of miss it is.

We first start with a BR-like method, which is very compatible with the class-specific extension of RELIEF-F discussed in Sect. 3.1. Different from BR, we do not generate $|\mathcal{C}|$ number of binary valued datasets. In

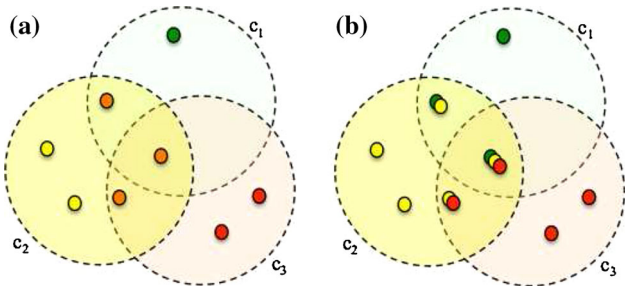

Fig. 2 Transforming multi-labeled samples into multiple single-labeled samples. *Small green*, *yellow* and *red circles* denote c_1, c_2 and c_3 instances, respectively. *Orange circles* in (**a**) are multi-labeled instances, each of which is transformed into multiple single-labeled instances in (**b**) (color figure online)

accordance with the class-specific extension, an intuitive way to deal with these cases is to transform each multi-labeled sample into multiple single-labeled samples with the same feature values but different classes (as illustrated in Fig. 2), and group the samples according to the class that they belong to. Thus we divide the training dataset into $|\mathcal{C}|$ number of subsets, each having the samples of a different class. During the execution of the algorithm, the random samples are selected among each subset iteratively, and finding the associated class of a sample is not problematic anymore, even if it is a multi-labeled sample originally. Thus, Case-1 is discarded. Actually, the use of a class-specific extension helps to prevent Case-1. For handling Case-2 and Case-3, the same transformation as with Case-1 is applicable. For Case-2, any multi-labeled neighbor instance y is replicated and transformed into y_{c_i} and y_{c_j}. Then, y_{c_i} is used as a hit instance and y_{c_j} is used as a miss instance, which actually means y is used both as a hit and a miss instance. Similarly, for Case-3, y is transformed into y_{c_j} and y_{c_k}, then y_{c_j} is used as a miss instance for class j, whereas y_{c_k} is used for class k.

Although this solution is an efficient approach to deal with the multi-labeled structure of training data and does not cause information loss as in BR transformation, it is still possible to lose some information due to the use of the same neighboring instances as both hits and misses (i.e., Case-2). Considering the weight estimation function of RELIEF-F given in Eq. (2), while calculating the weight of modality f using random sample x, the effect of a neighbor hit instance y is as follows:

$$\delta_{\text{hit}} = -\frac{\text{diff}(f,x,y)}{m \cdot k} \tag{8}$$

However, if the neighbor instance y is a multi-labeled one as in Case-2, the same instance is used both as a hit and a miss instance. Thus, the net effect of the neighboring hit instance becomes:

$$\delta'_{\text{hit}} = \left(\frac{P(c_j)}{1-P(c_i)} - 1\right)\frac{\text{diff}(f,x,y)}{m \cdot k} \tag{9}$$

In other words, the effect of the hit instance is decreased because of being a multi-labeled instance. The worst case of this situation, although practically impossible, occurs when the instance is labeled with all available classes. In such a situation, the effect of the instance equals to zero. In [28], Kong et al. propose to ignore the instances of Case-2, which is practically the same as assuming the situation is always the worst case. In our approach, we do not ignore such instances, since they may still provide some valuable information as long as the situation is not the worst case. We accept the decrease in the effect of the hit instances as a sort of noise and loss in the discrimination capability of the features.

In this aspect, we also consider the effect of noise in multimedia data. In addition to the fact that the multimedia data have an expected internal noise, the way we model the multimedia data can create an artificial noise. Since the multimedia data is usually large—even huge—some sub-sampling (i.e., using shots and keyframes instead of each particular frame) is done before processing it. The extracted features represent only subsamples from the video, whereas the ground truth labels are based on the full content of the video. Such a situation makes the evaluation of features complicated and eventually some of the ground truth instances appear as noisy instances. Similar to the multi-label issue, having noise in the samples prevents a correct discrimination between classes. In addition, depending directly on the distances between training instances affects the performance of the algorithm negatively, considering the noisy instances.

Consequently, the second step of our approach is based on strengthening the feature weighting mechanism of RELIEF-F. Thus, we introduce two new factors for the calculation of the weights, in addition to the discrimination capability: the representation and reliability characteristics. Having additional components in the weight calculation makes the algorithm less dependent on the discrimination capability, and provides better estimations. Hence, the class-specific weight of a feature, which was previously defined in Eq. (6), is updated as the following;

$$\varpi(c_u,f_i) = \begin{cases} (\omega(c_u,f_i))^{\alpha} \cdot \gamma(c_u,f_i) \cdot \eta(c_u,f_i), & \text{if } \omega_f^c > 0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where ω, γ and η functions provide the discrimination, representation and reliability-based weights, respectively. In addition, α is an experimental constant for tuning. Considering that RELIEF-F-based weights are in [−1, 1] and weights smaller than zero denote irrelevant features,

we discard these by assigning zero. The proposed functions are discussed in detail below.

3.2.1 Discrimination-based weight

The discrimination-based weight ($\omega(c_u, f_i)$) refers to the weight calculated using the data from all available classes with an aim to discriminate between those classes. The calculation of $\omega(c_u, f_i)$ is basically accepted as the way to calculate class-specific RELIEF-F (Eq. (6)).

3.2.2 Representation-based weight

The representation-based weight ($\gamma(c_u, f_i)$) refers to the weight calculated using the data only from any single class, with an aim to represent that class independent of other classes. To measure its effectiveness using only its characteristics and calculate such a weight, we assume that we can isolate the samples of a particular class from other classes. Here, isolation means that any sample labeled with other classes is always at a farthest location. Applying this idea to the class-specific RELIEF-F weight calculation gives the following: The distance of a random sample to any of its nearest misses always equals to 1 (note that $\text{diff}(f, x, y) \in [0, 1]$). Hence, the representation-based weight becomes the following:

$$\gamma(c_u,f_i) = \frac{1}{|\mathcal{R}_u|}\sum_{r\in\mathcal{R}_u}\left[1 - \sum_{v=1}^{k}\frac{\text{diff}(f_i, r, \mathcal{H}_v)}{m\cdot k}\right] \tag{11}$$

Equation (11) can also be interpreted as the complement of the mean distance of a class to itself, so the weight of a feature is inversely proportional to the mean distance of the class to itself. Here, the mean distance of a class to itself is the average of all distances from each sample of a class to its k neighbor hits. It is expected for a particular class that the features with lower mean distance values represent the class better. Thus, $\gamma(c_u, f_i)$ is a sound metric to estimate the representation capability of a feature.

3.2.3 Reliability-based weight

Reliability-based weight ($\eta(c_u, f_i)$) refers to the weight calculated using accuracy with respect to a feature for a particular class, with an aim to see whether it is reliable for that class or not. The idea of using the accuracies of features is based on the theoretical analysis of Fumera et al. [9]. Fumera et al. work on a late fusion scheme and show that the weight of a classifier for feature f_i should be inversely proportional to the error of the classifier.

Considering that RELIEF-MM is a filter method, and classification results are not available during the feature weighting, we propose to estimate the accuracy of each feature by comparing the intra-class distance of each class with the inter-class distances to other classes. The intra-class distance is defined as the mean distance of the samples in c_u to their nearest k hits, whereas the inter-class distance is the mean distance of the samples in c_u to their nearest k misses from each different class $c_{u'} \neq c_u$. It is important for a feature to give the lowest distance values for the instances in a class which is the same as the class of the query instances. Thus, $\eta(c_u, f_i)$ provides an estimation for reliability by finding the number of inter-class distances (by means of different classes) that has a larger value than the intra-class distance. The formal representation of $\eta(c_u, f_i)$ is given in Eq. (12)

$$\eta(c_u,f_i) = \frac{\left|\left\{\mu(c_u, c_{u'}, f_i)\big|_{\wedge\ \mu(c_u,c_{u'},f_i) > \mu(c_u,c_u,f_i)}^{c_{u'}\in\mathcal{C}-\{c_u\}}\right\}\right|}{s-1} \tag{12}$$

$$\mu(c_u, c_{u'}, f_i) = \frac{1}{|\mathcal{R}_u|}\sum_{r\in\mathcal{R}_u}\left[\sum_{v=1}^{k}\left(\text{diff}(f_i, r, \mathcal{N}_v^{c_{u'}})\right)\right] \tag{13}$$

where $\mathcal{N}_v^{c_{u'}}$ is the vth $c_{u'}$-labeled nearest instance of sample r. Thus, $\mu(c_u, c_u, f_i)$ refers to the mean distance to k hits (intra-class distance), whereas $\mu(c_u, c_{u'}, f_i)$ with $c_{u'} \in \mathcal{C} - \{c_u\}$ is the mean distance to the k misses of any other class (inter-class distance).

3.3 Unbalanced datasets

In multimedia datasets, some of the concepts occur less frequently than others, which causes the annotated training data to be unbalanced among different classes. We can consider the occurrences of *flag* versus *car* objects through a random video as an example; a *flag* object usually occurs less often than a *car* object. Thus, the number of *car* samples is usually larger than the number of *flag* samples. One important consequence of frequent occurrence is having more representative and descriptive data than the infrequent concepts, e.g. it is possible to find several different models and colors of *car* samples, but it is hard to find the variations of *flag* samples. Hence, having an unbalanced dataset may prevent an adequate learning process.

Although the unbalanced dataset problem is usually discussed in the scope of classification and learning [4], the RELIEF-F algorithm, as a feature selection method, is also negatively affected by unbalanced data. The reason why RELIEF-F is affected by the imbalance in the data is the use of k nearest neighbors during the weight calculation. As discussed in Sect. 2.3, RELIEF-F uses average distance to k nearest neighbors while calculating the weights, to eliminate the effect of outlier data. However, the placement of training samples in the multi-dimensional space and the amount of outliers are highly data-dependent, and can be very different for different domains and classes. Thus, we

point out that selecting k number of nearest neighbors for every class is not a fair preference, when each class has a different number of samples. Using the same k number of neighbors hinders the use of an equal amount of information from all classes. For instance, a certain value of k may provide for the acquiring of all available patterns of a particular class. However, for another class, the same k value may provide for the acquisition of only a small ratio of the available patterns. The situation is not different if we consider the outlier data. Selecting the same number of neighbor instances from different classes (each of which has a different number of samples) may result in different ratios of outlier data for each class.

Considering the above given issues, we propose to select the value of k dynamically, i.e., a class-specific k value. However, enabling a class-specific k selection makes the process more complicated, despite the potential improvement in the estimation of feature weights. Thus, we propose another promising idea; using the k value as a certain ratio of sample count in a class. By employing such an idea, the k value of class c_u can be calculated by

$$k_u = k_R \cdot |\mathcal{D}_u|, \tag{14}$$

where $\mathcal{D}_u = \{d \,|\, d \in \mathcal{D} \wedge C(d) = c_u\}$ is the set of training instances with class c_u, and $k_R \in [0, 1]$ is the nearest neighbor selection ratio, which is defined independently of the classes.

A weak point of this idea is that it requires us to assume approximately the same ratio of noise for all classes. Yet, this assumption can be practically applicable, considering that the datasets mostly do not suffer from the outliers because of mislabeling, but because of complexities related with the internal characteristics of video data, such as lighting variations, camera motion, occlusion, and noise in the sensed data. Mislabeling is a human-oriented noise, in which we cannot assume that the ratio of outliers are equal for different classes (e.g. it may be harder to annotate the samples with less frequently occurring classes). However, we can assume that the complexities in the video occur approximately in the same ratio for any class, especially when we have a broad range of videos.

3.4 The final algorithm

The finalized RELIEF-MM algorithm including all of the extensions that we describe above is given in Algorithm 3. In order not to make the presentation of the algorithm more complex, some of the calculation including loops are represented by some mathematical functions (e.g. sum operations). The $\varpi(c_u, f_i)$ in Eq. (10) is represented with W matrix in the algorithm. The other parameters for calculating $\varpi(c_u, f_i)$ are given as they are given in Eq. (10).

Algorithm 3: RELIEF-MM

Input: list of features $\mathcal{F} = \langle f_i \rangle_{i=1}^{n}$,
number of iterations m,
set of training instances $\mathcal{D} = \{d_j\}_{j=1}^{t}$,
list of classes $\mathcal{C} = \langle c_u \rangle_{u=1}^{s}$,
nearest neighbor selection ratio k_R,
tuning constant α
Output: the weight matrix W

begin
  // Initialization
  for $u \leftarrow 1$ **to** s **do** //for each class in $\mathcal{C}$
    for $i \leftarrow 1$ **to** n **do** //for each feature in $\mathcal{F}$
      $\omega[u][i] \leftarrow 0$;
      $\gamma[u][i] \leftarrow 1$;
      $\eta[u][i] \leftarrow 0$;
      for $u' \leftarrow 1$ **to** s **do** //for each class in $\mathcal{C}$
        $\mu[u][u'][f] \leftarrow 0$;
  // Calculations
  for $u \leftarrow 1$ **to** s **do** //for each class in $\mathcal{C}$
    $\mathcal{D}_u \leftarrow getClassInstances(\mathcal{D}, c_u)$;
    $k_u \leftarrow k_R \cdot size(\mathcal{D}_u)$;
    $m' \leftarrow m \cdot P(c_u)$; // P(c_u)=size($\mathcal{D}_u$)/size($\mathcal{D}$)
    for $j \leftarrow 1$ **to** m' **do**
      $r \leftarrow randomInstance(\mathcal{D}_u)$;
      $\langle \mathcal{H}, \mathcal{M} \rangle \leftarrow findNearestHitsMisses(r, \mathcal{D}, k_u, \mathcal{C})$;
      for $i \leftarrow 1$ **to** n **do** //for each feature in $\mathcal{F}$
        $\omega[u][i] \leftarrow \omega[u][i] - \sum_{v=1}^{k_u} \frac{diff(f_i, r, \mathcal{H}_v)}{m' \cdot k_u} + \sum_{\substack{u'=1 \\ u' \neq u}}^{s} \left(\frac{P(c_{u'})}{1 - P(c_u))} \sum_{v=1}^{k_u} \frac{diff(f_i, r, \mathcal{M}_v^u)}{m' \cdot k_u} \right)$;
        $\gamma[u][i] \leftarrow \gamma[u][i] - \sum_{v=1}^{k_u} \frac{diff(f_i, r, \mathcal{H}_v)}{m' \cdot k_u}$;
        $\mu[u][u][i] \leftarrow \mu[u][u][i] + \sum_{v=1}^{k_u} \frac{diff(f_i, r, \mathcal{H}_v)}{m' \cdot k_u}$;
        for $u' \leftarrow 1$ **to** s **do** //for each class in $\mathcal{C}$
          if $u' = u$ **then** continue;
          ;
          $\mu[u][u'][i] \leftarrow \mu[u][u'][i] + \sum_{v=1}^{k_u} \frac{diff(f_i, r, \mathcal{M}_v^u)}{m' \cdot k_u}$;
  // Finalization
  for $u \leftarrow 1$ **to** s **do** //for each class in $\mathcal{C}$
    for $i \leftarrow 1$ **to** n **do** //for each feature in $\mathcal{F}$
      for $u' \leftarrow 1$ **to** s **do** //for each class in $\mathcal{C}$
        if $u' \neq u \wedge \mu[u][u'][i] > \mu[u][u][i]$ **then**
          $\eta[u][i] \leftarrow \eta[u][i] + \frac{1}{s-1}$
      if $\omega[u][i] > 0$ **then**
        $W[u][i] \leftarrow (\omega[u][i])^{\alpha} \cdot \gamma[u][i] \cdot \eta[u][i]$;
      else
        $W[u][i] \leftarrow 0$;

The RELIEF-MM algorithm consists of three parts. First, the parameters of the weight estimation function (ω, γ and η) are initialized. Second, these parameters are updated iteratively by encountering random training samples in the total of m. This process is performed

separately for each class, thus some percentage of m (proportional to the prior probability of each class) is used for each class. Lastly, the calculated parameters are used to find the final values of weight estimations for each feature and class.

Here, it should be noted that the original RELIEF-F algorithm is an online algorithm, which means that the algorithm processes training instances one-by-one in a serial fashion, and can give an output after processing each instance. However, the RELIEF-MM algorithm presented in Algorithm 3 is offline, since the final weights are calculated as a batch instruction. Our choice eliminates further complexity in the algorithm. Yet, it is fairly straightforward to convert Algorithm 3 into an online version by moving the block between lines 24–32 into the *for* loop between lines 13–23, as the last instruction.

3.4.1 Complexity analysis

We assume that n denotes number of features ($n = |\mathcal{F}|$), m denotes number of iterations, k_R denotes nearest neighbor selection ratio, s denotes number of classes ($s = |\mathcal{C}|$) and t denotes number of training instances ($t = |\mathcal{D}|$). Considering the Algorithm 3, RELIEF-MM includes three main loops for initialization, calculation and finalization.

The first main loop (lines 2–8) initializes the parameter matrices and takes;

$$L_{init} = O(s^2 \cdot n) \tag{15}$$

The second main loop (lines 9–23) is basically used for iterating over m instances from any class in C. Inside the loop, there are three operations, which are not $O(1)$; (1) filtering c_u-labeled instanced in $\mathcal{D}$, in line 10, (2) selection of hits and misses, in line 15, (2) weight parameter calculations, between lines 16–23. The first operation is performed once for each class in $\mathcal{C}$. The operation checks whether each instance in $\mathcal{D}$ is labeled with c_u or not, and takes $O(t)$ time. The second operation includes the distance calculation from random instance to all instances in $\mathcal{D}$, heap construction using the distances and neighbor selection from the heap. This process is similar to the case for RELIEF-F in Sect. 2.4, and takes $O(t \cdot n + t + k_u \cdot s \cdot \log t)$ steps. The third operation contains four instructions and is repeated for each feature. It takes $O(2 \cdot k_u \cdot s \cdot n + 2 \cdot k_u \cdot n)$ steps in total. The bounds for the second and third operations include a k_u term which is dependent on a class c_u. In other words, for each class $c_u \in \mathcal{C}$, the k_u value gets a different value based on Eq. (14). Considering that these operations are repeated for m' instances of s number of classes, the total complexity of these three operations becomes;

$$\begin{aligned}
&= \sum_{u=1}^{s} \left[O\Big(mP(c_u)(tn + k_R|\mathcal{D}_u| s \log t + k_R|\mathcal{D}_u| sn) \Big) \right] \\
&= O\Big(m\,t\,n \;+\; m\,k_R\,s \log t \sum_{u=1}^{s} \big(P(c_u)|\mathcal{D}_u|\big) \\
&\quad + m\,k_R\,s\,n \sum_{u=1}^{s} \big(P(c_u)|\mathcal{D}_u|\big) \Big)
\end{aligned} \tag{16}$$

Considering that $P(c_u)$ is the prior probability of the classes and can be calculated using the instance counts in each class, the summation term $\sum_{u=1}^{s}(P(c_u)|\mathcal{D}_u|)$ in Eq. (16) can be rewritten as $\sum_{u=1}^{s}(|\mathcal{D}_u|^2/t)$. The minimum value of this term is obtained when the dataset is balanced. For such a case, the term equals to t/s. The maximum value of the term is obtained with an unbalanced dataset, where one of the classes contains all t instances and the other classes contain no instances, although this is practically impossible. In this case, the term equals to t. Thus the complexity bounds for the term is $\Omega(t/s)$ and $O(t)$. By applying this result in Eq. (16), the total complexity of the second main loop becomes

$$L_{\text{calc}} = O\big(m \cdot t \cdot n + m \cdot (k_R \cdot t) \cdot s \cdot \log t + m \cdot (k_R \cdot t) \cdot s \cdot n\big) \tag{17}$$

The third main loop (lines 24–32) calculates the final weights by looping over all features and classes. It also includes a s-sized loop for finding the final value of η. The total complexity of the third main loop is;

$$L_{\text{fin}} = O(s^2 \cdot n) \tag{18}$$

The total complexity of the RELIEF-MM algorithm can be obtained by adding the values in Eqs. (15), (17) and (18). Here, we consider that $t \geq s$ and $m \geq s$ should be true, since the algorithm implicitly makes an assumption that there should be at least one instance of each class, and also at least one instance should be selected from each class, in order to calculate the feature weights of each class. Hence, $m \cdot t \cdot n \geq s^2 \cdot n$. Consequently, the terms coming from the Eqs. (15) and (18) can be omitted for the calculation of the asymptotic upper bound. Then, the complexity of the RELIEF-MM algorithm equals

$$\begin{aligned}
&O(\text{RELIEF} - \text{MM}) \\
&= O\big(m \cdot t \cdot n + m \cdot (k_R \cdot t) \cdot s \cdot \log t + m \cdot (k_R \cdot t) \cdot s \cdot n\big)
\end{aligned} \tag{19}$$

If the complexity of RELIEF-MM is compared with the complexity of RELIEF-F given in Sect. 2.4, it can be seen that the only difference lies in the terms related with the nearest neighbor selection. RELIEF-MM includes $(k_R \cdot t)$, whereas RELIEF-F has k. Essentially, these two terms are asymptotically equal in terms of complexity since both reside in the same range. Furthermore, if we consider using RELIEF-MM with balanced datasets, the $(k_R \cdot t)$ term turns

into $(k_R \cdot \frac{t}{s})$, as described above. Thus, for small values of k_R, the complexity of RELIEF-MM for balanced datasets is $O(m \cdot t \cdot n)$, as it is for RELIEF-F. Here, the $m \cdot t \cdot n$ term is an asymptotic upper bound on $m \cdot (k_R \cdot t) \cdot \log t$ and $m \cdot (k_R \cdot t) \cdot n$, for small values of k_R. In conclusion, it can be said that the complexity of the RELIEF-MM algorithm is asymptotically the same as the original RELIEF-F algorithm.

3.5 Using RELIEF-MM with prediction scores

As mentioned before, in late fusion, the fusion is performed after a classification step. Thus, the inputs for the fusion process are the prediction scores obtained from the classifiers. In other words, the feature values of the samples may not available during the fusion process, in many cases. However, the original RELIEF algorithm uses the feature values of the samples to calculate the distances between them. Thus, in late fusion scenarios, where the feature values are not available, it is not possible to utilize the RELIEF algorithm. It is necessary to extend the weight calculation process of the RELIEF algorithm so that it can be used with the prediction score inputs.

Given that the classes are $\mathcal{C} = \langle c_u \rangle_{u=1}^{s}$, the modalities are $\mathcal{F} = \langle f_i \rangle_{i=1}^{n}$ and the training samples are $\mathcal{D} = \langle d_j \rangle_{j=1}^{t}$; the list of prediction probabilities for class c_u and modality f_i is $S_{c_u f_i} = \{s_j^{c_u f_i}\}_{j=1}^{t}$, where $0 \leq s_j^{c_u f_i} \leq 1$. Note that the order of the samples in $\mathcal{D}$ and the score values in S_{cu},f_i are given correspondingly.

While using RELIEF-MM with prediction score inputs, the algorithm remains the same, but the *diff* function calculation should be rewritten, since we do not have feature values anymore. Considering that an $s_j^{c_u f_i}$ value of a sample d_j corresponds to the similarity of the sample to a predefined class (c_u), we utilize the following idea: The difference between similarities of two samples to the same pattern corresponds to a reasonable distance metric of these samples. Thus the *diff* function in the RELIEF-MM algorithm can be updated as the differences of the score values of the samples. However, for each sample, there exist s number of scores of each modality, where each score is the similarity value for a different class. Thus, we consider that the RELIEF-MM algorithm iterates over the training samples, and we use the score list which corresponds to the class of the randomly selected sample, on each turn. Thus, the *diff* function becomes;

$$\text{diff}(f_i, d_x, d_y) = |s_x^{C(d_x),f_i} - s_y^{C(d_x),f_i}| \tag{20}$$

where d_x is the randomly selected sample, d_y is one of the hit/miss instances for d_x, and $C(d_x)$ function corresponds to the ground truth class value of the sample d_x given as parameter.

4 Empirical study

In this section, we evaluate the proposed modality weighting approach for semantic retrieval of multimedia data. For the retrieval task, the multimedia data is queried based on the semantic concepts. First, retrieval for each single modality is performed, then a multimodal retrieval is done. During the multimodal retireval, the modalities are combined with a linear (weighted averaging) combiner-based late-fusion approach, where the weights of the modalities are generated via different approaches.

To perform a detailed comparison, we carry out our empirical study in two major steps:

- Comparison with Other Approaches: We compare the retrieval accuracies of the RELIEF-MM-based linear weighted fusion approach with a RELIEF-F-based one, as well as the single modalities, basic approaches (simple averaging and maximum) and exhaustive search. Also, we compare the modality selection performance of RELIEF-MM with RELIEF-F in terms of the accuracies for each different number of feature selections.
- Tests for Each Extension Idea: After a comparison with alternative approaches, we focus on the issues that motivated us to develop RELIEF-MM, and perform tests comparing (i) class-common and class-specific selection, (ii) performances with multi-label, uni-label data and noisy cases (iv) using a dynamic vs. static nearest neighbor selection (k_R vs. k).

Considering that one of the important contributions of this study is the use of prediction scores with the RELIEF algorithm, the experiments are conducted with both of the following scenarios:

- We assume that the feature values are available, and use them to calculate the feature weights.
- We apply a pure late fusion scenario by assuming that the feature values are not available. Thus, the prediction scores are used for weight calculation.

4.1 Experimental setup

4.1.1 Datasets

Experiments are carried out on three frequently utilized benchmark datasets: TRECVID 2007 [35], TRECVID 2008 [36] and the Columbia Consumer Video (CCV) Database [19]. The dataset characteristics are summarized in Table 1. Further details and a performance comparison of TRECVID participants can be found in the corresponding references.

Table 1 Datasets

	TRECVID 2007	TRECVID 2008	CCV
Dataset length (hours)			
Train	~50	~100	~105
Test	~50	~100	~105
Number of videos			
Train	110	219	4659
Test	109	215	4658
Number of shots			
Train	21,532	39,674	N/A
Test	18,142	33,726	N/A

While using the TRECVID 2007 and 2008 dataset, we prefer using the outputs of common shot reference, for shot segmentation. For these datasets, the shots are used as the retrieval documents. Besides, for the CCV dataset, each video is accepted as a retrieval document. During the tests, the shots (for TRECVID 2007/2008) and the videos (for CCV) are considered as individual and independent documents, which means no contextual information or interaction is taken into account between shots/videos.

Each of the utilized datasets provides different sets of concept annotations. The annotations on all three datasets are provided in a multi-label manner, which means each shot can contain more than one label. A complete list of these concepts is given in Fig. 3 with sample images. The semantic queries performed during the tests are based on these semantic concepts.

4.1.2 Modalities

For all datasets, we consider a multimodal setting, and use features from different modalities. However, we prefer a relaxed definition for 'modality' [52]. The modalities of multimedia data are usually accepted as audio, visual and text modalities, but each of these modalities can be expanded. For instance, visual data can be defined with several modalities like color, shape, texture and face. Here, each of these modalities is a different type of information source, and contains a significant amount of complementary information. Thus, we accept each different type of information (i.e., each complementary feature) as a different modality. The multimodal features utilized during the test are listed in Table 2.

As presented on the table, visual, audial and textual features are extracted from the videos of the TRECVID 2007 dataset. For visual features, one key frame per shot is adopted and the middle frame for each shot is selected as the key frame. The feature extraction and distance calculation tasks of the visual features are performed using the MPEG-7 reference software (eXperimentation Model, XM)

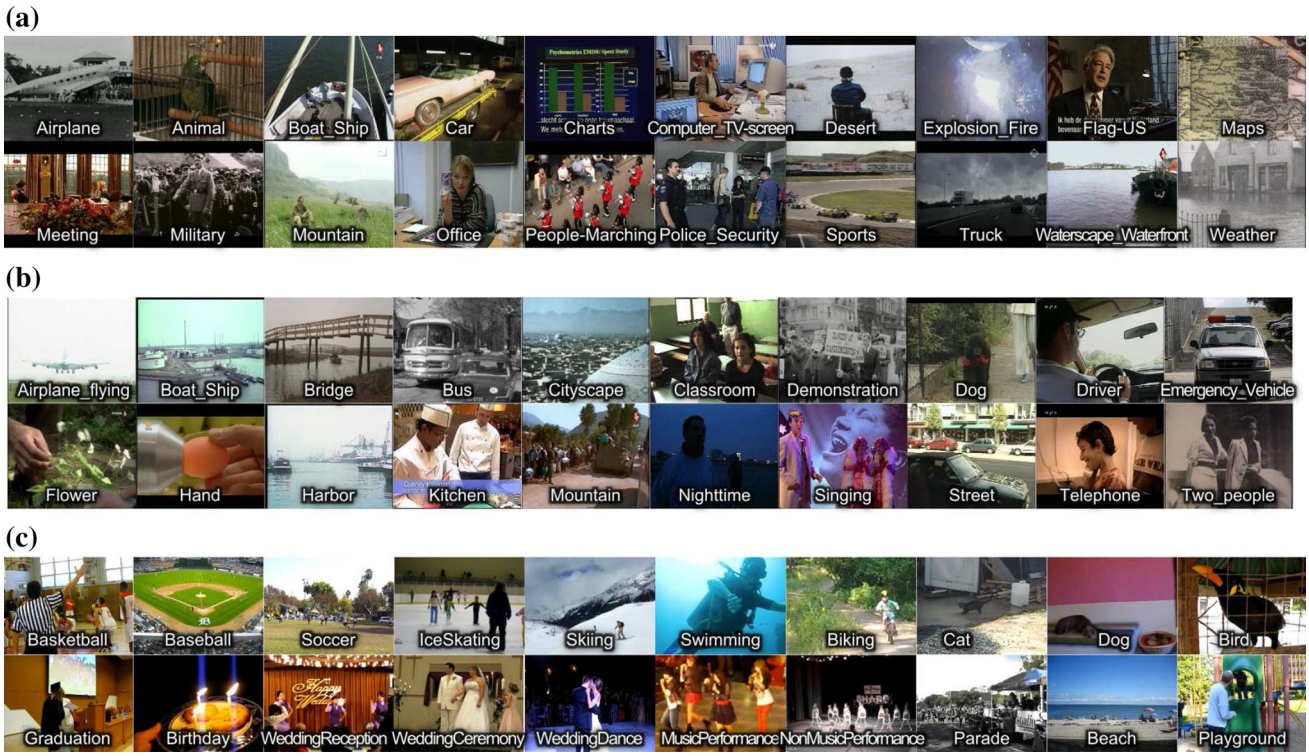

Fig. 3 Query concepts for each dataset and sample shot images from query concepts, **a** TRECVID 2007 dataset, **b** TRECVID 2008 dataset, **c** CCV dataset

Table 2 Modalities utilized for each dataset

Dataset	Modalities
TRECVID 2007	MPEG-7 Color Layout (CL)
	MPEG-7 Region Shape (RS)
	MPEG-7 Edge Histogram (EH)
	Zero Crossing Rate and Energy (ZCRE)
	Mel-freq. Cepstrum Coefficients (MFCC)
	Term Freq.–Inverse Doc. Freq. (TF-IDF)
TRECVID 2008	Gabor Texture (GT)
	Edge Direction Histogram (EDH)
	Scale Inv. Feature Transform (SIFT)
	Grid-based Color Moment (GCM)
	Grid-based Wavelet Texture (GWT)
CCV	Scale Inv. Feature Transform (SIFT)
	Spatial-Temporal Interest Points (STIP)
	Mel-freq. Cepstrum Coefficients (MFCC)

[33]. For audial features, the entire audio of each shot is processed and Yaafe toolbox [2] is utilized for feature extraction. For the textual features, the Automatic Speech Recognition and Machine Translation texts, which are provided by TRECVID, are employed. During the calculations, no stop-word filtering or preprocessing is done.

For TRECVID 2008, the features are not extracted; instead, the prediction score values of each shot for the concept queries are obtained from the CU-VIREO374 [18] dataset. In the CCV dataset some well-known features are already provided, as well as the videos and annotations. For more detailed explanations, interested readers can refer to [18, 19].

Considering that we combine the modalities with a late fusion process, features from each modality should be processed with a classifier and the prediction scores should be obtained before the combination (for TRECVID 2007 and CCV datasets). For the classification task, a support vector machine (SVM) classifier with appropriate radial basis function (RBF)-based kernels is preferred, and LIBSVM [3] is utilized.

4.1.3 Metrics

To measure the retrieval accuracy, *Precision*, *Recall*, *average precision* (AP) and *mean average precision* (MAP) are used. *Precision* is the fraction of retrieved documents that are relevant to the query concept, while *Recall* is the fraction of relevant documents that are retrieved. The *AP* is the sum of the precision at each relevant hit in the retrieved list, divided by the minimum of the number of relevant documents in the collection and the length of the list. Regarding the evaluation rules of TRECVID, AP is measured at 2000. *MAP* is the *AP* averaged over several query concepts. In other words, the *AP* of each concept is calculated separately and then the *MAP* is found by averaging them. Beyond the measurements of accuracy, we also present the statistical significance of the obtained results. To do so, we perform a *Student's t test* with paired samples, where the pairs are the accuracy results for different concept queries. A paired t test gives a p value which denotes the significance of the improvement between two tests. The smaller the p value, the more significant the difference of the two average values. We assume a confidence level at 0.95 and accept the results with p value <0.05 as significant.

We define another metric, named fusion gain (FG), to perceive the effect of the fusion process. Fusion gain gives the relative performance increase between two different configurations:

$$\mathrm{FG}(x, y) = \frac{\mathrm{MAP}(x) - \mathrm{MAP}(y)}{\mathrm{MAP}(y)} \tag{21}$$

where x and y denote different configurations (i.e., different feature selections). In our experiments we calculate two FGs:

- FG_{BS} The fusion gain is calculated by comparison with the best single modality.
- FG_{AVG} The fusion gain is calculated by comparison with the simple averaging approach.

4.2 Comparison with other approaches

To see the effectiveness of RELIEF-MM, we first compare its retrieval accuracy with the following alternative methods,

- Each single modality,
- Basic approaches like maximum (MAX) and averaging (AVG),
- Class-common exhaustive search (Exh-CC), Class-specific exhaustive search (Exh-CS)
- Original RELIEF-F algorithm.

Using each single modality and basic approaches represents the lower accuracy bounds for the fusion system. A fusion system is accepted as successful if it provides better accuracy than any of the single modalities. We also consider the MAX and AVG approaches as lower bounds, since these are the most frequently utilized fusion approaches due to their simplicity in calculation. In the MAX approach, the decision in the fusion process is calculated by taking the maximum score value of the available modalities. In the AVG approach, the mean of the score values of all available modalities is accepted as the final decision. On the other hand, we also present the accuracies of the exhaustive search for finding optimal modality weights, which provides an upper bound for the retrieval accuracies. For the

Table 3 Comparison of retrieval accuracies

	TRECVID 2007		TRECVID 2008		CCV	
		MAP (%)		MAP (%)		MAP (%)
S	CL	8.711	EDH	10.479	SIFT	49.676
	EH	9.032	GT	10.802	STIP	39.959
	RS	6.762	SIFT	19.032	MFCC	27.585
	ZCRE	6.385	GCM	13.027		
	MFCC	6.884	GWT	9.094		
	TFIDF	6.286				
B	MAX	6.639	MAX	17.126	MAX	52.071
	AVG	8.270	AVG	18.969	AVG	57.340
E	Exh-CC	10.322	Exh-CC	20.034	Exh-CC	57.403
	Exh-CS	12.988	Exh-CS	22.183	Exh-CS	57.783
F	RELIEF-F	9.847			RELIEF-F	56.027
	RELIEF-MM	10.563			RELIEF-MM	57.511
P	RELIEF-F	9.076	RELIEF-F	19.760	RELIEF-F	55.380
	RELIEF-MM	9.454	RELIEF-MM	20.559	RELIEF-MM	57.562

The first column denotes the configuration: *S* single modalities, *B* basic approaches, *E* exhaustive search, *F* RELIEF methods using feature values, *P* RELIEF methods using prediction scores

exhaustive search approach, we perform both class-common and class-specific weighting processes. Exh-CC evaluates every different weight set to find the optimal weight of each modality. In Exh-CS, the same process is repeated for each class, separately. Lastly, we compare our proposed approach with the original RELIEF-F algorithm, which exhibits the major contribution of this study. During these comparisons, for RELIEF-F and RELIEF-MM, the performances at the optimal k_R values are presented. The v value for RELIEF-MM is used as 2.

The use of an exhaustive search usually causes infeasible test situations. In our tests, the feasibility of the weight selection process via an exhaustive search depends on the precision of the weights, as well as the number of modalities. For instance, if we want to have a precision of 0.01 between weights with 6 modalities, we should check 100^6 cases. Assuming that we already have the prediction scores of each modality beforehand, such a process for TRECVID 2007 dataset would take so long that even parallelization of the process would not be a solution. Thus, we follow a computationally simpler search process without damaging the fairness of the comparisons. We perform the following two different near-exhaustive search processes[3], and then select the best one: (1) We first perform an exhaustive binary selection among available modalities, and select the best 4 modalities. Then, we perform a weight search on the selected 4 modalities with 0.01 precision ($w \in \{0, 0.01, 0.02, \ldots, 0.99, 1\}$). After finding the optimal weights and fixing them, we perform a weight search on the remaining 2 modalities. (2) We first perform an exhaustive weight search on all available modalities with 0.1 precision and find the optimal weights for each feature. Then, as a second step, we tune up the weights by performing a selection between $[w - 0.05, w + 0.05]$ with 0.01 precision.

To evaluate the proposed approach, one may argue that there should be comparisons with other available feature selection/weighting methods. However, as described in Sects. 1 and 2.2, currently available filter-based feature selection/weighting methods in the literature are not easily applicable to the modality weighting problem, due to the issues of the intrinsic multi-dimensionality of modalities and the multivariate inputs of fusion systems. Thus, adapting other approaches to modality weighting problem is beyond the focus of this study. Besides, we do not consider comparing our method with several different wrapper approaches since we perform a comparison with the exhaustive search, which gives the best possible accuracy. It is also known that any wrapper approach is much more computationally complex than our approach. Consequently, we think that the comparisons included in this study are enough to evaluate the effectiveness and efficiency of our proposed approach.

In Table 3, the MAP values of the above listed approaches are presented for the TRECVID 2007, TRECVID 2008 and CCV datasets. For a better understanding of which weighting approach provides more effective fusion, the Fusion Gains of these approaches are calculated and presented in Table 4. In addition to the accuracy results included here, a statistical significance analysis of the results is presented in Table 5. In the tables, (F) denotes the use of feature values as inputs to the RELIEF-based

[3] This two-step process is applied for the TRECVID 2007 and 2008 datasets, where the number of modalities lead to inefficient situations. For the CCV dataset, an exhaustive weight search process is performed with 0.01 precision.

Table 4 Fusion gains wrt Best single modality and AVG approach

	TRECVID 2007		TRECVID 2008		CCV	
	FG_{BS} (%)	FG_{AVG} (%)	FG_{BS} (%)	FG_{AVG} (%)	FG_{BS} (%)	FG_{AVG} (%)
(B)						
MAX	−26.500	−19.726	−10.013	−9.718	4.822	−9.188
AVG	−8.439		−0.327		15.428	
(E)						
Exh-CC	14.283	24.816	5.266	5.612	15.556	0.110
Exh-CS	43.792	57.045	16.558	16.941	16.321	0.773
(F)						
RELIEF-F	9.016	19.064			12.786	−2.289
RELIEF-MM	16.944	27.723			15.773	0.298
(P)						
RELIEF-F	0.483	9.744	3.829	4.170	11.483	−3.418
RELIEF-MM	4.669	14.316	8.023	8.378	15.877	0.388

Table 5 Statistical significance analysis using paired *T* test

	p value (BEST)	*p* value (AVG)	*p* value (RELIEF-F)
TRECVID 2007			
F			
RELIEF-F	7.39E-02	4.67E-02*	
RELIEF-MM	1.96E-02*	2.90E-02*	1.95E-02*
P			
RELIEF-F	4.75E-01	2.45E-01	
RELIEF-MM	2.65E-01	1.51E-01	1.89E-02*
TRECVID 2008			
P			
RELIEF-F	4.04E-01	5.60E-02	
RELIEF-MM	1.87E-02*	2.30E-03*	4.02E-04*
CCV			
F			
RELIEF-F	6.74E-06*	2.38E-03*	
RELIEF-MM	2.02E-06*	1.24E-01	4.43E-05*
P			
RELIEF-F	2.08E-04*	4.72E-13*	
RELIEF-MM	1.92E-06*	4.29E-02*	6.99E-10*

Pairs are based on query concepts. Statistically significant results according to the confidence level 0.95 (p value <0.05) are given with an * p value (BEST), (AVG) and (RELIEF-F) denote the p values with respect to the best single modality, simple averaging and RELIEF-F approaches, respectively

algorithms, whereas (P) represents the cases where the predictions scores are used as the inputs.

From these experimental results, we arrive at the following observations:

- Combinations of different modalities give more accurate results than the single modalities. However, selection of modalities is a critical issue. A wrong selection can lead to worse results than the best of the single modalities. For instance, AVG cannot provide a positive gain in the TRECVID 2007 and 2008 datasets, compared to the best single modality. Similarly, a MAX approach is not successful in any of the three datasets. This is because of the fact that these simple methods do not perform an effective evaluation on the modalities, and thus they cannot discard the unfavorable modalities. Although these approaches provide the most efficient solutions, they cannot always provide an effective solution and they are not robust against different datasets. Consequently, a more robust and effective approach is highly recommended despite the risk of some decrease in efficiency.
- RELIEF-F is significantly better than the best single modality in one of two datasets where feature values are used as input, and one of three datasets where predictions scores are used. If compared with the AVG approach, RELIEF-F has a significant improvement in only one case out of all five. Hence, RELIEF-F is not a robust solution against different datasets. Still, it can be accepted as an applicable modality selection approach, since it does not provide retrieval accuracies worse than the best single modality, and usually performs slightly better.
- RELIEF-MM provides a significant improvement over the best single modality for all datasets when the feature values are used as input, and two of three datasets when the prediction scores are used. If compared with AVG results, RELIEF-MM is significantly better in one of the two datasets with the feature value inputs, and in two out of three datasets with the prediction scores. When RELIEF-MM is compared with RELIEF-F, it is observed that RELIEF-MM obtains higher retrieval accuracies than RELIEF-F in all cases, each having a p value <0.05. In addition, it

Table 6 Approximate Execution Times of Exhaustive and RELIEF-based methods, on three different datasets

	RELIEF-F (s)	RELIEF-MM (s)	Exh-CC (h)	Exh-CS (h)	Exh-CC*	Exh-CS*
TRECVID 2007	3	6	17	340	19 years	380 years
TRECVID 2008	10	11	22	440	100 days	5.5 years
CCV	2	2	0.14	2.7	0.14 hours	2.7 hours

The column with an asterisk denotes estimated values for a real exhaustive search scenario

should be noted that RELIEF-MM achieves higher accuracy results than the best single modality, AVG and MAX approaches, and even slightly better results than the Exh-CC approach. Thus, there is strong evidence that the RELIEF-MM approach introduces a significant improvement and can be accepted as a robust and effective solution as a modality weighting approach for multimedia data. Therefore, RELIEF-MM can be regarded as a practical enhancement for the multimedia retrieval studies using simple averaging for fusion.

- An exhaustive search finds the optimal feature selection since it evaluates all possible combinations. The accuracy results show that the use of a class-specific approach in the exhaustive search (Exh-CS) helps to improve retrieval accuracy in all three datasets. Besides, being a class-specific approach, RELIEF-MM is not upper-bounded with Exh-CC, whereas the accuracies of RELIEF-F are always less than Exh-CC.
- The performance of using prediction scores instead of feature values for calculating the modality weights depends on the characteristics of the dataset. In our experiments, results with the TRECVID 2008 and CCV datasets are reasonably good. However, for TRECVID 2007 dataset, there exists a considerable decrease in accuracy according to the results of using feature values. Thus, it may be hard to give a conclusive decision about the effectiveness of using prediction scores, with the current evidence. Nevertheless, the accuracies with the prediction scores outperform best single modality, MAX and AVG approaches. Consequently, we observe that the results of using prediction scores is promising and they are applicable when the feature values are not available during the fusion process.

The efficiency of the proposed approach is another important concern. A running time comparison[4] of RELIEF-F, RELIEF-MM, Exh-CC and Exh-CS is presented in Table 6. The values given in the table correspond to the cases presented in Table 3. The table includes both the near-exhaustive search running times and the estimated

[4] The measurements are taken on a machine with "Intel(R) Xeon(R) CPU E5530 @2.40GHz". The values on the graph and table are obtained without a parallel programming approach.

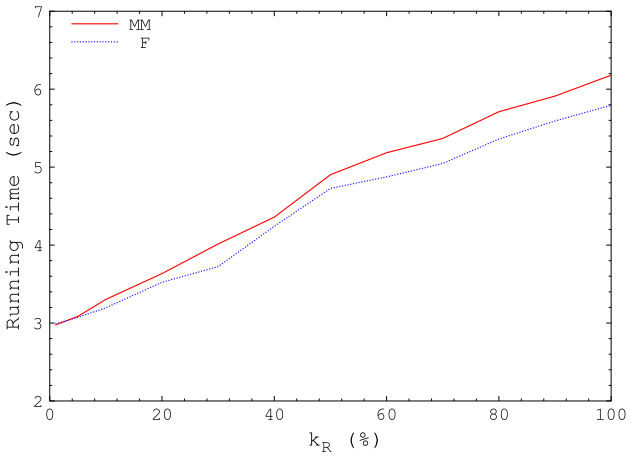

Fig. 4 Running time comparison of RELIEF-F and RELIEF-MM on TRECVID 2007 dataset for different k_R values

real exhaustive search times. The basic approaches (AVG and MAX) are not given in the table since they are done at no cost. Furthermore, a detailed comparison of RELIEF-MM and RELIEF-F, for different k_R nearest neighbor selections, is presented in Fig. 4. According to the given experimental results, execution of RELIEF-MM results in a small increase in time, which is in parallel with the complexity analysis given in Sect. 3.4.1. Besides, the exhaustive search methods, even the near-exhaustive search, require a high time cost, as expected. Hence, RELIEF-MM can be accepted as an efficient modality weighting approach, considering that the time cost is a polynomial function of the number of modalities, and thus much more efficient than the exhaustive search. When compared with basic approaches, the cost of RELIEF-MM is still acceptable, considering the improvement in the retrieval accuracy.

Up until now, the average query performances have been compared. To make a more detailed comparison, we also perform a concept-based analysis. Figure 5 illustrates a concept-based comparison and presents the number of concepts for which RELIEF-MM provides higher accuracy when compared with a particular approach. In addition, precision-recall graphs for some of the query concepts are given in Fig. 6. According to the given experimental results, RELIEF-MM achieves higher accuracies in a larger number of concepts than RELIEF-F, the best single modality and AVG approaches, regardless of the used

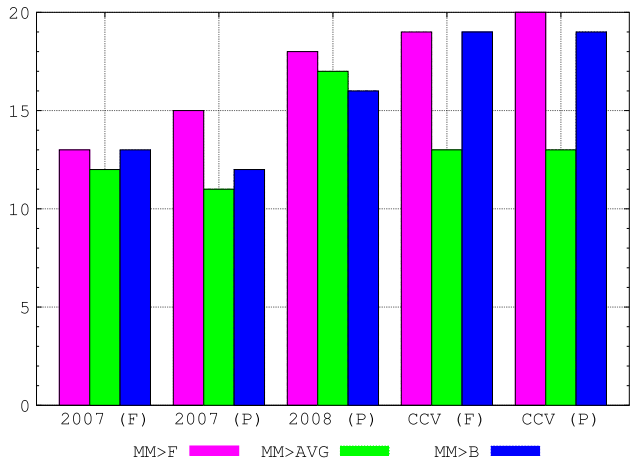

Fig. 5 Concept-based accuracy comparison of RELIEF-MM with other approaches. *Columns* indicate the number concepts that RELIEF-MM provides higher accuracy than the compared approach. *Each group* of *columns* denotes a different dataset with a specific input type. MM>F: RELIEF-MM vs. RELIEF-F, MM>AVG: RELIEF-MM versus Simple averaging, MM>B: RELIEF-MM versus best single modality

dataset and the input type (feature values vs. prediction scores). Nonetheless, the success rate of RELIEF-MM compared to RELIEF-F is more pronounced than the best single modality and AVG approaches. Under this observation, we can infer that the improvement provided by RELIEF-MM is reasonably good, due to the extensions introduced in this study. However the RELIEF idea in general may lead to difficulties in some particular data distributions and is open to improvement. Even though the RELIEF algorithm utilizes a margin-based nonlinear classifier [46] to evaluate the features and a margin-based nonlinear classifier is known to be successful in general, the way RELIEF uses the input data is based on a standard procedure of employing the distances from each training sample to its neighbors, and does not benefit from any feature transformations in kernel space. This approach may be inadequate for some particular concepts that have unique data distributions. Just as employing various kernel types in SVM classifiers according to the characteristics of data and features leads to more effective classification results, so performing some appropriate kernel transformations on the RELIEF input data will help to make the RELIEF approach superior in a larger number of query concepts. However, such a problem is not included within the scope of this study, and has been left for future work.

Beyond the discussion on kernel transformation, one may focus on the comparison between RELIEF-MM and RELIEF-F, and expect that a class-specific approach, i.e.,

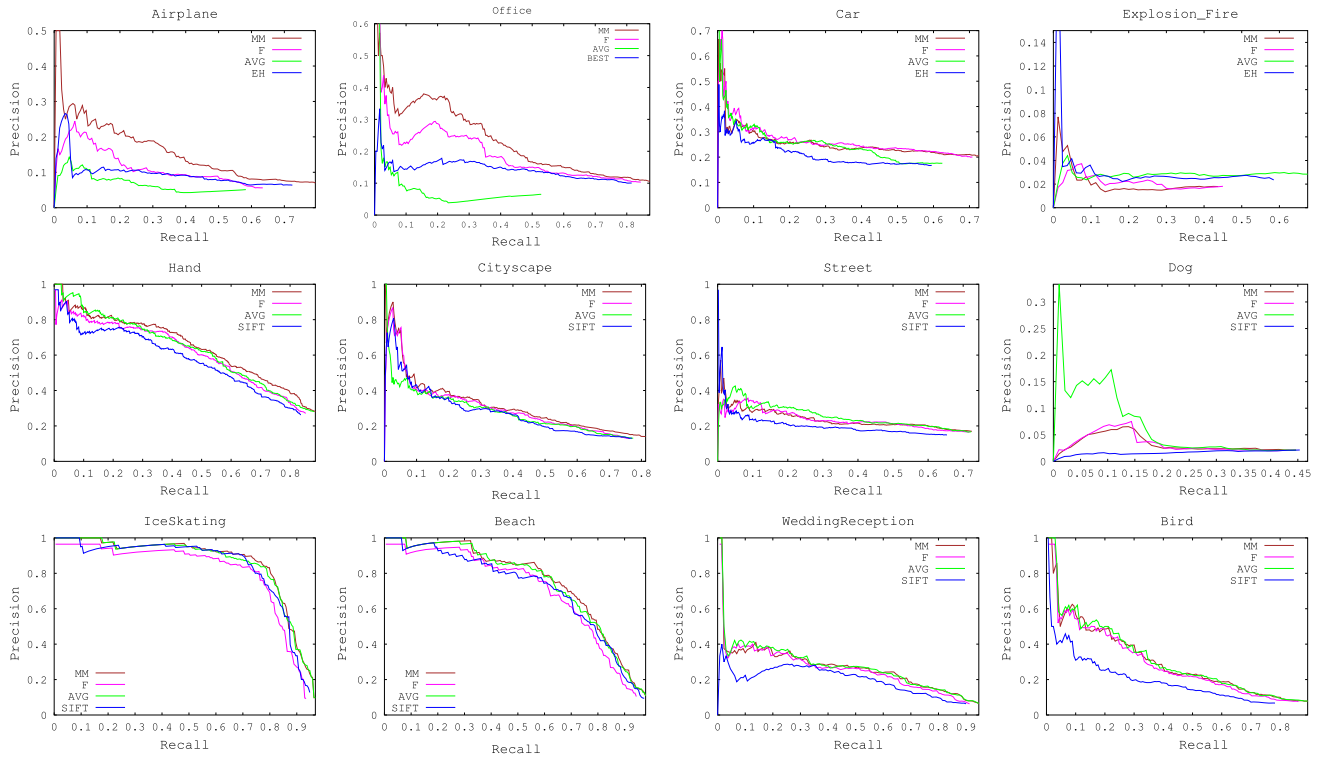

Fig. 6 Precision-–recall graphs of some selected concepts. The *rows* contain concepts from TRECVID 2007, TRECVID 2008 and CCV datasets, respectively. Concepts in the *first* and *second column* are best-case examples for RELIEF-MM, whereas the *third* and *fourth columns* show worst-case examples (in terms of accuracy)

RELIEF-MM, should have an ability to optimize the weights for every query concept individually and thus achieve higher retrieval accuracies in any concept. Insofar as our observations have shown, we think that there exist two important factors that prevent RELIEF-MM from giving the best accuracies in some of the concepts.

The first reason for RELIEF-MM's less-than-optimal accuracy with some concepts is the small number of training samples for some particular concepts, which lead to incomplete representation of the concept. As explained in Sect. 3.1, RELIEF-MM takes the samples of each concept into account for the weight calculation, whereas RELIEF-F uses all training samples without considering the concept that they belong to. As a result, the weight calculation of the concepts with a small number of training samples may lead to ineffective results. On the other hand, RELIEF-F gains a general insight into the effectiveness of each modality, which usually provides better results than the estimations of RELIEF-MM which are based on inadequate data. *Explosion_Fire*, *Desert*, *Flag* and *Truck* in the TRECVID 2007 dataset are some of the concepts for which RELIEF-F gives better accuracies. These concepts include 46, 67, 12 and 126 samples, respectively, whereas the dataset contains more than 350 samples per concept on average. A performance visualization for these kinds of concepts is given in the third column of Fig. 6. It is also worth noting that TRECVID 2008 and CCV include less concepts with a small number of samples, and thus the performance of RELIEF-MM is better in these two datasets than TRECVID 2007, as seen in Fig. 5.

Second reason for RELIEF-MM's weakness for some concepts is the intra-concept sample variety, which can be accepted as a side effect of including γ_f^c and η_f^c into the weight estimation function. As mentioned above, the way in which the margin-based classifier is utilized in RELIEF may be inadequate for some particular concepts that have unique data distributions. In RELIEF-MM we extend the weight estimation function and include γ_f^c and η_f^c into the formula. This preference increases the effect of margin-based calculations in the function since both γ_f^c and η_f^c are calculated using the intra-concept and inter-concept distances. Even though such preference makes the weight estimations better in most of the cases, increasing the effect of margin-based calculations without a feature transformation in the kernel space may lead to worse weight estimations. As a solution, two alternatives can be considered, a kernel transformation or including a non-margin-based variable into the weight estimation function, which can be considered for future work.

As a last comparison between RELIEF-MM and RELIEF-F, we try a different scenario from the previous tests, and combine the modalities with a simple averaging approach after a hard selection of the modalities instead of weighting. In modality selection for fusion, the ultimate goal is to find which subset of the modalities is more effective for the retrieval task. It is therefore important to rank the modalities correctly, and this scenario helps us to do so. In Fig. 7, the retrieval accuracies of RELIEF-F and RELIEF-MM are presented for those cases where a different number of modalities are selected and combined. During the test, first the weights of the modalities are obtained via the RELIEF-F and RELIEF-MM algorithms. Then a particular number of modalities are selected according the assigned weights. The results show that RELIEF-MM is clearly superior to the original RELIEF-F algorithm in this task. Hence, it can be said that the ranking capability of RELIEF-MM is more effective than that of RELIEF-F.

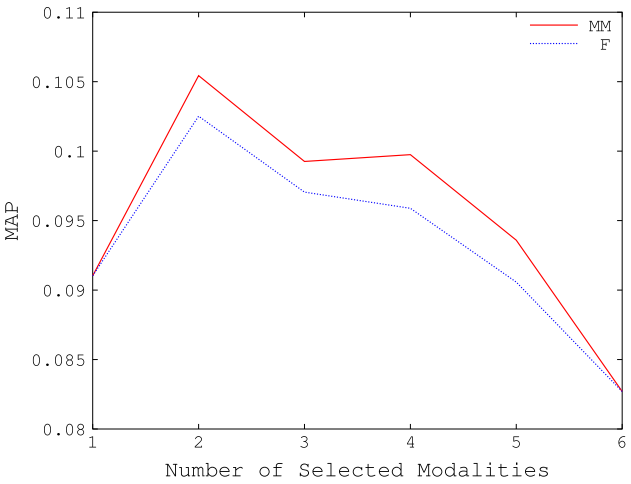

Fig. 7 Modality selection performances

4.3 Tests for each extension idea

To further analyze the improvements that RELIEF-MM provides, we compare our proposed algorithm with the baseline RELIEF-F algorithm with respect to each idea presented in Sect. 3. Below, each idea is discussed in a separate subsection. Through this evaluation, the TRECVID 2007 dataset is utilized.

4.3.1 Class-common versus class-specific feature weighting

The first improvement issue in RELIEF-MM is the conversion of the original RELIEF-F algorithm, which is a class-common approach, into a class-specific one. Thus, we compare the retrieval accuracies of the class-specific adaptation of RELIEF-F algorithm, which is introduced in Algorithm 2, with the original RELIEF-F. Moreover, we include the retrieval performances of RELIEF-MM algorithm to provide a more complete representation. In Fig. 8, precision–recall curves of these three methods are compared for optimized k selections. In addition, Fig. 9

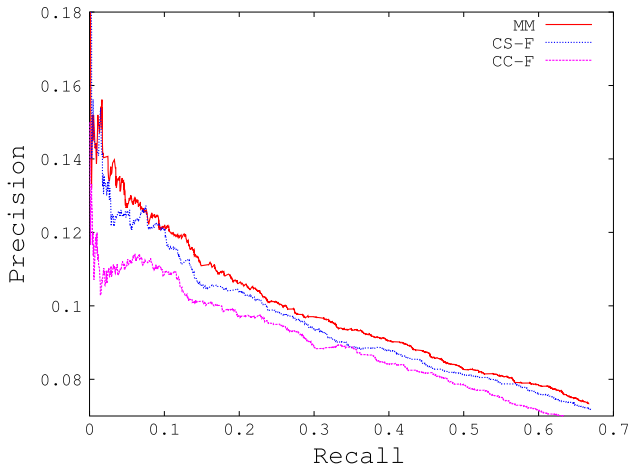

Fig. 8 Precision–recall curves of the original RELIEF-F (CC-F), class-specific RELIEF-F (CS-F) and RELIEF-MM (MM) algorithms

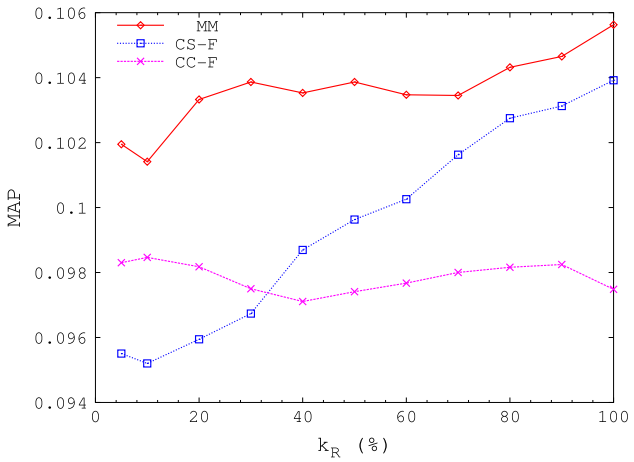

Fig. 9 Retrieval performances of original RELIEF-F (CC-F), class-specific RELIEF-F (CS-F) and RELIEF-MM (MM) algorithms

presents the retrieval performances of the given approaches with respect to different values of k nearest neighbors.

Figures 8 and 9 show that RELIEF-MM provides higher retrieval accuracies than both of the original RELIEF-F algorithm and the class-specific adaptation of RELIEF-F, for different values of nearest neighbors. Furthermore, Fig. 8 presents the clear superiority of the class-specific approach over the original one, and Fig. 9 shows that the accuracy of the original approach decreases, as the number of neighbors is increased. However, in the class-specific RELIEF-F, the accuracy is almost directly proportional to the number of neighbors. In addition, until some point around 33 % of nearest neighbors selection, the original RELIEF-F performs better than the class-specific RELIEF-F, which means that the original algorithm is more powerful than the class-specific approach for a small number of neighbors. The reason for this situation is discussed below.

In discrimination-based approaches, one of the most important factors that affects the success of the approach is the variety of the encountered samples. The original RELIEF-F algorithm estimates the weights by processing the randomly selected m samples and k neighbors of each sample from $s - 1$ classes. Equivalently, class-specific RELIEF-F allocates the randomly selected m samples into s classes according to the prior probabilities of each class, and processes the samples of each class separately. Hence, the weights of each class are estimated using a smaller number of samples according to m. If the number of nearest neighbors k is also small, the information obtained from the distances between samples becomes limited, which directly affects the success of class-specific RELIEF-F. If the number of nearest neighbors is increased, it is certain that the algorithm encounters some neighboring samples which have not been seen before, so that the algorithm obtains some adequate number of sample distances to estimate more effective weights. On the contrary, the original RELIEF-F usually does not encounter new samples when the number of nearest neighbors is increased, since many of the samples are seen through the m sample selection. If m is chosen as all training samples, there is no new instance that can provide new information while k is increased. Therefore, the only factor affecting the success of the original RELIEF-F algorithm becomes the noisy information obtained due to the increase in k. Consequently, it is more beneficial in this test to see which of the approaches can achieve higher accuracy in any configuration, since those upper bounds present how effectively they can use the available information. In Fig. 8, it is apparent that class-specific RELIEF-F uses the available information more effectively.

4.3.2 Performances with uni-label, multi-label and noisy data

Another improvement of RELIEF-MM is its ability to handle multi-label data. Thus, we compare the retrieval accuracies of the fusion systems using RELIEF-F and RELIEF-MM for weight generation in uni-label and multi-label data. This comparison helps us to understand whether RELIEF-MM is more effective in multi-label data.

To obtain a uni-label data, we first process the training dataset and remove the multi-labeled instances from the dataset. We use the newly constructed uni-label dataset only for the weight generation step of the fusion process. The classifiers, which give the inputs to the fusion process, are always trained with the multi-label dataset. Thus, we manage to compare only the effect of different weight generation methods. Furthermore, it should be noted that constructing the uni-label dataset by removing the multi-labeled instances may cause the loss of some information

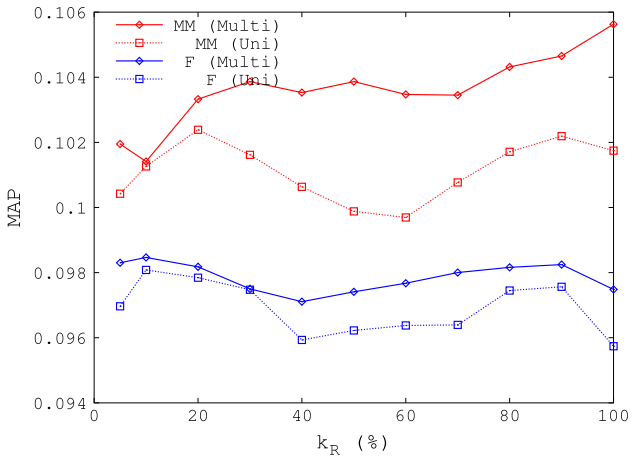

Fig. 10 Retrieval accuracies of RELIEF-F and RELIEF-MM for different k_R values, with uni-label and multi-label training data for weight generation. MM and F denote RELIEF-MM and RELIEF-F. (Uni) and (multi) denote the use of uni-label and Multi-label datasets for weight generation

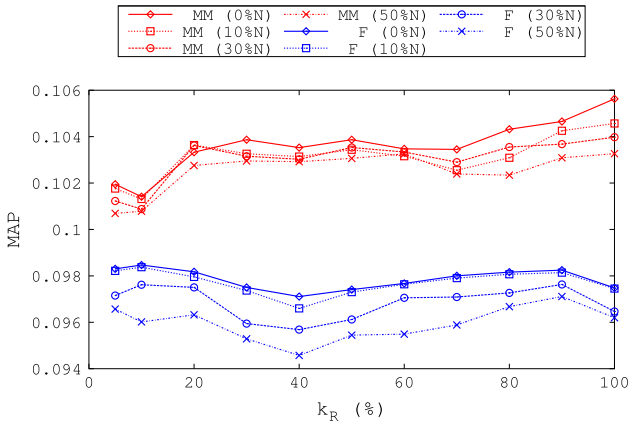

Fig. 11 Retrieval performances with different levels of noisy training data for weight generation

(e.g., approximately 40 % of the training instances is removed) and affect the performance of weight generation. Still, using a completely different uni-label dataset prevents us from comparing the accuracies of a weighting approach across datasets. Consequently, we find this setting fair enough to compare the effectiveness of RELIEF-MM and RELIEF-F.

The tests are conducted for several k_R nearest neighbor selections. Figure 10 presents the retrieval accuracies of RELIEF-MM and RELIEF-F using uni-label and multi-label data for weight generation. To understand the effect of using multi-label data, the differences between the accuracies of RELIEF-F and RELIEF-MM can be compared for uni-label and multi-label datasets. Such differences can be best understood by the area between the curves of RELIEF-F and RELIEF-MM in the given graph. As seen on the graph, the area between RELIEF-F and RELIEF-MM curves is larger for multi-label data, which can be evaluated as RELIEF-MM working better in multi-label data.

In addition to the uni-label vs. multi-label data comparison, we also consider the performances of the algorithms for noisy data. In Sect. 3 it is proposed that RELIEF-MM should perform better than RELIEF-F even in noisy data cases. Thus, we compare the performances of RELIEF and RELIEF-MM with noisy datasets. For this purpose, we manually add mislabeled instances into the multi-label dataset, and construct 10, 30 and 50 % noisy datasets. Similar to the tests for uni-label data, these noisy datasets are used only for the weight generation step. The retrieval accuracies at given noise levels are presented in Fig. 11, as well as the zero noise level.

Figure 11 demonstrates that the decrease in accuracy is usually larger for RELIEF-F, as the noise increases. Furthermore it is observed that RELIEF-MM is superior to RELIEF-F at any noise level. It can be stated that RELIEF-MM is more robust against noise.

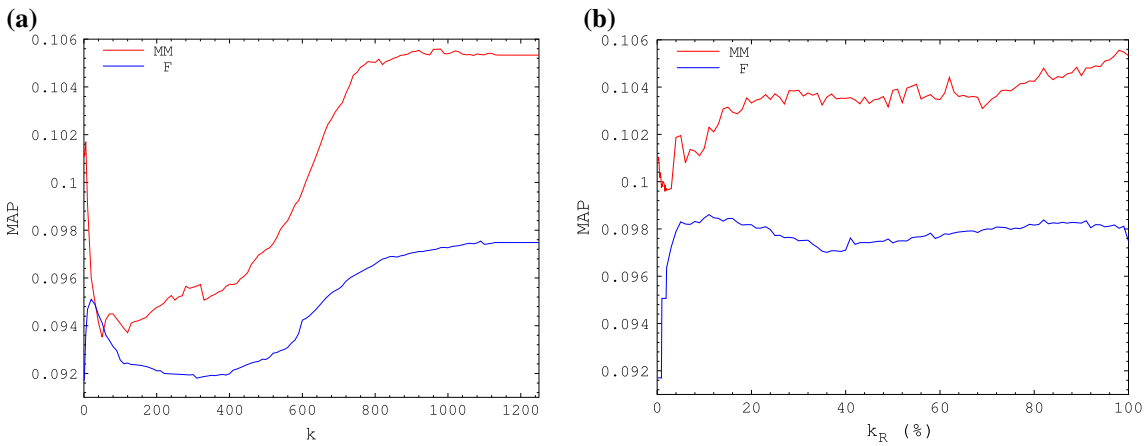

Fig. 12 Retrieval performances according to k versus k_R nearest neighbors, **a** k neareast neigbors, **b** k_R neareast neigbors

4.3.3 Using k versus k_R

One more improvement on the original RELIEF-F is the dynamic selection of k nearest neighbor as a ratio value of the class sample counts. The changes in retrieval accuracy change according to different k nearest neighbors are shown in Fig 12a. The change according to different k_R nearest neighbors is shown in Fig. 12b.

For each of the methods, it is expected that accuracy values will converge into the same value when k reaches the number of all training instances and k_R reaches 100 %. The improvement that k_R provides is more apparent in lower numbers of training samples. Figure 12 shows that both approaches exhibit a decrease in performance when 100–500 nearest neighbors are used. The main reason for the decrease is the use of imbalanced hit and miss instances for concepts that have a smaller number of samples, e.g., using $k = 400$ for a concept with only 200 samples causes the algorithm to use 200 hit instances, but 400 miss instances. Considering that RELIEF-MM works with class-specific preference, the decrease in accuracy becomes more dramatic for RELIEF-MM. On the other hand, the use of a dynamic selection with ratios (k_R) prevents such a decrease for both methods and enables more robust accuracy results against a different number of nearest neighbor selections. k_R is bounded by the number of samples in the class, thus the decrease caused by imbalanced hits/misses does not occur any more.

5 Conclusion

In this paper, the problem of modality weighting for multimodal information fusion is studied. As an effective and efficient modality weighting solution, a RELIEF-based approach is proposed. Considering the problems with RELIEF-F when using it with multimedia data for multimodal fusion, we focus on five crucial issues and extend the original RELIEF-F algorithm in these aspects. We first convert the original algorithm into a class-specific representation. Then we extend the algorithm and weight estimation function so that they estimate the modality weights better with multi-label and noisy data. For better estimations, we include the representation and reliability characteristics of modalities into the weight estimation function, in addition to the currently available discrimination capability. We also make an extension to make the algorithm more effective with unbalanced datasets. Lastly, we introduce a conversion procedure that enables the use of classifier predictions in RELIEF, considering that feature values may not be available during a fusion process.

Our approach is extensively tested on TRECVID 2007, TRECVID 2008 and CCV datasets with several modalities in a multimodal information fusion scenario. The results show that using RELIEF-MM guarantees higher accuracies than any single modality, and shows much better performance than simple averaging and RELIEF-F-based methods. In addition, RELIEF-MM provides slightly better performance than the class-common exhaustive search-based approach, although it is computationally much more efficient. We also perform several comparative tests against the RELIEF-F approach, aiming to examine each extension idea, and confirm that the proposed extensions lead to improvements on RELIEF-F. Consequently, we argue that our proposed approach is a timely efficient, accurate and robust way of modality selection.

The experiments carried out also exhibit some situations for future work. To further improve RELIEF-MM, we put forward the following ideas for future study:

- The RELIEF-MM algorithm utilizes a margin-based discrimination approach, like the original RELIEF, while evaluating features. Performing some appropriate feature transformations on the kernel space may improve the quality of the weight estimations, especially for those particular concepts that have unique data distributions. Performing such transformations separately for each concept type, like a one-vs.-all approach, may yield better results.
- Another improvement idea for RELIEF-MM is to include a non-margin-based evaluation metric into the weight estimation function (e.g., mutual information, information gain, correlation, etc). Any considerable metric may have its own complications when being used with modalities instead of features, deficiencies for multimedia data and extra computational complexity, however. All these factors should be analyzed in detail.
- It is possible to further increase the efficiency of the RELIEF-MM algorithm by employing some caching mechanisms (e.g., k-d trees, hashing).

References

1. Atrey, P.K., Kankanhalli, M.S., Oommen, J.B.: Goal-oriented optimal subset selection of correlated multimedia streams. ACM Trans. Multimedia Comput. Commun. Appl. **3**(1) (2007). doi:10.1145/1198302.1198304
2. Mathieu, B., Essid, S., Fillon, T., Prado, J., Richard, G.: Yaafe, an easy to use and efficient audio feature extraction software (2010). In: Proceedings of the 11th ISMIR Conference, Utrecht, Netherlands
3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011). Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor. Newsl. **6**(1), 1–6 (2004). doi:10.1145/1007730.1007733

5. Chen, Y.Y., Hsu, W., Liao, H.Y.: Automatic training image acquisition and effective feature selection from community-contributed photos for facial attribute detection. Multimedia, IEEE Transactions on **15**(6), 1388–1399 (2013). doi:10.1109/TMM.2013.2250492
6. Dietterich, T.G.: Machine-learning research: Four current directions. The AI Magazine **18**(4), 97–136 (1998)
7. Doquire, G., Verleysen, M.: Feature selection for multi-label classification problems. In: Proceedings of the 11th International Conference on Artificial Neural Networks Conference on Advances in Computational Intelligence-vol. Part I, IWANN'11, pp. 9–16. Springer, Berlin, Heidelberg (2011). http://dl.acm.org/citation.cfm?id=2023252.2023255
8. Ferri, F.J., Pudil, P., Hatef, M., Kittler, J.: Comparative study of techniques for large-scale feature selection. In: Gelsema, E.S., Kamal, L.N. (eds.) Pattern Recognition in Practice IV, Multiple Paradigms, Comparative Studies and Hybrid Systems, pp. 403–413. Elsevier, Amsterdam (1994)
9. Fumera, G., Roli, F.: A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE TPAMI **27**(6), 942–956 (2005). doi:10.1109/TPAMI.2005.109
10. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003). http://dl.acm.org/citation.cfm?id=944919.944968
11. Hall, M.A.: Correlation-based Feature Subset Selection for Machine Learning. Ph.D. thesis, Department of Computer Science, University of Waikato, New Zealand (1999)
12. Huang, K.C., Lin, H.Y.S., Chan, J.C., Kuo, Y.H.: Learning collaborative decision-making parameters for multimodal emotion recognition. In: Multimedia and Expo (ICME), 2013 IEEE International Conference, pp. 1–6 (2013). doi:10.1109/ICME.2013.6607472
13. Hunt, E.B., Stone, P.J., Marin, J.: Experiments in induction/Earl B. Hunt, Janet Marin, Philip J. Stone. Academic Press, New York (1966)
14. Inoue, N., Kamishima, Y., Wada, T., Shinoda, K., Sato, S.: Tokyotech+canon at trecvid 2011. In: NIST TRECVID Workshop. Gaithersburg, MD (2011)
15. Jain, A., Nandakumar, K., Ross, A.: Score normalization in multimodal biometric systems. Pattern Recognition **38**(12), 2270–2285 (2005)
16. Jain, A.K., Duin, R.P., Mao, J.: Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**, 4–37 (2000)
17. Jiang, Y.G., Bhattacharya, S., Chang, S.F., Shah, M.: High-level event recognition in unconstrained videos. Int. J. Multimedia Info. Retr. 1–29 (2012). doi:10.1007/s13735-012-0024-2
18. Jiang, Y.G., Yanagawa, A., Chang, S.F., Ngo, C.W.: CU-VIREO374: Fusing Columbia374 and VIREO374 for Large Scale Semantic Concept Detection. Tech. rep., Columbia University ADVENT #223-2008-1 (2008)
19. Jiang, Y.G., Ye, G., Chang, S.F., Ellis, D., Loui, A.C.: Consumer video understanding: a benchmark database and an evaluation of human and machine performance. In: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ICMR '11, pp. 29:1–29:8. ACM, New York, NY, USA (2011). doi:10.1145/1991996.1992025
20. Jiang, Y.G., Zeng, X., Ye, G., Ellis, D., Chang, S.F., Bhattacharya, S., Shah, M.: Columbia-ucf trecvid2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching. In: P. Over, G. Awad, J.G. Fiscus, B. Antonishek, M. Michel, W. Kraaij, A.F. Smeaton, G. Quénot (eds.) TRECVID. National Institute of Standards and Technology (NIST), Gaithersburg, MD (2010)
21. Kalamaras, I., Mademlis, A., Malassiotis, S., Tzovaras, D.: A novel framework for retrieval and interactive visualization of multimodal data. Electron. Lett. Comput. Vis. Image Anal. **12**(2) (2013). http://elcvia.cvc.uab.es/article/view/518
22. Kankanhalli, M., Wang, J., Jain, R.: Experiential sampling on multiple data streams. Multimedia, IEEE Transactions on **8**(5), 947–955 (2006)
23. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Proceedings of the 9th International Workshop on Machine Learning, ML '92, pp. 249–256. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1992). http://dl.acm.org/citation.cfm?id=645525.656966
24. Kittler, J.: Feature set search algorithms. In: Chen, C.H. (ed.) Pattern Recognition and Signal Processing, pp. 41–60. Sijthoff & Noordhoff International Publishers B.V., Alphen aan den Rijn, The Netherlands (1978)
25. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**, 226–239 (1998)
26. Kludas, J., Bruno, E., Marchand-Maillet, S.: Information fusion in multimedia information retrieval. In: Proceedings of the 5th International Workshop on Adaptive Multimedia Retrieval (AMR). Paris, France (2007)
27. Kludas, J., Bruno, E., Marchand-Maillet, S.: Can feature information interaction help for information fusion in multimedia problems?. Multimedia Tools Appl. **42**, 57–71 (2009)
28. Kong, D., Ding, C., Huang, H., Zhao, H.: Multi-label relieff and f-statistic feature selections for image annotation. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference, pp. 2352 –2359 (2012). doi:10.1109/CVPR.2012.6247947
29. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: Proceedings of the European Conference on Machine Learning, pp. 171–182. Springer, New York, Inc., Secaucus, NJ, USA (1994). http://dl.acm.org/citation.cfm?id=188408.188427
30. Liu, H., Motoda, H., Yu, L.: A selective sampling approach to active feature selection. Artif. Intell. 159, 49–74 (2004). doi:10.1016/j.artint.2004.05.009. http://dl.acm.org/citation.cfm?id=1039211.1039214
31. Atrey, P., Hossain, M., Saddik, A.E., Kankanhalli, M.: Multimodal fusion for multimedia analysis: a survey. Multimedia Systems **16**, 345–379 (2010)
32. Moulin, C., Largeron, C., Ducottet, C., Géry, M., Barat, C.: Fisher linear discriminant analysis for text-image combination in multimedia information retrieval. Pattern Recognit. **47**(1), 260–269 (2014). doi:10.1016/j.patcog.2013.06.003. http://www.sciencedirect.com/science/article/pii/S0031320313002550
33. MPEG: Mpeg-7 reference software experimentation model (2003). http://standards.iso.org/ittf/PubliclyAvailableStandards/c035364_ISO_IEC_15938-6(E)_Reference_Software.zip
34. Natarajan, P., Manohar, V., Wu, S., Tsakalidis, S., Vitaladevuni, S.N., Zhuang, X., Prasad, R., Ye, G., Liu, D., Jhuo, I., Chang, S., Izadinia, H., Saleemi, I., Shah, M., White, B., Yeh, T., Davis, L.: Bbn viser trecvid 2011 multimedia event detection system. In: NIST TRECVID Workshop. Gaithersburg, MD (2011)
35. Over, P., Awad, G., Kraaij, W., Smeaton, A.F.: Trecvid 2007—overview. In: Over, P., Awad, G., Kraaij, W., Smeaton, A.F. (eds.) TRECVID. National Institute of Standards and Technology (NIST), Gaithersburg, MD (2007)
36. Over, P., Awad, G., Rose, R.T., Fiscus, J.G., Kraaij, W., Smeaton, A.F.: Trecvid 2008—goals, tasks, data, evaluation mechanisms and metrics. In: Over, P., Awad, G., Rose, R.T., Fiscus, J.G., Kraaij, W., Smeaton, A.F. (eds.) TRECVID. National Institute of Standards and Technology (NIST), Gaithersburg, MD (2008)
37. Poh, N., Kittler, J.: Multimodal Information Fusion: Theory and Applications for Human-Computer Interaction, chap 8, pp. 153–169. Academic Press, (2010)
38. Quinlan, J.R.: Induction of decision trees. Mach. Learn. 1, 81–106 (1986). doi:10.1023/A:1022643204877. http://dl.acm.org/citation.cfm?id=637962.637969

39. Rahman, M., You, D., Simpson, M., Antani, S., Demner-Fushman, D., Thoma, G.: Multimodal biomedical image retrieval using hierarchical classification and modality fusion. Int. J. Multimedia Info. Retr. **2**(3), 159–173 (2013). doi:10.1007/s13735-013-0038-4
40. Robnik-Sikonja, M., Kononenko, I.: An adaptation of relief for attribute estimation in regression. In: Fisher, D.H. (ed.) ICML, pp. 296–304. Morgan Kaufmann, San Francisco (1997)
41. Robnik-Šikonja, M., Kononenko, I.: Theoretical and empirical analysis of relieff and rrelieff. Mach. Learn. **53**, 23–69 (2003). doi:10.1023/A:1025667309714. http://dl.acm.org/citation.cfm?id=940854.940876
42. Saeys, Y., Inza, I.n., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics **23**, 2507–2517 (2007). doi:10.1093/bioinformatics/btm344. http://dl.acm.org/citation.cfm?id=1349154.1349169
43. Sikonja, M.R.: Speeding up relief algorithm with k-d trees. In: Proceedings of Electrotechnical and Computer Science Conference (ERK'98), pp. 137–140 (1998)
44. Snidaro, L., Niu, R., Foresti, G., Varshney, P.: Quality-based fusion of multiple video sensors for video surveillance. SMC-B: Cybernetics, IEEE Trans. on **37**(4), 1044–1051 (2007)
45. Snoek, C.G.M., Worring, M.: Multimodal video indexing: A review of the state-of-the-art. Multimedia Tools and Applications **25**(1), 5–35 (2005)
46. Sun, Y.: Iterative relief for feature weighting: Algorithms, theories, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1035–1051 (2007)
47. Temko, A., Macho, D., Nadeu, C.: Fuzzy integral based information fusion for classification of highly confusable non-speech sounds. Pattern Recognit. **41**(5), 1814–1823 (2008). doi:10.1016/j.patcog.2007.10.026. http://www.sciencedirect.com/science/article/pii/S003132030700489X
48. Tsoumakas, G., Katakis, I., Vlahavas, I.P.: Mining multi-label data. In: Maimon, O., Rokach, L. (eds.) Data Mining and Knowledge Discovery Handbook, pp. 667–685. Springer US, Berlin (2010)
49. Tumer, K., Ghosh, J.: Linear and order statistics combiners for pattern classification. CoRR **cs.NE/9905012** (1999). http://dblp.uni-trier.de
50. Wang, L., Zhou, N., Chu, F.: A general wrapper approach to selection of class-dependent features. IEEE Transactions on Neural Networks **19**(7), 1267–1278 (2008)
51. Wu Q., Wang Z., Deng F., Chi Z., Feng D.: (2013) Realistic human action recognition with multimodal feature selection and fusion. Syst. Man Cybern. Syst. IEEE Trans. **43**(4), 875–885. doi:10.1109/TSMCA.2012.2226575
52. Wu, Y., Chang, E.Y., Chang, K.C.C., Smith, J.R.: Optimal multimodal fusion for multimedia data analysis. In: Proceedings of the 12th ACM Multimedia, pp. 572–579. ACM, New York, NY, USA (2004)
53. Yan, R., Hauptmann, A.G.: The combination limit in multimedia retrieval. In: Proceedings of the 11th ACM International Conference on Multimedia, MULTIMEDIA '03, pp. 339–342. ACM, New York, NY, USA (2003)
54. Yilmaz, T., Gulen, E., Yazici, A., Kitsuregawa, M.: A relief-based modality weighting approach for multimodal information retrieval. In: Proceedings of the 2nd ACM International Conference on Multimedia Retrieval, ICMR '12, pp. 54:1–54:8. ACM, New York, NY, USA (2012). doi:10.1145/2324796.2324858
55. Yilmaz, T., Yazici, A., Yildirim, Y.: Exploiting class-specific features in multi-feature dissimilarity space for efficient querying of images. In: Christiansen, H., Tré, G., Yazici, A., Zadrozny, S., Andreasen, T., Larsen, H. (eds.) Flexible Query Answering Systems, Lecture Notes in Computer Science, vol. 7022, pp. 149–161. Springer, Berlin, Heidelberg (2011). doi:10.1007/978-3-642-24764-4_14