Chapter 12
Data-Driven Stream Mining Systems for Computer Vision

Shuvra S. Bhattacharyya, Mihaela van der Schaar, Onur Atan, Cem Tekin and Kishan Sudusinghe

Abstract In this chapter, we discuss the state of the art and future challenges in adaptive stream mining systems for computer vision. Adaptive stream mining in this context involves the extraction of knowledge from image and video streams in real-time, and from sources that are possibly distributed and heterogeneous. With advances in sensor and digital processing technologies, we are able to deploy networks involving large numbers of cameras that acquire increasing volumes of image data for diverse applications in monitoring and surveillance. However, to exploit the potential of such extensive networks for image acquisition, important challenges must be addressed in efficient communication and analysis of such data under constraints on power consumption, communication bandwidth, and end-to-end latency. We discuss these challenges in this chapter, and we also discuss important directions for research in addressing such challenges using dynamic, data-driven methodologies.

12.1 Introduction

In this chapter, we address challenges involving the development of algorithms, models, and design methods for distributed and adaptive real-time knowledge extraction of information from high volume image streams. We focus on an important emerging class of "big data" systems called *adaptive stream mining* (*ASM*) systems,

S.S. Bhattacharyya (✉) · K. Sudusinghe
University of Maryland, College Park, MD, USA
e-mail: ssb@umd.edu

K. Sudusinghe
e-mail: kishans@umd.edu

M. van der Schaar · O. Atan · C. Tekin
University of California, Los Angeles, CA, USA
e-mail: mihaela@ee.ucla.edu

O. Atan
e-mail: oatan@ucla.edu

C. Tekin
e-mail: cmtkn@ucla.edu

B. Kisačanin and M. Gelautz (eds.), *Advances in Embedded Computer Vision*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-3-319-09387-1_12

and discuss the state-of-the-art and challenges in design and implementation of effective ASM systems for embedded computer vision. ASM systems can be viewed as real-time data mining systems that operate on streams of data and are constructed as topologies (directed graphs) of classifiers, where parameters associated with the topologies and constituent classifiers may be manipulated dynamically based on changes in data characteristics, operational constraints, and other relevant run-time considerations.

Intended applications of ASM systems for embedded computer vision are very diverse, ranging from medical services, to dynamic management of vehicular traffic, to real-time detection of events in home-based health-care, to many kinds of surveillance and environmental monitoring applications. Each of these applications requires a topology of classifiers (such as a chain or "pipeline" configuration) that analyzes streaming data (which dynamically changes over time) from a set of raw data sources to extract valuable information in real time.

The need for adaptivity in ASM systems is inherent in almost all practical knowledge extraction application areas as data characteristics and operating conditions often exhibit uncertain or time-varying behavior. Accurate assessment, understanding, and optimization of ASM systems generally requires extensive experimentation of how algorithms for data classification and classifier adaptation interact with the characteristics of input data, and how scheduling and buffer management for such algorithms should be performed to satisfy real-time constraints subject to given resource constraints.

Decomposing applications as topologies of distributed processing operators has merits that transcend the scalability, reliability, and performance objectives of large-scale, real-time stream mining systems [1, 11, 18, 27]. Specifically, many stream classification and mining applications implement topologies (ensembles such as trees or cascades) of low-complexity binary classifiers to jointly accomplish the task of complex classification [24]. Such a structure enables the successive identification of multiple attributes in the data, and also provides significant advantages in terms of reduced resource consumption through appropriate dynamic data filtering, based on the incrementally identified attributes.

It has been shown that using a tree of binary classifiers can achieve better performance compared to other techniques such as support vector machines or SVMs (e.g., see [10]), rule-based techniques, and neural nets for some applications [6, 11, 19, 26, 28, 31, 42]. Furthermore, using classifiers operating in series with the same model (boosting [31]) or classifiers operating in parallel with multiple models (bagging [19]) has resulted in improved classification performance.

12.2 ASM System Example

Consider the surveillance application depicted in Fig. 12.1. A straightforward approach to dealing with this application requires the cameras to acquire the images on a continuous basis with the highest resolution, and send them to a central processing unit that is responsible for analyzing the images with complex data analytics.

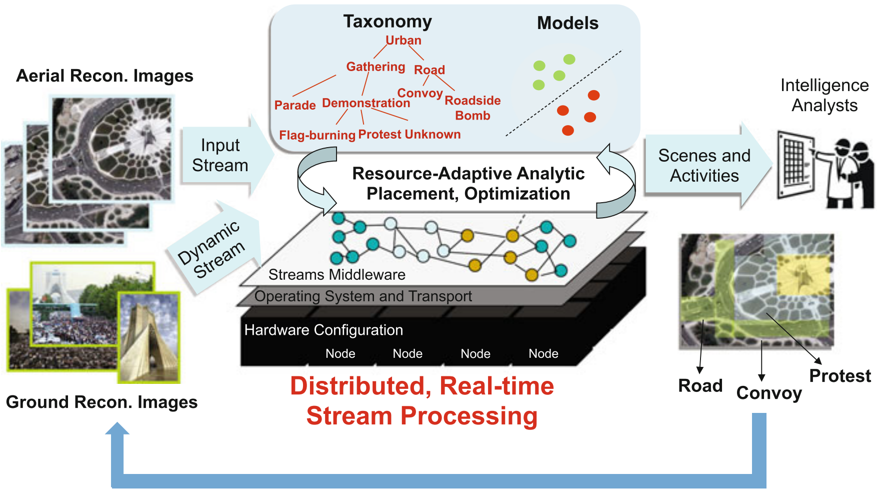


Fig. 12.1 An example of an ASM system for surveillance

Unfortunately, this approach is infeasible because it requires large communication bandwidths and energy consumption, and long transmission and processing delays. A feasible approach involves classifiers—localized in the same processing node of a camera—that are in charge of preprocessing the images. Based on the results of such preprocessing, the classifiers decide: (1) at which rate to acquire images, (2) whether or not to discard a specific image, and (3) in case the image is not discarded, the node to which the image must be sent for further processing and the resolution at which the image must be transmitted. Then the results of image processing can be exploited to trigger actions that modify the environment under observation (e.g., some roads are opened or closed) and even the stream mining system itself (e.g., additional cameras are turned on).

12.3 Challenges in ASM System Design

Key challenges in distributed real-time stream mining systems arise from the need to cope effectively with system overload due to large data volumes and limited system resources. There is a large computational cost incurred by each classifier (proportional to the data rate) that limits the rate at which the application can handle input video. Commonly used approaches to dealing with this problem in resource constrained stream mining are based on *load-shedding*, where algorithms determine when, where, what, and how much data to discard given the observed data characteristics, e.g. burst, desired Quality of Service (QoS) requirements [4, 5, 37–41], data value or delay constraints [12, 15].

An alternate approach to resource-constrained stream mining involves constructing topologies of classifiers based on hierarchical semantic concepts, and allowing individual classifiers in the topology to operate at different performance levels given the resources allocated to them. The performance level is determined by a classifier operating point that corresponds to the selected trade-off between probability of detection p_D and probability of false alarm p_F. Here, the probability of detection is defined as $p_D = p_{tp} + p_{tn}$, where p_{tp} and p_{tn} denote, respectively, the probability of a true positive, and the probability of a true negative.

This approach is illustrated in Fig. 12.2, where the curve on the right side shows a profile of the classifier accuracy in terms of the *detection error trade-off* (*DET*)—i.e., the trade-off of p_D versus p_F. Examples of operating points include decision thresholds for likelihood ratio tests or SVM normalized scores. Hence, instead of deciding on what fraction of the data to process, as in load-shedding approaches, such an approach determines *how* the available data should be processed given the underlying resource allocation. A solution based on this approach for configuring filtering applications that employ binary classifier chains has been proposed [14, 16–18].

Nevertheless, general binary tree topologies go significantly beyond linearly cascaded classifiers by providing greater flexibility in data processing, while also posing different challenges in terms of resource-constrained configuration. Specifically, while excess load can be easily handled within the optimization framework for a binary classifier chain, using a single operating point for each classifier in a tree generates two output streams with a total sum output rate that is fixed. Hence, it may not be possible to simultaneously meet tight processing resource constraints for downstream classifiers along both output edges when using only one operating point.

12.4 Dynamic, Data-Driven ASM Systems

Building on the conceptual framework of dynamically reconfigurable topologies of classifiers introduced in Sects. 12.1 and 12.3, an important direction for further work on stream mining for computer vision systems is in the rigorous integration of Dynamic Data Driven Applications Systems (DDDAS) into all aspects of processes for design and implementation. A significant class of future challenges for embedded computer vision therefore involves what may be referred to as *DDDAS-enabled ASM systems*.

12.4.1 DDDAS-Enabled ASM Systems

DDDAS is a paradigm that rigorously integrates application system modeling, instrumentation, and dynamic, feedback-driven adaptation of model and instrumentation parameters based on measured data characteristics [13]. DDDAS methods are

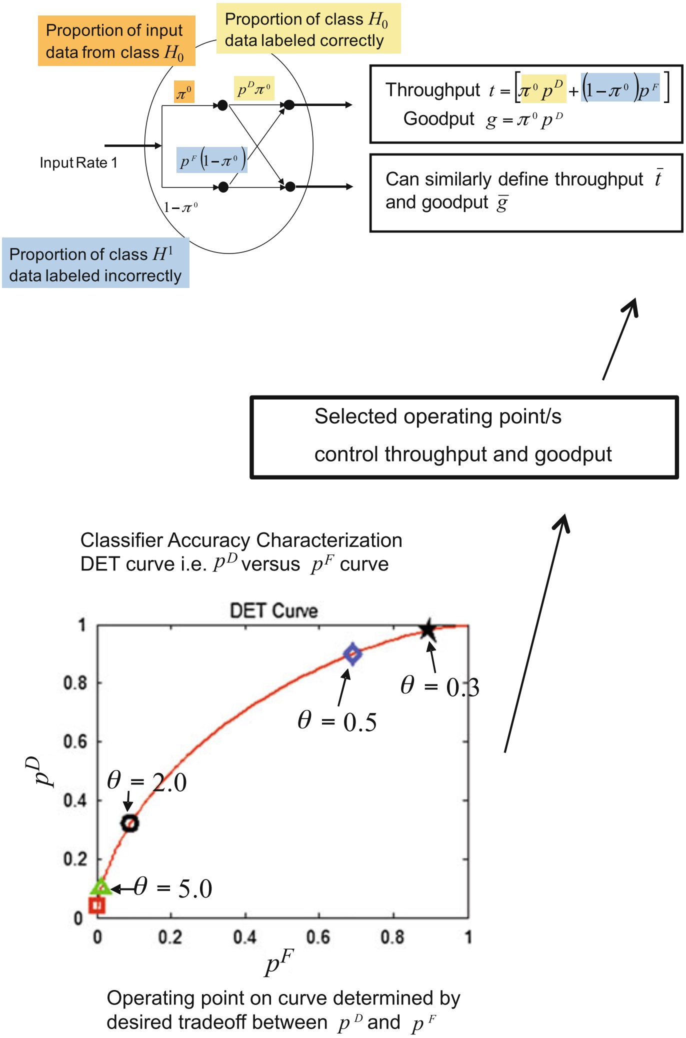


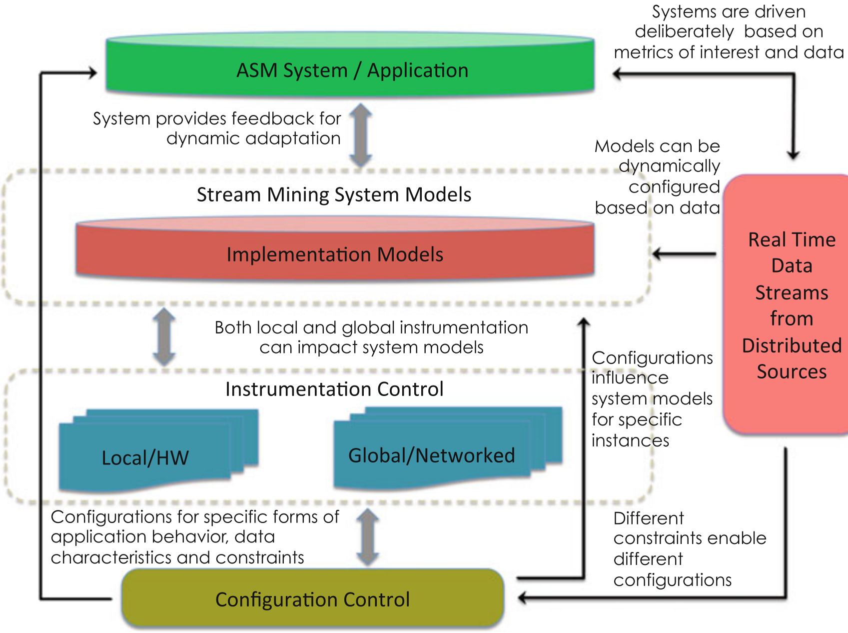
Fig. 12.2 An illustration of an adaptive and scalable classifier

highly relevant to design and implementation of ASM systems because they enable techniques for exploiting characteristics of the currently arriving set of image streams as well as characteristics of the overall operating environment to dynamically optimize critical trade-offs among key execution metrics, including power consumption, communication bandwidth, knowledge extraction accuracy, and end-to-end latency.

In ASM systems for embedded computer vision, DDDAS can be employed, for example, at network edges to systematically filter out image features that are not relevant to the current operational scenario or to adjust the resolution or frequency of captured images based on the type of object or amount of motion detected.

Such preprocessing at the network edges can help to reduce communication with a back-end server, and to improve overall system accuracy under communication and computation constraints. DDDAS techniques can also be employed at the server side. An example of such an application would be to dynamically determine the set of cameras at the network edges that should be active at a given time—e.g., to optimize trade-offs among energy efficiency, communication bandwidth requirements, and accuracy for the current image analysis scenario. In Sect. 12.5, we provide a detailed case study of DDDAS methods applied to a relevant application in embedded computer vision.

Use of DDDAS design techniques involves tightly integrated feedback from instrumentation. Use of DDDAS design techniques also involves application of dynamic parameters that are adapted based on such feedback, and that also control how subsequent rounds of instrumentation are performed. Figure 12.3 illustrates an abstract view of DDDAS as it relates to the class of stream mining systems addressed in this chapter.



Key challenges in integrating DDDAS principles into stream mining systems include the following.

- Development of abstract models for stream mining systems that can compactly and accurately represent the underlying design space of topological and classifier configurations. For this purpose, signal-processing-oriented dataflow models of computation are a promising starting point [8, 35].
- Development of methods to steer parameters of image stream acquisition (e.g., to select specific subsets of cameras or frame rates and resolutions for activated

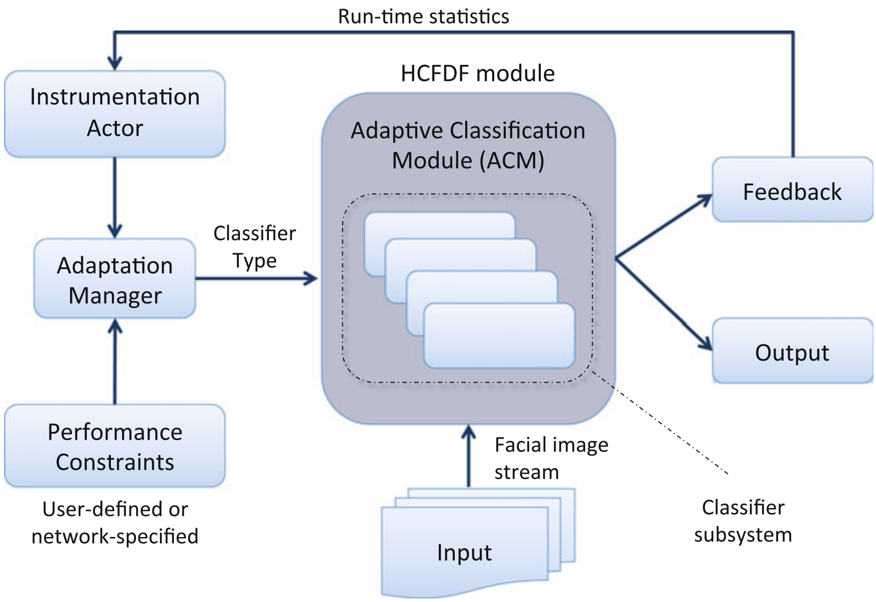


Fig. 12.3 An illustration of the LiD4E design tool and its application to DDDAS-enabled, multimedia, stream mining system design

cameras) based on the currently active regions of the stream mining design spaces, as estimated, for example, with the help of the abstract models described above.

- Development of methods to dynamically optimize the mapping of ASM topologies onto the targeted hardware platforms based on current configurations for the topologies, and their constituent classifiers. This mapping process may be especially challenging due to dynamics in stream mining topology characteristics, resource constraints on the target platforms or severe application requirements in terms of the volume of image data that needs to be processed, real-time constraints, etc.

Lightweight Dataflow for Dynamic Data-Driven Application Systems Environment (*LiD4E*) is a recently-developed design tool to help in the investigation of these challenges and other aspects of DDDAS-enabled stream mining systems. We discuss LiD4E next, in Sect. 12.4.2.

12.4.2 LiD4E

In this section, we provide an overview of LiD4E, which is a design environment that has been developed to facilitate experimentation with methods for DDDAS-enabled ASM system design, with emphasis on multimedia ASM systems [35].

A key feature of LiD4E is the provision for signal processing pipelines (i.e., chains of signal processing modules, such as classifiers, digital filters and transform operators) that can be data-dependent and dynamically changing. LiD4E employs *hierarchical core functional dataflow* (*HCFDF*) semantics as the specific form of dynamic dataflow [35]. HCFDF and the core functional dataflow (CFDF) model [29] that it extends belong to the class of signal-processing-oriented dataflow models of computation described in Sect. 12.4.1. HCFDF can be viewed as a hierarchical extension of CFDF. Through its emphasis on supporting structured, application-level dynamic dataflow modeling, HCFDF provides a formal, model-based framework through which stream mining applications can be designed and analyzed precisely in terms of integrated principles of DDDAS and dataflow.

In HCFDF graphs, actors are specified in terms of sets of processing modes, where each mode has static *dataflow rates*—i.e., each mode produces and consumes a fixed number of data values (tokens) on each actor port. However, different modes of the same actor can have different dataflow rates, and the actor mode can change from one actor execution (*firing*) to the next, thereby allowing for dynamic dataflow behavior (dynamic rates). Additionally, HCFDF allows dataflow graphs to be hierarchically embedded (nested) within actors of higher level HCFDF graphs, thereby allowing complex systems to be constructed and analyzed in a scalable manner. The design rules prescribed for hierarchical composition in HCFDF graphs ensure that actors at each level in a design hierarchy conform to the semantics of HCFDF or some restricted subset of HCFDF semantics, such as cyclo-static dataflow or synchronous dataflow (SDF) [9, 23]. For further details on HCFDF semantics, we refer the reader to [35].

As demonstrated in [35], HCFDF modeling enables run-time adaptation of signal processing topologies, including dataflow graphs that are constructed using arbitrary combinations of classifiers, filters, and transform units. Through the inclusion of a special HCFDF design component called an *adaptive classification module*, the designer can invoke multiple operating modes at run-time, and selection of such operating modes can be driven based on system feedback—e.g., based on instrumentation that monitors data characteristics, and guides selection based on desired trade-offs among performance, accuracy, and energy consumption.

Figure 12.3 provides an illustration of the LiD4E design tool and its application to DDDAS-enabled, multimedia, stream mining system design. For more details on LiD4E, we refer the reader to [35]. Extensions of the design principles in LiD4E to handle multi-mode stream mining systems are discussed in [34].

12.5 Case Study: Learning Based on Multi-armed Bandits

In this section, we present a case study in data-driven ASM techniques that are relevant for the emerging class of a ASM-enabled, embedded computer vision systems introduced in Sect. 12.1 through Sect. 12.4. The methods presented in the case study can be viewed as representative of the kinds of advances that are needed to address the challenges in providing robust, efficient, and integrated stream mining solutions for next-generation embedded computer vision systems.

The methods discussed in this section were originally presented in [2]. In this section, we provide a concise summary of the developments in [2] in the context of ASM systems for embedded computer vision. For full details on these methods, we refer the reader to [2].

12.5.1 Overview

In most video-based object or face recognition services on mobile devices, each device captures and transmits video frames over a wireless channel to a remote computing service (a.k.a. the "cloud") that performs the heavy-duty video feature extraction and recognition tasks for a large number of mobile devices. The major challenges of such scenarios stem from the highly-varying contention levels in the wireless local area network (WLAN), as well as the variation in the task-scheduling congestion in the cloud.

In order for each device to maximize its object or face recognition rate under such contention and congestion variability, a systematic learning framework based on *multi-armed bandits* has been developed [2]. Unlike well-known reinforcement learning techniques that exhibit very slow convergence rates when operating in highly-dynamic environments, this bandit-based, systematic learning approach quickly approaches the optimal transmission and processing-complexity policies based on feedback on the experienced dynamics (contention and congestion levels). The case study presented in this section centers on this bandit-based, systematic learning approach.

Many of the envisaged applications and services for wearable sensors, smartphones, tablets or portable computers in the next ten years will involve analysis of video streams for event, action, object or user recognition [21, 32]. In this process, they experience time-varying channel conditions, traffic loads and processing constraints at the remote cloud-computing servers where the data analysis takes place. Examples of early commercial services in this domain include Google Goggles, Google Glass, Facebook automatic face tagging [7], and Microsoft's Photo Gallery face recognition.

12.5.2 Application Example

Figure 12.4 presents an example of such deployments. Video content producers include several types of sensors, mobile phones, as well as other low-end portable devices, that capture, encode (typically via a hardware-supported MPEG/ITU-T codec) and transmit video streams to a remote computing server for recognition or authentication purposes. A group of M devices in the same WLAN comprises a

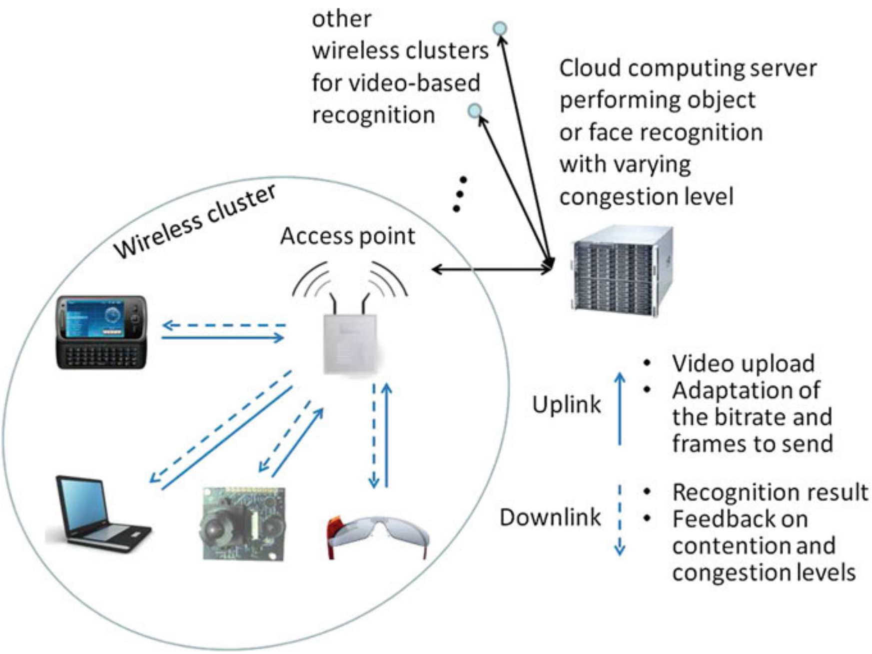


Fig. 12.4 Illustration of object or face recognition via adaptive wireless video transport to a remote computing server

wireless cluster. A server running openstack or Hadoop (or a similar runtime environment suitable for cloud computing) [25] is used for analyzing visual data from numerous wireless clusters, as well as other computing tasks unrelated to object or face recognition.

Each device can adapt the encoding bitrate, as well as the number of frames to produce (with the ensemble of N such settings comprising the set $\mathscr{A} = \{a_1, a_2, \ldots, a_N\}$), in order to alleviate the impact of contention in the WLAN. At the same time, the visual analysis performed in the cloud can be adapted to scale the required processing time to alleviate the impact of task scheduling congestion in the cloud [25, 30], with the sets of contention and congestion levels represented by the discrete sets $\mathscr{T}$ and $\mathscr{G}$, respectively. In return, each device receives from the cloud a label that describes the recognized object or face (e.g., the object or person's name), or simply a message that the object or person could not be recognized. In addition, each device or wireless cluster can also receive feedback on the experienced WLAN medium access control (MAC) layer contention and the cloud task scheduling congestion conditions.

Thus, the "reward" for each device is the recognition result at each time step. Given that each wireless access point and the cloud computing infrastructure serve many more requests than the ones from a given cluster of devices (as illustrated in Fig. 12.4), we can safely assume that for each device, the wireless contention and cloud congestion levels are both independent of the actions taken by the devices within their clusters. This makes each device independent, since the decisions made by other devices do not affect the reward.

12.5.3 Relation to Prior Work

Each mobile device of Fig. 12.4 seeks to maximize its own expected recognition rate at the minimum possible cost in terms of utilized wireless resources (i.e., MAC superframe transmission opportunities used). To this end, several approaches have been proposed that are based on reinforcement learning [36], such as Q-learning [30]. In these, the goal is to learn the state-value function, which provides a measure of the expected long-term performance (utility). However, they incur large memory overheads for storing the state-value function, and they are slow to adapt to new or dynamically changing environments. A better approach is to intermittently explore and exploit when needed, in order to capture such changes. Index policies for multi-armed bandit (MAB) problems, contextual bandits [22, 33], or epsilon-decreasing algorithms [3] can be used for this task. However, all existing bandit frameworks do not take into consideration the contention and congestion conditions as contexts in the application under consideration.

12.5.4 Learning Based on Multi-user Bandits

Motivated by the lack of efficient methods that fully capture the problems related to online learning in multi-user wireless networks and cloud computing systems with uncertain and highly-varying resource provisioning, an online systematic learning theory based on multi-user contextual bandits has been developed. This learning theory can be viewed as a natural extension of the basic MAB framework. Analytic estimates have been derived to compare its efficiency against the complete knowledge (or "oracle") benchmark in which the expected reward of every choice is known by the learner. Unlike Q-learning [36] and other learning-based methods, it is proven that the regret bound—the loss incurred by the algorithm against the best possible decision that assumes full knowledge of contention and congestion conditions—is logarithmic if users do not collaborate and each would like to maximize the user's own utility. Finally, the contextual bandit framework discussed here is general, and can be used for learning in various kinds of wireless embedded computer vision applications that involve offloading of selected processing tasks. Henceforth in this chapter, we refer to the contextual bandit framework by the abbreviation *CBF*.

Table 12.1 Average attempts (with the oracle bound given in parentheses) to obtain a recognition rate of 0.9 with 2D-PCA

Method	Iteration			
	$T = 50$	$T = 100$	$T = 250$	$T = 1{,}000$
CBF	3.3 (1.7)	3.1 (1.6)	2.4 (1.5)	1.9 (1.5)
CBF no context	3.1 (1.7)	2.8 (1.6)	2.6 (1.6)	2.4 (1.6)
Q-learning	3.5 (1.7)	2.8 (1.6)	2.7 (1.5)	2.2 (1.5)

12.5.5 Numerical Results

The CBF has been evaluated by simulation. The simulation environment comprises four mobile devices connected via an IEEE 802.11 WLAN to a cloud-computing server. Videos of human faces are produced by random images of persons taken from the extended Yale Face Database B (39 cropped faces of human subjects under varying illumination) [20]. Each video comprises 34 images from the same person, and is compressed to a wide range of bitrates via the H.264/AVC codec (x264 codec, crf $\in \{4, 14, 24, 34, 44, 51\}$). The 2D PCA algorithm [43] is used at the cloud side for face recognition from each decoded video (with the required training done offline as per the 2D PCA setup [43]). More than 80 % of the video frames have to match to the same person in the database to declare a given video as "recognized". There is a time window set for recognition, which limits the number of frames received by the cloud under varying WLAN contention levels (delay is increased under contention due to the backoff and retransmissions of IEEE 802.11 WLANs). Similarly, because of randomly varying congestion in the cloud, only a limited number of the received video frames is actually used by 2D PCA, thereby affecting the recognition rate.

Table 12.1 presents the average number of retries performed per recognition action by the CBF method (with and without using the cloud congestion information as context) in order to achieve a recognition rate of 90 %. Results are also presented in the following ways.

- An optimal solution that selects the transmission setting yielding the highest expected recognition rate [2]. This solution is defined as the *oracle solution*, since it assumes that all conditions for each case are precisely known beforehand.
- Q-learning [36, 44], as discussed in Sects. 12.5.3 and 12.5.4.

The results indicate that after 250 recognition attempts (each attempt comprises the retries listed), the CBF method approaches the oracle bound, and for the same recognition rate, incurs less retries per attempt in comparison to Q-learning.

12.5.6 Summary

In this section, we have examined in some detail a concrete case study of emerging methods for data-driven, ASM system design targeted to embedded computer vision. In particular, we have discussed a contextual bandit framework (CBF) for learning contention and congestion conditions in object or face recognition via wireless mobile streaming and cloud-based processing. Analytic results show that the CBF framework converges to the value of the oracle solution (i.e., the solution that assumes full knowledge of congestion and contention conditions). Simulations within a cloud-based face recognition system demonstrate that the CBF approach outperforms Q-learning, as it quickly adjusts to contention and congestion conditions. For more details on the CBF approach, we refer the reader to [2].

12.6 Future Directions in Stream Mining Systems for Computer Vision

Most existing solutions for designing and configuring computer vision and stream-mining systems based on the extracted visual data offload their processing to the cloud and assume that the underlying characteristics (e.g., visual characteristics) are either known, or that simple-yet-accurate models of these characteristics can be built. However, in practice, this knowledge is not available and models of such computer vision applications or the associated processing mechanisms are very difficult to build and calibrate for specific environments, since these characteristics are dynamically varying over time. Hence, despite applying optimization, these solutions tend to result in highly sub-optimal performance since the models they use for the experienced dynamics are not accurate. Hence, reinforcement learning (i.e., learning how to act based on past experience) becomes a vital component in all such systems. Some of the best-performing online reinforcement learning algorithms are Q-learning and structural-based reinforcement learning. In these, the goal is to learn the state-value function, which provides a measure of the expected long-term performance (utility) when it is acting optimally in a dynamic environment. It has been proven that online learning algorithms converge to optimal solutions when all the possible system states are visited infinitely often [36].

However, these methods have to learn the state-value function at every possible state. As a result, they incur large memory overheads for storing the state-value function and they are typically slow to adapt to new or dynamically changing environments (i.e., they exhibit a slow convergence rate), especially when the state space is large—as in the considered wireless transmission and recognition problem of Sect. 12.5. These memory and speed-of-learning deficiencies are alleviated in structural-based learning solutions. Despite this, a key limitation still remains: *all these schemes provide only asymptotic bounds for the learning performance—no speed-of-learning guarantees are provided*. Nevertheless, in most computer vision and recognition systems, users are interested in both short-term performance and long-term performance.

12.7 Conclusion

In this chapter, we have introduced the emerging area of adaptive stream mining systems for embedded computer vision, and we have discussed important research challenges in this area. We have emphasized key challenges in integrating methods of Dynamic Data Driven Applications Systems (DDDAS) rigorously in the design and implementation process for the targeted class of embedded computer vision systems. We have discussed the Lightweight Dataflow for Dynamic Data-Driven Application Systems Environment (LiD4E) as a recently-introduced design tool for experimenting with DDDAS-enabled stream mining methods. As a concrete example of recent advances in DDDAS-enabled adaptive stream mining, we have presented a case study involving learning based on multi-armed bandits. As motivated in this chapter, addressing the future challenges of adaptive stream mining systems for embedded computer vision will require interdisciplinary advances in areas that include machine learning, DDDAS design methods, and distributed embedded systems.

Acknowledgments This work is supported by the US Air Force Office of Scientific Research under the DDDAS Program.

References

1. Amini L, Andrade H, Eskesen F, King R, Park Y, Selo P, Venkatramani C (2005) The stream processing core. Technical Report RSC 23798
2. Atan O, Andreopoulos Y, Tekin C (2014) Bandit framework for systematic learning in wireless video-based face recognition. In: Proceedings of the international conference on acoustics, speech, and signal processing
3. Auer P, Cesa-Bianchi N, Fischer P (2002) Finite-time analysis of the multiarmed bandit problem. Mach Learn 47(2–3):235–256
4. Babcock B, Babu S, Datar M, Motwani R (2003) Chain: operator scheduling for memory minimization in data stream systems. In: ACM SIGMOD
5. Babcock B, Datar M, Motwani R (2003) Cost-efficient mining techniques for data streams. In: Proceedings of the workshop on management and processing of data streams
6. Balazinska M, Balakrishnan H, Madden S, Stonebraker M (2005) Fault tolerance in the borealis distributed stream processing system. In: Proceedings of the international conference on management of data
7. Becker BC, Ortiz EG (2008) Evaluation of face recognition techniques for application to facebook. In: Proceedings of the IEEE international conference on automatic face and gesture recognition, pp 1–6
8. Bhattacharyya SS, Deprettere E, Leupers R, Takala J (eds) (2013) Handbook of signal processing systems, 2nd edn. Springer, New York. http://dx.doi.org/10.1007/978-1-4614-6859-2. ISBN: 978-1-4614-6858-5 (Print); 978-1-4614-6859-2 (Online)

9. Bilsen G, Engels M, Lauwereins R, Peperstraete JA (1996) Cyclo-static dataflow. IEEE Trans Signal Process 44(2):397–408
10. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Knowl Discov Data Min 2(2):121–167
11. Cherniack M, Balakrishnan H, Carney D, Cetintemel U, Xing Y, Zdonik S (2003) Scalable distributed stream processing. In: CIDR
12. Chi Y, Yu PS, Wang H, Muntz RR (2005) Loadstar: a load shedding scheme for classifying data streams. In: Proceedings of the SIAM conference on data mining
13. Darema F (2005) Grid computing and beyond: The context of dynamic data driven applications systems. Proc IEEE 93(2):692–697
14. Ducasse R, Turaga D, van der Schaar M (2010) Adaptive topologic optimization for large-scale stream mining. IEEE J Sel Top Sign Proces 4(3):620–636
15. Eide V, Eliassen F, Granmo O, Lysne O (2003) Supporting timeliness and accuracy in distributed real-time content-based video analysis. In: Proceedings of the ACM international conference on multimedia
16. Foo B, van der Schaar M (2010) A distributed approach for optimizing cascaded classifier topologies in real-time stream mining systems. IEEE Trans Image Process 19(11):3035–3048
17. Foo B, Turaga D, Verscheure O, Amini L, van der Schaar M (2011) Configuring trees of classifiers in distributed multimedia stream mining systems. IEEE Trans Circuits Syst Video Technol 21(3):245–258
18. Fu F, Turaga D, Verscheure O, van der Schaar M, Amini L (2007) Configuring competing classifier chains in distributed stream mining systems. IEEE J Sel Top Sign Process 1(4):548–563
19. Garg A, Pavlovic V (2002) Bayesian networks as ensemble of classifiers. In: Proceedings of the international conference on pattern recognition, pp 779–784
20. Georghiades AS, Belhumeur PN, Kriegman D (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660
21. Girod B, Chandrasekhar V, Chen DM, Cheung N, Grzeszczuk R, Reznik Y, Takacs G, Tsai SS, Vedantham R (2011) Mobile visual search. IEEE Signal Process Mag 28(4):61–76
22. Langford J, Zhang T (2008) The epoch-greedy algorithm for multi-armed bandits with side information. In: Platt JC, Koller D, Singer Y, Roweis S (eds) Advances in neural information processing systems. Curran Associates, New York, pp 817–824
23. Lee EA, Messerschmitt DG (1987) Synchronous dataflow. Proc IEEE 75(9):1235–1245
24. Lienhart R, Liang L, Kuranov A (2003) A detector tree for boosted classifiers for real-time object detection and tracking. In: Proceedings of the IEEE international conference on multimedia and expo
25. Li A, Yang X, Kandula S, Zhang M (2010) CloudCmp: comparing public cloud providers. In: Proceedings of the ACM SIGCOMM conference on internet measurement, pp 1–14
26. Ntoulas A, Najork M, Manasse M, Fetterly D (2006) Detecting spam web pages through content analysis. In: Proceedings of the international conference on world wide web
27. Olston C, Jiang J, Widom J (2003) Adaptive filters for continuous queries over distributed data streams. In: ACM SIGMOD
28. Pang S (2004) SVM classification tree algorithm with application to face membership authentication. In: Proceedings of the international joint conference on neural networks
29. Plishker W, Sane N, Kiemb M, Anand K, Bhattacharyya SS (2008) Functional DIF for rapid prototyping. In: Proceedings of the international symposium on rapid system prototyping, Monterey, California, pp 17–23
30. Ren S, van der Schaar M (2013) Efficient resource provisioning and rate selection for stream mining in a community cloud. IEEE Trans Multimedia 15(4):723–734
31. Schapire RE (2003) The boosting approach to machine learning: an overview. In: Denison DD, Hansen MH, Holmes C, Mallick B, Yu B (eds) Nonlinear estimation and classification. Springer, New York
32. Siewiorek D (2012) Generation smartphone. IEEE Spectr Mag 49(9):54–58

33. Slivkins A (2011) Contextual bandits with similarity information. In: Proceedings of the conference on learning theory
34. Sudusinghe K, Cho I, van der Schaar M, Bhattacharyya SS (2014) Model based design environment for data-driven embedded signal processing systems. In: Proceedings of the international conference on computational science, Cairns, Australia, pp 1193–1202
35. Sudusinghe K, Won S, van der Schaar M, Bhattacharyya SS (2013) A novel framework for design and implementation of adaptive stream mining systems. In: Proceedings of the IEEE international conference on multimedia and expo, San Jose, California, pp 1–6. http://ieeexplore.ieee.org
36. Sutton RS, Barto AG (1998) Reinforcement learning: an introduction. MIT Press/Bradford Books, Cambridge
37. Tatbul N (2002) Qos-driven load shedding on data streams. In: XML-based data management and multimedia engineering
38. Tatbul N, Etintemel UC, Zdonik SB (2007) Staying fit: efficient load shedding techniques for distributed stream processing. In: Proceedings of the international conference on very large databases
39. Tatbul N, Etintemel UC, Zdonik SB, Cherniack M, Stonebraker M (2003) Load shedding in a data stream manager. In: Proceedings of the international conference on very large databases, pp 309–320
40. Tatbul N, Zdonik SB (2006) Dealing with overload in distributed stream processing systems. In: Proceedings of the IEEE international workshop on networking meets databases
41. Viglas S, Naughton J (2012) Rate-based query optimization for streaming information sources. In: Proceedings of ACM SIGMOD
42. Wu JRSY, Tseng B (2004) Ontology-based multi-classification learning for video concept detection. In: Proceedings of the IEEE international conference on multimedia and expo
43. Yang J, Zhang D, Frangi AF, Yang J (2004) Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans Pattern Anal Mach Intell 26(1):131–137
44. Zhu X, Lan C, van der Schaar M (2013) Low-complexity reinforcement learning for delay-sensitive compression in networked video stream mining. In: Proceedings of the IEEE international conference on multimedia and expo