New Methods for Estimating/Forecasting Link Bandwidths in 802.11b WLANs

Sridharan Muthuswamy, Ivan Marsic, A. Annamalai
Mobile Portable Radio Research Group (MPRG)
Virginia Polytechnic Institute and State University
432 Durham Hall (0350), Blacksburg, VA, 24061
Email: <smuthusw, annamala@vt.edu>, <marsic@caip.rutgers.edu>

Abstract—**Accurate assessment of the available link bandwidths in wireless networks is crucial for applications to negotiate, adapt and control quality-of-service parameters. In this paper we focus on new methods for estimating/predicting saturated throughputs (link bandwidths) in 801.11b wireless local area networks (WLANs). We propose the principle component analysis (PCA) and clustering approach for the estimation problem and the pattern modeling and recognition system (PMRS) and method of moment matching (MM) for the prediction problem. A trade off between the complexity and accuracy of the estimation/prediction algorithms is discussed and compared with the Neural Networks (NN) approach.**

I. Introduction

802.11b devices have become increasingly popular since they offer flexibility and convenience to mobile users in a wireless local area. Mobile users can realize the performance and speeds comparable to that of wired Ethernets. The most crucial issue slowing WLAN demand until now has been limited throughput which is primarily because of the time-varying nature of the wireless channel and multi-access communication. There has been significant effort in improving the performance of these 802.11 systems over the past few years. Adapting to the dynamics of the wireless link bandwidth (BW) is a frequently used approach to enhance the performance of applications and protocols [2]. BW is defined as the saturated throughput that can be achieved at the wireless link. Due to protocol and implementation overhead, this is slightly lower than the wireless link capacity, which is the maximum transmission bandwidth provided by the wireless link at the physical layer.

The 802.11b wireless system uses different modulation schemes at the physical layer at different data rates (2-FSK, 4-FSK at 1Mbps, 2Mbps and CCK at 5.5Mbps and 11Mbps). Also the 802.11b uses a RAKE receiver, with DFE (decision feedback equalizer). Further the data link layer uses forward error control codes and interleaving [3]. To add to the complication the wireless channel model is too complicated and not accurate. Also for estimating the BW of the 802.11b device at the MAC layer we have to consider queuing delays and media access control mechanism (CSMA/CA). To overcome these difficulties earlier works use round trip time measurements as reliable information for BW estimation with unjustified assumptions [1]. A more practical, passive method for BW estimation, non-intrusive to the wireless network is proposed in [4]. This method infers the maximum possible information about the link from *statistical models* of the data obtained from commercially available wireless network interface cards (NICs) such as signal level, noise level, and packet loss. NN and Bayesian estimators have been proposed for the estimation problem [4].

Due to dynamic channel sharing, fading and mobility, BW changes frequently and abruptly. Also wireless connection may experience cell handoff and even blackout due to user mobility [3]. For resource reservation in such environments, it is crucial to have the knowledge of the dynamics of BW ahead of time to perform admission control. A key component of this paper is end-to-end BW estimation/prediction for individual links. We adopt ideas from statistics viz. PCA and concepts from pattern recognition viz. clustering to estimate the BW. For predicting BW we apply ideas from pattern recognition viz. PMRS and concepts from stochastic processes viz. MM.

This paper is organized as follows. Section II presents a brief measurement setup. Section III describes estimation of BW using PCA. Section IV explains estimation of BW using the clustering approach. Section V describes error measures and gives accuracy vs. complexity trade offs of different estimation techniques. Section VI describes multiple forecasting of BW using PMRS technique. Section VII explains prediction of BW using MM method. Section VIII gives accuracy vs. length of prediction tradeoffs of combined estimation and prediction techniques. Section IX provides some concluding remarks.

II. Measurements in IEEE 802.11b

In our experiments, a constant bit rate (CBR) traffic generator runs on a wired sender (access point) and it keeps sending UDP traffic to the wireless receiver. To measure the maximum throughput that reflects the BW, we saturate the link by setting the CBR traffic rate to be slightly higher than the physical capacity of the wireless link as shown in Fig. 1.

0-7803-8521-7/04/$20.00 © 2004 IEEE

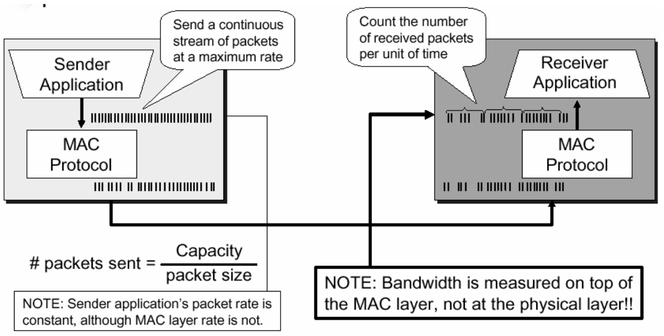
Send a continuous stream of packets at a maximum rate
Count the number of received packets per unit of time
Sender Application
Receiver Application
MAC Protocol
MAC Protocol
# packets sent = Capacity / packet size
NOTE: Sender application's packet rate is constant, although MAC layer rate is not.
NOTE: Bandwidth is measured on top of the MAC layer, not at the physical layer!!

Fig.1: Block schematic of the SNR-BW measurement process

The test bed has just one sender and one receiver to measure the saturated throughput in the absence of other contending users. This model however captures the effect of the unknown interference from other sources like a microwave oven located in an office space in the vicinity of the measurement set up or an adjacent cell user operating in the same frequency band. Every second, the wireless receiver reads the SNR from the wireless card driver. At the same time it calculates the BW of the received UDP traffic by multiplying the number of packets it has received during the one second period and the packet size as illustrated in Fig. 2. This experiment is repeated for different user mobility, terrain and weather conditions and interfering radio systems and the SNR, BW information collected and stored in a training data file. Several training data sets were collected. The times of day for the data collection were randomly distributed over the period of several weeks. All the recordings show similar relationship between SNR, BW as shown in Fig. 3.

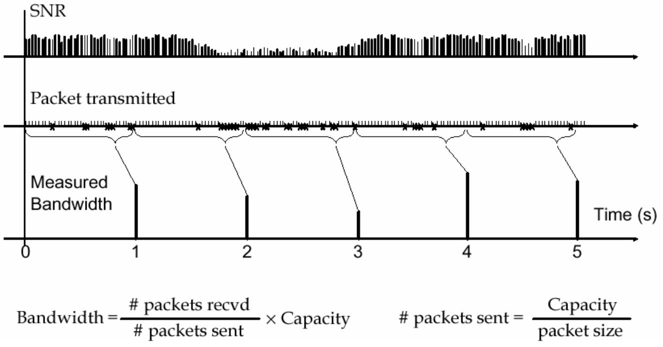
SNR
Packet transmitted
Measured Bandwidth
Time (s)
0
1
2
3
4
5
Bandwidth = # packets recvd / # packets sent × Capacity
# packets sent = Capacity / packet size

Fig. 2: Measuring SNR and BW

A mobile/wireless handset admitted to the network is provided with a copy of the training file. The handset can estimate BW in real-time by running its estimation algorithm using the reduced data file and SNR measured on the fly and monitor it for its adaptive applications. Ideally we would like to have *N*-step prediction of BW. Since in reality we can only observe SNR, we first predict *N* samples of SNR from already buffered past *K* SNR samples. We constrain *N* and *K* for different prediction algorithms in Section VI and Section VII. Using the *N* predicted SNR samples and the training file we estimate future *N* samples of BW using BW estimation algorithms. We need to have extremely accurate SNR prediction and BW estimation from predicted SNR since error propagation degrades performance. We use the clustering, Gaussian approximation (GA) BW estimation algorithm described in Section IV to estimate future BW samples.

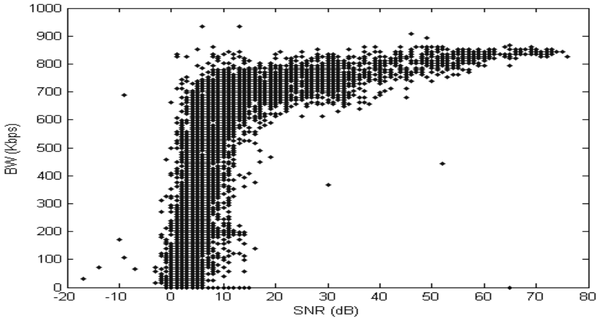

Fig. 3: BW (effective throughput) vs. SNR (signal –to-noise ratio)

III. PCA AND BANDWIDTH ESTIMATION

PCA is one of the useful techniques of multivariate statistics, being commonly used for the reduction of the dimensionality of datasets. The starting point of PCA is the spectral decomposition of a covariance matrix, with the objective of identifying only a few but most informative and mutually uncorrelated variables [5].

A. PCA on (SNR, BW) Training Data Set

Let $\underline{v}_1^{(t)} = \left\{ snr_0^{(t)}, snr_1^{(t)}, \ldots\ldots., snr_L^{(t)} \right\}$ be the vector of measured SNR and $\underline{v}_2^{(t)} = \left\{ bw_0^{(t)}, bw_1^{(t)}, \ldots\ldots., bw_L^{(t)} \right\}$ be the vector of measured BW using the training data set. The super script t denotes training data. The sample mean has been removed from these vectors. Form the matrix $V = \left[\underline{v}_1^{(t)}, \underline{v}_2^{(t)} \right]$. Matrix V has rank $r = 2$ and its "economy" singular value decomposition (SVD) [5] gives two orthonormal eigenvectors $\underline{w}_1, \underline{w}_2$ and singular values $\sigma_1 > \sigma_2$. The percentage variances: $p_1 = \sigma_1^2 / \left(\sigma_1^2 + \sigma_2^2 \right)$, $p_2 = 1 - p_1$ show that $p_1 >> p_2$. The percentage variance is a measure of the direction along which points are oriented. Inspecting Fig. 3 shows that the points are oriented along two prominent directions. Hence the training data set can be partitioned into two each one representative of the direction of clustering of points. The partition has been optimized (minimizing p_2 for both regions) and the percentage variance for the corresponding data set is tabulated in Table 1.

Range of SNR (dBm)	p_1	p_2
[-17,20]	0.9998	0.0002
[21, 76]	0.9996	0.0004

Table 1: SNR regions and the corresponding percentage variances

0-7803-8521-7/04/$20.00 © 2004 IEEE

B. Estimation using PCA

The orthonormal eigenvectors $\underline{w}_1^{(0)}, \underline{w}_2^{(0)}$ give the directions along which the points are clustered for the SNR region [-17, 20] (region I). The slope of the eigenvector $\underline{w}_1^{(0)}$, $s_1^{(0)}$ gives the dominant direction for this region I. Similarly the orthonormal eigenvectors $\underline{w}_1^{(1)}, \underline{w}_2^{(1)}$ give the directions for the SNR region [21, 76] (region II). The slope of the eigenvector $\underline{w}_1^{(1)}$, $s_1^{(1)}$ give the dominant direction for this region II. Suppose if $snr^{(d)}$ be the received SNR, where d denotes the data to be estimated and we need to estimate the BW corresponding to this SNR. The $snr^{(d)}$ is first mapped to the SNR region I or II. Let the two regions be defined by the indicator variable *J*. That is *J*=0 corresponds to region I and *J*=1 corresponds to region II. Let $bw^{(d,J)} = \left(s_1^{(J)}\left(snr^{(d)} - m_1^{(J)}\right) + m_2^{(J)}\right)$ and $u = (J+1)\,\mathrm{mod}(2)$

The estimate of the bandwidth can be compactly written as: $bw^{(d)} = (u)bw^{(d,0)} + (1-u)bw^{(d,1)}$ where $m_1^{(J)}$ and $m_2^{(J)}$ are the sample means of the SNR and BW training sequence vectors in region indicated by the indicator *J*.

PCA essentially fits a linear model to a non-linear SNR-BW relation and is a simple and fast technique. All we need to store are the SNR ranges and the slopes of the dominant direction of clustering of data points. A linear averaging *M*- filter is used to smoothen out the transitions. Fig. 4 shows the measured and estimated BW plot using PCA.

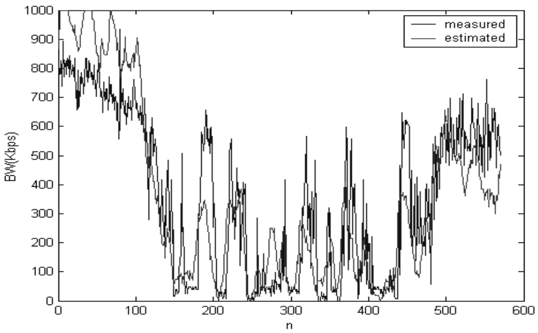

Fig. 4: Estimation of BW using PCA

IV. Cluster Formation and Bandwidth Estimation

A. The K-means Algorithm

The k-means algorithm for clustering in *N* dimensions produces k mean vectors that represent k classes of data. The algorithm relies on a distortion measure $d(\underline{x},\underline{y})$ between vectors in *N* dimensional real space, which could be a variety of norms like L_1, L_2, L_∞ or others specific to the problem [6]. We use the L_2 norm (Euclidean distance) as the distortion measure.

Let the set of training data be $X = \{\underline{x}_1, \underline{x}_2, \ldots\ldots, \underline{x}_L\}$. Given the cluster centroids $\underline{y}_i$, the set of points in *X* that is closer under the distortion measure d to $\underline{y}_i$ than to any other centroid is called the **Voronoi region** of $\underline{y}_i$ denoted as: $\Omega_i = \{\underline{x} \in X : d(\underline{x},\underline{y}_i) < d(\underline{x},\underline{y}_j), i \neq j\}$. The number of vectors in a Voronoi region is denoted as $|\Omega_i|$. The centroid of vectors in a Voronoi region is given by $\underline{y}_i = \frac{1}{|\Omega_i|}\sum_{\underline{x}\in\Omega_i}\underline{x}$. The centroid of the Voronoi region is used as the representative of all the data in the region. The algorithm can be stated as follows:

1. Choose an initial set of centroids $\{\underline{y}_1, \ldots\ldots\ldots, \underline{y}_k\}$ randomly.
2. Determine the Voronoi region for each $\underline{y}_i$.
3. Compute the centroid of each Voronoi region.
4. If the algorithm has not converged, go to step 2. Otherwise stop.

B. Clustering on (SNR, BW) Training Data Set

The k-mean clustering algorithm is applied to SNR and BW training data sets. Using the vector of measured SNR and BW $\underline{v}_1^{(t)}, \underline{v}_2^{(t)}$ as described in Section III, (A) form *N*-dimensional vectors as: $\underline{x}_i^{(t)} = \left[bw_i^{(t)}, snr_i^{(t)}, \ldots, snr_{i-N-2}^{(t)}\right]^T$ ($0 \le i \le L$-*N*-2). Let *C* be the number of clusters to be formed. Iterate the k-means algorithms on the set of vectors *X*=$\{\underline{x}_0, \underline{x}_1, \ldots\ldots\ldots, \underline{x}_{L-N-2}\}$ to generate *C* clusters.

C. Estimation using Clustering Approach

Suppose $snr_n^{(d)}$ be the n^{th} value of the received SNR. Form the following *N*-1 dimensional vector of SNR, $\underline{x}_n^{(d)} = \left[snr_n^{(d)}, snr_{n-1}^{(d)}, \ldots, snr_{n-N-1}^{(d)}\right]^T$. The past *N*-2 values of received SNR must be buffered. Generate *N*-dimensional vectors: $\tilde{\underline{y}}_i^{(t)} = \left[y_i^{(t)}, y_{i-1}^{(t)}, \ldots, y_{i-N-1}^{(t)}\right]^T$ by deleting the first component of the i^{th} centroid $\underline{y}_i^{(t)}$. Find to which Voronoi region the vector $\underline{x}_n^{(d)}$ belongs. That is $\underline{x}_n^{(d)} \in \Omega_i$ if $\left\|\underline{x}_n^{(d)} - \tilde{\underline{y}}_i^{(t)}\right\|^2 < \left\|\underline{x}_n^{(d)} - \tilde{\underline{y}}_j^{(t)}\right\|^2 \quad \forall i \neq j$. After determining to which Voronoi region the received SNR belongs we use the GA method to estimate BW as described in the following paragraph.

Consider the set of all vectors that belong to the Voronoi region Ω_j. Let $\mu_i^k = \frac{1}{|\Omega_i|}\sum_{\underline{x}_i^{(t)} \in \Omega_i} \left(bw_i^{(t)}\right)^k$, where $bw_i^{(t)}$ is the first component of the vector, $\underline{x}_i^{(t)}$. Generate a Gaussian random variable r with mean μ_i^1 and variance $\mu_i^2 - \left(\mu_i^1\right)^2$. The estimated BW is given by $bw_n^{(d)} = r$. Fig. 5 shows the measured and estimated plot using this GA method

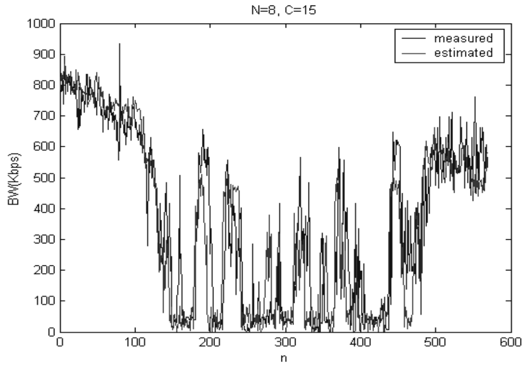

Fig. 5: Estimation of BW using GA Method

Estimation using the clustering approach shows interesting results. Fixing the number of clusters *C*, the number of dimensions of the vectors was increased from *N*=2 to *N*=20. It is observed in Fig. 6 that the mean relative error (described in next section) first drops, achieves a global minimum at *N*=8 and then increases. The increase in error with increasing *N* is because of negative correlation introduced by the past samples larger than *N*=8. Hence only past seven values of the SNR need to be considered in the BW estimation.

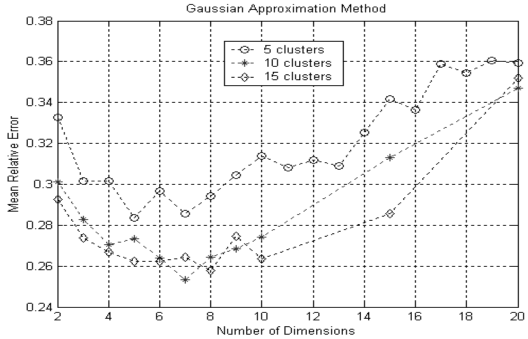

Fig. 6: Mean relative error vs. *N* for different clusters *C* for GA method

V. ACCURACY VS COMPLEXITY OF ESTIMATION METHODS

A. Error Measures

The accuracy of estimation can be quantified by a set of five error measures described below. There are two cases in estimation: either we estimate high or we estimate low. The estimation is scaled on the basis of the similarity of the match found. It is important to quantify the structural rather than the absolute differences between the two since we are primarily interested in studying the system behavior rather than minimizing a difference measure [7].

- Direction Error (positive) e_+: The total number of estimated series change in the positive direction (when $\tilde{y}_{i+1} > \tilde{y}_i$) is given as $\mu_+^{'}$. The total number of actual series changes in the positive direction (when $y_{i+1} > y_i$) is given as μ_+ ($n+1 \le i \le N$). The direction error (positive) is given as $e_+ = \frac{|\mu_+^{'} - \mu_+|}{\mu_+^{'} + \mu_+}$.
- Direction Error (negative) e_-: The total number of estimated series change in the negative direction (when $\tilde{y}_{i+1} > \tilde{y}_i$) is given as $\mu_-^{'}$. The total number of actual series changes in the positive direction (when $y_{i+1} > y_i$) is given as μ_- ($n+1 \le i \le N$). The direction error (positive) is given as $e_- = \frac{|\mu_-^{'} - \mu_-|}{\mu_-^{'} + \mu_-}$
- Average Proportional Error (APE) e_r: $e_r = \frac{1}{N}\sum_{i=n+1}^{N}\left(\frac{\tilde{y}_i}{y_i}\right)$
- Synchronization Error e_s: $e_s = \left(1 - \frac{a}{N}\right)$ where a is the total number of times the following conditions is satisfied: $\tilde{y}_{i+1} > \tilde{y}_i$ and $y_{i+1} > y_i$ or $\tilde{y}_{i+1} < \tilde{y}_i$ and $y_{i+1} < y_i$ ($n+1 \le i \le N$).
- Mean Relative Error $e_{mr} = \frac{\sum_i |y_i - \tilde{y}_i|}{\sum_i y_i}$ is an average measure of estimation error

The first two measures of error e_+ and e_- are self explanatory. Ideally, these two measures should be as close to zero as possible. The measure of average proportional error e_r records the average distance ratio between the estimated and actual value and ideally this should be as close to one as possible. Positive or negative divergence from one is equally bad. The measure of synchronization error e_s quantifies the synchronization between the estimated and actual behavior, that is do these profiles rise and decay in harmony. In practice, this measurement should be less than 50% representing that less than half of the time the two behaviors are out of synchronization (ideally the measure should have a zero value). The mean relative error is an average measure of the estimation error and should be close to zero.

The comparative study of the performance of the different estimation techniques viz. PCA, clustering and NN against these five different error measures is tabulated in Table 2 and Table 3 for a set of two randomly chosen test files.

Observation of the results shows that clustering, GA method gives the best accuracy. There is an improvement of about 1%-7% in the mean relative error over the NN method being currently used. This is because we have used past seven values to estimate the current BW in the clustering, GA method. The past history gives more information hence better accuracy in the results. It is also observed that the PCA method is comparable to the NN in terms of accuracy. This is because NN fits an optimal curve through the BW-SNR clutter plot and PCA optimally segments the clutter plot into two regions and fits a line to each region optimizing the total mean square error in each region.

Method	e_+	e_-	e_r	e_s	e_{mr}
GA	0.0424	0.0402	1.0725	0.5165	0.1394
NN	0.0201	0.0168	1.3472	0.5619	0.1482
PCA	0.1378	0.0953	1.2257	0.6851	0.2464

Table 2: Test file: InOrinFF1224_001.txt.mat

Method	e_+	e_-	e_r	e_s	e_{mr}
GA	0.0320	0.0281	1.3998	0.5272	0.2610
NN	0.0998	0.0692	2.4071	0.5903	0.3358
PCA	0.1049	0.0724	1.4296	0.6422	0.3383

Table 3: Test File: indoorC7Ar001.txt.mat

B. Complexity Measure

In this section we compare the computational complexity of the different estimation techniques in the estimation period. The training period can be usually carried offline and hence is not of much importance in assessing the computational needs of the different methods. Hence we focus on the estimation period. For the clustering approach given *C* clusters and past *N*-1values of SNR the number of multiplications is given by *N*m=*C*(*N*-1) and number of additions *N*a=*C*(2*N*-3). The optimal value of (*C*, *N*) is found to be (15, 8). For the PCA method *N*m=2 and for an FIR filter of length *M*, *N*a=*M*-1. The optimum *M* varies for each test file (usually $5 \le M \le 10$). For a 3 layer Neural Network *N*m=8 and *N*a=6. The results are tabulated in Table 4 for (*C*, *N*) = (15, 8) and *M*=10.

	Clustering	Neural Networks	PCA
Multiplications	105	8	2
Additions	195	6	9

Table 4: Comparison of complexity of different estimation techniques

VI. PMRS and Bandwidth Forecasting

A. Multiple Forecasting

Multiple forecasting refers to the process of making more than one prediction into the future. It should be noted that *N*-step multiple forecasting is different from *N*-step prediction. In *N*-step prediction, the system is trained to predict *N* steps on selected past actual values [7]-[11]. In multiple forecasting, past predictions generate future predictions. In such a process, the n^{th} future prediction $\widetilde{y}_n$ is based on the $(n-1)^{th}$ future prediction $\widetilde{y}_{n-1}$ rather than the $(n-1)^{th}$ actual value y_{n-1} since we do not have information about the actual value. We use the PMRS [7]-[8] method for *N*-step multiple forecasting of the SNR and use this to estimate *N* future BW samples.

B. Local Approximations using PMRS for the SNR Data

PMRS is based on matching neighborhoods in a time series data. The size of the neighborhoods matched is a parameter that is optimized empirically. Previous studies with PMRS [7]-[8] show that for single forecasts, it provides highly accurate predictions. We extend this algorithm to multiple forecasting.

If we represent the SNR time series $\underline{y}^{(t)} = \{snr_1^{(t)},...,snr_n^{(t)}\}$ which corresponds to the test data, then the current state of size one of the time-series is represented by its current value $snr_n^{(d)}$. In order to illustrate the matching process for series prediction, consider the SNR time series measured till time sample n. Form the segment defined as a difference vector $\underline{\delta}^{(d)} = \left(\delta_1^{(d)},...,\delta_{n-1}^{(d)}\right)$, where $\delta_i^{(d)} = snr_{i+1}^{(d)} - snr_i^{(d)}$ $1 \le i \le n$-1. In order to define any segment mathematically, we choose to tag the time series of SNR with a vector of change in direction. For this purpose, a value $snr_i^{(d)}$ is tagged with a '0' if $snr_{i+1}^{(d)} < snr_i^{(d)}$ and as a '1' if $snr_{i+1}^{(d)} \ge snr_i^{(d)}$. Formally, the segment is now represented as a pattern of '1's and '0's given by the vector $\underline{\rho}^{(d)} = \left(b_1^{(d)},...,b_{n-1}^{(d)}\right)$ where b is a binary value. Similarly form the string of '1's and '0's for the training data $\underline{\rho}^{(t)} = \left(b_1^{(t)},...,b_{K-1}^{(t)}\right)$ where $K >> n$. The algorithm for finding a match and predicting *N* step is described below:

1. Set $match = n$ and $i = 1$
2. Compute $new\_match = sum\left(XOR\left(\underline{\rho}^{(d)}, \left(b_i^{(t)},......b_{i+n-2}^{(t)}\right)\right)\right)$
3. If $new\_match < match$ then $match = new\_match$
4. If $match = 0$ then $position = i + n$, stop; else $i = i + 1$ and go to step 2
5. Compute the following sequences: $sign = \left\{2b_j^{(t)} - 1\right\}$ and $delta = \left\{\delta_j^{(t)}\right\}$; $position \le j \le position + N - 1$
6. Compute $beta = \dfrac{1}{N}\sum_{i=1}^{N}\left(\dfrac{\delta_{n-i}^{(d)}}{\delta_{position-i}^{(t)}}\right)$
7. Set $temp = snr_n^{(d)}$ and $j = 1$
8. $\widetilde{y}_{n+j} = temp + (beta)(sign_j)(delta_j)$

9. If $j = N$ then stop; else $j = j + 1$, $temp = \tilde{y}_{n+j}$ and repeat step 8
10. As new samples become available form $\underline{\rho}^{(d)}$ from the most recent past n SNR values and repeat step 1. If data gets exhausted, stop.

The error performance for BW prediction using the clustering GA approach for the PMRS method is tabulated in Table 5. The results show that PMRS can predict BW up to *N*=7 step ahead with acceptable error performance using an optimal pattern size of *K*=15 determined by simulations.

e_+	e_-	e_r	e_s	e_{mr}	(*K*,*N*)
0.0644	0.0594	1.1519	0.5111	0.2638	(2, 1)
0.0702	0.0660	1.2727	0.5512	0.2972	(15,7)

Table 5: Error performance for the predicted BW using PMRS method

VII. Moment Matching and Bandwidth Prediction

A. Prediction of SNR using Moment Matching

A non-stationary sequence (SNR) can be approximated as a stationary sequence to a first order by taking differences of adjacent samples of the sequence [6]. Suppose we need to predict *N* samples of SNR and have observed *K*=*N*+1 samples of SNR. Form the difference sequence $\underline{\delta} = \{\delta_1, \ldots, \delta_N\}$ where $\delta_i = snr_{i+1} - snr_i$ for $1 \leq i \leq N+1$. The $\underline{\delta}$ sequence can be approximated as a stationary sequence. Let $\underline{\tilde{\delta}} = \{\tilde{\delta}_1, \ldots \tilde{\delta}_N\}$ be the SNR samples to be predicted. Since the $\underline{\delta}$ sequence is stationary the *N* sample moments of $\underline{\delta}$ sequence must equal the *N* sample moments of $\underline{\tilde{\delta}}$ i.e. $\left(\tilde{\delta}_1^i + \ldots + \tilde{\delta}_N^i\right)/N = \left(\delta_1^i + \ldots + \delta_N^i\right)/N = s_i$, $1 \leq i \leq N$. We obtain a set of *N* power sum equations in *N* unknowns and adopt the Newton's identities [11], to solve for the unknowns. First form the polynomial $f(\lambda) = \lambda^N - p_1 \lambda^{N-1} - \ldots - p_N$ where $p_k = \frac{1}{k}\left(s_k - \sum_{i=1}^{k-1} p_i s_{k-i}\right)$ and $\{\tilde{\delta}_i\}$ are the roots of $f(\lambda)$=0. The predicted SNR samples are given as follows: $\tilde{s}_{N+2} = snr_{N+1} + \tilde{\delta}_1$ and $\tilde{s}_{N+i} = \tilde{s}_{N+i-1} + \tilde{\delta}_{i-1}$; $1 \leq i \leq N+1$.

N	e_+	e_-	e_r	e_s	e_{mr}
2	0.0571	0.0528	1.6114	0.5148	0.2838
5	0.0498	0.0592	1.7721	0.5411	0.3211
10	0.0601	0.0594	3.1148	0.5662	0.3642

Table 6: Error performance for the predicted BW using MM method

The error performance for the MM method is tabulated in Table 6. It is observed that we can predict up to *N*=10 step in MM method with acceptable error performance.

VIII. Conclusion

In this paper we have given new methods to enhance the performance of BW estimation with respect to the accuracy of the estimation such as the clustering and PCA approach and also considered prediction models for BW such as PMRS and MM methods. The past values of the observed SNR give more information and optimally utilizing these give better estimation of BW. The clustering, GA approach is better and PCA is comparable to NN for most test data files. In respect to the computational complexity PCA is the simplest and clustering method more complex than the rest. For the prediction problem we first predict SNR samples and then use the predicted SNR to estimate future BW samples since SNR is the only observable. We can predict *N*=7 step using the PMRS and *N*=10 step using MM method with acceptable error performance. PMRS is attractive for simple hardware implementation using digital logic circuitry since it searches for a matching pattern in a sequence of '1's and '0's. The primary disadvantage in PMRS is that error propagation is severe. Efficient routines exist for implementing MM method. If the sequence of first differences of SNR were strictly stationary we could have theoretically predicted *N*=∞ step. The sequence of first differences is stationary only over a few time samples since the SNR is a strictly time-varying signal.

References

[1] Sridharan Muthuswamy, "Wireless Link Bandwidth Estimation and Prediction in 802.11b WLAN: Practical Considerations", *M.S. Thesis*, Electrical and Computer Engineering, Rutgers University, October'03.

[2] R. Katz, "Adaptation and Mobility in Wireless Information Systems," *IEEE Personal Communications*, vol. 1, no. 1, pp. 6–17, 1994.

[3] IEEE 802.11b Wireless LANs, Technical Report 3Com; http://www.wlana.com

[4] L. Cheng and I. Marsic, "Lightweight Models for Prediction of Wireless Link Dynamics in Wireless/Mobile Local Area Networks," *Proceedings of the IEEE Sarnoff Symposium on Advances in Wired and Wireless Communications*, pp. 98-101, NJ, USA, March 13, 2002.

[5] S.J. Orfanidis, "Optimum Signal Processing", 2nd Edition, McGraw-Hill, New York, 1988

[6] Todd K. Moon, Wynn C. Stirling, "Mathematical Methods and Algorithms for Signal Processing", Prentice-Hall, Upper Saddle River, New Jersey, USA, 2000

[7] S. Singh, "Multiple Forecasting Using Local Approximations", *Pattern Recognition*, vol. 33

[8] S. Singh and E. Stuart, "Pattern Matching Tool for Forecasting", Proceedings 14th International Conference on *Pattern Recognition ICPR'98, Brisbane*, *IEEE Press*, vol. 1, pp. 103-105, 1998

[9] B. S. Motnikar, T. Pisanski, and D. Cepar, "Time Series Forecasting by Pattern Imitation or Spektrum", 18(1), pp. 43-49, 1996

[10] S. Delurgio, *Forecasting: Principles and Applications*, McGraw-Hill, 1998

[11] M. E. Azoff, *Neural Network Time Series Forecasting of Financial Markets*, John Wiley and Sons, 1994

0-7803-8521-7/04/$20.00 © 2004 IEEE