

Merging Rank Lists from Multiple Sources in Video Classification*

Wei-Hao Lin and Alexander Hauptmann
 Language Technologies Institute
 School of Computer Science
 Carnegie Mellon University
 5000 Forbes Avenue
 Pittsburgh, PA 15213, U.S.A.
 {whlin, alex}@cs.cmu.edu

Abstract

Multimedia corpora increasingly consist of data from multiple sources, with different characteristics that can be exploited by specialized applications. This paper focuses on video classification over multiple-source collections, and addresses the question whether classifiers should train from individual sources or from a full data set across all sources. If training separately, how can rank lists from different sources be merged effectively? We formulate the problem of merging ranked lists as learning a function mapping from local scores to global scores, and propose a learning method based on logistic regression. In our experiments we find that source characteristics are very important for video classification. Moreover, our method of learning mapping functions perform significant better than merging methods without explicitly learning the mapping functions.

1. Introduction

Multimedia collections are quickly accumulated with the ease of creating multimedia content. While most research has focused on uniform corpora, heterogeneous sources are typical in the real world. In broadcast news, various networks and channels can be accumulated and searched, but the results must be combined and delivered to the users in a single ranked list. This paper is concerned with the combination of results from different sources in order to exploit the source characteristics, and hopefully to improve the performance of video classification. Previous works in Text

*This work was supported in part by the Advanced Research and Development Activity (ARDA) under contract number MDA908-00-C-0037.

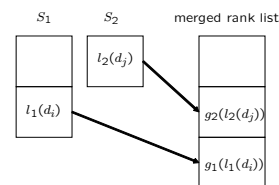


Figure 1. Merging two ranked lists

Retrieval often make strong assumptions that scores or ranks of the classifiers from each source are comparable [2], and techniques utilizing text-specific statistics [1] or summaries [8] cannot be directly applied to multimedia domain. In this paper, we formulate the problem of merging rank lists as score mapping, and propose a method to learn a mapping function using logistic regression. The experimental results show that merging methods based on learned mapping functions significantly outperform the methods without learning such mapping functions.

2. Merging Rank Lists

We formulate the task of merging rank lists as a score mapping problem, as illustrated in Figure 1. Suppose all videos in the corpora belong to one of the sources S_k ($k = 1, 2$ here). A classifier is trained separately for each source, which learns a function $l_k(d)$ that assigns a similarity score to each video shot d in the source, and returns a rank list in the order of the local scores. Since the scores in one ranked list are not necessarily comparable to scores from the other sources, a mapping function $g_k(x)$ is required to map the local scores to comparable global scores. The final merged list is sorted in the order of the global scores. Many widely-used merging methods, as well as our proposed

methods based on logistic regression, can be explained in this framework with different choices of $l_k(d)$ and $g_k(x)$, as listed in Table 1.

Merging Method	$g(x)$	$l(d)$
Round Robin	x	$-\text{rank}(d)$
Raw Score	x	$\text{score}(d)$
Linear Scaling	$\frac{x - \min_i \text{score}(d_i)}{\max_i \text{score}(d_i) - \min_i \text{score}(d_i)}$	$\text{score}(d)$
Logistic Regression	$\frac{1}{1 + \exp(-a - b \cdot x)}$	$\text{score}(d)$

Table 1. Merging methods can be explained as different combinations of the mapping function $g(x)$ and local score $l(d)$

One principle of designing a mapping function is to preserve rank before and after mapping. If a video clip d_i is ranked lower than d_j in the single-source rank list, the rank of d_i has to be lower than the rank of d_j in the final merged rank list. If additional knowledge is available to alter the order of a rank list and improve performance, the knowledge should reasonably be able to be applied locally before the merging stage. Therefore we should always preserve rank.

2.1. Merging Methods without Learning Mapping Functions

Round Robin Round Robin works as follows: we pick up the top-ranked video shot in the first rank list, and then select the top-ranked shot in the second list. After all top ranked shots have been selected, we start to select the second-ranked shot in the first rank list, and so on until add shots are selected.

Raw Score The degree of confidence that a classifier assigns to a shot may be better reflected in scores rather than rank. Raw Score takes the local scores from each rank list, and sorts the combined rank list in the order of the scores.

Linear Scaling Without explicit mapping functions, Raw Score assumes that the local scores from one source are comparable to the scores from all other sources, which usually does not hold true in practice. Linear Scaling is a crude way to normalize the local scores into the range between zero and one, which satisfies the rank preserving principle.

2.2. Learning Mapping Functions

We propose to learn the score mapping functions instead of relying on strong assumptions or simple normalization. Learning mapping functions can be seen as a regression problem.

Logistic Regression For each source, a classifier is trained in k -fold cross-validation fashion. In each fold, the trained classifier is applied to the testing data of the corresponding fold, and the local scores, as well as their labels (positive as one and negative as zero) are collected as training data for logistic regression¹. Logistic regression is fit with two parameters a and b .

The first reason of choosing logistic regression over linear regression is that the range of output values of logistic regression is restricted to be between zero and one, which makes scores comparable across sources, while linear regression does not limit the range. Secondly, the non-linearity of the Sigmoid function fits data with zero/one values better than linear regression. Thirdly, the classification performance of a classifier on the training data is reflected in the curve fitting. The fitting of logistic regression on a well performed classifier will have output values more close to zero or one, while the output values from a poorly fit classifier will be more spread. Note that the Sigmoid function is a monotonic function, which obeys the rank preserving principle.

2.3. Optimal and Random Merging Performance

One may be curious as to the best performance we can achieve in merging multiple lists as well as a random baseline. The random performance of merging rank lists, by definition, is to merge individual rank lists from each source and shuffle the resulting merged list randomly. The optimal way of merging rank lists in terms of maximizing the evaluation metric, average precision, can be formulated as the following search problem.

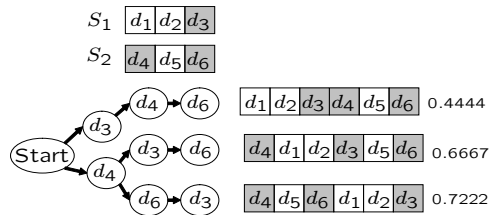


Figure 2. Searching for the optimal way to merge two rank lists from S_1 and S_2

Suppose we are looking for the optimal way to merge two rank lists (d_1, d_2, d_3) and (d_4, d_5, d_6) returned by

¹We do not use the local scores generated by a classifier built on full training data because these scores will be over-fit.

the classifiers trained for sources S_1 and S_2 , respectively, in the decreasing order of classifier’s “confidence”, as illustrated in Figure 2. The actually positive clips are drawn in a shadow box. According to the principle of preserving rank, first we choose either d_3 or d_4 into the merged list, but not d_6 . If we choose d_3 first, we have to include d_1 and d_2 and rank them higher than d_3 , again, by the principle of preserving rank. This search process can continue and be well presented as a search tree, as shown in the lower part of Figure 2. The number at the end of each merged rank list is the average precision.

Any search algorithms [7] can be used to find the solution based on the requirements of time and space complexity, optimality, and completeness. In this paper we choose an approximation method based on greedy search that always expands the positive clip with the highest precision. The merged list found by the greedy algorithm, called Greedy Bound, may be suboptimal, but the algorithm is very efficient in terms of time and space complexity.

3. Experiments

3.1. Testbed, Tasks, and Evaluation Metric

We choose the video corpus for TRECVID 2003 [6] as the testbed in this paper. The corpus is consisted of broadcast news programs from three sources: ABC, CNN, and C-SPAN. The definition of two classification tasks are listed as follows,

Sporting Event shot contains video of one or more organized sporting events

Weather News shot reports on the weather

The basis statistics in the training and testing set² are listed in Table 2. The shot boundaries of the training set are defined by common annotations, and those of the testing data by NIST. C-SPAN data are not included here because there are no sporting event or weather news shots in the source. The labels are from collaborate annotations by TRECVID 2003 participants. Note that the positive examples are very rare in the training data, around 1% in both tasks, which classifier will have hard time learning the concept.

We adopts Average Precision (AP) our evaluation metric as TRECVID does. AP of a rank list \mathcal{A} is de-

²The number of the positive examples in the test set is underestimated because TREC used a pooling method to evaluate participants’ submissions.

Set	Source	Task	Positive	Total
Training	ABC	Sporting Event Weather News	303 71	25630
	CNN	Sporting Event Weather News	303 215	21696
Testing	ABC	Sporting Event Weather News	26 7	16593
	CNN	Sporting Event Weather News	559 159	15282

Table 2. Basic statistics of two video classification tasks in the TRECVID 2003

finied as follows,

$$AP(\mathcal{A}) = \frac{1}{|\mathcal{A}^+|} \sum_{d \in \mathcal{A}^+} \frac{U^+(d) + 1}{U(d) + 1} \quad (1)$$

where \mathcal{A}^+ is a set of all positive examples in \mathcal{A} , $U(d)$ is a function returning the number of examples ranked higher than the example d in \mathcal{A} , and $U^+(d)$ is a function returning only the number of positive examples ranked higher than d . The upper bound of AP of merging multiple rank lists can be found as described in Section 2.3.

3.2. Features

We extract two types of features for each video shot.

Text Feature News programs in the TRECVID 2003 corpus come with closed captions or transcripts from speech recognition systems. The words in each shot are represented as a feature vector. Stop words are removed, and the Porter stemming algorithm is used to remove morphological variants. Term Frequency is used to reflect the importance of the words in the shot. The whole feature vector is normalized by unit length.

Color Feature One keyframe is chosen for each shot, and color features are extracted from this keyframe. The keyframe is dissected into 5 by 5 grids, and color histogram in HVC color space are calculated (using only H and C values). The feature vector is consists of the mean and the variance of the 125-bin color histogram in the grid, resulting in 50-dimensional vectors.

3.3. Classifiers

We use Support Vector Machine (SVM) as the classification algorithm. SVM has been widely used and very effective in many domains, including Text Categorization[4] and Video Classification[5]. The basic idea behind SVM is to select the decision hyperplane

Merging Methods	Sporting, Text	Weather, Text	Weather, Color
Round Robin	0.034	0.449	0.225
Raw Score	0.042	0.859	0.435
Linear Scaling	0.008	0.745	0.385
Logistic regression	0.064	0.854	0.467
Uni-Modal Training	0.027	0.858	0.384
Greedy Bound	0.067	0.864	0.467
Random Baseline	0.022	0.008	0.008

Table 3. Experimental Results of Merging Rank Lists from Multiple Sources

in the feature space that can separate two classes of data points while keeping the margin as large as possible. The process of finding the hyperplane can be formulated as the following optimization problem,

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i \quad (2)$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1, \dots, l$$

where x_i is a feature vector, $i = 1, \dots, l$, l is the size of the training data, $y_i \in \{+1, -1\}$, y_i is +1 when the shot is positive example, and -1 otherwise, ϕ is the kernel function that maps the feature vector into higher dimension, ξ_i is the degree of misclassification when the data points fall in the wrong side of the decision boundary, and C is the penalty parameter that tradeoff between two terms. More details can be found in [3]. Note that the choice of the classifier depends only on the task at hand and any classifiers can be plugged in the learning method in Section 2.2.

3.4. Results

The results of merging rank lists using different merging methods under various combinations of features and tasks³ are shown in Table 3. In each column, we also list the random baseline and the upper bound found by the greedy search algorithm. The best performance other than Greedy Bound is marked in bold.

Overall speaking, Logistic Regression methods are the best among four methods, which suggest the effectiveness of learning global mapping functions. Blindly assuming the local scores are comparable across sources (Round Robin, Raw Score) or simply scaling the local scores (Linear Scaling) hurt the performance greatly and should be avoided in practice.

³The results of merging ABC and CNN’s color classifiers in the Sporting Event classification task are so close to the random baseline that the table was omitted here

We also conducted experiments to compare the performance of merging separate sources vs. training on all data without discerning video sources (Uni-Modal Training in Table 3). The results show that merging rank lists significantly outperforms uni-modal training that ignores the source differences, which strongly suggests the importance of exploiting source characteristics.

Merging text-based classifiers in the Weather News task seems to be an exception, but it is not surprising at all because different news channels usually use very similar terminology to present weather reports, such as “temperature”, “snow”, etc, and thus there are little source characteristics left to be exploited.

4. Conclusions

In this paper we showed that source characteristics can provide valuable information for video classification. Furthermore, merging methods should be carefully designed and chosen because the choice significantly affects performance. Among all merging methods, our proposed method of learning the mapping function using logistic regression significantly outperforms those without learning the mapping function.

References

- [1] J. Callan. *Advances in Information Retrieval*, chapter Distributed Information Retrieval, pages 127–150. Kluwer Academic Publishers, 2000.
- [2] A. Chen. Cross-language retrieval experiments at CLEF 2002. In *Working Notes for the CLEF 2002 Workshop*, 2002.
- [3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
- [4] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning (ECML)*. Springer, 1998.
- [5] W.-H. Lin and A. Hauptmann. News video classification using SVM-based multimodal classifiers and combination strategies. In *Proceedings of the tenth ACM international conference on multimedia*, Juan-les-Pins, France, December 1-6 2002.
- [6] NIST. Guidelines for the TRECVID 2003 evaluation. Webpage, 2003. <http://www-nlpir.nist.gov/projects/tv2003/tv2003.html>.
- [7] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2nd edition, 2002.
- [8] X. M. Shou and M. Sanderson. Experiments on data fusion using headline information. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, University of Tampere, Finland, August 11 - 15 2002.