

# RESOURCE-ADAPTIVE MULTIMEDIA ANALYSIS ON STREAM MINING SYSTEMS

*D. S. Turaga<sup>1</sup>, R. Yan<sup>1</sup>, O. Verscheure<sup>1</sup>, B. Foo<sup>2</sup>, F. Fu<sup>2</sup>, H. Park<sup>2</sup>, M. van der Schaar<sup>2</sup>*  
<sup>1</sup>IBM Research, Hawthorne, NY, USA. <sup>2</sup>University of California, Los Angeles, CA, USA.  
Email: {turaga, yanr, ov1}@us.ibm.com, {bkungfoo, fwfu, hgpark, mihaela}@ee.ucla.edu

## ABSTRACT

Large-scale multimedia semantic concept detection requires real-time identification of a set of concepts in streaming video or large image datasets. The potentially high data volumes of multimedia content, and high complexity associated with individual concept detectors, have hindered the practical deployment of many current solutions. In this paper, we present a summary of our work in building systems and applications for resource adaptive semantic concept detection in multimedia using large-scale distributed stream mining systems. We construct such concept detection applications as a hierarchical topology of individual concept detectors, and deploy them on distributed processing infrastructure. We then focus on dynamically configuring individual concept detectors to meet system imposed resource constraints while minimizing a penalty defined in terms of the misclassification cost. We present multiple centralized and distributed algorithms for this configuration, and describe the implemented application and system. We also verify through simulations that significant improvement in terms of accuracy of classification can be achieved through our approach.

## 1. INTRODUCTION

Recently, there has been the emergence of several applications that require processing and classification of continuous, high volume multimedia streams. These include online photo and video streaming services, search engines, spam filters, security services, etc. Each application may be viewed as a processing pipeline that analyzes streaming data from a set of raw data sources to extract valuable information in real time. In order to handle the naturally distributed set of data sources and jobs, as well as high computational burdens for the analytics, distributed stream mining systems have been recently developed [1]. These systems leverage computational resources from a set of distributed processing nodes and provide the framework to deploy and run different stream mining applications on various resource topologies. In such systems, complex jobs are decomposed into a network of operators performing feature extraction, classification, aggregation, and correlation. Such decomposition and distributed deployment has significant merits in terms of scalability, reliability, and performance objectives of large-scale, real-time stream mining applications.

In this paper we focus on applications for real-time semantic concept detection in multimedia streams, deployed on a stream mining system. We construct these applications as topologies of networked binary concept detectors, where the detectors are organized into topologies based on the semantic relationships between concepts of interest. A key research challenge lies in the management of limited system resources (e.g. CPU, memory, I/O bandwidth etc.) while providing desired application performance. Prior research in this area relies mostly on load-shedding, where algorithms determine a discard policy given the observed data characteristics e.g. burst, and the desired Quality of Service (QoS) requirements [5] [2]. These approaches are limited by their assumption that the impact of load

shedding on performance is known a-priori. They also impose significant overheads for computing any relevant metrics, and consider only locally available information, which may lead to sub-optimal end-to-end performance.

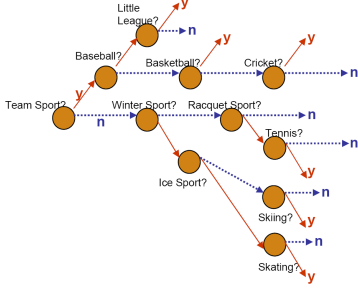
Instead of deciding on *what fraction* of the data to process, as in load-shedding based approaches, we determine *how* the available data should be processed given the underlying resource allocation. Hence, we allow individual classifiers in the topology to operate at different *performance levels* given their resource allocation. We define application utility, as a function of these performance levels, in terms of the end-to-end accuracy, i.e. desired tradeoff between probability of detection and probability of false alarm. Using this utility, our resource management problem may be formulated as a network optimization problem (NOP). However, unlike traditional NOPs, that determine a transmission rate per job, we configure the performance level, in terms of the operating point, of each classifier to meet system resource constraints while maximizing the end-to-end performance. We develop solutions for this problem using both a centralized optimization solution using Sequential Quadratic Programming (SQP), as well as distributed solution based on game theoretic principles. We then implement a multimedia analysis application, for semantic concept detection in sports images, along with our optimization algorithms on IBM's distributed stream mining system [1], and benchmark performance in terms of scale as well as misclassification penalty.

The paper is organized as follows. We introduce our application of interest in Section 2. In Section 3 we describe our models for classifiers, classifier tree topologies, and define utility and resource constraints for the optimization problem. In Section 4, we include a summary of our designed centralized as well as distributed solutions.

## 2. SEMANTIC CONCEPT DETECTION APPLICATION

We consider an application for hierarchical semantic concept detection [3] in streaming sports images. Incoming images of several different types are classified into six classes of interest: *Little League Baseball*, *Basketball*, *Cricket*, *Skating*, *Skiing* and *Tennis*, each of which identifies a specific type of sport. By introducing a set of additional intermediate concept detectors, we may then construct a hierarchical topology of classifiers such that not all classifiers need to process all the images. For instance, using a set of additional classifiers for *Team Sports*, *Winter Sports*, *Ice Sports*, *Racquet Sports* and *Baseball* concept identification, we can build a classifier tree as shown in Figure 1.

The *Team Sports* classifier filters data relevant to the *Little League*, *Cricket* and *Basketball* classifiers, the *Winter Sports* classifier filters data relevant to *Skating* and *Skiing* and the *Racquet Sports* classifier filters data relevant to *Tennis*. The mutually exclusive nature of concepts *Team Sports*, *Winter Sports* and *Racquet Sports* allows identifying these in series, i.e. passing only data that does not belong to a class to the next class. Using this hierarchy, the amount of data each classifier needs to process is significantly lower than the



**Fig. 1.** Hierarchical Classification Tree.

total data volume - depending on the a-priori probability of concept occurrence, leading to savings in resource consumption.

### 3. RESOURCE-ADAPTIVE CONCEPT DETECTION

Consider a binary classifier (concept detector)  $C_i$  in the topology shown in Figure 1. The total data rate or throughput entering  $C_i$  is labeled  $t_{i-1}$ . Since classifiers make mistakes, the throughput consists both of correctly labeled as well as incorrectly labeled data. The total goodput entering  $C_i$  is labeled  $g_{i-1}$ , where the goodput consists only of that portion of the throughput that is correctly classified. Classifier  $C_i$  labels data as belonging to the *yes* class or the *no* class, and forwards data appropriately. The a-priori conditional probability of the data belonging to the *yes* class for  $C_i$  (given the processing the data has already undergone) is labeled  $\phi_i$ . Correspondingly, the probability for the *no* class is  $1 - \phi_i$ .

The operation of  $C_i$  is characterized by its Receiver Operating Characteristic (ROC) curve, a curve that represents a tradeoff between probability of detection  $p^D$  and probability of false alarm  $p^F$ . Note that a binary classifier may have two such curves, one corresponding to the *yes* class and the other to the *no* class. These are coupled together when the classifier uses for instance one score threshold, i.e. if the classification score for a data object falls above a threshold it is labeled *yes* otherwise it is labeled *no*. In the most general case though, the classifier may use two separate thresholds, i.e. when the score falls above the first threshold, the data is labeled *yes*, if the data falls below the second threshold the data is labeled *no*. The use of two independent thresholds allows the classifier to both replicate data across its output branches (i.e. say both *yes* and *no*), as well as discard data (i.e. say neither *yes* nor *no*) as required. Hence the performance of the classifier is represented by two curves  $(p_i^D, p_i^F)$  for the *yes* class, and  $(\bar{p}_i^D, \bar{p}_i^F)$  for the *no* class.

Given these definitions, the output of  $C_i$ , i.e.  $t_i$  and  $g_i$  on the *yes* and  $\bar{t}_i$  and  $\bar{g}_i$  on the *no* may be determined recursively as:

$$\begin{bmatrix} t_i \\ g_i \end{bmatrix} = \begin{bmatrix} p_i^F & \phi_i(p_i^D - p_i^F) \\ 0 & \phi_i p_i^D \end{bmatrix} \begin{bmatrix} t_{i-1} \\ g_{i-1} \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \bar{t}_i \\ \bar{g}_i \end{bmatrix} = \begin{bmatrix} \bar{p}_i^D & (1 - \phi_i)(\bar{p}_i^F - \bar{p}_i^D) \\ 0 & \phi_i \bar{p}_i^D \end{bmatrix} \begin{bmatrix} t_{i-1} \\ g_{i-1} \end{bmatrix} \quad (2)$$

Given this recursive relationship, we can then determine the throughputs and goodputs at each terminal (leaf) classifier to determine the end-to-end throughput and goodput. The misclassification penalty may be defined in terms of the two types of errors - a penalty  $c^M$  per unit rate of missed detection, and a penalty  $c^F$  per unit rate of false alarm. For the *yes* output of  $C_k$ , we can compute this as:

$$\mathcal{U}_k = c_k^M (t_0 \pi_k - g_k) + c_k^F (t_k - g_k), \quad (3)$$

where  $\pi_k$  represents the a-priori probability of data actually belonging to the *yes* class of  $C_k$  and  $t_0$  represents the input data rate into the

tree. We can similarly define the penalty for the *no* class. The end to end misclassification penalty may thus be defined as a sum of these penalties for all the leaf classifiers. Note that this misclassification penalty function is non-concave in nature.

In addition to the misclassification penalty, we also need to model the resource consumption of each classifier. Given an underlying model, e.g. SVM, Bayes, Decision Tree etc., the computational requirements of a classifier  $C_i$  may be modeled as being directly proportional to the rate of data entering it. Hence we use a linear model:  $\rho_i = t_{i-1} \alpha_i$ , where  $\alpha_i$  represents the per-unit rate computations for the classifier. Finally, given a set of  $N$  classifiers, organized into a tree topology, placed on  $M$  resource-constrained nodes, the configuration problem involves determining the right set of operating points for each classifier such that the end to end misclassification penalty is minimized, while the  $M$  resource constraints are satisfied.

### 4. SUMMARY OF SOLUTIONS

In [6] we solve the topology configuration optimization problem using a centralized approach based on Sequential Quadratic Programming. In this case, the central optimizer has access to all the ROC curves, a-priori probabilities as well as resource placements and constraints. The gradient descent based approach guarantees convergence to local minima for the utility, however, the non-concavity of the utility as well as the constraints do not allow analytical bounds on performance. The approach clearly outperforms load-shedding approaches - leading to savings of a factor 1.5-2 times the misclassification penalty. This factor increases with increasing cost for false alarms, and with tightening resource constraints. Alternately, in [4] we propose a distributed solution to this problem, using game-theoretic principles. In this solution, individual classifiers select their operating points to maximize a local utility function. The utility may be purely local to the current classifier, corresponding to a *myopic* strategy, or may include the impact of the classifier actions on successive classifiers in the tree, corresponding to a *foresighted* strategy. We analytically show that foresighted actions of any classifier improve end-to-end performance over myopic strategies, and derive an associated probability bound. We then evaluate our algorithms on an application for hierarchical sports scene classification. We compare centralized, myopic and foresighted solutions and show that foresighted strategies outperform myopic strategies, and also asymptotically approach the centralized optimal solution as the number of actions available to each classifier increase. In this presentation, we will describe the application, the system, and these two different approaches in detail. We will include simulation results with comparisons against load-shedding approaches, and conclude with a summary of interesting directions for future research.

### 5. REFERENCES

- [1] L. Amini, H. Andrade, and et al. The stream processing core. Technical Report RSC 23798, IBM T.J. Watson Research Center, November 2005.
- [2] Y. Chi, P. S. Yu, H. Wang, and R. R. Muntz. Loadstar: A load shedding scheme for classifying data streams. *SDM*, 2005.
- [3] M. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. Hsu, L. Kennedy, A. Hauptmann, and J. Curtis. Large-scale concept ontology for multimedia. *IEEE MultiMedia*, 13(3):86-91, 2006.
- [4] H. Park, D. Turaga, O. Verscheure, and M. van der Schaar. A framework for distributed stream mining using coalition-based foresighted strategies. *IEEE ICASSP*, 2009.
- [5] N. Tatbul, U. Çetintemel, and S. B. Zdonik. Staying fit: Efficient load shedding techniques for distributed stream processing. *VLDB*, 2007.
- [6] D. Turaga, B. Foo, O. Verscheure, and R. Yan. Configuring topologies of distributed semantic concept classifiers for continuous multimedia stream processing. *ACM Multimedia*, 2008.