Learning to Modulate for Non-coherent MIMO

Ye Wang, Toshiaki Koike-Akino

IEEE ICC 2020
International Conference on Communications

This document does not contain Technology as defined in EAR Part 772.
Overview

- Apply machine learning to design modulation and detection
  - Target: space-time constellations for non-coherent MIMO channel
  - Recent trend: encode/decode with Deep Neural Networks (DNN)

- Simulation-driven, end-to-end, encoder/decoder optimization
  - Minimizing cross-entropy loss $\Leftrightarrow$ maximizing mutual info
  - We compare DNN-based vs DNN-free systems

- Learned schemes can outperform traditional designs at some SNRs
  - DNNs can be avoided altogether while keeping similar performance
  - Feasibility of non-coherent MIMO with only two time slots
Non-Coherent MIMO Channel

Using \( m \) Tx and \( n \) Rx antennas over \( L \) time slots

\[
Y = HX + Z
\]

- Signal \( X \in \mathbb{C}^{m \times L} \) sent over \( m \) Tx antennas and \( L \) time slots
- Unknown block fading channel \( H \in \mathbb{C}^{n \times m} \) iid \( \sim \mathcal{CN}(0, 1/m) \)
- Gaussian noise \( Z \in \mathbb{C}^{n \times L} \) iid \( \sim \mathcal{CN}(0, \sigma^2) \)
- Power constraint: \( \mathbb{E}[||X||^2/(mL)] = 1 \), average SNR = \( 1/\sigma^2 \)
- Receive \( Y \in \mathbb{C}^{n \times L} \) on \( n \) Rx antennas

Goal: design \( k \)-bit modulation and non-coherent detection scheme

\[
m \in \{0, 1\}^k \rightarrow \text{ENC} \rightarrow X \rightarrow \text{MIMO} \rightarrow Y \rightarrow \text{DEC} \rightarrow \hat{m}
\]
Encoding Signal Constellation via a Lookup Table

For small $k$ (#bits), lookup table most effective and efficient

- Encoder: $\{0, 1\}^k \rightarrow \mathbb{C}^{m \times L}$ fully captured by table $C \in \mathbb{C}^{2^k \times m \times L}$
- Subtract centroid (mean across first axis) to get centered $\overline{C}$
- Normalize average power of codebook $\overline{\tilde{C}} := \overline{C} \sqrt{2^k m L / \|C\|}$
- For message $\mathbf{m} \in \{0, 1\}^k$, signal $X_{\mathbf{m}}$ is selected from $\tilde{C}$
Encoding Signal Constellation via a Lookup Table

For small $k$ (#bits), lookup table most effective and efficient

- Encoder: $\{0, 1\}^k \rightarrow \mathbb{C}^{m \times L}$ fully captured by table $C \in \mathbb{C}^{2^k \times m \times L}$
- Subtract centroid (mean across first axis) to get centered $\overline{C}$
- Normalize average power of codebook $\tilde{C} := \overline{C} \sqrt{2^k m L / \|C\|}$
- For message $m \in \{0, 1\}^k$, signal $X_m$ is selected from $\tilde{C}$

Common alternative: DNN-encoder with one-hot message input

- Vastly over-parameterizes the codebook with extraneous layers
- Binary input encoding would scale better for large $k$
- Our work uses lookup table to avoid encoder DNN
Decoding with or without DNN

We optimize two soft-decision decoders: DNN-based vs DNN-free

- Both output unnormalized, log-likelihoods for each message
- Apply softmax to yield approximate posterior $P^\theta_m|Y$

Neural Network (NN) Decoder:

- $\theta$ is network parameters
- Used Multi-Layer Perceptron (MLP) or Residual MLP (ResMLP)
- Blind decoder trained end-to-end with cross-entropy loss

Pseudo-ML (pML) Decoder: based on orthonormal code ML decoding

- If codeword orthonormal, i.e., $\forall m, X_m X_m^\dagger = L \cdot I_m$, then $\|Y X_m^\dagger\|_2$ is proportional to unnormalized, log-likelihood $\log \alpha P(Y|m)$
- Outputs $\{\theta \|Y X_m^\dagger\|_2\}_{m \in \{0, 1\}^k}$ where $\theta > 0$ captures confidence

- Requires additional codebook orthonormality constraint
Decoding with or without DNN

We optimize two soft-decision decoders: DNN-based vs DNN-free

- Both output unnormalized, log-likelihoods for each message
- Apply softmax to yield approximate posterior $P^\theta_{m|Y}$

Neural Network (NN) Decoder: $\theta$ is network parameters

- Used Multi-Layer Perceptron (MLP) or Residual MLP (ResMLP)
- Blind decoder trained end-to-end with cross-entropy loss
Decoding with or without DNN

We optimize two soft-decision decoders: DNN-based vs DNN-free

- Both output unnormalized, log-likelihoods for each message
- Apply softmax to yield approximate posterior $P^\theta_{m|Y}$

Neural Network (NN) Decoder: $\theta$ is network parameters

- Used Multi-Layer Perceptron (MLP) or Residual MLP (ResMLP)
- Blind decoder trained end-to-end with cross-entropy loss

Pseudo-ML (pML) Decoder: based on orthonormal code ML decoding

- If codeword orthonormal, i.e., $\forall m, X_mX_m^\dagger = L \cdot I_m$, then $\|YX_m^\dagger\|^2$ is proportional to unnormalized, log-likelihood $\log \alpha P(Y|m)$
- Outputs $\{\theta\|YX_m^\dagger\|^2\}_{m \in \{0,1\}^k}$ where $\theta > 0$ captures confidence
- Requires additional codebook orthonormality constraint
Cross-Entropy Loss Training Maximizes Mutual Info

\[ \text{min}_{C, \theta} \mathbb{E} \left[ - \log P_{m|Y}(m|Y) \right] \]

End-to-end optimization with cross-entropy loss
Cross-Entropy Loss Training Maximizes Mutual Info

\[
m \rightarrow \text{ENC}_C \rightarrow X_m \rightarrow \text{MIMO} \rightarrow Y \rightarrow \text{DEC}_{\theta} \rightarrow P_{m|Y}^\theta
\]

End-to-end optimization with cross-entropy loss

\[
\min_{C,\theta} \mathbb{E} [-\log P_{m|Y}(m|Y)]
\]

Equivalent to maximizing mutual info \(I(m;Y)\) since

- \(\mathbb{E} [-\log P_{m|Y}(m|Y)] = \mathcal{H}(m|Y) + \text{KL}(P_{m|Y}||P_{m|Y}^\theta)\)
- \(I(m;Y) = \mathcal{H}(m) - \mathcal{H}(m|Y)\), with constant \(\mathcal{H}(m) = k\)
Cross-Entropy Loss Training Maximizes Mutual Info

End-to-end optimization with cross-entropy loss

\[
\min_{C, \theta} E \left[ - \log P_{m|Y}^\theta (m|Y) \right]
\]

Equivalent to maximizing mutual info \( I(m; Y) \) since

- \( E \left[ - \log P_{m|Y}^\theta (m|Y) \right] = \mathcal{H}(m|Y) + \text{KL}(P_{m|Y} \parallel P_{m|Y}^\theta) \)
- \( I(m; Y) = \mathcal{H}(m) - \mathcal{H}(m|Y) \), with constant \( \mathcal{H}(m) = k \)

For pML decoder, we also enforce orthonormality with soft penalty

\[
\min_{C, \theta} E \left[ - \log P_{m|Y}^\theta (m|Y) \right] (1 + \lambda \ell(C)), \quad \lambda > 0
\]

\[
\ell(C) := \frac{1}{2^k m^2} \sum_{m \in \{0,1\}^k} \| X_m X_m^\dagger / L - I_m \|^2
\]
Experimental Evaluation

- Non-coherent MIMO parameters
  - Bits $k \in \{2, 4, 6, 8\}$, time slots $L \in \{2, 4\}$
  - RX $n \in \{2, 3, 4\}$, TX $m = n$ for $L = 4$, TX $m = 2$ for $L = 2$
- Encoders/decoders optimized over a range of hyperparameters
  - MLP: varied depth and width of fully-connected layers
  - ResMLP: MLP with additional skip connections and batch-norm
  - pML: varied $\lambda$ parameter controlling code orthonormality
- Compare performance vs codes constructed by [Liang, Xia '02]
  - Existing Grassmann code designs require $L > m$
  - We demonstrate novel feasibility of learning for $L = m = 2$
- Cross-entropy loss gives approximate lower-bound on throughput:
  \[
  k - \mathbb{E}
  \left[
    - \log P_{m|Y}(m|Y)
  \right] \\
  L \approx 
  \frac{I(m; Y)}{L}
  \]
  - Compare with capacity lower-bounds of [Yang, Durisi, Riegler '13]
Comparison for $L = 4$ Time Slots, TX-RX $(m, n) = (2, 2)$
NN Decoder Performance for $L = 4$
NN Decoder Throughput for $L = 4$
NN Decoder Performance for $L = 2$
NN Decoder Throughput for $L = 2$
pML Decoder Performance for $L = 4$
pML Decoder Throughput for $L = 4$
Codebook via pML Decoder for $k = 2$, $L = 4$, $(m, n) = (4, 4)$
Summary

- Applied learning to non-coherent MIMO modulation and detection
  - Simulation-driven, end-to-end, space-time constellation optimization
  - Minimizing cross-entropy loss $\Leftrightarrow$ maximizing mutual info
  - We offer a DNN-free system as alternative to using DNN

- Learned schemes can outperform traditional designs at some SNRs
  - DNNs can be avoided altogether while keeping similar performance
  - Non-coherent MIMO feasible with only two time slots