2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)

Video Quality Prediction under Time-varying Loads

Obinna Izima Ruairí de Fréin Mark Davis
Communications Network Research Institute
School of Electrical and Electronic Engineering, Dublin Institute of Technology, Ireland

***Abstract*—We are on the cusp of an era where we can responsively and adaptively predict future network performance from network device statistics in the Cloud. To make this happen, regression-based models have been applied to learn mappings between the kernel metrics of a machine in a service cluster and service quality metrics on a client machine. The path ahead requires the ability to adaptively parametrize learning algorithms for arbitrary problems and to increase computation speed. We consider methods to adaptively parametrize regularization penalties, coupled with methods for compensating for the effects of the time-varying loads present in the system, namely load-adjusted learning. The time-varying nature of networked systems gives rise to the need for faster learning models to manage them; paradoxically, models that have been applied have not explicitly accounted for their time-varying nature. Consequently previous studies have reported that the learning problems were ill-conditioned –the practical, undesirable consequence of this is variability in prediction quality. Subset selection has been proposed as a solution. We highlight the short-comings of subset selection. We demonstrate that load-adjusted learning, using a suitable adaptive regularization function, outperforms current subset selection approaches by 10% and reduces computation.**

***Index Terms*—Machine Learning, Load-adjusted Learning, Network Analytics, Cloud Services and Applications.**

I. Introduction

Advances in widespread internet connectivity enables users to watch online videos anytime and anywhere. Given the consumer's range of choice, competition for subscription fees for cloud-based services such as Internet Protocol Television (IPTV) and Video-on-Demand (VoD) is fierce. The scale of this problem can be inferred from the Cisco Visual Networking Index [1], which predicts that annual global IP traffic will reach about 3.3 Zettabytes by 2021; video traffic will contribute 82% of all consumer internet traffic.

Motivation: Growth of cloud-based video traffic will bring about a strain on the capacity of cloud-based services to deliver good quality video; users who seek-out better Quality of Service (QoS) will increase the dynamicity of these systems. Delivering video streams over dynamic IP-based cloud infrastructures may introduce impairments such as packet loss, delay and/or jitter which may deteriorate the QoS received by the end-users. To meet the dynamicity challenge, service providers who wish to remain competitive must (1) over-provision the system and (2) be able to automatically and accurately predict the quality of video being received, so that they can leverage this redundancy. Improving learning algorithms without increasing their computational cost is important for cloud services and application community because responsivity to poor QoS is a differentiating factor for consumers.

Dynamically changing systems that host cloud-based services are challenging to model. IPTV delivered in this way are challenging to predict. The learning algorithms (such as [2]) used in this setting should have low complexity in order to facilitate timely network management [3]. The authors of [4] considered video service-level prediction. They used statistical learning to predict client-side metrics for a video streaming service, VLC [5] using variants of Lasso [6], Linear Regression, Random Forests and Ridge Regression [2].

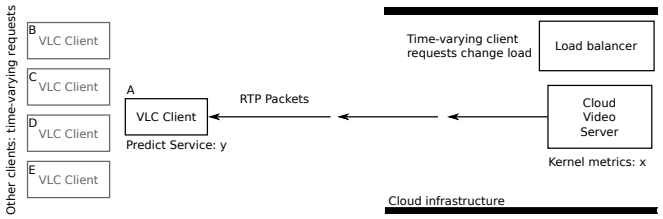

Fig. 1: Goal: predict the service levels of client A, y, given a time-varying number of other clients B, C, D, E, etc using the kernel metrics, x from the video server in the cloud. Previous works have not modeled the time-varying number of clients.

Contribution: Using the approach in [4] as our base-line, we contribute adaptive learning methods that have lower computation complexity and yield more accurate predictions. This is achieved by considering (1) the role of regularization via the Elastic Net [7] which automates the choice of regression parameter and function; (2) the efficacy of load-adjusted learning, originally contributed in [8], e.g. using the TCP socket count to improve the prediction algorithm's performance; and finally, (3) determining whether or not subset selection, our base line method [4], or load-adjusted learning improves the condition number of a range of learning algorithms using the traces in [4].

Organization: In Section II we discuss the related literature. In Section III we introduce load adjusted learning strategies. In Section IV we evaluate the efficacy of each approach and make recommendations.

II. Related Works

Predicting service-level metrics from cloud hosted device statistics, that give better estimates of future performance is of crucial importance and has been addressed in the signal processing [9] and SL [4] literature. Yanggratoke et al. in [4], hypothesized that by collecting thousands of kernel variables, they could learn and predict the behaviour of the client system. No detailed knowledge of the system components and interactions was assumed. This assumption is appealing because it may allow the network manager to deploy the predictor *blindly*, e.g. without significant supervision or set-up. Recent work [9] used new signal processing results to develop a system load model, in a pre-processing step, in order to aid the subsequent service level prediction step. This approach

978-1-5386-7899-2/18/$31.00 ©2018 IEEE
DOI 10.1109/CloudCom2018.2018.00035

Authorized licensed use limited to: Dublin Business School. Downloaded on February 21,2024 at 22:32:32 UTC from IEEE Xplore. Restrictions apply.

was called load-adjusted learning and it improved predictions. It did not however automate the selection of the learning function. The simplest form of a load adjusted learning algorithm is one that trains prediction weights conditional on the value of the load which changes (cf. Fig. 1). These results suggest that the hypothesis that the system statistics, and thus any model learned from them, change as the number of users accessing video content changes and that any learning algorithm that captures this effect will perform better. Modelling the effect of the load on the system performance is straightforward given that the TCP socket count is readily available as one of the server metrics. The scenario that we investigate is applicable in many other situations. Estimating service level metrics using SL has been investigated for another cloud hosted service, Voldemort, a Key-Value store in [4]. This paper addresses the following open questions: (1) Can we automatically choose the best learning function to use? Parametrization of learning algorithms is a major open problem. (2) Is subset selection a credible approach for reducing computational complexity? (3) Can we speed-up learning?

One approach taken in the literature to reduce variance in the performance of a predictor is to examine the feature set via subset selection [2]. This variance may be explained by the poor condition number or low rank of the feature set matrix used to learn the prediction coefficients. Starting out with the full feature set of the infrastructure statistics which included device statistics from the cluster and the network the authors of [4] reduced the infrastructure feature set successively to improve prediction. Their subset selection approach did not specifically account for the changing system load. We investigate if a load-adjusted learning approach performs a similar role, more effectively.

The authors discuss a scenario in [10] for learning from a set of network-level metrics, e.g. delay, loss, and jitter measurements, in order to predict the QoS metrics for IPTV streaming clients. They concluded that their prediction approach was accurate, as long as the packet loss ratio was minimized. The work in [11] uses SL models to predict Quality of Experience metrics of a multimedia service. Closely related to our work here is [8] in which the authors characterize the system we seek to investigate. A case is made for modeling the presence of the load, which had not been considered before. As the number of subscribers of a system increases an increase in usage this causes some level of excitation in the system- the load effect. We consider whether or not subset selection [2] combined with load-adjusted learning [9] improves prediction.

III. Learning Strategies

A client accesses a VoD service which runs on the server in the cloud in Fig. 1. Device statistics x are collected on the server. They consist of operating level metrics such as the number of processes and the TCP socket count, TCPSCK. The load signal, the number of active clients (A, B, C, etc in Fig. 1), can be measured using the TCPSCK rate of x. Device statistics, x are used to predict the service-level metrics, y, the RTP packet count in this instance, at client A. Prediction weights may also be learned for other clients. Fig. 2 illustrates the RTP Packet count recorded during 25000 seconds. It also illustrates the TCPSCK kernel parameter. TCPSCK plays an important role on the performance of the service-level metrics. As the load increases, the TCPSCK increases and may lead to a decrease in the number of RTP packets at the client because the system does not have unlimited resources. Other kernel metrics exhibit a similar behaviour.

Load-adjusted model: The authors of [8] modelled the relationship between the device statistics, the service-level metrics and the load using a linear model. They expressed the response of the server, with respect to kernel metric n, the n-th feature, to one request for a video at time i as the sum of a load-based component $\hat{u}_i[n]$ and a feature specific component $\epsilon_i[n]$,

$$x_i[n] = \hat{u}_i[n] + \epsilon_i[n], \quad \text{where} \quad i \in \mathbb{Z}, x_i[n], \hat{u}_i[n] \in \mathbb{R}. \quad (1)$$

A feature refers to a metric on the operating system level, for instance, the number of active TCP connections. The feature set $x_i[n]$ was constructed in [4] using the System Activity Report which computes system metrics over a given time interval. In Eqn. 1, $x_i[n]$ refers to the n-th feature at time index i. The RTP packet rate, y_i is the observed application level metric, at time i. The signal $\hat{u}_i[n]$ in Eqn. 1 corresponds to an increase in the CPU workload; for instance, an additional α_n units for each user for the duration of the video requested by the user. The deviations from the expected performance are captured by the noise signal $\epsilon_i[n]$. We assume that $K(i)$ is the number of users requesting the service at time i. For example, when clients A, B and C are receiving video at time i, $K(i) = 3$. The response of the n-th feature to the time-varying load is

$$x_i[n] = \alpha_n K(i) + \sum_{k=1}^{K(i)} \epsilon_i[n,k]. \quad (2)$$

The load signal $\alpha_n K(i)$ denotes the number of active users at time i times the resources one user uses, α_n. A more in-depth treatment of this model is given in [9].

Un-adjusted learning: Previous methods do not model the time-varying load. They assume that $K(i)$ is constant C.

$$x_i[n] = \alpha_n C + \sum_{k=1}^{K(i)} \epsilon_i[n,k]. \quad (3)$$

Problem: Our objective is to predict the RTP packet rate y_i using the features $x_i[n]$ given a time varying load $K(i)$.

We examine four different regression methods, which are well-understood and fast to achieve this. We start with Linear Regression (LR), a baseline for most SL techniques, which gives good prediction performance. LR models the relationship between the metric we want to predict y, also known as the dependent variable and the independent variable of predictors x as a linear function of the form:

$$\hat{y}_i = \sum_{n=1}^{N} x_i[n]\beta[n] \quad (4)$$

where $x_i[1]$ represents the intercept and the remaining features represent the feature space of the predictors. The model coefficients, which we use for prediction, are $\beta[n]$ where $n = 1, \ldots N$. LR computes the coefficients that minimize the residual sum of squares; we evaluate the performance of the resulting predictor using the Root Mean Square Error (RMSE).

Authorized licensed use limited to: Dublin Business School. Downloaded on February 21,2024 at 22:32:32 UTC from IEEE Xplore. Restrictions apply.

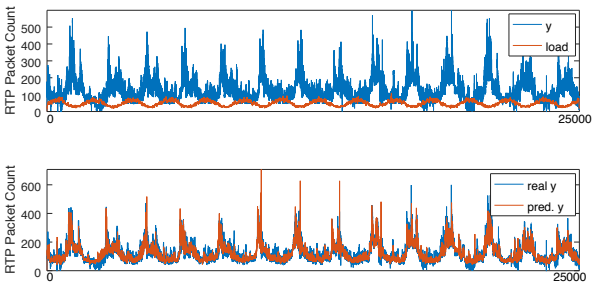

Fig. 2: Row 1: The RTP packet count, y, is illustrated for 25000s along with the system load, TCPSCK. TCPSCK is feature of x. The request patterns change periodically. As TCPSCK increases the RTP packet count decreases, which illustrates the dependence between these statistics. Row 2: The LA Elastic Net prediction is compared to the true y.

LA Learning: The LR model is load-adjusted by training a set of weights for each value of the load signal.

$$\hat{y}_i\Big|_{K(i)=k} = \sum_{n=1}^{N} x_i[n]\beta[n]\Big|_{K(i)=k} \tag{5}$$

It is un-adjusted when all samples irrespective of the load are used during training (Eqn. 5). The feature space we examine is a high dimensional one. To increase speed, the authors of [4] (which serves as a second base line), pruned the feature space by determining a subset of the predictors that contributed the most (cf. [2]). We then apply Lasso and Ridge Regression (RR) methods which are variants of LR. We first evaluate the model using RR which includes an ℓ_2-norm penalty on the coefficients, which maintains a small amount of energy in each coefficient. We evaluate the Lasso method which imposes an ℓ_1-norm on the regression coefficients. In contrast with RR the Lasso attempts to force some of the coefficients towards zero [6]. Both RR and the Lasso are shrinkage methods. The Lasso performs a form of automatic variable selection and continuous shrinkage, using zero coefficients to *turn-off* features.

One limitation of the Lasso in terms of prediction performance occurs when the number of observations is much larger than the number of predictors. We examine 50000 observations and 264 predictors. The challenge lies in choosing between RR and the Lasso. We apply the Elastic Net (EN) model [7] owing to its ability to perform shrinkage and variable selection just like the lasso method. The EN approach is suitable as high correlations exist between our predictors; the lasso has been empirically observed to suffer from poor prediction accuracy. In summary by using the EN we automatically chooses the learning algorithm and load-adjusted learning increases the computation speed.

IV. Numerical Evaluation

Our evaluation is organized as follows: (1) We compare the performance of the models learned using LR, RR, the Lasso and EN, e.g the base-line methods. These methods are collectively called UnAdjusted (UA) learning algorithms here. These experiments are based on the data in [4] where the effect of the load is not taken into account. However, in this paper we use the EN to automate the selection of learning algorithm, which was not done in [4]. (2) In comparison with the UA algorithms, we demonstrate the performance of Load-Adjusted (LA) learning using LR, RR, the Lasso and EN, by explicitly taking into account the effect of the load on the learning algorithms. This is the first time that this UA versus LA learning study has been considered. (3) We consider the efficacy of subset selection on learning and prediction performance. Previous works proposed to improve prediction performance of UA algorithms by using subset selection routines to reduce the variability in prediction accuracy. We evaluate the effect of subset selection on both UA and LA learning. Prediction accuracy is assessed using the RMSE and Adjusted R-squared score.

TABLE I: RMSE for LA Versus UA Learning

Model	Load-Adjusted RMSE	Un-Adjusted RMSE
Ridge	**28.49**	30.74
Lasso	**28.46**	29.85
Elastic Net	**28.36**	28.53

Model Fitting, Analysis and Evaluation Procedure: The feature set was first pre-processed to remove all non-numeric and constant value features. We then applied LR, RR, the Lasso and EN to the traces to learn UA prediction models for the RTP packet rate, our service level metric. All four models were implemented in RStudio. We evaluated the models using the validation set approach using a 60-40 training-test data split. The poor performance of LR estimates motivated our study of shrinkage methods.

Our implementation of RR, the Lasso and EN required a method for selecting the regularization parameter, λ, for the penalty function. Recall that RR, the Lasso and EN use an ℓ_2-norm, ℓ_1-norm and combination of both norms as a weighted (by λ) penalty term. The entire path for the results for these models was calculated using path-wise cyclical coordinate descent algorithms. Computationally efficient and effective approaches for evaluating these convex optimization problems were implemented using the glmnet package in R. We employed 10-fold cross-validation to find the optimal value of λ for each model. This value was used in subsequent learning and prediction experiments. Different values for λ were determined for both the UA and LA algorithms. We selected a sequence of values between 0.0001 and 1 and applied cross-validation to automate λ selection.

Notably the Lasso exhibited less sensitivity to the selection of λ for the LA case than for the UA case. The tendency of the Lasso to *turn-off* components made the tuning of the λ parameter more sensitive to changes in the load. In short, the choice of the best value of λ depends more of the load on the system in the UA case than for the LA case. It is encouraging that when we LA the data the Lasso is easier to tune. The best value of λ for RR was approximately the same for both the UA and LA case. One explanation for this is that RR turns-on many coefficients; therefore changes in the load do not change the sensitivity to λ as more coefficients are active.

LA learning was implemented based on [9], but extended to each of the base-line UA approaches above. We obtained a subset of the entire data set for which the load value, namely the TCPSCK, was fixed and we had greater than 500 samples. These samples were then divided into a 60-

Authorized licensed use limited to: Dublin Business School. Downloaded on February 21,2024 at 22:32:32 UTC from IEEE Xplore. Restrictions apply.

TABLE II: LA models Versus LA with Subset Selection

Load Values	Load-Adjusted		LA & Subset Selection	
	RMSE	R Squared	RMSE	R Squared
30	38.43	71.2	40.11	69.2
40	25.48	64.4	32.73	42.1
50	24.20	42.2	24.34	37.7
60	19.15	47.9	24.61	2.3
70	18.91	43.4	23.46	4.2

40 training-test split as before. We utilized the same cross-validation procedures described above to find the best λ values for each of the LA algorithms. We learned a different model for each value of the load, which removed the conditional dependence on the load within each of these sets of data and reduced the computational complexity or learning.

Results: Table I lists the performance of RR, the Lasso and EN for both LA and UA learning on 20000s of test data. The RMSE is better for all prediction algorithms if the algorithm is LA (boldfont in table). The majority of the values of y lie in the range 20 to 100 RTP packets/s. The RR LA estimates are over 2 RTP packets/s better than for UA RR-based learning. Expressed in percentages, the improvement in prediction performance is $\approx 10\%$ to $\approx 2\%$ better for LA RR learning. The EN gives the best predictions. RR performs the worst. We conclude that LA learning improves service level prediction in this case. We illustrate the accuracy of LA Elastic Net predictions of the RTP packet/s in order to demonstrate what a RMSE of 28.36 means for this service level metric trace. The true RTP packet count is illustrated for comparison in Fig. 2 (Row 2). The main point is that the LA EN prediction algorithm seems to make more accurate predictions when the RTP packet count is high, when the TCPSCK is low. Using LA learning RTP packet/s predictions are 2 packets/s better than UA learning. This prediction improvement is significant; service level agreements that give guarantees about the expected levels of RTP packets/s a client should receive would be affected by misprediction.

We have given evidence that LA learning improves the UA learning methods and demonstrated that ENs give the best prediction performance. We consider the effect of subset section on LA learning to determine if load-adjusting the features or performing subset selection explains the performance gain.

We tabulate the performance of LA learning versus LA learning where the subset selection has been applied to the features in Table II. We use the EN, the best performing learning algorithm, as a learning method in these experiments. We list the results for predictions when the TCPSCK on the system is $30, 40, 50, 60, 70$ in order to demonstrate the behaviours of the predictors under different system loads. In summary, LA learning without subset selection performs better than LA with subset selection learning. LA learning gives RTP packet count predictions which are 2, 7, .1, 5 and 5 RTP packets/s more accurate than when LA learning is used in conjunction with subset selection for successive load values. These results are also evaluated using the adjusted R-squared score. They show that these performance gains are not an artifact of the measurement approach. Higher adjusted R-squared scores are better than lower scores; the performance gain achieved by LA learning over LA learning combined with subset selection is confirmed by the adjusted R-squared score.

Discussion: The best model for predicting RTP packet/s is the EN. The EN boasts the ability to overcome the limitations of the closely matched Lasso by automatically tuning its objective function so that it performs better predictions based on the data. Our results demonstrate that using subset selection to improve the predictions from various regression algorithms may not be the best strategy. In a novel approach we evaluated how Load-Adjusted Elastic Nets could be applied to training data to improve the performance of (1) all UA learning algorithms and (2) LA learning algorithms with subset selection. Note that this improvement in performance does not make any additional assumptions about that data. We will investigate how weakened forms of the independence assumption made by these regression methods can improve prediction.

V. Conclusions

A method for improving predictions of the level of RTP packet/s received by a client by up to 7 RTP packets/s was introduced. This performance gain was achieved by adopting a LA learning approach. We demonstrated that subset selection reduced the effectiveness of the prediction. Given the increased dynamicity of modern service delivery and host infrastructures these late-breaking results are of interest to practitioners as they demonstrate that off-the-shelves approaches may be improved and automated without additional sources of data.

References

[1] Cisco Systems 2017, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021 White Paper," *Cisco*.

[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, USA: Springer, 2001.

[3] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *J. Int. Ser. App.*, vol. 9, no. 1, p. 16, Jun 2018.

[4] R. Yanggratoke, J. Ahmed, J. Ardelius, C. Flinta, A. Johnsson, D. Gillblad, and R. Stadler, "Predicting service metrics for cluster-based services using real-time analytics," *CNSM*, pp. 135–143, 2015.

[5] VLC. [Online]. Available: http://www.videolan.org/vlc/

[6] R. Tibshirani, "Regression Shrinkage and Selection Via the Lasso," *J. Roy. Stat. Soc. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[7] H. Zou and T. Hastie, "Regularization and variable selection via the Elastic-Net," *J. Roy. Stat. Soc.*, vol. 67, no. 2, pp. 301–320, 2005.

[8] R. de Fréin, "Effect of system load on video service metrics," *IEEE ISSC*, pp. 1–6, 2015.

[9] ——, "Source separation approach to video quality prediction in computer networks," *IEEE Comm. Let.*, vol. 20, no. 7, pp. 1333–6, Jul. 2016.

[10] S. Handurukande, S. Fedor, S. Wallin, and M. Zach, "Magneto approach to QoS monitoring," *Proc. 12th IFIP/IEEE Int. Symp. Integrated Network Management*, pp. 209–216, 2011.

[11] P. M. Arun Kumar and S. Chandramathi, "Intelligent video QoE prediction model for error-prone networks," *Indian J. of Scien. and Tech.*, vol. 8, no. 16, 2015.

Authorized licensed use limited to: Dublin Business School. Downloaded on February 21,2024 at 22:32:32 UTC from IEEE Xplore. Restrictions apply.