FINAL project: linear interpolation –vs- linear regression

Play by the rules:

1. **The deadline for submission to AIS submission area is 18th December 2023 23:59. Do NOT submit by email.**
2. **The use of any form of cycles or symbolic toolbox functions in you implementation is prohibited. Using the vector features of MATLAB is mandatory.**
3. **It is prohibited to use any toolbox function when implementing `linterpCoef()`. It is mandatory to implement it using the equations derived in the lecture.**
4. **It is prohibited to use the function `polyfit()` or the \ operator when implementing the `linreg()` function. It is mandatory to implement it using the equations derived in the lecture, potentially using the built-in functions `mean()`, `var()`, `cov()`.**

The points obtained for the exercises are distributed in the following way (together max. 40 points):

Exercise 1: 0 points.
Exercise 2: 8 points.
Exercises 3 - 10: 4 points each.
Exercise 11: 0 points, but Ex. 5 – 10 must work with step value set by a variable (not a literal).

Exercise 1: Preparation

Read the documentation for the following MATLAB functions: `rand, randn, randi, mean, var, cov, polyfit`. Read the short help provides by the `>>help function` command, and also the more detailed help provided by the `>>doc function` command. Except where explicitly stated otherwise, implement all code into a single .m file using cell mode (use the cell separator:`%%`). For further examples look at the MATLAB linear regression documentation online (in english): https://www.mathworks.com/help/matlab/data_analysis/linear-regression.html
Further documentation in slovak available here: https://kurzy.kpi.fei.tuke.sk/nm/student/13.html

Implement exercise 2 into `linterpCoef.m` and `linterp.m`, use `lin_interpolation.m` as a unit test, do not modify this file. Implement exercises 3,4,5,7,8,10 directly into `lin_regression.m`.

Exercise 2: Linear interpolation

Into a separate .m file implement a function `linterpCoef()` that takes two points *p1* and *p2* in 2D plane as input parameters and returns the coefficients *a,b* of a linear function $f(x) = a.x + b$ that crosses these points. The points are defined as vectors of two values: $p_i = [x_i, y_i]$. Into another .m file implement a second function with a prototype given: `[X,Y] = linterp(a,b,Xl,n)` that calculates the values of the linear interpolation function $f(x) = a.x + b$ for an interval of x values given by $X_l = [x_{min}, x_{max}]$ where `n` >= 1 is the number of intervals into which the interval X_l will be divided. The function should return two vectors: the interpolated coordinates x and corresponding values of $f(x)$ at these points. The length of the resulting vectors should be n+1 and the x values should include the boundary values x_{min} and x_{max}.

Exercise 3: Generate pseudo-random matrices

Generate a matrix $\mathbf{R_u}$ of pseudo-random floating point values from a uniform distribution, a matrix $\mathbf{R_i}$ of pseudo- random integer values from a uniform distribution and a matrix $\mathbf{R_n}$ of pseudo-random values from a normal distribution. Let the size of all matrices be M x N. Let the expected value of $\mathbf{R_n}$ be $m = 8$ and variance $v = 3$. Let the values of $\mathbf{R_u}$ an $\mathbf{R_i}$ be from the interval $< a, b > = < 4, 9 >$.
Calculate the sample mean and sample variance of all pseudo-random matrices:

a) for each column (The results will be row vectors of sample averages and variances.)
b) for the whole matrix (The results will be two scalar numbers per each matrix: sample mean and sample variance.)

Note: Yout don't have to implement the formulas defined in the section below, just use the MATLAB built-in functions.

Exercise 4: Calculate basic statistics

Generate a pseudo random column vector V of varying length N (using the already prepared `for` cycle containing also a print statement) and mean and variance given: $m = 10$, $v = 3$. For each vector length N calculate sample mean m_s and variance v_s and for each calculated m_s and v_s calculate their relative errors in percent (That means error with respect to the given values $m = 10$, $v = 3$). Also print these values. The errors should converge to zero with large N.

Exercise 5: Generate noisy data

Generate values of a linear function $f(x) = ax + b$ for the x variable taking on values from the interval <-4,10> with step 0.5. Create two noisy copies (simulating the real-world measurement): Y_1, and Y_2 by adding pseudo-random noise from the normal distribution: one with a variance $v_1 = 0.1$, the other with $v_2 = 2$. The mean value of both noise vectors should be set to zero. Plot three plots side-by-side within one figure: first the noiseless linear function $f(x)$ using `plot()`, second the scattered noisy observations Y_1 using `scatter()`, and finally the scattered observations of the more noise-degraded signal Y_2 (again using `scatter()`). Use `subplot()` for plot separation.

Note: Do not implement the linear function $f(x)$ to a separate .m-file. Just use the vector x in a simple formula.

This exercise has nothing to do with the symbolic toolbox.

Exercise 6: Implement linear regression coefficient calculation algorithm

Open the provided .m file `linreg.m` with a function prototype given as :

`function [alpha, beta] = linreg( x, y )`. Implement the body of this function to perform the calculation of simple linear regression coefficients α, β. The necessary formulas are provided below in the "Equations" section. Use, however, the MATLAB built-in functions `mean(), var(), cov()` in your implementation.

Note that when using linear interpolation, we denoted the coefficients of the linear function: $f(x) = ax + b$. Then, when describing regression, we denoted the coefficients: $f(x) = \alpha + \beta x$. Furthermore, various MATLAB functions may use another notation, such as: $f(x) = p_1x + p_2$. It is our duty to not be confused by this fact of life :)

Exercise 7: Perform linear regression – find the actual coefficients

For the x coordinates and both noisy datasets Y_1, Y_2 from Exercise 5 find the coefficients α, β of the simple linear regression:

a) using your own function `linreg()`
b) using the MATLAB function `polyfit()`
c) using the \ operator

Observe the results.

Note: The result for each of the points a) b) c) will be the tuple (α, β) of the linear regression coefficients. Together 6 values. The results obtained in a) may be very slightly different from results of b) and c). This is a known property of floating point calculations and number representation. To compare the equality of these results, the == operator is not an appropriate tool.

Exercise 8: Perform linear regression - plot fitted data

Use the `polyval()` function to find the data Y_{f1} and Y_{f2} for both noisy datasets Y_1 and Y_2 using the previously found linear regression coefficients α, β for the x vector defined in Exercise 5.

Use the `plot(),subplot()` and `scatter()` functions to plot three plots side by side in one figure window:

1. The noisy data Y_1 along with the fitted linear function values Y_{f1}.
2. The noisy data Y_2 along with the fitted linear function values Y_{f2}.
3. The linear fitted values (vectors) Y_{f1} and Y_{f2}.

Exercise 9: Calculate coefficient of determination

Implement the body of the function `r_squared()`, provided in a separete .m-file `r_squared.m`, so that it first calculates the residuals as defined by eq. (7) and uses these to perform the calculation of the coefficient of determination R^2 as defined by eq. (6).

Note: The symbol Y_f is used here in the text to denote the fitted data while in MATLAB source code the symbol `yf` is used (potentially further indexed as: `yf1, yf2`). Equation (7) uses the symbol $\hat{y}$ (y hat).

Exercise 10: Plot the residuals

For the fits Y_{f1} and Y_{f2}, and know values Y_1 and Y_2, calculate residuals for r_1 and r_2 and plot them using the `stem()` (for r_1) and `scatter()` (for r_2) functions. Which of these functions appears to be more suitable for this data visualization ?

Also use the `hist()` function to calculate and plot the histogram of one the residuals vectors of your choice.

Note: The vectors of residuals r_1 and r_2 are vectors of the same length as vectors Y_1 and Y_2 (which is the same length as Y_{f1} and Y_{f2}).

Exercise 11: Change the resolution

Change the value of the x step in exercise 4 from 0.5 to 0.1 and run the exercises 4 to 9 again. Also try step 0.001. Observe results.

FINAL projekt: lineárna interpolácia a regresia

Pravidlá:

1. **Deadline na odovzdanie FINAL projektu do miesta odovzdania v AIS je 18.12. 23:59. NEodovzdávajte mailom.**
2. **Pri implementácií úloh je zakázané používať akékoľvek cykly, alebo funkcie symbolického toolboxu. Využitie vektorových operácií systému MATLAB je povinné.**
3. **Pri implementácií funkcie `linterpCoef()` je zakázané používať akékoľvek funkcie toolboxov, implementovať je ju nutné pomocou vzorcov odvodených na prednáške.**
4. **Pri implementácií funkcie `linreg()` je zakázané používať funkciu `polyfit()` a operátor \, implementovať je ju nutné pomocou vzorcov odvodených na prednáške s možným využitím funkcií `mean()`, `var()`, `cov()`.**

Za úlohy je možné získať max. 40 bodov nasledovne:

- **Cvičenie 1: 0 bodov**
- **Cvičenie 2: 8 bodov**
- **Cvičenie 3 – 10 : 4 body každé**
- **Cvičenie 11: 0 bodov, ale cvičenia 5 – 10 musia používať krok nastavený ako premennú (nie literál).**

Cvičenie 1: Príprava

Prečítajte si dokumentáciu k nasledujúcim funkciám MATLABu: `rand`, `randn`, `randi`, `mean`, `var`, `cov`, `polyfit`. Prečítajte si nie len krátku verziu dokumentácie poskytovanú pomocou príkazu: `>>help` *`function`*, ale aj detailnejšiu dokumentáciu, ktorú zobrazuje príkaz: `>>doc` *`function`* . Okrem úloh, kde je explicitne uvedený iný postup, implementujte všetky úlohy do jedného spoločného .m súboru, pričom použite bunkový režim (oddelenie pomocou: `%%`) pre oddelenie implementácie jednotlivých úloh. Prečítajte si podrobnú dokumentáciu k implementácii lineárnej regresie v angličtine: https://www.mathworks.com/help/matlab/data_analysis/linear-regression.html.
Dokumentácia v slovenčine je dostupná tu: https://kurzy.kpi.fei.tuke.sk/nm/student/13.html

Implementujte cvičenie 2 so súborov `linterpCoef.m` and `linterp.m`, použite `lin_interpolation.m` ako unit test - pre kontrolu správnosti (tento súbor nemodifikujte). Implementujte cvičenia 3,4,5,7,8,10 priamo do súboru `lin_regression.m`.

Cvičenie 2: Lineárna interpolácia

Do samostatného .m súboru implementujte funkciu `linterpCoef()`, ktorá má vstupné parametre dva body v 2D rovine: *p1* a *p2*, a ktorej výstupné parametre sú koeficienty *a*, *b* lineárnej funkcie $f(x) = a.x + b$, ktorá prechádza týmito bodmi. Body sú definované ako vektory p_i = [x_i, y_i]. Do ďalšieho .m súboru implementujte druhú funkciu s prototypom `[X,Y] = linterp( a, b, Xl, n )`, ktorá vypočíta hodnoty lineárnej interpolácie pre zadaný interval x hodnôt $X_l = [x_{min}, x_{max}]$ kde n >=1 je počet podintervalov rovnakej dĺžky, na ktoré sa má rozdeliť interval X_l. Funkcia má vracať dve hodnoty: vektor interpolovaných x-súradníc a vektor korešpondujúcich hodnôt funkcie *f(x)*. Dĺžka oboch týchto vektorov má byť n+1 a krajné hodnoty výstupného vektora X majú byť x_{min} a x_{max} .

Cvičenie 3: Generovanie pseudo-náhodných matíc

Vygenerujte maticu $\mathbf{R_u}$ pseudonáhodných hodnôt (desatinných čísel typu double) s rovnomerným rozdelením pravdepodobnosti. Ďalej vygenerujte maticu pseudonáhodných *celých čísel* $\mathbf{R_i}$ s rovnomerným rozdelením pravdepodobnosti a maticu pseudonáhodných desatinných čísel $\mathbf{R_n}$ s normálnym rozdelením. Nech rozmery všetkých matíc M x N sú určené preddefinovanými premennými M a N. Očakávaná stredná hodnota pre maticu $\mathbf{R_n}$ nech je daná $m = 8$ a variancia $v = 3$. Zabezpečte, aby všetky prvky vygenerovaných matíc $\mathbf{R_u}$ a $\mathbf{R_i}$ ležali v intervale: $< a, b > = < 4, 9 >$.
Vypočítajte výberový priemer a varianciu pre všetky vygenerované pseudo-náhodné matice:

a) pre každý stĺpec matice zvlášť (Výsledkom bude riadkové vektory výberových priemerov a variancií.)
b) naraz pre celú maticu (Výsledkom budú dve skalárne čísla pre každú maticu: výberový priemer a variancia.)

Poznámka: Vzorce uvedené ďalej v samostatnej sekcii nemusíte implementovať, stačí zavolať už existujúce funkcie MATLABu.

Cvičenie 4: Výpočítajte základné štatistiky

Vygenerujte pseudonáhodný stĺpcový vektor *V*, ktorého dĺžku danú premennou *N* budete meniť v cykle (v programe je už predpripravený cyklus `for` aj s príkazom pre výpis). Nech očakávané hodnoty strednej hodnoty a variancie sú konštantné $m = 10$, $v = 3$. Pre každú dĺžku *N* vektora vypočítajte výberový priemer m_s a výberovú varianciu v_s a pre každú hodnotu m_s a v_s vypočítajte ich relatívnu chybu v percentách (Teda chybu voči očakávaným hodnotám $m = 10$ a $v = 3$). Vypíšte tieto hodnoty. Hodnoty chýb by mali klesať spolu so zvyšujúcou sa hodnotou *N*.

Cvičenie 5: Vygenerujte zašumené dáta

Vygenerujte hodnoty lineárnej funkcie $f(x) = ax + b$ pre hodnoty vektora x z intervalu <-4,10> s krokom 0.5. Vytvorte dva zašumené vektory Y_1 a Y_2 (simulujúce merania skutočných veličín) tým, že k hodnotám lineárnej funkcie $f(x)$ pripočítate pseudonáhodný šum s normálnym rozdelením pravdepodobnosti: jeden s varianciou $v_1 = 0.1$, druhý s varianciou $v_2 = 2$. Stredná hodnota oboch šumových vektorov nech je nastavená na nulu. Vykreslite všetky tri priebehy do jedného okna vedľa seba: najprv čistú lineárnu funkciu $f(x)$ pomocou funkcie `plot()`, potom mierne zašumený priebeh pozorovaní Y_1 pomocou funkcie `scatter()`, a nakoniec veľmi zašumený priebeh Y_2 (opäť pomocou `scatter()`). Použite `subplot()` pre oddelenie grafov.
Poznámka: Pre funkciu $f(x)$ nevytvárajte samostatný .m-súbor. Len použite dosadenie vektora x do jednoduchého vzorca.
Toto cvičenie nemá nič spoločné s použitím symbolického toolboxu.

Cvičenie 6: Implementujte algoritmus výpočtu koeficientov lineárne regresie

Otvorte si priložený .m-súbor `linreg.m` obsahujúci funkčný prototyp daný ako:
`function [alpha, beta] = linreg( x, y )`. Implementujte telo tejto fukcie tak aby táto vypočítavala koeficienty jednoduchej lineárnej regresie $f(x) = \alpha + \beta x$. Potrebné vzorce sú uvedené dole v sekcii "Rovnice". Tieto vzorce ale nemusíte sami implementovať, vo Vašej implementácií použite hotové funkcie MATLABu: `mean()`, `var()`, `cov()`.
Všimnime si kontrast pri označení: pri lineárnej interpolácií sme si označili koeficienty lineárnej funkcie $f(x) = ax + b$, zatiaľ čo pri regresii používame označenie $f(x) = \alpha + \beta x$,pričom rôzne funkcie MATLABu používajú ešte aj iné označenia (napr. $f(x) = p_1x + p_2$). Je našou úlohou si to nepopliesť :)

Cvičenie 7: Vykonajte lineárnu regresiu – nájdite koeficienty

Pre hodnoty x a obidve zašumené pozorovania Y_1, Y_2 z Cvičenia 5 nájdite koeficienty jednoduchej lineárnej regresie α, β:

a) pomocou vlastnej funkcie `linreg()`
b) pomocou funkcie MATLABu `polyfit()`
c) pomocou operátora \

Pozorujte výsledky.
Poznámka: Pre každý z horeuvedených bodov a) b) c) bude výsledkom dvojica koeficientov (α, β) lineárnej funkcie. Spolu teda 6 hodnôt. Výsledky dosiahnuté v bode a) sa môžu nepatrne líšiť od výsledkov z bodu b) a c). Toto je známa vlastnosť výpočtov a reprezentácie čísel s pohyblivou desatinnou čiarkou. Pre porovnanie týchto výsledkov nie je operátor == vhodným nástrojom.

Cvičenie 8: Vykonajte lineárnu regresiu - vykreslite lineárne priebehy

Použite funkciu `polyval()` pre nájdenie hodnôt Y_{f1} and Y_{f2}, získaných dosadením do lineárnych funkcií s koeficientami α, β nájdených pomocou jednoduchej lineárnej regresie pre všetky hodnoty vektora x, ako je definovaný v Cvičení č. 5.
Použite funkcie `plot()`, `subplot()` a `scatter()` na vykreslenie troch grafov v jednom okne vedľa seba:

1. Zašumené hodnoty Y_1 zároveň s preloženými hodnotami regresiou získanej lineárnej funkcie Y_{f1}.
2. Zašumené hodnoty Y_2 zároveň s preloženými hodnotami regresiou získanej lineárnej funkcie Y_{f2}.
3. Hodnoty regresiou získaných lineárnych funkcií (vektorov) Y_{f1} a Y_{f2}.

Cvičenie 9: Vypočítajte koeficient determinácie

Implementujte telo funckie `r_squared()`, dodanej v samostatnom m.-súbore `r_squared.m`, tak aby táto vypočítala najpr reziduá a tieto potom použila na výpočet koeficientu determinácie R^2 podľa vzorca (6).
Poznámka: Symbol Y_f je v texte používaný pre linearizované dáta, zatiaľ čo v zdrojovom MATLAB súbore je použitý symbol `yf` (prípadne ďalej indexovaný ako: `yf1`, `yf2`). V rovnici (7) je pre túto istú veličinu použitý symbol $\hat{y}$ (y so strieškou).

Cvičenie 10: Vykreslite reziduá

Pre regresiou získané hodnoty Y_{f1} and Y_{f2} a známe hodnoty Y_1 and Y_2 vypočítajte reziduá r_1 and r_2 a tieto vykreslite pomocou funkcií `stem()` (pre r_1) a `scatter()` (pre r_2). Ktorá z týchto funkcií sa Vám javí ako vhodnejšia pre vizualizáciu týchto dát ?
Použite funkciu `hist()` pre výpočet a vykreslenie histogramu Vami zvoleného vektora reziduí.
Poznámka: Vektory reziduí r_1 a r_2 sú vektory rovnakej dĺžky ako vektory Y_1 a Y_2 (ktoré sú rovnakej dĺžky ako Y_{f1} a Y_{f2}).

Cvičenie 11: Zmeňte rozlíšenie

Zmeňte hodnotu kroku pri vektore x v cvičení č. 4 z 0.5 na 0.1 a spustite kód cvičení 4 až 9 znova. Skúste aj krok 0.001. Pozorujte výsledky.

Equations / Rovnice

Basic statistics / Základné štatistiky

Sample mean, výberový priemer

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

Sample variance, výberová variancia

$$s_x{}^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2 \tag{2}$$

Sample covariance, výberová kovariancia

$$s_{x,y} = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right) \tag{3}$$

Calculating simple linear regression coefficients
Výpočet koeficientov jednoduchej lineárnej regresie

$$\beta = \frac{s_{x,y}}{s_x{}^2} \tag{4}$$

$$\alpha = \bar{y} - \beta\bar{x} \tag{5}$$

Calculating residuals and coefficient of determination
Výpočet reziduí a koeficientu determinácie

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2} \tag{6}$$

$$r_i = y_i - \hat{y}_i \tag{7}$$