Machine Learning for Link Adaptation in Wireless Networks

Robert W. Heath Jr.
The University of Texas at Austin
http://www.profheath.org
April 2011

In collaboration with Robert C. Daniels, Kuma Signals LLC
It is an Exciting but Scary Time in Wireless

Almost 2X / year growth in data traffic projected for years to come

How is wireless evolving to meet performance demands?
Cellular Architectures are Becoming More Complex

Expanding network complexity

Multiuser / multicell / femtocell / relay are changing topology

PHY techniques e.g., MIMO, OFDM, becoming understood

It’s not the size (of your technology) but how you use it
Link Adaptation Remains a Frontier

Spatial configuration

Frequency configuration

Architecture

Impairments
MIMO-OFDM AMC

**Possible Modes**

$(1/2, 4$-QAM, 2 streams)

**AMC** = Adaptive modulation and coding

- **Convolutional Coding**
- **Interleaving**
- **Modulation**
- **MIMO OFDM**

**Code rate**

**Constellation Size**

**# Streams**

**SNR**
Conventional Approach

\[ \{H[k]\}_{k=0}^{N-1} \]

- Threshold-based/look-up-table offline AMC
  - SNR based adaptation [GolChu98]
  - Link quality metrics in coded OFDM [SimoEtA06]
  - Link quality metrics in MIMO-OFDM [KanJen07]
Limitations

- Difficulty in determining the optimum mappings
  - Challenging with MIMO, OFDM, coding, interleaving
- Model mismatch
  - Variations in hardware, e.g. phase noise, nonlinearities
Proposed Solution

- **Machine learning** with real-time training
  - Adapt classification over time based on performance data

- **Advantages**
  - Can collect data in real-time
  - Adapt to real impairments over time
Machine Learning and Classification

- Supervised learning: training data gives correct mapping
- Classification: mapping of features to labels (i.e. class)
  - Features represent state of system being modeled
Machine Learning in Wireless

- Link adaptation response evolves from performance training data
- More flexible: allows multi-dimensional link quality metrics
- No need for excessive design to capture potential impairments
Challenges in Learning for LA

- What is the right feature space?
  - Avoid curse of dimensionality
  - Requires domain knowledge of wireless communication
- How is reliability incorporated?
  - Traditional classification strives for 100% correct
  - Link adaptation maximizes rate subject to error constraints
- How does adaptation proceed in real-time?
  - Memory, reliability, adaptation to impairments
Part #1: MIMO-OFDM Link Adaptation through Offline Learning

Traditional MIMO AMC Concept

Goal: Maximize throughput, while maintaining a target packet error rate

1. Transmitter Probes Channel
2. Recommended Parameter Feedback
3. Packet Transmission w/ Recommended Parameters

Recommended parameters may include modulation order, coding rate, spatial multiplexing order, precoding, bandwidth, carrier frequency, etc.

AMC (adaptive modulation and coding)
MIMO-OFDM Link Adaptation

Prior Work

• FER approximation techniques
  ▶ BICM, pairwise error probability [Song2006, Sung2007, Li2008]
  ▶ FER expressions are not universal, accurate, or practical
  ▶ Fading assumptions, bounds on coding performance
  ▶ Model irregularities [Razavi1997]

• Simplistic link quality metrics (LQM)
  ▶ EESM (most popular) [Tsai2003, Liu2007]
  ▶ Mutual information, capacity [Ericsson2003, Sayana2007, Choi2008]
Offline Learning for AMC Concept

1. Transmitter Probes Channel
2. Receiver Polls Learning Engine
3. Machine Learning Algorithm Recommends Parameters
4. Recommended Parameter Feedback
5. Packet Transmission w/ Rec. Params

Algorithm trained offline, prior to implementation
System Model and Link Adaptation

- Input/output model

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

- Post-processing SNR

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

- Coding over subcarrier and spatial streams

- Reliability constraint (frame error rate): FER(·) ≤ F

- Throughput: (1-FER)x(Data Rate)

- Modulation and Coding Scheme (MCS)
  - Same constellation over all subcarriers/streams
  - Adaptive modulation and coding (AMC) [Catreux2002]
New FER Expression

- Convolutional coding performance with interleaving

\[
\text{FER} \triangleq 1 - (1 - P_B)^{N_f} \quad P_B < \frac{1}{q} \sum_{d=d_f}^{\infty} C_d P_d
\]

\[
Z(\gamma) \triangleq \exp(-\gamma)
\]

\[
P_B < \frac{1}{q} \sum_{d=d_f}^{\infty} \sum_c \prod_{l=1}^{d} Z \left( \gamma \left[ s_1(\ell_l^{(c)}), \text{mod} \left( s_2(\ell_l^{(c)}), N \right) \right] \right)
\]

\[
\max_{c \in S_d} \left\{ \prod_{l=1}^{d} Z \left( \gamma \left[ s_1(\ell_l^{(c)}), \text{mod} \left( s_2(\ell_l^{(c)}), N \right) \right] \right) \right\} \leq \max_{a_1, a_2, \ldots, a_d} \left\{ \prod_{l=1}^{d} Z \left( \gamma[a_l, n_l] \right) \right\}
\]

\[
P_B < \frac{1}{q} \sum_{d=d_f}^{d_L} C_{d,L} \prod_{l=1}^{d} Z \left( \gamma^{(l)} \right)
\]

- Key observation: FER bound does not depend on subcarrier location!
Sorted Post-Processing SNR
Link Quality Metrics

- Sorting over subcarriers and streams uncovers correlation
- Brute-force search used to determine best indices
- Increase dimensions to closely approximate profile
  - Does not necessarily correspond to improved performance
  - Curse of dimensionality for feature sets

Pick fixed indices to represent entire profile
k-NN Classification Algorithm

• Modification of \(k\)-Nearest Neighbor (\(k\)-NN)
  - Compute the feature for a channel realization
  - Find \(k\) closest features in each MCS database
  - Determine \(e_i\), the number of \(k\)-nearest neighbors corresponding to unsuccessful transmissions

\[
\text{Target FER} = \mathcal{F}
\]
Performance

- k-NN classifiers
- 10% FER constraint with 2x2 IEEE 802.11n training set
- EESM with learning outperforms look up table approach
- 4-dim ordered SNR feature sets provide superior performance
Part #2:
Online Link Adaptation

Prior Work & Motivation

- Link adaptation with backoff characterization [Das2007]
- Stochastic learning and greedy optimization [Misra2006, Aggarwal2009]

PA nonlinearity (variable backoff)

Not captured in Monte Carlo simulations which define AMC thresholds
- May evolve over time due to interference conditions, device temperature
Online Learning for AMC Concept

1. Transmitter Probes Channel
2. Receiver Polls Database
3. Machine Learning Algorithm in Database Recommends Parameters

4. Recommended Parameter Feedback
5. Packet Transmission w/ Rec. Params
6. Add Packet Performance to Database
Classification: MIMO OFDM AMC

- **Classes**
  - Modulation and coding schemes (MCS)

- **Challenges**
  - What are the right features?
  - What is the optimization criterion?
  - How can past data be exploited?

### IEEE 802.11n MCS List (2x2)

<table>
<thead>
<tr>
<th>MCS_i</th>
<th>N_s</th>
<th>M</th>
<th>Code Rate</th>
<th>R_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>i = 0</td>
<td>1</td>
<td>2</td>
<td>1/2</td>
<td>6.5 Mbps</td>
</tr>
<tr>
<td>i = 1</td>
<td>1</td>
<td>4</td>
<td>1/2</td>
<td>13.0 Mbps</td>
</tr>
<tr>
<td>i = 2</td>
<td>1</td>
<td>4</td>
<td>3/4</td>
<td>19.5 Mbps</td>
</tr>
<tr>
<td>i = 3</td>
<td>1</td>
<td>16</td>
<td>1/2</td>
<td>26.0 Mbps</td>
</tr>
<tr>
<td>i = 4</td>
<td>1</td>
<td>16</td>
<td>3/4</td>
<td>39.0 Mbps</td>
</tr>
<tr>
<td>i = 5</td>
<td>1</td>
<td>64</td>
<td>2/3</td>
<td>52.0 Mbps</td>
</tr>
<tr>
<td>i = 6</td>
<td>1</td>
<td>64</td>
<td>3/4</td>
<td>58.5 Mbps</td>
</tr>
<tr>
<td>i = 7</td>
<td>1</td>
<td>64</td>
<td>5/6</td>
<td>65.0 Mbps</td>
</tr>
<tr>
<td>i = 8</td>
<td>2</td>
<td>2</td>
<td>1/2</td>
<td>13.0 Mbps</td>
</tr>
<tr>
<td>i = 9</td>
<td>2</td>
<td>4</td>
<td>1/2</td>
<td>26.0 Mbps</td>
</tr>
<tr>
<td>i = 10</td>
<td>2</td>
<td>4</td>
<td>3/4</td>
<td>39.0 Mbps</td>
</tr>
<tr>
<td>i = 11</td>
<td>2</td>
<td>16</td>
<td>1/2</td>
<td>52.0 Mbps</td>
</tr>
<tr>
<td>i = 12</td>
<td>2</td>
<td>16</td>
<td>3/4</td>
<td>78.0 Mbps</td>
</tr>
<tr>
<td>i = 13</td>
<td>2</td>
<td>64</td>
<td>2/3</td>
<td>104.0 Mbps</td>
</tr>
<tr>
<td>i = 14</td>
<td>2</td>
<td>64</td>
<td>3/4</td>
<td>117.0 Mbps</td>
</tr>
<tr>
<td>i = 15</td>
<td>2</td>
<td>64</td>
<td>5/6</td>
<td>130.0 Mbps</td>
</tr>
</tbody>
</table>
Creating a Database of Past Performance

Database holds past packet information and associated channel feature

ARQ and ACK/NACK messages determine successful packets

Features associated with successful packets

Features associated with packet failures

---

DATA

TX Node

RX Node

(N)ACK

---

Index | Timestamp | Features (d-dimensional)

MCS i Database

---

Sunday, May 1, 2011
Training Online: Challenges

- Use received frames to create training data for classifiers in real-time
- Information from received packets is limited
  - Only single MCS for a single channel
  - FER information not available (only success/failure)

Each frame only contains information for a single MCS for a single channel for one TX/RX pair.
Training Data for Online Classifiers

- Instead of a single-machine approach, use many binary classifiers
  - Proposed one-vs-none binary classification
  - Predict whether packet fails or succeeds
  - Failure reinforces null MCS
  - Success reinforces chosen MCS
  - Number of classifiers
- How do we address the FER constraint?
  - Calculate posterior class probability of null MCS

\[
\hat{\text{FER}}_i = \Pr[\text{failure}_i | \mathbf{q}] \quad \text{is posterior class probability for null MCS when compared to } \text{MCS}_i \text{ given feature set query } \mathbf{q}
\]
Database Learning

- Idea: use database to save relevant information
- Databases needed for each MCS, packet length, TX/RX pair

Use $k$-NN to extract posterior class probability of frame failure

$$\overline{FER_i} = \frac{e_i}{k}$$

$0 \leq e_i \leq k$ number of nearest neighbors due to frame failure
Preservation and Exploration

- Use density metrics to preserve feature set diversity
  - Density of $D$ feature sets, $\{z_l\}_{l=1}^D$ relative to reference $z_0$
    
    $$
    \rho(z_0, \{z_l\}_{l=1}^D) \triangleq \frac{D}{(\max_{l\in\{1,2,...,D\}} \{\sum_{m=1}^p |z_{l,m} - z_{0,m}|^2\})^{p/2}}
    $$

- Explore to prevent conservative MCS selection
  - Lower bounds on FER, low network utilization

$E(\hat{i})$ is randomly selected MCS in candidate set $\mathcal{E}(\hat{i})$
Reducing Complexity Through Support Vector Machines (SVMs)

- Problem: database learning search/memory complexity
- Observation: database learning does not explicitly construct class boundaries
- Solution: use SVM classifiers which offer potential to reduce complexity
  - SVMs preserve information only about class boundaries
Support Vector Machines (SVMs)

- SVMs maximize the margin between classes in training set
  - Boundary captured w/ inference function \( h(q) = \sum_{n=1}^{N} y_n \alpha_n \kappa(x_n, q) + b \)
- Kernel function generalizes boundary shape between classes
- Dual optimization variable, \( \alpha_n \), reveals concept of support vector

\[ N = \text{training set dimension} \]
\[ y_n = \text{training set entry } n \text{ class label} \]
\[ x_n = \text{training set entry } n \text{ feature set} \]
\[ b = \text{inference function bias} \]
\[ \kappa(x_n, x_{n'}) = \text{kernel function} \]
\[ \kappa_{\text{lin}}(x_n, x_{n'}) := x_n^T x_{n'} \]
\[ \kappa_{\text{rbf}} := \exp \left( -\gamma \|x_n - x_{n'}\|^2 \right) \]
Addressing the FER Constraint

- Usually inference outputs mapped directly to class (+1/-1)
  - This only predicts packet success or failure, not FER
- Normally distributed inference outputs yield FER form
  \[
  \Pr[y = 1|h(q)] = \left(1 + \exp(A_1 h(q)^2 + A_2 h(q) + A_3)\right)^{-1}
  \]
  constants $A_1$, $A_2$, and $A_3$
- Propose regression onto Platt sigmoid to determine FER
  - Platt sigmoid: $(1 + \exp(B_1 h(q) + B_2))^{-1}$, constants $B_1$ and $B_2$
  - Platt sigmoid maintains monotonicity for all constant values
Online AMC with SVMs: Procedure

- **Optimization**
  1. Complete optimization to define inference functions for each MCS
     - Training data for optimization captured from frame transmissions
  2. Regress inference function outputs on Platt sigmoid

- **Prediction**
  1. Complete inference mapping for prediction feature set for each MCS
  2. Use regression functions to predict FER for each MCS
  3. Select best MCS using FER constrained adaptation criterion
Complexity Comparison

- Prediction complexity very critical for practical implementation
- SVM with linear kernels offers best complexity 🌟

\[ M = \# \text{MCS}, p = \text{feature set dim}, b = \text{bit resolution of data} \]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Processing</th>
<th>Memory (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM (linear)</td>
<td>( M(p+2) ) mults</td>
<td>( M b(8p+4) )</td>
</tr>
<tr>
<td></td>
<td>( M ) divs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M(p+4) ) adds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M ) exp maps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M ) length sort</td>
<td></td>
</tr>
<tr>
<td>SVM (RBF)</td>
<td>( M(p+303) ) mults</td>
<td>( M b(300p+5) )</td>
</tr>
<tr>
<td></td>
<td>( M ) divs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M(p+304) ) adds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M \times 2 ) exp maps</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M ) length sort</td>
<td></td>
</tr>
<tr>
<td>kNN</td>
<td>( M(p300+1) ) mults</td>
<td>( M b(300p+1) )</td>
</tr>
<tr>
<td></td>
<td>( M(2p300+k+1) ) adds</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M ) divs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M ) 300-length sorts</td>
<td></td>
</tr>
<tr>
<td></td>
<td>( M ) length sort</td>
<td></td>
</tr>
</tbody>
</table>

RBF kernels do not limit number of support vectors
Memory for alternatives uses entire training set
kNN processing scales with training set size
Potential Chipset Implementation

Training/optimization/prediction all on host; bad for latency critical adaptation.

Example WiFi Chipset

Based on Ericsson CW1200 IEEE 802.11a/b/g/n
Demonstrating the Advantage of Online AMC

- 2x2 IEEE 802.11n including PA nonlinearity and Laplacian noise
  - Rapp solid-state model with \( p=3, \{0,2\} \) dB backoff
- Offline AMC from Part #1, online AMC through SVMs

**Graphs:**

- **1-tap channel**
- **4-tap channel**

**Axes:**
- Throughput after Link Adaptation (in Mbps)
- Average SNR (dB)

**Legend:**
- Nonlinear Power Amplifier (0 dB Backoff)
- Nonlinear Power Amplifier (2 dB Backoff)
- Laplacian Noise
- NPA with Online Learning (0 dB Backoff)
- NPA with Online Learning (2 dB Backoff)
- Laplacian Noise with Online Learning
Part #3: Prototyping Online Link Adaptation

MIMO-OFDM System Demonstration

• Adaptation procedure

RTS sent with training symbols for channel estimation
DATA payload transmitted using received adaptation setting

CTS returns adaptation setting after learning database query
ACK and receiver measurements update learning database
MIMO-OFDM Demonstration

- Bi-directional ping traffic between nodes
- Online learning LA with subcarrier ordering
- System parameters:
  - IEEE 802.11n 2x2 physical layer
    - MCS 0-15 corresponding to data rates 6.5-130 Mbps
  - IEEE 802.11 DCF medium access control
  - GNU radio software defined radio
  - Universal software radio peripheral hardware
Measurements

- Measured performance in repeatable channels
- Compared to offline/online baseline ($k$-NN/ARF)
- Online learning clearly best in practice

### Static Channel

<table>
<thead>
<tr>
<th>Packet Sequence Index</th>
<th>Throughput (Mbps)</th>
<th>Packet Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offline Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Dynamic Channel

<table>
<thead>
<tr>
<th>Packet Sequence Index</th>
<th>Throughput (Mbps)</th>
<th>Packet Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Offline Learning</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Video Demo

• http://www.youtube.com/watch?v=mnK9Pft4fQw
Conclusions

• Summary
  ▶ Machine leaning is flexible approach for link adaptation
  ▶ Offline learning has more accurate adaptation
  ▶ Online learning offers flexibility and reduced memory

• Future work
  ▶ Incorporating packet losses due to interference
  ▶ More advanced PHY algorithms
  ▶ Sophisticated network topologies
Our Related Publications


References (1/3)

References (2/3)


References (3/3)


