Addressing the Problems of Bayesian Network Classification of Video Using High-Dimensional Features

Ankush Mittal and Loong-Fah Cheong

Abstract—Bayesian theory is of great interest in pattern classification. In this paper, we present an approach to aid in the effective application of Bayesian networks in tasks like video classification, where descriptors originate from varied sources and are large in number. In order to extend the application of conventional Bayesian theory to the case of continuous and nonparametric descriptor space, dimension partitioning into attributes by minimizing the discrete Bayes error is proposed. The partitioning output goes to the dimensionality reduction module. A new algorithm for dimensionality reduction for improving the classification accuracy is proposed based on the class pair discriminative capacity of the dimensions. It is also shown how attributes can be weighed automatically in a single-label assignment based on comparing the class pairs. A computationally efficient method to assign multiple labels on the samples is also presented. Comparison with standard classification tools on video data of more than 4,000 segments shows the potential of our approach in pattern classification.

Index Terms—Content-based retrieval, discrete bayes error, partitioning, dimensionality reduction, multiple labels assignment, Bayesian networks.

1 Introduction

Classification is an essential step in pattern recognition tasks and diagnosis systems. Designing accurate classifiers from preclassified data has become a very active research topic in machine learning and data mining.

One of the best known classifiers is the Bayesian network. Bayesian network shows the dependence-independence relations in an understandable form that renders the tasks of decomposition, feature selection, or transformation more principled [3], besides providing a sound inference mechanism. However, Bayesian Network requires a priori knowledge of many probabilities, which are usually estimated based on assumptions about the form of the underlying distributions. Second, there is a significant computational cost required to determine Bayes optimal hypothesis in the general case of multiple-label. This paper presents new approaches to pattern classification based on Bayesian theory with application to the field of Content-Based Retrieval (CBR).

The goal of CBR systems is to retrieve images or video sequences (called, in short, segments) as per the interest of the user (for review on CBR, refer to [4]). The challenges inherent in video classification in CBR systems include, among others, 1) forming close association between the descriptor space and the meaningful classes, 2) performing automatic meaningful dimension evaluation on only a few relevant dimensions (in contrast to the user assigning weights to the dimensions or presuming equal weights for all the dimensions), 3) dealing with the large dimensionality of the descriptor space in an efficient and effective manner, and 4) providing coexistent labels to a multimedia content, which can be later on matched with the user's query.

The novelty of our approach lies in how the conditional probabilities are estimated by discretizing the descriptor space followed by dimensionality reduction. The theoretical foundation for dimension partitioning and dimensionality reduction is presented; it is premised on the reduction of Bayes error of classification. The complexities arising from the characteristic descriptor space are also considered in the knowledge extraction module. The resulting Bayesian Network discerns the data into cliques by learning with the preclassified samples and associates each clique of data points in the multidimensional space to one of the classes. Multiple labels assignment and a meaningful evaluation for the single label assignment are achieved too.

We will be using the following notations for feature, descriptor, dimension, and attribute in this paper (consistent with MPEG-7 [29]). A feature is a perceptual attribute of the video that signifies something to a human observer, i.e., color, texture, shape, motion, etc. A descriptor is a numerical structure that describes a feature, i.e., average color, histogram color, etc. A dimension is one of the dimensions of a multidimensional descriptor. For example, the descriptor "average color" might have three dimensions, one each for average red component, average blue component, and average green component. A dimension when discretized into a number of partitions forms several attributes, which are binary in nature.

- *A. Mittal is with the Department of Computer Science, National University of Singapore, 3 Science Drive 2, Singapore 117576. E-mail: dcsam@nus.edu.sg.*
- *L.-F. Cheong is with the Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576. E-mail: eleclf@nus.edu.sg.*

Manuscript received 22 Sept. 2000; revised 5 Aug. 2001; accepted 10 Oct. 2002.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 112896.

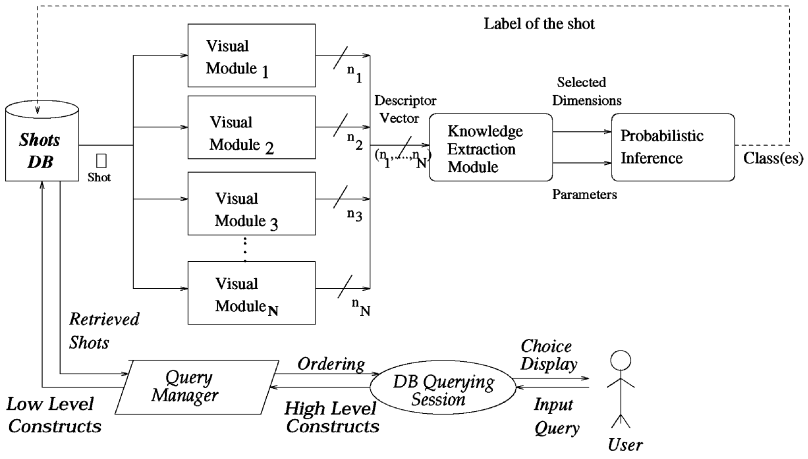

Fig. 1. Proposed content-based retrieval system.

The paper is organized as follows: Section 2 presents a background to CBR system along with the feature extraction process. Section 3 deals with the discretization of dimensions and dimensionality reduction, which are the parts of the knowledge extraction module. Probabilistic modeling and inference are discussed in Section 4. Experiments and comparison with a few classification tools are presented in Section 5. Conclusion and the scope for future work follow in Section 6.

2 Background

2.1 Overview of the CBR System

Fig. 1 depicts an overview of the CBR system. The video data is stored in a database in the form of segments (i.e., a continuous unit of video data representing one single class or meaning). A number of descriptor extractors labeled as Visual $\mathrm{Module}_i, (\mathrm{i} = 1, 2, \ldots, \mathrm{N})$ derive characteristics of a segment, and the output of these descriptor extractors is a N-dimensional vector, some dimensions of which are probably redundant. The knowledge extraction module performs partitioning of the descriptor vectors based on increasing the discrimination among classes. This is followed by a selection of the best few dimensions which can help distinguish classes when considered in pairs. During the learning or modeling phase, this module discerns the probabilities of the classes against each attribute (or subdimension). On the other hand, during the querying phase, it instantiates the Bayesian Network with appropriate parameters.

Mapping of these descriptors with higher-level association is required to furnish high-level classes and is performed by the Bayesian network. Bayesian Network assigns a single label or multiple labels like "interesting," "soccer," "tennis," etc., on the segment as per the design paradigm of the system. These labels are stored in the database along with the segments and through query manager and querying session are matched to the user's choice and retrieved.

2.2 Feature Extraction

Descriptors can be classified as global or local [22]. Global or coarse-grained feature extraction techniques transform the whole image into a functional representation where minute details within the individual portion of the multimedia are ignored. It offers low-computational complexity at the cost of high percentage of false matches. At another level of granularity there exist local descriptors, which exhibit a fine-grained approach in analyzing data into segmented smaller regions. Although working with local descriptors implies increasing the complexity in the feature extraction process and increasing the dimension of the descriptor space, local descriptors are nevertheless employed in our CBR system as they provide more effective characterization of a class.

In order to have complete understanding of the work, a discussion of the descriptors is presented. It can be seen that the descriptors come from highly varied sources and a priori normalization might not work. The high dimensionality associated with these descriptors poses further challenges. The descriptors which we use in our work are given in Table 1. Feature extraction process was done for a domain comprising of diverse video classes was selected as shown in Table 1. The sequences were recorded from TV using VCR and grabbed in MPEG format. The size of the training database was 3,600 sequences each with frame dimension 352×288. The size of the test database was 900 sequences comprising of an equal number of sequences for each class. The results presented throughout this paper are based on these descriptors and the video database. The details of these descriptors are as follows, some of which can be found in MPEG-7 documentation [29]:

1. **Region Shape**. This descriptor can describe any shape consisting of either a single region or a set of regions. It employs a set of ART (Angular Radial Transform) coefficients. Twelve angular and three radial functions are used to give 36 dimensions to this descriptor.
2. **Homogeneous Texture**. Many natural and man-made objects are distinguished by their texture. It uses the energy and energy deviation in a set of frequency channels. The extraction is done by first filtering the image with a bank of orientation and

TABLE 1
Set of Descriptors Used and Set of Video Classes

Descriptor	Dimension
Region shape	36
HomoTexture	62
GoF Color	96
Color Layout	12
Color structure	128
Edge Histogram	80
Average Color	3
Histogram Color	32*3
Motion	16
Intensity Variation	16*3
Edge Density	16
Total	**593**

Class no.	Class
1	Basket
2	Educational
3	MTV (Music)
4	News
5	Soccer
6	Swimming
7	Tennis
8	Table Tennis
9	Volleyball

scale-tuned filters (modeled using Gabor functions) using Gabor filters to give 62 dimensions.

3. **GoF Color**. The group-of-frames (GoF) color descriptor defines a structure required for representing the color descriptors of a collection of video frames by means of the scalable color descriptor. The individual histogram of a video frame is an 32-bin quantized HSV color histogram. The three subdescriptors used are average histogram, median histogram, and intersection histogram.
4. **Color Layout**. This descriptor specifies the spatial distribution of the color of a representative frame. Six, three and three DCT coefficients of the color component Y, Cb, and Cr, respectively, constitute the dimensions of this descriptor.
5. **Color Structure**. It is a color descriptor that captures both the color content and the structure of the content via the use of a structuring element. Color structure descriptor containing 128 bins is computed based on unification of the bins of the 256-bin HMMD color space histogram.
6. **Edge Components Histogram**. This descriptor represents the spatial distribution of four directional edges and one nondirectional edge in each 16 nonoverlapping subimages. For each subimage, a local edge histogram with five bins is generated.
7. **Other descriptors**. In these descriptors, the image is represented in RGB color space. The descriptor **Average color** has three dimensions containing the average value of three components R, G, and B. **Histogram color** is the 32-bins quantized histogram for the three components R, G, and B. Other descriptors are based on a grid layout strategy of MPEG-7. Each frame is split into a set of equally sized 4×4 rectangular regions and each region is described separately. Motion Mot_n^k is the number of pixels moved from frame $n-1$ to the next frame n given as:

$$Mot_n^k = \sum_i \sum_j \langle (f_n(i,j) - f_{n-1}(i,j)) > T_m \rangle, \quad (i,j) \in kth\ subimage, \tag{1}$$

where $f_n(i,j)$ is the pixel value at (i,j) coordinates of nth frame. T_m is a suitably chosen threshold. The predicate $\langle (f_n(i,j) - f_{n-1}(i,j)) > T_m \rangle$ is either 0 or 1. For calculating **intensity variation** IV_n^k, the average of intensity difference from frame $n-1$ and frame n is taken:

$$IV_n^k = \frac{1}{\text{size(subimage)}} \sum_i \sum_j f_n(i,j) - f_{n-1}(i,j), \quad (i,j) \in kth\ subimage. \tag{2}$$

Edge density is calculated by finding the edges in a frame. Then, a percentage count of pixels with edge intensity greater than a threshold is taken for each subimage.

Let us present the difficulty in estimating the probability density functions (PDFs) and in the subsequent classification with an example of descriptor values for a class. Fig. 2 shows a typical histogram plot for a descriptor. Unlike the

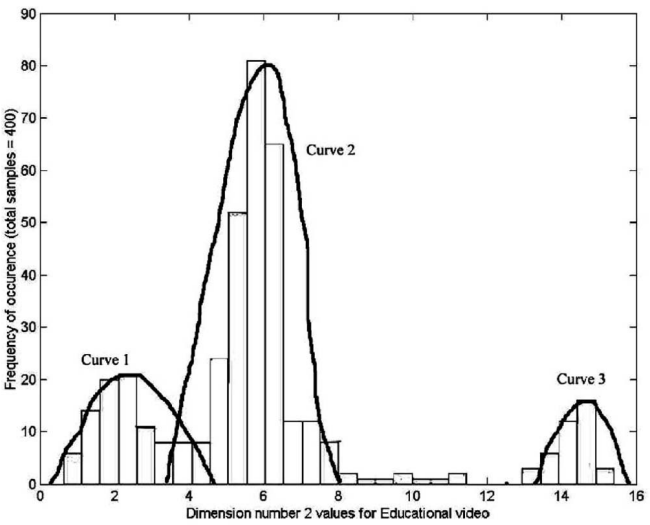

Fig. 2. Histogram plot of frequency of occurrence against the values of a descriptor for an Educational video. On X-axis are histogram bins. Each bin has data points within a range of values. On Y-axis is the number of points in the bin with the total number of samples being 400. Note that Curve 1, Curve 2, and Curve 3 belong to the same class.

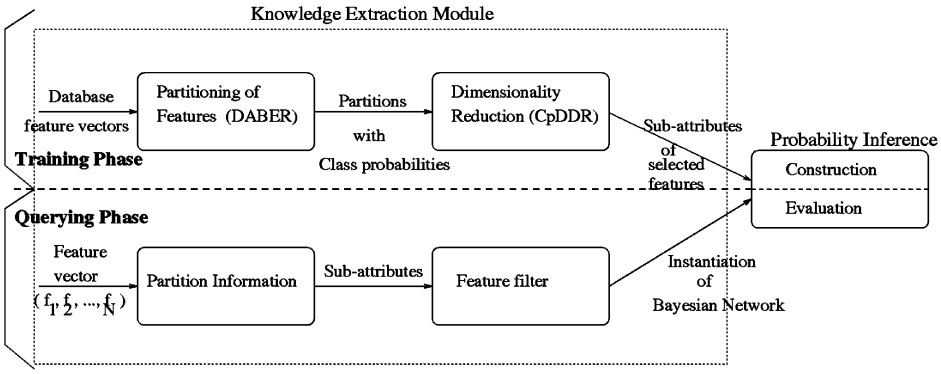

Fig. 3. Knowledge extraction module.

assumed unimodal Gaussian distribution, the distribution could be multimodal. Estimating the PDFs in such situations is an extremely difficult problem. On the other hand, techniques that are based on dividing the descriptor space into small regions and estimating the probability could be effective.

3 Knowledge Extraction

In this section, details of discretization of descriptors and dimension selection methods are presented. Fig. 3 depicts the functionality of the knowledge extraction module during the two phases, the training phase and the querying phase. During the training phase, the descriptor vectors corresponding to all the classes are passed to the discretization algorithm (DABER) and the resulting partitions along with the corresponding probabilities of the classes form the input to the dimensionality reduction algorithm (CpDDR). By employing discretized partitions, CpDDR can make an effective decision in selecting the dimension set $\mathcal{F}$, which can distinguish one class against another. During the querying phase in which the instantiation of the Bayesian network and the subsequent evaluation have to take place, the partitions (i.e., attributes) corresponding to each of the selected dimensions $s_f \in \mathcal{F}$ are passed to the inference module.

3.1 Discretization of Dimension

Dealing with multimodal continuous variables constituting a multidimensional descriptor space without making assumptions about the underlying probability is a complicated task. When descriptors have continuous values, a standard approach (like in [31]) is to compute the conditional probability density by assuming that, within each class, the values for a descriptor are normally distributed about some mean. Mean and standard deviation of a class for a descriptor are evaluated using a common statistical approach. It was shown in Pazzani [30] that Gaussian assumption of numeric data may lead to poor performance in many practical systems like electrical faults. He suggested the discretization of the variables into a small fixed number of partitions for each descriptor.

It is not optimal to have fixed partitions all of equal sizes as some partitions become densely populated leading to poor discrimination (see several examples and analysis in [8]). Thus, we propose a statistical approach for finding the boundary points of a variable number of partitions. The discretization algorithm gives optimum number of partitions and with good discriminability between the classes based on the Discretely Approximated Bayes Error Rate (DABER). The derivation of DABER is shown below followed by the description of the algorithm in detail.

Let π_i denote the a priori class probability of video class i, $1 \leq i \leq |V|$, and $p(\mathbf{s}\,|\,v_i)$ be the conditional probability density of $\mathbf{s}$, the descriptor vector for a video segment, given that it belongs to class i. $|V|$ is the number of video classes. $p(\mathbf{s})$, the probability distribution function of $\mathbf{s}$, is given by: $\sum_{i=1}^{|V|} p(\mathbf{s} \mid v_i)\pi_i$. The Bayes error which is associated with Bayes classifier is given by [18]:

$$E_{\mathbf{s}} = \int_R [\ 1 - \max_i p(v_i \mid \mathbf{s})\ ]\ \ p(\mathbf{s}) d\mathbf{s}, \tag{3}$$

where R is the descriptor space and $p(v_i \mid \mathbf{s})$ is the a posteriori probability of class v_i, $i = 1, 2, 3, \ldots, |V|$.

Evaluating the Bayes error E_s might entail the complexity of evaluating the multidimensional integral of unknown multivariate functions and therefore, in practice the Bayes error can be computed directly only for a limited number of problems. Approximations and bounds on the Bayes error are instead commonly calculated. In [37], the outputs of various classifiers are used to calculate the upper and the lower bounds on the Bayes error rate. Similarly, an approximation of the Bayes error was used by Kohn et al. [23] based on a nonhomogeneous, nonlinearly separable terminal multidimensional box as a class discriminability measure after the completion of the partitioning process.

In our approach, the computation of the Bayes error rate and the partitioning process are made interdependent to each other and the partitioning process iterates on each dimension. Let there be $N_p^j(t)$ partitions of dimension $s_j \in \mathbf{s}$ at the particular time instant t of the process. Consider any arbitrary partition l resulting from the discretization process. This partition might contain data points from more than one class. Let $m(l)$ be the total number of data points of all classes in the partition l and $m(i \mid l)$ be the total data points from class i in the partition l.

The Bayes error rate corresponding to dimension s_j alone is given by

$$E_{s_j} = \int_{R_j} [\ 1 - \max_i p(v_i \mid s_j)\ ]\ \ p(s_j) ds_j, \tag{4}$$

where $p(v_i \mid s_j)$ is the a posteriori probability of class v_i and can be approximated to $m(i \mid l)/m(l)$ for lth partition of dimension s_j. Using $ds_j \cong \triangle s_j$, the probability distribution of dimension can be written as

$$p(s_j) \cong \frac{\mathrm{m(l)}}{\left[\sum_{\mathrm{k=1}}^{\mathrm{N_p^j(t)}} \mathrm{m(k)}\right] \times \triangle \mathrm{s_j}}. \quad (5)$$

This assumes that, in a partition, the feature points are uniformly distributed. In general, we could try to estimate the probability distribution within each partition using the mixture of Gaussian distributions. However, it is complicated and computationally expensive task. Of course, if our classification step considers each of these partitions as a binary attribute as we do, the two approaches would give almost identical final performance. E_{s_j} can be, therefore, written as:

$$\begin{aligned} E_{s_j} &\cong \sum_{l=1}^{N_p^j(t)} [1 - \max_i \ m(i \mid l)/m(l)] \times \frac{\mathrm{m(l)}}{[\sum_{\mathrm{k=1}}^{\mathrm{N_p^j(t)}} \mathrm{m(k)}] \times \triangle \mathrm{s_j}} \triangle s_j \\ &= \frac{1}{\sum_{\mathrm{k=1}}^{\mathrm{N_p^j(t)}} \mathrm{m(k)}} \sum_{l=1}^{N_p^j(t)} [m(l) - \max_i \ m(i \mid l)]. \end{aligned} \quad (6)$$

This equation implies that the Bayes error would be less if the partition contains most of the sample points from one of the classes. The discrete approximated Bayes error $E_{s_j}^l$ for partition l alone can be written as:

$$E_{s_j}^l \ \cong \ \frac{1}{\sum_{\mathrm{k=1}}^{\mathrm{N_p^j(t)}} \mathrm{m(k)}} [m(l) - \max_i \ m(i \mid l)] \ = \ DABER_j^l. \quad (7)$$

The DABER algorithm is delineated below in detail.

DABER Algorithm

Formation Phase :

```
while (some-dimensions-left) {
  choose one of the unprocessed dimension s_j;
  sort_on_dimension(data,s_j);     /* Sort data on s_j */
  /* start forming a new partition l */
  while (Upper_limit(l) < maximum_value(s_j,data) {
    if (DABER_j^l > θ_err)     /* discriminability is poor */
      if (density(l) > ρ )
      /* Only high density partition should be divided */
      {Pt_opt = Find-optimum-DABER-point(l);
      divide_the_partition(l, Pt_opt);
      /* Divide at optimum point */
      Set_Lower_limit(l) = Pt_opt; }
      /* Start growing from optimal point */
    Upper_limit(l) = Upper_limit(l) + step_increment; }
  If (number_of_partitions(s_j) > τ) discard_dimension(s_j);
  /* Irrelevant dimension */
  }    /* Take another dimension */
```

Merging Phase : /* See if two consecutive partitions $l1$ and $l2$ can be merged */

```
if combined_DABER(l1, l2) < ϑ    /*  ϑ > θ_err */
  merge_partitions(l1, l2);
```

The DABER algorithm discretizes the data on each of the dimension s_j separately in two phases: formation phase and merging phase. In formation phase, first the data is sorted on s_j. An expanding window that starts from minimum value of s_j is then considered. The window expands in the direction of the increasing value of dimension s_j in steps of $step\_increment$. A partition is formed when the DABER value falls below a threshold θ_{err} and the partition density is at least greater than ρ threshold. An optimal point for the partition is found iteratively on all the dimension value in the partition by minimizing the sum of the two partitions that would result from choosing that point. The partition is formed with the lower limit set to the least value of the expanding window and the upper value set to the optimal point. The expanding window then considers all points above optimum value for possibility of new partition and the process repeats till the maximum value points are considered. Large number of partitions could serve as an indication that there is a large overlap between the probability distributions of classes and, therefore, the dimension is not useful. The merging phase serves to refine the partitions in second pass. Adjacent partitions are combined with the resulting partition has DABER value below threshold ϑ. Let $|s|$ be the number of features and t be the number of training samples, then it can be seen that DABER algorithm is $O(|s|\,[t\log(t) + t^2])$, i.e., $O(|s|\,t^2)$.

The DABER threshold θ_{err} and the density threshold ρ control the number of discretized dimensions (i.e., attributes). On increasing the value of θ_{err} or alternatively on decreasing ρ, the number of partitions can be increased. The trade off in performance by increasing the number of partitions is the increase in size of Bayesian CBR network. It has been argued in [19] that the structure of AI tools like Neural networks and support vector machines needs to be altered with changes in the dimensionality of the descriptor space, i.e., the size of the input vector, or with changes in the size of the training data.

In contrast, changing the two parameters of the CBR system, θ_{err} or ρ, do not affect the performance of our CBR system as long as θ and ρ are fixed such that the number of partitions is not too small leading to poor discrimination. This would be demonstrated later by experimental results on varying training size and varying the dimensionality. Since, each dimension is treated independently of the other dimensions in the discretization algorithm, the number of dimensions does not affect the output of an individual dimension as well as the processing time per dimension. Varying the training size could alter the performance significantly if the distribution characteristics of classes are altered. However, assuming the classes (which is the case with domain like multimedia) have structure in their elements, a reasonable set of training samples can represent the data.

DABER algorithm constructs a set of hyperrectangular parallelepiped partitions such that the faces are perpendicular to the dimension axes and located at the maxima and minima of the samples in the partition. These hyperrectangles are disjoint in relation to each other, except for their boundaries. For example, if there are only two dimensions, a partition can be defined as rectangle $\{(x_{min}, y_{min}),$

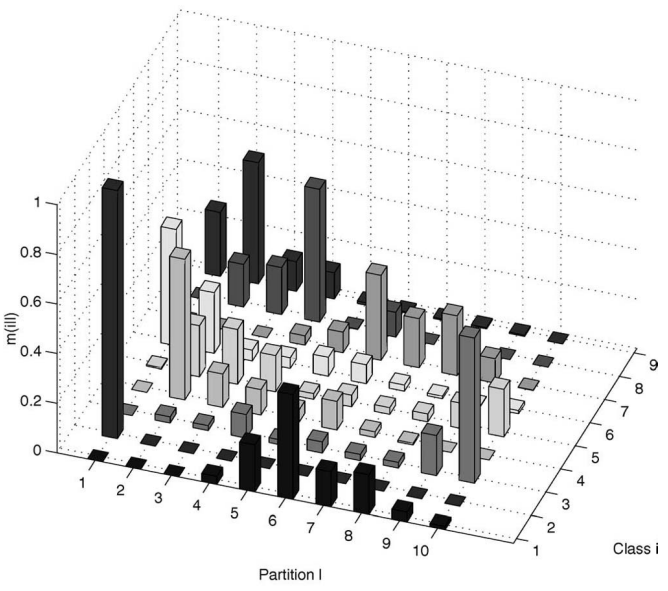

Fig. 4. The probability distribution $m(i|l)$ of class i on a single dimension of histogram descriptor partition l using DABER algorithm. For example, for partition number 10, conditional probability of class 3 is around 0.6 and of class 4 is 0.2. The description of the descriptor and the set of classes is given in Section 2.2.

$(x_{max}, y_{max})\}$, where (x_{min}, y_{min}) and (x_{max}, y_{max}) are the coordinates of the diagonal ends of the rectangle. The value x_{min} and x_{max} are the minimum and the maximum values, respectively, for dimension $\mathbf{x}$ in the partitions. Since DABER algorithm considers each dimension separately, it can work even in tasks with a large numbers of dimensions or with correlated data. Correlation of data is handled appropriately in our dimensionality reduction algorithm presented in the next section (Section 3.2).

Corresponding to the above algorithm, the probability distribution of the classes over the partitions of a chosen dimension of histogram descriptor is depicted in Fig. 4. The X-axis is marked by the partitions 1 to 10 while the classes (1 to 9) are marked on the Y-axis. Thus, a class distribution along the partitions can be determined by viewing parallel to the X-axis from the mark of the class.

Many noteworthy observations can be made from the figure. A significant number of partitions have only a few of the classes as the most dominant classes. For example, partition 1 has class 2 as the distinctly prominent class and similarly partition 10 has class 3 as the prominent class. Still there is ambiguity in making clear decision on the basis of one dimension. However, the ambiguity can be resolved with good accuracy when more than one dimensions of the descriptor space are considered because each meaningful dimension would add cues for characterization. The second observation from Fig. 4 is that the relevance of a dimension for each class is appropriately calculated based on distribution of the class over the descriptor space. For a class like 1, which has wide distribution, this particular dimension is not so relevant as this is meaningful for class 2 or class 8. Thus, for the purpose of distinguishing class 2 points from that of class 8, the dimension could be effectively used.

DABER algorithm strategy differs from several other well known algorithms such as by Pfahringer [33] and by Liu and Setiono [25]. Pfahringer partitions the variable value to a large number of partitions in a binary tree and uses the MDL metric in a best first search to determine best partitions. Liu and Setiono partition the data in large number of intervals and merge them based on χ^2 statistical test. DABER algorithm takes a more fundamental approach in the sense that the initial formation of partitions themselves is a meaningful exercise for increasing the classification performance.

3.2 Dimensionality Reduction

Multimedia classification is representative of a domain of tasks involving high dimensionality of the descriptor space and a large dissimilarity between the descriptors in their range and distribution. Dimensionality reduction can eliminate some irrelevant and/or redundant dimensions of the descriptors. By using feature selection, classification algorithms can in general improve their predictive accuracy (as in [1]), shorten the learning period [25], and result in saving in the memory requirements and the computation time.

There are two techniques which are commonly found for dimensionality reduction: one is to select a limited set of "features" (here, "feature" is used as is generally used in pattern recognition task and is equivalent to dimension in our notation) out of the total set ([12]) and the other is to extract a smaller set of "features" as linear or nonlinear functions of the original set of "features" using Principal Component Analysis (PCA) [39] or discriminant analysis [17]. The present approach is based on the former technique i.e., choosing few "features" from the original set because of two reasons: First, the approach is more appropriate for meaningful "feature" evaluation executed in the inference module discussed later on and, second, in multimedia classification, the queries could be partially specified, which makes the task of obtaining transformed "features" using PCA or discriminant analysis a formidable exercise. Some of the well known "features" selection algorithms are not apposite in their application in the domain of multimedia classification. FOCUS [2] is intractable in data mining applications with thousands or even hundreds of "features" because it selects the minimal subset of "features" by exhaustively examining all the subsets of "features." PRESET [28] works only in a noise-free domain. We devise a class-pair distinctive "feature" selection which is efficient in computation even for a large data and can work on noisy data as well.

3.2.1 Relationship of Dimensionality Reduction with Bayes Error

Let us revisit the basics of "feature selection" process that is specifically employed to improve the classification accuracy. We would consider the "best" classifier (from the theoretical standpoint), the Bayesian classifier. If the cost of all types of correct classification is zero and the cost of all types of incorrect classification is one, the optimal Bayes decision rule assigns the sample to the class with the highest a posteriori probability. Thus, the Bayes risk associated with a given dimension set $\mathbf{s}$ reduces to the probability of error, $E_{\mathbf{s}}$, which was defined in (3). In order to minimize the classifier error rate, the most appealing function to evaluate

TABLE 2
Classification Problem to Illustrate the Working of E_s Measure

Classes \ Attribute (0/1)	s_1	s_2	s_3	s_4	s_5
v_1	0.75	0.90	0.92	0.40	0.10
v_2	0.10	0.45	0.52	0.40	0.90
v_3	0.80	0.45	0.60	0.40	0.75
v_4	0.85	0.01	0.05	0.40	0.80

the potency of a dimension to differentiate between the classes is the E_s function (see chapter 5 of [12]). A brute force method to best select the dimensions using E_s function would be to select a combination with minimum classification error among $2^n - 1$ combinations of all dimensions. Therefore, the E_s function has not been used in practical applications because of its computational complexity; instead, other measures for class separability were sought such as Bhattacharya distance, Matusita distance, Patrick-Fisher distance, and Shannon measure [5]. Note that all these measures are related to the E_s function as they provide an upper bound to the error probability. However, they are not ideal as they are not directly derived from the expression of classification error. Besides, the reliability of a separability measure depends on how tightly it bounds the error probability. Thus, we would like to employ a measure directly derived from the E_s. Let us illustrate the strength and weakness of E_s function as a criteria for dimensionality reduction.

Consider a classification problem with four video classes $(v_1, \ldots, v_4)$ and five binary dimensions (or, in this case, attributes) $(\mathbf{s_1}, \ldots, \mathbf{s_5})$ as presented in Table 2. The entries in the table represent the respective conditional probabilities for the presence of the attribute given the classes. If the prior probability of all the classes is equal to 0.25, the E_s values for all the attributes are given in Table 3 using (3) for discrete case (i.e., attribute value is 0/1). An attribute $\mathbf{s_i}$ is preferred to $\mathbf{s_j}$ if $E_{s_i} < E_{s_j}$. Thus, the attribute $\mathbf{s_2}$ is preferred over all other attributes. The a posteriori probabilities $P(v_k|s_i)$ is calculated as

$$\frac{\pi_k p(s_i|v_k)}{\sum_1^{|V|} \pi_1 p(s_i|v_1)},$$

where π_k is the a priori class probability of video class k and $p(s_i|v_k)$ is the conditional probability density of attribute s_i. Suppose, if $\mathbf{s_2}$ is selected and it becomes active (i.e., $s_2 = 1$), then the approximate a posteriori probabilities for the four classes are: $P(v_1 \mid \mathbf{s_2}) = 0.5$, $P(v_2 \mid \mathbf{s_2}) = 0.25$, $P(v_3 \mid \mathbf{s_2}) = 0.25$, and $P(v_4 \mid \mathbf{s_2}) = \mathbf{0.005}$. In other words, the likelihood of class v_1 is doubled, while class v_4 is virtually eliminated. On the other hand, if $\mathbf{s_4}$ is selected, the posterior probabilities of all the classes remain the same as the priors regardless of the results observed for $\mathbf{s_4}$. Thus, the E_s rule is a good measure of how much a dimension contribute to differentiating among the classes.

TABLE 3
The Ordering of the Attributes Using E_s Function

Feature	s_2	s_3	s_5	s_1	s_4
E_{s_j}	0.52	0.53	0.55	0.56	0.75

Though the individual merit of a dimension can be assessed by the E_s rule, the selection of a group of dimensions for reducing the overall error is not straightforward. For example, if $\mathbf{s_2}$ and $\mathbf{s_3}$ are both selected, they both have almost the same discrimination characteristics (i.e., the ability to distinguish class v_1 against all other classes especially class v_4). It can be seen from this example that a good set of dimensions should discriminate every class from other classes or in other words, they should reduce the overlapping between the class probabilities. We discuss in the following section our algorithm which involves rating the dimensions based on the E_s rule with the final selection of good dimensions decided by how they differentiate various different class pairs.

3.2.2 The Algorithm

In the two-class case, the error rate in (3) can be expressed as:

$$E_s = \frac{1}{2}\left[1 - \int_R \mid p(\mathbf{s} \mid v_1)\pi_1 - p(\mathbf{s} \mid v_2)\pi_2 \mid d\mathbf{s}\right], \quad (8)$$

where π_i is the prior probability of class v_i. The integral in the above equation is known as the Kolmogorov distance [36] which is theoretically a sound distance measure as compared to the other measures. Our distance measure is a modification of the Kolmogorov distance for the discrete case. To map the problem of multiple classes into two classes, we consider all class pairs (v_I, v_j), $i \neq j$ at a time.

Our dimensionality reduction algorithm is based on choosing α dimensions (in our case $\alpha = 2$) for each class pair (v_I, v_j) such that these α dimensions possess the highest discriminability measure of class v_i against class v_j. Since the probability inference module is based on comparing the a posteriori probability of one class with that of another class, this would result in optimally correct classification.

To derive discriminatory measure, we consider Kolgomorov distance KD^{s_f} for a dimension s_f for discrete case (from (8) considering two-class (v_i, v_j) case):

$$KD^{s_f} = \sum_{l=1}^{N_p^f} \mid p(s_f^l \mid v_i)\pi_i - p(s_f^l \mid v_j)\pi_j \mid, \quad (9)$$

where N_p^f is the number of partitions of the dimension s_f. The conditional probability density $p(s_f^l \mid v_i)$ for a partition l of a feature s_f is the probability of a sample belonging to class v_i in partition l and is given by

$$p(s_f^l \mid v_i) = \frac{\mathrm{m(i \mid l)}}{\sum_{\mathrm{l=1}}^{\mathrm{N_p^f}} \mathrm{m(i \mid l)}}. \quad (10)$$

The discriminatory capacity $|P_{ij}^{s_f}|$ of a dimension s_f for the class pair (v_i, v_j) is simply KD^{s_f}. $|P_{ij}^{s_f}|$ is the absolute difference between the probabilities of class v_i and class v_j

taken over all the partitions of a dimension $s_f \in \mathbf{s}$. If the prior probabilities of all the classes are equal (i.e., there are equal training samples for the video classes), $|P_{ij}^{s_f}|$ can be written as:

$$|P_{ij}^{s_f}| = \sum_{l=1}^{N_p^f} \frac{|\, p(s_f^l \mid v_i) - p(s_f^l \mid v_j)\,|}{2}. \tag{11}$$

Note that the discriminatory capacity $|P_{ij}^{s_f}|$ is a relative measure that lies between 0 and 1. A high value of $|P_{ij}^{s_f}|$ signifies that dimension s_f can distinguish well class v_i samples and class v_j samples.

Next, the Class-Pair Discrimination Measure ($CPDM_{ij}$) for each class pair (v_i, v_j) is evaluated by summing $|P_{ij}^f|$ over each dimension f, where $\mathrm{f} \in \mathbf{s}$, as shown below:

$$CPDM_{ij} = \frac{1}{|\,\mathbf{s}\,|} \sum_{f=1}^{|\mathbf{s}|} |P_{ij}^f|\,. \tag{12}$$

A high value of $CPDM_{ij}$ indicates the relative ease of classifying the pair (v_i, v_j). Such a class pair is called a "highly separable" class pair. All class pairs (v_i, v_j), $1 \le i, j \le |V|$, $i \ne j$, are sorted according to the magnitude of $CPDM_{ij}$. The class pair (v_i, v_j) with minimum CPDM and those not yet covered are first examined to ensure that the dimensions which distinguish "less-separable" class pair are first chosen. The α dimensions which have maximum discriminatory capacity $|P_{ij}^{s_f}|$ in the class pair (v_i, v_j) are selected among the best $\gamma\,(\ge \alpha)$ dimensions. Among γ best dimensions, if some dimensions are already included in the selected set, they are included in the α count. This optimizes the selection algorithm for choosing the minimum number of dimensions. Other class pairs are similarly covered.

Notice that even class separability measures like Bhattacharya, Mitusita, etc., are for two-classes and an extension to multiple class requires separate formulation. A common strategy [12] is to calculate the overall discrimination power $D(\mathbf{s})$ of a dimension s by summing the distance $d_{ij}(\mathbf{s})$ between the two classes as follows:

$$D(\mathbf{s}) = \sum_i \sum_j \pi_i \pi_j d_{ij}(\mathbf{s}). \tag{13}$$

However, the major disadvantage of this approach is that one large value of $d_{ij}(\mathbf{s})$ may dominate the value for $D(\mathbf{s})$ and impose a ranking that reflects only the distance between the most separable class pair. The details of the Class-pair Discriminative Dimensionality Reduction (CpDDR) algorithm based on using the CPDM are as follows:

CpDDR Algorithm

```
while (any-class_pair-unprocessed) {
  choose_class_pair(v_i, v_j) ; /* i ≠ j */
  CPDM_ij = calculate_CPDM(v_i, v_j)
  /* Calculate using Eq. 12 */
  mark_class-pair_processed(v_i, v_j) }
/* Sort CPDM in nondescending order */
CPDM_sorted = sort_order(CPDM);
  /* CPDM_sorted is an array */
while (all_pairs_covered(CPDM_sorted) ==FALSE)
  Take-the-next-uncovered-pair-in-order(v_i, v_j) /* i ≠ j */
  { /* Sort the dimensions index(f) on |P_ij^sf| for pair (v_i, v_j)
    in descending order */
    SORTED_F_ARRAY = sort_dimension_index(i,j);
    /* Choose the α dimensions with highest
    discriminability value |P_ij^sf| */
    chosen_SET = chosen_SET ∪
    choose_α_dimension(SORTED_F_ARRAY)
  }    /* Take another pair */
OUTPUT(chosen_SET)
/* chosen_SET has the selected dimensions */
```

The above algorithm is based on selecting the dimensions that are most effective in distinguishing one class from another. The algorithm first computes the $CPDM_{ij}$ values for all class pairs. In each step, the least separable class pair (v_I, v_j) is chosen among the class pairs that are not yet considered. Next, all the dimensions are sorted based on their discriminatory capacity. The next step is to select α dimensions with highest discriminatory capacity (and include them in output set of features). These α dimensions best discriminate the least separable class pair (v_i, v_j).

In our experiments with nine classes and 593 dimensions, the above algorithm selected 59 dimensions. For experimentation, when the number of dimensions were reduced to 59, the algorithm selected 29 dimensions. One key observation from the experiment was that a few class pairs had some common selected dimensions, which makes dimensionality reduction process more meaningful.

Note that the CpDDR algorithm is based on the assumptions that at least a few dimensions are relevant to each class. This is because a simple comparison was performed among the dimensions for selection. For making the algorithm more generic for other purposes than pattern classification, thresholding can be done on the discriminability measure of a dimension.

3.2.3 Comparison with Two Common Approaches

Our approach to dimensionality reduction is superior to the standard statistical methods which have been used in diverse applications involving regression or classification tasks like classification of wood defects [14], or numeral recognition [20]. These methods use measures like intraclass variation, interclass variation, or correlation, etc., to differentiate how well the dimensions differentiate between the classes. Generally, the dimensions are rejected if their intraclass variation is above a given threshold or their interclass variation is below a given threshold. Additionally, correlated dimensions can be eliminated from the selection set. The normalized intraclass variation of dimension s_f is given by:

$$IACV(f) = \sum_{i=1}^{|V|} \sigma_f^2(i), \tag{14}$$

where $|V|$ is the number of classes and $\sigma_f^2(i)$ is the variance. The total interclass variation of dimension s_f is given by:

$$IRCV(f) = \sum_{i=1}^{|V|} \sum_{j=1, j\ne i}^{|V|} \frac{|\,\tilde{\mu}_{\mathrm{f}}(\mathrm{i}) - \tilde{\mu}_{\mathrm{f}}(\mathrm{j})\,|}{\sqrt{\sigma_{\mathrm{f}}^2(\mathrm{i}) + \sigma_{\mathrm{f}}^2(\mathrm{j})}}. \tag{15}$$

Item	Value
Mean IACV of all features	0.21
Mean IACV of selected features (CpDDR)	0.26
Mean IACV of features selected by IACV algorithm	0.003
Number of features common in CpDDR and IACV algorithm	7 (12 %)
Number of CpDDR selected features which have IACV more than mean	43 (73 %)

Fig. 5. CpDDR versus IACV algorithm. The total number of dimensions was 593. The threshold for IACV algorithm was kept such that it selected the same number of dimensions as CpDDR.

Item	Value
Mean IRCV of all features	42.42
Mean IRCV of selected features (CpDDR)	68.66
Mean IRCV of features selected by IRCV algorithm	83.6
Number of features common in CpDDR and IRCV algorithm	31 (53%)
Number of CpDDR selected features which have IRCV less than mean	7 (11%)

Fig. 6. CpDDR versus IRCV algorithm.

Fig. 5 makes the comparison between the IACV algorithm with $\omega = 59$ (the number of dimensions selected by CpDDR) and CpDDR. It is interesting to note that the mean IACV of dimensions selected by CpDDR, i.e., 0.26 is even higher than the mean IACV of all dimensions, i.e., 0.21. Only 12 percent of the dimensions are common in the two algorithms. The difference in the functioning of the two algorithms can be explained as follows: The IACV algorithm is based on selecting the dimensions which have the least scatter of data *around one cluster* within each class while CpDDR is based on selecting dimensions which have minimum overlap within the class pairs and, thus, highest discriminability. The distribution of data (Fig. 2) shows that although the data is distributed in the clusters for classes, each class might have several of such cliques. Second, IACV makes no reference to the distribution of data of another class.

The IRCV algorithm would choose ω dimensions with maximum IRCV values. Fig. 6 presents the comparison between CpDDR and IRCV. We found that the mean of CpDDR selected dimensions, i.e., 68.66 is significantly higher than the mean IRCV of all dimensions, i.e., 42.42. The number of common dimensions selected by both algorithms is 53 percent. This shows that the dimensions selected by CpDDR have high IRCV values, in general.

However, there are two significant advantages of the CpDDR over the IRCV algorithm: 1) The set of dimensions selected by CpDDR necessarily has dimensions to distinguish every class-pair, which is not the case in IRCV. The IRCV algorithm may choose dimensions, which may distinguish only a few class pairs. 2) IRCV is based on the "one cluster for one class" assumption. $\tilde{\mu}_f(i)$ used in (15) would generally fail to give accurate representation for the mean value of class data with arbitrary multimodal distribution. The CpDDR, on the other hand, is more generic for it can work with multimodal distribution without making a priori assumption about the data.

Fisher's Discriminant Analysis (FDA) approach is based on both maximizing the between-class scatter and minimizing the within-class scatter of the selected features. However, FDA does not have a direct relationship to the probability of error for the Bayes classifier, which is the optimum measure of feature effectiveness. Second, in case of higher dimension problems such as in CBR systems, use of FDA is not suitable as regardless of the dimension of the original patterns, the FDA transforms a pattern vector onto a feature vector, whose dimension can be at most $|V| - 1$, where $|V|$ is the number of classes [17]. Besides, within-class scatter and between-class scatter are difficult to compute effectively when the probability distribution is multimodal (see Fig. 2) and the class patterns are distributed in cliques.

4 The Probabilistic Inference

In this section, the application of Bayesian network for achieving coupling between the low-level dimensions and the high-level classes is presented. The structure of the network for CBR is first presented followed by the inference mechanisms to assign either multiple-labels or single-label to a video segment.

4.1 Bayesian Network

The practical difficulty in applying Bayesian methods is that they typically require initial knowledge of many probabilities. A second practical difficulty is the significant computational cost required to determine Bayes optimal hypothesis in the general case (multiple-labels).

Let the set of Video classes be denoted by V, the set of dimensions by $\mathbf{s}$, and the relation $\mathrm{C} \subseteq \mathrm{V} \times \mathbf{s}$ represent the pairwise causal associations between Video classes and dimension values. Video class v_i giving rise to dimension value s_f corresponds to a link in the graph. A subset of $\mathbf{s}$, denoted $\mathbf{s}^+$, represents the set of all dimension values that are present, while $\mathbf{s}^- = \mathbf{s} - \mathbf{s}^+$ represents the set of dimension values assumed to be absent. The computation of $\mathbf{s}^+$ and $\mathbf{s}^-$ from the possibly continuous values of the dimensions is discussed later in this section. Each causal link between a video class v_i and a dimension value s_f is associated with a number $c_{if} \in (0, 1]$, called causal strength from v_i to s_f, representing how frequently v_i causes s_f. In other words, $c_{if} = P(v_i \text{ causes } s_f \mid v_i)$. If there is no causal link between v_i and s_f, then c_{if} is assumed to be zero. The evaluation of c_{if} is done in the learning module, which also constructs dynamically the structure of the Network as discussed.

Let dimension s_f be partitioned into N_p^f subdimensions (or attributes), s_{f_k} for $1 < \mathbf{k} < N_p^f$. After the execution of the DABER discretizing algorithm, each $\text{partition}(s_{f_k})$ of a dimension can be considered as a Boolean variable (see

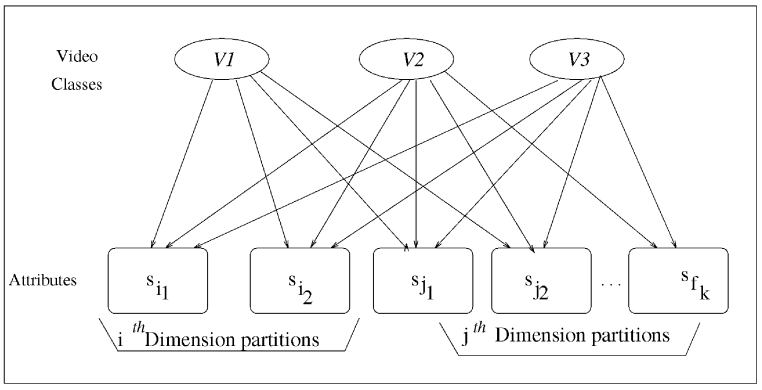

Fig. 7. Schematic CBR Belief Network. Each of the partition of a dimension becomes an independent binary attribute (active 1 or passive 0) which can be caused by one or more classes.

Fig. 7). At a particular instance, one and only one of the attributes, denoted by $s^{\tau}_{f_k}$, will have the status of being present (i.e., True). The link probabilities are also redefined as c_{if_k} denoting the causal strength from v_i to attribute s_{f_k}. Similarly, after partitioning, $\mathbf{s}^{+}{}_k$ refers to the set of all $s^{\tau}_{f_k}$ present and $\mathbf{s}^{-}{}_k$ refers to the set of other attributes (i.e., those which are not present).

4.2 Noisy-OR Model of the Network

The effect of multiple classes on a common attribute is modeled by a Noisy-OR gate. This assumes that the mechanisms, whereby different classes can result in a given attribute assignment, operate independently. Noisy-OR gate is suitable when an attribute can be present only by the class set, V. However, in real life, there is always a possibility that an attribute may be present even when none of the classes in V is present. Therefore, we permit leakage in the Noisy-OR gate which assigns null hypothesis (H_ϕ) with a finite value not necessarily equal to zero. A hypothesis H consists of a set of classes (none, one, or more than one). As discussed below, it would become clear that null hypothesis (H_ϕ) is assigned when no class or a set of class has sufficiently high probability. If π_i is prior probability of occurrence of video class v_i, the probability of null hypothesis $P(H_\phi)$ is $\prod_{\forall v_i \in V}[1-\pi_i]$. The prior probability $P(H)$ of a hypothesis H can be decomposed as follows:

$$P(H) = \prod_{\forall v_i \in H} \pi_i \times \prod_{\forall v_i \notin H} [1-\pi_i]. \tag{16}$$

This can be expressed in terms of prior probability for H_ϕ as follows:

$$P(H) = \prod_{\forall v_i \in H} \frac{\pi_{\mathrm{i}}}{1-\pi_{\mathrm{i}}} \times \prod_{\forall v_i \in V} [1-\pi_i] \;=\; P(H_\phi) \prod_{\forall v_i \in H} \frac{\pi_i}{1-\pi_{\mathrm{i}}}. \tag{17}$$

Leak probability corresponding to null hypothesis is defined as $c_{\phi f_k} = P(s_{f_k} \mid H_\phi)$. Thus, the probability that s_{f_k} will occur given any combination of classes is

$$P(s_{f_k} \mid H) = 1 - (1 - c_{\phi f_k}) \times \prod_{\forall v_i \in H} (1 - c_{if_k}). \tag{18}$$

The Noisy-OR model has two assumptions: All the dimensions and attributes are conditionally independent of each other given any hypothesis, and the classes are marginally independent. Using these assumptions along with conditional independence property and letting q_{if_k} denote $1 - c_{if_k}$, $P(H, \mathbf{s})$ can be calculated as follows:

$$\begin{aligned} P(H,\mathbf{s}) = P(H)P(\mathbf{s} \mid H) \;&=\; P(H)P(\mathbf{s}^{-}{}_k \mid H)P(\mathbf{s}^{+}{}_k \mid H) \\ &= P(H_\phi) \times \prod_{\forall v_i \in H} \frac{\pi_{\mathrm{i}}}{1-\pi_{\mathrm{i}}} \times \prod_{s_{f_k} \in \mathbf{s}^{-}{}_k} q_{\phi f_k} \times \prod_{s_{f_k} \in \mathbf{s}^{-}{}_k} \prod_{\forall v_i \in H} q_{if_k} \\ &\quad \times \prod_{\forall s_{f_k} \in \mathbf{s}^{\tau}_{f_k}} \left(1 - q_{\phi f_k} \prod_{\forall v_i \in H} q_{if_k}\right) \text{ (using (17) and (18)).} \end{aligned} \tag{19}$$

The implication of Noisy-OR model is that the truth of one hypothesis upon observing the evidence depends strongly on the truth of the other. Consider Fig. 7, if $c_{1i_1} = 1$ and $c_{2i_1} = 0$, i.e., V1 class causes attribute s_{i_1} while V2 does not. If attribute s_{i_1} becomes active on some assignment, the likelihood of V2 reduces (this is called "explaining away" effect [32]).

4.3 Multiple Labels Assignment

The computational complexity of exact inference on Bayesian networks is NP-hard. For small networks, inference is still practical. However, for large, richly connected networks, exact inference becomes intractable with increasing evidence. Given the intractability of exact inference on large, complex networks, researchers have pursued general purpose approximate methods based on stochastic sampling such as likelihood weighting ([21]) and Markov chain Monte Carlo simulations (Pearl [32]). For multiply connected networks, the standard ways of dealing with loops are clustering and conditioning. Clustering (as given in Lauritzen and Spiegelhalter [24]) involves forming compound variables in such a way that the resulting network of clusters is singly connected. Conditioning involves breaking the communication pathways along the loops by instantiating a select group of variables. Both the methods are liable to combinatorial problems if there are many intersecting cycles. Cooper [10] has shown that the problem of inference to obtain conditional probabilities in an arbitrary belief network is NP-hard. This suggests that it

will be quite useful to look for approximate or bounding methods rather than exact algorithms.

To find the most probable hypothesis, a search in the space of $2^{|V|}$ possible hypotheses is required. An efficient algorithm could be implemented by requiring that a video class ν be added to a hypothesis H (initially H_ϕ) only when there is an increase in the posterior probability, i.e., when $P(H \cup \nu \mid \mathbf{s})$ is greater than $P(H \mid \mathbf{s})$. Formulating the comparative probability M_ν^H as follows:

$$M_\nu^H = \frac{\mathrm{P(H} \cup \nu \mid \mathbf{s})}{\mathrm{P(H} \mid \mathbf{s})} = \frac{\mathrm{P}(\mathbf{s} \mid \mathrm{H} \cup \nu)\mathrm{P(H} \cup \nu)}{\mathrm{P}(\mathbf{s} \mid \mathrm{H})\mathrm{P(H)}} = \frac{\mathrm{P(H} \cup \nu, \mathbf{s})}{\mathrm{P(H}, \mathbf{s})}.$$

The M_ν^H is a measure of the increase or decrease in the degree to which the hypothesis H explains dimension value $\mathbf{s}$ due to the addition of the video class ν. Using (19), we derive a simple form of M_ν^H for the CBR Bayesian Network. Note that the first term and the third term are independent of the hypothesis in (19) and, hence, they will cancel out on taking ratio in M_ν^H. The second term deals with the prior probabilities of the classes only and, therefore, will evaluate to $\frac{\pi_\nu}{1-\pi_\nu}$ which is the a priori odds to class ν.

The fourth term involves more computation than any other term in (19) as it involves multiplying over all attributes that are not present and all classes. Due to the special structure of the CBR network, it reduces to $\prod_{\forall s_{f_k} \in \mathbf{s}^-{}_k} q_{\nu s_{f_k}}$ when we take the ratio.

Thus, M_ν^H reduces to the following:

$$M_\nu^H = \frac{\pi_\nu}{1-\pi_\nu} \times \prod_{\forall s_{f_k} \in \mathbf{s}^-{}_k} q_{\nu s_{f_k}} \times \frac{\prod_{\forall s_{f_k} \in s^\tau_{f_k}} (1 - \mathrm{q}_{\phi \mathrm{f_k}} \mathrm{q}_{\nu \mathrm{f_k}} \prod_{\forall \mathrm{v_i} \in \mathrm{H}} \mathrm{q}_{\mathrm{if_k}})}{\prod_{\forall s_{f_k} \in s^\tau_{f_k}} (1 - \mathrm{q}_{\phi \mathrm{f_k}} \prod_{\forall \mathrm{v_i} \in \mathrm{H}} \mathrm{q}_{\mathrm{if_k}})}. \tag{20}$$

The M_ν^H of a class ν for a hypothesis H cannot be less than the $M_\nu^{H^+}$ of ν for any extension of H, i.e., H^+ (for proof, see the appendix). This implies that, if $M_\nu^H \leq 1$, then ν can be eliminated as a path for exploring as an extension to H since it cannot lead to a more probable hypothesis. Thus, the only classes which need to be considered are those for which the M_ν^H is greater than 1. This provides a strong pruning heuristic for limiting the size of search tree.

4.4 Single Label Assignment

If we let $v_t, v_t \in \mathrm{V}$, be interpreted as the class that is present, a relative likelihood measure for a label v_t given $\mathbf{s}^+$ can be derived from (19) as

$$L(v_t, \mathbf{s}^+) = \frac{\pi_\mathrm{t}}{1-\pi_\mathrm{t}} \prod_{s_f \in \mathbf{s}^+} c_{tf}^{w_f} \prod_{s_l \in \mathbf{s}^-} (1 - c_{tl})^{w_l}. \tag{21}$$

This is the special case for a system where a segment is exclusively and necessarily (i.e., leak probability $c_{\phi f_k}$ is zero) labeled by one of the video class v_{max} with the highest likelihood measure L_{max}. There is refinement in (19) where it was assumed that all the dimensions had the same weightage in computation of the likelihood measure. In practice, different dimensions play different degree of importance in making a final decision for v_{max} and the relevance of a dimension s_f is reflected in its weight w_f.

This approach of weighing the dimensions is similar to that of some of the present CBR systems such as QBIC [16], Virage [15], and JACOB [7] which use a weighted linear method to combine the similarity measures of different dimensions. They rely on the user to specify the relative weights to the dimensions. However, a user has to be knowledgeable in the details of the system to be able to assign accurate weights. In contrast, in our system the relevant dimensions for a class are automatically assigned more weightage as compared to the other dimensions. The relative probability RPC(t,z) of one class t over another class z can be written as:

$$\begin{aligned} RPC(t,z) &= \frac{\mathrm{L(v_t}, \mathbf{s}^+)}{\mathrm{L(v_z}, \mathbf{s}^+)} \\ &= \frac{\pi_\mathrm{t}}{1-\pi_\mathrm{t}} \times \frac{1-\pi_\mathrm{z}}{\pi_\mathrm{z}} \times \frac{\prod_{\mathrm{s_f} \in \mathbf{s}^+} \mathrm{c}_{\mathrm{tf}}^{\mathrm{w_f}} \prod_{\mathrm{s_l} \in \mathbf{s}^-} (1 - \mathrm{c_{tl}})^{\mathrm{w_l}}}{\prod_{\mathrm{s_f} \in \mathbf{s}^+} \mathrm{c}_{\mathrm{zf}}^{\mathrm{w_f}} \prod_{\mathrm{s_l} \in \mathbf{s}^-} (1 - \mathrm{c_{zl}})^{\mathrm{w_l}}}. \end{aligned} \tag{22}$$

The classification task can be restated as assigning the class "t" if $RPC(t,z) \geq 1, \ \forall \ z$. Since the task involves iterative comparison among class pairs, weights can be assigned to the dimensions in a manner to enhance the distinction between the two classes. One approach which takes care of that is by employing the discriminatory capacity $|P_{tz}^{s_f}|$ of a dimension s_f for class $\mathrm{pair}(v_t, v_z)$ as weight where $0 \leq |P_{tz}^{s_f}| \leq 1$. The dimensions which have higher discriminatory capacity are more relevant in the computation of relative probability of one class over another. The logarithm of RPC can be taken after assigning weights in the above manner, as the number of dimensions could be very large and computation could easily result in an overflow beyond the numeral representation of a computer.

$$\begin{aligned} \log(RPC(t,z)) = {} & \log(\pi_t) - \log(\pi_z) + \log(1-\pi_z) - \log(1-\pi_t) \\ & + \sum_{s_f \in \mathbf{s}^+} |P_{tz}^{s_f}| \left[\log(c_{tf}) - \log(c_{zf})\right] \\ & + \sum_{s_l \in \mathbf{s}^-} |P_{tz}^{s_l}| \left[\log(1-c_{tl}) - \log(1-c_{zl})\right], \end{aligned} \tag{23}$$

where $\log(0)$ should be treated as $\log(\epsilon)$ where $\epsilon \ll 1$.

5 Experiment and Comparison

In this section, experiments using real video sequences are discussed. A comparison in performance is made with some standard classification tools. Some advantages of the present framework are highlighted in the ensuing discussion. The experimental details were presented in Section 2.2.

The ideas of performing association and classification in content-based classification are beginning to develop with the application of tools like Neural networks (for example, see Doulamis et al. [13]), decision trees (see Demsar and Solino [11]) and K-nearest neighbor classifier (see Yang and Kuo [40]). These works have different paradigms of operation from our CBR system in the sense that they do not envisage autonomous development of high-level classes from the knowledge extraction processes as we do. We

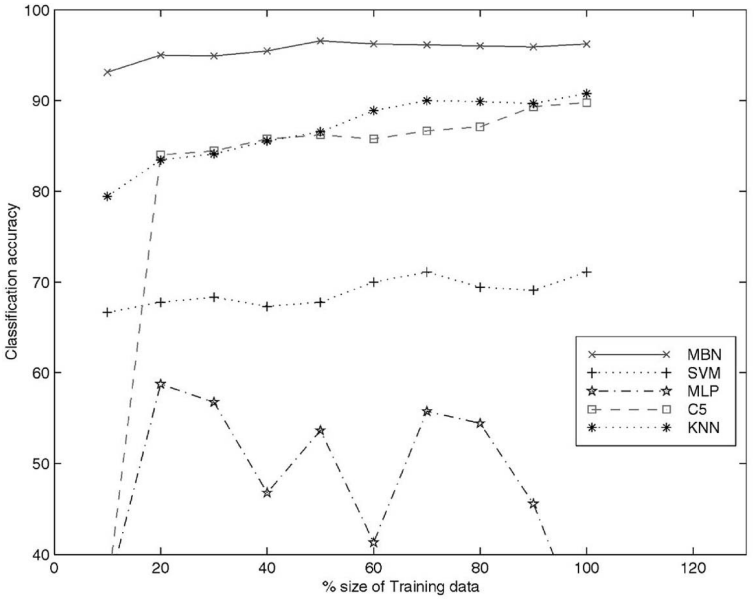

Fig. 8. Performance with varying size of training data. The number of classes was 9 and the dimensions was fixed at 593.

would like to present a comparison of our approach with Neural Networks (ANN), Support vector machines (SVM), K-Nearest Neighbor classifier (KNN), and decision trees.

5.1 Comparison

Some of the most well-known decision tree algorithms are C4.5 [34] and its improved successor C5.[1] We chose a C5 decision tree package for the purpose of comparison since it has many nice features like accurate and fast rule sets and fuzzy thresholding. The application of SVM to a domain of more than two target classes is still in the development phase; however, we use SVMTorch[2] C++ package [9] where the iterative process is performed by treating one class as $+1$ and the others as -1, thereby getting $|V|$ SVM models, where V is the set of classes. In neural networks, a feedforward backpropagation network was used with two hidden layers consisting of 150 neurons and 100 neurons. The training function used was the resilient backpropagation ("trainrp"). It is one of the best learning functions for classification problems, is not sensitive to the fine settings of training parameters and converges faster than other functions. The transfer functions employed were $\{tansig, tansig, tansig\}$, where $tansig$ is the hyperbolic tangent sigmoid transfer function. Single-label derivation (24) was used that performs meaningful dimension evaluation on Bayesian networks.

For the purpose of comparing performance, two sets of experiment were done under these conditions: a) with varying size of the training data and b) with varying number of dimensions. The type a experiment evaluates the generalization properties of the classification approach in relation to the nonlinear input-output mapping while the type b experiment demonstrates the effect of dimensionality on the performance.

Fig. 8 illustrates the comparison of the percentage classification accuracy on the type a experiment. The number of dimensions kept for this experiment set was 593 and the size of training data was varied by randomly selecting equal percentage of sequences for each class. Our approach is denoted as MBN (Modified Bayesian Network) and backpropagation networks as MLP. MBN has overall best performance, whereas KNN is the second best. It is interesting to note that the MBN performance is above 90 percent even with 10 percent of training samples, when the dimension space is relatively sparse. While in other methods, exact values are used; in the MBN approach, a partition is the basic unit of representation, which can provide good approximation when query point lies in the vicinity of the exact point. The second best performance of KNN could be attributed to the strategy of working in local regions as opposed to estimating some parameters for the entire descriptor space. However, in KNN, the boundaries are not so well defined as in MBN and, therefore, the boundary points are misclassified. Besides, in MBN, only meaningful dimensions are employed for inference.

Fig. 9 shows the classification accuracy of the various tools for a varying number of dimensions. The number of dimensions selected in MBN and its accuracy are also shown. The strategy in selecting dimensions was to select equal number of dimensions, as much as possible, from the descriptors in Table 1. With 10 dimensions only, the best performance was that of MBN and KNN (90 percent). It is noteworthy to observe that increasing the number of dimensions from 5 to 20 results in better performance by most of the tools and it appears that with a very few dimensions, information on distinction between the classes was less. The distinction achieved by a large number of dimensions shows the effectiveness of local feature extraction over the global one. On increasing the number of dimensions to more than 20, MBN gets sufficient meaningful dimensions and its performance is consistently more than 93 percent.

1. http://www.rulequest.com/see5-unix.html.
2. http://www.idiap.ch/learning/SVMTorch.html.

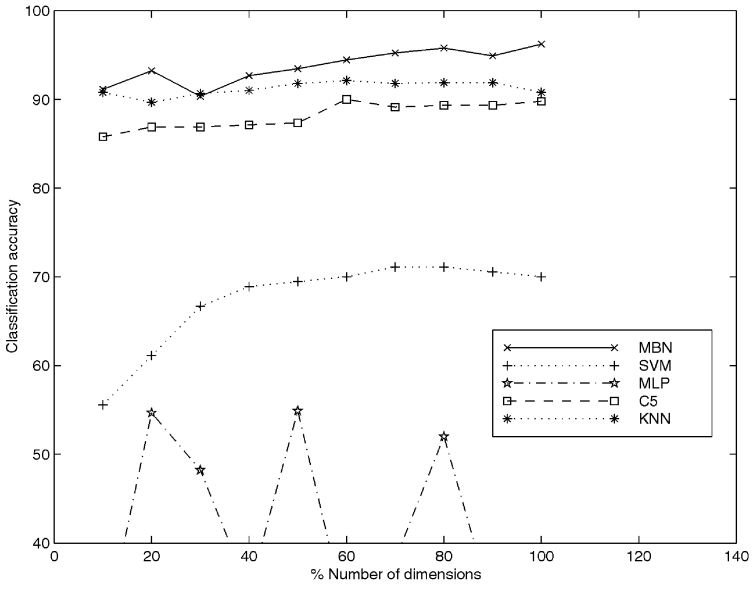

Fig. 9. Performance with varying number of dimensions.

5.2 Discussion

MBN offers significant advantages over other approaches in the domain of multimedia classification as it can be concluded from the experiments that MBN has good generalization capability as well as consistently good performance. The mechanism to support partial query by the user and also to label multiple classes on a single image/video segment, which gives multiple perspective to one image/segment, is provided. MBN neither require normalization of different dimensions nor the aid of an expert knowledge base.

On the other hand, tools like Neural networks suffer from the curse of dimensionality and it has been suggested in the literature [19] that the only practical way to overcome it is to incorporate prior knowledge about the function over and above the training data, which is known to be *correct*. However, this exercise is very difficult. In SVM, only a fixed (usually small) number of training-set vectors determine the parameters of the decision rule and, since no probability density is estimated, it becomes highly sensitive to the curse of dimensionality [35]. With a finite training sample, a high-dimensional feature space is almost empty [6] and many separators in SVM tool may perform well on the training data, but only few would generalize well. It has been shown by Wetson et al. [38] that both linear SVMs and nonlinear SVMs perform badly in the situation of many irrelevant features and they show how SVM performance can be improved by feature selection. In fact, we have shown in our previous work [27] that feature selection can improve SVM accuracy to 88 percent on similar video classification problem.

The performance of tools is also dependent on the distribution of the data. For instance, SVM is a useful tool to classify populations characterized by abrupt decreases in the density functions. However, in the real world, we have neither Gaussian populations nor data with sharp linear boundaries. KNN works well when there is a distribution of class clusters far apart from each other.

KNN also depends on the computation of distance measure and/or normalization of dimensions, which is not a simple task. Decision trees do not perform optimally with multimodal distribution. They also lack the correlation among dimensions in higher dimensions. While MBN considers the projection of the whole problem space, the decision tree, when selecting the best dimension, only considers a subregion corresponding to the current path in the tree. MBN performance is not critically dependent on any small part of the model, while decision trees are much more susceptible to small alterations in the model. It is also important to note that it is not a straightforward task in the AI tools considered, except MBN, to assign multiple labels to a segment.

6 Conclusion and Future Work

The main contribution of this paper is in presenting an approach where the steps in classification are derived from Bayesian theory. The discretization process developed was based on reducing the classification error on each dimension and it generalizes the application of the Bayesian network to a larger domain than those having Gaussian distribution of data or discrete data. A new dimensionality reduction method was presented which selects the dimensions with high class-pair discrimination capacity. It was compared with some standard algorithms. It was shown how multiple labels can be assigned to a multimedia data unit using Bayesian inference. The technique for single-label assignment was based on meaningful dimension evaluation in accordance with their significance in distinguishing the classes. Finally, the performance of our approach was compared with some standard classification tools.

Our work was based on assuming that the dimensions have direct causal relationship with the classes. However, in some practical situations, there could be intermediate nodes required to model the CBR system. Hidden nodes and, subsequently, the structure learning algorithm (like

[26]) would be then necessary in such cases. The discretization and dimensionality reduction algorithms would have to be extended for these cases.

APPENDIX

On taking the ratio of M_ν^H with M_ν^H, we get:

$$\frac{\mathrm{M}_\nu^{\mathrm{H}}}{\mathrm{M}_\nu^{\mathrm{H}^+}} = \prod_{\forall s_{f_k} \in \mathbf{s}_{f_k}^\tau} \frac{(1 - \mathrm{q}_{\phi \mathrm{f_k}} \mathrm{q}_{\nu \mathrm{f_k}} \prod_{\forall \mathrm{v_i} \in \mathrm{H}} \mathrm{q}_{\mathrm{if_k}})}{(1 - \mathrm{q}_{\phi \mathrm{f_k}} \prod_{\forall \mathrm{v_i} \in \mathrm{H}} \mathrm{q}_{\mathrm{if_k}})} \times \frac{(1 - \mathrm{q}_{\phi \mathrm{f_k}} \prod_{\forall \mathrm{v_i} \in \mathrm{H}} \mathrm{q}_{\mathrm{if_k}} \prod_{\forall \mathrm{v_j} \in \mathrm{H}^+, \mathrm{v_j} \notin \mathrm{H}} \mathrm{q}_{\mathrm{jf_k}})}{(1 - \mathrm{q}_{\phi \mathrm{f_k}} \mathrm{q}_{\nu \mathrm{f_k}} \prod_{\forall \mathrm{v_i} \in \mathrm{H}} \mathrm{q}_{\mathrm{if_k}} \prod_{\forall \mathrm{v_j} \in \mathrm{H}^+, \mathrm{v_j} \notin \mathrm{H}} \mathrm{q}_{\mathrm{jf_k}})}.$$

Substituting $q_{\phi f_k} \prod_{\forall v_i \in H} q_{if_k}$ as x_{f_k} and $\prod_{\forall v_j \in H^+, v_j \notin H} q_{jf_k}$ as y_{f_k}, the above equation can be written as

$$\frac{\mathrm{M}_\nu^{\mathrm{H}}}{\mathrm{M}_\nu^{\mathrm{H}^+}} = \prod_{\forall s_{f_k} \in \mathbf{s}_{f_k}^\tau} \frac{1 - \mathrm{x}_{\mathrm{f_k}} \mathrm{q}_{\nu \mathrm{f_k}}}{1 - \mathrm{x}_{\mathrm{f_k}}} \times \frac{1 - \mathrm{x}_{\mathrm{f_k}} \mathrm{y}_{\mathrm{f_k}}}{1 - \mathrm{x}_{\mathrm{f_k}} \mathrm{q}_{\nu \mathrm{f_k}} \mathrm{y}_{\mathrm{f_k}}} = \prod_{\forall s_{f_k} \in \mathbf{s}_{f_k}^\tau} \frac{1 + \mathrm{x}_{\mathrm{f_k}}^2 \mathrm{q}_{\nu \mathrm{f_k}} \mathrm{y}_{\mathrm{f_k}} - \mathrm{x}_{\mathrm{f_k}} (\mathrm{q}_{\nu \mathrm{f_k}} + \mathrm{y}_{\mathrm{f_k}})}{1 + \mathrm{x}_{\mathrm{f_k}}^2 \mathrm{q}_{\nu \mathrm{f_k}} \mathrm{y}_{\mathrm{f_k}} - \mathrm{x}_{\mathrm{f_k}} (1 + \mathrm{q}_{\nu \mathrm{f_k}} \mathrm{y}_{\mathrm{f_k}})}.$$

Since $0 \leq q_{\nu f_k}, y_{f_k} \leq 1$, $(1 + q_{\nu f_k} y_{f_k}) \geq (q_{\nu f_k} + y_{f_k})$. Each term in the iteration of $s_{f_k} \leq 1$ and, hence, $\frac{\mathrm{M}_\nu^{\mathrm{H}}}{\mathrm{M}_\nu^{\mathrm{H}^+}} \geq 1$.

ACKNOWLEDGMENTS

A. Mittal would like to express his deep gratitude to Dr. P.V. Krishnan, I.I.T. Delhi, for providing inspiration, motivation, and encouragement to take up this work. Both authors would like to thank the CIT Multimedia Laboratory for providing the facilities for digitization of the video data.

REFERENCES

[1] D.W. Aha and R.L. Bankert, "Feature Selection for Case-Based Classification of Cloud Types: An Empirical Comparison," *Proc. Am. Association for Artificial Intelligence Conf.*, pp. 106-112, 1994.

[2] H. Almuallim and T.G. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features," *Artificial Intelligence*, vol. 69, pp. 279-305, 1994.

[3] P. Antal, "Construction of a Classifier with Prior Domain Knowledge Formalized as Bayesian Network," *Proc. IEEE Conf. Industrial Electronics Soc.*, pp. 2527-2531, 1998.

[4] Y.A. Aslandogan and C.T. Yu, "Techniques and Systems for Image and Video Retrieval," *IEEE Trans. Knowledge and Data Eng.*, vol. 11, no. 1, pp. 56-63, Jan./Feb. 1999.

[5] M.B. Bassat, "Use of Distance Measures, Information Measures and Error Bounds in Feature Evaluation," *Handbook of Statistics*, Krishnaiah and Kanal, eds., pp. 773-791, 1982.

[6] P.S. Bradley and O.L. Mangasarian, "Feature Selection via Concave Minimization and SVMS," *Proc. Int'l Conf. Machine Learning*, pp. 82-90, 1998.

[7] M. L. Cascia and E. Ardizzone, "JACOB: Just a Content-Based Query System for Video Databases," *Proc. Int'l Conf. Acoustics, Speech, and Signal Processing*, pp. 1216-1219, 1996.

[8] E.J. Clarke and B.A. Barton, "Entropy and MDL Discretization of Continuous Variables for Bayesian Belief Networks," *Int'l J. Intelligent Systems*, vol. 15, pp. 61-92, 2000.

[9] R. Collobert and S. Bengio, "SVMTorch: Support Vector Machines for Large-Scale Regression Problems," *J. Machine Learning Research*, vol. 1, pp. 143-160, 2001.

[10] G.F. Cooper, "Probabilistic Inference Using Belief Networks is NP-Hard," Technical Report KSL-87-27, Stanford Univ., 1987.

[11] J. Demsar and F. Solina, "Using Machine Learning for Content-Based Image Retrieval," *Proc. Int'l Conf. Pattern Recognition*, vol. 3, pp. 138-142, 1996.

[12] P.J. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach.* Prentice Hall, 1982.

[13] N.D. Doulamis, A.D. Doulamis, and S.D. Kollias, "A Neural Network Approach to Interactive Content-Based Retrieval of Video Databases," *Proc. Int'l Conf. Image Processing*, vol. 2, pp. 116-120, 1999.

[14] P.A. Estevez, M. Fernandez, R.J. Alcock, and M.S. Packianather, "Selection of Features for the Classification of Wood Board Defects," *Proc. Int'l Conf. Artificial Neural Networks*, pp. 347-352, 1999.

[15] J. Bach, et al., "The Virage Search Engine: An Open Framework for Image Search Engine," *Proc. SPIE Conf. Storage and Retrieval of Image and Video Databases*, pp. 76-87, 1996.

[16] M. Flickner, et al., "Query by Image and Video Content: The QBIC System," *Computer*, pp. 23-32, Sept. 1995.

[17] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press 1990.

[18] F. Garber and A. Djouadi, "Bounds on the Bayes Classification Error Based on Pairwise Risk Functions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp. 281-288, 1988.

[19] S. Haykin, *Neural Network: A Comprehensive Foundation*, pp. 178-210, Macmililan, 1999.

[20] L. Heutte, J.V. Moreatu, B. Plessis, J.L. Plagnaud, and Y. Lecourtier, "Handwritten Numeral Recognition Based on Multiple Feature Extractors," *Proc. IEEE Int'l Conf. Document Analysis and Recognition*, pp. 167-170, 1993.

[21] K. Kanazawa, D. Koller, and S. Russell, "Stochastic Simulation Algorithms for Dynamic Probabilistic Networks," *Uncertainity in Artificial Intelligence*, pp. 346-351, 1995.

[22] W.A. Khatib, Y.F. Day, A. Ghafoor, and P.B. Berra, "Semantic Modeling and Knowledge Representation in Multimedia Databases," *IEEE Trans. Knowledge and Data Eng.*, vol. 11, pp. 64-80, 1999.

[23] A.F. Kohn, L.G. Nakano, and M.O. Silva, "A Class Discrimability Measure Based on Feature Space Partitioning," *Pattern Recognition*, pp. 873-887, 1996.

[24] S.L. Lauritzen and D.J. Spiegelhalter, "Local Computations with Probabilities on Graphical Structures and Their Applications to Expert Systems," *J. Royal Statistical Soc.*, pp. 157-224, 1988.

[25] H. Liu and R. Setiono, "Dimensionality Reduction via Discretization," *Knowledge Based Systems*, no. 9, pp. 67-72, 1996.

[26] C. Meek and D. Heckerman, "Structure and Parameter Learning for Causal Independence and Causal Interaction Model," *Uncertainity in AI*, pp. 366-375, 1997.

[27] A. Mittal and L. -F. Cheong, "Achieving Semantic Coupling in the Domain of High-Dimensional Video Indexing Application," *Proc. SPIE Conf. Applications of Artificial Neural Networks in Image Processing VI*, pp. 97-107, 2001.

[28] M. Modrzejewski, "Selection Using Rough Sets Theory," *Proc. European Conf. Machine Learning*, pp. 213-226, 1993.

[29] ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, "Overview of the MPEG-7 Standard," *Int'l Organization for Standarisation*, Oct. 2000.

[30] M. Pazzani, "An Interative Improvement Approach for the Discretization of Numeric Attributes in Bayesian Classifiers," *Proc. Int'l Conf. Knowledge Discovery and Data Mining*, pp. 228-233, 1995.

[31] M. Pazzani, C. Merz, K. Ali, and T. Hume, "Reducing Misclassification Costs," *Proc. Int'l Conf. Machine Learning*, 1994.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, 1988.

[33] B. Pfahringer, "Compression Based Discretization of Continuous Variables," *Proc. Int'l Conf. Machine Learning*, pp. 456-463, 1995.

[34] J.R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[35] S. Raudys, "How Good are SVMS?" *Neural Networks*, vol. 13, pp. 17-19, 2000.

[36] V.I. Smirnov and A.M. Tikheyeva, "The Connection between the Bayesian Risk and the Kolmogorov Distance and a Modification of It in Recognition Problems," *Eng. Cybernetics*, vol. 15, pp. 147-150, 1977.

[37] K. Tumer and J. Ghosh, "Estimating the Bayes Error Rate through Classifier Combining," *Proc. Int'l Conf. Pattern Recognition*, pp. 695-699, 1996.

[38] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, V. Vapnik, and T. Poggio, "Feature Selection for SVMs," *Neural Information Processing Systems*, pp. 668-674, 2000.

[39] N. Wyse, R. Dubes, and A.K. Jain, "A Critical Evaluation of Intrinsic Dimensionality Reduction Algorithms," *Pattern Recognition in Practise*, pp. 415-425, 1980.

[40] Z. Yang and C.C.J. Kuo, "A Semantic Classification and Composite Indexing Approach to Robust Image Retrieval," *Proc. Int'l Conf. Image Processing*, vol. 1, pp. 134-138, 1999.

Ankush Mittal received the BTech and Masters (by Research) degrees in computer science and engineering from the Indian Institute of Technology, Delhi. He received the PhD degree from The National University of Singapore in 2001. Since March 2001, he has been a faculty member in the Department of Computer Science, National University of Singapore. Prior to his PhD degree, he was working as an assistant professor in the Department of Computer Science, Birla Institute of Technology, India. His research interests are in multimedia indexing, machine learning, and motion analysis.

Loong-Fah Cheong received the BEng. degree from the National University of Singapore, and the PhD degree from University of Maryland, College Park, Center for Automation Research, in 1990 and 1996, respectively. In 1996, he joined the Department of Electrical and Computer Engineering, National University of Singapore, where currently he is an assistant professor. His research interests are related to the basic processes in the perception of three-dimensional motion, shape, and their relationship, as well as the application of these theoretical findings to specific problems in navigation and in multimedia systems, for instance, in the problems of video indexing in large databases.