The final version of this manuscript appeared in the Proceedings of the IEEE IFIP Annual Mediterranean Ad Hoc Networking Workshop, Med-Hoc-Net 2014, June 2-4, 2014, Piran, Slovenia.
DOI: not yet available

A Machine Learning Approach to QoE-based Video Admission Control and Resource Allocation in Wireless Systems

Alberto Testolin[‡], Marco Zanforlin[†], Michele De Filippo De Grazia[‡]
Daniele Munaretto[†], Andrea Zanella[†], Marco Zorzi[‡], Michele Zorzi[†*]
† Department of Information Engineering, University of Padova, Italy
‡ Department of General Psychology, University of Padova, Italy
* CalIT2, University of California at San Diego, USA
E-mail: {*firstname.lastname*}@unipd.it

***Abstract*—The rapid growth of video traffic in cellular networks is a crucial issue to be addressed by mobile operators. An emerging and promising trend in this regard is the development of solutions that aim at maximizing the Quality of Experience (QoE) of the end users. However, predicting the QoE perceived by the users in different conditions remains a major challenge. In this paper, we propose a machine learning approach to support QoE-based Video Admission Control (VAC) and Resource Management (RM) algorithms. More specifically, we develop a learning system that can automatically extract the quality-rate characteristics of unknown video sequences from the *size* of H.264-encoded video frames. Our approach combines unsupervised feature learning with supervised classification techniques, thereby providing an efficient and scalable way to estimate the QoE parameters that characterize each video. This QoE characterization is then used to manage simultaneous video transmissions through a shared channel in order to guarantee a minimum quality level to the final users. Simulation results show that the proposed learning-based QoE classification of video sequences outperforms commonly deployed off-line video analysis techniques and that the QoE-based VAC and RM algorithms outperform standard content-agnostic strategies.**

I. Introduction

Mobile data and video services are taking a key role in everyone's daily life. According to the latest global mobile data traffic forecast in [1], in 2012 the mobile video traffic already exceeded 50% of the total data traffic in the Internet and a further 75% increment of the video traffic is expected by 2017, accounting for over 66% of the total mobile data traffic by the end of the forecast period. Furthermore, widespread heterogeneous high-speed wireless coverage by means of LTE femto-cells and WiFi hotspots will increase the number of users that require mobile access to high quality video services, with dramatic impact on the access network performance in both uplink and downlink. Therefore, mobile operators face the issue of supporting high quality video services with the available network resources.

A possible way to reach this goal is to dynamically adapt the video coding rate to the available transmission resources in order to always optimize the QoE perceived by the final video consumers. As observed in [2], reducing the encoding rate of a video is indeed much less critical in terms of QoE degradation than increasing the packet loss probability or the delivery delay. However, the perceived QoE at a certain encoding rate depends on the video content, e.g., scene and source dynamics and frame-by-frame motion and, therefore, it is not easy to predict.

In this paper, we consider a large set of H.264-AVC [3] video clips coded at different source rates, which correspond to different perceived quality levels. We then assess the quality level of each video in terms of the average Structural SIMilarity (SSIM) index [4]. After a suitable normalization and rescaling of the encoded source rate, we are able to analytically approximate the empirical SSIM-to-bitrate characteristics of each video by means of a polynomial expression, which is then used by QoE-aware VAC and RM algorithms. Unfortunately, this method requires to calculate the SSIM rates for each video and to fit the corresponding polynomial function, which is computationally prohibitive in realistic scenarios. However, we show that the polynomial coefficients can be reliably estimated using a machine learning approach [5]. Crucially, the proposed method does not require to process the original content of the video frames, but only uses network information available after the encoding process, namely the video *frame size*. The rationale is that the SSIM-to-bitrate function of a video is closely related to the dynamics of its content, and this information is reflected in the structure of the corresponding sequence of frame sizes after the encoding [6]. Indeed, the content of a video influences the structure of its compressed version (e.g., highly-dynamic videos, containing complex spatial and temporal structure, will likely result in larger frame sizes). Thus, we build a training dataset containing the frame sizes of the different Group-of-Pictures (GOPs) of the test videos, and upon this dataset we train a Restricted Boltzmann Machine (RBM; [7]) in an unsupervised fashion. The RBM captures the latent features such as input data, thus providing a high-level representation that can be exploited by supervised learning algorithms to estimate the polynomial coefficients that estimate the SSIM-to-bitrate characteristics of unknown videos, which is then used by the aforementioned QoE-aware VAC and RM algorithms.

As a proof of concept, we apply our approach to a simple transmission scenario with a congested link shared by multiple video flows, e.g., a wireless downlink video streaming scenario. We show that, after an off-line learning phase, our approach can run online, performing VAC of unknown videos with basically negligible computational complexity.

To summarize, following [5], we extend the approach in [6] by using a machine learning scheme to estimate the SSIM-to-bitrate characteristics of unknown videos from the distribution of the coded frame sizes, and to use this characterization in QoE-aware VAC and RM algorithms. By means of simulations, we show that combining unsupervised feature extraction and linear classification provides better results than a more basic approach that tries to extract the SSIM characteristics directly from the raw data. Furthermore, we show that QoE-based VAC and RM algorithms make a better use of the available transmission resources than content-agnostic schemes.

The remainder of the paper is organized as follows. In Section II we review the related work. Our video analysis is presented in Section III. The machine learning approach is described in Section IV and validated in Section V. In Section VI we describe the QoE-based and QoE-agnostic resource management algorithms, whose performance is compared by simulations in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

Prior works on video detection over communication networks mainly focus on extracting objective networking and quality metrics. In [8] the authors classify videos based on selected common spatial-temporal audio and visual features described by the MPEG-7 compliant content descriptors. Due to the complexity of the method, the authors make use of the principal component analysis (PCA) to reduce the set of features under study. Nevertheless, this work is strictly dependent on the MPEG-7 multimedia format. Scene detection mechanisms were developed in recent years based on predictive analytical models. In [9], the authors propose a scene-change detector for video-conference traces that works based on the average number of bits generated during the scenes, and is modeled with a two-state Markov chain. The proposed low complexity method comes at the cost of requiring full knowledge of video content to properly set the thresholds for the scene recognition.

Further related works focus on quality prediction models to capture the behavior of video scenes. In [10], an objective model to predict the quality of the lost frames for 3D videos is designed based on the header information of the video packets at different ISO/OSI layers. This model is able to roughly capture the SSIM of some video clips based on the size of the lost frames and via deep packet inspection, which is usually avoided by operators in cellular deployments due to the complexity and national privacy rules. Nevertheless, in [11], the authors claim that the frame loss probability, which is mainly a network metric, provides only limited insight into the video quality perceived by the user. Moreover, the authors state that the rate distortion curves drawn using the Peak Signal-to-Noise Ratio (PSNR) provide a limited representation of the perceived video quality, thus improved quality metrics to better represents videos are needed.

In our work, we analyze and group video test sequences based on the relation between video compression rate and SSIM. It is widely recognized that the SSIM index improves traditional objective QoE metrics like PSNR and mean square error (MSE), which have proven to be inconsistent with the human eye perception. Although the SSIM characterization of a video sequence is computationally expensive, in [6] we showed that it can be compactly represented by means of polynomial curves that can be associated to each video. Tagged videos can then be handled by simple traffic shaping mechanisms in case of network congestion or under-provisioned network resources.

Despite its appeal, a major drawback of this approach is that it requires to tag all the videos with the corresponding polynomial coefficients [6]. Computing the SSIM-rate for each video being transmitted is infeasible even in medium-scale scenarios. An alternative approach is to use automatic methods to support the tagging process [5]. Machine learning algorithms represent the state-of-the-art in many classification tasks, especially when the structure of the domain is difficult to characterize. However, extracting information from visual sequences has proven to be a challenging problem for machine learning algorithms. In the so-called “content-based” video retrieval [12], a range of different techniques can be applied depending on the task of interest, e.g., video indexing, scene recognition and/or classification, object tracking, and motion detection.

The problem of automatic video processing is closely related to that of image recognition, with the additional complexity given by the temporal dimension of the data. In recent years, advances in the theory and practice of probabilistic graphical models and statistical learning led to the development of extremely powerful learning systems, which achieve state-of-the-art performance in several machine vision tasks [13], [14]. Although the main application of these systems has been primarily focused on still frames, there have also been successful extensions to the temporal domain [15]. However, all the above-mentioned machine learning methods are usually applied at the pixel level, or to some higher-level representations obtained after additional pre-processing of the raw images. Nevertheless, for the task of classifying different videos depending on the dynamics of their content, we assume that the relevant information is still preserved after the video has been encoded to be sent on a transmission channel. In this work, which builds on our preliminary position paper in [5], we therefore propose to automatically extract a set of features that can be used to describe the relevant characteristics of the original videos, using only information available at the network level. To our knowledge, this is the first attempt to apply machine learning algorithms on this type of data for such a purpose.

III. Video analysis

For the reader’s convenience, we report here the video analysis framework described in [6]. We evaluate the objective

TABLE I
MAPPING SSIM TO MEAN OPINION SCORE SCALE

SSIM	MOS	Quality	Impairment
≥ 0.99	5	Excellent	Imperceptible
$[0.95, 0.99)$	4	Good	Perceptible but not annoying
$[0.88, 0.95)$	3	Fair	Slightly annoying
$[0.5, 0.88)$	2	Poor	Annoying
< 0.5	1	Bad	Very annoying

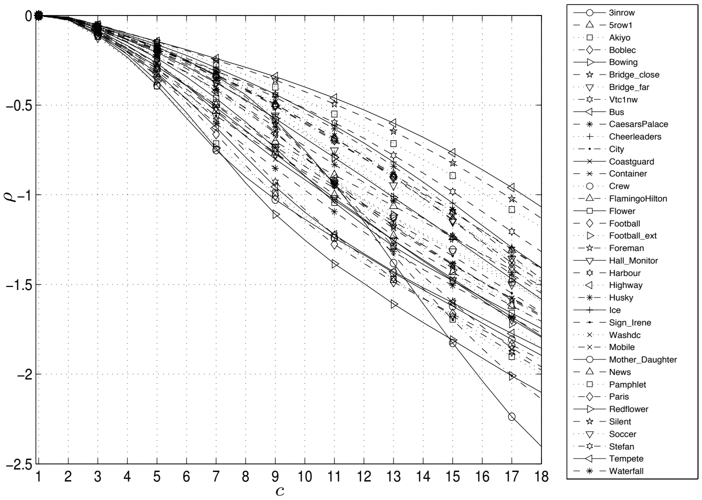


Fig. 1. Logarithm of the normalized rate $\rho_v(c)$ *versus* compression level c for different video clips.

TABLE II
VIDEO TEST SET

Name	Full quality rate [kbit/s]	Duration [s]
3inrow	11856	12
5row1	11135	12
Akiyo	5387	10
Boblec	11504	12
Bowing	10325	10
Bridge_close	18246	66
Bridge_far	18304	70
Vtc1nw	11210	12
Bus	16954	5
CaesarsPalace	17001	12
Cheerleaders	21757	12
City	14139	10
Coastguard	16570	10
Container	12229	10
Crew	16179	10
FlamingoHilton	25622	12
Flower	16335	8
Football	15806	3
Football_ext	18092	12
Foreman	14642	10
Hall_Monitor	16291	10
Harbour	17929	10
Highway	17529	66
Husky	24065	8
Ice	9517	8
Sign_Irene	14091	18
Washdc	12948	12
Mobile	19172	10
Mother_Daughter	11348	10
News	7824	10
Pamphlet	10917	10
Paris	12450	35
Redflower	14168	12
Silent	11586	10
Soccer	14063	10
Stefan	17589	3
Tempete	17850	8
Waterfall	14950	8

QoE of the videos with the SSIM index, which is a full reference metric that measures the image degradation in terms of perceived structural information change, thus leveraging the tight inter-dependence between spatially close pixels which contain the information about the objects in the visual scene [4]. SSIM is calculated via statistical metrics (mean, variance) computed within a square window of size $N \times N$ (typically 8×8), which moves pixel-by-pixel over the entire image. The measure between the corresponding windows X and Y of two images is computed as follows:

$$SSIM(X,Y) = \frac{(2\mu_X \mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)} \tag{1}$$

with μ and σ^2 denoting the mean and variance of the luminance value in the corresponding window, and c_1 and c_2 being variables to stabilize the division with weak denominator (we refer the interested reader to [4] for details).

The range of the SSIM index goes from 0 to 1, which represent the extreme cases of totally different or perfectly identical frames, respectively. Tab. I shows the mapping between SSIM and Mean Opinion Score (MOS) scale, which assesses the subjective perceived video quality on a scale of 5 values, from 1 (bad) to 5 (excellent), as reported in [16].

We consider a pool of $V = 38$ CIF video clips, taken from standard reference sets.[1] Each video has been encoded with the Joint Scalable Video Model (JSVM) reference software [18] into H.264-AVC format at $C = 18$ increasing compression levels (i.e., quantization points), which correspond to as many quality levels. The list of video names, full quality transmit rate and duration are provided in Tab. II. Note that there are no scene transitions inside each video sequence. The SSIM of a frame encoded at compression level c is obtained by comparing the decoded frame with the full quality version of the same frame. For practical reasons, we take the average values of the SSIM index for each video.

We denote by $r_v(c)$ the transmit rate of video $v \in \{1, \ldots, V\}$ encoded at rate $c \in \{1, \ldots, C\}$, with $r_v(1)$ being the maximum (i.e., full quality) rate. To ease the comparison between different video clips, it is convenient to normalize the video rates to the full quality rates. Moreover, following the Weber-Fechner's law that postulates a logarithmic relation between the intensity and the subjective perception of a stimulus, we introduce a logarithmic measure of the normalized rate, here named *Rate Scaling Factor* (RSF) and defined as

$$\rho = \log(r_v(c)/r_v(1)) \,. \tag{2}$$

Fig. 1 shows ρ when varying the compression level c for the different videos. We can see that the same compression level c corresponds to different rates, depending on the content of the videos. In general, given c, the more dynamic the video sequence the larger the RSF ρ. Indeed, dynamic sequences exhibit lower spatial and temporal correlation of consecutive video frames and, hence, are less amenable to compression.

The dynamics of the video content also impact the perceived QoE for a certain RSF value, as clearly shown in Fig. 2 that reports the average SSIM of each video clip when varying ρ_v (markers). We observe that the SSIM characteristics of a video v can be approximated with an n-degree polynomial

[1] Video traces can be found in [17], ftp://132.163.67.115/MM/cif

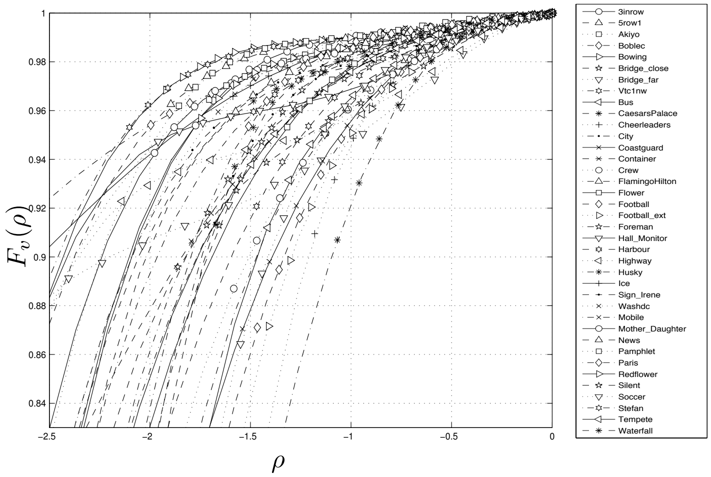


Fig. 2. SSIM of the different video clips when varying the RSF: markers show empirical values, lines are obtained by the 4-degree polynomial approximation $F_v^{(4)}(\rho)$.

expression, which takes the form

$$F_v^{(n)}(\rho) \simeq 1 + a_{v,1}\rho + a_{v,2}\rho^2 + a_{v,3}\rho^3 + \ldots + a_{v,n}\rho^n\,. \quad (3)$$

The vector of coefficients $\mathbf{a}_v = \{a_{v,i}\}$ provides a compact description of the relation between the perceived QoE and the RSF of a video v. It is hence conceivable to tag each video with such a compact representation of its QoE characteristics that can then be used by RM and VAC algorithms, as discussed in the next section. We observe that, in general, a 4-degree polynomial provides a quite accurate approximation of the SSIM values in the range of ρ of practical interest (lines in Fig. 2). Hence, in the following we consider $F_v^{(4)}(\rho)$ as the reference (exact) QoE characteristics of video $v \in \{1, \ldots, V\}$. However, in this paper we will also consider 3-degree and 2-degree polynomial approximations of the SSIM that, while providing a less accurate approximation of the SSIM curve, are likely simpler to be estimated by the machine learning model described in Sec. IV.

IV. Machine Learning approach to video classification

The computation of the exact SSIM characterization of a video sequence is computationally demanding and infeasible in many practical cases. To overcome this problem, following the rationale described in [5], we propose a machine learning approach that provides a fairly accurate estimate of the SSIM characteristics of a video from the *size* of the frames coded in a GOP. As previously mentioned, we postulate that the SSIM characteristics of a video are closely related to the dynamics of its content, and that this information is preserved in the structure of the corresponding sequence of frame sizes after the encoding. However, extracting the SSIM characteristics of a video directly from the raw data, i.e., the frame sizes, is problematic because of the non-linear and hidden interrelations between the two quantities.

The fundamental idea behind our approach is to learn a generative model to capture these non-linearities, providing an alternative representation of the input data that is amenable to classification even by means of linear discrimination methods. This strategy has been shown to be very effective in several machine learning scenarios, and resembles that used by the so-called "kernel methods" like *Support Vector Machines* [19], which first perform a non-linear projection of the data into a (usually higher-dimensional) feature space, and then exploit a linear optimization method to classify the data based on their structural similarities.

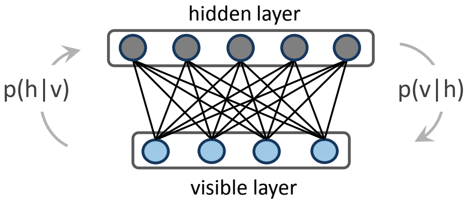


Fig. 3. Graphical representation of a Restricted Boltzmann Machine.

More specifically, our learning framework consists of two main phases. First, *unsupervised learning* is used to extract an abstract representation of the raw data that captures descriptive features of the video. Then, *supervised learning* is performed to create a mapping between the abstract representations and the corresponding SSIM coefficients of the related videos.

A. Unsupervised phase: the Restricted Boltzmann Machine

Our approach relies on a powerful family of generative models, which can be implemented as stochastic recurrent neural networks known as Boltzmann Machines [20]. They can be interpreted as probabilistic graphical models, where connections between units are symmetric, i.e., with equal weight in either direction. The input to the network is given through a layer of visible (i.e., observed) units, which are fully-connected to another layer of hidden units that are used to model the latent features of the data. If there are no connections among units of the same layer, we obtain the so-called Restricted Boltzmann Machine (RBM) [7], which is graphically represented in Fig. 3.

RBMs can be trained in a particularly efficient way, which consists in iterating a positive and a negative phase [21]. During the positive phase, visible units are clamped to the values of the data observed in the training set. The network then propagates activations to hidden units, according to the weights of the connections. If we consider binary units for simplicity, each hidden unit h_j is activated according to the conditional probability

$$p(h_j = 1|v) = \sigma(c_j + \sum_i v_i w_{ij})\,, \quad (4)$$

where σ is the sigmoid logistic function, c_j is the bias term of the hidden unit h_j, and w_{ij} is the weight of its connection with the visible unit v_i. The entire vector of hidden unit activations constitutes an *internal representation* of the pattern observed in the visible units. During the negative phase, instead, hidden units are fixed and activations are propagated backward to the visible units in a similar fashion, in order to accurately *reconstruct* the original input vector. The objective of the learning algorithm is therefore to iteratively adjust the values of the connection weights until the network is able to generate good reconstructions of the input patterns. From a probabilistic standpoint, this corresponds to fitting a generative model to the observed data using a maximum likelihood approach. In

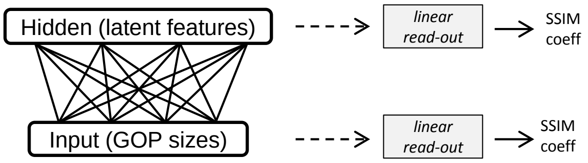


Fig. 4. Scheme of the proposed learning framework, on which unsupervised feature extraction (left) is followed by a supervised linear read-out (right).

practice, this is usually accomplished by performing some form of gradient descent over the likelihood function of the training data. The reader could refer to [22], [23] for more details about learning in RBMs and for the explanation of important additional parameters of the algorithm.

In our case, the training set consists of the vectors of frame sizes in each GOP of the videos in the dataset. Unsupervised learning tunes the RBM model parameters (i.e., the connections weights) with the objective of reproducing the patterns presented in the visible layer, thereby minimizing the reconstruction error. At the beginning, weights are randomly initialized to small values (close to zero) and the reconstructions will be completely wrong. However, the learning process iteratively adapts the weights until the network is able to accurately reproduce the observed patterns. At the end of this unsupervised learning phase, the values taken by the units in the hidden layer provide an alternative and, hopefully, more expressive representation of the input vector, i.e., of a certain sequence of frame sizes in a GOP.

B. Supervised phase: the linear classifier

The estimate of the SSIM coefficients for a GOP is obtained by placing a simple linear classifier on top of the hidden layer of the trained RBM, and performing a supervised classification task. The idea is that some characteristics of the data are not directly visible in the raw input patterns, but can be discovered by the feature extraction process during the unsupervised learning phase. Once the RBM has learned good internal representations of the patterns by modeling their underlying causes, it should be easier to perform a supervised classification task starting from those abstract representations.

We use a simple linear classifier as read out module. The discrimination between the possible classes is therefore performed by exploiting a linear combination of the data features. This choice is motivated by observing that the non-linearities of the data should be captured by the generative model during the unsupervised learning phase, which creates more separable representations that could be easily read out even by a linear method. Within this perspective, accuracy of linear read-out can be considered as a coarse measure of how well the relevant features of the data are explicitly captured by the generative model [22]. Therefore, the use of a linear classifier makes it easier to understand the quality of the internal representations learned by the RBM, because we can directly compare the classification accuracy obtained using the raw input patterns with that obtained from the internal representations of the RBM. A schematic representation of this process is given in Fig. 4.

V. Learning framework performance

In this section we evaluate the performance of the proposed RBM-based learning framework with respect to a linear classifier that acts directly on the raw data, i.e., the frame sizes contained in a GOP.

A. Dataset and learning parameters

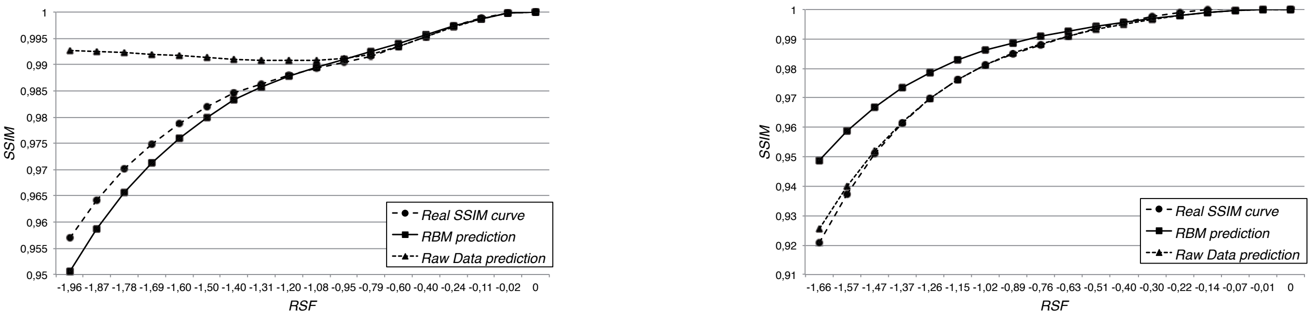
The system is tested on the video dataset described in Section III. In order to make the size of the data uniform, we consider the first 15 GOPs of each video, thereby discarding shorter videos. Thus, we use 34 videos for a total of 510 data patterns, i.e., GOPs. Each GOP is formed by an inter-coded frame (I), followed by 15 predicted frames (P). Due to the limited size of the dataset, we test the performance of the system using a *k-fold cross-validation* technique [24]. To this aim, we partition the dataset into 34 subsets (folds), each including the 15 GOPs of a specific video. The RBM is then trained using 33 folds (training set), and its generalization performance is computed on the left-out fold (test set). This way, 34 different RBMs are trained, each time changing the left-out video to be used as test, and we report the mean estimation accuracy over all the 15 GOPs.

The input to the RBM consists of 32 visible units. Each input vector is obtained by concatenating the sizes of the 16 frames in a GOP, coded with compression levels $c = 1$ (full quality) and $c = 9$ (intermediate quality). The I and P frame sizes of each GOP are normalized between 0 and 1, as this is the usual format of the input patterns used for training neural networks. The size of the hidden layer determines the complexity of the generative model, since the number of free parameters in the model is given by the number of connection weights. We test different layer sizes, with a number of units varying between 50 and 200, and we find that our results are robust with respect to this parameter. We present results for a network with 70 hidden units. We use a publicly available efficient implementation of RBMs that exploits Graphic Processing Units (GPUs) to parallelize the learning algorithm [25]. With the current settings of the machine learning parameters, the learning phase converges after about 50 epochs without exceeding one minute of running time. Regarding the supervised phase, a linear classifier can be implemented as a single layer perceptron, on which iterative learning is performed using the delta-rule. We use an equivalent but computationally more efficient method, which relies on the calculation of a pseudo-inverse matrix and is readily available in some high-level programming languages such as Python or MATLAB [22].

We remark that the unsupervised and supervised learning processes are performed only once. When the RBM and the coupled linear classifier are trained, the estimate of the SSIM coefficients for unknown videos is extremely simple, and can be performed online in negligible time.

B. Coefficients estimation accuracy

We assess whether the internal representation learned by the RBM allowed to estimate the n coefficients of the polynomial



(a) Predicted and real curves for video number 2: RBM prediction shows better precision.

(b) Predicted and real curves for video number 11: raw data prediction shows better precision, but RBM prediction is still acceptable.

Fig. 6. Examples of predicted polynomial curves with respect to ideal curve for two different videos.

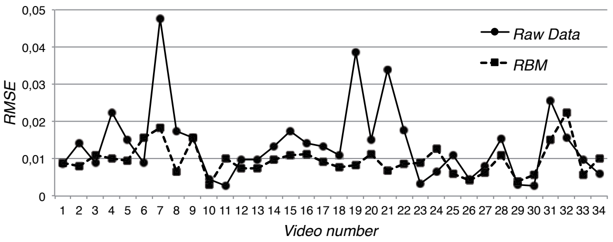


Fig. 5. Coefficients prediction error in terms of mean RMSE between the ideal and predicted curve for each GOPs of the test video.

SSIM function for each GOP in the test set. To evaluate the quality of the estimation, we compute the Root Mean Square Error (RMSE) between the exact SSIM curve, i.e., $F_v^{(4)}(\rho)$, and the curve generated using the n coefficients estimated by the classifier, which we denote by $\tilde{F}_v^{(n)}(\rho)$. In Fig. 5 shows the mean estimation accuracy on the 15 GOPs contained in each of the 34 videos of the test set (dashed line). To better appreciate the performance of the RBM-based learning architecture, we also report the RMSE for the SSIM curves obtained by applying the linear classifier directly on the raw data patterns (solid line). We see that the internal representation learned by the RBM model is indeed capable of capturing critical features of the data, thereby allowing to increase the estimation accuracy for almost all test videos.

Fig. 6 offers a visual comparison between the exact and estimated SSIM curves for two different videos. Fig. 6(a) shows that the curve estimated using the RBM internal representations (solid line) clearly exhibits a better alignment with the exact SSIM curve (dashed line) than the curve obtained directly from raw data (dotted line). Even in the few cases where the RMSE is worse for RBM prediction, as that reported in Fig. 6(b), the RBM estimate of the SSIM curve still remains acceptable.

VI. SSIM-based RM and VAC Algorithms

In this section, we revisit the approach presented in [6], which in this paper is used in conjunction with the learning framework of Sec. V. Given a mechanism to infer the QoE characteristics of a video, we here develop VAC and RM mechanisms that can make use of such information. We consider a framework where different video clips are multiplexed into a shared link of capacity R by a control unit that performs VAC and RM. More specifically, the RM module detects changes of the link capacity (e.g., due to concurrent data flows or fading phenomena in wireless channels) and triggers an optimization procedure that adapts the video rates to maximize a certain utility function. Similarly, the VAC module determines whether or not a new video request can be accepted without decreasing the QoE of any video below a threshold F^* negotiated, for instance, between operator and video consumers. To this end, the VAC invokes the RM module to get the best resource allocation for all the videos potentially admitted into the system and, then, computes the expected SSIM of each video by using (3). If the estimated SSIM is below F^* the last video admission request is refused, otherwise the video is accepted and the rates of the videos in the system are adapted to the new allocation of the transmission resources determined by the RM module.

Formally, let R denote the transmission capacity that needs to be allotted to the videos, and let $\Gamma = \{\gamma_v\}$ be an allocation vector that assigns to the vth video a fraction γ_v of R, with $\gamma_v = 0$ indicating that the video is not accepted into the system. Although the H.264 encoding can only offer a discrete set of transmit rates (see Fig. 1), in the formulation of the optimization problem we assume that video rates can change in a continuous manner. Under this assumption, the RSF of the vth video can be expressed as

$$\tilde{\rho}_v = \log\left(\frac{\gamma_v R}{r_v(1)}\right) . \quad (5)$$

The optimization problem addressed by the RM module can then be defined as follows:

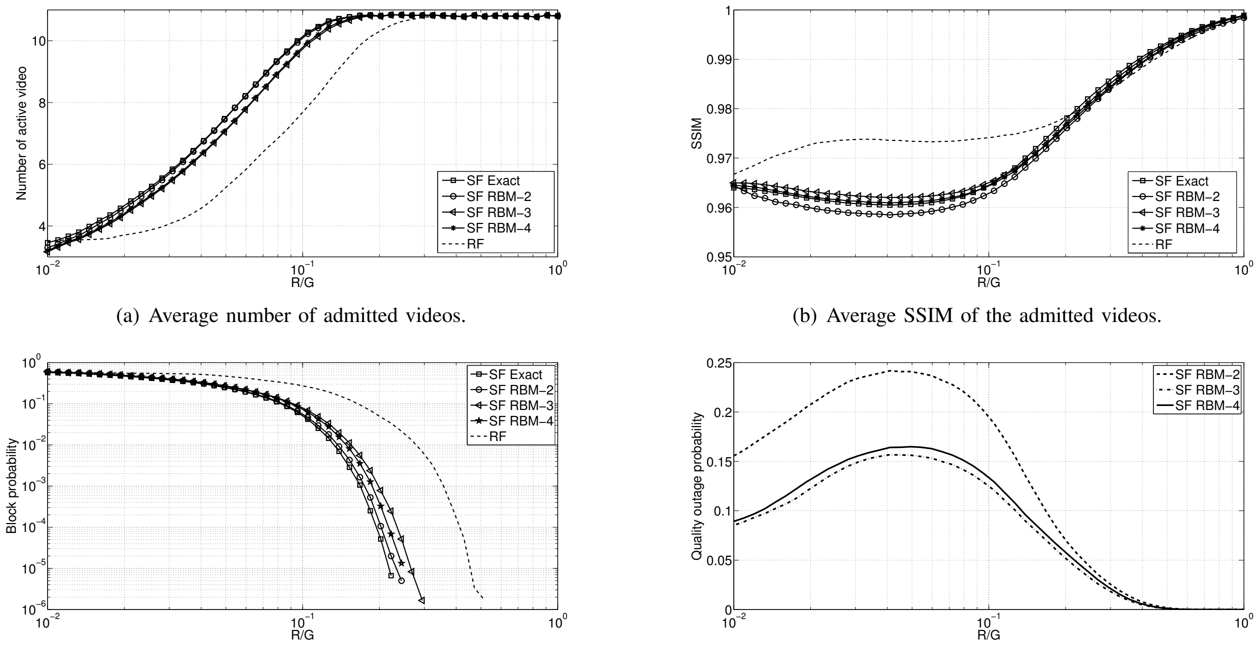
$$\Gamma_{\text{opt}} = \arg\max_{\Gamma} U(\Gamma, R, \{F_v\}) \quad \text{s.t.} \quad \sum_v \gamma_v \le 1 \quad (6)$$

where $\{F_v\}$ denotes the set of SSIM functions of the videos, while $U(\cdot)$ denotes the *utility function* considered by the optimization algorithm. We consider two baseline utility functions that reflect different optimization purposes:

Rate Fairness (RF): Resources are distributed to all active videos proportionally to their full quality rate, without considering the impact on the perceived QoE. In this case, the optimal rate allocation for the ith video is simply given by

$$\gamma_{\text{opt},v} = \frac{r_v(1)}{\sum_j r_j(1)} \quad (7)$$

so that the RSF of each video equals $\tilde{\rho} = \log(R/\sum_j r_j(1))$.



(a) Average number of admitted videos.

(b) Average SSIM of the admitted videos.

(c) Probability that video request is blocked.

(d) Probability that video quality is under threshold (at least once during the video session).

Fig. 7. Performance comparison of our proposed algorithms when varying the channel capacity.

SSIM Fairness (SF): Resources are allocated according to a max-min fairness criterion with respect to the SSIM of the different videos:

$$U(\Gamma, R, \{F_v\}) = \min_v F_v(\tilde{\rho}_v)\,. \tag{8}$$

Note that under the assumption of continuous rate adaptation, the SF criterion yields the same SSIM, say φ, to all active videos. Given this target SSIM, the RSF for each video can be easily found as $\tilde{\rho}_v = F_v^{-1}(\varphi)$, where F_v^{-1} is the inverse of the QoE function F_v (which is monotonic in the range of interest). Therefore, the optimization problem can be easily solved by searching for the maximum φ that satisfies the rate constraint in (6), i.e., such that

$$\frac{1}{R}\sum_v r_v(1)10^{F_v^{-1}(\varphi)} \leq 1\,. \tag{9}$$

VII. Cognitive Video admission control performance

In this section we compare the performance of the VAC and RM algorithms described in Sec. VI by means of simulation.

We consider a scenario where a transmission link is shared among the users, e.g., the wireless downlink channel of a cellular system, and the VAC mechanism allows the admission into the system and the transmission of a video under the constraint that the quality does not fall below a certain SSIM threshold that we set to $F^* = 0.95$, which corresponds to good quality (MOS of 4, see Tab. I).

The video generation process is simulated as a Poisson process with $\lambda = 0.66$ requests/s, where each video request refers to a video randomly picked from the dataset. Denoting by T the average duration of a video sequence, we then have an offered load of $\lambda T \simeq 11$ videos, which corresponds to an aggregate rate request for full video quality of about $G \simeq 161$ Mb/s.

Video requests are processed by the VAC algorithms described in Sec. VI, and resources are allocated accordingly. In particular, we consider four different flavors of the SF algorithm, corresponding to different choices of the SSIM function $F_v(\rho)$, namely:

- *SF-Exact* based on the exact SSIM curve, i.e., $F_v(\rho) = F_v^{(4)}(\rho)$;
- *SF-RBM-n* based on the n-degree polynomial estimation given by the RBM model, i.e., $F_v(\rho) = \tilde{F}_v^{(n)}(\rho)$, with $n \in \{2, 3, 4\}$.

A. Results

We compare the algorithms in terms of: (i) average number of admitted videos, (ii) average SSIM of admitted videos, (iii) blocking probability of a video request, and (iv) quality outage probability, i.e., probability that the quality of an accepted video drops below the minimum threshold F^* during the session. Note that with SF-Exact there is no quality outage, therefore this performance index captures the impact of the SSIM estimate errors of the RBM-based methods.

Fig. 7 shows the performance indices when varying the channel rate R with respect to the nominal average rate request G for full-quality videos. At first glance, we observe that the SF policies always perform better than RF, and accept more videos with above-threshold quality. This confirms that content-aware admission and resource allocation policies are much more effective than traditional content-agnostic policies in a QoE framework. It is interesting to observe in Fig. 7(b) that the average SSIM of the active videos is well above the minimum required quality threshold F^*. The reason is that we considered the actual video rates obtained with the different compression levels, so that resource allocation occurs with a granularity that prevents the "water filling" effect of the channel and leaves part of the capacity unused. This effect is minimized when $R/G \simeq 0.05$. From Fig. 7(d) we also note that the smaller the margin between the mean SSIM and F^*, the larger the quality outage probability of the SF-RBM schemes. This is a consequence of the smaller robustness to the SSIM estimate errors.

For what concerns the SF algorithms, we observe in Fig. 7(a) that, on average, the SF-RBM polynomial approximations perform quite closely to the SF-Exact scheme. Hence, the RBM-based prediction is nearly-optimal and proves the goodness of the training phase. A closer look at the results reveals that SF-RBM-2 is slightly looser than the other SF schemes in the admission process, allowing a moderately larger number of videos in the system, with a little lower average SSIM, as shown in Fig. 7(b). From Fig. 7(d), however, we note that the 2-degree approximation exhibits the largest quality outage probability, which negatively impacts the system performance due to the aforementioned nearly-optimal number of admitted videos. Conversely, the SF-RMB-3 and SF-RMB-4 schemes perform in a comparable manner, with a very small advantage of SF-RBM-3 over SF-RBM-4 in terms of quality outage probability. Thus, we might suggest the use of 3-degree predictions due to the slightly lower computational complexity and amount of signaling required in the system.

VIII. Conclusions and future directions

We designed a framework for video admission control in wireless systems that exploits machine learning algorithms to optimize resources management. By means of simulation, we showed that our proposal outperforms offline video analysis techniques in terms of the trade-off between QoE delivered and computational costs.

One promising future direction to further improve the proposed method could be to extend the unsupervised learning phase by using a deeper architecture, thereby considering a hierarchical generative model of the data distribution [13]. However, more complex models usually need larger training datasets, which must provide enough statistical information to extract a good set of descriptive features. An important step would therefore be to also increase the amount of data used to train the generative model, which can be accomplished by collecting more videos or integrating other available datasets into the framework. Finally, exploiting unsupervised learning to build an expressive set of high-level features allows great flexibility to the proposed framework, which can be used to transfer knowledge across several tasks [26].

IX. Acknowledgement

This work was partially supported by the U.S. Army Research Office, Grant No. W911NF-11-1-0538, and by the project "A Novel Approach to Wireless Networking based on Cognitive Science and Distributed Intelligence," funded by Fondazione CaRiPaRo under the framework *Progetto di Eccellenza 2012*.

References

[1] CISCO, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017*. White Paper, Feb. 2013.

[2] N. Amram, B. Fu, G. Kunzmann, T. Melia, D. Munaretto, S. Randriamasy, B. Sayadi, J. Widmer, and M. Zorzi, "QoE-based transport optimization for video delivery over next generation cellular networks," in *IEEE ISCC*. IEEE, 2011, pp. 19–24.

[3] "Advanced Video Coding for Generic Audiovisual Services," *ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC*.

[4] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, pp. 600 – 612, Apr. 2004.

[5] L. Badia, A. Testolin, A. Zanella, M. Zorzi, and M. Zorzi, "Cognition-based networks: applying cognitive science to wireless networking," in *Video Everywhere (VidEv) Workshop of IEEE WoWMoM*, Sidney, Australia, June 2014.

[6] M. Zanforlin, D. Munaretto, A. Zanella, and M. Zorzi, "SSIM-based video admission control and resource allocation algorithms," in *WiVid Workshop of IEEE WiOpt*, Hammamet, Tunisia, May 2014.

[7] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel distributed processing: Explorations on the microstructure of cognition. Volume 1: Foundations*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986.

[8] L.-Q. Xu and Y. Li, "Video classification using spatial-temporal features and PCA," in *IEEE ICME*, Baltimore, MD, July 2003.

[9] I. Spanou, A. Lazaris, and P. Koutsakis, "Scene change detection-based discrete autoregressive modeling for MPEG-4 video traffic," in *IEEE ICC 2013*, Budapest, Hungary, June 2013.

[10] B. Feitor, P. Assuncao, J. Soares, L. Cruz, and R. Marinheiro, "Objective quality prediction model for lost frames in 3D video over TS," in *IEEE ICC 2013*, Budapest, Hungary, June 2013.

[11] P. Seeling, M. Reisslein, and B. Kulapala, "Network performance evaluation using frame size and quality traces of single-layer and two-layer video: a tutorial," *IEEE Communications Surveys and Tutorials*, vol. 6, pp. 58 – 78, Oct-Dec 2004.

[12] S. Antani, R. Kasturi, and R. Jain, "A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video," *Pattern recognition*, vol. 35, no. 4, pp. 945–965, 2002.

[13] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 24, 2012, pp. 1–9.

[15] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Modeling human motion using binary latent variables," *Advances in neural information processing systems*, vol. 19, p. 1345, 2007.

[16] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld, "Impact of frame rate and resolution on objective QoE metrics," in *Workshop on Quality of Multimedia Experience (QoMEX)*, Trondheim, Norway, June 2010.

[17] "Test media repository." [Online]. Available: http://media.xiph.org/video/derf/

[18] "Joint scalable video model - reference software." [Online]. Available: http://ip.hhi.de/imagecom_G1/savce/downloads/SVC-Reference-Software.htm

[19] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[20] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.

[21] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[22] M. Zorzi, A. Testolin, and I. P. Stoianov, "Modeling language and cognition with deep unsupervised learning: a tutorial overview," *Frontiers in Psychology*, vol. 4, 2013.

[23] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," in *Neural Networks: Tricks of the Trade*. Springer, 2012, pp. 599–619.

[24] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, vol. 14, no. 2, 1995, pp. 1137–1145.

[25] A. Testolin, I. Stoianov, M. De Filippo De Grazia, and M. Zorzi, "Deep unsupervised learning on a desktop PC: a primer for cognitive scientists," *Frontiers in Psychology*, vol. 4, 2013.

[26] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning." *Journal of Machine Learning Research - Proceedings Track*, vol. 27, pp. 17–36, 2012.