Fast Video Retrieval Under Sparse Training Data

Yan Liu and John R. Kender

450 Computer Science Building, 1214 Amsterdam Avenue, New York, NY, 10027, U.S.A
{liuyan, jrk}@cs.columbia.edu

Abstract. Feature selection for video retrieval applications is impractical with existing techniques, because of their high time complexity and their failure on the relatively sparse training data that is available given video data size. In this paper we present a novel heuristic method for selecting image features for video, called the Complement Sort-Merge Tree (CSMT). It combines the virtues of a wrapper model approach for better accuracy with those of a filter method approach for incrementally deriving the appropriate features quickly. A novel combination of Fastmap for dimensionality reduction and Mahalanobis distance for likelihood determination is used as the induction algorithm. The time cost of CSMT is linear in the number of features and in the size of the training set, which is very reasonable. We apply CSMT to the domain of fast video retrieval of extended (75 minutes) instructional videos, and demonstrate its high accuracy in classifying frames.

1 Introduction

The rapid growth and wide application of digital video has led to a significant need for efficient video data set management. The problem of efficient retrieval and manipulation of semantically labelled video segments is an important issue.

One typical approach is to use existing image retrieval algorithms, starting from a good segmentation of the video into shots and then selecting certain images of the shots as key-frames [1]. But as Lew et al mentioned in [3], the gap between the high level query from the human and the low-level features persists because of a lack of a good understanding of the "meanings" of the video, of the "meaning" of a query, and of the way a result can incorporate the user's knowledge, personal preferences, and emotional tone. Machine learning methods such as classification [4] and boosting [5] are introduced to help retrieve matching video sequences semantically, and some methods use audio information analysis and text extraction and recognition as well. But there appears to be little work that supports efficient feature selection for video retrieval, due to the huge volume of data.

Researchers therefore work on speeding up their algorithms; one way is by seeking efficient ways of reducing the dimensionality of the data prior to classification and retrieval. Vailaya et al [6] and Smeulders et al [7] discuss this problem from the view of image processing and computer vision. They assume that some features, such as color histograms or texture energies, are more sensitive than others, based on the

researchers' intuition. They provide theoretical analyses and empirical validations for their choices, but this approach is difficult to extend to other domains where the relationships between features and categories are unclear and changeable.

The heart of this paper is a novel feature selection algorithm, which focuses on selecting representative features in the massive and complex dataset automatically; no manual definition or construction of features is required [8]. This form of learning has received significant attention in the AI literature recently and has been applied to moderately large data sets in applications like text categorization [8] and genomic microarray [9] analysis. Another emphasis of this paper is that our novel selection algorithm also addresses the problem of sparse and noisy training data. The training sets available for the learning of semantic labels is a very small fraction of the total in video retrieval. Classification using sparse training data is a classical problem of machine learning and few papers [9] support feature selection under these circumstances.

This paper is organized as follows. Section 2 introduces some related work in feature selection. Section 3 proposes the feature selection algorithm, CSMT, and provides a framework for video retrieval using this algorithm. Section 4 presents empirical validation of the accuracy of algorithm when applied to the particular genre of instructional videos, and validates the algorithm in a generic data. We close the paper with discussion and planned future work in section 5.

2 Related work of feature selection

There appears to be two major approaches to the feature selection problem. The first emphasizes the discovery of any relevant relationship between the features and the concept, whereas the second explicitly seeks a feature subset that minimizes prediction error of the concept. The first is referred to as a filter method, and the second approach is referred to as a wrapper method. In general, wrapper methods attempt to optimize directly the classifier performance so that they can perform better than filter algorithms, but they require more computation time. Seen in this context, this paper proposes a wrapper feature selection method with time cost considerably less than that of filter methods. For an alternative viewpoint, see [13].

Feature selection methods are typically designed and evaluated with respect to the accuracy and cost of their three components: their search algorithm, their statistical relationship method (in the case of filter methods) or their induction algorithm (in the case of wrapper methods), and their evaluation metric (which is simply prediction error in the case of wrapper methods). The dominating cost of any method, however, is that of the search algorithm, since feature selection is fundamentally a question of choosing one specific subset of features from the power set of features. So far, three general kinds of heuristic search algorithms have been used: forward selection, backward elimination, and genetic algorithms.

Sparse training data is a hard problem in machine learning. As Xing et al mentioned in [9], the number of replicates in some experiments is often severely limited; he gives a real world problem in which only 38 observation vectors exist, each one encoding the expression levels of 7130 features. Feature selection when there are so few observations on so many features is very different from the more general cases typical in the learning literature; it even renders some powerful algorithms ineffective. This is easy to see in Xing's case: since there are only 38 observations, no matter what feature set has been chosen the prediction error is severely quantized to one of 39 levels. With 7130 features, on average we could expect about 183 features to produce each of these error levels; either a forward or backward wrapper method will be forced to choose randomly over this large set at each iteration. If ultimately we wish only a small set of about. 50 features to avoid overleaning as in [9], too much randomness is introduced by these methods. Moreover, randomness accumulates, with the chose of each feature heavily influencing the choice of its successors.

The alternatives to wrapper methods are filter methods, which select feature subset independently of the actual induction algorithm. Koller and Sahami in [2] employ a cross-entropy measure, designed to find Markov blankets of features using a backward greedy algorithm. In theory going backward from the full set of features may capture interesting features more easily [12], especially under sparse training data. However, in Xing's case this means that if we want a target feature space with 50 features, we have to remove 7080. To avoid this expensive time cost, Xing proposes to sort the 7130 features based on their individual information gain in classification (a wrapper method), but then abandons the wrapper approach, and uses only the best N features in a series of filter methods. Additionally, he selects N=360 manually. It is not clear how well such a technique generalizes or how effective it is, given its mixture of models.

3 Feature Selection for Video Retrieval

This section presents a method of efficient semantic video retrieval, based on automatically learned feature selection. This novel feature selection algorithm, called CSMT combines the strengths of both filter and wrapper models, and exploits several properties unique to video data.

3.1 Complement Sort-Merge Tree

Our overall approach is to use an outer wrapper model for high accuracy, and an inner filter method for resolving the problem of random selection when the training set is relatively small and errors are quantized (as they inevitably are for video data).

This Complement Sort-Merge Tree (CSMT) algorithm combines the features of forward selection, backward elimination, and genetic algorithms. To avoid irrevocable adding or subtracting, it always operates on some representation of the original feature space, so that at each step every feature has an opportunity to impact

the selection. To avoid heuristic randomness, at each step a complement test is used to govern subset formation. The tree structure of the CSMT leads to low time cost. Further, the recursive nature of the method enables the straightforward creation of a hierarchical family of feature subsets with little additional work. The entire CSMT of progressively more accurate feature subsets can be stored in space O(N), to be accessed when needed at a later time.

The CSMT algorithm can be divided into two parts: the creation of the full tree of feature subsets, and subsequent manipulation of the tree (if necessary) to create a feature subset of desired cardinality or accuracy. Each part uses a different heuristic greedy method.

Table 1 shows the CSMT basic algorithm. Initially, there are N singleton feature subsets. Using a wrapper method, their performance is evaluated on training data, and they are sorted in order of performance. Features are then paired into N/2 subsets of cardinality 2, by merging them according to the complement requirement. After another round of training, sorting, and pair-wise merging according to complement requirement, a third level of N/4 subsets of cardinality 4 are formed. The process continues until it attains a level or condition prespecified by the user, or until the entire tree is constructed.

Initialize level = 0
    Create N singleton feature subsets.
While level < $\log_2 N$
    Induce on every feature subset.
    Sort subsets based on their classification accuracy.
    Choose pairs of feature subsets based on the complement requirement.
    Merge to new feature subsets.

Table 1. Complement Sort-Merge Tree

Select the leftmost branch of size $2^{\lfloor \log 2r \rfloor}$.
Initialize cutout = $2^{\lceil \log 2\, r \rceil}$ - r.
While cutout >0
    Let branch-size = $2^{\lfloor \log 2\ \text{cutout} \rfloor}$.
    For all remaining branches of this size, evaluate the induction result of removing those branches individually.
    Remove the branch with best result.
    Let cutout = cutout – branch-size.

Table 2. Algorithm to select exactly r features from the tree of feature subsets.

Figure 1 illustrates the algorithm with an initial set of features with cardinality N = 256. Table 2 shows the related algorithm that further manipulates the full CSMT tree

if it is necessary to select exactly r features (r not a power of 2) from the hierarchy of feature subsets. It is not hard to show that the time cost of the search algorithm of CSMT is linear in the number of nodes in the Sort-Merge tree, i.e., $T \sim O(N*T_m)$, where T_m is the induction time complexity using m training data.

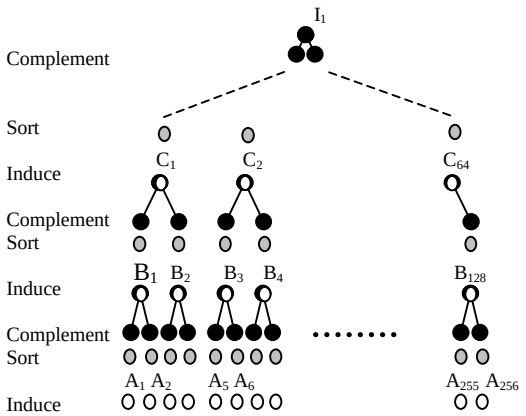


Fig. 1. CSMT for N=256. Leaves correspond to singleton feature subsets. White nodes are unsorted feature subsets and gray nodes are the white nodes rank-ordered by performance. Black nodes are the pairwise mergers of gray nodes, with pairs formed under the complement requirement.

Figure 2 illustrates the complement test, which uses a filter method to inform the otherwise random selection of feature subsets. It employs a heuristic approximation to a markov blanket that attempts to maximize classification performance on the m training samples. An m-length performance vector records for each feature subset correct classifications with a 1 and failures with a 0. Any feature subset seeking a complementary feature subset will examine all unpaired feature subsets sharing identical error rates with it. It then selects from these that feature subset which maximizes the number of 1s in the OR of their two performance vectors. These complementary feature subsets are then merged. This step of the CSMT method is a greedy algorithm, but one that is more informed than random choice.

3.2 Induction Algorithm for Feature Selection

The performance of a wrapper feature selection algorithm not only depends on the search method, but also on the induction algorithm. For our induction method during the course of the learning we use a novel, low-cost, and scalable combination of Fastmap for dimensionality reduction with Mahalanobis maximum likelihood for classification. We refer readers to the literature for a detailed explanation of these two component methods, but summarize their significance here.

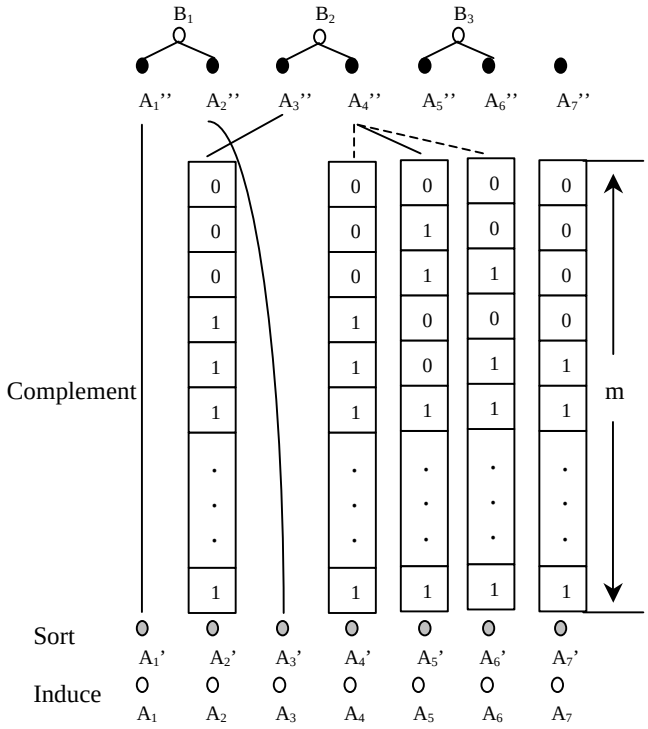


Fig. 2. The complement requirement, illustrated for the A level of Figure 1. The sorted singletons A_1' and A_3' have already been paired to form pair B_1. To find the best complementary feature subset for A_2', examine all sorted subsets (A_4,' A_5', A_6') with the same error rate on the m training samples. The bitwise OR of performance vectors of A_2' and A_5' maximizes performance coverage; A_5' complements A_2' for B_2.

The fastmap method proposed in [10] approximates Principal Component Analysis (PCA), with only linear cost in the number of reduced dimensions sought, c, and in the number of features, N. The method heuristically replaces the computation of the PCA eigenvector of greatest eigenvalue, which represents the direction in the full feature space that has maximum variation, with a (linear) search for the two data elements that are maximally separated in the space. The vector between these two elements is taken as a substitute for the eigenvector of greatest eigenvalue, and the full space is then projected onto the subspace orthogonal to this substitute vector for the first eigen dimension. The process then repeats for a desired and usually small number of times. By the use of clever bookkeeping techniques, each additional new dimension and projection takes time approximately linear in the number of features.

In brief, as defined in statistical texts Duda et al. [11], or in the documentation of Matlab, the Mahalanobis distance computes the likelihood that a point belongs to a distribution that is modeled as a multidimensional Gaussian with arbitrary covariance. During training, each image frame in a training set for a video category is first mapped to a point in the space of reduced dimension c. Then the distribution of these

mapped points is approximated by a c-dimensional Gaussian with a non-diagonal covariance matrix. Multiple categories and training sets are represented each with their own Gaussian distribution. The classification of a test image frame is obtained by mapping it, too, into the reduced c-dimensional space, and then calculating the most likely distribution to which it belongs. The time cost is also linear with the number of features N.

3.3 Framework of Video Retrieval Using CSMT

The linear time cost and the increased accuracy of the complement requirement allow an efficient and effective implementation of video retrieval under sparse training data. In this section, we demonstrate the CSMT on two retrieval tasks on MPEG1-encoded instructional videos.

First, in our application and in general, the video may be down-sampled temporally, spatially, and/or spectrally. We temporally subsample by using only every other I frame (that is, one I frame per second). We spatially subsample by a factor of 16 in each direction by using only using the DC terms of each macro-block of the I frame (consisting of six terms, one from each block: four luminance DC terms and two chrominance DC terms); this subsampling is very popular in video retrieval [1]. This gives us, for each second of video, 300 macroblocks (15 by 20) of 6 bytes (4 plus 2) of data: 1800 initial features. For convenience of decoding, we consider the 6 DC terms from the same macro-block to be an un-decomposable vector, so our initial data consists more accurately of 300 six-dimensional features per second of video. Each of these 300 features is placed into its own subset to initialize the CSMT algorithm.

Second, using Fastmap, the dimensionality of each feature subset is reduced to a pre-specified small number, c, of dimensions. (This makes more sense after the first several steps.) Third, for each feature subset at this level, using the reduced dimensionality representation, the training sets of the video train the induction algorithm to classify the test sets of the video. Fourth, the feature subsets are sorted by accuracy. Pair-wise feature subsets are then merged, based on complement requirement. Fifth, the process repeats again, starting at the Fastmap step. It is clear that at most O(log N) iterations of this CSMT algorithm are necessary. Sixth, if needed, exactly r features are extracted from the tree of the feature subsets. Seventh, the frames of the learned category are retrieved from the video only using these r features.

4 Experiment

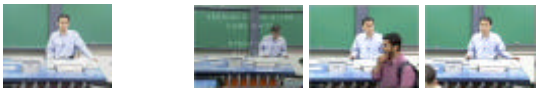
The extended instructional video mentioned above is of 75 minutes duration, which has approximately 4500 frames of data, with 300 six-dimensional features for each frame. Existing feature selection methods, which typically have been reported to run for several days on features sets of cardinality at least one decimal order of magnitude smaller, are intractable on video data; see Koller and Sahami [2]. Therefore, we

compared the retrieval accuracy of our novel method against an imperfect but feasible method, random feature selection; see the work of Xing et al who were similarly forced into such benchmarks [9]. These experiments use the same data and same classifiers; the only difference is how the feature subset was chosen.

In the first experiment, we attempt to retrieve about 200 “announcement” frames from the 4500 frames, without any prior temporal segmentation or other pre-processing. “announcement” frames look like Figure 3 (a); other video frames look like Figure 4 (b). Although it should not be difficult to find a very small set of features to make these distinctions, we want to do so rapidly and accurately. Only 80 training frames are provided (40 “announcement” and 40 others), and as shown in Figure 4, they include considerable noise.

(a) **(b)**

Fig. 3. Task: Retrieve “announcements”(a) from an entire video with competing image types (b).



Normal data Noise data

Fig. 4. Training data also includes noise.

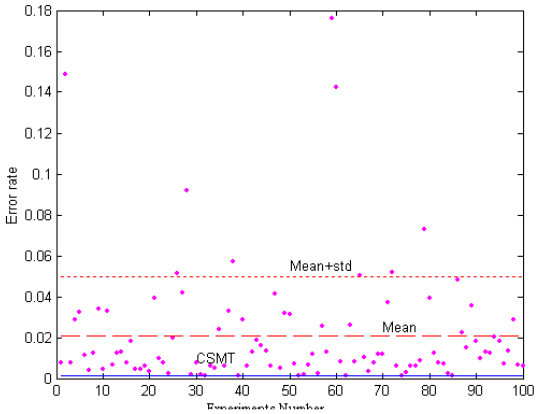
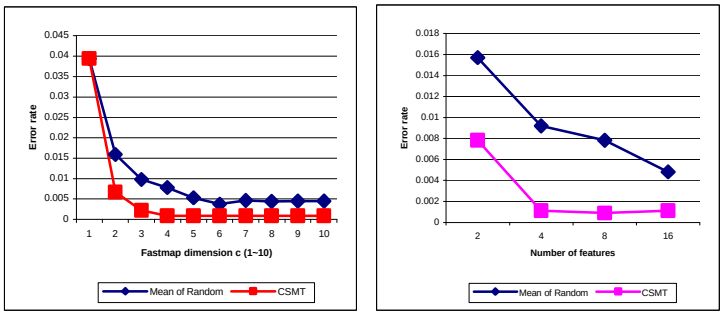


Fig. 5. Error rate of CSMT vs. random for retrieval of “announcements” with features r = 4 and Fastmap dimension c = 2.

Figure 5 compares the retrieval results using only 4 features, when Fastmap dimension c is equal to 2. Points show the error rate of 100 experiments that select the features randomly. As expected, the rate of error is highly variable, with the standard deviation being larger than the mean. The error rate using features selected by CSMT, as a solid line, is clearly better. None of the results of random feature selection is better than CSMT. Figure 6 (a) compares the performance of different Fastmap dimensions from 1 to 10 using the same number of features. Figure 6 (b) fixes the Fastmap dimension c=4 and compares the classification error rate of different numbers. The performance of CSMT is much better than that of random selection in all cases.



(a) Fix r = 8 and c from 1 to 10 **(b)** Fix c = 4 and r = 2, 4, 8, 16

Fig. 6. Error rate of CSMT vs. random for retrieval of "announcements", with r features and c Fastmap dimensions.

In the second experiment not related to video retrieval, we tested the generality of the CSMT method on Xing's original data. Using his data, his definitions, and his evaluation method, CSMT obtains the same error rate compared with his 5.9%. but with much lower time complexity.

5 Conclusion

We have presented a low-cost feature selection algorithm CSMT that is well-suited for large data sets with sparse training data. It relies on the three algorithms working together in linear time of the features: Fastmap for dimensionality reduction, Mahalanobis distance for classification likelihood, and a sort-complement-merge sequence for combining relevant and non-redundant feature subsets into more accurate ones. CSMT combines the performance guarantees of a wrapper method with the speed and logical organization of a filter method. It therefore leads to new feasible approaches for rapid video retrieval. We have demonstrated some of its results on an extended video. We intend to investigate its utility both across a library

of videos of this kind, and also on other genres such as situation comedies which share a similar categorization structure.

References

1. Irena Koprinska and Sergio Carrato,: Temporal video segmentation: A survey. Signal processing: Image communication 16, (2001) 477-500.
2. Koller, D. & Sahami,M.: Toward optimal feature selection. Proceedings of the Thirteenth International Conference on Machine Learning (1996).
3. Michael S. Lew, Nicu Sebe, John P. Eakins.: Challenges of Image and Video Retrieval, International Conference on Image and Video Retrieval. Lecture Notes in Computer Science, vol. 2383, Springer (2002) 1-6.
4. Wensheng Zhou, Asha Vellakial and C.-C. Jay Kuo.: Rule-based video classification system for basketball video indexing. ACM Multimedia (2000).
5. Pickering, M., Ruger, S., Sinclair, D.: Video Retrieval by Feature Learning in Key Frames. International Conference on Image and Video Retrieval. Lecture Notes in Computer Science, vol. 2383, Springer (2002) 316-324.
6. A. Vailaya, M. Figueiredo, A. K. Jain, and H.-J. Zhang.: Image Classification for Contnet-Based Indexing. IEEE Transactions on Image Processing, vol. 10, no. 1, January, (2001) 117-130.
7. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain.: Content-based image retrieval: the end of the early years. IEEE trans. PAMI, 22 – 12 (2000) 1349 -- 1380.
8. Yiming Yang and Jan O. Pedersen: A comparative study on feature selection in text categorization. Proceedings of the Fourteenth International Conference on Machine Learning (1997) 412-420.
9. Eric P. Xing, Michael I. Jordan, Richard M. Karp: Feature selection for high-dimensional genomic microarray data. Proceedings of the Eighteenth International Conference on Machine Learning (2001).
10. Christons Faloutsos and king-Ip (David) Lin: FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. Proceedings of ACM SIGMOD (1995) 163-174.
11. Richard O. Duda, Peter E. Hart and David G. Stork: Pattern classification, Wiley, New York (2000).
12. R. Kohavi and G. H. John: Wrappers for feature subset selection. Artificial Intelligence, special issue on relevance (1997) 273-324.
13. Douglas Zongker and Anil K. Jain: Algorithms for Feature Selection: An Evaluation. In Proceedings of the 13th International Conference on Pattern Recognition, 1996.