



Gépi tanulás a gyakorlatban

Octave



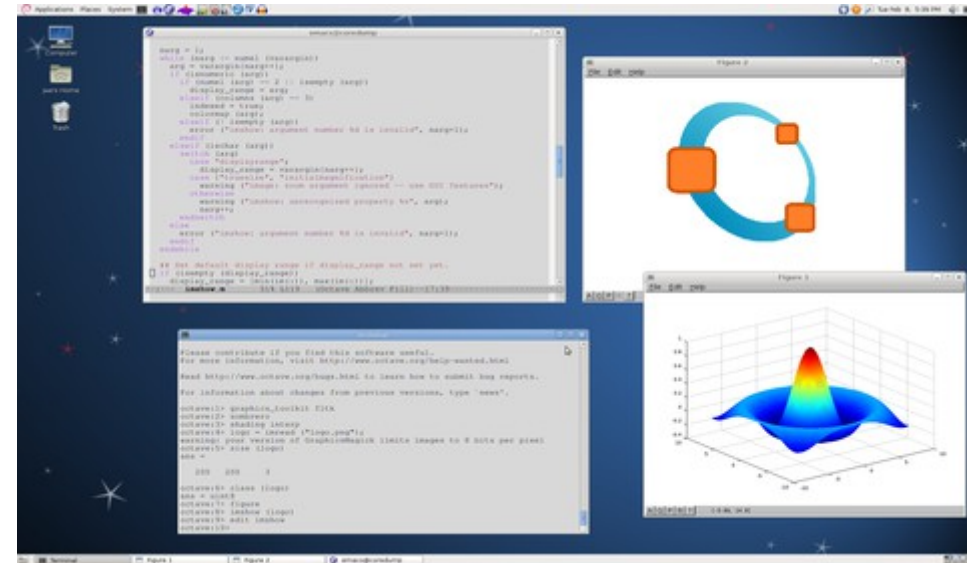
Mi az Octave?

- Az Octave:

- *numerikus* számításokra és
- *grafikus* megjelenítésére kihegyezett
- *interaktív*
- *programozási* környezet
- nyílt forráskódú → teljesen *ingyenes*

- Mi *nem* az Octave:

- (elsősorban) nem alkalmas *szimbolikus* számításokra → nincs egzakt megoldás
- nem *általános célú* nyelv → gyorsabb programozás, kevesebb optimalizáció (pl. natív támogatás)





Elérhetőség, telepítés, indítás

- Ingyenesen letölthető:
 - <http://www.gnu.org/software/octave/download.html>
- Elérhető:
 - Linux, Windows, Mac rendszerekre
- Telepítési útmutató:
 - Windows: megszokott telepítés, útmutató hamarosan a weboldalon
 - Linux (Debian alapú rendszerek): `apt-get install octave`
- Indítás:
 - Windows: dupla klikk
 - Linux: `octave`



Lináris algebra ismétlés

- Lineáris algebra ismétlés (vektor = oszlop vektor):
 - v és w vektorok belső szorzata (ha a dimenziójuk megegyezik):
 $v_1*w_1 + v_2*w_2 + \dots = v'*w \rightarrow$ eredmény egy szám
 - transzponálás:
 - főátlóra történő tükrözés
 - sorok \rightarrow oszlopokba
 - mátrix szorzás: A $n \times m$ -es mátrix, B $m \times k$ mátrix esetén $AB=A*B$
 $n \times k$ -s mátrix $AB(i,j) = A$ **i -dik** sorának és **B j -dik** oszlopának mint
 - mátrix inverze: A $n \times n$ négyzetes mátrix inverze $invA$ $n \times n$, melyre $A*invA = invA*A = E \leftarrow$ egység mátrix főátlóban 1, egyébként 0 **(inverz nem mindig létezik!)**



Octave programozás

- Alapok:
 - Minden mátrix!
 - Minden művelet mátrixokon értelmezett!
 - Egy sorba több művelet is írható
 - ; zárt műveletek eredménye nem jelenik meg
 - , vagy – sor végén – üresen hagyott lezárás hatására az eredmény a képernyőre íródik
 - Értékadás művelete: =
 - Konstans: 1x1-es mátrix pl. **A=10**
 - Mátrix létrehozás: [és] között,
 - Oszlop elválasztó: , vagy ' ' (whitespace) pl. **A=[10 20; 30 10]**
 - Sor elválasztó: ;
 - Szöveg konstans: ” között, pl. **str = ”valami”**



Octave programozás

- Környezet:
 - Megjelenítés:
 - **format long/format short**
 - Munkamenet mentése:
 - **save/load fileName.mat**
 - Változók kezelése:
 - **who** - változók listázása
 - **whos** - változók listázása (részletes)
 - **clear v** - v nevű változó törlése
 - **clear** - minden változó törlése
 - Könyvtárak kezelése:
 - **pwd** - aktuális könyvtár
 - **cd path** - könyvtár váltás path-ban adott könyvtárra
 - **ls** - aktuális könyvtár listázása



Octave programozás

- Elemi aritmetikai műveletek:
 - Összeadás: $3+2$
 - Kivonás: $5-3$
 - Szorzás: $3*7$ (← mátrix szorzás), $3.*7$ (← elemenkénti szorzás)
 - Osztás: $6/7$ (← inverzzel való szorzás) $6./7$ (← elemenkénti osztás)
 - Hatványozás: 2^3 (← mátrix szorzás) $2.^3$ (← elemenkénti hatványozás)
 - Transzponálás: $[10\ 20; 30\ 10]'$
- Logikai műveletek:
 - Egyenlő-e: `==`
 - Nemegyenlő: `~=`
 - Logikai nem: `~`
 - Logikai és: `&&`
 - Logikai vagy: `||`
 - Logikai kizáró vagy: `xor(0,1)`

Mátrix műveleteknél a dimenzió egyezésekre figyelni kell!



Octave programozás

- Adottak $u=[1, 2]'$ és $v=[3, 1]'$ *oszlopvektorok*.
 1. Kvíz: Mi a dimenziója az $u * v'$ szorzatnak?
 2. Kvíz: Mi a dimenziója az $u * v$ szorzatnak?
 3. Kvíz: Mi a dimenziója a $v' * u$ szorzatnak?
 1. 1×1 , azaz szám
 2. Nem lehet elvégezni a műveletet
 3. 2×2 -es mátrix
 4. 2×1 -es vektor
- Mindhárom kvíz válasz halmaza a fenti négy alternatíva.



Octave programozás

- Dimenzió
 - sorok száma/vektor hossza: **length(A)**
 - dimenzió: **size(A)**
- Indexelés
 - i,j pozíció elérése: **A(i,j)**
 - i-k intervallumba eső sorok és j-dik oszlop elérése: **A(i:k,j)**
 - : teljes intervallum: **A(:,j)** (← összes sor és j-dik oszlop elérése)
 - end intervallum jobb oldali „vége” (bal mindig 1) – csak indexelésnél: **A(2:end,j)**
 - intervallum bejárása lépésközzel: **1:step:end**
 - i1,i2,...,in sorok elérése és j-dik oszlop elérése: **A([i1 i2 ... in],j)**
 - Mátrix „sorosítása” (mátrix → vektor) indexeléssel: **A(:)**
- Adatok mozgatása:
 - Blokk mátrix alapú bővítés: **[A [1;2]; [100 200 300]]**
 - Túlindexeléssel: **A(end+1, :)= [5 6]**



Octave programozás

- Legyen adott az A és a B mátrix (oldalt).

4. Kvíz: Melyik művelet a helyesek?

1. $A+B$
2. $A*B$
3. $B'*A$
4. $A.*B$

$$A = \begin{bmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}$$

4. Kvíz: Melyik index kiválasztás eredményezi B-t?

1. $A(:, [1 \ 2])$
2. $A(:, 1:3)$
3. $A(:, 1:2)$
4. $A(1:2, :)$

$$B = \begin{bmatrix} 16 & 2 \\ 5 & 11 \\ 9 & 7 \\ 4 & 14 \end{bmatrix}$$

- Jelöld meg az összes helyes megoldást.



Octave programozás

- Függvények:
 - 0-val feltöltött tömb deklaráció: **zeros(n,m)**, n és m a dimenzió méret
 - 1-kel feltöltött tömb deklaráció: **ones(n,m)**
 - egység mátrix deklaráció: **eye(n)**, nxn dimenziós egység mátrix
 - Oszloponkénti összegzés/szorzás: **sum(A)**, **prod(A)**
 - Matematikai függvények elemenként hajtódnak végre: **exp()**, **log()**, **log10()**, **log2()**, **sin()**, **cos()**, ...
 - Segítség a függvényekről: **help függvénynév**
 - Függvények több paraméterrel is visszatérhetnek, pl. **[V,L] = eig(A)**



Octave programozás

- Saját függvények írás:
 - Aktuális könyvtárban, függvény nevével megegyező nevű fájl + m kiterjesztés
 - pl. `fv` függvény megvalósítása (paraméter 2 mátrix; visszatér 2 értékkel, ami a paraméter mátrixok elemeinek összegét tartalmazza)

```
function [sA, sB]=fv(A,B)
```

```
    sA = sum(sum(A));
```

```
    sB = sum(sum(B));
```

```
end
```



Octave programozás

- Rajzolás:
 - `x = [0:0.01:0.98]; y = sin(2*pi*4*x); plot(x,y);` ← jóval több lehetőség, nem tárgyaljuk részletesen
- Vezérlési szerkezetek:
 - for:
`for i=from:step:until,`
`v(i) = 2^i;`
`end`



Octave programozás

- Vezérlési szerkezetek:

- while:

```
i = 1;
```

```
while i <= 5,
```

```
    v(i) = 100;
```

```
    i = i+1;
```

```
end
```

- If-then-else:

```
if v(1)==1,
```

```
    disp('The value is one!');
```

```
elseif v(1)==2,
```

```
    disp('The value is two!');
```

```
else
```

```
    disp('The value is not one or two!');
```

```
end
```

- Megtanultunk Octave-ban programozni, bővebben:

<http://www-mdp.eng.cam.ac.uk/web/CD/engapps/octave/octavetut.pdf>



Keretrendszer bemutatása

- Minden HF-ben lesz egy submit.m, ami a feltöltést biztosítja Octave-ból:

```
octave:1> submit
```

Gépi tanulás a gyakorlatban feladat beküldés.

Elérhető feladatok:

1 Mátrix szorzás (submit.m)

2 Mátrix invertálás (invert.m)

Feltöltendő [1-2]: 2

E-mail cím: ormandi.robert@gmail.com

Jelszó: 8e1a77970cc8990fbc39550f89a1132f9a9df590

Sikertelen feltöltés, próbálkozz újra!



Weka programozási alapok

- A múlt alkalommal megnéztük, hogy hogyan tudunk tanulni a Weka GUI használatával.
- Most megnézzük azt, hogy hogyan tudjuk – majdnem – ugyanezt megtenni API szinten.
- Forráskódok elérhetőek lesznek a weboldalon.
- Továbbiak:

<http://weka.wikispaces.com/>

Use+WEKA+in+your+Java+code