DIPARTIMENTO DI INFORMATICA
E SISTEMISTICA ANTONIO RUBERTI

SAPIENZA
UNIVERSITÀ DI ROMA

Path Loss Prediction in Urban Environment Using Learning Machines and Dimensionality Reduction Techniques

Mauro Piacentini
Francesco Rinaldi

Technical Report n. 11, 2009

Path Loss Prediction in Urban Environment Using Learning Machines and Dimensionality Reduction Techniques

M. Piacentini, F. Rinaldi

Dipartimento di Informatica e Sistemistica
Sapienza Universitá di Roma
Via Ariosto 25 - 00184 Roma - Italy

e-mail (Piacentini): piacentini@dis.uniroma1.it
e-mail (Rinaldi): rinaldi@dis.uniroma1.it

Abstract

Path loss prediction is a crucial task for the planning of networks in modern mobile communication systems. Learning machine-based models seem to be a valid alternative to empirical and deterministic methods for predicting the propagation path loss. As learning machine performance depends on the number of input features, a good way to get a more reliable model can be to use techniques for reducing the dimensionality of the data. In this paper we propose a new approach combining learning machines and dimensionality reduction techniques. We report results on a real dataset showing the efficiency of the learning machine-based methodology and the usefulness of dimensionality reduction techniques in improving the prediction accuracy.

Keywords Path Loss Prediction, Learning Machines, Dimensionality Reduction Techniques.

1 Introduction

The problem of predicting the propagation path loss frequently occurs in the design and planning of networks for communication systems (e.g. mobile systems, wireless-access systems). A prediction based on theoretical models is really important since it allows to determine optimum base locations without the need of any measurement. In order to plan a cellular system, several propagation methods have recently been developed. Prediction models can be divided into three classes: empirical, deterministic and semi-deterministic. Empirical models [14, 19] describe from a statistical point of view the relationship between the path loss and the environment. Results are usually obtained by means of measurement campaigns. In deterministic models [16, 24] the field strength is calculated using the Geometrical Theory of Diffraction (GTD). It is obtained as the superposition of direct, reflected and diffracted rays at the point of interest. Semi-deterministic models are half-way between deterministic and empirical models [8, 10]. Empirical models are easier to implement and usually require less computational effort but are less sensitive to the environment. Deterministic ones, on the other hand, are more accurate but require detailed information about the environment and more computational effort.
The prediction of propagation path loss can be basically viewed as a regression problem. In fact, information about transmitter, receiver, buildings, frequency, etc. represents the inputs and the propagation loss represents the output to be calculated (see Fig. 2). The goal is finding a suitable input vector x and an estimate $f(x)$ that best approximates the propagation loss.

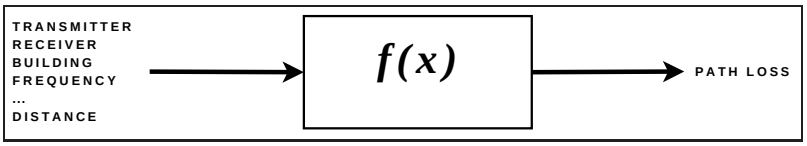

Figure 1: Path Loss Prediction as a Regression Problem.

Learning Machines, which are useful tools for solving regression problems, can be efficiently applied for obtaining a reliable prediction of wave propagation [1, 6, 7, 13, 20].
In regression, nothing is known about the function we want to represent. The only information available is in the inputs, or features, of the vectors x. As relevant inputs are unknown *a priori*, many candidate features are usually included in order to better describe the domain. Unfortunately, many of these features are *irrelevant* or *redundant* and their presence does not provide more discrimination ability. Furthermore, data sets with a large number of inputs and a limited number of training examples lead to the "curse of dimensionality": the data are very sparse and provide a poor representation of the mapping [5]. Then the only way to construct a good estimator f is to transform input data into a reduced representation set of features by using *Dimensionality reduction techniques* [5, 18]. Dimensionality reduction techniques are usually divided into two classes: linear methods (e.g. Independent component analysis (ICA), Principal Component Analysis (PCA), Singular value decomposition (SVD)) and nonlinear methods (e.g. Nonlinear PCA, Kernel PCA).
In this work we propose a two-step approach, which combines learning machines and dimensional reduction techniques, for predicting the propagation path loss in an urban environment. Once the input vector is built, a dimension reduction strategy is applied to obtain a new vector in a smaller space. Then a suitably trained learning machine is fed with reduced data in order

to obtain the path loss prediction (see Fig. 2).
We briefly describe the contents of the paper. In section 2 and 3 a brief overview of Artificial Neural Networks and Support Vector Machines is given. In section 4 we describe two dimensionality reduction techniques, namely PCA and nonlinear PCA. A new method for generating input data in path loss prediction is introduced in section 5. Finally, in section 6 we report the numerical results on a real test problem showing the usefulness of the new method in predicting wave propagation.

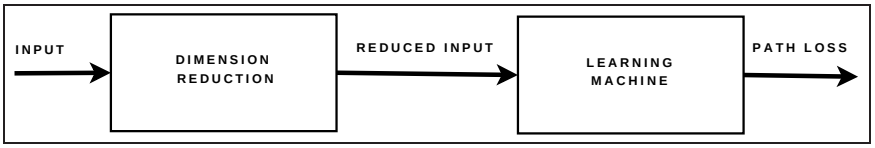

Figure 2: A Two-step Strategy for Path Loss Prediction.

2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are adaptive statistical tools that model the way biological nervous systems, such as the brain, process information. Similarly to people, ANN can learn by example, namely they can learn how to represent a given process just by using some examples related to it. As a result of their simplicity and flexibility, they have been successfully applied to tough problems (e.g. regression, pattern recognition) in a variety of different fields, such as engineering, econometrics, statistics, physics and medicine.
Generally, ANNs consist of several elementary processing units called neurons, which are located in different layers and interconnected by set of weighted edges (see figure 3). Each neuron transforms its input information into an output response by a nonlinear function g, called transfer function. Two well-known transfer functions are reported in Table 1. Overall, the inputs of the process are combined, propagated and processed through all layers and so converted as the output of the process.

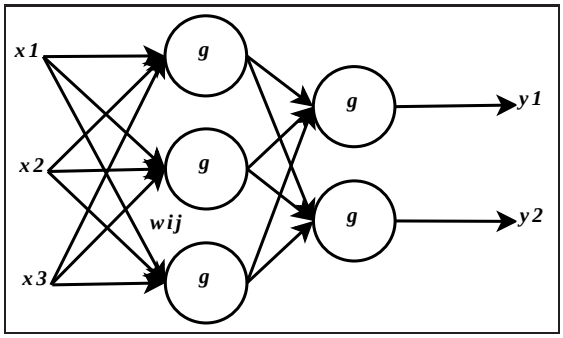

Figure 3: Neural Network Architecture.

One of the most popular architectures in neural networks is the multi-layer perceptron (see [5, 15]). A multilayer network typically consists of an *input layer*, which is basically a set of source nodes, one or more *hidden layers*, composed by various computational nodes, and an *output layer* of computational nodes. Once the architecture is chosen, the output of the network depends only on the weight vector w.
We consider a certain phenomenon described by a nonlinear regressive model of the following form:

$$\tilde{y} = f(x) + \nu$$

where f is an unknown function to be approximated and ν is an additive noise term statistically independent of the vector x.
The problem we deal with is a supervised learning problem. Formally, given a *training set*, namely a set of examples

$$TS = \{(x^i, y^i),\ x^i \in R^n,\ y^i \in R^k,\ i = 1, \ldots, N\},$$

we want to build an estimate of the dependence describing the phenomenon. This is equivalent to solve the following optimization problem

$$\min_{w \in \Re^m} E(w) = \sum_{i=1}^{P} E_i(w),$$

where each E_i measures the distance between the target output y^i and the network response for x^i, that is $y(w; x^i)$. A well-known function is the quadratic loss function, that is

$$E_i(w) = \frac{1}{2}\|y^i - y(w; x^i)\|^2,$$

although other types of structure for the error can be used.
We can construct (*train*) the desired network in a supervised manner by using a popular algorithm known as the *backpropagation algorithm* [21, 15]. Methods for large unconstrained optimization, such as Quasi Newton, Conjugate gradient and non monotone Barzilai-Borwein gradient methods [2, 12] represent another widely-used class of methods for solving the problem described above.
The hope is that the neural network obtained after the training process will *generalize*. A neural network is said to generalize well when it is able to compute correctly the input-output mapping for *test data* not included in the training set. The generalization ability of a learning machine, in general, is strictly connected with its complexity. In fact, a complex network usually approximates the process poorly on points not in the training set. Such a phenomenon is referred to as *overfitting* or *overtraining*. A model which is too simple, however, is also not preferred as it gives too poor a fit to the training data. In order to find the optimal complexity for our learning machine, we can utilize the *Occam's Razor*[4]. This model selection criteria favors the simplest model possible that still grants good performance on the training data. Finally, in order to evaluate the generalization ability of a learning machine, we can use a *cross-validation* procedure (see [22] for further details).

Type of T. Function	$g(t)$
Logistic	$\frac{1}{1+e^{-ct}}$
Hyperbolic Tangent	$\frac{1-e^{-t}}{1+e^{-t}}$

Table 1: Transfer Functions used in ANN.

3 Support Vector Machines for Nonlinear Regression: A Brief Review

Support Vector Machines (SVMs) were first introduced by Vapnik [23]. Like multilayer perceptrons and radial basis functions, SVMs represent an efficient tool for pattern recognition and nonlinear regression.
In the SVMs case, the estimate usually assumes the following form:

$$\tilde{y} = \sum_{j=1}^{m} w_j \phi_j(x)$$

where $\{\phi_j(x)\}_{j=1}^{m}$ is a set of nonlinear basis functions. The loss function [15, 23] used for determining the estimate is

$$L_\epsilon(\tilde{y}, y) = \begin{cases} |\tilde{y} - y| - \epsilon & |\tilde{y} - y| > \epsilon \\ 0 & otherwise \end{cases} \tag{1}$$

with ϵ a small value. The problem can be formally stated as follows:

$$\begin{aligned} \min \quad & \frac{1}{N} \sum_{i=1}^{N} L_\epsilon(\tilde{y}^i, y^i) \\ s.t. \quad & \|w\| \leq \alpha \end{aligned} \tag{2}$$

where $w \in R^m$, and $\alpha \in R_+$ is an arbitrarily chosen constant. It is possible, by introducing some slack variables, to reformulate problem (2) as follows:

$$\begin{aligned} \min_{\xi^i, \bar{\xi}^i, w} \quad & C \sum_{i=1}^{N} \{\xi^i + \bar{\xi}^i\} + \frac{1}{2}\|w\|^2 \\ s.t. \quad & y^i - w^T \phi(x^i) \leq \epsilon + \xi^i \quad i = 1, \ldots, N \\ & w^T \phi(x^i) - y^i \leq \epsilon + \bar{\xi}^i \quad i = 1, \ldots, N \\ & \xi^i, \bar{\xi}^i \geq 0 \quad i = 1, \ldots, N \end{aligned} \tag{3}$$

Type of SVM	$K(x, x^i)$	Parameters
Polynomial	$(x^T x^i + 1)^p$	p
Radial Basis	$\exp(-\frac{1}{2\sigma^2}\|x - x^i\|^2)$	σ
Two Layer Perceptron	$\tanh(\beta x^T x^i + \gamma)$	β, γ

Table 2: Kernels used in SVM. Sigmoid function does not satisfy Mercer's condition on all β and γ.

Then we consider the dual problem of (3):

$$\begin{aligned} \max_{\alpha^i, \bar{\alpha}^i} \quad & Q(\alpha^i, \bar{\alpha}^i) = \sum_{i=1}^{N} y^i(\alpha^i - \bar{\alpha}^i) - \epsilon \sum_{i=1}^{N} (\alpha^i + \bar{\alpha}^i) - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\alpha^i - \bar{\alpha}^i)(\alpha^j - \bar{\alpha}^j) K(x^i, x^j) \\ s.t. \quad & \textstyle\sum_{i=1}^{N} y^i(\alpha^i - \bar{\alpha}^i) = 0 \\ & 0 \le \alpha^i \le C \quad i = 1, \ldots, N \\ & 0 \le \bar{\alpha}^i \le C \quad i = 1, \ldots, N \end{aligned} \tag{4}$$

where ϵ and C are arbitrarily chosen constants, and $K(x^i, x^j)$ is the inner-product kernel

$$K(x^i, x^j) = \phi(x^i)^T \phi(x^j)$$

defined in accordance with the Mercer's condition [23]. In Table 2 we report three widely-used types of kernel functions. Once we solve problem (4), we use optimal values α^i, $\bar{\alpha}^i$ to determine the approximating function:

$$f(x, w) = \sum_{i=1}^{N} (\alpha^i - \bar{\alpha}^i) K(x, x^i).$$

We define support vectors as those data points for which we have $\alpha^i - \bar{\alpha}^i \neq 0$. Parameters ϵ and C control in some way the machine complexity. Anyway, complexity control in nonlinear regression is a very tough task and still represents an open research area.

4 Dimensionality Reduction Techniques

Learning Machine performance can be highly improved if some kind of preprocessing is applied to the raw data. In fact, a reduction in the input dimensionality can offset the effects of the *curse of dimensionality*: a learning machine with fewer inputs has fewer weight parameters to be estimated and owns better generalization properties.
Dimensionality Reduction is justified from the fact that the actual dimension may be larger than the intrinsic dimension (i.e. the minimum dimension that explains more the non-random variation in the population data). Then, the goal of *Dimensionality Reduction* is to transform input

data into a reduced representation set of features, while keeping as much relevant information as possible.
In this section, we describe two different dimensionality reduction techniques: *Principal Component Analysis* and *Nonlinear Principal Component Analysis*.

4.1 Principal Component Analysis

The principal component analysis, also known as Karhunen Loève Transformation, is a dimensionality reduction technique, which acts as a linear transformation of the original space into a new space of smaller dimension, while accounting for as much of the variability in the data as possible.
PCA in practice makes a projection along the directions where the data varies the most. These directions are determined by the eigenvectors of the covariance matrix corresponding to largest eigenvalues (see e.g.[5]). Moreover, eigenvalues are important because their magnitude can be used to estimate the intrinsic dimension of the data. Indeed, if r eigenvalues have a magnitude much larger than the remaining ones, it can be assumed r as the true dimension of the data.
Obviously, a linear technique is only able to catch linear correlations between variables. This is the reason why, when complex relations arise, this method usually overestimates the intrinsic dimension and fails to provide a compact representation of the data.

4.2 Nonlinear Principal Component Analysis using Autoassociative Neural Networks

When the phenomenon to be represented is highly complex, a general nonlinear mapping between the original variables and the new ones is considered, and *Nonlinear Principal Component Analysis* [17] is used.
NPCA ca be implemented using a cascade of two artificial neural networks: the first one to approximate a function $G(x)$, which maps the original space in the reduced space, and the second one to approximate its inverse mapping $F(y)$.
As $G(x)$ and $F(y)$ are unknown, the outputs of the training set for the first network are not available as well as the inputs for the second network. Then, a supervised learning of such networks is not possible. The idea is to use the outputs of the first net as the inputs for the second one. So, we obtain a new net given by the concatenation of the two networks. This network is usually called *autoassociative network* (see figure 4). Once the autoassociative network is trained, we use the first subnet to project the original space into a smaller space.
However, *NPCA* presents some drawbacks. First of all, autoassociative network has at least 4 layers, so that the learning phase becomes very difficult as the dimension of the space gets large. Secondly, despite an efficient trial and error, there is no way to decide the number of principal components. Moreover, the method does not explicitly provide a parameter (like eigenvalues in conventional *PCA*) to measure the contribution of each principal component to the data representation.

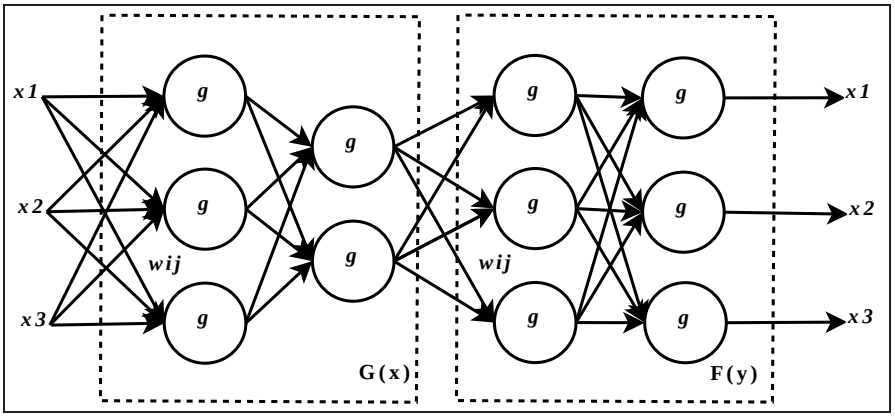

Figure 4: Autoassociative Neural Network.

5 A New Model for Path Loss Prediction

Learning machines, which represent a very powerful tool for solving regression problems, can be efficiently applied for obtaining a reliable prediction of wave propagation. As highlighted in [1], the key aspect when dealing with learning machines is the choice of the inputs.
In macrocellular models [16, 24], propagation loss is calculated as the sum of the free space path loss (L_0) and an attenuation term ($\alpha_{buildings}$) which takes into account the effect of the buildings:

$$L(db) = L_0 + \alpha_{buildings} = 32.4 + 20 \cdot \log(d) + 20 \cdot \log(f) + \alpha_{buildings}, \tag{5}$$

where d is the radio-path length (in km) and f is the radio frequency (in MHz). The attenuation term depends on several parameters, such as height of base station and receiver, distance between consecutive buildings, height of buildings, etc.
When designing our model, we need to use these parameters as inputs of our learning machine. In our approach the path from transmitter to receiver is divided into n intervals (n is set by user). For each interval the highest building is selected as the main obstacle and its features are included into the input vector (see Fig. 5). Inputs are divided into two groups:

1. **global inputs**: inputs related to the global path (i.e. distance between transmitter and receiver *TrxRx*, portion through the buildings *ptb*);

2. **interval inputs**: inputs related to each interval i of the path (i.e. height h_i, thickness l_i, distance from transmitter d_i).

Then the total number of inputs used (I_{tot}) is directly related to the number n of intervals:

$$I_{tot} = 2 + 3 \cdot n.$$

The portion through the buildings is defined as the portion of the straight line drawn between transmitter and receiver extending through the buildings (see Fig. 6).
However, when calculating the propagation loss, we do not consider the absolute height of the buildings, but their height relative to transmitter and receiver. Hence, the heights relative to the straight line linking transmitter and receiver are selected as inputs. In other words, if there

is no building exceeding this straight line, then there will be no obstacle to take into account. After the input vector is generated, a dimensionality reduction technique (i.e. PCA, nonlinear PCA) is used for extracting a new vector of smaller dimension (that carries as much as the information possible). The new vector is given to a suitably trained learning machine (i.e ANN, SVM) that determines the propagation path loss.

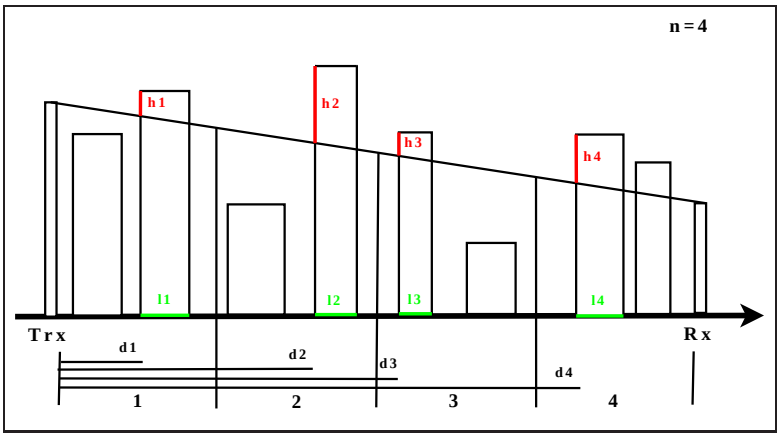

Figure 5: Interval Inputs. height h_i, thickness l_i, distance from transmitter d_i.

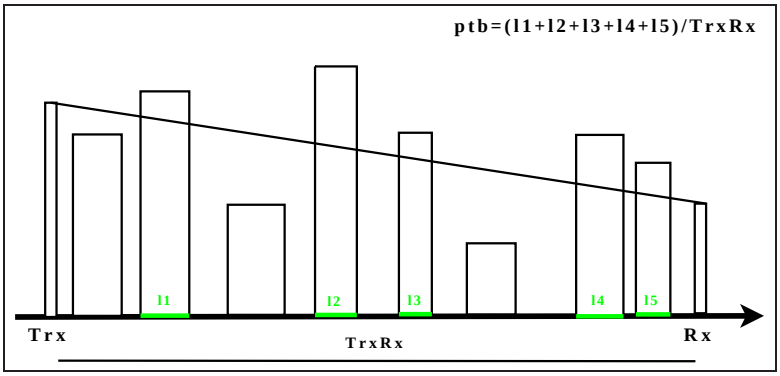

Figure 6: Global Inputs. Distance Trx-Rx (*TrxRx*); portion through the buildings (*ptb*).

6 Computational Experiments

In order to test the performance of our model, we used measurements from a campaign carried out in Munich (Germany). All measurements correspond to a frequency of 947 MHz. The transmitter ground altitude is 515m, the transmitter and receiver heights above ground are respectively 13m and 1.5m (see [9] for further details).
The dataset, composed of 2356 points, was split into two parts. 2000 points were used for training the selected learning machine; the remaining 356 points were used to evaluate the model.

Implementation details

1. Algorithms for dimensionality reduction techniques were implemented using *Fortran 90*;
2. SVMs were trained using *LIBSVM A Library for Support Vector Machines*, developed by the Machine Learning Group at National Taiwan University (see [11] for the details). The kernel selected for training SVMs was the radial basis one. The parameters C and σ have been determined by a standard cross-validation procedure;
3. A Multilayer Perceptron with 2 layers (a hidden layer with hyperbolic tangent transfer functions and a linear ouput layer) was chosen as architecture of the network for predicting the path loss. A standard cross-validation procedure was used for determining the architecture of the network (i.e. number of neurons in the hidden layer). A *Fortran 90* implementation of L-BFGS, a Quasi-Newton method with limited memory [3, 25], was used for training the networks.

All the experiments were carried out on Intel Pentium 4 3.2 GHz 512 MB RAM.

Results

In Table 3 we report the results obtained for both SVM and ANN, in terms of the mean squared error (**MSE**), using no dimensionality reduction. We indicate with **i** the number of intervals chosen for the model, and with **n** the number of related inputs. These preliminary results show the effectiveness of the learning-based approach. Furthermore, as we can easily notice analyzing the table, ANNs has slightly better performance than SVMs.
In Table 4-6 we report the results obtained for both SVM and ANN, in terms of the mean squared error, using PCA and nonlinear PCA over 5, 10 and 20 intervals dataset respectively. We indicate with **n** the number of features extracted from the original input vector. The results highlight the ability of the model that combines dimensionality reduction techniques and learning machines in reducing the original mean squared error. Moreover, learning machines trained over PCA-compressed data generally grant better performance than those ones trained over NPCA-compressed data. The best result is obtained with ANNs trained using 20 intervals PCA-compressed data with 50 extracted features (see Fig. 7).

i	**n**	**SVM**	**ANN**
5	17	60,36	60,25
10	32	63,04	62,16
20	62	69,92	66,81

Table 3: Results obtained for SVM and ANN using no dimensionality reduction.

5 Int	SVM		ANN	
n	PCA	NPCA	PCA	NPCA
5	58,11	71,33	**52,66**	79,7
10	**57,62**	67,36	61,15	**54,34**
15	61,03	**64,29**	62,30	65,14

Table 4: Results obtained over 5 intervals for SVM and ANN using dimensionality reduction techniques.

10 Int	SVM		ANN	
n	PCA	NPCA	PCA	NPCA
5	65,26	109,95	61,70	103,77
10	**56,88**	**56,88**	59,98	**53,25**
20	60,80	63,68	**54,34**	68,99
30	62,13	62,18	77,84	78,15

Table 5: Results obtained over 10 intervals for SVM and ANN using dimensionality reduction techniques.

7 Conclusions

In this work we have developed a new approach for the prediction of the path loss in an urban environment based on dimensionality reduction techniques and learning machines. Tests were designed in order to evaluate the effectiveness of dimensionality reduction in improving the path loss prediction accuracy as well as to compare performances of SVMs and ANNs on the regression problem. The ANN classifiers yielded slightly better results than the SVM classifiers. Furthermore, using dimensionality reduction before the prediction step led to a significant improvement of the learning machine accuracy. PCA-based prediction models generally granted better performance than NPCA-based ones.

References

[1] T. Balandier, A. Caminada, V. Lemoine and F. Alexandre, *170 MHz Field Strength Prediction in Urban Environments Using Neural Nets*, Proc. IEEE Inter. Symp. Personal, Indoor and Mobile Radio Comm., vol. 1, pp. 120-124, Sept. 1995.

[2] Bertsekas B., *Nonlinear programming*, Athena Scientific, Boston, 1999.

[3] Byrd R. H., Lu P., Nocedal. J. , *A Limited Memory Algorithm for Bound Constrained Optimization* , SIAM Journal on Scientific and Statistical Computing, 16, 5, pp. 1190-1208, 1995.

[4] A.Blumer,A.Ehrenfeucht,D.Hausler,andM.K.Warmuth, *Occam's razor*, Information Procesing Leters,24:377–380,1987.

[5] C.M.Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, NewYork,1995.

20 Int	SVM		ANN	
n	PCA	NPCA	PCA	NPCA
5	77,55	97,42	77,77	93,67
10	73,95	86,34	64,99	74,20
20	71,81	70,87	64,95	62,38
30	65,47	71,21	58,80	58,40
40	**62,50**	74,79	65,66	**58,26**
50	65,15	72,20	**51,15**	67,46
60	66,45	**66,95**	64,93	76,19

Table 6: Results obtained over 20 intervals for SVM and ANN using dimensionality reduction techniques.

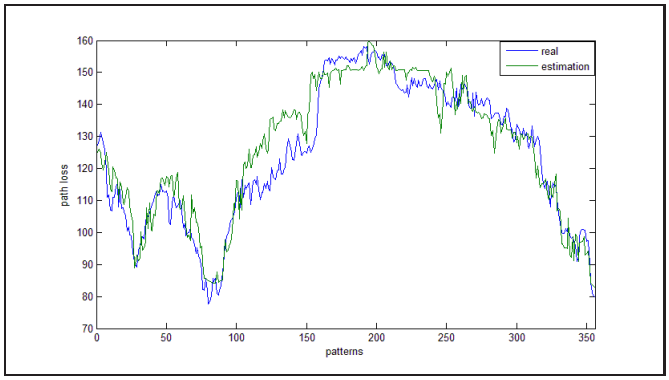

Figure 7: Comparison between measurements and prediction for the best case.

[6] N. Cardona, R. Fraile, *Macrocellular coverage prediction for all ranges of antenna heightusing neural networks.*, Universal Personal Communications, 1998. ICUPC '98. IEEE 1998 International Conference. vol.1, pp. 21-25 , 1998.

[7] P-R. Chang, W-H Yang, *Environment-Adaptation Mobile Radio Propagation Prediction Using Radial Basis Function Neural Networks*, IEEE Trans. Vech. Technol., vol. 46, No. 1, pp. 155-160, Feb. 1997.

[8] Chan G.K., *Propagation and coverage prediction for cellular radio systems.* IEEE Trans. on Vehicular Technology, Vol. 40, No. 4, pp. 665-670, nov. 1991.

[9] COST 231: **http://www2.ihe.uni-karlsruhe.de/forschung/cost231/cost231.en.html**.

[10] Dersch U., Braun W.R., *A physical radio channel model.* IEEE CH2944-7/91/0000/0289, 1991.

[11] Fan R.-E. , Chen P.-H. , and Lin C.-J. *Working set selection using the second order information for training SVM.* Journal of Machine Learning Research 6, 1889-1918, 2005.

[12] Grippo L., Sciandrone M., *Nonmonotone Globalization Techniques for the Barzilai-Borwein Gradient Method.* Computational Optimization and Applications, Volume 23 , Issue 2, pp. 143-169, Nov. 2002.

[13] B. E. Gschwendtner and F. M. Landstorfer, *Adaptive Propagation Modeling using a Hybrid Neural Technique*, Electronics Letters, Feb. 1996, vol. 32, No. 3, pp. 162-164.

[14] Hata M., *Empirical formula for propagation loss in land mobile radio services.* IEEE Trans. on Vehicular Technology, Vol. 29, No. 3, pp. 317-325, aug. 1980.

[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice-Hall, 1999.

[16] Ikegami F., Yoshida S., *Analysis of multipath propagation in urban mobile radio environment.* IEEE Trans. Antennas and Propagation, IEEE Trans. Antennas and Propagation, Vol. 28, NO. 4, pp. 531-537, 1980.

[17] Kramer M.,.*Nonlinear principle component analysis using autoassociative neural networks*, AIChE Journal, vol. 37, no. 2, pp. 233-243, 1991.

[18] Liu H., Motoda H., *Feature selection for knowledge discovery and data mining*, Kluiver Academic Publishers, 1998.

[19] Okumara Y., Ohmori E., Kawano T., Fukura K., *Field strength and its variability in VHF and UHF land mobile radio service.* Rev. Elec. Comm. Lab., Vol. 16, No. 9-10, pp. 825-873, 1968.

[20] Popescu, I. Kanstas, A. Angelou, E. Nafornita, L. Constantinou, P., *Applications of generalized RBF-NN for path loss prediction.* The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002. vol.1 pp. 484–488, 2002.

[21] Rumelhart D., Hinton G., Williams R., *Learning representations of back-propagation errors*, Nature, vol. 323, pp. 533-536, 1986.

[22] M. Stone, *A Cross-validatory choice and assessment of statistical predictions.* Journal of the Royal Statistical Society,36:111–147, 1974.

[23] Vapnik V., *Statistical Learning Theory.* Wiley, New York , 1998.

[24] Walfish J., Bertoni L., *A theoretical model of UHF propagation in urban environments.* IEEE Transactions on Antennas and Propagation, Vol. 36, No. 12, pp. 1788-1796, dec. 1988.

[25] C. Zhu, R. H. Byrd and J. Nocedal., *L-BFGS-B: Algorithm 778: L-BFGS-B, FORTRAN routines for large scale bound constrained optimization* , ACM Transactions on Mathematical Software, Vol 23, Num. 4, pp. 550-560, 1997.