Best Compact SNR For MIMO

November 23, 2021

In this article we want to find the best compact achievable SNR for MIMO and its condition.

1 Definition Of Compact SNR For MIMO Channel

There are several definitions for the SNR of a MIMO channel and they are not necessarily equivalent. For example, we can define effective SNR to be

$$\mathrm{SNR}_{\text{eff}} = 2^{\sum_i \log(1+\mathrm{SNR}_i)} - 1 = \prod_i (1 + \mathrm{SNR}_i) - 1$$

where SNR_i is the SNR for the i-th (scalar) stream.

MATLAB 5G Toolbox™ uses another weird definition, which I would name it sum SNR, defined by

$$\mathrm{SNR}_{\text{MATLAB}} = \sum_i \mathrm{SNR}_i$$

And they use this SNR to compute the BLER-SNR curve. The physical interpretation of the sum SNR is not clear to us yet. Maybe MATLAB have their reasons.

But in the following we will define our so called compact SNR which makes good senses to us. Consider a MIMO channel

$$y = Hx + w$$

where y and w are vectors of the same dimension, x is of another dimension not necessarily the same with the former two, and the size of H is compatible with them. After linear estimation by an estimator G, we have

$$\hat{x} = Gy = GHx + Gw$$

The (equalized) compact SNR of the MIMO channel is defined by

$$\gamma_{\text{cp}} = \frac{\mathcal{E}||GHx||^2}{\mathcal{E}||Gw||^2} \tag{1}$$

So our goal is to find the G to maximize γ_{cp}.

We expand equation 1 into

$$\begin{aligned} \gamma_{\text{cp}} &= \frac{\mathcal{E}(x^*H^*G^*GHx)}{\mathcal{E}(w^*G^*Gw)} \\ &= \frac{\mathcal{E}\text{tr}(GHxx^*H^*G^*)}{\mathcal{E}\text{tr}(Gww^*G^*)} \\ &= \frac{\text{tr}(GH\mathcal{E}(xx^*)H^*G^*)}{\text{tr}(G\mathcal{E}(ww^*)G^*)} \\ &= \frac{\text{tr}(GHK_xH^*G^*)}{\text{tr}(GK_wG^*)} \end{aligned} \tag{2}$$

Let $C = GK_w^{1/2}$ and substitute it into 2 and we get

$$\gamma_{\text{cp}} = \frac{\text{tr}(CK_w^{-1/2}HK_xH^*K_w^{-1/2}C^*)}{\text{tr}(CC^*)} = \frac{\text{tr}(CAC^*)}{\text{tr}(CC^*)} \tag{3}$$

where we have defined $A = K_w^{-1/2}HK_xH^*K_w^{-1/2}$ for simplification.

Now the question becomes finding the C that maximizes 3. We will adopt two approaches and see that they lead to the same result.

2 Setting Derivatives To Zero

A brute force way to find the extreme value of a function is taking the derivatives and setting it to zero. But here things need to be treated carefully as we are dealing with matrices or even complex ones.

2.1 Derivatives Of Traces

In recent years with the rising of machine learning, more and more people on the internet start to talk about taking derivatives with respect to matrices. The most mentioned reference they use to support their calculation is *The Matrix Cookbook*. This book defines derivatives by the partial derivatives of each component. And many books in communication engineering or related subjects follow similar ways. Personally I do not like the way they define derivatives w.r.t. matrices. Because those definitions do not reveal the algebraic (or geometric) nature of derivatives.

From the algebraic point of view, derivatives have a simple, concrete, unified definition that can be applied to any kind of functions. Consider the general case. An operator F maps from vector spaces V to W

$$V \xrightarrow{F} W$$

At a point $x \in V$, F induces a linear map $f(x) \in \mathrm{Hom}\left(V, W\right)$

$$V \xrightarrow{f(x)} W$$

and is defined by

$$\lim_{\Delta x \to 0} \frac{F(x+\Delta x) - F(x) - f(x)[\Delta x]}{||\Delta x||} = 0 \tag{4}$$

where $f(x)[\Delta x]$ means applying the linear map $f(x)$ on Δx and $|| \cdot ||$ is an arbitrary norm. It is easy to see that under this definition

$$\nabla_x \left(x^T A x\right) = 2x^T A \tag{5}$$

where x could be a scalar or vector or even matrix.

Go back to functions of matrices. Denote the derivatives of $F(X)$ with $f(X)$, then we study the derivatives of traces.

First let us consider a simple case. Find $\nabla_X \operatorname{tr}(F(X))$.

It follows that

$$\begin{aligned} \operatorname{tr}(F(X+\Delta X)) &= \operatorname{tr}(F(X) + f(X)[\Delta X]) + o(\Delta X) \\ &= \operatorname{tr}(F(X)) + \operatorname{tr}(f(X)[\Delta X]) + o(\Delta X) \end{aligned}$$

Hence $\nabla_X \operatorname{tr}(F(X))$ is given by

$$\nabla_X \operatorname{tr}(F(X))[Y] = \operatorname{tr}(f(X)[Y]) \tag{6}$$

for any Y in the domain. Unlike equation 5, here we cannot write the derivatives as a matrix under our definition 4, it is still a linear map though. In books including *The Matrix Cookbook* the derivatives can be represented by matrices under their definitions. But they do not fit well in the sense of algebra.

Next we will show an result that is relevant to the optimization problem 3.

Claim 1. *Let f and g be the respective derivatives of F and G that take matrices as input. Then for any Y in the domain of $\frac{\mathrm{tr}(F(X))}{\mathrm{tr}(G(X))}$, we have*

$$\nabla_X \left\{ \frac{\mathrm{tr}(F(X))}{\mathrm{tr}(G(X))} \right\} [Y] = \mathrm{tr} \left\{ \frac{\mathrm{tr}(G(X)) f(X)[Y] - \mathrm{tr}(F(X)) g(X)[Y]}{\mathrm{tr}(G(X))^2} \right\} \tag{7}$$

Proof. Substitute

$$\mathrm{tr}(F(X + \Delta X)) = \mathrm{tr}(F(X)) + \mathrm{tr}(f(X)[\Delta X]) + o(\Delta X)$$

and

$$\mathrm{tr}(G(X + \Delta X)) = \mathrm{tr}(G(X)) + \mathrm{tr}(g(X)[\Delta X]) + o(\Delta X)$$

into

$$\frac{\mathrm{tr}(F(X + \Delta X))}{\mathrm{tr}(G(X + \Delta X))} - \frac{\mathrm{tr}(F(X))}{\mathrm{tr}(G(X))}$$

and the rest are just tedious calculations. □

2.2 Optimal Solutions

We find the solution to 3 by setting the derivatives to zero. Note that equating 7 to zero is equivalent to setting

$$\mathrm{tr}(G(X)) f(X) = \mathrm{tr}(F(X)) g(X) \tag{8}$$

To apply equation 8 on 3, we can form the correspondence: $X = C^*$, $F(X) = X^* A X$, $G(X) = X^* X$.

Special care should be taken here because we are playing with complex matrices. So we have to check the differentiability in the first place. Unfortunately, usually the functions involve conjugation are not complex differentiable.

We restrict our problem to be on real matrices. By equation 5, we have

$$f(X) = 2X^T A$$

and

$$g(X) = 2X^T$$

Substitute them into equation 8 and after some calculation we get

$$AX = \frac{\operatorname{tr} F(X)}{\operatorname{tr} G(X)} X$$

Look back at equation 3, this is exactly

$$AX = \gamma_{\text{cp}} X \tag{9}$$

It becomes interesting. We see that the columns of X should be the eigenvectors (including zero vectors) of A and all belong to a same eigenvalue. What's more, the maximum achievable SNR is just the maximal eigenvalue of A. From 3, the positive semi-definiteness of A ensures that all eigenvalues are non-negative.

Note also that for any diagonal matrices (could be complex) D, if X is a solution to 9, then Xe^{jD} is also a solution. Indeed, multiplying each column of X by a complex number does not affect the vectors being eigen.

3 Spectral Decomposition Theorem

We are going to solve the optimization problem in a similar way as the proof for Rayleigh-Ritz Theorem (Not Rayleigh-Ritz Method appeared on Wikipedia). which relies on Spectral Decomposition Theorem. We state it without proof

Theorem 1 (Spectral Decomposition Theorem)**.** *If real matrix A is symmetric (or Hermitian if complex), then there exists an orthogonal (or unitary if complex) matrix whose column vectors are all eigenvectors of A.*

Since A is positive semi-definite (actually Hermitian would suffice), by spectral decomposition theorem, the eigenvalue decomposition and SVD coincide

$$A = U\Lambda U^*$$

Let $S = UC^*$, then equation 3 becomes

$$\gamma\text{cp} = \frac{\text{tr}(S^*\Lambda S)}{\text{tr}(S^*S)}$$

Let s_i denotes the i-th column vector of S and λ_i denotes the eigenvalue of Λ corresponding to s_i (be careful, λ_i is not necessarily the i-th diagonal entry of Λ), then we have

$$\gamma\text{cp} = \frac{\sum_i \lambda_i s_i^* s_i}{\sum_i s_i^* s_i}$$

which is the weighted average of $\{\lambda_i\}$ with the weights $\{s_i^* s_i\}$.

Apparently,

$$\gamma\text{cp} \le \lambda\text{max}$$

with equality if $\lambda_i = \lambda\text{max}$ for all i's. In other word, all s_i's have the same eigenvalue which is the maximum one.

Therefore we see that the results of the two approaches are identical.

To summarize. The procedure to construct a linear decoder that maximizes compact SNR for MIMO is:

1. Apply eigenvalue decomposition (or SVD) on ${K_w^{-1/2}}HK_xH^*{K_w^{-1/2}}$ and find the maximum eigenvalue with the corresponding eigenvectors;

2. Arrange the eigenvectors include zero vector in an arbitrary way to form a matrix Q (dimension compatible with the MIMO channel);

3. The decoding matrix is then given by

$$G = Q^* {K_w^{-1/2}}$$

The resulting G is definitely not unique since we can scale the eigenvectors and arrange them at will if there are no further restrictions.