Demo Abstract: Explaining Web users' QoE with Factorization Machines

Alexis Huet and Dario Rossi
Huawei Technologies Co. Ltd — {alexis.huet,dario.rossi}@huawei.com

***Abstract*—Whereas most of the literature employs classic machine learning techniques (such as C4.5 trees, Random Forest and Support Vector Machines) to *improve forecast accuracy* of QoE models, in this demo we explore the use of an information filtering system (Factorization Machine) to get *fundamental insights and explain* the relationship between QoE and different features.**

I. Introduction

Management of users' Quality of Experience (QoE) has become an important asset in the commercial war among Internet Service Providers (ISPs), to engage and keep customers locked in their userbase, especially given the smaller revenue margins with respect to Content Service Providers (CSPs). On the one hand, user QoE is notoriously difficult to measure, part of which is due to the continuous evolution of services and applications, which require to specialize QoE models to particular applications. Consider the World Wide Web for example: in recent years, a plethora of objective application-level QoS metrics have been proposed beyond Page Load Time (PLT), such as those normalized at the W3C Navigation Timing or newer ones proposed in the industry such as SpeedIndex [1] (SI) and variants [3], [5]. To assert whether such QoS metrics might have predictive power of the user QoE, a number of work started polling real users opinions (e.g., collecting and aggregating MOS [3] or user perceived page load times [5], [8]) to develop models linking objective application-QoS metrics to subjective user-QoE (such as simple ITU-T G1030 or more advanced models [3]–[5]). Generally, the goal of the above work is to *propose accurate models* to forecast QoE scores from a feature vector $\overline{QoS}$ which may contain tens to hundreds of variables $QoE = f(\overline{QoS})$, possibly using machine learning techniques [3], [5] to maximize the model prediction accuracy. These models may have high prediction power, but nevertheless are obscure in nature: we argue that a complementary approach is also desirable, such as employing collaborative filtering systems like Factorization Machines (FM) [6], to *systematically explain* the relative importance of the different dimensions that the QoE problem involves.

Particularly, in this demo we apply FM on the dataset made publicly available by [3] to investigate with a scientifically principled approach, very practical yet relevant questions such as the following: **(Q1)** *What is the relative contribution of network (protocol, delay, loss) vs application-level (PLT, TTI, ATF, SI) metrics in the prediction accuracy?* **(Q2)** *To what extent webpage-agnostic models are accurate?*

(Q1) is motivated by the fact that, whereas CSPs can relatively easily collect such application-level metrics, ISPs generally can only collect network-level metrics: one crucial question is thus whether ISPs and equipment vendors should invest into developing inference techniques able to approximate rather complex objective application-level metrics [1], [5], or whether the development cost is not justified as easily accessible objective network-level metrics could already provide most of the discriminative power.

(Q2) is motivated by [3], which suggests that given the wide diversity among pages of the world "wild" web, one could expect to have better accuracy by developing webpage-specific models. This could be manageable for a set of particularly relevant websites (e.g., top-100) and could assist ISPs to develop fairly accurate and scalable models to proactively detect QoE degradation: since the top-100 websites are frequently visited, they would offer ISPs a continuous and statistically relevant stream of QoE samples.

Clearly, **(Q1)** and **(Q2)** are just to illustrational examples out of the different possible questions that can be answered through the FM model: the aim of this demo is provide intuitive answers to **(Q1-2)** and similar practical questions, and to further let scientists, researchers and practitioners interact with the provided FM models, by either altering inner parameters of the FM model (e.g., number of latent factors, capping SGD iterations, altering regularization, etc.), or by altering the variables that are given as input to FM (e.g., network-level vs application-level, binning strategies, etc.). An (early stage) prototype of the demonstration is available at [2].

II. Factorization Machines (FM)

We leverage Factorization Machines (FM) [6] to quantitatively study the different embedding dimensions of the QoE problem. FM is a kind of recommender system which is well-suited for our experiment: (i) it allows estimation of parameters under very *sparse data*, without loosing information by aggregating samples into mean scores; (ii) it seamlessly integrate *arbitrary contextual features* such as application vs network-level metrics; (iii) finally, it has *linear complexity* in the overall number of classes n of the contextual features.

For example, let consider a user U navigating on webpage W experiences a measurable application-QoS performance A and a network-QoS performance N, rating his QoE with a grade $y \in [\![1, 5]\!]$. Without loss of generality, in the following we quantize application-level QoS A by discretizing the value of PLT (or SI, etc.) into 5 *quintiles* of the populations. The different features U, W, A, N are encoded into one-hot vectors (though this is not a hard constraint of the model)

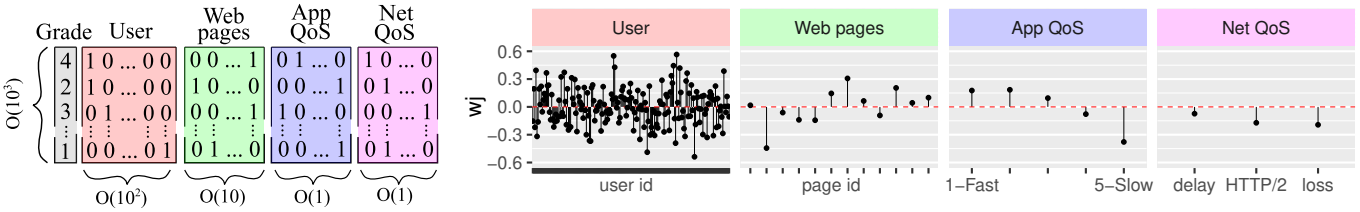

Fig. 1. Illustration of Factorization Machines on the QoE dataset provided in [3] (interactive demo interface accessible online at [2])

and concatenated into a vector $\mathbf{x} = (x_1, \ldots, x_n)$ of length n. The resulting sparse matrix is illustrated in the left of Fig. 1, where m rows correspond to the number of collected user grades, and the n columns to the features collected during each experiment. For instance, the first row consists of a QoE grade $y = 4$ given by the first user $U = (1, 0, \ldots, 0)$, for the page with the last index $W = (0, \ldots, 1)$, with measured QoS values for application A falling in the [20%,40%) range, and a certain configuration N of network-QoS performance.

The FM model is defined as:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{j=1}^{n} w_j x_j + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

where w_j represents the weight of the j-th feature, $\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} v_{j,f}$ is the factorized interaction weight of the feature-pair (i, j), and $k \geq 0$ is a hyperparameter defining the number of latent factors in the model. The FM model weights to be estimated are thus w_0, w_j, and $v_{j,f} \in \mathbb{R}$ for $j \in [\![1, n]\!], f \in [\![1, k]\!]$. The gradient of the FM model with respect to each parameter is straighforward to get and allows estimating weights via Stochastic Gradient Descent (SGD) with low computational complexity (cfr. Sec. III-C of [6]).

III. Assessment of Web QoE factors

Dataset We leverage the dataset of [3] consisting, after sanitization, of over 2000 Web browsing sessions from 188 volunteers. Each session collects QoE feedback (1-Bad to 5-Excellent) along with various contextual features (user ID, visited website, several application and network QoS features). For the sake of illustration, we consider one application QoS feature (PLT) and three network QoS features (HTTP protocol, additional delay, presence of loss). PLT values are separated into 5 bins of equal sizes indexed from 1-Fast to 5-Slow.

Methodology We train FM models with libFM [7] using SGD for 64 different feature selections (including or not user ID, webpage, PLT, HTTP protocol, presence of delay, presence of loss) and 16 different sets of hyperparameters (latent space of dimension, number of iterations and L2 regularization). For each set of parameters, we compute Root-mean-squared error (RMSE) using 5-fold cross-validation, and separately extract final weights using the whole data set. RMSE is computed between true grades y and estimated grades $\hat{y}$, without any intermediary aggregation (for brevity, RMSE and weights are discussed only in the best configuration of hyperparameters).

Demo highlights The demo interface [2] allows (i) interactive selection of features to be integrated into the FM model, as well as (ii) tuning of FM hyperparameters. For a selected model, the demo outputs a description of the corresponding model, model accuracy and RMSE, and a visual summary of weights w_j as illustrated in the right of Fig. 1. Interestingly, weights have a *direct physical interpretation* as they linearly add to the forecasted grade, so that the magnitude of their dispersion immediately conveys the relative importance of the feature under investigation.

Answer to (Q1) The two rightmost weight plots of Fig. 1 clearly show that, once application-level QoS information is provided to the model, additionally providing network-level information only brings limited benefits: indeed, while the weights vary drastically depending on the PLT bin, the weights associated with network-level measurement are very small. This clearly answers (Q1), prioritizing the inference of accurate L7 application-level measurement, as opposite as to just collecting L3 network-level measurement – which opens new challenges for operators, since these statistics are not easy to collect at layer L3/L4 that are oblivious to encrypted application requests of Web browsers.

Answer to (Q2) The two leftmost weight plots of Fig. 1 additionally show that knowledge of the webpage has a dramatic impact as well: notice that weight dispersion has about the same magnitude than the user or the L7 application QoS metrics. This fact alone answers (Q2), suggesting that page-oblivious models such as those proposed by ITU-T G1030 are largely simplistic and not sufficient to faitfully represent user QoE – an interesting question that remains open is to what extent webpage properties related to page complexity (e.g., number of objects, javascript and domains) can be leveraged to further harness and explain QoE.

References

[1] https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index.

[2] https://huet.shinyapps.io/webqoe/.

[3] D. N. da Hora, et al. Narrowing the gap between QoS metrics and web QoE using above-the-fold metrics. In *PAM*. 2018.

[4] M. Fiedler, et al. A generic quantitative relationship between quality of experience and quality of service. *IEEE Network*, 24(2), 2010.

[5] Q. Gao, et al. Perceived performance of top retail webpages in the wild. In *ACM Internet-QoE*. 2017.

[6] S. Rendle. Factorization machines. In *IEE ICDM*. 2010.

[7] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1, 2012.

[8] M. Varvello, et al. Eyeorg: A platform for crowdsourcing web quality of experience measurements. In *ACM CoNEXT*. 2016.