Reliability Enhancement of 3G Radio Network Prediction by a Conditional Distribution Discrimination Tree

Berna Sayrac, Zakaria Nouir and Benoît Fourestié
France Telecom Research & Development Division
e-mail: berna.sayrac@orange-ftgroup.com

Alain Pétrowski
GET/Institut National des Télécommunications (INT)
e-mail: Alain.Petrowski@int-edu.eu

Abstract— This paper presents a constructive learning system that enhances the reliability and precision of radio network predictions. This task is achieved by finding a correspondence between the probability density distributions of simulated predictions and real measurement data collected from the radio network. Once this correspondence is found, it is possible to arrive at more realistic prediction values from simulation results. After carrying out non-parametric estimations of the probability distributions of the simulations, feature vectors are computed from these estimations, followed by a supervised learning that finds a mapping between the feature vectors issued from the simulations and the estimations of conditional probability distributions of the measurements. The proposed method is evaluated on a 3G radio network using indicators such as UpLink (UL) and DownLink (DL) base station loads. Results show that the proposed scheme is able to yield distributions that are much closer to measurements than simulations. With such a technique, it is possible to predict with enhanced accuracy new configurations and conditions for which we don’t have observations.

I. INTRODUCTION

Radio network planning requires accurate prediction of network behaviour under different conditions and environments. With the deployment of 3G systems whose structure is more intricate than those of 2G systems, the need for accurate prediction has become of paramount importance for a mobile operator. Radio network prediction has so far been carried out by one of two approaches. The first approach is based on computer simulations which use simplified mathematical and/or statistical models representing real, physical phenomena like propagation, traffic, etc. (\textit{a priori} knowledge) \cite{1}. The consistency and applicability of the obtained results to network deployment depends on the modelling quality. However, no model perfectly reflects the reality and hence, utilisation of models in radio network prediction causes inherent discrepancies between simulation results and reality. Radio Network Prediction (RNP) tools constitute a good example of this first approach. The second approach is based on the utilisation of empirical data in the form of observations and/or measurements (\textit{a posteriori} information) to train an automatic learning system like an Artificial Neural Network (ANN) that models the whole radio network \cite{2}. The transfer function found by the automatic learning system is a simple mathematical operator without any physical reality. Since a considerably complex system like a radio access network is learned by the automatic learning system, this approach may lead to robustness and stability problems.

In a previous work, a technique that overcomes the abovementioned drawbacks of the existing two approaches by simultaneous utilisation of network measurements together with simulation results has been proposed \cite{3}-\cite{6}. Its aim is to make the radio network predictions more reliable and more realistic by benefiting from the respective advantages of the two approaches, i.e. the handiness of models and the accurateness of measurements. The scheme basically consists of finding the mapping between the probability distributions of measurements and simulations via a Multi-Layer Perceptron (MLP) trained with Back Propagation (BP) algorithm. Compared with the abovementioned second approach where the learning is done on the complex radio network, this method alleviates the stability and robustness problems by performing learning on the relatively simpler mapping between measurements and simulations. Independent Component Analysis (ICA) and \textit{k}-means clustering algorithms are used to decrease the learning complexity and to enhance the generalization capability respectively. Generalization with different traffic values \cite{3} and with different network parameters \cite{4} on a 3G network yields significantly good results, enabling us to predict with higher precision and accuracy, even for cases in which we do not have the opportunity to perform measurements.

In this paper, we propose a new method that learns the relation between probability distributions of measurements and simulations for the same purpose of decreasing the discrepancy between them, and hence, increasing the reliability and precision of the predictions. The proposed method has several advantages over the MLP used in the abovementioned previous works: it is very fast and parameter-free. The first step of the proposed algorithm consists in non-parametric estimation of the probability distributions of the simulations. Then, in second step, feature vectors are computed from the estimations. The third step involves a supervised learning that finds a mapping between the feature vectors and the estimations of conditional probability distributions of the measurements. The learning algorithm of the third step is an improved version of the
“Conditional Distribution Discrimination Tree” (CDDT) algorithm whose purpose is to associate input vectors to conditional probability distributions of output vectors. The CDDT learning algorithm has been successfully applied to predict the sunspot number [7].

The paper is organized as follows: first, an overview on radio network prediction is given with a focus on 3G networks, followed by the description of the proposed method. Next, application of the proposed method to prediction of a 3G radio network is presented and finally concluding remarks are given.

II. RADIO NETWORK PREDICTION

Radio network prediction is mostly carried out by Radio Network Prediction (RNP) tools that perform Monte Carlo simulations of the Radio Access Network (RAN) according to predetermined traffic, mobility and service distributions, and a propagation model. They have two operating modes: static and dynamic. In the static mode the mobiles are positioned according to the traffic distribution and they are assigned services. Then, their conditions of access to the network are checked by calculating the UpLink (UL) and DownLink (DL) quality metrics and transmission characteristics such as transmission powers, signal-to-interference ratios, error rates, throughput etc. For performing these calculations, the RNP tool takes into account processes such as admission control, link control, load control etc. This procedure is repeated a number of times for successive independent snapshots. In dynamic mode, the simulator performs correlated snapshots to account for the time evolution of the network, with the following characteristics: after each time step that can typically vary between one to tens of seconds, the new mobile positions (due to mobility with different speeds) and the powers transmitted in the network are computed, as in a static simulator. For both modes of operation, indicator statistics are given at different entity levels: mobile, base station, network, operator etc.

In this work, we use a 3G radio network simulator in the static mode of operation. We are interested in two performance indicators: the UL Load (ULL) and the DL Load (DLL) whose expressions for an arbitrary base station \(m\) can be written as [1]:

\[
ULL_m = \sum_{i=1}^{K_m} \frac{1}{1 + \text{SIR}_{UL,i}} (1 + f)
\]

\[
DLL_m = \sum_{i=1}^{K_m} \left[ \text{SIR}_{DL,i} \left( 1 - \alpha_i \right) + \sum_{b=1,b \neq m}^{B_m} \frac{L_{bi}}{L_{bi}} \right]
\]

where \(K_m\) is the number of mobiles connected to base station \(m\), \(\text{SIR}_{UL,i}\) is the required UL (DL) SIR (signal-to-interference ratio) for mobile \(i\), \(\alpha_i\) is the orthogonality factor, \(L_{mi}\) is the path loss from the serving base station \(m\) to mobile \(i\), \(L_{bi}\) is the path loss from the neighbouring base station \(b\) to mobile \(i\), \(B_m\) is the number of neighbouring base stations of base station \(m\), and \(f\) is the proportionality factor between the powers received from the mobiles connected to other base stations, \(I_{oth}\), and the power received from the mobiles connected to base station \(m\) \(I_{mon}\), i.e. \(f = I_{oth}/I_{mon}\).

ULL and DLL are important indicators that show how much loaded the base stations are. The ULL is a measure of the amount of UL interference that the base station receives with respect to the thermal noise. The DLL measures the amount of transmit power allocated by the base station with respect to its maximum transmit power. The calculation of these two indicators by the RNP tool from expressions (1) and (2), does not yield exactly the same values observed in the RAN. The real values that correspond to ULL and DLL are reflected in the following measurement metrics:

\[
ULL_m \rightarrow 1-P_N/RTWP_m
\]

where \(RTWP_m\) (Received Total Wideband Power) indicates the amount of total received power at base station \(m\), and \(P_N\) is the thermal noise.

\[
DLL_m \rightarrow P_{tot,m}/P_{max,m}
\]

where \(P_{tot,m}\) is the total transmit power of base station \(m\), and \(P_{max,m}\) is the maximum transmit power of base station \(m\).

These measurements metrics are communicated between different entities of the RAN to ensure proper network operation. It is possible to make use of the relation between the computed \(ULL_m\) (DLL\(m\)) values and the measured \(1-P_N/RTWP_m\) \((P_{tot,m}/P_{max,m})\) values to decrease the discrepancy between them and obtain predictions that are closer to real values. In the following sections, we will describe a fast and easy-to-use learning system that achieves this aim.

III. THE CONDITIONAL DISTRIBUTION DISCRIMINATION TREE

The “Conditional Distribution Discrimination Tree” (CDDT) is a conditional probability distribution estimator of a multivariate random variable \(Y\) defined in an output space given a vector \(x\) defined in an input space. The conditional distribution of \(Y\) given a vector \(x\) is denoted by \(P(Y \mid x)\). The CDDT can be viewed as a supervised learning algorithm, which associates any input vector \(x\) to a probability distribution \(P(Y \mid x)\) thanks to a mapping \(x \rightarrow P(Y \mid x)\) built during a training phase. This training process uses a training set composed of \(s\) indexed examples of couples \((x_i, P(Y \mid x_i))\), where \(P(Y \mid x_i)\) is the desired conditional distribution of \(Y\) given \(x_i\).

Basically, it is easy to build an estimator of \(P(Y \mid x)\) from the couples \((x_i, P(Y \mid x_i))\), where all \(x_i\) are distinct and \(i\) belongs to the interval \([1, s]\). For example, we can proceed in two steps:

1. The input space is partitioned into regions in such a way that each of them contains one and only one sample vector \(x_i\). The region which contains \(x_i\) is denoted by \(R_i\). It is labeled by \(P(Y \mid x_i)\).

2. The estimation of the conditional distribution for any input vector \(x\) is obtained by the label \(P(Y \mid x_i)\) of a region \(R_i\) such that \(x\) is a vector of \(R_i\).

However, in most concrete cases, the distributions \(P(Y \mid x_i)\) are not given, only estimations \(\hat{P}_i(Y \mid x_i)\) are available. These
estimations can be biased or characterized by high variance. In this context, we have to accept that a region can contain several identical \( x_j \) with different \( \hat{P}_j(Y|x_j) \). To overcome this problem, the associated distributions to a same region \( R_i \) is obtained by mixing the distributions \( \hat{P}_j(Y|x_j) \) for all \( x_j = x_i \).

The quality of a distribution estimator can be measured by its mean squared error (MSE). Its value is the sum of its variance and of the square of its bias. However, it is known that attempting to reduce the variance of an estimator leads to an increase in its bias, and vice versa [8]. This property is known as the bias variance dilemma. So minimizing the MSE would consist in finding the best tradeoff between bias and variance. But, in the context of distribution estimation, the bias is not known because, obviously, the exact distribution is not given. For this reason, the quality of an estimator is generally evaluated during a validation phase where the MSE is computed with data samples that have not been presented for the estimator setup.

It is possible to reduce the variance of an estimator by increasing the size of the data sample used to build it. Thus, mixing the distributions \( \hat{P}_j(Y|x_j) \) as mentioned above when vectors \( x_j \) are equal has the virtue to reduce the variance without increasing the bias, leading thus to a reduction of the estimator MSE. If the distribution \( P(Y|x) \) is continuous, the mixing process of \( \hat{P}_j(Y|x_j) \) can be generalized by applying it when the distance between the vectors \( x_j \) are lower than a given threshold. Its value must be chosen in such a way that the bias of the estimator is not increased too much. This is obtained when the density function is constant or varies slightly for all points in a ball the radius of which is the threshold value. But, this value is a priori strongly problem dependent and moreover it is unknown because the exact conditional distributions are not given.

The CDDT approach implements another way to increase the data sample size to estimate a distribution without notably increasing the bias. It attempts to merge adjacent regions \( R_i \) into “super-regions” according to a criterion of distribution homogeneity: two regions \( R_i \) and \( R_j \) are merged if the distribution associated \( \hat{P}_j(Y|x_j) \) and \( \hat{P}_j(Y|x_i) \) are not significantly different according to a goodness of fit statistical test. Then, the distribution estimation associated to a super-region is obtained by mixing the estimations \( \hat{P}_j(Y|x_j) \) of the component regions. This process can be applied recursively by merging super-regions to form wider “super-regions” if the associated conditional distributions are not significantly different. In this way, it is possible to reach a good trade-off between the bias and the variance of the estimator, providing thus a good generalization ability for the learning algorithm.

The partitioning of the input space to define the regions \( R_i \) is implemented with a “tree structured vector quantization” algorithm [9]. The regions \( R_i \) are then associated to the leaves of the tree while the super-regions are associated to internal nodes. The union of the regions associated to all the leaves reachable from a given internal node is its associated super-region.

From the above discussion, the CDDT training algorithm proceeds in three stages:

1. Partitioning of the input space with a tree structured vector quantization algorithm from the set of input vectors \( x_i \). The regions obtained are associated to the leaves of the tree.
2. Labeling of every region \( R_i \), associated to the leaves by the estimations \( \hat{P}_i(Y|x_i) \).
3. Recursive pruning of the tree in such a way that, if the labels of the leaves having the same parent node are not significantly different, these leaves are removed and their parent node becomes a leaf with a label obtained by mixing the labels of the nodes removed.

When the training has built the tree, it is used as a decision tree to quickly find the leaf \( i \) associated to a given vector \( x \) and thus the estimation of the associated conditional distribution \( \hat{P}_i(Y|x) \). Such a tree is quite well balanced by construction. So finding an output distribution associated to an input vector has a computational complexity of the order of the logarithm of the tree size.

### IV. Integrating the CDDT System in a Radio Network Prediction Context

The task of enhancing radio network predictions using measurements, requires finding a mapping between event samples provided by the simulation and samples obtained by the measurements. Now, the CDDT training algorithm is able to learn a mapping between a set of input vectors and a set of distribution estimations for the outputs. So, there are two requirements in order to be able to use a CDDT system for radio network prediction enhancement:

- finding a way to produce input vectors for the CDDT system from simulation samples,
- computing distribution estimations, \( \hat{P}(Y|x) \) from the measurement samples to define the output labels.

#### A. The output labels

The second requirement mentioned above consists in building an estimator from a data sample. In fact, it is possible to envisage the direct use of the samples as the labels of the leaves of the CDDT instead of computing distribution estimations \( \hat{P}(Y|x) \) because the CDDT learning algorithm only needs:

- to test if two leaf labels are significantly different or not,
- to mix two or more labels at pruning.

These two operations are readily possible on data samples instead of distribution estimations. To test if two leaf labels are significantly different, the Kolmogorov-Smirnov statistical test [10] can directly be used on two data samples. Moreover, the mixture of several data samples obviously permits us to have an estimation of a mixture of distributions. Hence, handling data samples as output labels for the CDDT system is simpler and more reliable than building estimators for each of them. That is why the version of the CDDT system for the proposed scheme uses data samples as output labels.
B. Producing the input vectors from simulation data samples

The basic idea of producing input vectors for the CDDT system from simulation samples consists in estimating the probability distribution for every sample and extracting a feature vector from each estimation. The set of these vectors can then be applied to the inputs of the CDDT system.

Let $X_i$ be the multivariate random variable in $\mathbb{R}^n$ for the simulation sample $S(X_i)$ of the training set. A probability distribution $P(X)$ models the distribution of events in $S(X_i)$. A $m$-component feature vector $f_i$ should be determined in such a way that it is associated to one and only one distribution $P(X_i)$. Assume that the distributions are continuous, $f_i$ is defined as a vector of probability densities: $f_i = (f(x_1), ..., f(x_m))$ for $x_j \in \mathbb{R}^n$, where $f(x)$ is the density of $P(X)$ at point $x$. The indexed set of points $x_j$ must be the same for all the distributions $P(X)$. The definition of $f_i$ is similar if the distributions are discrete.

In the context of this work, the indexed set $\{x_j\}$ is chosen in a simple way: the $m$ points $(x_i)$ are randomly chosen according to the distribution estimated from the sample which is the union of all the simulation samples $S(X_i)$ of the training set. Preliminary experiments have shown that the component number $m$ is of the order of 30. More details are given in section V. Of course, this method is not optimal: $m$ should be minimized in order to save computation time. However, the gain in time would be negligible compared to the treatment of the numerous events per sample (10 000 for this RNP task) involved in the statistical tests during the training phase of the CDDT.

Often, only estimations $\hat{P}(X_i)$ of $P(X_i)$ are available. This is the case in the context of this study. So, feature vectors $f_i$ become multivariate random variables and, if the estimator quality is low, the complexity of the mapping $f_i \rightarrow P(Y \mid f_i)$ performed by the CDDT learning can be unnecessarily increased. No particular form is required for the estimators $\hat{P}(X_i)$ computed from the simulation samples, they should only be of sufficiently high quality. This requirement implies that the simulation sample sizes in the training set should be large enough.

V. RADIO NETWORK PREDICTION ENHANCEMENT USING THE CDDT LEARNING ALGORITHM

In order to evaluate the performance of the proposed scheme on radio network prediction, we have conducted experiments with 3G (UMTS) radio network data where the goal is to estimate the generalization error with the “leave one out” cross-validation, also known as the $N$-fold cross-validation. It consists of conducting a series of learning and test phases. For each learning phase, all samples of the available data set are used, with the exception of the one that is then used in the test phase. The generalization error is defined as the error made during this test phase. A couple "learning/test" constitutes a fold of the $N$-fold cross-validation. It performs as many folds as there are samples available. The overall generalization error is computed as the average of the generalization errors of each fold. The distribution of the generalization errors over the N-folds is also computed in the form of histograms. In the radio network prediction context, the input of the proposed method that undergoes a pre-processing to build feature vectors comprises of the simulation samples generated by Monte Carlo simulations, i.e. the prediction results of the RNP tool. These samples are composed of events that are 2D vectors ($ULL_m$, $DLL_m$). On the other hand, the outputs presented to the proposed scheme during the training phase are samples of 2D vectors ($1-P_{ULR}/RSSI_m$, $P_{tot}/P_{max}$) coming from measurements. In the test phase, the outputs computed by the proposed scheme for an input sample is a sample of 2D vectors representing couples that are statistically closer to ($1-P_{ULR}/RSSI_m$, $P_{tot}/P_{max}$) than ($ULL_m$, $DLL_m$).

The 3G network used in our experimentation is made up of 115 base stations. Each base station has one input pair (simulation ULL and DLL) and one output pair (measurement ULL and DLL) of data sample. For practical reasons, we have used synthetic measurement data that is produced by the RNP tool. To obtain the synthetic measurement data, we have modified the target UL signal-to-interference ratio: it is set to -20 dB for the simulation data and to -18 dB for the measurement data, accounting for a bias between field parameters and RNP parameters. For each base station, there are 10 000 Monte Carlo draws obtained from uncorrelated snapshots of the RNP tool.

In order to determine the correct size of the feature vectors extracted by the pre-processing step described in section IV, the variation the generalization error has been observed with increasing feature vector sizes. These observations have shown that the generalization error stabilizes starting from 32 components. In most cases, a few tens of components are sufficient for satisfactory performance. This size has little impact on the learning time and performance, if it is higher than the minimum required value. The performance criterion chosen for estimating the generalization error is the Kolmogorov-Smirnov distance between the desired data sample (measurement ULL and DLL) and the output sample of the proposed scheme (enhanced prediction ULL and DLL) for a test sample submitted to its input (simulation ULL and DLL).

VI. RESULTS

A. Computation times

The training system has been implemented using C++ programs on a computer running a dual-core 2GHz processor.

Before carrying out the cross-validation, the set of input feature vectors is computed from the set of simulation samples during a preprocessing phase. This one involves as many estimator setups as samples in the training set. Each of the 115 samples contains 10000 2D vectors. This preprocessing phase takes 405 seconds.

Then, the 115-fold cross-validation is launched for a global computation time equal to 30 seconds. That means that a learning task with a training set composed of 114 examples followed by one test takes 0.26 second.
B. Quality of the trainings

At the end of the N-fold cross-validations, the distributions of the KS distance values are calculated and put in the form of histograms. Figure 1 and 2 depict the KS distance histograms for the simulations (RNP tool outputs) and for the enhanced predictions (outputs of the proposed method) respectively.

Comparison of the two histograms reveals that the proposed scheme provides a significant improvement in the KS distance distributions of ULL and DLL: the average KS distance is decreased to one fourth of the initial value. In other words, the ULL and DLL distributions obtained by the proposed scheme are much closer to real distributions than the distributions of the RNP tool simulations.

A closer examination of the histogram in figure 2 shows that the generalization is relatively bad in a few cases in which the KS distance between the measurement sample and the enhanced prediction sample is greater than 0.5. Nevertheless, these cases are rare in the order of 2.6%.

VII. Conclusion

In this paper, we have proposed a method that builds a correspondence between data from a radio network simulator and data from field measurements having the same settings. The goal is to correct the simulator data by rendering it closer to measurements, without being obliged to change the structure of the simulator for each network configuration. This objective is achieved by the proposed CDDT (Conditional Distribution Discrimination Tree) algorithm which constructs a link between feature vectors of simulation data and multivariate distribution estimations of measurement data. The CDDT includes a pre-treatment phase that performs probability density estimation and feature vector extraction of the simulation data. The extracted feature vectors are then used to find a correspondence between the distributions of measurements and simulations. The performance of the proposed system is tested and evaluated on a data set coming from a 3G network of 115 stations. The metric used for evaluating the performance is the Kolmogorov-Smirnov (KS) distance that is a measure of stochastic similarity between samples of random vectors. We could afford to carry out an N-fold cross validation thanks to the fast learning achieved by CDDT. The results show significant improvement in generalization (average KS distance decreased to one-fourth). The CDDT proves that it is able to render distributions much closer to measurements than simulation tools.

REFERENCES