Machine Learning in Wireless Relay Channels

Steven W. Peters and Kien T. Truong

I. Introduction

Our course project for CS395T has made substantial progress since the project proposal was submitted. The first phase, which consists of implementing the communication protocols and algorithms, is nearly complete, and work on the second phase, which consists of implementing and running the classifiers, is about to begin. This report details the progress we have made, the challenges we have faced, and our expectations for the remainder of the project. We begin with a brief recap of the motivation and goals of the project.

II. Motivation

Recall that any digital communication system can be modeled as a system with discrete-time input x and discete-time output y. The input x is uniformly drawn from an *alphabet* $\mathcal{X}$. Since x can take one of $|\mathcal{X}|$ values, the successful transmission of x communicates exactly $log_2|\mathcal{X}|$ bits.

By increasing $|\mathcal{X}|$, we increase the number of bits transmitted in a single transmission, and thus the "speed" of communication. However, doing so increases the number of classes, so the probability of misclassification also increases. Thus, all else equal, increasing the speed of communication will reduce the reliability of communication, and vice versa.

For our purposes, we use *packet error rate* as the metric of reliability. In most wireless sytems, data is communicated via packets; an error in any symbol or bit in the packet results in an error in the entire packet, which must somehow be retransmitted. Thus, we set a target 10% packet error rate as the maximum allowable. While arbitrary, this number is reasonable and practical.

When a channel has very low noise, one may be able to increase $|\mathcal{X}|$ without exceeding the maximum error rate. The problem of choosing $\mathcal{X}$ (and implicitly $|\mathcal{X}|$) is called *adaptive modulation*. Basic systems have analytical expressions or easily-computed lookup tables. One type of system, however, does not have tractable expressions. This system is the relay channel, where another receiving device decodes the signal and forwards it to the final destination. Our goal is to use machine learning to find the decision boundaries between alphabets in an adaptive relay system.

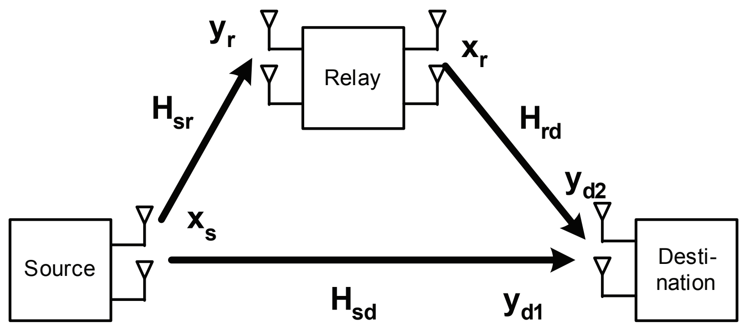

Fig. 1. Wireless relay channel.

III. Related work

Lately, machine learning techniques have been used to solve both regression and classification problems formulated to extract and utilize useful information from channel data in wireless communications. For example, support vector machines (SVM) has been used widely to improve the performance of channel estimation [1], [2] and of beamforming [3].

More notably, due to the varying nature of wireless channels, many issues in wireless systems can be modeled as classification problems. In [4], the author proposed a new pattern recognition-based handoff algorithm that results in considerably fewer handoffs in cellular networks. Handoff is defined as the change of responsibility for monitoring of a mobile station from a base station to another nearby base station. This reduction of handoff frequency decreases the amount of overhead in networks. Another application of machine learning in wireless networks is for routing optimization. In [5], [6], [7], routing protocols in wireless sensor networks are optimized based on wireless link quality classification. The basic idea of the papers is to formulate the link quality prediction as a classification problem, e.g. predicting link to be “good” or “bad”, and to apply various statistical classification problem (k-Nearest Neighborhood, Kernel methods, and SVM) to solve it.

IV. Goals and System Model

The wireless relay channel is shown in Figure 1. The helper, called a *relay*, is capable only of aiding communication between the other two devices.

In a relaying system, communication of a block of data from the source to the sink consists of two stages. First, the source transmits the data using a signal set $\mathcal{X}_S$. The sink will store the signal it receives in the first stage for later processing. In the second stage, the relay decodes its signal and re-encodes it using signal set $\mathcal{X}_R$ for transmission over the relay-sink channel. Note that, although the data is sent using two different alphabets, the exact same data is being sent over the two stages.

The sink intelligently combines the signals from both stages and is able to decode the message more reliably than if it only received directly from the source. Note that both the decoding stage at the relay and the combining stage at the sink are nonlinear processes that are not easily modeled analytically.

The input-output relationship between the source and the relay can be modeled as

$$y_r = h_{sr} x_s + v_r, \tag{1}$$

where y_r is the received signal at the relay, h_{sr} is the complex channel between the source and relay, x_s is the signal from the source and drawn from $\mathcal{X}_S$, and v_r is the additive white Gaussian noise at the relay. Similarly, for the source-sink relationship,

$$y_{d1} = h_{sd} x_s + v_{d1}, \tag{2}$$

where the subscript of "1" denotes reception in stage 1.

The relay decodes its signal from Equation 1, maps it to $\mathcal{X}_R$, and transmits this to the destination in stage 2:

$$y_{d2} = h_{rd} x_r + v_{d2}. \tag{3}$$

The destination adds the log-likelihood ratios of each bit from y_{d1} and y_{d2}, and decodes based on these.

Using this model, the goal of our classifier is to input h_{sr}, h_{sd}, h_{rd}, and output $\mathcal{X}_S$ and $\mathcal{X}_R$ such that $R = (|\mathcal{X}_S|^{-1} + |\mathcal{X}_R|^{-1})^{-1}$ is maximized within the constraint of the maximum packet error rate, where R is the overall transmission rate.

V. Progress

A. Design Decisions

We have made several decisions that will affect the implementation of the project. First, we have decided to use Rayleigh channels. That is, each h is drawn from an identical, independent Gaussian distribution. Such a model approximates a physical propagation environment with many objects scattering the electromagnetic wave. It also has the advantage of being analytically tractable and easily simulated computationally.

We have also decided to initially restrict ourselves to 3 signal-to-noise (SNR) values per link: 2 dB, 10 dB, 20 dB. These SNR values are representative in the sense that they represent all typical regimes of SNR values: low , medium, and high. Since there are 3 SNR values on each of 3 links, we actually have 27 possible effective SNRs. After testing some classifiers, depending on the available time we may investigate a higher-resolution set of SNR values. Note that, however, if the number of SNR values is p, then the number of system configurations to be investigated will be p^3.

We have chosen the packet error rate (PER) as the measure of reliability. A packet is said to be successfully transmitted from the source to the destination if the destination can decode and demodulate to extract from the received signal the same bit stream as the original bit stream at the source. If even a single bit at the destination is different from that sent at the source, a packet error has occurred. With the packet length of 512 bytes, we have set the maximum acceptable PER to 10 percent. Because of the very large simulation times that we are currently seeing, we may decrease the packet length and increase the PER accordingly.

As far as alphabets, we have chosen to use standard alphabets (more commonly known as constellations) often used in practice. For now, we have chosen to use two constellations, called Quaternary Phase Shift Keying (QPSK) and 16-level Quadrature Amplitude Modulation (16-QAM). The cardinality of these two constellations is 4 and 16, respectively. Again, if time allows, we will add to this list.

Finally, we have decided to use Matlab to generate the channel data and to use C for the other tasks. The use of C will significantly speed up the run time for the code, especially in the initial phase in which we aim to determine the optimal modulation schemes for each relay channel realizations (see discussion below). In addition, since the simulation for wireless systems requires a lot of vector and matrix manipulation work, we utilize the GNU Scientific Library (GSL), a numerical library for C programmers.

B. Implementation

We have broken up our software implementation into two phases. The first phase consists of programming the wireless system protocol and algorithms. This consists of generating random channels, data to transmit, and noise; modulating the data; decoding the data using a maximum likelihood estimator; finding log-likelihood ratios for each received bit; and finding error statistics on each received packet. Everything listed above is done except the log-likelihood estimator and the error calculator.

Importantly, in order to train any classifier, we need to have an optimal target class for each training input. Since there are no analytical or computational expressions for finding such a target for our model, we must run Monte Carlo simulations to find them. The software for these simulations is complete, and only the incomplete pieces listed above are needed to run these simulations. A timeline on when this will be completed, along with what tasks are left, is given in the next section.

C. Future Tasks & Timeline

The remaining communications algorithms are some of the most challenging parts to implement in a communication system. The log-likelihood ratio estimator requires several folding and rotation operations on the constellation. The authors have experience in buidling it, however, so we expect to complete it by April 5. At this time, the Monte Carlo simulations will be ready.

These simulations will require a significant amount of time. For each of 27 system SNRs, we have generated 10,000 channels to be partitioned into training, testing, and validation. For each of these channels, to reliably predict packet error rate will require approximately 10,000 iterations. At current speeds, this will take approximately three weeks to complete. We have several optimizations in mind, however, including compiler optimizations, running code for different SNRs on different computers, and simplifying key parts of the C code. We expect the simulations to be finished by April 12th.

While the simulation is running, we will work on implementing the classifiers (along with studying for the midterm and working on the upcoming assignment). We plan to finish the classifier implementation by April 20 so that we can have enough time, counted to the first week of May, to run enough simulations and gather enough data to make conclusions about the success of the project. Tasks need to be done after April 20 are running tests to validate the results, producing meaningful plots to compare the performance of different classifiers, writing up the final project report, and preparing for the project presentation.

References

[1] M. J. F.-G. Garcia, J. L. Rojo-Alvarez, F. Alonso-Atienza, and M. Martinez-Ramon, "Support vector machines for robust channel estimation in OFDM," *IEEE Signal Processing Letters*, vol. 13, no. 7, pp. 397–400, 2006.

[2] M. Sanchez-Fernandez, M. de Prado-Cumplido, J. Arenas-Garcia, and F. Perez-Cruz, "SVM multiregression for nonlinear channel estimation in multiple-input multiple-output systems," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2298–2307, 2004.

[3] M. M. Ramon, M. M. Ramon, N. Xu, and C. G. Christodoulou, "Beamforming using support vector machines," *IEEE Antennas and Wireless Propagation Letters*, vol. 4, pp. 439–442, 2005.

[4] K. D. Wong and D. C. Cox, "A pattern recognition system for handoff algorithms," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 7, pp. 1301–1312, 2000.

[5] Y. Ma, Y. Yu, G.-H. Lu, and Z.-L. Zhang, "Improving wireless link delivery ratio classification with packet SNR," in *2005 IEEE Int. Conf. on Electro Info. Tech.*, 2005.

[6] Y. Wang, M. Martonosi, and L.-S. Peh, "A supervised learning approach for routing optimizations in wireless sensor networks," in *Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*. Florence, Italy: ACM, 2006, pp. 79–86.

[7] ——, "Predicting link quality using supervised learning in wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 3, pp. 71–83, 2007.