



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

*Instrukce k programovací úloze
lineární regrese*

Patrik Drhlík

Liberec 2015

Úvod

V poslední sekci kurzu Lineární regrese (VI. Programovací cvičení a test) se nachází odkaz s názvem *Programování lineární regrese*. V tomto dokumentu je popsáno prostředí, v kterém se programuje. Dále je zde uveden návod, podle kterého byste po absolvování všech předchozích přednášek měli naprogramovat zadané úlohy. Programování bude probíhat v jazyce Python. Pro ty, kteří s tímto jazykem nikdy nepřišli do styku, budou k dispozici materiály k pochopení základních principů jazyka.

Není potřeba instalace žádného softwaru do vlastního počítače – všechno programování lze jednoduše absolvovat přímo v prohlížeči v prostředí VPL.

2 Popis prostředí VPL na e-learningovém portále ALS

V poslední šesté sekci se nachází ikona počítače s lidskou hlavou, která značí prostředí VPL (Virtual Programming Lab). Jakmile do této laboratoře vstoupíte, tak uvidíte základní popis celé úlohy. Ten je krátký a pouze se odkazuje na tento dokument. Mimo to zde uvidíte předpřipravené soubory, které budete potřebovat ke splnění úkolů.

Další je záložka *Submission*, kterou nebudete potřebovat (odesílají se zde soubory). Poslední záložka *Submission View* slouží pro přehled odevzdaných úkolů. Nejvíce Vás ale bude zajímat záložka *Edit*. V té už uvidíte zdrojové kódy jednotlivých souborů. Každý soubor je reprezentován vnitřní záložkou (tabem).

K tomuto programovacímu prostředí je k dispozici několik funkcí (v liště nad oknem). Není potřeba nic nastavovat, případně si pro lepší přehled můžete okno přepnout do režimu fullscreen. Zajímat Vás ale bude potřeba každý soubor po úpravě před spuštěním uložit (*Save*). Pokud chcete program jen spustit bez odeslání k hodnocení, tak k tomu slouží možnost *Run* a k odeslání všech úloh k ohodnocení slouží možnost *Evaluate*.

3 Programování v jazyce Python

Nezáleží na verzi tutoriálů (Python 2 i 3). Oproti ostatním je Python na první pohled jiný v tom, že se na konci řádků nepiší středníky a těla funkcí, cyklů nebo podmínek se na novém řádku odsazují. Zbytek záležitostí je vesměs podobných ostatním klasickým programovacím jazykům. Více informací naleznete na následujících odkazech.

Trochu specifitější pro Vás může být práce s knihovnou numpy. Pokud jste ale někdy dělali v MATLABu nebo alespoň základní kód v MATLABu dokážete přečíst, tak Vám velmi pomůže třetí z uvedených odkazů, v kterém jsou popsány ekvivalentní příklady. Pomůže Vám to samozřejmě i bez znalosti MATLABu. Součástí je i popis funkcí.

- České tutoriály na <http://python.cz/>
- Anglické tutoriály na oficiální dokumentaci <https://docs.python.org/2/tutorial/>

- Knihovna numpy, srovnání funkcí s MATLABem - http://wiki.scipy.org/NumPy_for_Matlab_Users

4 Způsob odesílání a hodnocení úloh

Odesílání i hodnocení probíhá automaticky v prostředí VPL. V záložce Test Activity -> Edit - Evaluate je možné program odeslat k hodnocení. Výsledky otestuje server a okamžitě Vám vrátí výsledek. Odeslání k hodnocení vždy zhodnotí všechny úlohy najednou. Pokud úlohy nechcete rovnou hodnotit, ale pouze zkontrolovat, jestli v nich není chyba nebo jestli vrací rozumný výsledek, tak stačí spustit *Run* a v konzoli zadat a potvrdit příslušné číslo úlohy (č. 1–3 viz 6).

Počet bodů za každou úlohu bude popsán až v kapitole 6.

5 Popis souborů

Upravujte pouze soubory označené s hvězdičkou. Ostatní soubory slouží k nastavení a jejich změnou byste mohli znefunkčnit chod a hodnocení programu. Jsou tu pro Vás, abyste nemuseli řešit zbytečné věci a zabývali se pouze jádrem algoritmů.

- `main.py` - hlavní soubor s nastavením a voláním Vámi implementovaných funkcí.
- * `hypothesis.py` - soubor s první úlohou, v které je úkolem naprogramování regresní funkce.
- * `computeCost.py` - soubor s druhou úlohou, v které je úkolem naprogramování hodnotové funkce.
- * `computeGradient.py` - soubor s třetí úlohou, v které je úkolem naprogramování metody postupného klesání.
- `data.txt` - datový soubor, z kterého se načítají proměnné X a y .
- `setup.py` - soubor s doplňkovým nastavením, který načítá proměnné z datového souboru, které máte dále k dispozici.

Poslední dva typy komentářů jsou *INIT* a *ASSIGNMENT*. Po prvním zmíněném jsou inicializované hodnoty, s kterými budete v programu pracovat. Pro odevzdání úloh tyto hodnoty neměňte, jinak Vám úkol nebude uznán. Po druhém bloku komentáře už je Váš prostor, kam máte za úkol naprogramovat příslušnou úlohu. Nemusíte řešit vracení hodnot ani jejich zaokrouhlování, protože to už je za Vás vyřešené.

6 Popis úloh

Každá z úloh má číslo, pod kterým si můžete vyzkoušet spuštění pouze jedné funkce bez odeslání k hodnocení. Ve třech úloh si vyzkoušíte naprogramovat regresní a hodnotovou funkci a metodu postupného

klesání. Ke splnění třetí úlohy musíte mít vyřešenou hodnotovou funkci, protože ji tato metoda využívá. Doporučuji volání Vámi již vypracované funkce. Celkový počet možných bodů ze všech úloh je 100 a každá úloha má svůj bodový příděl uvedený v závorce.

6.1 Regresní funkce (15 b)

V této úloze máte za úkol spočítat hodnotu regresní funkce ze zadaných parametrů. Pro připomenutí má tato funkce následující tvar:

$$h_{\theta}(x) = \theta_0 + \theta_1 x.$$

6.2 Hodnotová funkce (35 b)

V této úloze budete řešit výpočet hodnotové funkce na připraveném datasetu. Můžete ji počítat libovolným způsobem, ale doporučuji výpočet udělat vektorově skrze knihovnu numpy. Hodnotová funkce má tvar

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2.$$

6.3 Metoda postupného klesání (50 b)

Poslední úloha je za největší počet bodů z toho důvodu, že je k ní potřeba mít vypočítanou i předchozí úlohu. Vaším úkolem je naprogramovat metodu postupného klesání tak, abyste našli optimální parametry θ_0 a θ_1 . Postupujte podle poslední přednášky. Cílem je postupná minimalizace hodnotové funkce pomocí jejích parciálních derivací:

$$\theta_j = \theta_j - \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1).$$