Sort-Merge Feature Selection for Video Data

Yan Liu and John R. Kender
Department of Computer Science
Columbia University
New York, NY 10027
{liuyan, jrk}@cs.columbia.edu

Abstract

Applying existing feature selection algorithms to video classification is impractical. A novel algorithm called Basic Sort-Merge Tree (BSMT) is proposed to choose a very small subset of features for video classification in linear time in the number of features. We reduce the cardinality of the input data by sorting the individual features by their effectiveness in categorization, and then merging pairwise these features into feature sets of cardinality two. Repeating this Sort-Merge process several times results in the learning of a small-cardinality, efficient, but highly accurate feature set. As the wrapper model, this paper exploits a novel combination of Fastmap for dimensionality reduction and Mahalanobis distance for likelihood determination. The time complexity of this induction part is linear in the number of training data. We provide theoretical proof of time cost and empirical validation of the accuracy.

1. Introduction

The rapid growth and wide application of digital video has led to a significant need for efficient video data management. Bridging the semantic gap between the high level query from the human and the low-level information from the video data is a fundamental challenge in video understanding.

To reach these semantic goals, some machine learning methods such as classification and boosting have been attempted, in order to find features that associate image properties with user labels. But due to the high volume of video data, the time complexity of these methods has been prohibitively high. Researchers therefore have worked on speeding up their algorithms; one way has been by seeking efficient ways of reducing the dimensionality of the data prior to processing.

Vailaya and Smeulders, among others, 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 analysis and empirical validations for their choices, but this approach is difficult to extend to other domains because of need for human interaction.

The heart of this paper is an automatic feature selection algorithm, called the Basic Sort-Merge Tree (BSMT). The problem of feature selection has received significant attention in the AI literature recently, and various algorithms have been devised and applied to moderately large data sets. Learning research is not often carried out in video indexing and retrieval--although Lew et al [1] used a feature selection method to refine features for stereo image matching. This is because the sheer magnitude of video data has limited the choice and application of existing feature selection algorithm, which have been designed for smaller databases and which run inordinately long even on those. One emphasis of this paper is the low time cost of our heuristic method, which can exploits several properties unique to video data to induce appropriate but small feature sets.

This paper is organized as follows. Some related work in feature selection is introduced in Section 2. Section 3 proposes the feature selection algorithm, BSMT, and provides a framework for video analysis using this algorithm. Section 4 presents empirical validation of the accuracy and efficiency of algorithm when applied to the particular genre of instructional videos. We close the paper with discussion section 5.

2. Related work

2.1 Filter methods and wrapper method

There appears to be two major approaches to the feature selection problem. The first emphasizes the discovery of any relevant relationship between features and concept, whereas the second explicitly seeks a feature subset that minimizes prediction error. The first is referred to as a filter method, and it finds a feature subset independently of the actual induction algorithm that will use this subset for classification. Ordinarily, filter methods use simple statistics computed from the empirical feature distribution to select strongly relevant features and to filter out weakly relevant features before induction occurs; see Blum and Langley [2]. In contrast, a wrapper method searches the space of feature subsets, using cross-validation to compare the performance of a trained classifier on each tested subset, directly optimizing the induction algorithm that uses the subset for classification. As Xing et al. state in [3], wrapper methods attempt to optimize directly the predictor performance so that they can perform better than filter algorithms, but they require more computation time. Seen in this context, this paper proposes a novel sort-merge feature selection method with accuracy approaching that of a wrapper method but with a cost comparable to a filter method.

2.2. Feature selection algorithm design and evaluation

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. Forward selection starts with the empty set of features and successively adds individual features, usually following a variant of a greedy algorithm, terminating when no improvement is possible. However, it can not remove any features, and therefore ends up making what amounts to local optimizations to the growing set. Backward elimination, which does the reverse, starts with the full set of features and heuristically subtracts individual features. It suffers from a similar problem of local optimization, as removal of a feature is irrevocable. A genetic algorithm, which permits both the addition and deletion of features to a surviving population of evolving subsets of limited cardinality, is more likely to seek a global optimum. But it is computationally costly, and requires a more elaborate definition of algorithm convergence.

2.3. Apply feature selection to video data

No doubt, as general methods, feature selection algorithms can apply to any data. But some applications may have special characters and grow up to individual topics, such as feature selection of Genomic data [3] and feature selection of text data [4].

In this part, we will discuss feature set used in video classification currently. It is also the original feature space from which we will select feature subset. Because of length limitation of this paper, we only provide some background introduction of compressed video. Video in other format will be ignored and it is also easier to makeup the background by readers themselves; see [5].

The Moving Picture Expert Group (MPEG) standard is the most widely accepted international standard for digital video compression. Speaking in the simple way, an MPEG stream can be considered of a serious of GOPs, usually two GOP/sec. GOP (Group of Picture) consists of three types of pictures – I frame, P frame and B frame. In general, each GOP is led by one I frame, which is coded using information present in the picture itself, and followed by several P frames and about ten B frames coded using reference I or P frames; see [6].

Every picture can be divided into 8 * 8 blocks with 64 pixels and provided by Discrete Cosine Transform (DCT). The first DCT coefficient is called DC term, which is 8 times the average intensity of the respective block. Each four neighbors blocks form Macro-blocks (MB). The DCT coefficients of each MB are presented by four luminance arguments and two chrominance arguments. Similarly, the DC terms of each MB are consisting of four luminance DC terms, one from each block and two chrominance DC terms, sharing by four blocks. The other important definition should be mentioned is motion vector (MV), which is used in P frames as coefficient of motion compensation.

Features that are often used for compressed video classification have been grouped to six types [10]:

- DCT coefficients
- DC terms
- DC terms, MB coding modes and MVs
- DCT coefficients, MB coding mode and MVs
- MB coding mode and MVs
- MB coding mode and bit-rate information

Obviously, no matter which kind of feature sets are selected for classification, the native feature space and the instance space are massive. For example, Down-sample a MPEG-1 video of 320*240 pixels per frame both temporally and spatially by only using the DC terms of each macroblock of every other I frame. This gives us, for each second of video, 300 macroblocks (15 by 20) of 6 bytes (4 plus 2) of data. For the video of one hour, there are 7200 instances and each instance has 1800 initial features.

3. Feature selection for video data

3.1. Frame of applying BSMT to video data

The first difficulty of feature selection for video classification is data type of features. For feature selection of text categorization, features are Boolean value. The choice of feature selection methods is not limited by data type of features and the computational cost is much lower than video data, which are chrominance arguments from -256 to 256 and luminance arguments having larger range from -1024 to 1024. Of course, we can transfer continuous attributes to enumerable ones by looking for optimal threshold using information gain or other criteria, but it means more error will be introduced. In this paper, we use Mahalanobis likelihood as induction metric in the wrapper model since the Mahalanobis metric has good performance in classifying data with multiple dimensions, even if each dimension has a different range of feature values.

We refer readers to the literature for a detailed explanation of this method, but summarize its significance here. In brief, as defined in statistical texts Duda et al. [7], 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.

However, Classification using Mahalanobis likelihood requires the cardinality of the training set should be much larger than the number of features; the usual lower bound given by P. A. Devijver and J. Kittler in [8] for this cardinality is N(N-1)/2, where N is the number of features. It means providing mahalanobis metric directly in the original feature space is impossible while we only have limited training data.

Principal Component Analysis (PCA) is the usual method of choice for dimensionality reduction, but it carries high computational complexity. Instead, the Fastmap method proposed in [9] approximates PCA, with only linear cost in the number of reduced dimensions sought, c, and in the number of training data, m.

The heart of the feature selection algorithm for video data is how to search the high-dimensional feature space to get one specific feature subset. This is an exponentially hard problem since the completed search is the power of the features while heuristic approach seldom work well for video data since categorization of entire video frames, however, does not appear to be either straightforward or logical, and is further complicated by the redundancy of neighboring pixels. Next part addresses these two problems and present a novel algorithm to select features suitable for video classification with low time cost and space cost.

3.2. Basic Sort-Merge Tree

BSMT 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 greedy algorithm is used to govern subset formation. Further, the recursive nature of our method provides an additional advantage over existing methods, in that it enables the straightforward creation of near-optimal feature subsets of any or all given cardinalities or accuracies, with little additional work.

BSMT can be divided into two parts: the creation of a tree of feature subsets, and the manipulation of the tree to create a feature subset of desired cardinality or accuracy. Each part uses a heuristic greedy method.

Table 1 shows the algorithm of setting up the tree. Figure 1 illustrates a tree with N=256. Table 2 shows the algorithm of cutting the tree based on the application requirement, for example, to create a feature space with exactly r features.

```
Initialize level = 1
        N singleton feature subsets.
While level < $\log_2 N$
        Induce on every feature subset.
        Sort subsets based on their
        classification accuracy.
        Combine, pairwise, feature subsets.
```

Table 1. Setup Basic Sort-Merge Tree.

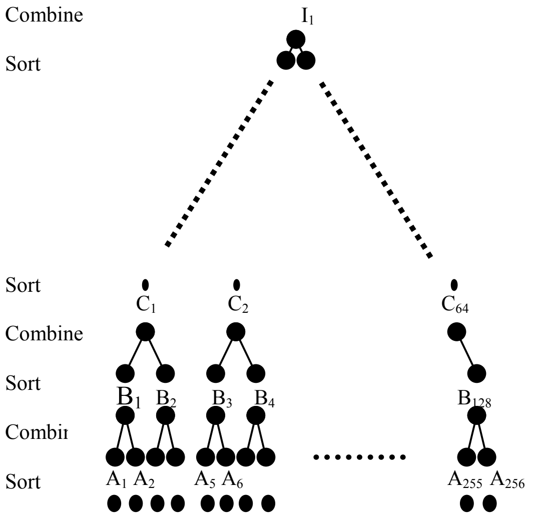

Figure 1. Setup of the Basic Sort-Merge Tree

```
Select the leftmost branch of size $2^{\lceil \log2r \rceil}$.
Initialize cutout = $2^{\lceil \log2 r \rceil}$ - r.
While cutout >0
        Let branch-size = $2^{\lfloor \log2 cutout \rfloor}$.
        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. Select exactly r features from BSMT.

3.3 Analysis of BSMT time complexity

We use the following definitions to analyze the time cost of BSMT.

N: Number of dimensions of the original feature space
r: Number of dimensions of the reduced feature space
m: Number of the training data
c: Number of dimensions extracted using the Fastmap algorithm
L: level number of Sort-Merge tree
T_m: Time of induction using m training data in the Mahalanobis classifier
T_{basic}: Time of the basic CSMT
T_{select}: Time of algorithm to select r features from the tree

We first show search algorithm is linear in the number of features, i.e., $T_{basic} = O(NT_m)$.

As shown in Table 1, we begin our induction with N feature subsets using m training data; the time complexity is $O(NT_m)$. Since in general, the time complexity of inducing using m training data is much larger than the time cost of sorting, and pair-wise merging, the time cost at this level can be replaced by $O(NT_m)$. As shown in Figure 1, the number of subsets in each level is $N/(2^{L-1})$. Since the cost of each level is proportional to the number of subsets, the time cost of each level is equal to $N/(2^{L-1}) * T_m$. It is clear that there are at most O(log N) levels. Summing these costs yields a total cost of $O(NT_m)$.

Similarly, one can show that the additional cost of $T_{select} = O(rT_m)$; this is dominated by T_{basic}. The argument is based again on the sum of a geometrically decreasing series of costs, proportional to evaluations of effects of pruning ever smaller numbers of subtrees.

Using Fastmap-Mahalanobis, the induction step is also linear in the number of training data, i.e., $T_m = O(mc^2)$. The cost of the Fastmap per subset is O(mc), based on the proof given in [9], and the cost of the Mahalanobis classification is $O(mc^2)$, based on the proof given in [7]. Thus, the cost of the induction is a fixed $T_m = O(mc^2)$.

4. Experiment

In our application, we have approximately 4500 seconds (units) of video to classify, 300 features for each unit, four classification categories, and about 400 units of training. 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 this dataset. Therefore, we compared the classification accuracy of our new method against two imperfect but feasible benchmarks, random feature selection, and hand feature selection: see the work of Xing et al who were similarly forced into such benchmarks [3].

For random feature selection, we ran 100 experiments in which 30 features were selected randomly and calculated the mean of classification error rate. For hand selection, 30 macro-blocks selected by hand, based on the intuition of the researchers. Figure 2 is a grand summary. The classification error rate of BSMT is not only less than that of hand selection and random selection, but also appears to be very stable as the Fastmap dimension varies: this is critical, as C must be fixed before hand. Figure 3 fixes the Fastmap dimension at c=4, and compares the classification error rate of different values of r.

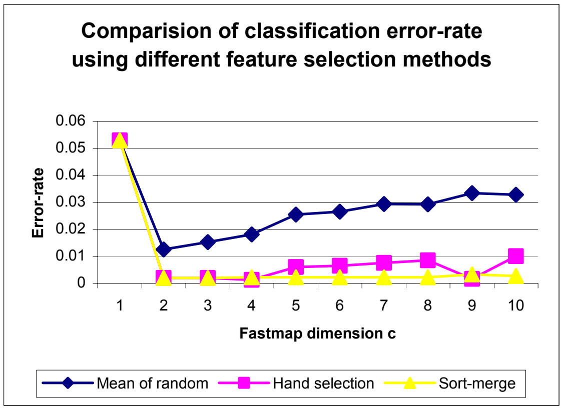

Figure 2. Classification error rate with same feature subset size (r=30) and different Fastmap dimensions c.

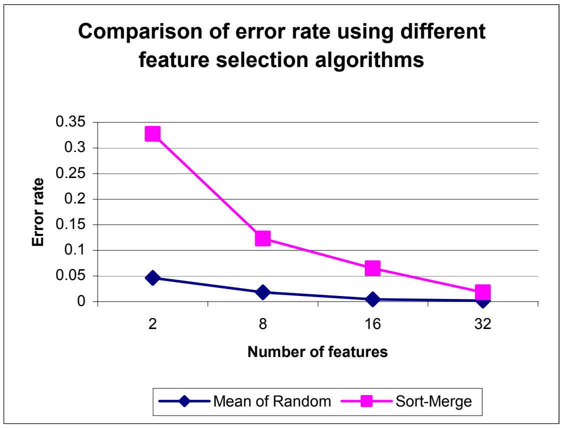

Figure 3. Classification error rate with different feature subset size and same Fastmap dimensions c=4.

Sort-Merge feature selection algorithm is designed for video classification, which is impractical using existing feature selection algorithms. But as a generic wrapper method, it is also applicable to other datasets. In this part, we apply Sort-Merge feature selection algorithm to Ionosphere Dataset of UCI machine learning repository with 351 data of 34 attributes to classify if the radar will return from the ionosphere. First 200 data is chosen for training and feature selection and all these 351 data is used for test. Different from video data, the number of features in this data set is much smaller. It has more training data comparing with the dimension of the data and the number of test data while the target concept is only Boolean value. The classifier is also changed to kNN.

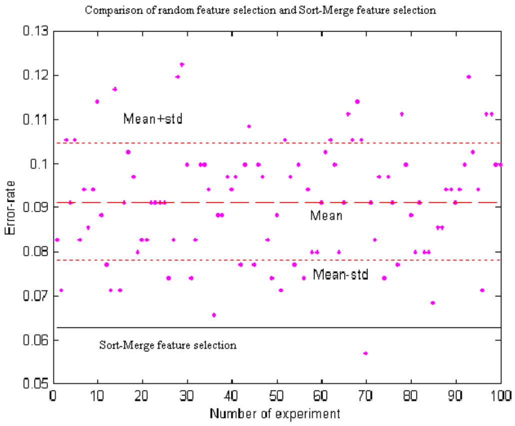

Figure 4. Classification error rate with feature subset size (r=8).

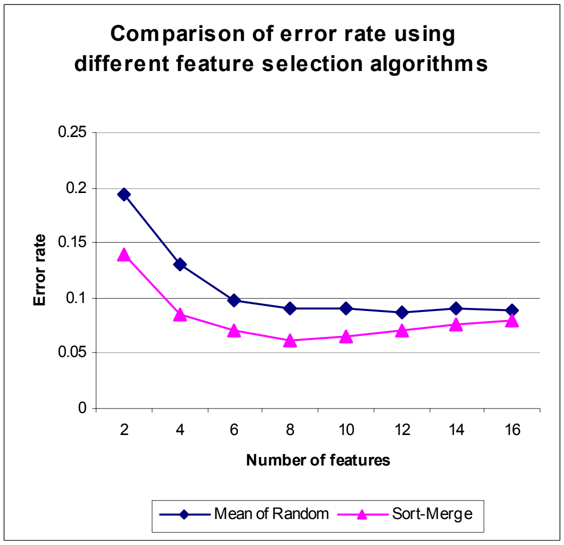

Figure 5. Classification error rate with different feature subset size.

In Figure 4, we fix sample rate r=8 and compare the performance of random feature selection and Sort-Merge feature selection algorithm. Points represent the classification error rate of 100 experiments using random feature selection and the solid line replaces the result using BSMT. Figure 5 shows BSMT has better performance than random feature selection under different sample rate especially when features are sparse.

5. Conclusion

We have presented a novel feature selection method that is well-suited to the difficult domain of video classification. Three novel characteristics that are well-adapted to this large and continuous-valued domain, and which work together in linear time: Fastmap for dimensionality reduction, Mahalanobis distance for classification likelihood, and a Sort-Merge approach to combining relevant and non-redundant feature subsets into more accurate ones. Together, they combine the performance guarantees of wrapper methods with the speed and logical organization of filter methods. The method is shown to be linear in the number of features and in the size of the training set, and it constructs a complete hierarchy of increasingly accurate classifiers. We intend to pursue this work theoretically, in proving some theorems about the limits of its near-optimality, and experimentally, by exercising it on different video genres to derive heuristics about the most appropriate way to set the value of r.

Acknowledgments:

This research was supported by NSF grant EIA-00-71954.

Reference:

[1] Michael S. Lew, Thomas S. Huang, Kam W. Wong, "Learning and Feature Selection in Stereo Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence on Learning in Computer Vision, September, 1994,* pp. 869-881.

[2] Avrim L. Blum and Pat Langley, "Selection of Relevant Features and Examples in Machine learning", *Artificial Intelligence*, 97, pp.245-271.

[3] 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.

[4] 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,* pp.412-420.

[5] Yao Wang, Joern Ostermann, and Ya-Qin Zhang, *Video Processing and Communications*, Prentice Hall, 2002.

[6] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of ACM, April 1991, Vol 34, No. 4,* pp. 46-58.

[7] Richard O. Duda, Peter E. Hart and David G. Stork, *Pattern classification*, Wiley, New York, 2000.

[8] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall International, Englewood Cliffs, NJ, 1980.

[9] 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,* pp 163-174.

[10] Koller, D. and Sahami,M. "Toward optimal feature selection", *Thirteenth International Conference on Machine Learning* , 1996.