1.1) Show that $\max: \mathbb{R}^n \to \mathbb{R}$ is convex.

Wikipedia definition of convex function:

for $0 \le t \le 1$, $f: X \to \mathbb{R}$, $\forall\, x_1, x_2 \in X$

$$f(tx_1 + (1-t)x_2) \le tf(x_1) + (1-t)f(x_2)$$

We'll use x, y instead of x_1 & x_2 to avoid multiple subscripts

Show: $\max(tx + (1-t)y) \le t\max(x) + (1-t)\max(y)$

$\max(x) = x_i$ for some $i \to t\max(x) = tx_i$

$\max(y) = y_j$ for some $j \to (1-t)\max(y) = (1-t)y_j$

So,

$$\max(tx + (1-t)y) \le tx_i + (1-t)y_j$$

if $y_j \ne y_i$ & $x_i \ne x_j$, it is clear that the LHS will always be less than the right. In other words, if the max values of x & y occur at the same index, LHS = RHS at this index — otherwise LHS < RHS.

If $y_j \ne y_i$ & $x_i \ne x_j$

$$\max(tx_k + (1-t)y_k) < tx_i + (1-t)y_j \quad \forall\, k$$

1.2)

2. [Writing] An equity analyst at a hedgefund is developing a trading strategy on a stock S with price S_t at time t.

He has a signal α_t as well, which we uses to determine how much to long/short in the next period. In other words, his holdings at time t are given by $f(\alpha_t, S_t)$, for some "trading strategy" f.

The strategy's PnL is then given by

holdings change in price

$$\mathrm{PnL}(f) = \sum_{t=0}^{N} f(\alpha_t, S_t)(S_{t+1} - S_t)$$

where $\{\alpha_t\}_{t=0\ldots N}$ and $\{S_t\}_{t=0\ldots N}$ are the historical signal and stock price series and we assume interest rates are zero.

The analyst backtests $n > 1$ trading strategies $\{f_i : i = 1\ldots n\}$, and picks best performing one f_*. He tells his boss that his strategy can make $\mathrm{PnL}(f_*)$.

Show using part $\boxed{1}$ that *on average*, this is an overestimate.

$$E[PnL(f)] = E\left[\sum_{t=0}^{N} f(\alpha_t, S_t)(S_{t+1} - S_t)\right]$$

$$= \sum_{t=0}^{N} E\left[f(\alpha_t, S_t)(S_{t+1} - S_t)\right]$$

using the assumption that S is based on BM w/ independent increments $\sim N(0, dt)$

$$= \sum_{t=0}^{N} E[f(\alpha_t, S_t)]\cancel{E[S_{t+1} - S_t]}^{\,0} = 0$$

From part 1, we showed that the max function is convex. For convex functions, we can apply Jensen's inequality:

$$E[\max_i PnL(f_i)] \geq \max_i \cancel{E[PnL(f_i)]}^{\,0}$$

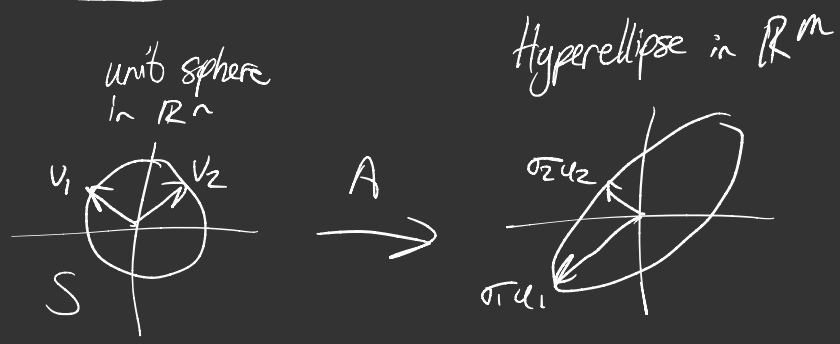
$$E[PnL(f_*)] \geq 0$$

The expectation of the PnL of the chosen strategy is non-negative, while the expectation of each individual strategy is 0. Thus, we are picking from the positive tail of the distribution ~~based~~ on in sample performance. This is an overestimate of out of sample performance.

2.1) Notes from the reading:

SVD

- Idea:



$m \times n$

$$A = U \Sigma V^{-1}$$

V^{-1} ← unitary $n \times n$ matrix
"preserves" sphere

Σ ← $m \times n$ diagonal matrix
· stretches the sphere into an ellipse aligned w/ canonical basis

U ← unitary $m \times m$ matrix
· rotates or reflects hyperellipse

This full form can have silent columns of U & rows of Σ w/ 0 values of σ. Can alternatively use reduced form

Every matrix has an SVD & the singular values (σ_j) are uniquely determined. If A is square & σ_j's distinct, then u_j & v_j are uniquely det. up to complex signs.

Least Squares:

- over determined system — $m > n$ equations & n unknowns
- no true solution, choose b that minimizes $\|Ax-b\|_2$

Polynomial Interpolation:

$p(x) = c_0 + c_1 x + \dots + c_{m-1} x^{m-1}$ $\qquad p(x_i) = y_i$

Vandermonde system:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{m-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{m-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Solution guaranteed to be non-singular if $\{x_i\}$ are distinct

- Fit can be quite bad & tends to get worse w/ more data points.
- Sensitive to perturbations in the data.

Polynomial Least Squares fitting:

$p(x) = c_0 + c_1 x + \dots + c_{n-1} x^{n-1}$ for some $n < m$

$n-1$ degree polynomial

minimize $\sum_{i=1}^{m} |p(x_i) - y_i|^2 = \|r\|_2^2$ for V-system:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{m-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_m & x_m^2 & \cdots & x_m^{m-1} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

2.4) [Writing] Prove that β as given in equation (2) satisfies equation (1) and that $\hat{y}$ as given in equation (3) equals $X\beta$. Do not assume in your proof that either U or V are square matrices or that X has full rank.

Explain (with math and formulas) the meaning of the `ComputeThinU`, `ComputeThinV` flags.

Part 1) Prove $X^T X \beta = X^T y$ for $\beta = D^{-1} V \Sigma^{-1} U^T y$

$$\beta = D^{-1} V \Sigma^{-1} U^T y$$

$$X^T X \beta = D^T \hat{X}^T \hat{X} \underbrace{D D^{-1}}_{I} \underbrace{V \Sigma^{-1} U^T}_{\hat{X}^{-1}} y$$

$$= \underbrace{D^T \hat{X}^T}_{X^T} \underbrace{\hat{X} \hat{X}^{-1}}_{I} y$$

$$X^T X \beta = X^T y \checkmark$$

$$X = \hat{X} D$$

$$X^T X = D^T \hat{X}^T \hat{X} D$$

$$\hat{X} = U \Sigma V^T$$

$$\hookrightarrow \hat{X}^{-1} = V \Sigma^{-1} U^T$$

Part 2) Prove $\hat{y} = X\beta$ for $\hat{y} = U U^T y$

$$\hat{y} = X\beta$$

$$= X D^{-1} V \Sigma^{-1} U^T y$$

$$= \hat{X} \underbrace{D D^{-1}}_{I} V \Sigma^{-1} U^T y$$

$$= U \Sigma \underbrace{V^T V}_{I} \Sigma^{-1} U^T y$$ V is orthogonal, so $V^T V = I$

$$\hat{y} = U \underbrace{\Sigma \Sigma^{-1}}_{I} U^T y = U U^T y \checkmark$$

Part 3) ComputeThinU & ComputeThinV tell Eigen to use the reduced form of SVD rather than the full form.

<u>Reduced SVD:</u> (if $m \geq n$)

$$\left[A_{m\times n}\right] = \left[\hat{U}_{m\times n}\right]\left[\hat{\Sigma}_{n\times n}\right]_{diag}\left[V^{-1}_{n\times n}\right]$$

<u>Full SVD:</u> (if $m \geq n$)

the 0's in Σ mean these columns don't impact the result matrix.

$$\left[A_{m\times n}\right] = \left[\hat{U}_{m\times n}\ \underbrace{\begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \\ \hline & 0 & \end{bmatrix}}_{U_{m\times m}}\right]\underbrace{\begin{bmatrix} \hat{\Sigma}_{n\times n}\ {}_{diag} \\ \hline 0 \end{bmatrix}}_{\Sigma_{m\times n}}\left[V^{-1}_{n\times n}\right]$$

$$= \hat{U}_{m\times m}\,\hat{\Sigma}_{m\times n}\,V^{-1}_{n\times n}$$

So we get an equivalent result, but better computational efficiency using these flags since we avoid unnecessary computations when A isn't full rank.