Multiclass Support Vector Machines for Adaptation in MIMO-OFDM Wireless Systems

Sungho Yun
Wireless Networking & Communications Group
Electrical and Computer Engineering
University of Texas at Austin
Austin, Texas 78712
Email: shyun@mail.utexas.edu

Constantine Caramanis
Wireless Networking & Communications Group
Electrical and Computer Engineering
University of Texas at Austin
Austin, Texas 78712
Email: caramanis@mail.utexas.edu

Abstract

In MIMO-OFDM systems, by matching transmitter parameters such as modulation order and coding rate, link adaptation can increase the throughput significantly. However, creating a tractable mathematical mapping model from environmental variables to transmitter parameters that allows the latter to be optimized in any sense, presents serious challenginges due to the large number of variables involved, as well as the complexity required in any model with the ability to accurately capture and explain all factors that affect performance. Machine learning algorithms, which make no mathematical assumptions and use only past observations to model the input-output relationship, have recently been explored for adaptation in MIMO-OFDM systems. In this paper we propose a novel machine learning algorithm based on multi-class support vector machines (SVMs). Our algorithm has considerably smaller operational overhead (including storage requirements) and better performance for link adaptation. With IEEE 802.11n simulations we show that our new algorithm outperforms existing machine-learning based algorithms. Moreover, we show that our algorithm is (asymptotically) consistent, in the sense that as the number of training data used increases, our algorithm obtains the performance-optimal classifier.

I. Introduction

The limited available frequency spectrum and the demand for higher data rate, require future systems to provide significantly enhanced spectral efficiency in order to increase link throughput and network capacity. Multiple-input, multiple-output (MIMO) systems increase the throughput by simultaneously transmitting different streams of data on the different transmit antennas [1], [2]. They can be used to achieve multiple-fold improvement in the peak data rate provided that the MIMO channel is well conditioned. And indeed, this is the case in rich-scattering environments. Frequency-selective fading caused by multipath scattering can be handled by Orthogonal Frequency Division Multiplexing (OFDM). The effects of frequency-selective fading can be considered as flat over an OFDM subcarrier if the subcarrier is sufficiently narrow-banded. This makes equalization much simpler at the receiver in OFDM in comparison to conventional single-carrier modulation. As a result, the combination of MIMO with OFDM is a promising technique to enhance data traffic rate in physical layer.

However, maximizing network throughput in higher layers requires systems to meet reliability constraints to reduce overhead caused by retransmissions. Hence, both high data rate and high reliability have to be achieved simultaneously. By matching transmitting parameters such as symbol modulation order, error control coding rate, and spatial multiplexing order to time varying channel conditions, adaptive modulation and coding (AMC) can increase the transmission rate considerably while meeting the reliability constraints at the same time [3], [4]. Unfortunately, the sheer number of environmental parameters such as signal energy, noise variance, channel state information for each subcarrier, time tap, and spatial stream, make it difficult to tune the transmission parameters appropriately. Moreover, many other additional and potentially subtle factors such as quantization error, non-gaussian noise effect, and non-linearity of systems make it almost impossible to obtain a mathematical model which can be tractably optimized to find the optimal (or even near-optimal) parameters to operate the system. Hence, link adaptation to a time-varying channel and environment conditions is challenging.

Recently, there have been new flexible approaches to use machine learning algorithms for effective link adaptation [5]–[8]. The authors of [5] have proposed a non-parametric supervised learning algorithm based on k-nearest neighbor (k-NN). There, they show that a subset of ordered post-processing SNR can explain the frame error rate (FER) well, and moreover can do this with very low dimensions. Using this as a feature space, they further show that an adaptation of the k-NN algorithm provides accurate mapping from features to modulation and coding schemes (MCSs) and significantly outperforms other link adaptation algorithms in MIMO-OFDM systems.

In this paper we use the feature set extraction scheme shown in [5], namely ordered post-processing SNR, and develop a new machine learning algorithm based on multi-class support vector machines. The link-adaptation problem, unlike traditional classification problems, is in fact an optimization problem, in the sense that we seek to classify in order to optimize an objective (e.g., expected rate) as opposed to simply aiming to maximizing the probability of determining the "correct label."

Our new algorithm allows the optimization of such objectives, and our numerical results demonstrate that our algorithm does in fact outperform algorithms that focus on maximizing probability of correct classification. Indeed we show that our algorithm is statistically consistent, in the sense that as the number of training data goes increases, we asymptotically compute the performance-optimal solution. In addition to performance improvement, our algorithm presents significant advantages important in practice, including reducing time-overhead and significantly reducing memory usage and requirements.

The remainder of this paper is organized as follows. In Section II we describe the system model for the machine learning link adaptation algorithms and look into the characteristic of the system to give the motivation of our algorithm. Section III provides the framework for multi-class SVMs, after which point we give our classification algorithm. We report the results of extensive computations on IEEE 802.11n systems we conducted, in Section IV. Here we show the gains possible using algorithms that optimize performance rather than classification correctness. Finally, Section V concludes this paper.

II. MIMO-OFDM AMC Using Machine Learning

A. System Model

1) MIMO-OFDM Systems: In MIMO-OFDM systems with N_t transmit antennas and N_r receive antennas, data will be transmitted over $N_s \leq \min\{N_t, N_r\}$ spatial streams. In frequency domain, a baseband signal $\mathbf{x}[m,n]$ for the nth subcarrier of the mth OFDM symbol, will be multiplied by a pre-coding matrix, $\mathbf{F}[n]$ then transmitted over a wireless channel, $\mathbf{H}[n]$. At the receiver, we use a linear equalizer, $\mathbf{G}[n]$ to recover the transmitted signal and complex Gaussian noise, $\mathbf{v}[m,n] \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$ will be added. Then,

$$\mathbf{y}[m,n] = \sqrt{E_s}\mathbf{G}[n]\mathbf{H}[n]\mathbf{F}[n]\mathbf{x}[m,n] + \mathbf{G}[n]\mathbf{v}[m,n], \tag{1}$$

where $n \in \{0, \cdots, N-1\}$ and $m \in \{0, \cdots, N_{OFDM}-1\}$. We assume that all the modulation orders and coding rates are same for all spatial streams and the wireless channel is constant for all OFDM symbols in a single packet.

2) Learning Model: With a feature set X, a label set Y and n training samples $T = ((x_1, y_1), \cdots, (x_n, y_n)) \in (X \times Y)^n$, a machine learning algorithm creates a mapping $\mathscr{A} : X \mapsto Y$ from features to labels and predicts labels for new samples. Since the resulting mapping completely depends on training samples, it is important to extract relevant features from the data available. The number of features, however, should be limited, primarily for two reasons. First, a high dimensional feature set may allow overfitting of the data, and thus requires larger training sets in order to obtain similar predictive performance, also known as generalization performance. Second, the running time for training grows quickly as the number of features, i.e., the dimensionality, gets larger. The authors of [5] have proposed a low dimensional feature set, namely ordered post-processing SNR, that they have shown represents the frame error rate (FER) performance metric very well. In MIMO-OFDM systems (1), the post-processing SNR for spatial stream a and subcarrier n is defined as follows:

$$\gamma[a,n] = \frac{E_s|[\mathbf{G}[n]\mathbf{H}[n]\mathbf{F}[n]]_{a,a}|^2}{\sum_{a' \neq a} E_s|[\mathbf{G}[n]\mathbf{H}[n]\mathbf{F}[n]]_{a,a'}|^2 + \sigma^2 N_s|[\mathbf{G}[n]]_{a,a}|^2}. \tag{2}$$

With zero forcing (ZF) equalizer, $\mathbf{G}_{ZF}[n] = (\mathbf{F}^*[n]\mathbf{H}^*[n]\mathbf{H}[n]\mathbf{F}[n])^{-1}\mathbf{F}^*[n]\mathbf{H}^*[n]$, it reduces to

$$\gamma_{ZF}[a,n] = \frac{E_s}{\sigma^2 N_s[(\mathbf{F}^*[n]\mathbf{H}^*[n]\mathbf{H}[n]\mathbf{F}[n])^{-1}]_{a,a}}. \tag{3}$$

Following the notation of [5], we define $\gamma^{(t)} \in \{\gamma[1,0], \cdots, \gamma[N_s, N-1]\}$ as the t^{th} smallest post-processing SNR for all subcarriers and spatial streams. In IEEE 802.11n systems, there are $N = 52$ subcarriers and the selected feature set is

$$\begin{cases} x = [\gamma_{ZF}^{(5)}, \gamma_{ZF}^{(10)}, \gamma_{ZF}^{(23)}, \gamma_{ZF}^{(40)}] & \text{, if } N_s = 1 \\ x = [\gamma_{ZF}^{(6)}, \gamma_{ZF}^{(13)}, \gamma_{ZF}^{(24)}, \gamma_{ZF}^{(56)}] & \text{, if } N_s = 2 \end{cases}. \tag{4}$$

Hence we have a 4-dimensional feature set. The goal of a learning algorithm is to map any point in the feature space to the "best" MCS. Hence, the set of modulation and coding schemes (MCS) of IEEE 802.11n form our set of labels. MCS ranges from MCS_0 to MCS_7 for 1 spatial stream, and MCS_8 to MCS_{15} for 2 spatial streams. Given $\mathbf{H}, E_s, \sigma^2$ and target FER $\mathscr{T}$, the "best" MCS is the MCS_i with highest rate $\mathscr{R}_i$ such that the target FER constraint is met, i.e.,

$$i = \arg\max_j \{\mathscr{R}_j : \text{FER}(\mathbf{H}, E_s, \sigma^2, \text{MCS}_j) \leq \mathscr{T}\}. \tag{5}$$

So far, we have defined a feature set and a label set for IEEE 802.11n MIMO-OFDM systems. Given training samples, machine learning algorithms produce mappings that will predict the best MCSs for future channel realizations.

B. Optimization as Classification: An Example

In the standard classification setup, machine learning algorithms such as k nearest neighbor and support vector machines try to maximize the classification accuracy, i.e., they seek to maximize the probability that the next point generated is classified correctly. If we think of this as a reward function, it means we obtain the same reward for correctly classifying a sample, regardless of the label, and also that we gain nothing by wrongly classifying samples. Yet in our link-adaptation setting, this assumption is not consistent with actual system-performance, since the penalty for misclassification may not be symmetric. We illustrate this point in the following simple example.

Example 1: A sample x has a 30% probability to be labeled A and a 70% probability to be labeled B. When the correct label is A, we get reward 100 by correctly classifying the sample, while if we misclassify it as B we still get reward 50. On the other hand, correctly classifying a sample with label B gives reward 80 and misclassifying it as A gives only reward 10. In a classification problem, we classify x as B to minimize the classification error. However in order to maximize the expected reward, we choose label A since by choosing A, we expect $0.3\times 100+0.7\times 50=65$ which is higher than what we expect by choosing B: $0.3\times 10+0.7\times 80=59$.

Because some of the MCSs are comparable, in the sense of maximum rate and also FER (namely, for the same channel conditions, we may have that the FER of one MCS is always at least as great as another's FER) the above example is precisely illustrative of the scenario in the MIMO-OFDM link-adaptation problem. First, each correctly classified MCS results in different performance, its own rate. Second even when we misclassify it, we may get some degraded performance if we classify it as a lower rate MCS. (By choosing higher rate MCS we will violate the target FER and observe zero performance due to the high layer overhead.)

In Example 1, there is ambiguity about which label the sample x will be labeled as. If there is no ambiguity, classification algorithms will do just fine. Hence the situations mentioned above will occur mostly on the mutual boundaries of different label's areas. Therefore we need a machine learning algorithm that compares every pair of labels instead of comparing them altogether because we want to impose asymmetric weights on each pair of labels.

III. Multi-class SVM with Asymmetric Weights

In this section, we present a multi-class SVM-based algorithm. The key idea is to introduce asymmetric penalties in order to "favor" making certain classification mistakes over others. We show how the weights for these penalties should be chosen, based on the objective we would like to maximize. Note that if we are interested in simply maximizing the probability of correct classification, the weights should all be chosen equal to one another. Then we show that if the weights are properly chosen, our algorithm is statistically consistent, that is, as the number of training data grows, our classifier is asymptotically optimal in the sense of the objective to be optimized.

A. Asymmetric Weights for Binary Classification Algorithms

In order to motivate the method of putting asymmetric weights in our algorithm, let us begin with a simpler binary classification case. Given a feature set X and a label set Y, we assume a probability distribution on $(X\times Y)$, i.e., $(x,y)\sim(X\times Y)$. The $|Y|=2, Y=\{+1,-1\}$ case is referred to as binary classification. With a sample set, $T=((x_1,y_1),\cdots,(x_n,y_n))\in(X\times Y)^n$, we would like to compute a real-valued function f to minimize

$$\mathbb{E}[\Psi_y(f(x))],$$

where the expectation is taken over the unknown distribution that generates the sample points x and their labels y. Keeping to the standard PAC-learning setup, we assume we know nothing about the underlying generative distribution, but we do assume that the training data we see are all generated *iid* according to the true distribution. In particular, we assume that the testing data are also generated according to the same distribution that generates the training data. One possibility is to attempt to find a function f that minimizes the *empirical loss* rather than the true expected loss:

$$\frac{1}{n}\sum_i \Psi_{y_i}(f(x_i)) \tag{6}$$

For the standard classification problem, our objective is to minimize the probability of misclassification. Thus, the loss function of interest is $\Psi_y(f(x))=I(y\neq \mathrm{sign}(f(x)))$. This function is non-convex, however, and as a consequence, one can show that the problem (6) becomes a combinatorial optimization problem, and in particular is NP-hard. As a result, alternate tractable (and in particular, convex) loss functions such as $\Psi_y(f(x))=\phi(yf(x))$, where $\phi(t)=(1-t)_+$ are typically used.

Thus there are two difficulties that must be overcome. First, we do not have access to the underlying distribution against which performance is judged, and thus must rely on the empirical distribution. Second, if the true objective is non-convex and intractable, we must resort to training our algorithm using a tractable loss function. Statistical consistency, which we revisit in the next section below, requires showing that as the number of samples goes to infinity, both of these problems are overcome. In the context of binary classification, these problems are well-studied.

Under suitable conditions, minimizing a regularized version of (6) over a sequence of function classes, minimizes the Ψ-risk, $R_\Psi(f) = \mathbb{E}_{XY}[\Psi_y(f(x))]$. If our objective is classification correctness, we also want the probability of misclassification $R(f)$ of that minimizer to approach the optimal risk (*Bayes risk*). It is shown in [9] and graphically explained in [10] that if we have a convex loss function $\phi : \mathbb{R} \mapsto [0,\infty)$ which is differentiable at 0 and $\phi'(0) < 0$, any minimizer f^* of Ψ-risk yields a Bayes consistent classifier, i.e.,

$$\begin{cases} f^*(x) > 0, & \text{if } \mathbb{P}(Y=+1|X=x) > 1/2 \\ f^*(x) < 0, & \text{if } \mathbb{P}(Y=-1|X=x) > 1/2. \end{cases} \tag{7}$$

Now, moving away from the classification correctness problem, we consider the expected reward maximization problem. Let us fix an x and the two conditional probabilities $\mathbb{P}(Y=+1|X=x)$ and $\mathbb{P}(Y=-1|X=x)$ by p_+ and p_-, respectively. Let r_{ij} be the reward obtained when we classify an i-labeled sample as j. For example, we gain r_{+-} when we misclassify a $+1$-labeled sample as -1. Then the expected reward by classifying x as $+1$ is $p_+ r_{++} + p_- r_{-+}$ and the expected reward by classifying it as -1 is $p_+ r_{+-} + p_- r_{--}$. Therefore, the reward-optimizing classifier f^* is as follows:

$$\begin{cases} f^*(x) > 0 & \text{,if } p_+ > \frac{r_{--}-r_{-+}}{(r_{++}+r_{--})-(r_{+-}+r_{-+})} \\ f^*(x) < 0 & \text{,if } p_- > \frac{r_{++}-r_{+-}}{(r_{++}+r_{--})-(r_{+-}+r_{-+})} \end{cases}. \tag{8}$$

Note that in the standard classification problem, $r_{++} = r_{--} = 1$ and $r_{+-} = r_{-+} = 0$, hence (8) reduces to (7). Now given a sample point x, we define asymmetric loss functions, $\phi_y(yf) = \alpha_y \phi(yf) = \alpha_y(1-yf)_+$ omitting the argument in $f(x)$. Then Minimizing Ψ-risk is equivalent to finding f that minimizes the following form.

$$p_+ \phi_+(f) + p_- \phi_-(-f) \tag{9}$$

These loss functions penalize f differently according to asymmetric weights α_+ and α_-. If we define the set $\mathscr{R} \in \mathbb{R}^2$ as

$$\mathscr{R} = \{(\phi_+(f), \phi_-(-f)) : f \in \mathbb{R}\}, \tag{10}$$

then the above minimization can be written as

$$\min_{\mathbf{z} \in \mathscr{R}} \langle \mathbf{p}, \mathbf{z} \rangle \tag{11}$$

where $\mathbf{p} = (p_+, p_-)$.

Lemma 1: If asymmetric weights, α_+ and α_-, are as follows,

$$\frac{\alpha_+}{\alpha_-} = \frac{r_{++} - r_{+-}}{r_{--} - r_{-+}}, \tag{12}$$

then the classifier f^* that minimizes (9) maximizes the expected reward, i.e., f^* follows (8).

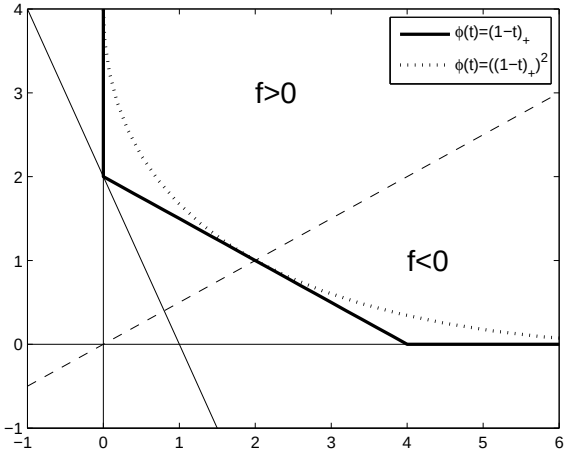


Fig. 1. set $\mathscr{R}$ with hinge loss and squared hinge loss

Proof: The set $\mathscr{R}$ is shown in Fig. 1 for the hinge loss function $\phi(t) = (1-t)_+$ and the squared hinge loss function $\phi(t) = ((1-t)_+)^2$ where $\alpha_+ = 2$ and $\alpha_- = 1$. Note that the slope of the line going through the transition area from $f > 0$ to $f < 0$ is $-\frac{\alpha_-}{\alpha_+}$. By taking a line $\langle \mathbf{p}, \mathbf{z} \rangle = c$ and sliding it until it touches $\mathscr{R}$, we obtain the solution to (11).

Suppose $p_+ > \frac{r_{--}-r_{-+}}{(r_{++}+r_{--})-(r_{+-}+r_{-+})}$. Then we have $\frac{p_+}{p_-} > \frac{r_{--}-r_{-+}}{r_{++}-r_{+-}} = \frac{\alpha_-}{\alpha_+}$, which means the line is inclined more towards the vertical axis and the point of contact is in the area of $f > 0$, hence the first condition of (8) holds. Similarly, it can be shown that the second condition holds, too. ■

B. Multi-class Support Vector Machines

For multi-class classification problems, many generalizations of binary SVM have been proposed ([11]–[16]). Among those, the one-against-one and the Weston/Watkins methods compare every pair of labels unlike others that compare one label against the others altogether. As discussed in the previous section, in order to maximize the expected reward, we require asymmetric weights on the comparison of each pair of labels. Therefore, for our purposes, these form an appropriate basis for our classification algorithms. In this section we develop the framework for the algorithms with asymmetric weights introduced in Section III-A. We show that these algorithms actually maximize the expected reward. That is, they are consistent. In order to do this, we show that the convexified objective asymptotically minimizes the actual objective (which we see is also non-convex), and moreover minimizes this with respect to the unknown actual underlying distribution.

1) Multi-class SVM framework: The One-against-one method constructs $K(K-1)/2$ binary SVMs where K is the number of classes. Given a sample set T of size n, in order to compare label i and j we construct a smaller sample set $T^{ij} = \{(x_t, y_t) \in T : y_t = i \vee y_t = j\}$ and let $n^{ij} = |T^{ij}|$. Then the corresponding binary SVM problem is

$$\begin{array}{lll} \min_{f^{ij}\in\mathscr{H}, b^{ij}\in\mathbb{R}} & \frac{\lambda_n}{2}\|f^{ij}\|^2_{\mathscr{H}} + \frac{1}{n^{ij}}\sum_{t:(x_t,y_t)\in T^{ij}} \alpha^{ij}_{y_t}\xi^{ij}_t & \\ \text{s.t.} & f^{ij}(x_t) + b^{ij} \geq 1 - \xi^{ij}_t & y_t = i \\ & f^{ij}(x_t) + b^{ij} \leq -1 + \xi^{ij}_t & y_t = j \\ & \xi^{ij}_t \geq 0 & \end{array} \tag{13}$$

where $\mathscr{H}$ is a reproducing kernel Hilbert space (RKHS). Following the observation of Section III-A, asymmetric weights α^{ij}_i and α^{ij}_j are defined as follows:

$$\alpha^{ij}_i = \sqrt{\frac{r_{ii}-r_{ij}}{r_{jj}-r_{ji}}}, \qquad \alpha^{ij}_j = \sqrt{\frac{r_{jj}-r_{ji}}{r_{ii}-r_{ij}}}. \tag{14}$$

In the testing phase, for a sample x, we vote for either label i or j according to the classifier f^{ij}. Then after $K(K-1)/2$ votes, we predict that x is labeled as the one with the largest vote sum.

The Weston/Watkins method constructs a multi-dimensional classifier $f : X \mapsto \mathbb{R}^K$ by solving one big optimization problem. The idea is that we make the ith element of f, f_i, to separate samples of label i from the others by maximizing the sum of gaps between the samples of label i and the samples of the other labels. The formulation without offsets is as follows.

$$\begin{array}{ll} \min_{f\in\mathscr{H}^K} & \frac{\lambda_n}{2}\sum_{m=1}^K \|f_m\|^2_{\mathscr{H}} + \frac{1}{n}\sum_{i=1}^n \Psi_{y_i}(f(x_i)) \\ \text{s.t.} & \Psi_y(f(x)) = \sum_{y'=1}^K r_{yy'}\Phi_{y'}(f(x)) \\ & \Phi_y(f(x)) = \sum_{y'\neq y}\phi(f_y(x) - f_{y'}(x)) \\ & \phi(t) = ((1-t)_+)^2 \end{array} \tag{15}$$

and the decision rule is

$$y^* = \arg\max_{y=1,\cdots,K} f_y(x). \tag{16}$$

2) Consistent Classifiers: In Section III-A, we have shown that minimizing Ψ-risk leads to the expected reward maximization in binary classification problems. It is by now well known that minimizing the empirical risk asymptotically minimizes the true expected risk for the case where the loss function is the indicator of misclassification. Putting these two together, we have the consistency result for the case of binary classification for reward maximization. In this section, we show that the consistency result of binary classification problems can be generalized for multi-class classification problems.

Before we go further, let us define some notation. The empirical Ψ-risk is defined as $\hat{R}_\Psi(f) = \frac{1}{n}\sum_i \Psi_{y_i}(f(x_i))$ and the Ψ-risk is $R_\Psi(f) = \mathbb{E}_{XY}[\Psi_y(f(x))]$. To avoid overfitting we use a squared norm regularizer and the classifier $\hat{f}^*_\lambda$ minimizes the regularized empirical Ψ-risk, $\hat{R}^{reg}_{\Psi,\lambda}(f) = \frac{\lambda}{2}\sum_{m=1}^K \|f_m\|^2_{\mathscr{H}} + \hat{R}_\Psi(f)$ [17], [18]. Similarly, the classifier f^*_λ minimizes the regularized Ψ-risk, $R^{reg}_{\Psi,\lambda}(f) = \frac{\lambda}{2}\sum_{m=1}^K \|f_m\|^2_{\mathscr{H}} + R_\Psi(f)$. Let $R(f)$ be the expected reward of a function f over the underlying distribution of $(X \times Y)$ and the largest achievable expected reward is defined as $R^* = \sup\{R(f)|f : X \mapsto \mathbb{R}^K \text{measurable}\}$.

Theorem 2 shows that the classifier we compute by minimizing regularized empirical Ψ-risk also minimizes Ψ-risk, and Theorem 4 generalizes the consistency result of Section III-A to multi-class classification problems.

Theorem 2: Given a RKHS $\mathscr{H}$, let $K : X \times X \mapsto \mathbb{R}$ be a corresponding kernel. If K is universal and $\lambda_n \to 0$ slowly enough as $n \to \infty$, then the classifier $\hat{f}^*_{\lambda_n}$ from (15) holds the following condition in probability for all distributions on $(X \times Y)$.

$$\lim_{n\to\infty} R_\Psi(\hat{f}^*_{\lambda_n}) = \inf\{R_\Psi(f)|f : X \mapsto \mathbb{R}^K \text{measurable}\} \triangleq R^*_\Psi \tag{17}$$

We adapt the consistency proof for binary classification in [19]. The proof consists of the following five steps.

$$R_\Psi(\hat{f}^*_{\lambda_n}) \quad \leq \quad R^{reg}_\Psi(\hat{f}^*_{\lambda_n}) \quad \leq \quad \hat{R}^{reg}_\Psi(\hat{f}^*_{\lambda_n})+\varepsilon \quad \leq \quad \hat{R}^{reg}_\Psi(f^*_{\lambda_n})+\varepsilon \quad \leq \quad R^{reg}_\Psi(f^*_{\lambda_n})+2\varepsilon \quad \leq \quad R^*_\Psi+3\varepsilon \tag{18}$$

The first inequality is obvious since the regularizer is positive and the third one is due to the definition of $\hat{f}^*_{\lambda_n}$. The second and fourth inequalities hold by the so-called "concentration" theorem we will prove soon. Finally, the last step holds as λ_n goes to 0.

Proof: Before we prove the theorem, we define the following notations and upper bound the norm of the solutions to (15) and the norm of Ψ-risk functions by Lemma 3.

$$\begin{aligned}
&\bar{r} = \max_{i,j} r_{i,j} \quad ,i,j \in \{1,\cdots,K\}\\
&M = \sup_x \sqrt{k(x,x)} \quad ,x \in X\\
&\delta_\lambda = \sup\{\|\mathbf{t}\|_2 | \mathbf{t} \in \mathbb{R}^K, \tfrac{\lambda\|\mathbf{t}\|_2^2}{2} \leq \sup_y \Psi_y(0)\}\\
&\|\Psi_\lambda\|_\infty = \sup\{|\Psi_y(\mathbf{t})| | y \in \{1,\cdots,K\}, \|\mathbf{t}\|_2 \in [-\delta_\lambda M, \delta_\lambda M]\}
\end{aligned}$$

Lemma 3:

$$\textstyle\sum_{m=1}^K \|\hat{f}^*_{\lambda,m}\|^2_{\mathscr{H}} \leq \delta_\lambda^2, \quad \sum_{m=1}^K \|f^*_{\lambda,m}\|^2_{\mathscr{H}} \leq \delta_\lambda^2 \tag{19}$$

Proof: Due to the definition of $\hat{f}^*_\lambda$, we have $\frac{\lambda}{2}\sum_{m=1}^K \|\hat{f}^*_{\lambda,m}\|^2_{\mathscr{H}} + \frac{1}{n}\sum_{i=1}^n \Psi_{y_i}(\hat{f}^*_\lambda(x_i)) \leq \frac{1}{n}\sum_{i=1}^n \Psi_{y_i}(0) \leq \sup_y \Psi_y(0)$. Since the loss function is positive, $\frac{\lambda}{2}\sum_{m=1}^K \|\hat{f}^*_{\lambda,m}\|^2_{\mathscr{H}} \leq \sup_y \Psi_y(0)$, and by the definition of δ_λ, the first inequality of the lemma holds. A similar argument can be formulated for the second one. ■

If we define a space $\mathscr{H}^K \triangleq ((\mathscr{H}, \|\cdot\|_{\mathscr{H}})^K, \|\cdot\|_2)$, then the lemma implies that $\hat{f}^*_\lambda, f^*_\lambda \in \delta_\lambda B_{\mathscr{H}^K}$, where $B_{\mathscr{H}}$ is the unit ball in $\mathscr{H}^K$ centered at the origin. Also by Hoeffding's inequality, $\forall f \in \delta_\lambda B_{\mathscr{H}^K}$,

$$Pr(|\hat{R}_\Psi(f) - R_\Psi(f)| \geq \varepsilon) \leq 2e^{\frac{2\varepsilon^2 n}{\|\Psi_\lambda\|^2_\infty}}. \tag{20}$$

Next, we define the covering number of a RKHS $\mathscr{H}$, $\ell = \mathscr{N}((\mathscr{H}, \|\cdot\|_{\mathscr{H}}), \varepsilon)$, i.e., there exist $f_1,\cdots,f_\ell$ such that the disks D_i centered at f_i with radius ε cover $\mathscr{H}$. Then we claim that the covering number of $\mathscr{H}^K$ is at most ℓ^K with radius $\varepsilon\sqrt{K}$. The proof is as follows.

$$\begin{aligned}
& \exists f_1,\cdots,f_\ell \text{ s.t. the disks } D_i(f_i,\varepsilon) \text{ cover } \mathscr{H}\\
\Rightarrow\ & \forall f \in \mathscr{H}, \exists i \in \{1,\cdots,\ell\} \text{ s.t. } \|f - f_i\|_{\mathscr{H}} \leq \varepsilon\\
\Rightarrow\ & \forall \mathbf{f} \in \mathscr{H}^K, \forall j \in \{1,\cdots,K\}, \exists i \in \{1,\cdots,\ell\} \text{ s.t. } \|\mathbf{f}_j - f_i\|_{\mathscr{H}} \leq \varepsilon\\
\Rightarrow\ & \forall \mathbf{f} \in \mathscr{H}^K, \exists \mathbf{f}' \in \{f_1,\cdots,f_\ell\}^K \text{ .s.t. } \|\mathbf{f} - \mathbf{f}'\|_2 \leq \varepsilon\sqrt{K}\\
\Rightarrow\ & \mathscr{N}((\mathscr{H}^K, \|\cdot\|_2), \varepsilon\sqrt{K}) \leq \ell^K
\end{aligned} \tag{21}$$

Now, to verify the second and the fourth inequalities of (18) and prove the theorem, we claim that

$$Pr\left\{\sup_{f \in \delta_\lambda B_{\mathscr{H}^K}} |\hat{R}_\Psi(f) - R_\Psi(f)| \geq \varepsilon\right\} \leq 2\left(\mathscr{N}\left(\delta_\lambda B_{\mathscr{H}}, \frac{\varepsilon}{4\Delta_\Psi M\sqrt{K}}\right)\right)^K e^{\frac{\varepsilon^2 n}{2\|\Psi_\lambda\|^2_\infty}}, \tag{22}$$

where $\Delta_\Psi = \sup_{f \in \delta_\lambda B_{\mathscr{H}^K}} \|\bigtriangledown_f \Psi_y(f)\|_2$.
First, we upper bound the difference between risk functions of two classifiers. $\forall f_1, f_2 \in \delta_\lambda B_{\mathscr{H}^K}$, we have

$$|\Psi_y(f_2(x)) - \Psi_y(f_1(x))| \leq \sup_{f \in \delta_\lambda B_{\mathscr{H}^K}} |\bigtriangledown_f \Psi_y(f)^\top (f_2(x) - f_1(x))| \leq \Delta_\Psi \|f_2 - f_1\|_2 M. \tag{23}$$

By simply integrating it over the underlying distribution and the sample distribution the following two inequalities hold.

$$\begin{aligned}
|R_\Psi(f_2) - R_\Psi(f_1)| &\leq \textstyle\int_{X\times Y} |\Psi_y(f_2(x)) - \Psi_y(f_1(x))| \leq \Delta_\Psi \|f_2 - f_1\|_2 M\\
|\hat{R}_\Psi(f_2) - \hat{R}_\Psi(f_1)| &\leq \textstyle\frac{1}{n}\sum_{i=1}^n |\Psi_{y_i}(f_2(x_i)) - \Psi_y(f_1(x_i))| \leq \Delta_\Psi \|f_2 - f_1\|_2 M
\end{aligned} \tag{24}$$

Using triangular inequality, we have,

$$|(\hat{R}_\Psi(f_2) - R_\Psi(f_2)) - (\hat{R}_\Psi(f_1) - R_\Psi(f_1))| \leq 2\Delta_\Psi \|f_2 - f_1\|_2 M. \tag{25}$$

Let $\ell = \mathscr{N}(\delta_\lambda B_{\mathscr{H}}, \frac{\varepsilon}{4\Delta_\Psi M\sqrt{K}})$. Then by (21), $\mathscr{N}(\delta_\lambda B_{\mathscr{H}^K}, \frac{\varepsilon}{4\Delta_\Psi M}) \leq \mathscr{N}((\delta_\lambda B_{\mathscr{H}})^K, \frac{\varepsilon}{4\Delta_\Psi M}) \leq \ell^K$. Defining $f_i, i \in \{1,\cdots,\ell^K\}$ to be the centers of the disks that cover the space $\delta_\lambda B_{\mathscr{H}^K}$, $\forall f \in \delta_\lambda B_{\mathscr{H}^K}$ we have

$$|(\hat{R}_\Psi(f) - R_\Psi(f)) - (\hat{R}_\Psi(f_i) - R_\Psi(f_i))| \leq 2\Delta_\Psi \|f - f_i\|_2 M \leq \frac{\varepsilon}{2}. \tag{26}$$

Therefore by (20)

$$Pr\left\{\sup_{f \in D_i} |\hat{R}_\Psi(f) - R_\Psi(f)| \geq \varepsilon\right\} \leq Pr\left\{\|\hat{R}_\Psi(f_i) - R_\Psi(f_i)| \geq \frac{\varepsilon}{2}\right\} \leq 2e^{\frac{\varepsilon^2 n}{2\|\Psi_\lambda\|^2_\infty}}. \tag{27}$$

Finally, plugging the covering number we developed earlier into (27), we conclude the proof.

$$Pr\left\{\sup_{f\in\delta_\lambda B_{\mathscr{H}^K}}|\hat{R}_\Psi(f)-R_\Psi(f)|\geq\varepsilon\right\}\leq\sum_{i=1}^{\ell^K}Pr\left\{\sup_{f\in D_i}|\hat{R}_\Psi(f)-R_\Psi(f)|\geq\varepsilon\right\}\leq 2\left(\mathscr{N}\left(\delta_\lambda B_{\mathscr{H}},\frac{\varepsilon}{4\Delta_\Psi M\sqrt{K}}\right)\right)^K e^{\frac{\varepsilon^2 n}{2\|\Psi_\lambda\|_\infty^2}}, \quad (28)$$

■

In case of (15) equipped with the Gaussian RBF kernel, we have $\delta_\lambda\leq K\sqrt{\frac{2}{\lambda}}$, $\Delta_\Psi\sim\frac{1}{\sqrt{\lambda}}$, and $\|\Psi_\lambda\|_\infty\sim\frac{1}{\lambda}$. Using the upper bound for the covering number shown in [20], the term "slowly enough" in Theorem 2 can be clarified as $\lambda_n\to 0$ and $n\lambda_n^2|\log\lambda_n|^{-d-1}\to\infty$ where d is the dimension of the feature space, X.

Theorem 4: Let $\Psi(\cdot)$ be a loss function of the Weston/Watkins method equipped with the squared hinge loss as in (15). Then,

$$R_\Psi(\hat{f}^*_{\lambda_n})\to R^*_\Psi \qquad \text{in probability}$$

implies

$$R(\hat{f}^*_{\lambda_n})\to R^* \qquad \text{in probability}$$

Proof: This proof follows closely the proof of classification case in [10].

Let us rewrite the asymmetric Ψ-risk as

$$R_\Psi(f)=\mathbb{E}_X[\mathbb{E}_{Y|x}[\Psi_y(f(x))]] \quad (29)$$

Now let us fix an arbitrary $x\in X$. We write $\mathbf{f}$ instead of $f(x)$ and let p_y be the conditional probability of label y given a sample point x. Then the inner expectation of (29) is $\sum_y p_y\Psi_y(\mathbf{f})$. Since both p_y and $\Psi_y(\mathbf{f})$ are nonnegative in (15), we're guaranteed to have the infimum of the inner expectation. If we define the subsets $\mathscr{R}$ and $\mathscr{S}$ of $\mathbb{R}^K_+$ as

$$\begin{aligned}\mathscr{R}&=\{(\Phi_1(\mathbf{f}),\cdots,\Phi_K(\mathbf{f}))^\top:\mathbf{f}\in\mathbb{R}^K\}\\ \mathscr{S}&=\text{conv}(\mathscr{R})=\text{conv}(\{(\Phi_1(\mathbf{f}),\cdots,\Phi_K(\mathbf{f}))^\top:\mathbf{f}\in\mathbb{R}^K\})\end{aligned} \quad (30)$$

and a matrix $\mathbf{R}\in\mathbb{R}^{K\times K}$ as

$$\mathbf{R}=\begin{pmatrix}r_{11}&r_{21}&\cdots&r_{K1}\\ \vdots&\vdots&\ddots&\vdots\\ r_{1K}&r_{2K}&\cdots&r_{KK}\end{pmatrix} \quad (31)$$

then we have

$$\begin{aligned}\inf_{\mathbf{f}\in\mathbb{R}^K}\sum_y p_y\Psi_y(\mathbf{f})&=\inf_{\mathbf{f}\in\mathbb{R}^K}\sum_y p_y(\textstyle\sum_{y'=1}^K r_{yy'}\Phi_{y'}(\mathbf{f}))\\ &=\inf_{\mathbf{f}\in\mathbb{R}^K}\textstyle\sum_y p_y(\mathbf{R}^\top\vec{\Phi}(\mathbf{f}))_y\\ &=\inf_{\mathbf{z}\in\mathscr{R}}\langle\mathbf{Rp},\mathbf{z}\rangle=\inf_{\mathbf{z}\in\mathscr{S}}\langle\mathbf{Rp},\mathbf{z}\rangle\end{aligned}, \quad (32)$$

where $\mathbf{p}=(p_1,\cdots,p_K)^\top$. The last equation holds because the inner product is a linear function. It is shown that the Weston/Watkins method with a squared hinge loss function is universally consistent [10], i.e., $\forall\mathbf{p}\in\Delta_K$ all sequences $\{\mathbf{z}^{(n)}\}\in\mathscr{S}$ such that $\langle\mathbf{p},\mathbf{z}^{(n)}\rangle\to\inf_{\mathbf{z}\in\mathscr{S}}\langle\mathbf{p},\mathbf{z}\rangle$, we have $\arg\max_y p_y$ as a predicted label when we use a decision rule that chooses a label $y^*=\arg\min_y z_y$. Notice that this decision rule is equivalent to ours (16) since the loss function $\Phi(\cdot)$ is non-increasing. Moreover, one can easily see that $\mathbf{p}$ being in $\mathbb{R}^K_+$ and bounded is enough for their proof instead of $\mathbf{p}\in\Delta_K$. Therefore, since $\mathbf{Rp}\in\mathbb{R}^K_+$ and bounded, by solving (32) we get a predicted label y that achieves

$$\max_y(\mathbf{Rp})_y=\max_y\sum_{y'}p_{y'}r_{y'y} \quad (33)$$

, which means the expected reward is maximized by choosing the label y. Hence our algorithm is consistent in the sense of maximizing the reward. ■

These two theorems assure that with a sufficiently large number of samples, the solution to (15) maximizes the expected reward.

IV. Application to MIMO-OFDM Systems

As we have discussed earlier, Modulation and Coding Schemes for MIMO-OFDM systems have asymmetric rate performance. and MCSs for one spatial stream case are listed in TABLE I. Aggressive MCS selection can achieve high spectral efficiency but may have overall worse performance because of unacceptably high FER. On the other hand conservative MCS selection can guarantee at least a fraction of the performance of the ideal selection, but nevertheless performance is sacrificed. Our algorithm balances between those two schemes and maximizes the expected throughput.

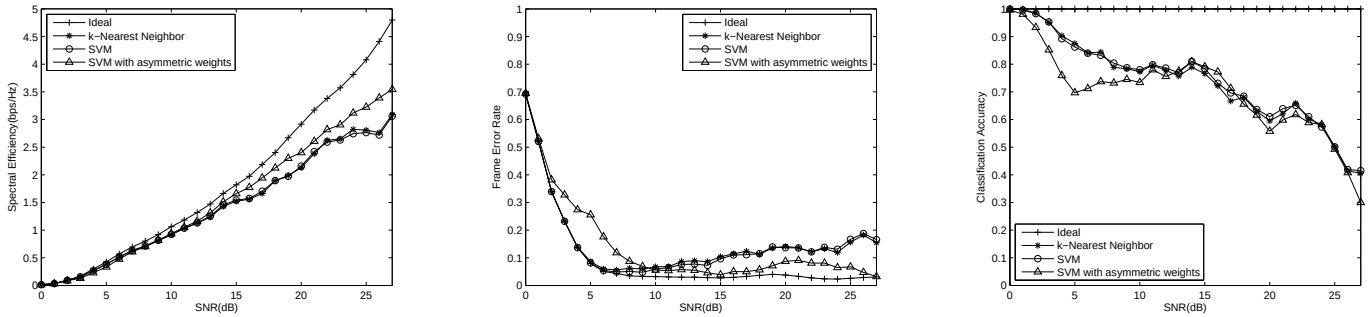
In this section, we evaluate the performance of our algorithm using IEEE 802.11n based simulation study. We use the packet error rate simulation data of [21]. Under 2×2 MIMO-OFDM and 4 taps frequency selective fading, 2 sets of 28,000 channels are generated according to the zero-mean complex-Gaussian distribution with SNR varying from 0 to 27. Then, packet error rate

is simulated for every pair of channel realization and MCS. We extract features from channel state information and associate them to their ideal MCSs with a target FER 0.1 as we discussed in Section II-A2. Also we use LIBSVM [22], a library for support vector machines, to construct SVM classifiers. We evaluate the rate performance of our algorithm as well as other practical advantages such as reducing time overhead and memory usage.

TABLE I
RATE PERFORMANCE FOR ONE SPATIAL STREAM CASE

	Ideal MCS							
	0	1	2	3	4	5	6	7
0	6.5	6.5	6.5	6.5	6.5	6.5	6.5	6.5
1	0	13.0	13.0	13.0	13.0	13.0	13.0	13.0
2	0	0	19.5	19.5	19.5	19.5	19.5	19.5
3	0	0	0	26.0	26.0	26.0	26.0	26.0
4	0	0	0	0	39.0	39.0	39.0	39.0
5	0	0	0	0	0	52.0	52.0	52.0
6	0	0	0	0	0	0	58.5	58.5
7	0	0	0	0	0	0	0	65.0

A. Spectral Efficiency



(a) (b) (c)

Fig. 2. SNR vs. (a) Spectral Efficiency, (b) Frame Error Rate, and (c) Classification Accuracy for different link adaptation algorithms

Spectral efficiency, frame error rate and classification accuracy for different link adaptation algorithms over varying SNR are shown in Fig. 2 (a), (b) and (c) respectively. In this simulation, Gaussian RBF kernel, $K(\mathbf{x_1},\mathbf{x_2}) = e^{-\|\mathbf{x_1}-\mathbf{x_2}\|^2/2\sigma^2}$ is used and parameters for SVM and k-NN algorithms such as the regularization coefficient, the kernel coefficient and the number of neighbors are chosen by cross validation methods. Also, a comparison of multi-class support vector machines of [23] shows that the One-against-one method requires much shorter testing time than the Weston/Watkins method while their performances are just comparable. Thus, we use the One-against-one method for all the simulations in this paper. As one can see, even though classification accuracy of our algorithm is worse than the others, it outperforms them in terms of spectral efficiency performance. (We have gained about 0.5 bps/Hz at higher SNR.) This verifies that maximizing the expected performance is a different problem from minimizing the classification error rate in machine learning schemes. As shown in Fig. 2 (b), frame error rate of our algorithm is high at lower SNR and low at higher SNR, which means that our algorithm is aggressive at lower SNR and conservative at higher SNR.

B. Memory Usage

Since the memory size for mobile devices is limited, small memory usage is desirable. The k-NN algorithm, however, needs all the pre-observed data to be stored. In our IEEE 802.11n AMC framework, we need to store 28,000 samples. Using a SVM algorithm with a kernel function, the resulting classifier is expressed as follows.

$$f(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_i \Phi(\mathbf{s_i}) \cdot \Phi(\mathbf{x}) + b = \sum_{i=1}^{N_s} \alpha_i K(\mathbf{s_i},\mathbf{x}) + b \tag{34}$$

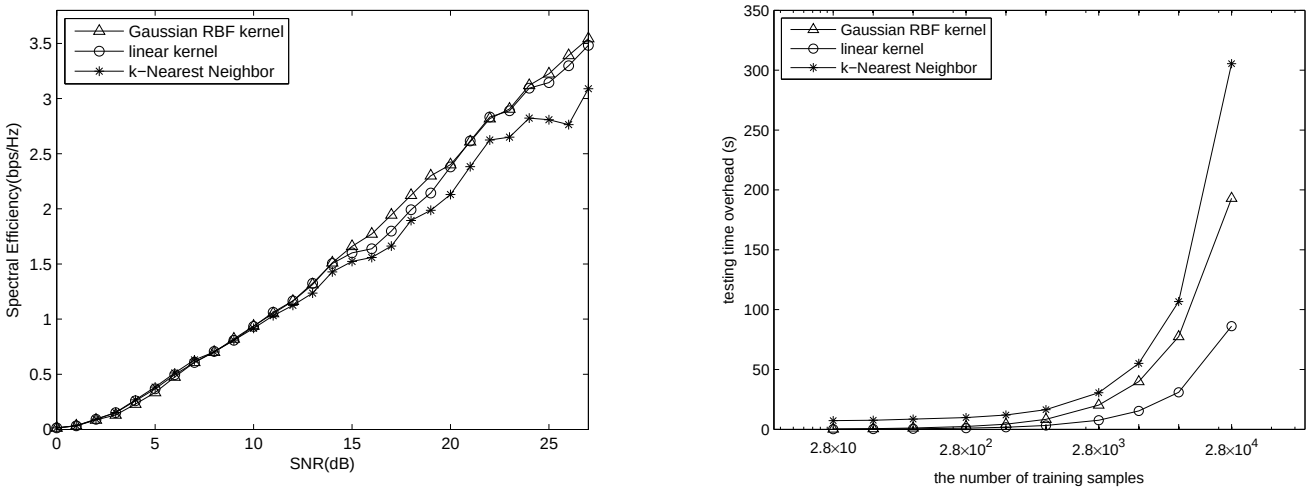
where N_s is the number of support vectors. Hence the memory size needed is proportional to the number of support vectors. However it is not predictable and can be as large as the number of samples, which means we may not have a considerable memory reduction by using SVM over k-NN. Hence many approximations to reduce the number of support vectors have

been proposed, and the authors in [24] have provided an exact simplification of support vector solutions, in which the linearly dependent support vectors in kernel feature space are merged into one support vector, and only effective support vectors remain without any approximation. Using this simplification method, we can upper bound the number of effective support vectors.

Lemma 5: The number of effective support vectors is at most the dimension of the kernel feature space.

Proof: Finding effective support vectors, $\mathbf{s_1}, \cdots \mathbf{s_{N_s}}$, such that $K(\mathbf{s_1}, \cdot), \cdots, K(\mathbf{s_{N_s}}, \cdot)$ determine a unique classifier, is equivalent to finding $\Phi(\mathbf{s_1}), \cdots, \Phi(\mathbf{s_{N_s}})$ that determine a unique subspace in the kernel feature space. Since linearly independent d vectors can determine a unique subspace in d dimensional space, the minimal number of effective support vectors is at most the dimension of the kernel feature space. ■

The dimensions of kernel feature space, i.e., the upper bound of the number of effective support vectors, for well-known kernel functions such as homogeneous polynomial, inhomogeneous polynomial and Gaussian RBF functions are listed in TABLE II. As one can see, even high degree polynomial kernel functions show much less memory usage than k-NN's 28,000. Although it is a common thought that the classifiers lie in higher dimensional feature space perform better, depending on problems simple kernels can be comparably effective, too. To compare the two extremes, Fig. 3(a) shows the spectral efficiency performance of our algorithms with a Gaussian RBF kernel function and a linear kernel function. As shown in the figure, two algorithms have almost the same performance outperforming k-NN algorithm. Therefore, we can reduce the memory usage significantly by using a linear kernel function.



(a) (b)

Fig. 3. (a) SNR vs. Spectral Efficiency for algorithms with different kernel functions, (b) The number of training samples vs. testing time for different algorithms

TABLE II
THE UPPER BOUND OF THE NUMBER OF EFFECTIVE SVS

p	1	2	3	4	5	6	7
$(\mathbf{x_1} \cdot \mathbf{x_2})^p$	4	10	20	35	56	84	120
$(\mathbf{x_1} \cdot \mathbf{x_2} + 1)^p$	5	15	35	70	126	210	330
$e^{-\|\mathbf{x_1}-\mathbf{x_2}\|^2/2\sigma^2}$	∞						

C. Testing Time Overhead

MCS selection occurs in real time, hence the testing phase overhead for learning algorithms should be minimized. Testing phase of k-NN algorithm consists of computing distances between a new sample and the training samples and sorting the distances to find k nearest neighbors. Therefore it requires $\mathcal{O}(n \log n)$ time complexity where n is the number of training samples. On the other hand, in the testing phase of SVM algorithm, we compute N_s kernel functions, thus only $\mathcal{O}(N_s)$ time complexity is required where N_s is the number of support vectors. As we have seen in Section IV-B, the effective number of support vectors can be reduced significantly depending on the kernel functions. Fig. 3(b) shows the actual testing time to choose MCSs for 28,000 new channel realizations with different machine learning algorithms. Almost a 70% reduction in testing overhead is achieved by using our algorithm with a linear kernel function over the k-NN algorithm.

V. Conclusions

In this paper we have shown that machine learning algorithms with appropriately weighted labels are suitable for AMC in MIMO-OFDM. We've developed a consistent learning algorithm that does not minimizes the classification error but maximizes the expected reward. Asymmetric weights play a key role to achieve high performance by balancing between aggressive and conservative label selection. In addition to the performance improvement, our algorithm has practical advantages for implementation over other machine learning algorithms.

One issue we did not mention is the channel estimation error at the receiver. Perfect channel state knowledge is assumed in this paper which may not be the case in real systems. Although how badly the estimation error impacts on the performance is not known, applying robust optimization techniques can be a potential extension of our work to handle this problem. Another extension is to make our algorithm to adapt to changing target reliability. In reality, different types of data and changing demand for data rate keep the target reliability changing over time. Since our algorithm cannot tune itself according to different target error rates, we have to consider more flexible and adaptable algorithms. Reinforcement learning algorithms may work in this case.

Acknowledgments

We would like to thank Bob Daniels and Robert Heath Jr. for valuable conversations, suggestions, and also for sharing their considerable expertise on MIMO wireless systems. We would also like to thank them for making available the data set, without which exploring new algorithms would be considerably more difficult.

References

[1] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1, no. 2, pp. 41–59, 1996.

[2] G. G. Raleigh and J. M. Cioffi, "Spatio-temporal coding for wireless communication," *IEEE Trans. Commun*, vol. 46, pp. 357–366, 1998.

[3] M. S. Alouini and A. Goldsmith, "Adaptive modulation over nakagami fading channels," 1998.

[4] S. T. Chung and A. Goldsmith, "Degrees of freedom in adaptive modulation: A unified view," *IEEE Trans. Commun*, vol. 49, pp. 1561–1571, 2001.

[5] R. C. Daniels, C. M. Caramanis, and R. W. Heath, Jr, "Adaptation in convolutionally-coded MIMO-OFDM wireless systems through supervised learning and SNR ordering," *to appera in Vehicular Technology, IEEE Transactions on*, December.

[6] Y. Ma, "Improving wireless link delivery ratio classification with packet snr," in *Electro Information Technology, 2005 IEEE International Conference on*, May 2005, pp. 6 pp.–6.

[7] M. Haleem and R. Chandramouli, "Adaptive stochastic iterative rate selection for wireless channels," *Communications Letters, IEEE*, vol. 8, no. 5, pp. 292–294, May 2004.

[8] A. Misra, V. Krishnamurthy, and S. Schober, "Stochastic learning algorithms for adaptive modulation," in *Signal Processing Advances in Wireless Communications, 2005 IEEE 6th Workshop on*, June 2005, pp. 756–760.

[9] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Large margin classifiers: Convex loss, low noise, and convergence rates," in *Advances in Neural Information Processing Systems 16*. Cambridge, MA: MIT Press, 2004.

[10] A. Tewari and P. L. Bartlett, "On the consistency of multiclass classification methods," *J. Mach. Learn. Res.*, vol. 8, pp. 1007–1025, 2007.

[11] J. Weston and C. Watkins, "Multi-class support vector machines," 1998.

[12] E. J. Bredensteiner and K. P. Bennett, "Multicategory classification by support vector machines," *Computational Optimizations and Applications*, vol. 12, pp. 53–79, 1999.

[13] K. Crammer, Y. Singer, N. Cristianini, J. Shawe-taylor, and B. Williamson, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, p. 2001, 2001.

[14] T. Zhang, "An infinity-sample theory for multi-category large margin classification," in *Advances in Neural Information Processing*. MIT Press, 2004, p. 16.

[15] T. Zhang and B. Scholkopf, "Statistical analysis of some multi-category large margin classification methods," *Journal of Machine Learning Research*, vol. 5, pp. 1225–1251, 2004.

[16] Y. Lee, Y. Lin, and G. Wahba, "Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data," *Journal of the American Statistical Association*, vol. 99, pp. 67–81, 2004.

[17] V. Vapnik and A. Chervonenkis, "The necessary and sufficient conditions for consistency in the empirical risk minimization method," *Pattern Recognition and Image Analysis*, vol. 1, no. 3, pp. 260–284, 1991.

[18] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 171–203.

[19] I. Steinwart, "Consistency of support vector machines and other regularized kernel classifiers," *Information Theory, IEEE Transactions on*, vol. 51, no. 1, pp. 128–142, Jan. 2005.

[20] D.-X. Zhou, "The covering number in learning theory," *Journal of Complexity*, vol. 18, no. 3, pp. 739 – 767, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/B6WHX-46P42XB-7/2/9a3dcbba67446da51cc045b47af0a46c

[21] R. C. Daniels, C. M. Caramanis, and R. W. Heath, Jr. (2008) PER simulations for 2x2 IEEE 802.11n system. [Online]. Available: http://128.83.198.111/mlearn/4_tap_files/index.htm

[22] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[23] C. W. Hsu and C. J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, Mar 2002.

[24] T. Downs, K. E. Gates, and A. Masters, "Exact simplification of support vector solutions," *J. Mach. Learn. Res.*, vol. 2, pp. 293–297, 2002.