

# ASTRA: Application of Sequential Training to Rate Adaptation

Hui Liu, Jialin He, Pengfei Cui, Joseph Camp, and Dinesh Rajan  
Electrical Engineering, Southern Methodist University  
{huil, jhe, pcui, camp, rajand}@smu.edu

**Abstract**—The application of machine learning algorithms in wireless communications has attracted increasing attention due to the promising performance gains recently achieved. Static classification algorithms have been successfully applied to training protocols that adapt transmission parameters according to context information. However, in reality, there are many time-varying reasons for fading channel quality including mobility of sender, receiver, and/or obstacles within the environment. Moreover, time-varying noise further exacerbates the dynamics of the channel. These problems pose new challenges for the application of static classification algorithms in context-aware algorithms and suggest that sequential classifiers which leverage the temporal dynamics and correlation of context information might be more appropriate. In this paper, we apply sequential training to rate adaptation (ASTRA), leveraging the temporal correlation of context information. In particular, linear and non-linear sequential coding schemes are used in the training process for selecting the modulation/coding rate that achieves the highest throughput for the given context. Experimental results on measurements from emulated and in-field channels demonstrate that ASTRA can significantly increase the accuracy of selecting these target rates by up to 175% and increase the resulting throughput by up to 66% over rate adaptation training which uses static classifier-based methods.

## I. INTRODUCTION

Wireless channels are known to have time-varying quality, especially in mobile and vehicular networks. In such scenarios, algorithms attempt to adapt the transmission rate to by measuring either the packet losses [1], [2] or the channel quality [3], [4], [5]. As channel fluctuations increase, the ability to converge to optimality becomes more and more elusive [6]. Thus, recent works have proposed using the context information and machine learning to quickly converge to optimality [7], [8].

Context-aware rate adaptation schemes attempt to leverage existing patterns in the collected context information to adjust the transmission parameters to improve performance. Examples of such schemes include neural networks and genetic algorithms for parameter adaptation in cognitive radio networks [9], [10], distributed classification with data from different sensors [11], and static classification-based rate adaptation [7]. Existing works in context-aware rate adaptation have mainly focus on operations on static sets of attributes. these works treat measurements from sensors as independent and identically distributed data points, using them to infer decision rules.

However, the mapping of context information with network performance in the field is not *static*, because in-field channels and contexts change over time. Since the fluctuating channel state cannot sensibly be represented as a fixed set of measurements, it is not sufficient to simply determine when

classification is obsolete. Rather, the patterns changing over time can be used as clues for the cause of the channel variation. Learning these patterns can help rate adaptation mechanisms to better adapt to dynamic channels. Moreover, when noise is introduced to the transmitter or receiver, an awareness of temporal patterns can lessen its effect.

The temporal correlation embedded in context information has not yet been solved. In this paper, we design, implement and test an algorithm on the application of sequential training to rate adaptation called ASTRA. ASTRA leverages the temporal correlation among context information to improve the performance of rate adaptation. In particular, linear and non-linear sequential coding schemes are proposed to capture and exploit these temporal properties. To the best of our knowledge, our work is the first to exploit temporal information for rate prediction and adaptation. Our main contributions are summarized as follows:

- We qualify the importance of temporal information to predict which rate will achieve the highest throughput for the given context and form a sequential classification-based model for adaptation. ASTRA exploits the temporal correlation in the training data, building a decision structure which can then predict the best transmission mode based on the current contextual measurements.
- We survey the performance of a series of widely-used static classifiers on contextual measurements. We compare the performance of different classifiers on accuracy of rate prediction in different situations using two platforms, a custom FPGA-based platform and an off-the-shelf platform.
- We implement linear and non-linear sequential coding schemes for the purposes of rate adaptation (ASTRA-L and ASTRA-N, respectively). We then demonstrate the advantage of the non-linear coding scheme over the linear coding scheme in exploiting temporal information to improve the performance of rate adaptation.
- We verify the proposed ASTRA rate adaptation schemes on measurements from emulated and in-field channels to demonstrate the significant impact of utilizing temporal information in rate adaptation. ASTRA-N improves the accuracy and throughput of rate adaptation by up to 175% and 66.11%, respectively.

While in this paper we focus on the application to rate adaptation to show gains, ASTRA has other possible applications to transmission parameter adaptation based on context, such as transmission power control. The remainder of the paper is organized as follows. We introduce our sequential

classification-based framework for rate adaptation, ASTRA, as well as certain sequential coding approaches in Section II. In Section III, we compare the performance of various static classifiers in classification-based rate adaptation. Then, we compare ASTRA-N and ASTRA-L with the static classification-based method on emulated channels. In Section IV, we evaluate ASTRA-N based on measurements from in-field channels. Finally, we discuss related work in Section V and conclude in Section VI.

## II. SEQUENTIAL CLASSIFICATION MODEL

In this section, we exploit temporal information embedded in contextual data and develop linear and non-linear sequential coding based classification algorithms to train rate adaptation protocols for dynamic environments. The proposed ASTRA increases the accuracy of selecting the rate which has the highest throughput for a given context.

### A. Problem Formulation

The context information that we utilize in this paper are the channel type, node velocity and SNR. In this paper, we define that in the same channel type the effective performance of various transmission modes exhibits similar behavior for various values of other context attributes. Many factors (e.g., multi-path, path loss, and shadowing) have a substantial influence on the characteristics of the channel type. However, SNR and velocity fluctuate frequently while channel type remains unchanged in a small time scale and changes in a larger time scale. We use the suite of ITU channels, which are widely-accepted as representative channel types for urban and suburban settings [12]. Moreover, velocity gradually changes in the field. For example, it is far more likely to find a situation in which the velocity changes from 30 km/h to 35 km/h over a second rather than a sudden shift in velocity from 30 km/h to 120 km/h. This gradually change ensures that contextual measurements have an embedded temporal correlation. Using context information and this temporal correlation, we adapt the transmission mode for each scenario based on the desired performance metric (throughput, in our case).

Traditional static classifiers aim to find a mapping function,  $f$ , to predict the optimal transmission mode with the highest throughput as represented by:

$$f : \{Channel\ type, SNR, Velocity\} \rightarrow optimalmode, \quad (1)$$

This function,  $f$ , inherently assumes that the data points with different timestamps are independent. However, in our application, we consider the temporal correlation of contextual data points, which can be exploited to further improve performance of the classification-based rate adaptation. The proposed method can be represented using a simple two-stage model. In the first stage, we perform sequential coding, which can be formally stated as finding a sequential coding function,  $g$ , to capture the temporal correlation among contextual measurements as represented by:

$$vector_{seqcod} = g(Channel\ type, SNR, Velocity). \quad (2)$$

In the second step, we perform static classification on the new representation of the input vector after sequential coding. With

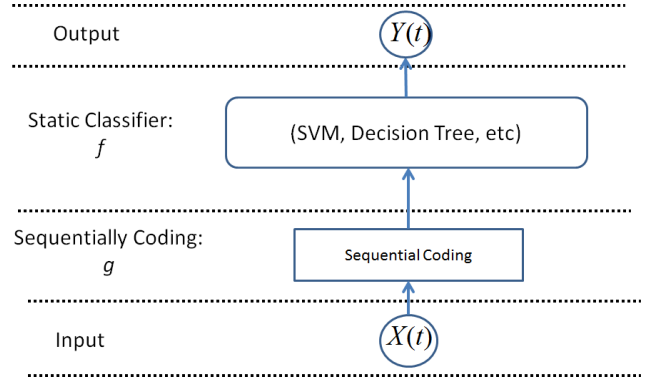


Fig. 1. ASTRA framework.

the notations and definitions above, the problem solved by ASTRA can be formulated as:

$$f : \{g(Channel\ type, SNR, Velocity)\} \rightarrow optimalmode. \quad (3)$$

Now, we formulate the problem with a generalized mathematical representation. Let  $X(t) = [x^1(t), x^2(t), \dots, x^m(t)]$  be the data at time  $t$  where  $m$  is the number of attributes. In our application,  $m$  equals 3, corresponding to channel type, SNR, and velocity. Each attribute is a variable which can be represented by a numerical value where  $x^i(t)$  represents the value of the  $i^{th}$  attribute at time  $t$  and  $Y(t)$  is the corresponding optimal transmission mode as prediction target at time  $t$ . Here, we define the optimal mode as the modulation/coding rate that achieves the highest throughput for the given context information set. Fig. 1 depicts the main scheme of ASTRA. The figure can be represented by the following notation:

$$f : \{g(X(t))\} \rightarrow Y(t). \quad (4)$$

### B. Background: Static Classification

In our rate adaptation model, the collected context information (channel type, SNR, and velocity) is the input to the sequential coding block. Also, each of the modulation/coding rates will be numbered as the set of available transmission modes. In the training set, for each given set of channel type, SNR and velocity values, the labeled mode represents the rate with the highest throughput. The static classifier extracts the relationship between the output of the sequential coding block and optimal modes.

Before considering the temporal information, we have implemented the static classification-based rate adaptation algorithm, in which we directly use the classification algorithm to extract the relationship between context information and the target mode without considering the temporal information. If the context information is classified into the appropriate category, the transmitter can avoid poor settings and quickly converge to optimality.

We consider 3 different static classification schemes, namely, Support Vector Machine (SVM), and Adaboost in our experiments [13], [14]. SVM and Adaboost have good performance and are used in many commercial applications, especially for binary classification problems. In contrast, a decision tree is able to choose subsets of available attributes

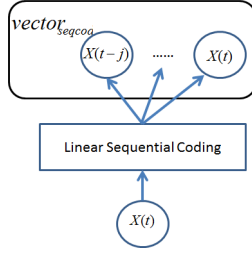


Fig. 2. Linear sequential coding.

when there are abundant attributes. In the static classification-based algorithm, we use the C4.5 algorithm, which has been used in commercial applications to implement decision trees [15].

### C. Sequential coding

Sequential coding attempts to code the temporal information in continuous contextual data. In this work, we investigate two sequential coding approaches: linear coding and habituation, one of the most widely-used non-linear coding schemes.

1) *Linear Coding*: Static classifiers are frequently used to classify data at one point in time. However, when the variation of data over time is correlated with the data values, the correlation of current data with the historical data is important in classification. In this situation, the classification performance can be improved if the classifier takes the data collected in the past  $j$  time units into consideration.

In a linear coding scheme, as an alternative representation of the data, the historical data in the past  $j$  time units and the data at the current time are treated as input to predict  $Y(t)$ . Linear coding can be represented by:

$$f : \{X(t-j), X(t-j+1), \dots, X(t)\} \rightarrow Y(t), \quad (5)$$

The static classifier is one instance of a sequential classifier based on linear coding with  $j = 0$ .

With linear coding, sequential classification utilizes the same set of data as the static classification but with a different order to the data. Since the measurements over a period of time can reflect sudden changes of measurements, linear coding can improve the reliability of classification when errors and noise occur in the data. A known problem with classical linear sequential coding is that it treats the past data equally as current data. However, past data has less effect on the current performance than current data. A weighting process only changes the values of the data rather than its *effect* on classification, because the classifiers would not distinguish the effect of two attributes on classification results based on their order of magnitude. Also, fluctuations of measurements due to external reasons can not be represented as a linear model in most situations. In other words, non-linear coding (described below) is an alternative approach which uses a decaying function to retain the activation from past data points [16].

2) *Habituation Coding*: Habituation comes from the natural biological process when humans and animals make decisions or respond to temporally patterned stimuli. When the animal or human receives the stimulus at the first time, the

synapses will release chemical substances to stimulate the connected neurons in the direction of signal transmission. When the stimulus is given repeatedly, the neuron would become habituated to release less chemical substances for stimulation. Primarily, habituation is a means by which biological neural systems vary their synaptic strengths in order to ignore repetitive, irrelevant stimuli. It can filter large amount of information from the surroundings, making repeated stimulus become less important. In this way, the animal will be able to focus on the more important features of the surroundings [17].

In our model, habituation acts as a model of learning which we can leverage for making rate adaptation decisions by exploiting temporal properties in the training data. The application of a sequential coding block based on habituation in the proposed framework is to encode temporal information and does not necessarily imply that our method has any psychological or biological relevance. Habituation has been used in machine learning for novelty detection, recency detection, and temporal learning [16]. Our design considers mechanisms for learning when the temporal information among contextual measurements is important. The function of temporal learning is what we are most interested in in our application. In our scheme, sequential classification is implemented by feeding the outputs of the sequential coding block into the static classifier.

There are two categories of habituation with respect to the time duration. Long-term habituation is used to describe the behavioral changes over days or weeks. Due to the properties of fluctuating channels, data points separated by long periods of time have little correlation [18]. Sequential coding based on long-term habituation makes the activation of the historical data decay slowly. Thus, the effect of long-term habituation should be avoided for our application. Short-term habituation is preferred to code the short-term correlation in time among contextual data over millisecond or second time scales.

Researchers in neurophysiology have developed a simplified mathematical expression for habituation. A discrete-time version of a short-term habituation model for varying the strength,  $w(t)$ , is summarized as a simple mathematical model [19]:

$$w_k(t+1) = w_k(t) + \tau_k(\alpha_k(1 - w_k(t)) - w_k(t)x(t)), \quad (0 < k \leq n) \quad (6)$$

where  $x(t)$  is the input,  $\tau_k$  is a constant used to vary the habituation rate, and  $\alpha_k$  is a constant used to vary the ratio between the rate of habituation and the rate of recovery from habituation. These two parameters can be used to control the trade-off between the temporal range and resolution of the correlated data points. Both  $\tau_k$  and  $\alpha_k$  are in the range of  $(0, 1)$ . Note that, (6) is a non-linear model due to the product term. The dimension for the output vector from the sequential coding block is  $n$ . When  $n > 1$  the habituation block might encode sequential information better, which we will evaluate in the following section by experimentation. In (6), we formulate the problem when the size of the input vector  $m$  is 1. When the input is multidimensional, one vector of size  $n$  will be derived for each dimension. Fig. 3 is the block diagram of non-linear coding based on habituation. For each attribute,  $x^i$ , sequential coding derives a vector  $[w_1^i, w_2^i, \dots, w_n^i]$ .

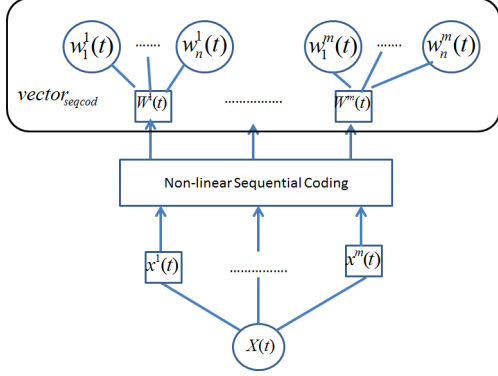


Fig. 3. Habituation-based sequential coding.

The recursion in (6) makes the procedure of sequential coding inexplicit. After the elimination of the recursion, the equivalent (7) helps us to understand the procedure as well as the selection of the key parameters in the habituation block. The term  $w_k(t)$  can be derived as follows:

$$w_k(t) = \tau_k \alpha_k + \tau_k \alpha_k \sum_{j=1}^{t-1} \prod_{h=j}^{t-1} (1 - \tau_k \alpha_k - \tau_k x(h)) + \prod_{i=0}^{t-1} (1 - \tau_k \alpha_k - \tau_k x(i)), \quad (7)$$

where  $\alpha_k$  and  $\tau_k$  control the rate of habituation and we assume  $m = 1$ . In our application, the sequential coding block is used to encode the short-term temporal information reflecting the system and channel variations. For each attribute in the input vector, such as the SNR, an  $n$ -dimensional vector is derived by the sequential coding block. Then the input to the static classifier is a vector of size  $m * n$ . The dimension of the original input vector to the sequential coding block is  $m$  and represented by:

$$f_{seq} : \{w_1^1(t), w_2^1(t), \dots, w_n^1(t), \dots, w_1^m(t), w_2^m(t), \dots, w_n^m(t)\} \rightarrow Y(t) \quad (8)$$

The input  $x^i(t)$  is normalized before processing by the sequential coding block in our application. We set  $w_k^i(0) = 1$  for all  $k$  and  $i$ . We choose  $\tau_k$  and  $\alpha_k$  from the range of (0,1) and satisfying  $\alpha_k \tau_k + \tau_k < 1$ . All these specifications guarantee that  $w_k^i(t) \in [0, 1]$  and the habituation process is stable for all values of  $k$  and  $t$ . This model is not directly derived from the biological reality, nor does it model all the expected functions of habituation. Also, the habituation model alone does not fulfill the expectation for learning behavior. Only when it is cascaded with a learner, the habituation can find its application in machine learning [16].

### III. EXPERIMENTS ON EMULATED CHANNELS

In this section, we use emulated channels for repeatability and control to directly compare and evaluate rate adaptation performance when using static- and sequential-based methods for training. The experimental results indicate that the application of non-linear sequential coding can significantly enhance the performance of linear sequential coding and static classifiers in training of rate adaptation protocols.

#### A. Experimental Set-up for Controlled, Repeatable Channels

In order to evaluate the efficacy of exploiting the temporal correlation in context information, we use an experimental setup where two wireless nodes communicate across emulated channels. The Azimuth ACE-MX is used for channel emulation, allowing controllable propagation and fading characteristics with a broad range of industry-standard models for our experiments [20]. The channel emulator can create repeatable channels for testing each transmission mode to measure the performance of a given wireless context. Each mode represents a modulation/coding scheme and packet size combination. For a given channel, we can exhaustively search for the best transmission mode in each scenario to produce a training set for the classification mechanisms. We then use randomized (but reproducible) channel settings to evaluate the rate adaptation algorithms according to the different types of training.

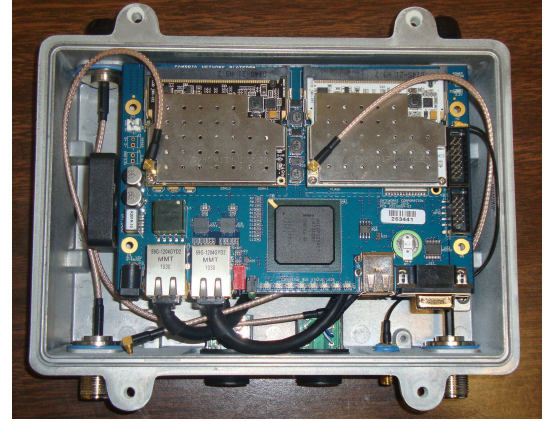


Fig. 4. Gateworks 2358 with Ubiquiti XR2 and XR5 Radios.

To ensure that our results are broadly applicable across wireless devices, we use both a FPGA-based, fully custom wireless platform as well as an off-the-shelf, 802.11-based testbed. For the custom platform, we use the Wireless Open-Access Research Platform (WARP) [21], which allows users to define the physical [22], media access [23], network layer behavior [24]. WARP also enables programmability and observability at each layer, permitting detailed per-packet channel information to be collected. For the off-the-shelf platform, we use a Linux-based Gateworks 2358 with Ubiquiti XR-2 and XR-5 radios (shown in Fig. 4). For the purposes of the emulator experiments, the Gateworks/Ubiquiti testbed allows increased number of transmission modes as compared to WARP. In the following section, we will leverage the increased transmission power and built-in GPS of the Gateworks/Ubiquiti platform.

Fig. 5 shows the experimental set-up and data flow of the experimental setup of either platform with the channel emulator. A computer captures the variation of the SNR and the throughput values from the wireless receiver and the velocity value from the channel emulator according to its current setting. With the packet error rate (PER) collected at the receiver for one minute, the throughput is calculated as

follows:

$$G_{th} = (1 - PER) * R_{th} * \frac{l_{payload}}{l_{packet}} \quad (9)$$

where  $R_{th}$  is the physical data layer rate, and  $G_{th}$  is the throughput at this rate.  $l_{payload}$  and  $l_{packet}$  represent the length of payload and the length of packet, respectively.

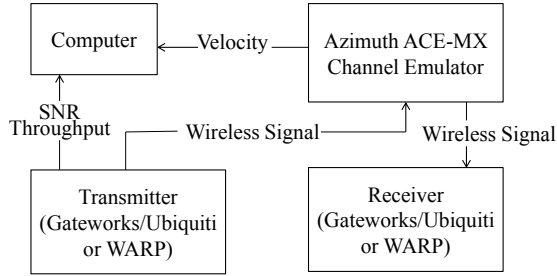


Fig. 5. Experimental Set-up for Emulated Channels.

In the training of each of the classification-based schemes, the mode with the highest throughput for context (i.e., channel type, SNR, and velocity) is the target of prediction. The learning algorithm will extract the relationship between the context information and target mode. Later, in the testing phase, the learned classifier is used to predict the target mode.

#### B. Performance of Static Classification-Based Algorithms

The performance of static classifiers depends on the scenario used, because for different kinds of data, the inherent relationship and the property are different. To verify the applicability of classifiers in our framework, we now evaluate the performance of different classifiers when training rate adaptation algorithms according to diverse contexts. As discussed previously, the available transmission modes are different based on the capabilities of the hardware platform (see Table I). For WARP, we use 6 transmission modes: 3 modulation schemes and 2 packet sizes. For Ubiquiti, we use 18 modes: 9 coding and modulation pairs and 2 packet sizes. We use both platforms for our static-classifier experiments where each test is run for a given contextual data set consisting of channel type, attenuation (SNR), and velocity for one minute per transmission mode. We average measured SNR values in every minute independently.

TABLE I  
CHANNEL SCENARIOS AND TRANSMISSION MODES

Types	Values
Channel Models (ITU)	Ch. A, Ch. B, Ch. C, Ch. D
Velocities (km/h)	0, 30, 60, 90, 120
Attenuation (dB)	0, 6, 12, 18, 24, 30, 36, 42
Modulation (WARP)	BPSK, QPSK, 16-QAM
Rates (Ubiquiti) (Mbps)	6, 9, 11, 12, 18, 24, 36, 48, 54
Packet Size (Byte)	100, 1000

For these experiments, we use 4 channel types corresponding to 4 different environments where each channel type is a pedestrian or vehicular channel model as specified by the ITU standard. For our training data set, each channel type is

emulated with each of the 40 different pairs of attenuation and velocity values specified in Table I for one minute. In total, there are 160 points in the training set. We prepare a test set of the same size as the training set with 160 unique pairs of velocity and SNR values, 40 for each of the 4 channel types. In the test set, we choose a random velocity from 0 km/h to 120 km/h for each of the fixed values of attenuation specified in Table I. Similarly, we choose a random attenuation from 0 dB to 42 dB for each of the fixed velocities found in the training set.

Table II shows the performance of rate adaptation trained using the three static classification methods of Decision Tree, SVM, and Adaboost on the Ubiquiti and WARP platforms. The metrics used are: accuracy, throughput improvement, and gap from maximum. In Table II, accuracy refers to the percentage of rate adaptation decisions that match the target rate. The throughput improvement refers to the percentage of throughput gain over an SNR-based rate adaptation scheme such as [4], [5]. Finally, the gap from the maximum refers to throughput percentage from the throughput achieved by the optimal rate. We find that when the number of available modes is small, SVM outperforms Decision Tree in prediction accuracy, and Adaboost outperforms both in accuracy and throughput improvement. When the number of available transmission modes increases (e.g., the number of classes increases) Decision Tree achieves superior performance over SVM and Adaboost. These observations are consistent with findings from the machine learning community: SVM and Adaboost were designed for binary-label classification problems and then extended to multi-label problems. Thus, it is expected that the performance will deteriorate with increasing number of modes. In summary, considering both accuracy and throughput, the performance of Decision Tree in our application is better than the other two algorithms. Particularly, Decision Tree is very stable for different situations. Thus in the following experiments, we incorporate the Decision Tree for comparison with our sequential classification-based rate adaptation scheme, ASTRA. In contrast with these schemes, ASTRA will build the relationship between the context and rate adaptation decisions using sequential coding.

In Table II, we also observe that the accuracy of SVM is higher than that of Decision Tree, but the throughput improvement and throughput gap from the maximum achievable throughput of SVM are less than those of the decision tree. One reason for this discrepancy is that the maximum throughput used for the gap is different for each data point in the testing set. SVM would give wrong predictions on points which are classified correctly by the decision tree. The throughput gap from these points are larger than the gap from the points which are given wrong predictions by the decision tree but correct predictions by SVM. Another reason for this peculiarity is that both the classifiers may fail to yield the target rate. However, the prediction from the classifier with lower accuracy provides a throughput that has a smaller gap from the maximum achievable throughput than the classifier with the higher accuracy.



TABLE II  
PERFORMANCE OF STATIC CLASSIFIERS ON WARP AND UBIQUITI MEASUREMENTS

Classifier	Platform	Number of modes	Accuracy of predicting optimal mode	Throughput improvement over SNR-based rate adaptation	Gap from maximum possible throughput
Decision Tree	<b>WARP</b>	6	72.5%	70.35%	6.28%
SVM	<b>WARP</b>	6	73.75%	68.53%	7.28%
Adaboost	<b>WARP</b>	6	78.75%	74.43%	4.04%
Decision Tree	<b>Ubiquiti</b>	18	73.1%	17.1%	16.6%
SVM	<b>Ubiquiti</b>	18	59.62%	0.53%	28.37%
Adaboost	<b>Ubiquiti</b>	18	48.27%	0.42%	33.83%

### C. Performance of Sequential Classification-based Algorithms

We now evaluate the performance of using linear and non-linear sequential coding (ASTRA-L and ASTRA-N, respectively) for training rate adaptation protocols. Since sequential training exploits the temporal properties of the channel, we must form a testing environment on the emulator with representative mobility patterns where the relative velocity between the two nodes is increasing on a particular channel type (e.g., Ch. A from Table I). We choose to use the Ubiquiti platforms for the following two reasons: (i) it has more transmission modes available, thereby limiting the effect of static classification and clearly demonstrating the impact of temporal information, and (ii) it is better suited for deployment in field trials due to the platform's with increased transmission power and built-in GPS.

We consider operation on 11 data rates with the Ubiquiti radio: 1, 2, 5.5, 11, 6, 9, 12, 18, 24, 48 and 54 Mbps. To increase the granularity of context-data, we modify the device driver of the wireless card so that it can report SNR and throughput values for each packet. At the transmitter, we count the number of transmissions (including retransmissions) per successful packet. With the ratio of this successful packet to total transmissions, we can calculate the throughput according to (9). For each rate, the total number of successful packets in each experiment duration is different, leading to a different number of throughput samples per experiment. To account for the rates with less throughput samples, we downsample the data points including SNR, velocity, and throughput before feeding them into the sequential coding block. To do so, we measure the time between two consecutive packets of the lowest transmission rate and average the throughput values and contextual data of the other transmission rates which have been collected during this time. Even with the aforementioned averaging, the fluctuation of values in the training points greatly exceed that of the training points used in Section III-B.

Next, we compare the performance of two sequential coding schemes: ASTRA-L and ASTRA-N. Similar to the static classifier case, we use 160 data points for training. Each data point includes the context information of channel type, velocity and SNR and the transmission mode that achieves the highest throughput (e.g., the target mode). We use a test set of the same size as the training set. The results of ASTRA-L are shown in Fig. 6. The static classifier is a special case of linear sequential classifier when the number of used historical data points equals 0. Thus, in Fig. 6, the 1<sup>st</sup> point where  $i = 0$ , shows the performance of static classifier on this pair of training and testing sets for comparison. As before, accuracy is the percentage of time the classifier chooses the

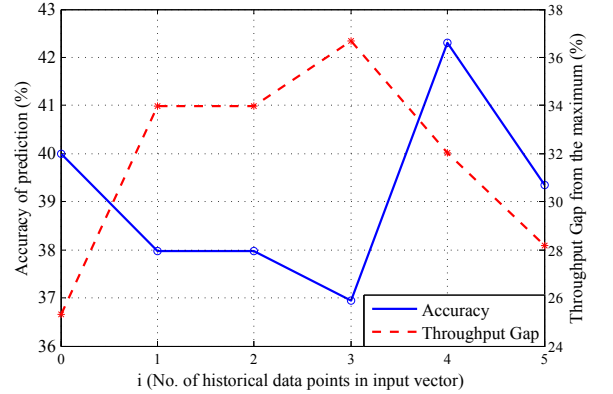


Fig. 6. Performance of ASTRA-L on data from emulated channels.

target rate and throughput gap refers to the percentage from the maximum achievable. ASTRA-L with different  $i$  values does not show desirable improvements over the static classification-based method. The degradation of performance demonstrates that the correlation between continuous measurements can not be interpreted as a linear relationship.

To test the performance of ASTRA-N and the effect of the parameter  $n$  in the habituation-based analysis block (i.e., the output vector as described in Section II), we use the same pairs of training and test sets as used to compare the static classification-based schemes and ASTRA-L. The results are shown in Fig. 7. With the increase of  $n$ , the performance shows some patterns: increasing initially, reaching its peak value, and then decreasing. These patterns can be used to determine the optimal  $n$  in practice.

For Fig. 7: nonlinear results, it can be observed that:

- ASTRA-N can significantly improve the performance of rate adaptation. When  $n = 3$ , the throughput improvements over the static classification-based method and ASTRA-L when  $i = 4$  (when ASTRA-L has the highest accuracy) are 26.47% and 21.90%, respectively. And the accuracy improvement over the static classification-based method is 10.93%. These results reveal the advantages of ASTRA-N over the static classification-based method and ASTRA-L.
- When  $n = 1$ , we do not increase the dimension of data, which means the sequential coding block does not increase any time complexity for the static classification, and the throughput is still improved. With the same time complexity of classification, ASTRA-N can still improve the performance, which demonstrates the applicability of temporal information for rate adaptation.

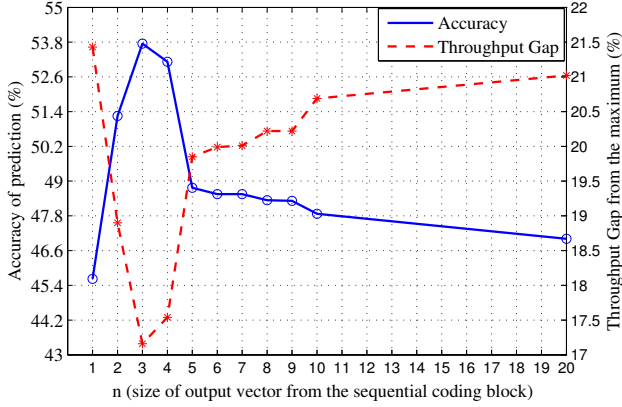


Fig. 7. Performance of ASTRA-N.

- When  $n$  increases from 1 to 3, the performance increases dramatically. When  $n = 3$ , ASTRA-N has the best performance in target mode prediction. The result is very consistent with the observations in [19]. Namely, multi-dimensional habituation coding (i.e.,  $n$  is 2 or greater) can obtain better performance over habituation coding with a single dimension.
- When  $n$  increases from 4 to 20, the performance decreases but is still better than the performance when  $n = 1$ . Multi-dimensional habituation coding has better performance in terms of capturing temporal information. We note that when  $n$  increases beyond 4, the performance changes very slowly.

The preceding analysis indicates that multi-dimensional habituation coding obtains better performance, but slightly increases the dimension of input samples, from  $m$  to  $n*m$ , and thus increases the complexity of classification. As the size of the output vector from the sequential coding block  $n$  increases, the time complexity increases: very slowly to  $n = 5$  and then dramatically beyond that point. However, this increasing time complexity does not limit the applications of ASTRA-N for two reasons: (i) ASTRA-N achieves its peak performance when  $n$  is small—usually from 2 to 4. The time complexity of ASTRA-N when  $n = 3$  is almost the same as that when  $n = 1$ . (ii) The most time-consuming phase is training the decision tree [15]. After training, the testing process has the same complexity as a static classification method (i.e., is very fast) [15]. Thus, the time complexity increased by ASTRA-N is limited to the training process. Considering the performance of classification as well as the time complexity, we set  $n = 3$  in the following in field experiments.

#### IV. IN-FIELD EXPERIMENTATION

While in the previous section, the emulator was configured to model an acceleration process with gradually changing velocity and fixed attenuation, in field trials the velocity exhibit random variations. Also, the change of one contextual attribute may interact with the change of another attribute. These variations are critical for the temporal property of in-field measurements and the corresponding evaluation of our algorithms. To test the performance of ASTRA-N on in-field



Fig. 8. Testbed on a campus bus for repeatable in-field trials.

channels, we use a bus on SMU campus with a repeatable mobility pattern for data collection and show that significant gains can be achieved for rate adaptation.

##### A. Experimental Design for In-Field Data Collection

For our experimental set-up in the field, we deploy the Gateworks/Ubiquiti platform on a campus bus as shown in Fig. 8 which makes a loop from off-campus graduate apartments to the center of campus. We installed multiple mobile-mount antennas on the roof of the bus, however, in this test we only use the 5 GHz antenna specifically in this section. Another node and antenna is located on the roof of a 3-story building near campus. The bus takes about 45 minutes to complete one loop and runs on the hour everyday from 7:00 am to 9:00 pm. Wireless traffic is sourced from the bus node to the building node (i.e., uplink). The performance data is transmitted to backbone network automatically, using a different frequency, when the bus approaches around a 3-story building. We match the throughput to location, velocity, and timestamp from the GPS built-in to the Gateworks board. SNR is collected from the Ubiquiti XR5 radio.

In Fig. 9, we place measurements for a single day on a map according to their geographical locations. Different styles of pins and circles represent different transmission modes (data rates) at which the throughput value is measured. We set the transmission power to 17 dB and use 8 data rates for our in-field data collection: 6, 9, 12, 18, 24, 36, 48 and 54 Mbps. Since the bus repeats many loops of its route every day, we have collected extensive measurements over the course of a week. We use data points in the *evaluation region* where the transmitter and the receiver can communication effectively. We divide the evaluation region into smaller regions and assume that the communication is experiencing the same type of channel condition every time the bus is located in that particular region (as signified by GPS coordinates). We evaluate ASTRA-N based on the collected contextual measurements by training and testing the sequential classifier with data from the same region.

##### B. In-Field Evaluation of Sequential Classification

For each of the 7 regions, we collect 40 data points. We choose the first 20 points for training and the remaining 20

TABLE III  
PERFORMANCE OF ASTRA-N ON DATA FROM IN-FIELD CHANNELS

Region	Accuracy of predicting optimal mode (static-based)	Accuracy of prediction optimal mode (ASTRA-N)	Gap from maximum possible throughput (static-based)	Gap from maximum possible throughput (ASTRA-N)
Region 1	100%	100%	0%	0%
Region 2	45%	55%	29.55%	19.50%
Region 3	90%	90%	0.13%	0.13%
Region 4	95%	95%	0.15%	0.15%
Region 5	35%	55%	43.33%	5.86%
Region 6	20%	55%	31.24%	29.26%
Region 7	30%	45%	14.95%	9.17%

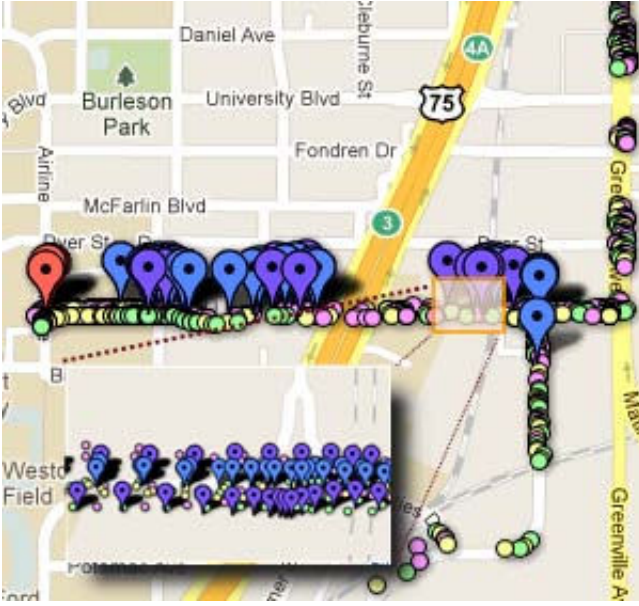


Fig. 9. Map of performance data measured from the bus.

points for testing the rate adaptation accuracy of our sequential training. We show the accuracy and gap from the maximum achievable for both the static-based training and ASTRA-N for each of the 7 regions in Table III. Results for Regions 1, 3 and 4 have a high accuracy of predicting the optimal mode no less than 90% (including 100%) when all points in the testing set and training set are assigned to the default class. The default class is the one that would be selected by majority voting in the training set. The high accuracy indicates that we can achieve minimal additional performance gains with ASTRA-N over the static classification-based algorithm for that particular region.

However, ASTRA-N achieves significant improvements in accuracy (up to 175%) and throughput (up to 66.11%) over static classification-based training in Regions 2, 5, 6 and 7. The relative improvement of ASTRA-N over the static classification-based algorithm is shown in Table IV. We note that for some regions the improvements in terms of accuracy are much higher than throughput. A key reason for this discrepancy is that the throughput gap achieved by static classification-based method and corresponding potential improvement thereof is small. For example, in Region 7, the gap for static classification-based method is 14.95%, which is much smaller than that in Region 5, i.e. 43.33%.

Thus, the advantages of ASTRA-N are more obvious when both the accuracy and throughput are low. For instance, in Region 5, ASTRA-N gains more than 50% improvement for both accuracy and throughput.

TABLE IV  
IMPROVEMENT OF ASTRA-N OVER STATIC CLASSIFICATION

Improvement in:	Region 2	Region 5	Region 6	Region 7
Accuracy	18.18%	57.14%	175%	50%
Throughput	14.27%	66.11%	2.88%	6.80%

Compared to the performance on data collected from emulated channels in last section, the performance of ASTRA-N on data from in-field channels is far more promising. To collect data from emulated channels, we configure the channel emulator to model an acceleration process with increasing relative velocity, which can result in losing temporal information. However, real measurements exhibit the natural changes of velocity values on system's performance, showing strong temporal correlation. This experiment on in-field data directly demonstrates that ASTRA-N is more suitable and very robust to the practical situations and could achieve even greater results for more dynamic channels.

## V. RELATED WORK

Machine learning borrows many concepts from the biological area to provide efficient solutions to problems in the real world. SVM and Adaboost are the most widely used classifiers in commercial applications [13], [25]. SVM represents data for classification as points in space and maps them, into a higher dimensional space to "space" them or separate them. Adaboost constructs a "strong" classifier by a combination of several "simple" and "weak" classifiers. Decision trees represent the relationship of categorical attributes and classes as a tree-like structure. Each decision node in the tree represents an attribute. Each branch will lead the deduction to another decision node or a final decision based on thresholds or conditions from training. The user can easily deduce the decision following branches with the given categorical attributes.

Machine learning has been applied to communication systems [26], [27], [28]. The classification algorithms can establish a connection between the context information and optimal rate. Cognitive Radio (CR) uses learning and adaptation of parameters according to the propagation environment. A CR model incorporating a machine learning engine into the radio architecture is described in [10]. A series of algorithms are based on this model (e.g., a CR evolved as the chromosome in genetic algorithms [29]). The KNN classifier is modified for



online classification [26]. KNN is a classification algorithm of low complexity, but the training process is implemented offline and the classification model is not adaptive online to a new data point [26]. In [27], the authors design a modulation and coding adaptation scheme with an SVM for binary classifiers.

Some works on context awareness have discussed the existence of temporal correlation among contextual measurements. The conditional probability of losing the  $k^{th}$  packet following a lost packet has been calculated in [18]. Also, the mutual information between two packets separated by a time interval is used to demonstrate the statistical correlation in system behaviors. These works indicate that the fate of a later packet depends on a previous packet to some extent, because during a relatively short transmission period, the propagation environment is relatively stable. In contrast, we utilize this observation to select the best transmission rate with the sequential classification methods.

Context information is time-varying due to the fluctuation of channel property. A central issue is how “historical” information is represented and stored. This issue can be solved by storing measurements in the recent past and presenting them for processing along with current measurements. The historical weather information is used in weather forecasting based on a temporal classification scheme in [19]. Alternatively, past information can be indirectly represented by a suitable memory device such as changes in the internal states of the processing cells [30]. Temporal classification has been widely used in the biomedical field to explore the temporal information in biomedical signals [30]. Some researchers have also studied spatio-temporal sequence recognition mechanisms in other applications, such as speech recognition [31].

## VI. CONCLUSION

In this work, we applied sequential training to rate adaptation (ASTRA) to leverage the temporal correlation of wireless channels in different environments. We did so by first testing and comparing the performance of different static classifiers to ASTRA. In our experimental analysis, we evaluated the performance of rate adaptation mechanisms on two different hardware platforms over emulated and in-field channels. Experimental results demonstrate that ASTRA-N can significantly increase the accuracy and throughput of rate adaptation over the static classification-based method by up to 175% and 66.11%, respectively. Since the size of output vector has an effect on the prediction performance and time complexity, in future work we plan to adapt its size in different situations. Finally, we also plan to consider the spatial information in contextual measurements.

## ACKNOWLEDGEMENTS

This work has been supported in part by a grant from Toyota InfoTech and by the National Science Foundation under grant CNS-0958436. We also thank buses by Bill for outdoor experiments.

## REFERENCES

- [1] J. Kim, S. Kim, S. Choi, and D. Qiao, “CARA: Collision-aware rate adaptation for IEEE 802.11 WLANs,” in *Proc. of IEEE INFOCOM*, Catalunya, Spain, Apr. 2006.
- [2] S. Wong, S. Lu, H. Yang, and V. Bharghavan, “Robust rate adaptation for 802.11 wireless networks,” in *Proc. of ACM MobiCom*, Los Angeles, CA, Sep. 2006.
- [3] J. C. Bicket, “Bit-rate selection in wireless networks,” M.S. Thesis, MIT, February 2005.
- [4] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, “Opportunistic media access for multirate ad hoc networks,” in *Proc. of ACM MobiCom*, Atlanta, GA, Sep. 2002.
- [5] G. Holland, N. Vaidya, and P. Bahl, “A rate-adaptive MAC protocol for multi-hop wireless networks,” in *Proc. of ACM MobiCom*, Rome, Italy, Jul. 2001.
- [6] J. Camp and E. Knightly, “Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation,” *IEEE/ACM Transactions on Networking*, vol. 18, no. 6, pp. 1949–1962, Dec. 2010.
- [7] J. He, H. Liu, P. Cui, J. Landon, O. Altintas, R. Vuyyuru, D. Rajan, and J. Camp, “Design and experimentation of context-aware link-level adaptation,” in *Proc. of IEEE INFOCOM*, Orlando, FL, Mar. 2012.
- [8] S. Eswaran, A. Misra, and T. La Porta, “Utility-based adaptation in mission-oriented wireless sensor networks,” in *Proc. of IEEE SECON*, San Francisco, CA, Jun. 2008.
- [9] S. Haykin, “A comprehensive foundation,” *Neural Networks*, vol. 2, 1994.
- [10] C. Clancy, J. Hecker, E. Stuntebeck, and T. O’Shea, “Applications of machine learning to cognitive radio networks,” *IEEE Wireless Communications*, vol. 14, no. 4, pp. 47–52, 2007.
- [11] J. Predd, S. Kulkarni, and H. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 23, no. 4, pp. 56–69, 2006.
- [12] “ITU-R M. 1225: Guidelines for evaluation of radio transmission technologies for imt-2000,” *ITU-R.M.1034*.
- [13] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [14] G. Rätsch, T. Onoda, and K. Müller, “Soft margins for adaboost,” *Machine learning*, vol. 42, no. 3, pp. 287–320, 2001.
- [15] J. Quinlan, *C4.5: programs for machine learning*. Morgan kaufmann, 1993.
- [16] S. Marsland, “Using habituation in machine learning,” *Neurobiology of learning and memory*, vol. 92, no. 2, pp. 260–266, 2009.
- [17] A. Fielding, *Machine learning methods for ecological applications*. Springer Us, 1999.
- [18] L. Ravindranath, C. Newport, H. Balakrishnan, and S. Madden, “Improving wireless network performance using sensor hints,” in *Proc. of USENIX NSDI*, San Jose, California, Apr. 2010.
- [19] B. Stiles and J. Ghosh, “Habituation based neural networks for spatio-temporal classification\* 1,” *Neurocomputing*, vol. 15, no. 3-4, pp. 273–307, 1997.
- [20] “Aximuth ACE - MIMO Channel Emulator,” <http://www.aximuthsystems.com>, Mar. 2011.
- [21] P. Murphy, A. Sabharwal, and B. Aazhang, “Design of WARP: Wireless open-access research platform,” in *Proc. of EUSIPCO*, Florence, Italy, Jun. 2006.
- [22] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly, “Design and experimental evaluation of multi-user beamforming in wireless LANs,” in *Proc. of ACM MobiCom*, Chicago, IL, Sep. 2010.
- [23] C. Hunter, J. Camp, P. Murphy, A. Sabharwal, E. Knightly, and C. Dick, “A flexible framework for wireless medium access protocols,” in *Proc. of Asilomar*, Monterey, CA, Nov. 2006.
- [24] S. Gupta, C. Hunter, P. Murphy, and A. Sabharwal, “WARPnet: Clean slate research on deployed wireless networks,” in *Proc. of ACM MobiHoc*, 2009.
- [25] J. Tang and H. Liu, “Feature selection with linked data in social media,” in *Proc. of SDM*, Anaheim, CA, Apr. 2012.
- [26] R. Daniels and R. Heath, “An online learning framework for link adaptation in wireless networks,” in *Proc. of IEEE ITA Workshop*, San Diego, CA, Feb. 2009.
- [27] —, “Online adaptive modulation and coding with support vector machines,” in *Proc. of IEEE EW*, Lucca, Italy, Apr. 2010.
- [28] J. Tang, H. Gao, and H. Liu, “mtrust: Discerning multi-faceted trust in a connected world,” in *Proc. of WSDM*, Seattle, WA, Feb. 2012.
- [29] T. Rondeau, B. Le, C. Rieser, and C. Bostian, “Cognitive radios with genetic algorithms: Intelligent control of software defined radios,” in *Proc. of SDR Forum Technical Conference*, Phoenix, AZ, Nov. 2004.
- [30] P. Revesz and T. Triplet, “Temporal data classification using linear classifiers,” in *Advances in Databases and Information Systems*, 2009, pp. 347–361.
- [31] R. Lippmann, “Review of neural networks for speech recognition,” *Neural computation*, vol. 1, no. 1, pp. 1–38, 1989.