Predicting QoS in LTE HetNets based on location-independent UE measurements

Jessica Moysen, Lorenza Giupponi, Nicola Baldo, Josep Mangues-Bafalluy
Centre Tecnològic de Telecomunicacions de Catalunya-CTTC
Av. Carl Friedrich Gauss 7, 08860 Castelldefels (Spain)
$\{jessica.moysen, lorenza.giupponi, nicola.baldo, josep.mangues\}@cttc.es$

Abstract—**This paper aims to find patterns of knowledge from physical layer data coming from Heterogeneous Long Term Evolution (LTE) networks. We discuss how the collected data is employed in such a manner that improves Minimization of Drive Tests (MDT) functionality in LTE networks. In particular we aim to predict Quality of Service (QoS) expressed in terms of throughput of the User Datagram Protocol (UDP) traffic flow. We propose regression models to estimate QoS, by extrapolating information independently of the user's physical location. In particular our approach allows to estimate the QoS in any location, based on measurements collected at anytime in the past, or anywhere in the network. This will allow to significantly reduce costs of future network deployments, even in complex and heterogeneous scenarios, such as those foreseen in stadiums, events, etc. We identify three feasible regression models, and we compare results in terms of prediction accuracy.**

I. Introduction

Minimization of human intervention in cellular networks is achieved through the implementation of Self-Organizing Network (SON)s [1]. This concept has been introduced by 3rd Generation Partnership Project (3GPP) in Release 8 and it has been expanding across subsequent releases. The main objective of SON is to reduce the costs associated with network operations, by diminishing human involvement, while enhancing network performance, in terms of network capacity, coverage and service quality. One of the most important SON use cases identified by [1] is the MDT. MDT enables operators to collect User Equipments (UEs) measurements together with location information, if available, with the purpose of optimizing network management, while reducing operational effects and maintenance costs. This feature has been introduced by 3GPP since Release 10, among the targets there are the standardization of solutions for coverage optimization, mobility, capacity optimization, parametrization of common channels, and QoS verification [2]. Since operators are also interested in estimating QoS performance, in Release 11, MDT functionality has been enhanced to properly dimension and plan the network by collecting measurements indicating throughput and connectivity issues [3].

The problem of QoS prediction, estimation and verification has been studied in the literature in [4][5]. Here, the authors address the MDT QoS verification use case by identifying and estimating different KPIs and correlating them with common nodes measurements, to establish whether a UE is satisfied with the received QoS. Other works on QoS prediction focus on WiFi and mobile networks in general [6] [7]. We observe that the works available in literature cover only traditional macrocell scenarios and do not focus on more complex multi layer heterogeneous networks. In addition, available solutions in LTE networks mainly focus on the use case of QoS verification, trying to estimate the QoS perceived by the users in the network, without having to monitor network state and performances through expensive drive tests.

In this paper we propose to move forward, and we propose an approach that not only is able to verify the QoS level experienced by the users, through physical layer measurements of the UEs, but it is also able to predict it based on measurements collected in different moments in time, and from different regions of the heterogeneous network. We propose then to make predictions independently of the physical location, in order to exploit the experience gained in other sectors of the network, to properly dimension and deploy heterogeneous nodes. Target use cases could be the future deployment of heterogeneous nodes, the construction of new infrastructures, e.g., new highways, railways, buildings etc., the satisfaction of customer's complaints, estimation for extraordinary deployments, e.g., stadiums, events, etc. This approach promises high reduction in Operational Expenditure (OPEX) according to MDT philosophy. Without loss of generality, we focus on the throughput as a metric to be predicted, but other interesting indicators could also be considered following the same approach, like the Physical Resource Block (PRB)/ Megabit (MB), as it is proposed in [8] by AT&T.

For our purposes, in this paper we propose to use Supervised Learning (SL) solutions, which are Machine Learning (ML) techniques very useful to identify relations between input and output variables [9]. Our problem is a regression problem, since we want to analyze the relationship between a continuous dependent variable (throughput), and more independent variables (UE measurements). Many regression techniques have been developed in the SL literature, and criteria to select the most appropriate method include aspects such as the kind of relation that exists between the input and the output, or between the considered features, the complexity, the dimension of the dataset, the ability to separate the information from the noise, the training speed, the prediction speed, the accuracy

This work was made possible by NPRP grant No. 5-1047-2-437 from the Qatar National Research Fund (a member of The Qatar Foundation). The work of J. Moysen is also supported by grant TEC2014-60491-R.

in the prediction, etc. It is very difficult to predict the kind of dependency between physical layer measurements in a LTE heterogeneous network and the performances that may derive from them. This is why, for this preliminary study, we focus on two families of regression models, linear and nonlinear regression models [9], and we select the most representative approaches from these families, prioritizing criteria such as the low complexity and the high accuracy: (1) K-Nearest Neighbors (KNN), (2) Generalized Linear Models (GLM), and (3) Support Vector Machines (SVMs). We perform an empirical comparison of these algorithms and we analyze results from these three approaches observing the impact on the prediction of the different kinds and amounts of UE measurements. We benchmark our approach to a physical distance based prediction, and to the Kriging spatial interpolation technique [10].

The outline of the paper is organized as follows. Section II introduces the considered SL algorithms. Section III describes the simulation framework set-up, together with the procedure we followed to collect and prepare the data. Section IV presents meaningful simulation results. Finally, Section V summarizes the conclusion.

II. Supervised Learning

SL is a Machine learning technique which takes training data, organized into input and desired output, to develop a prediction model, by inferring a function $f : \mathbf{x} \rightarrow \mathbf{y}$, returning the predicted output $\mathbf{y}$. The input space is represented by a n-dimensional input vector $\mathbf{x} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)})^T \in \mathbf{R}^n$. Each dimension is an input variable. In addition a training set involves m training samples $((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m))$. Each sample consists of an input vector $\mathbf{x}_i$, and a corresponding output $\mathbf{y}_i$. Hence $x_i^{(j)}$ is the value of the input variable $x^{(j)}$ in training sample i, and the error is usually computed via $\mathbf{y}_i - \mathbf{y_i}$.

In this paper we focus on regression analysis, i.e., $\mathbf{y}_i$ is continuous in nature. It is hard to define the relationship that exists between input and output variables in our regression problem, as in the complex heterogeneous cellular network, the relationship between physical layer measurements and QoS perceived is very complicated, and affected by random propagation effects, as well as multiple transmission, communication and networking aspects. We empirically evaluate different representative models, belonging to the family of linear and nonlinear regression, prioritizing aspects in this selection such as the low complexity, the high accuracy and the speed of training and predicting. On the one hand, linear models are adequate when a linear trend exists in the data, they can be adapted also for nonlinear relations, but with some limitations. On the other hand, nonlinear approaches, do not require any prior model of the nonlinearity. We focus on three representative schemes from these families [9]:

A. K-Nearest Neighbors

KNN is a nonlinear method where the input consists of the k closest training samples in the input space. The predicted output is the average of the values of its k nearest neighbors. A commonly used distance metric for continuous variables is Euclidean distance. The KNN method has the advantage of being easy to interpret, fast in training, and the amount of parameter tuning is minimal. However, the accuracy of the prediction is generally limited.

B. Generalized Linear Model

The linear model describes a linear relationship between the output and one or more input variables, and where the approximation function h_θ maps from $\mathbf{x}$ to $\mathbf{y}$ as follows,

$$\mathbf{y} = h_\theta(\mathbf{x}) = \theta_0 + \theta_1 x^{(1)} + \ldots + \theta_n x^{(n)} \tag{1}$$

where θ_i are the unknown parameter.

Assuming that we only have one input variable, the idea is to choose θ_0, θ_1 so that $h_\theta(\mathbf{x})$ minimizes the following function,

$$\min_{\theta_0, \theta_1} \; J(\theta_0, \theta_1) \tag{2}$$

where $J(\theta_0, \theta_1)$ is the cost function, and is defined as,

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(\mathbf{x}_i) - \mathbf{y}_i))^2 \tag{3}$$

Linear regression type models are highly interpretable, are fast in training and prediction and do not need parameter tuning. However, they can be limited in their usefulness. These models are appropriate if the input-output relation falls along a hyperplane (i.e. the straight line defined above, in case of only one input variable). If though, the relation is not linear, the model should be generalized, in an attempt to capture this relationship.

C. Support Vector Machines

The main motivation of SVM for regression, referred hereafter as a Support Vector Regression (SVR), is to find a function $f(\mathbf{x})$, i.e., an optimal hyperplane, which approximates all training samples with ε precision. Similarly to other regression methods, this is done by minimizing a cost function. The proposal is that the problem which is stated in a finite dimensional space, is mapped onto a higher dimensional space, where the fit is supposed to be easier. The cost function used by SVM is different by the ones used in other regression methods, to save in computational complexity. The cost function can use nonlinear kernels, so that it allows for nonlinear regression. This process is based on the kernel trick and the representation of the solution is obtained in the dual domain. SVR methods in general show high accuracy in the prediction, and with appropriate kernels, they behave very well also with nonlinear problems. However, they are much harder to interpret than the other methods discussed above, and they need more tuning.

III. Simulation framework set-up

We consider a heterogeneous wireless network, whose system performance has been evaluated on the ns3 LTE-EPC Network Simulator (LENA) platform based on LTE Release 10. The scenario that we set up consists of 1 Enhanced Node

Base station (eNB) with three sectors, which results in 3 cells and 19 UEs with transmit power equal to 46 dBm. The small cell network is based on the dual stripe scenario with 1 block of 2 buildings. We consider 30 blocks in the coverage area of the macrocell. Each building has one floor, with 20 apartments, which results in 40 apartments per block. The Home eNodeB (HeNB) deployment ratio is 0.5, and the activation factor is 1, which results in 20 HeNBs, each one located in an independent apartment. The HetNet scenario is given in Table I, and the parameters used in the simulations are given in Table II.

TABLE I: HetNet scenario.

Macrocell scenario	**Value**
eNB Tx Power	46 dBm
Nº of cells	6
Nº of macro UEs	19
Small cell scenario	**Value**
HeNB Tx Power	23 dBm
Nº Femto blocks	30
Nº of HeNBs per block	20
Nº of home UEs per HeNB	4
Nº of home UEs per block	80
Total Nº of HeNBs	600
Total Nº of home UEs	2400

TABLE II: Simulation parameters.

Parameter	**Value**
PropagationLossModel	HybridBuildings
ShadowSigmaOutdoor	1
ShadowSigmaIndoor	1.5
Scheduler	Round Robin
AMC model	4-QAM, 16-QAM, 64 QAM
Transport protocol	UDP
LTE	**Value**
Cell layout	radius: 500m
Bandwidth	5MHz
No. of RBs	25
TTI	1ms
CQI	period: 1ms; No. of RBs per CQI:2
Simulation time	0.25s

Our approach is based on 3 phases. First we collect the data, then we prepare them, and finally we analyze them through the proposed regression analysis methods.

A. Collect the data

We collect for each UE: (1) the Reference Signal Received Power (RSRP), and (2) the Reference Signal Received Quality (RSRQ) coming from the serving and neighboring cells. The RSRQ is defined as,

$$RSRQ = \frac{nRB \times RSRP}{BW \times RSSI} \tag{4}$$

where nRB is the number of resource blocks, BW is the system bandwidth, and $RSSI$ is the reference signal strength indication, and contains the power received from co-channel serving and non serving cells, adjacent channel interference and thermal noise. As a result, the RSRQ is an indicator of the portion of useful reference signal power received by the UE over the measurement bandwidth BW. Finally, in order to test the QoS performance, the throughput per user is obtained by using UDP Client application, which takes care of the generation of Radio Link Control (RLC) Protocol Data Units (PDUs) allowing multiple flows belonging to different QoS classes.

The size of the input space is $[2400 \times 1200]$, where the number of rows is the number l of UEs in the scenario, and the number of columns corresponds to the number of UE measurements n. In particular, for the serving and the neighbouring cells, each UE reports the RSRP and the RSRQ. As a result, in the data set one column corresponds to the RSRP of the serving cell, the second one corresponds to the RSRQ of the serving cell, 599 correspond to the RSRP of the neighboring cells, and 599 correspond to the RSRQ of the neighboring cells. The size of the output space is $[2400 \times 1]$, which corresponds to the throughput.

B. Preparing the data

Once data are collected, we proceed with the data preparation.

1) The three selected methods for evaluation will benefit in performances if the input variables of the different measurements are on a similar scale and range. So, a common practice is to normalize every variable between $-1 \leq x^{(j)} \leq 1$ range, and replace $x^{(j)}$ with $x^{(j)} - \mu^{(j)}$ over the range (max-min), where $\mu^{(j)}$ is the average of the input variable j in the training set.
2) We create a random partition for test validation from the l sets of input. This partition divides the observations into a training set of m samples, and a test set $p = l - m$ samples. We randomly select approximately $p = \frac{1}{5} \times l$ observations for the test set.
3) For each test value, we predict the throughput, and evaluate performances against the actual value in terms of the Root Mean Squared Error (RMSE) as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{p}(\mathbf{y}_i - \mathbf{y}_i)^2}{p}}$$

where p is the length of the test set, $\mathbf{y}_i$, indicates the predicted value, and $\mathbf{y}_i$ is the testing value of one data point i. In order to compare the RMSE with different scales, the input and output variable values are normalized as follows,

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}}$$

where y_{max} and y_{min} represent the maximum and minimum values in the output set.

IV. Results

We benchmark the three selected regression methods, which take Location Independent (LI) input, based on UE measurements, to the following Location Dependent (LD) schemes, which take the physical position of the samples as input.

1) *Kriging*: The predicted value is evaluated considering Kriging as a spatial interpolation technique. Kriging is an interpolation technique, which selects weights for each point according to its distance from the unknown value. This technique aims at minimizing the error variance, and set the mean of the prediction errors to zero, where the spatial variation is quantified by the variogram [10].
2) *Distance*: The predicted value is that corresponding to the closest training sample.

For our empirical evaluation, we consider 6 cases, each one considering different features in the data set:

- (a) RSRP from the neighboring cells,
- (b) RSRP from the serving and neighboring cells,
- (c) RSRQ from the neighboring cells,
- (d) RSRQ from the serving and neighboring cells,
- (e) RSRP and RSRQ from the neighboring cells, and
- (f) RSRP and RSRQ from the serving and neighboring cells.

For each case, we consider inputs from a variable number of cells, which results in a variable number of columns in the data set, and consequently in a variable number of features in the input space. In particular, with the following notation we consider measurements from:

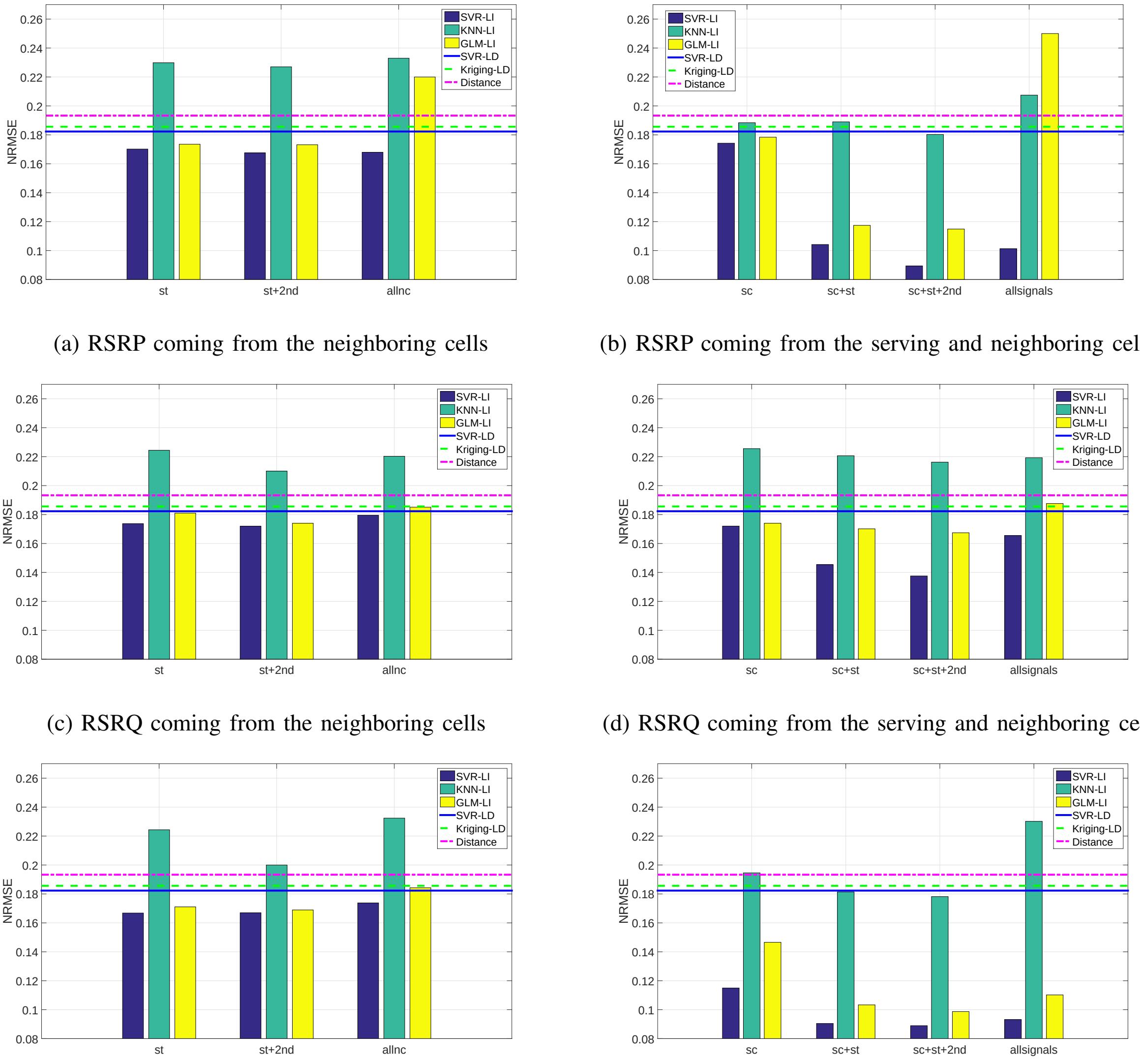
- st: the strongest cell,
- $st + 2nd$: the two strongest neighboring cells,
- $allnc$: all the neighboring cells,
- sc: the serving cell,
- $sc + st$: the serving cell, and the strongest neighboring cell,
- $sc + st + 2nd$: the serving cell and the two strongest neighboring cells,
- $allsignals$: all the serving and neighboring cells.

We show performance results for different algorithms. In particular, we evaluate LI approaches, which do not take into account information about the physical position of the data, and LD algorithms, already presented as the benchmarks. We want to show that abstracting from the physical position of the measurements we can provide better estimations in a LTE heterogeneous network, when considering physical layer measurements. For LI approaches we consider 1) SVR-LI, 2) GLM-LI, 3) KNN-LI. These approaches are benchmarked to the following LD schemes: 1) Distance-LD, 2) Kriging-LD, 3) SVR-LD. The most relevant observations are summarized in the following.

1) From Figures 1 (a) to (f), we observe that the error decreases if we consider more than one input variable, but if we take into account the whole input space the error generally increases, and this behavior is observed for all the algorithms. The reason is that an underspecified model produces biased estimates, while too many features lead to an overspecified model, which tends to have less precise estimates.
2) From Figures 1 (a) to (f), we observe that the SVR-LI tends to provide the best results. This was to expect, as SVR is a powerful regression model, which shows high accuracy in the prediction. Also, properly using the kernels, it allows for proper nonlinear regression, without the need to model a priori the nonlinear trends in the data. This is the main reason why, in our scenario, where the nonlinear relationship between input and output is not known a priori, SVR has provided the most promising results. Since SVR is the scheme which better performs, we propose also a SVR-LD implementation as a benchmark. We observe that SVR-LI provides better results than SVR-LD, which means that, independently of the regression technique the recognition of patterns among UE measurements allows more precise predictions than LD estimations.
3) GLM allows to adapt a linear regression to nonlinear trends in data. However, the model performs worse than SVM, since it is necessary to foresee the specific nature of the nonlinearity in the data, which is unknown in our scenario. Also, Figures 1 (a) to (f) show that GLM-LI provides better accuracy results than all LD schemes, in all cases except for the $allnc$ and $allsignals$ ones, where the model is overspecified. In fact, one of the drawbacks of GLM is that it has difficulties in properly handling redundant features [9].
4) KNN is a very simple algorithm, and as it was to expect, it is the one providing poorest accuracy performances. In particular, it provides lowest accuracy than LD methods.
5) Introducing heterogeneity in the data set (Figure (e) and Figure (f)), i.e., using both RSRP and RSRQ signals as input variables, we improve the accuracy of the results. This is because, due to the complexity of the heterogeneous scenario and to the different effects, at propagation, communication and networking levels, which affect the UE measurements, they provide independent information. Also, the algorithms tend to provide better accuracy when measurements from the serving cell are included in the data set (i.e. Figures 1 (b), (d), (f)), since the signal coming from the serving cell is less affected by propagation effects, and so the associated information is more stable.

V. Conclusion

In this paper we have presented an approach, based on regression analysis, which allows to predict QoS in Heterogeneous LTE networks for UEs, independently of the physical location of the UE, and only based on physical measurements already available in MDT data base. Our approach allows then to predict QoS in a given region, based on measurements extracted from other areas in the network, and in other moments in time, as long as RSRP and RSRQ patterns are identified. We compare results from different regression techniques, namely SVR, GLM, KNN for different amounts and kinds of input/features. We benchmark the results to LD prediction models, where the prediction is strictly related to the position of the UE. We have shown that: 1) the dimension of the input space has an impact in the error, but not necessarily,



(a) RSRP coming from the neighboring cells

(b) RSRP coming from the serving and neighboring cells

(c) RSRQ coming from the neighboring cells

(d) RSRQ coming from the serving and neighboring cells

(e) RSRP/RSRQ coming from the neighboring cells

(f) RSRP/RSRQ coming from the serving and neighboring cells

Fig. 1: NRMSE as a function of different kinds and amount of information.

the bigger the input space, the lower the error, 2) SVR is the most accurate approach for our application, since it allows to fit nonlinearities in the data, without a priori modelling them, 3) considering heterogeneous kinds of inputs (e.g. RSRP and RSRQ), we benefit the prediction, as in the complexity of the proposed scenario, they provide independent information.

References

[1] 3GPP, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self- configuring and self-optimizing network (SON) use cases and solutions," Tech. Rep. 3GPP TR 36.902 v9.3.1, 2011.

[2] Seppo Hämäläinen, Henning Sanneck, Cinzia Sartori, *LTE Self-Organising Networks (SON): Network Management Automation for Operational Efficiency*. John Wiley and Sons, 2012.

[3] Johansson, J., Hapsari, W.A., Kelley, S., Bodog, G., "Minimization of drive tests in 3GPP release 11," *IEEE Communications Magazine*, November 2012.

[4] Fedor Chernogorov and Timo Nihtilä, "QoS Verification for Minimization of Drive Tests in LTE Networks," in *Proceedings of the 75th IEEE VTC Spring, Yokohama, Japan* , May 2012, pp. 6–9.

[5] Fedor Chernogorov and Jani Puttonen, "User satisfaction classification for Minimization of Drive Tests QoS verification," in *24th IEEE PIMRC 2013, London, United Kingdom*, September 2013, pp. 2165–2169.

[6] Rattaro, Claudina and Belzarena, Pablo, "Throughput prediction in wireless networks using statistical learning," in *Latin-American Workshop on Dynamic Networks. Buenos Aires*, 2010.

[7] Mirza, M. and Springborn, K. and Banerjee, S. and Barford, P. and Blodgett, M. and Xiaojin Zhu, "On The Accuracy of TCP Throughput Prediction for Opportunistic Wireless Networks," in *6th Annual IEEE Communications Society; Sensor, Mesh and Ad Hoc Communications and Networks*, June 2009, pp. 1–9.

[8] Gordon Mansfield, "AT&T: Het-Net-Small Cell Placement and the resulting performance," *Small Cell Forum*, March 2015.

[9] Ian H. Witten, Eibe Frank, Mark A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.

[10] Williams C. K. I., "Prediction with Gaussian Processes: From Linear Regression to Linear Prediction and Beyond," *Learning in Graphical Models*, pp. 599–621, 1998.