MIMO detection algorithms

EE359 staff

November 14, 2017

Consider the MIMO channel

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n},$$

where $\mathbf{y} \in \mathbf{C}^{M_r}, \mathbf{x} \in \mathcal{X}^{M_t}$ and $\mathbf{H} \in \mathbf{C}^{M_r \times M_t}$. Here $n_i \sim \mathcal{CN}(0, 1)$ and $\mathrm{E}[\mathbf{x}\mathbf{x}^H] = \frac{\rho}{M_t}\mathbf{I}$. The receiver estimates $\hat{\mathbf{x}}(\mathbf{y})$, an estimate for the transmitted vector $\mathbf{x}$, based on its knowledge of the channel matrix $\mathbf{H}, \mathcal{X}$, and the observation $\mathbf{y}$.

We consider the following detection algorithms:

1. **maximum likelihood (ML)**: This is the optimal detector from the point of view of minimizing the probability of error (assuming equiprobable $\mathbf{x}$). The maximum likelihood detector with IID Gaussian noise at the receiver antennas solves the following problem.

$$\hat{\mathbf{x}}(\mathbf{y}) = \mathrm{argmin}_{\mathbf{x} \in \mathcal{X}^{M_t}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2. \tag{1}$$

The minimization is over $\mathbf{x} \in \mathcal{X}^{M_t}$, i.e. over all possible transmitted vectors. Unfortunately, solving this problem involves computing the objective function for all $\mathcal{X}^{M_t}$ potential values of $\mathbf{x}$. Hence the ML detector has prohibitive (exponential in M_t) complexity.

2. **linear receivers**: The complexity of linear detectors is the same as the complexity of inverting or factorizing a matrix of dimensions $M_r \times M_t$, hence the name. They work by spatially decoupling the effects of the channel by a process known as MIMO equalization. This involves multiplying $\mathbf{y}$ with a MIMO equalization matrix $\mathbf{A} \in \mathbf{C}^{M_t \times M_r}$ to get $\tilde{\mathbf{x}}(\mathbf{y}) \in \mathbf{C}^{M_t}$. To get the estimate $\hat{\mathbf{x}}(\mathbf{y})$ from $\tilde{\mathbf{x}}(\mathbf{y})$, we perform coordinatewise decoding of $\tilde{\mathbf{x}}(\mathbf{y})$. The coordinatewise decoding operation is given by:

$$\hat{\mathbf{x}}(\mathbf{y})_i = \mathrm{argmin}_{s \in \mathcal{X}} |\tilde{\mathbf{x}}(\mathbf{y})_i - s| \text{ for all } i, \tag{2}$$

i.e., it maps each coordinate to the closest constellation point.

We describe two common ways of obtaining $\tilde{\mathbf{x}}(\mathbf{y})$ from $\mathbf{y}$:

- **zero forcing (ZF)**: In zero forcing, the following problem is solved:

$$\tilde{\mathbf{x}}(\mathbf{y}) = \mathrm{argmin}_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2. \tag{3}$$

Comparing with (1), we note that the constellation constraints on $\mathbf{x}$ have been removed. This significantly reduces the complexity. For a square invertible matrix, the solution is given by

$$\tilde{\mathbf{x}}(\mathbf{y}) = \mathbf{H}^\dagger \mathbf{y},$$

where $\mathbf{H}^\dagger$ is just $\mathbf{H}^{-1}$ if the matrix is square and invertible. If the matrix is not invertible or not square we use the **pseudo inverse** instead. When $M_t \leq M_r$, and there are at least M_t linearly independent columns in $\mathbf{H}$ (we see this case often) the pseudo inverse (sometimes called the Moore-Penrose pseudoinverse) is given by

$$\mathbf{H}^\dagger = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H.$$

The complexity of obtaining $\mathbf{H}^\dagger$ from $\mathbf{H}$ is roughly cubic in M_t for a square matrix. However obtaining $\hat{\mathbf{x}}(\mathbf{y})$ from $\tilde{\mathbf{x}}(\mathbf{y})$ is done in a time linear in M_t.

- **linear minimum mean squared error (L-MMSE)**: Zero forcing can cause noise amplification if the minimum singular value of $\mathbf{H}$ is too small. This may be quantified by the notion of the condition number of the matrix $\mathbf{H}$. The condition number of the matrix $\mathbf{H}$ is a measure of the relative magnitudes of the singular values of $\mathbf{H}$. It is defined as the ratio between the largest and the smallest singular values of $\mathbf{H}$. When the condition number is unity or close to unity, the matrix is said to be well conditioned. When the condition number is large, the matrix is ill conditioned.

  To reduce the sensitivity of linear receivers to the conditioning of the matrix $\mathbf{H}$, we can add a regularization term to the objective function in (3) (in red).

  $$\tilde{\mathbf{x}}(\mathbf{y}) = \text{argmin}_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda\|\mathbf{x}\|^2,$$

  for some $\lambda > 0$. The solution to this is given by

  $$\tilde{\mathbf{x}}(\mathbf{y}) = (\mathbf{H}^H\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^H\mathbf{y} = \mathbf{H}^H(\mathbf{H}\mathbf{H}^H + \lambda\mathbf{I})^{-1}\mathbf{y}.$$

  For $\lambda = \frac{M_t}{\rho}$, this is called the **L-MMSE** detector since it minimizes the mean squared error in the estimate of $\mathbf{x}$ among all linear detectors, i.e., it solves the following problem:

  $$\begin{aligned} \tilde{\mathbf{x}}(\mathbf{y}) &= \text{argmin}_{\mathbf{s}}\ \mathrm{E}_{\mathbf{x},\mathbf{n}}[\|\mathbf{s} - \mathbf{x}\|^2], \\ &\quad \text{such that } \mathbf{s} = \mathbf{A}\mathbf{y} + \mathbf{b}, \end{aligned}$$

  for any $M_t \times M_r$ matrix $\mathbf{A}$ and vector $\mathbf{b} \in \mathbf{C}^{M_t}$. Note that the minimization is only over all affine functions of $\mathbf{y}$, which is parametrized by $\mathbf{A}$ and $\mathbf{b}$. The expectation is over the randomness in $\mathbf{x}$ and $\mathbf{n}$ (the channel matrix $\mathbf{H}$ is assumed to be known and non random). If $\mathbf{x}$ were to be Gaussian (instead of being from discrete constellation points), this is also the MMSE detector.

  Compared to the ML detector, both the linear detectors are simpler to implement, but the BER performances are worse.

3. **sphere decoders (SD)**: The sphere decoder trades off performance versus complexity by controlling a parameter r. By choosing a large enough r, the performance of SD approaches that of the ML detector. For small r, the search space (and hence complexity) of the SD is much smaller than that of the ML detector, but it suffers a performance degradation as a result. The sphere detector exploits the following factorization of the matrix $\mathbf{H}$:

   $$\mathbf{H} = \mathbf{Q}\mathbf{R},$$

   where $\mathbf{Q}$ is unitary and $\mathbf{R}$ is upper triangular. Since the squared-distance norm does not change under multiplication by a unitary matrix $\mathbf{Q}^H$, we have

   $$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = \|\mathbf{Q}^H\mathbf{y} - \mathbf{Q}^H\mathbf{Q}\mathbf{R}\mathbf{x}\|^2 = \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{x}\|^2 = \sum_{i=M_t}^{1} |\tilde{\mathbf{y}}_i - \sum_{j=i}^{M_t} \mathbf{R}_{i,j}\mathbf{x}_j|^2,$$

   where $\tilde{\mathbf{y}} \triangleq \mathbf{Q}^H\mathbf{y}$, and we use the upper triangular nature of $\mathbf{R}$ in the last step. The sphere decoder solves the following problem.

   $$\hat{\mathbf{x}}(\mathbf{y}) = \text{argmin}_{\mathbf{x} \in \mathcal{X}^{M_t}, \|\mathbf{Q}^H\mathbf{y} - \mathbf{R}\mathbf{x}\| \le r} \|\mathbf{Q}^H\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \tag{4}$$

   Note that choosing $r = \infty$ gives us the ML decoder. For a smaller r, the solver can exploit the upper triangular nature of $\mathbf{R}$ to "prune" many candidate solutions (using depth-first-search or breadth-first-search or a combination of the two), thereby reducing the detection complexity significantly. One surprising property of the SD is that if it finds a valid solution, it is the same solution that the ML detector would have returned.