Supervised Learning for Link Adaptation in Multi-antenna Wireless Links

Anonymous Author(s)

Abstract

In this paper, we apply supervised learning algorithms for link adaptation in wireless communications where nodes possess multiple antennas. Link adaptation maximizes throughput while maintaining target reliability by adaptively selecting the modulation order and coding rate. Due to physical channel impairments, hardware non-linearities, and non-Gaussian noise effects, it is often difficult to analytically model the link quality metrics such as error rate. In this work, we consider machine learning framework that exploits past observations of error rate and associated channel–state information to predict the best modulation order and coding rate for new channel realizations. In particular, we study three classifiers: nearest neighbor algorithm, support vector machines, and random forest (RF). We show classification accuracy and performance of all the schemes. The accuracy performance of all the classifiers are consistent. In particular, we develop reduced feature dimension in RF classifier based on out–of–bag importance which simplifies the training procedure a lot.

1 Introduction

Maximizing the performance of multiantenna wireless links requires aggressive yet reliable selection of modulation and coding parameters based on the channel state information (CSI). Selecting proper modulation and coding schemes (MCS) based on the observed CSI is referred to as adaptive modulation and coding (AMC) [user other]. The conventional approach to fast AMC is to maximize a rate–proportional objective function (e.g., a closed-form representation of link throughput) subject to functional constraints that express link reliability (e.g., upper bounds on error rate) [1]. The objective function and functional constraints are appropriately defined in terms of observable domains, often called link quality metrics. In practice, AMC finds the optimal communication parameters through look-up-tables filled with optimal parameter mappings for link quality metric realizations. However, the lookup table approach may be inadequate because the error rate of practical links is intrinsically linked to an ever-growing set of potential system configurations. For example, wide-band wireless links with multiple-input multiple-output orthogonal frequency division multiplexing (MIMO–OFDM) create frequency and spatial selective link quality metrics. Analysis has not been able to provide a universal, low-dimensional link quality metric in these systems due to link performance sensitivity in terms of wireless communication parameters. Moreover, for wireless systems with radio frequency (RF) and analog circuits that interface to the wireless medium, mitigating circumstances such as amplifier nonlinearity, transmission frequency instability, and non-Gaussian additive noise limit tractable mathematical modeling of the complete wireless system. Hence, adaptation solely based on the analysis of mathematical models struggles to accurately account for all the factors that determine performance. Furthermore, machine learning algorithms are also flexible enough to observe nonlinearities in the system model.

Motivated by this fact, in this paper, we consider link adaptation in the form of adaptive modulation and coding (AMC), where the symbol modulation order, error-control coding rate, and spatial multiplexing order are assigned to maximize the data rate under frame-error-rate (FER) reliability constraints imposed by higher layers. In particular, we consider IEEE 802.11n (WiFi) channels but the techniques can be easily extended to IEEE 802.16e (WiMax), and the Third-Generation Partnership Project Long Term Evolution (3GPP–LTE) platform as well. In particular, we adopt a
supervised learning framework for the MIMO-OFDM AMC problem. Every wireless channel realization (for each packet) is classified or matched to a distinct MCS through the use of a feature set extracted from the CSI. We will treat the system as a black box and use observations (features) associated with past system performance (i.e., the training set) to predict the performance output of the black box given a new set of feature values. Few contributions have been reported in the literature on machine learning algorithms for link adaptation [2], [3], [4]. The authors of [2] 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 (which are based on average throughput) in MIMO–OFDM systems. A multi–class support vector machine (SVM) method for link adaptation is proposed in [3] however the objective was to maximize throughput rather than maximizing the probability of correct classification and it was shown that this approach achieve lower accuracy compared to k–NN algorithm. Also note that SVM techniques were originally defined for binary classification problems, but recently researchers proposed few techniques (eg., one vs. one, one vs. all) to adapt to multiple classes. The methods reported in [2] and [3] operate on an off–line basis, i.e., learning framework is developed offline and it is then applied to unseen test data. However, online realizations are also possible as reported in [3]. In this work, we focus on offline supervised learning framework and study different popular classification techniques and assess their performance. We obtained simulated data sets from Robert Daniels of UT Austin which includes simulated 32,000 realization of the channels for 2x2 MIMO–OFDM system and corresponding frame error rate data for different chosen MCS labels. We study k–NN algorithm, SVM, and classification forests [5] for the considered problem. As classification forests are inherently suitable for multi–class problems, we propose to apply it for link adaptation.

2 System Model

In this section, we present the MIMO–OFDM operation, derive the post–processing SNR, and provide an expression for an upper bound on the Frame Error Rate (FER). We adopt regular statistical and vector notations in the paper.

2.1 MIMO–OFDM Operation

MIMO–OFDM is a key technology for next-generation cellular communications (3GPP–LTE, Mobile WiMAX) as well as wireless LAN (IEEE 802.11n), wireless PAN, and broadcasting (DAB, DVB, DMB). We use the same notations as adopted in [2]. This section defines the MIMO–OFDM system model under consideration, , which represents the communication procedure of wireless signals in a practical MIMO–OFDM wireless link such as IEEE 802.11n. Each physical layer frame of the MIMO–OFDM transceiver\(^1\) is processed as follows:1 The \(N_f\) source bits for a single frame \(b[p_n]_{n=0}^{N_f-1}\) are delivered to the convolutional encoder. The convolutionally encoded bits \(c[n_c]_{n_c=0}^{C-1}\) with coding rate \(C\) are sent to the spatial bit parser. Each spatial branch \(a \in \{1, 2, \ldots, N_s\}\) of the transmitter receives an equivalent number of bits (we assume that each spatial branch has the same modulation order and coding rate). The spatial parsing operation not only assigns bits from codewords to multiple spatial streams but also interleaves bits over the spatial streams and sub–carriers. We assume the spatial parser consecutively process \(L\) bits.

After the spatial parsing operation, the bits in each spatial branch \(a \in \{1, 2, \ldots, N_s\}\) enter the \(M\)-ary quadrature amplitude modulator (M–QAM). Therefore, \(K = \log_2(M)\) bits are modulated at a time in each spatial branch. Using matrix notation, we represent the set of \(M–QAM\) samples across all spatial branches as \(X[m, n]_{m=0}^{N_O-1} N_{O-1}^{N_{0}}\) where \(X[m, n] \in \mathbb{C}^{N_s}\). The spatial mapping block transforms each \(N_s\)-dimensional complex vector into \(N_t\) complex dimensions that distinctly map to each transmit RF chain \((N_{t} \leq \min(N_s, N_f))\) through a linear precoding matrix \(F[n] \in \mathbb{C}^{N_t \times N_s}\) for each subcarrier \(n\). The choice of \(F[n]\) will be discussed later. The last digital processing at the transmitter is the inverse discrete Fourier transform (IDFT) and cyclic prefix (CP) addition. In each of the \(N_t\) transmit branches, the IDFT algorithm processes \(N\)-length blocks.

At the receiver, the equalized samples are equivalently mapped into the discrete Fourier domain, and the frequency-domain representation of OFDM symbol \(m\) and subcarrier \(n\) is given by \(Y[m, n]\).

---

\(^1\)Due to space constraints, we do not provide a detail theory on MIMO–OFDM. Interested readers may consult [6] for background on MIMO–OFDM.
After mapping the QAM symbols to soft bits, the receiver calculates the sequence of log-likelihood ratios which are then utilized by the Viterbi decoder to generate the estimated frame \( \hat{b} \in \{0, 1, \ldots, N_f - 1\} \).

Given MIMO–OFDM symbol \( m \in \{0, 1, \ldots, N_f - 1\} \), sub–carrier \( n \in \{0, 1, \ldots, N - 1\} \), \( \mathbf{H}[n] \in \mathbb{C}^{N_s \times N_r} \) as the wireless channel, \( \mathbf{G}[a] \in \mathbb{C}^{N_s \times N_r} \) as the linear equalization matrix at the receiver, and \( \mathbf{V}[m, n] \in \mathbb{C}^{N_r} \) as the thermal noise, then

\[
\mathbf{Y}[m, n] = \sqrt{E_s} \mathbf{G}[a] \mathbf{H}[n] \mathbf{F}[n] \mathbf{X}[m, n] + \mathbf{G}[a] \mathbf{V}[m, n].
\]

Let us define \( \tilde{\mathbf{H}}[n] = \mathbf{G}[n] \mathbf{H}[n] \mathbf{F}[n] \) which denotes the combined effect of spatial mapping, channel, and equalization. All modulated signals have unit average power, i.e., \( \mathbb{E}[\mathbf{X}^\ast[m, n] \mathbf{X}[m, n]] = 1 \). Here, \( E_s \) denotes the expected total transmit power by ensuring \( ||\mathbf{F}[n]||^2_F = 1 \) holds. The noise term is modeled as \( \mathbf{V}[m, n] \sim \mathcal{CN}(0, \sigma^2 I) \).

### 2.2 Post–Processing SNR

For multiple spatial stream transmission, the SNR is defined per subcarrier and per spatial stream. For analysis, linear receivers and linear precoders are convenient since the performance can be described with the postprocessing SNR per spatial stream \( a \in \{1, 2, \ldots, N_s\} \) as [7]

\[
\gamma[a, n] = \frac{\left|\mathbb{H}[n]\right|_{a,a}^2}{\sum_{a' \neq a} \left|\mathbb{H}[n]\right|_{a,a'}^2 + \frac{\sigma^2}{E_s} \sum_{a'=1}^{N_s} \left|\mathbb{G}[n]\right|_{a,a'}^2}
\]

in sub–carrier \( n \). One of the most commonly adopted choices for \( \mathbf{G}[n] \) is the zero–forcing (ZF) spatial equalization, where \( \mathbf{G}_{ZF}[n] = (\mathbf{F}^\ast[n] \mathbf{H}^\ast[n] \mathbf{H}[n] \mathbf{F}[n])^{-1} \mathbf{F}^\ast[n] \mathbf{H}^\ast[n] \). For systems that use Eigen Value Decomposition (EVD) precoding at the transmitter, linear receivers achieve ML performance [34]. According to this criterion, \( \mathbf{F}[n] \) collects the \( N_s \) largest Eigen vectors of \( \mathbf{H}^\ast[n] \mathbf{H}[n] \) which ensures that \( \mathbf{F}^\ast[n] \mathbf{H}^\ast[n] \mathbf{H}[n] \mathbf{F}[n] \) becomes a diagonal matrix and multiple streams can be easily decoupled at the receiver. For ZF receivers, the post–processing SNR simplifies to

\[
\gamma_{ZF}[a, n] = \frac{E_s}{\sigma^2 \sum_{a'=1}^{N_s} \left|\mathbb{G}[n]\right|_{a,a'}^2},
\]

which will be exploited later in the FER formulation and learning algorithms.

### 2.3 Frame Error Rate

Let us define \( \gamma^{(t)} \in \{\gamma[1, 0], \gamma[1, 1], \ldots, \gamma[N_s, N - 1]\} \) as the \( t \)th smallest postprocessing SNR from (3) for all sub–carriers and spatial streams such that

\[
\gamma^{(1)} \leq \gamma^{(2)} \leq \ldots \leq \gamma^{(N_s, N)}
\]

where \( \gamma^{(1)} = \min_{a, n} \gamma[a, n] \) and \( \gamma^{(N_s, N)} = \max_{a, n} \gamma[a, n], \forall a, n \). In [2], the authors derived a FER upper bound for the considered MIMO–OFDM system as

\[
\text{FER} \leq 1 - \left( 1 - \frac{1}{q} \sum_{d=d_L} d_{x} C_{d_L, d} \prod_{l=1}^{d} Z(\gamma^{(l)}) \right)^{N_f},
\]

where \( C_{d_L, d} \) is the number of source bit errors due to \( d \)–distance error events of length at most \( L \), \( d_L \) as the maximum distance error event with length \( \leq L \), and \( Z(\gamma) \triangleq \exp(-\gamma) \). The FER approximation enables performance characterization and is useful for reliability constraint evaluation. Note that \( C_{d_L, d} \) depends on \( \{\mathbf{H}[n]\}_{n=0}^{N_f-1}, E_s, \) and \( \sigma^2 \). For (5) to hold, we assume that spatial parsing block length \( L < N_s N \) according to IEEE 802.11n standards.

Remark 1: FER bound in (5) does not depend on exact sub–carrier location, rather it is a function of the ordered postprocessing sub–carrier SNR. This is an important observation for feature set selection which will be explained in the next section.

### 3 AMC Class Labeling and Feature Set Selection

In this section, we formulate AMC as a classification problem and construct a feature space based on the FER analysis presented in Section 2.
3.1 AMC Class Labeling

For the purpose of AMC, we have chosen a supervised learning framework (i.e., where a training set containing features or attributes and correct class labels is available). Therefore, in the context of AMC, a training set that consists of channel realizations associated with the ideal coding rate, modulation order, and number of spatial streams must be available. To create the training set, the FER is simulated/measured from experiments for each potential choice of coding rate, modulation order, and number of spatial streams by repeated transmission of packets. In particular, AMC in our MIMO–OFDM system model is the process of selecting the QAM modulation order \( M \), convolutional coding rate \( C \), and number of spatial streams \( N_s \) for a given realization of the channel state information. We denote the class label index as \( i \) which distinctly maps to a realization of \( M, C, \) and \( N_s \) to define \( MCS_i \), where \( i \in \mathcal{I} \) and cardinality of \( \mathcal{I} \) denotes the number of available \( M, C, \) and \( N_s \) triplets. Therefore, in terms of classification, AMC is the process of selecting a class \( i \), corresponding to \( MCS_i \), to maximize \( R_i \) (the data rate provided by the values of \( M, C \), and \( N_s \) in \( MCS_i \)) under a reliability constraint for different realizations of the channel state. As FER is widely chosen as a performance metric in past AMC work, we can formulate the class labeling procedure as

- class \( i \) is only selected if the corresponding FER of \( MCS_i \) is less than or equal to \( \mathcal{P} \) (i.e., \( FER_i \leq \mathcal{P} \)). Therefore, classification with this performance metric uses class

\[
\arg \max_i \{ R_i : FER_i \leq \mathcal{P} \} \tag{6}
\]

for a given channel realization.

If no MCS satisfies (6), the most reliable MCS (i.e., with lowest FER) will be selected.

3.2 Feature Set Selection

The choice of appropriate feature set selection is critical. In IEEE 802.11n systems with two transmit and two receive antennas, using the full channel in the feature space would require 416 dimensions (52 data sub–carriers two transmit radio–frequency (RF) chains two receive RF chains two dimensions of complex channel), resulting in complete failure of classification with training sets of reasonable size. It is necessary to find a low dimensional feature set that captures the essence of (6) in these 416 dimensions. Given 5, we know that the set of ordered SNR \( \{ \gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(N,N)} \} \) uniquely parameterizes the approximate FER in MIMO–OFDM systems. Furthermore, there is a chance of significant correlation among the ordered SNR due to sorting process and finite depth interleaving [1]. Hence, it may be possible that a low dimensional feature set can accurately approximate the whole domain. We perform a mapping to obtain a \( p \) dimensional feature set out of \( N_s N \) dimensional ordered SNRs, i.e.,

\[
\mathcal{F}(\{ \gamma^{(1)}, \gamma^{(2)}, \ldots, \gamma^{(N,N)} \}) \triangleq \{ \gamma^{(n_1)}, \gamma^{(n_2)}, \ldots, \gamma^{(n_p)} \} \tag{7}
\]

where \( n_k \in \{ 1, 2, \ldots, N_s N \}, k \in \{ 1, \ldots, p \} \). The procedure for discovering the best indices \( \{ n_k \} \) will be discussed in the next section.

4 Supervised Learning Algorithms

The functional mapping between the link quality metrics and the feature set drawn from the ordered postprocessing SNR is not obvious. Hence, we adopt supervised learning algorithms to provide accurate class estimates without knowledge of the functional mapping between the feature sets and the class. In particular, we consider three widely used classifiers: a) k nearest neighbor (k–NN) algorithm, b) support vector machines (SVM), and c) random forests.

4.1 k–NN AMC

The k–NN is among the simplest of all machine learning algorithms where an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its \( k \) nearest neighbors (\( k \) is a positive integer). The k–NN system is trained with \( W \) distinct realizations of the feature set and its associated class (i.e., the training set). We define a realization index \( w \in \{ 0, 1, 2, \ldots, W \} \) for each feature set in the training set. The feature set corresponding to index \( w \) in the training set, i.e., \( z_w \in \mathbb{R}^p \) is assigned to a class \( i \in \mathcal{I} \) according to (6) using the realization of the channel state from which \( z_w \) was evaluated. The corresponding MCS for each element of
the training set follows the mapping \( \{ z_w \}_{w=1}^{W-1} \rightarrow \{ i(w) \}_{w=0}^{W-1} \), where \( i(w) \in I \) is the real AMC classification according to (6).

The \( k \)-NN AMC predicts the class for a new realization of the CSI using a query \( q \in \mathbb{R}^p \) which represents the feature set of test channel. For a given depth of search \( k \) and distance metric \( d(\cdot, \cdot) \), the \( k \)-NN algorithm is given by A critical feature of the algorithm is the rule for breaking ties. If two distinct classes \( i' \neq i'' \) return the same number of neighbors to the query, the ties are broken such that \( i' \) is selected if \( R_{i'} < R_{i''} \). Hence, the proposed algorithm is biased towards lower rate selection, which empirically often implies higher reliability [1].

### 4.2 SVM AMC

The SVMs are one of the most popular and widely used binary classifiers as they guaranty maximum–margin separation, and often this property yields good generalization with relatively little training data [9]. However, despite the success of SVMs, they do not extend naturally to multiple class problems, such as the considered link adaptation problem. To this end, researchers have proposed multi–class SVMs by combining several binary classifiers, such as one vs. all, one vs. one etc. In this work, we consider one vs. one multiclass SVM where \( I(I-1)/2 \) SVM models are constructed where each one is trained on data from two classes. We have \( W \) training data \( \{ z_w, i(w) \} \), \( w \in \{1, \ldots, W\} \), where \( z_w \in \mathbb{R}^p \) and \( i(w) \) are the corresponding class labels. For training data from the \( mn \)-th and the \( m'n \)-th classes, we solve the following binary classification problem:

\[
\min_{\nu^{mn}, b^{mn}, \xi^{mn}} \frac{1}{2} (\nu^{mn})^T \nu^{mn} + C \sum_w \xi^{mn}_w (\nu^{mn})^T \phi(z_w) + b^{mn} \geq 1 - \xi^{mn}_w, \text{ if } i(w) = n \\
(\nu^{mn})^T \phi(z_w) + b^{mn} \leq -1 + \xi^{mn}_w, \text{ if } i(w) = n \\
\xi^{mn}_w \geq 0.
\]

Here, minimizing \( \frac{1}{2} (\nu^{mn})^T \nu^{mn} \) means that we would like to maximize \( 2/||\nu^{mn}|| \), the margin between two groups of data. When data are not linearly separable, the penalty term \( C \sum_w \xi^{mn}_w (\nu^{mn})^T \phi(z_w) \) is introduced which can reduce the number of training errors. \( C \) is the control parameter which is usually obtained by cross–validation over the training data. here, \( \phi \) denote the feature mapping function. Since the algorithm is written entirely in terms of inner products, an appropriate Kernel formulation is necessary, i.e., \( K(z_w, \phi(z_u)) \equiv \phi(z_w)^T \phi(z_u) \). For this work, we consider radial basis functions (RBF): \( K(z_w, z_u) = \exp(-\lambda \| z_w - z_u \|^2) \), where the parameter \( \lambda > 0 \) is also optimized via cross–validation.

### 4.3 RF AMC

In contrast to SVMs, RF classifiers [5] work, unmodified with any number of classes. More importantly, RF classifiers generate probabilistic output for class prediction of the test data, as they return not just a single class point prediction but an entire class distribution. Here we consider the random forest version as used by Breiman [10], which combines bagging and random feature selection. More formally, a bootstrap procedure is used to create multiple randomized training sets, by sampling with replacement from the original training set, such that the bootstrap replicates contain an equal amount of instances as the original instance set. The random decision tree learner has a parameter \( 1 < v < p \) which is a positive integer. At each step of the construction of the tree, \( v \) variables are selected uniformly at random from the \( p \) candidates \( z_{w(1)}, \ldots, z_{w(p)} \). Recall that each training observation \( z_w \in \mathbb{R}^p \).
To illustrate the performance of our AMC algorithm, we follow the MCS specifications from IEEE 802.11n with two receive antennas and two transmit antennas. In Table 5.1, we list eight MCS where each class \(i\) for \(MCS_i\) defines a convolutional coding rate, QAM modulation order \(M\), and number of spatial streams \(N_s\) to determine the physical layer data rate \(R_i\). For this work, we assume \(N_s = 1\) only. The dataset\(^2\) is obtained for highly frequency selective (8–tap) 20 MHz channel. The channels are generated, both for training and test sets, with 1000 realizations of the \(N_r \times N_t\) channel matrix for each tap. Every matrix element for every tap is complex Gaussian distributed with zero mean and unit variance. Furthermore, each of the 1000 channel realizations in the training and test sets is combined with 32 values of \(E_s/\sigma^2\), resulting in 32000 observations in each of the training and test sets.

### 5.1 \(k\)–NN

For training the \(k\)–NN, a brute–force search over all possible ordered subcarrier feature sets up to four dimensions are performed. This is completed as follows for fixed feature space dimension \(d\), and fixed neighbor size \(k\). In particular, the sub–carrier indices with the lowest number of errors for each dimension \(d\) are chosen. In Table 5.1, the result for this search for 8–tap channels are listed. For each channel and \(E_s/\sigma^2\) realization in the test set, we compute the postprocessing SNR in (3), sort according to (4), and select the ordered postprocessing SNR values corresponding to the sub–carrier indices for a fixed \(d\) to construct a query \(\mathbf{q}\). The class labels for each observation in test set are then obtained following the algorithm is Section 4.1. In Fig. 1, we present the results for \(d = 2\) and \(d = 4\). We observe that as \(k\) increases, the classification accuracy increases as expected. It can be mentioned that as \(k\) grows, the complexity of this scheme increases exponentially. Also \(k\)–NN requires lots of memory storage.

\(^2\)The dataset is obtained from Robert Daniels, author of [2].
Table 2: BEST SUBCARRIER ORDERING SETS FOR AMC in IEEE 802.11n [2]

<table>
<thead>
<tr>
<th>Ns</th>
<th>d</th>
<th>8 Tap (n₁, n₂, ..., n₄)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>(17)</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>(10,31)</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>(6,17,39)</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>(5,12,24,38)</td>
</tr>
</tbody>
</table>

Table 3: Accuracy and Cross-Validation Result for SVM

<table>
<thead>
<tr>
<th>d</th>
<th>log₂ C</th>
<th>log₂ γ</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>-16</td>
<td>75.4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>-17</td>
<td>77.4</td>
</tr>
</tbody>
</table>

5.2 SVM

For simulating data for SVM, we use LibSVM [11]. Before, performing SVM, we scale the data to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. For example, scale the training data from [0,10] to [0,1]. For RBF Kernel, we perform cross–validation over C and λ. Then we use the best parameters C and λ to train the whole data set. A Fine grid search for C and λ is shown in Fig. 3. In Table 5.2, we show accuracy and cross–validation result for feature dimension d = 2 and d = 4. Note that for training SVM, we use the best sub–carrier indices for different feature dimension d, listed in Table 5.1. This choice is motivated by the fact the authors in [3] applied the same feature sets for training multiclass SVM. However, note that the multiclass SVM in [3] use asymmetric weights for different streams whereas in our work we assume equal weight as all streams have same modulation orders.

5.3 RF classifier

Application of RF classifier for link adaptation is not available in the literature. Here, we do not assume the feature set listed in Table 5.1 is optimum for RF classifier as well. For training the RF, we follow Breiman’s [10] approach, where many bootstrap replicas of the dataset are generated and decision trees are grown on these replicas. We draw 50% data at random with replacement for training each tree and select √d features at random for each decision split. Note that for applying RF to the link adaptation, our main objective is to identify best feature set for RF classifier and accurate low dimensional mapping. From the out-of–bag importance of feature sets in Fig. 3, we observe that out of N = 52 ordered sub–carrier SNRs, γ₁ and γ₃₃ are most important. Now, we grow a larger ensemble on this reduced feature set. In Fig. 4, we observe that compared to the choice in Table 5.1 for d = 2, the important features obtained from out-of-importance chart results in better error performance.

Remark2: We observe that all three schemes result in more or less consistent accuracy level, with SVM resulting in a slightly higher accuracy. Note that obtaining a very high classification accuracy is difficult in wireless link adaptation as performance is governed by many parameters. The study in this work is based on the fact that FER depends on ordered post–processing SNR. However, other practical impairments such as non–Gaussian noise, non–linearity in the transmit power amplifiers, channel estimation errors may also cause accuracy to fall. Hence, investigation of effect of these parameters may be incorporated into feature set to better predict appropriate class according to the channel fading realizations. Out of three algorithms, k–NN has a major disadvantage regarding neighbor search and memory requirements. In SVM, often less training is needed due to reduced
number of support vectors compared to training data size, hence it is much more efficient compared to $k$–NN. In the proposed RF classifier, the two dimensional feature set results in significant amount of training overhead and larger efficiency.

6 Conclusions

In the paper, we study three different supervised classification algorithms: $k$–NN, SVM, and RF for link adaptation in MIMO–OFDM wireless links. In particular, we consider IEEE 802.11n channel models for training and testing the classifiers. For training $k$–NN and SVM, we used the sub–carrier indices proposed in [2] for different feature dimensions. In RF classifier, we locate two important feature dimensions and use them for training larger ensemble of trees. All the methods result is accuracy close to 76%. We did not consider channel estimation errors in this work which can be good topic for future work. Overall, we feel more investigation is necessary in identifying critical system parameters other than post–processing SNR to boost the accuracy of the considered classifiers.

References