2020 16th International Conference on the Design of Reliable Communication Networks (DRCN) 978-1-7281-6300-0/20/$31.00 ©2020 IEEE 10.1109/DRCN48652.2020.1570609737

A Unified View to Machine Learning and Control for Measurement-based Equivalent Bandwidth

Maurizio Mongelli, Marco Muselli*, Mario Marchese†,
*Institute of Electronics, Computer and Telecommunication Engineering
National Research Council of Italy, Genova, 16134, Italy
Email: maurizio.mongelli, marco.muselli }@ieiit.cnr.it
†Department of Electrical, Electronic and Telecommunication and Naval Engineering,
University of Genova, Genova, 16143, Italy
Email: mario.marchese@unige.it

***Abstract*—This paper outlines a unified view of machine learning and control for the optimization of a communication system. The problem of equivalent bandwidth is taken as a reference. A dedicated classification technique is used to derive insights into the structure of the problem by means of boolean rules over the variables of the system. The approach is of particular interest for many settings in which only measurements of the performance are available. Simulations corroborate the quality of the proposed technique.**

I. Introduction

Classification techniques developed in machine learning research are often used in many communication disciplines to assess the influence of system variables when analytical models can be hardly derived. The acquired knowledge can then be used to drive adaptive control of performance metrics. Quite a few works follow this principle (see, e.g., [1], [2], [3]), without entering into the details of the generalization capabilities of the adopted approach.

In this letter, classification and control are formulated in a unified way. The focus is on the concept of *Equivalent Bandwidth* (EqB), namely, the satisfaction of *Quality of Service* (QoS) for a set of traffic connections in a high-speed network, but the inherent generalization to other communication settings (overlay or ad-hoc networks) is straightforward because the methodology is only based on performance samples available from the system.

The choice of formulating the control problem for EqB has a motivation on its own. Analytical models are not available for some QoS metrics (loss, delay, jitter) when the stochastic behavior of input traffic does not allow closed-form expressions (e.g., under the Poisson assumption of packets interarrival time). In those cases, measurement-based algorithms are in general adopted [4], [5], but their application is not always immediate because they may need an accurate tuning of the algorithms (see, e.g., the adoption of the Dominant Time Scale principle of [5] or the setting of the gradient stepsize in [4] or of the ‘proportional-integral-derivative controller’ parameters in [6]).

The proposed approach guarantees a high degree of accuracy, even if it is based on a simple incremental control paradigm, whose setting of the parameters does not require critical insight. It consists of exploiting a classification problem to infer the next step in time of the control variable, on the basis of the measurable quantities of the system. The term *incremental* stems from the fact that at every step a small perturbation of the current control signal is generated.

II. The problem

The problem of service rate dimensioning of a finite traffic buffer is considered. The stochastic input rate process of the buffer is $\alpha(t)$; no specific assumptions are made for it. The service rate of the buffer is denoted by $\theta(t)$. The overall optimization objective is to find $\theta^*(t)$ so that the following functional cost is minimized:

$$\int_0^T \Delta(l(t), l^*(t))\ dt \tag{1}$$

where $l(t)$ is the chosen performance index measured at time t, $l(t) = l(\alpha(t), \theta(t))$, $l^*(t)$ is the performance target and $\Delta(\cdot)$ is a function measuring the distance of $l(t)$ from $l^*(t)$, e.g., $\Delta(l(t), l^*(t)) = (l(t) - l^*(t))^2$.

Since $\alpha(t)$ is not known a-priori, a sequence of reallocation steps $\theta(k), k = 1, 2, ...$ is defined, on the basis of feedback acquired during the system evolution. The feedback law $f(\cdot)$ decides the reallocation at time k, $\theta(k) = f(\theta(k-1), I(k))$, as a function of an information vector $I(k)$ collecting observations of some features of interest acquired during the system evolution up to instant k.

The components of $I(k)$ must concern quantities correlated with the performance, such as indications about the statistical properties of $\alpha(t)$ (e.g., number of active sources, burstiness) or, more simply (and often less effectively), its mean and variance. For instance, $I(k)$ may assume the following form:

$$\begin{aligned} I(k) = [\ & l(k), N(k), Bp(k), \overline{\tau}(k), \\ & \overline{\phi}(k), m(k), \sigma(k), B(k), B_{Max}(k)\ ] \end{aligned} \tag{2}$$

where $l(k)$ is the measured performance (averaged over the $[k-1, k]$ horizon), $N$is the number of active traffic sources giving origin to α, Bp, $\overline{\tau}$ and $\overline{\phi}$ are the peak bandwidth, the average burst size and the average silence duration of the sources, respectively, m and σ are the average and standard deviation of α, B and B_{Max} are the current and maximum buffer size, respectively. The presence of $\overline{\tau}$ and $\overline{\phi}$ constitutes a certainty equivalent assumption concerning the presence of on-off traffic sources.

In addition, any other parameter involved in the stochastic behavior of the sources can be taken into account (e.g., some video sources are modeled by multiple active states with constant bit rate generation). However, some of the features involved in a precise description of the sources may be or may be not exploited in the information vector and this has an impact on the control performance, as evidenced later.

The problem is thus to find the optimal sequence $\theta^*(k)$ of bandwidth reallocations over consecutive discrete time instants $k = 1, 2, ...$ In a generic setting when many performance metrics may be of interest (loss, delay, jitter of the packets) and no specific statistical properties of $\alpha(t)$ is assumed, the functional cost can be only estimated by means of measurements on the system. This holds true even more for the end-to-end QoS on a network [7].

III. Classification and Control

A feedback law of the form:

$$\theta(k) = f(\theta(k-1), I(k)), k = 1, 2, ...; \tag{3}$$

$$f(\theta(k-1), I(k)) = (1 + \delta \cdot r(I(k))) \cdot \theta(k-1) \tag{4}$$

is considered, being δ a pre-defined quantity allowing changes in θ between consecutive reallocation instants (e.g., of 5%), $I(k)$ the above defined information vector, and $r(I(k))$ a function called δ-mapping assuming values in the set $\{-1, 0, +1\}$. This function chooses at each time instant k the best action to be performed: in particular, if $r(I(k)) = -1$ the value of $\theta(k)$ will be decreased of $\delta \cdot \theta(k-1)$ with respect to $\theta(k-1)$; if $r(I(k)) = +1$, θ will be increased of the same quantity, whereas it will remain unchanged when $r(I(k)) = 0$. The approach described by (4) will be called *incremental control* (IC).

The δ-mapping can be derived through the solution of a standard classification problem. A training set $\{(I^f, o^f), f = 1, ..., F\}$ is collected by analyzing the behavior of the system in presence of several realizations I^f of the information vector; in every sample o^f is the choice for $r(I^f)$ which leads to the best change in the considered performance. Then, a classification algorithm is used to induce from the training set the value of the δ-mapping in the whole input domain. The proposed approach reveals to be enough robust even if no certainty equivalent assumption are made on the traffic sources, namely, if only basic stochastic indicators, such as mean and variance of α, are used.

IV. Remarks

A. Training set

The choice of the value for o^f is derived by measurement-based inspection on the system, e.g., through simulation analysis or derived from a data set of measures available from previous working periods of the system. In the former case, a random generation of samples of the features in the information vector, under a uniform distribution, may help the generalization capabilities of the classifier; on the other hand, the generation of the training set under a known control heuristic reduces the size of the training set and the computational complexity involved by the inherent classification problem.

B. δ parameter

The δ parameter is set to a small value in order to avoid excessive control oscillations. One may therefore argue that it cannot lead to fast reactions to system changes. In this perspective, it is worth noting that reallocations in (3) can be applied more than once, e.g, m times, between two subsequent updates of the information vector $I(k)$. It is sufficient to substitute (3) with the following equation:

$$\theta(k) = f^m(\theta(k-1), I(k)), k = 1, 2, ...; \tag{5}$$

being f^m, with $m \geq 1$, the composition of m functions $f(\cdot, \cdot)$ having the same information vector $I(k)$, i.e.

$$\begin{aligned} &f^1(\theta(k-1), I(k)) = f(\theta(k-1), I(k)) \\ &f^2(\theta(k-1), I(k)) = f(f(\theta(k-1), I(k)), I(k)) \\ &f^3(\theta(k-1), I(k)) = f(f^2(\theta(k-1), I(k)), I(k)) \\ &\ldots \end{aligned}$$

If reliable simulative or analytical models of the system exist, the application of (5) may occur by following the *receding horizon* approach, i.e., by evaluating the effect $f^m(\theta(k-1), I(k))$ of m consecutive reallocations of θ before applying them on the real system through equation (5).

C. Other approaches

Measurement-based EqB can be addressed in different ways. On the one hand, traditional EqB closed-form expressions can be applied under statistically homogeneous traffic trunks and some QoS constraints (e.g., loss) [5]. On the other hand, other metrics (e.g., delay) and heterogeneous conditions may require the adoption of numerical approximations. Among them, approaches driven by *Perturbation Analysis* (PA) may be of great interest. In on-line gradient descents driven by PA [4], an accurate tuning of the gradient stepsize is needed as a trade off between fast reactions to traffic changes and convergence [8]. The extended-Ritz method [9] may overcome this drawback, by mapping the solutions found by PA on a neural network, which then applies the suboptimal control on line, almost instantly. However, the approach requires an off-line training phase, which consists of solving a regression problem, whose functional cost may suffer from the presence of local minima [10]. This drawback has been recently experienced on Gaussian sensor networks [11], for which non-linear coding-decoding strategies are known to be optimal, but are difficult to be found via numerical approximations [12].

The same concept may hold for *Reinforcement Learning* (RL), which consists in approximating dynamic programming cost-to-go functions through a regression algorithm. Many regression algorithms have been investigated to this aim and

RL has been extensively adopted in several communication fields. However, the inherent regression problem may suffer of numerical instability as for the extended-Ritz case.

D. Advantages

In this perspective, the main advantage of the IC approach relies on the adoption of a classification problem for determining the correct action to be performed at any instant. In fact, the amount of information needed to solve a classification problem is considerably lower than that required in a regression problem [13]. For this reason, to avoid overfitting many approximating structure of RL are chosen as linear functions, even if non-linear ones should be, in theory, more appropriate [14]. Furthermore, categorical variables (for which an ordering relationship among its values is not defined) can be included in a classification analysis, whereas this is not possible when regression estimate is searched for. It is also remarkable that the decision about which of the 3-classes ($\{-1, 0, +1\}$) is the best choice at a given instant is generally less sensitive to noise with respect to the determination of a precise value for the control function θ.

Another advantage of IC derives from feature selection capabilities provided by some classification techniques, which can automatically offer insight into the structure of the problem at hand (which parameters are of main interest for inferring control and in what conditions). In particular, rule generation (RG) techniques that develop classifiers described by a set of intelligible rules allow to understand the conditions involved in a specific decision about the control action to be performed. Recent encouraging results have been obtained in several applicative fields by employing Switching Neural Networks (SNN) [15] trained via Shadow Clustering algorithm [16]. Intensive simulations have shown that intelligible models produced by SNN present a better generalization ability with respect to those provided by other RG approaches, such as decision trees (DT) [17].

E. Drawbacks

The main source of error for the IC comes from possible noise of the training set, when deriving from performance data collected on a history database of the system. In that case, however, the classification algorithm returns clear indications about inadequacy of the available information (poor classification performance or very complex ruleset).

V. Performance Evaluation and Discussion

A. System setting

On-off traffic is considered with respect to the *packet loss probability* (PLP) metric. The PLP target is 10^{-2}. Each source is an on-off process with time durations exponentially distributed on ($\overline{\tau}$) and off ($\overline{\phi}$). For example, according to ITU-T P.59, VoIP on-off periods amount to 1.008 s and 1.587 s, respectively, whereas VoIP peak bandwidth of a single source is in $[5.25, 64]$ kbps, depending on the codec. Consequently, $\overline{\tau}$ has been set to 1.0s whereas $\overline{\phi}$ has been considered as a variable. Traffic enters an IP buffer whose length and service rate (set by the traffic peak bandwidth) guarantees no packet loss rate. The resulting stream is then encapsulated over DVB, thus generating the process $\alpha(t)$, which then enters a DVB buffer, whose size is variable (the size is in DVB cells, of 188 bytes each). All the other parameters of the system are variable as well: peak bandwidth $Bp \in [5, 50]$ kbps, number of connections $N \in [70, 120]$, buffer size $B_{Max} \in [5, 500]$ and average silence periods $\overline{\phi} \in [0, 5]$.

B. Perturbation Analysis used for comparison

The following technique, called *Reference Chaser Bandwidth Controller* (RCBC) [1], is used for performance comparison with IC:

$$\theta(k) = \theta(k-1) + \eta_k \cdot \left.\frac{\partial \Delta(\theta)}{\partial \theta}\right|_{\theta(k-1)}$$

It consists of a gradient descent approach, whose gradient is approximated through PA and whose stepsize η_k is empirically tuned in order to optimize performance. The choice of RCBC stems from the fact that it is more efficient than other algorithms, such as PID or regular EqB [4], [8], [18]. The setting of η_k here follows [8]. The notation RCBC$_{fs}$ refers to the adoption of RCBC with a fixed stepsize set to s ($\eta_k = s, \forall k$); RCBC$_\nu$ defines the adoption of the Vogl method (whose tunable parameter is ν) to let the stepsize be adaptive to the current value of the functional cost [19]: $\eta_k = \nu \cdot |l(k) - l^*(k)|$. According to [18], the bandwidth in correspondence of three consecutive times with PLP zero is decreased of 15%; this leads to counteract the error introduced by PA when it approximates with zero the gradient of the cost function when the loss is zero, thus avoiding long situations with overprovisioned bandwidth.

C. Simulation

The mentioned system is simulated using an ad-hoc C++ simulator implementing RCBC and IC to set the service rate of the DVB buffer and measuring the achieved PLP over time. Simulation time is 5 hours. Every 15m, a new system setting is chosen by following a uniform distribution according to the mentioned ranges (the same realization for each variable is applied under each different usage of RCBC or IC). Reallocations are performed every minute, unless otherwise stated. The VoIP loss rate in VoIP packets (of 80 bytes each) is measured every minute. The following figures show the behavior of considered algorithms; each point corresponds to the achieved PLP with the inherent bandwidth allocation, sampled every minute. For the sake of picture clarity, points in Fig. 1 and Fig. 3 correspond to averages of 20 points of PLP, measured every 60s. Quantitative metrics of Tab. I, such as the average and variance of PLP and of the bandwidth over the simulation period, may help the interpretation of the qualitative behavior inferred from the figures. These quantities are identified in Tab. I through the notations $\bar{l}$, σ_l^2, $\overline{\theta}$ and σ_θ^2. Also the percentage of the periods where PLP is over threshold (l^*_{over}) and the average difference between measured PLP and the target ($|\bar{l} - l^*|$) are considered.

D. RCBC

Fig. 1 and Fig. 2 show the performance and bandwidth allocation using RCBC (the horizontal line of Fig. 1 highlights the target). RCBC gradient descent is initialized by the VoIP average bandwidth of 70 sources, multiplied by the percentage overhead of DVB. As also clear from Tab. I, RCBC is not always able to guarantee the target. The loss peaks are due to sudden reductions of B_{Max} or sudden increase of Bp; this, in turn, leads to large bandwidth increments (quite evident for RCBC$_{0.5}$). More complicated settings of ν may result in better performance, but this would reveal that RCBC is very sensitive to the high variable conditions of the system. In other words, finding the best mapping among ν and the actual system setting to guarantee the given PLP target is a hard task.

E. Incremental control

The δ-mapping for the problem under investigation is derived as follows. The δ quantity is set to 15% for both training and control. The points of the training set are derived from consecutive simulations under quasi-random extractions of the variable system parameters, according to a Sobol sequence. One simulation step for training (SST$_f$) corresponds to a point (I^f, o^f) of the training set. The duration of each SST is 4 minutes. The first minute is considered as the transient period to achieve steady state. The second minute allows collecting the samples of the information vector (I^f), the last 2 minutes are used to collect the PLP. The bandwidth is initialized with: $\theta = N \cdot B_p \cdot b$, being $b = (\overline{\tau} + \overline{\phi})/\overline{\tau}$ the sources' *burstiness* (giving indication about how much a source is "bursty", i.e., generates packets more irregularly over time). This constitutes the first collection of the PLP (under $o^f = 0$). Then, two replications with the same realizations of stochastic processes are repeated while increasing $(o^f = 1)$ and decreasing $(o^f = -1)$ the value of θ; o^f is then chosen in correspondence of the measured PLP closest to the target (among the three replications of SST$_f$).

The training set is built in around 2 hours over an Intel Q6600 2.4Ghz CPU; in virtue of the independence of SSTs, this processing time can be significantly reduced using multiple CPUs. A *Logic Learning Machine* (LLM) implementing the paradigm of the SNN (subsection IV.E) is then used to solve the inherent classification problem by using the *Rulex* software [20]. F=10^4 is chosen to guarantee 95% of classification accuracy over a test set of 5000 points (only half of the training set is used to train the LLM); it is worth noting that the same accuracy is achieved for the training set under investigation with a reduced size, roughly speaking, of about 3000 points. The execution time needed to train LLM is around 15 minutes. Intelligible rules generated by LLM to drive control are presented and discussed in the following subsection. Two different cases are considered with respect to two possible settings of the information vector: in the first one (denoted in the figures with the subscript "...all"), all the quantities in (4) are used for determining the output. In the second case (denoted in the figures with the subscript "...reduced"), only the quantities corresponding to no certainty

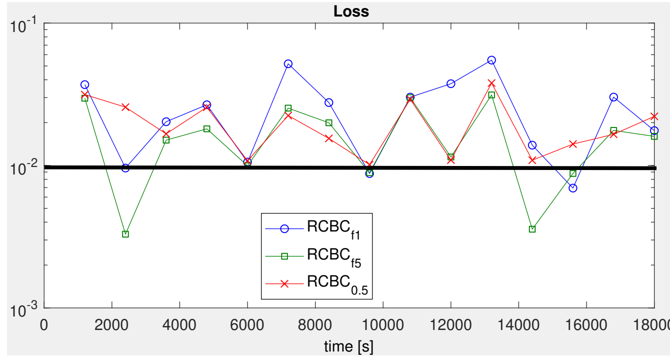

Fig. 1. RCBC loss over time.

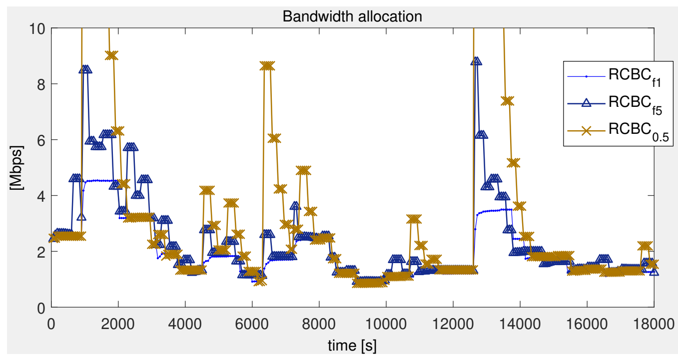

Fig. 2. RCBC bandwidth allocation over time.

equivalent assumptions on the sources are used; consequently, the variables N, Bp, $\overline{\tau}$, $\overline{\phi}$ are ignored when deriving the rules.

Fig. 3 and Fig. 4 show the performance and bandwidth allocation obtained by using IC in the same simulation setting of RCBC (the same realizations are used). In the case of the reduced information vector, the IC is applied with a smaller time granularity than the regular horizon used to collect samples of the information vector and of PLP (of 1 minute), as explained in subsection IV.C: twice times a minute and five times a minute (denoted in the figures with "...*2..." and "...*5...", respectively). The achieved performance is satisfying in all the cases as also confirmed by Tab. I, except for IC reallocations every 1 minute with reduced $I(\cdot)$. Using $I(\cdot)$ with all the features of (4) guarantees an average PLP below the target with a well balanced margin; on the other hand, the average PLP with faster IC and under the reduced $I(\cdot)$ converges to the target.

F. Rule generation

Similar results can be obtained by adopting for IC other classification techniques, such as standard neural networks (this has been validated by other results, not shown here). However, LLM allows inferring some insights into the structure of the problem by means of RG. After classification, the ruleset is simplified by removing all the rules whose covering (i.e. the fraction of samples in the training set that verify it) is less than 15%.

TABLE I
AVERAGE PERFORMANCE

	$\bar{l}$	σ_l^2	l^*_{over} [%]	$\lvert\bar{l}-l^*\rvert$	$\bar{\theta}$ [Mbps]	σ_θ^2
RCBC$_{f1}$	**2.56**$\cdot10^{-2}$	**5.80**$\cdot10^{-2}$	**56**	**1.86**$\cdot10^{-2}$	**1.92**	**0.96**
RCBC$_{f5}$	**2.00**$\cdot10^{-2}$	**5.70**$\cdot10^{-2}$	**28**	**1.61**$\cdot10^{-2}$	**2.22**	**1.43**
RCBC$_{0.5}$	**1.66**$\cdot10^{-2}$	**5.53**$\cdot10^{-2}$	**27**	**1.29**$\cdot10^{-2}$	**4.10**	**6.83**
IC$_{all}$	**5.60**$\cdot10^{-3}$	**3.85**$\cdot10^{-2}$	**3**	**5.28**$\cdot10^{-3}$	**3.63**	**1.06**
IC$_{reduced}$	**1.76**$\cdot10^{-2}$	**6.92**$\cdot10^{-2}$	**14**	**1.50**$\cdot10^{-2}$	**2.13**	**1.11**
IC*2$_{reduced}$	**1.10**$\cdot10^{-2}$	**5.32**$\cdot10^{-2}$	**12**	**8.84**$\cdot10^{-3}$	**2.36**	**1.52**
IC*5$_{reduced}$	**1.00**$\cdot10^{-2}$	**4.13**$\cdot10^{-2}$	**19**	**6.89**$\cdot10^{-3}$	**3.40**	**3.88**

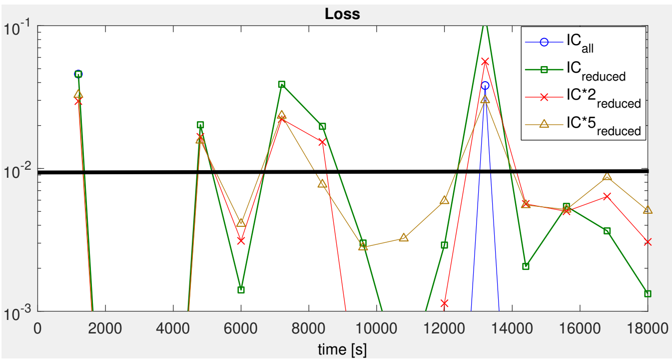

Fig. 3. Incremental Control (IC) loss over time.

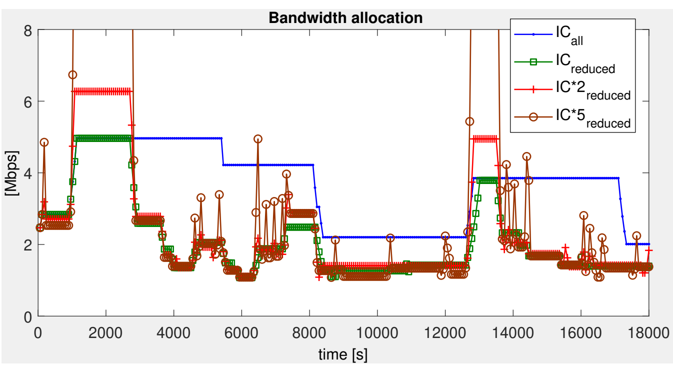

Fig. 4. Incremental Control (IC) bandwidth allocation over time.

When LLM employs the whole set of input variables to build the model, the following rules are generated (Bp is expressed in kbps, m and θ in Mbps):

if $((l > 2\cdot10^{-2}) \wedge (\overline{\phi} \leq 4.44))$ $r =$ **1**;

if $((m > 0.57) \wedge (l > 2\cdot10^{-2}) \wedge (B \leq 138))$ $r =$ **1**;

if $((m > 0.57) \wedge (l > 2\cdot10^{-2}) \wedge (B_{Max} > 251) \wedge (Bp \leq 44.4))$ $r =$ **1**;

if $((N > 84) \wedge (\sigma \leq 0.13) \wedge (l > 2\cdot10^{-2}))$ $r =$ **1**;

if $((N \leq 90) \wedge (l > 2\cdot10^{-2}) \wedge (B > 77))$ $r =$ **1**;

if $((\sigma \leq 0.06) \wedge (l > 2\cdot10^{-2}))$ $r =$ **1**;

....

if $((l \leq 5\cdot10^{-4}) \wedge (\theta > 2.7) \wedge (B \leq 266) \wedge (\overline{\phi} > 3.4))$ $r =$ **-1**;

if $((N > 87) \wedge (l \leq 10^{-3}) \wedge (\theta > 1.7) \wedge (B_{Max} > 69) \wedge (\overline{\phi} > 3.1) \wedge (Bp \leq 40.1))$ $r =$ **-1**;

It is interesting to note that condition $l > 2\%$ to drive bandwidth increase has been inferred by LLM without any explicit indication about the PLP target (1%). Recognizing the value of 4.44 as an important threshold on $\overline{\phi}$ (first rule) is another interesting outcome; the same concept holds true for $m > 0.57$. The covering of the first two rules is 94% and 62%, respectively. Rule intelligibility helps appreciate the influence of many variables (see, in particular the first rules). The most relevant features are (in order): l, θ, $\overline{\phi}$ and Bp, whereas the other variables have much less importance in inferring the right control action. Surprisingly, any further knowledge on N or σ is not so helpful as expected.

The bandwidth decrease is driven by the last rules, characterized by the 15% of covering, and, surprisingly, no rule is present to state bandwidth equilibrium ($r = 0$, i.e., leave the bandwidth as it is). The rationale behind these outcomes relies on the training configuration when θ is initialized with the average rate of α. As a consequence of this initialization, over 5000 points give indication about $r = 1$, 2000 points are relative to $r = -1$ and less than 800 points correspond to the $r = 0$ case. However, this does not preclude both classification accuracy and reliable RG; as evidenced by Fig. 4, an appropriate balance between bandwidth increase and decrease is guaranteed, even if no enough points with $r = -1$, and $r = 0$, in particular, are available. It is also worth noting that some $r = 0$ points are misclassified with the $r = 1$ prediction, thus giving robustness to IC (misclassification with $r = -1$ would lead to wrong bandwidth decreases with a consequent detrimental effect on the PLP). In other words, posing the problem under the δ-mapping framework leads to robust IC, even if the performance database available for training is characterized by a lack of precision (missing data, polarization on some classes).

The ruleset for the reduced information vector leads to:

if $((l > 2\cdot10^{-2}) \wedge (\theta \leq 1.63))$ $r =$ **1**;

if $((\sigma \leq 0.17) \wedge (l > 2\cdot10^{-2}) \wedge (B_{Max} > 164))$ $r =$ **1**;

if $((l > 2\cdot10^{-2}) \wedge (B > 83) \wedge (B_{Max} \leq 455))$ $r =$ **1**;

if $((m \leq 2.0) \wedge (l > 2\cdot10^{-2}) \wedge (B_{Max} \leq 455))$ $r =$ **1**;

....

if $((m \leq 2.0) \wedge (\sigma \leq 0.21) \wedge (l \leq 2\cdot10^{-4}) \wedge (\theta > 2.2) \wedge (B_{Max} > 164))$ $r =$ **-1**;

if $((m \leq 1.7) \wedge (l = 0) \wedge (\theta > 2.2))$ $r =$ **-1**;

if $((m \leq 2.4) \wedge (l = 0) \wedge (\theta > 2.75) \wedge (B > 30))$ $r =$ **-1**;

Four rules of bandwidth increase ($r = 1$) with less coverage are not reported for the sake of synthesis. Similar comments can be provided, with emphasis on the more intricate adoption of m, l and B_{Max} to guarantee the appropriate $r = 1$ predictions, being those the variable with the higher relevance in the resulting feature ranking. Again, the simplicity of the first rule, whose covering is 58%, is outstanding. The adoption of σ in the second rule is counterintuitive, but its covering is 57%, thus corroborating the importance of RG in support of human intuition. Some more words are necessary for the $r = -1$ predictions. The last three rules, whose covering is 20%, are intuitive (apart from the adoption of σ in one case only), but the resulting calculation of the most appropriate thresholds for all the variables involved is an important result. Overall, it is interesting to notice the reliability of these rules, despite the used features do not correspond to any certainty equivalent assumption of the sources, and their power of synthesis even in the presence of the large range of stochastic behaviors considered.

VI. Related Literature

The largest part of the applications of classification algorithms to communication systems relies on behavioral classification (see [21], for a recent example). Nevertheless, driving control on the basis of machine learning inference is intuitive and has been applied in some cases. Reinforcement learning has been exploited for bandwidth control in DiffServ networks and it is still under study, see, e.g., [1]. Self-learning optimization of the *Transport Control Protocol* (TCP) has been always considered a formidable problem in virtue of the complexity of the protocol rules, which can be hardly mapped onto a mathematical model (see, e.g., appendix A of [6]). The authors of [2] show how classification can drive differentiation of TCP losses in optical burst switching either for congestion or contention and how to slightly modify TCP' rules on the basis of this knowledge. Knowledge-based self-configuration of DiffServ networks has been studied recently with: a tree-based model checking in [22], a Bayesian network in [23], inductive logic programming in [3]. Among them, [3] has a direct impact on control, but its application to guarantee strict QoS deserves further analysis.

VII. Conclusions and Future Work

The approach proposed here concerning the joint adoption of machine learning and control allows to both optimize system performance and derive important insights into the structure of the problem. The application to equivalent bandwidth is just an example as many other communication settings can be considered in virtue of the general capabilities of the approach.

References

[1] L. Cruvinel, T. Vazao, "Profile-based Adaptive DiffServ Policing with Learning Techniques," 20th International Conference on Computer Communications and Networks (ICCCN) 2011, Maui, Hawaii, July 31 2011-Aug. 4 2011.

[2] A. Jayaraj, T. Venkatesh, C. S. R. Murthy, "Loss Classification in Optical Burst Switching Networks using Machine Learning Techniques: Improving the Performance of TCP," IEEE Journal on Select. Ares in Commun., vol. 26, no. 6, Aug. 2008, pp. 45-54.

[3] M. Mbaye, F. Krief, H. Soldano, "Collaborative Strategy Learning for Distributed Network Self-Configuring," 1st Internat. Conf. on Commun. and Networking, 2009, Hammamet, Tunisia, 3-6 Nov. 2009.

[4] M. Marchese, M. Mongelli, "Measurement-based Computation of Generalized Equivalent Bandwidth for Loss Constraints," IEEE Comm. Letters, vol. 11, no. 12, Dec. 2007, pp. 1007-1009.

[5] S. Georgoulas, P. Trimintzios, G. Pavlou, K. Ho, "An Integrated Bandwidth Allocation and Admission Control Framework for the Support of Heterogeneous Real-time Traffic in Class-based IP Networks," Computer Communications, vol. 31, no. 1, Jan. 2008, pp. 129-152.

[6] J. Aweya, M. Oullette, D. Y. Montuno, "A self-regulating TCP acknowledgment (ACK) pacing scheme," Int. Journal of Network Manag.," vol. 12, no. 3, May/June 2002, pp. 145-163.

[7] P. Belzarena, L. Aspirot, "End-to-end quality of service seen by applications: A statistical learning approach," Computer Networks, vol. 54, 2010, pp. 3123-3143.

[8] M. Marchese, M. Mongelli, "Performance Evaluation of Bandwidth Adaptation over DVB Satellite Channels,"Proc. IEEE Global Communication Conference 2011 (Globecom 2010), Houston, Texas, USA, 5-9 Dec. 2011.

[9] M. Baglietto, F. Davoli, M. Marchese, M. Mongelli, "Neural Approximation of Open-loop-Feedback Rate Control in Satellite Networks,"IEEE Trans. On Neural Networks, vol. 16, no 5. Sept. 2005, pp. 1195-1211.

[10] X. Glorot, Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks,"Proc. 13th Internat. Conf. on Artificial Intelligence and Statistics (AISTATS 2010), May 2010, Chia Laguna, Sardinia, Italy, in Journal of Machine Learning Research, 2010, vol. 9, pp. 249-256.

[11] F. Davoli, M. Marchese, M. Mongelli "Non-linear Coding and Decoding Strategies Exploiting Spatial Correlation in Wireless Sensor Networks," IET Commun., in press.

[12] E. Akyol, K. Rose, T. A. Ramstad, "Optimized Analog Mappings for Distributed Source-Channel Coding," Proc. Data Compression Conference (DCC), 2010.

[13] L. Devroye, L. Gyorfi, G. Lugosi, A Probabilistic Theory of Pattern Recognition, New York: Springer-Verlag, 1997.

[14] P. Marbach, O. Mihatsch, and J. N. Tsitsiklis, "Call Admission Control and Routing in Integrated Service Networks Using Neuro-Dynamic Programming," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 2, February 2000, pp. 197-208.

[15] M. Muselli "Switching Neural Networks: A New Connectionist Model for Classification", in WIRN/NAIS 2005, vol. 3931 of Lecture Notes in Computer Science, 2006, Eds. B. Apolloni, M. Marinaro, G. Nicosia, R. Tagliaferri, Berlin: Springer-Verlag, pp. 23-30.

[16] M. Muselli, E. Ferrari, "Coupling Logical Analysis of Data and Shadow Clustering for partially defined positive Boolean function reconstruction," IEEE Transactions on Knowledge and Data Engineering, 2011, vol. 23, pp. 37-50.

[17] E. Ferrari, M. Muselli "Efficient constructive techniques for training Switching Neural Networks," in Constructive Neural Networks, vol. 258 of Studies in Computational Intelligence, 2009, Eds. L. Franco, D.A. Elizondo, J.M. Jerez, Berlin: Springer-Verlag, pp. 25-48.

[18] M. Marchese, M. Mongelli, "Reference Chaser Bandwidth Controller for Wireless QoS Mapping under Delay Constraints,"EURASIP Journal on Wireless Communications and Networking, vol. 4, 2010.

[19] T.P. Vogl, J.K. Mangis, A.K. Rigler, W.T. Zink, D.L. Alkon, "Accelerating the Convergence of the Back-Propagation Method," Biological Cybernetics, Springer Verlag, 1988.

[20] http://www.impara-ai.com/.

[21] P. Bermolen, M. Mellia, M. Meob, D. Rossi, S. Valenti, "Abacus: Accurate behavioral classification of P2P-TV traffic," Computer Networks," vol. 55, 2011, pp. 1394-1411.

[22] T. Samak, A. El-Atawy, E. Al-Shaer, "QoS Policy Verification for DiffServ Networks," 19th IEEE International Workshop on Quality of Service (IWQoS), 2011, San Jose, CA, 6-7 June 2011.

[23] X. Lin, B. Cheng, J. Chen, "Context-aware end-to-end QoS qualitative diagnosis and quantitative guarantee based on Bayesian network," Computer Commun., vol. 33, 2010, pp. 2132-2144.