



# Вычислительные модели с использованием научных библиотек Python Линейная алгебра



# Базовые типы, dense matrix

# 1

```
>>> import numpy as np
>>> from scipy import linalg
>>> A = np.array([[1,2],[3,4]])
>>> A
array([[1, 2], [3, 4]])
>>> linalg.inv(A)
array([[ -2. ,  1. ], [ 1.5, -0.5]])
>>> b = np.array([[5,6]]) #2D array
>>> b
array([[5, 6]])
>>> b.T
array([[5], [6]])
>>> A*b #not matrix multiplication!
array([[ 5, 12], [15, 24]])
>>> A.dot(b.T) #matrix multiplication
array([[17], [39]])
```

# 2

```
>>> import numpy as np
>>> A = np.mat('[1 2;3 4]')
>>> A
matrix([[1, 2], [3, 4]])
>>> A.I
matrix([[ -2. ,  1. ], [ 1.5, -0.5]])
>>> b = np.mat('[5 6]')
>>> b
matrix([[5, 6]])
>>> b.T
matrix([[5], [6]])
>>> A*b.T
matrix([[17], [39]])
```

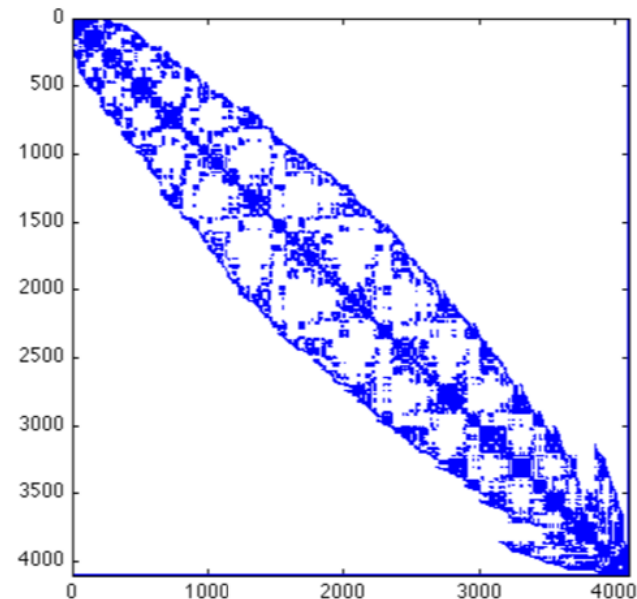


# Базовые типы, sparse matrix

## Способы хранения

1. csc\_matrix: Compressed Sparse Column format
2. csr\_matrix: Compressed Sparse Row format
3. bsr\_matrix: Block Sparse Row format
4. lil\_matrix: List of Lists format
5. dok\_matrix: Dictionary of Keys format
6. coo\_matrix: COOrdinate format (aka IJV, triplet format)
7. dia\_matrix: DIAGONAL format

```
>>> import numpy as np
>>> import scipy.sparse as sps
```



## Постановка задачи

$$\mathbf{A}\mathbf{u} = \mathbf{f}$$

## Число обусловленности матрицы $\mathbf{A}$

$$\mu(\mathbf{A}) = \|\mathbf{A}^{-1}\| \|\mathbf{A}\|$$

$$\mu \approx 1 \div 10$$

-хорошо обусловленная СЛАУ

$$\mu \gg 10^2 \div 10^3$$

-плохо обусловленная СЛАУ



# СЛАУ, точные методы

## LU-разложение

$$\mathbf{A} = \mathbf{LU}$$

$$\mathbf{L}\mathbf{v} = \mathbf{f}, \mathbf{U}\mathbf{u} = \mathbf{v}$$

```
>>> import numpy as np
>>> from scipy import linalg
>>> A = np.array([[1, 2], [3, 4]])
>>> A = np.array([[1, 2], [3, 4]])
>>> b = np.array([[5], [6]])
>>> b
array([[5], [6]])
>>> linalg.inv(A).dot(b) # slow
array([[-4. ], [ 4.5]])
>>> np.linalg.solve(A, b) # fast
array([[-4. ], [ 4.5]])
```

```
>>> import numpy as np
>>> from scipy.sparse import linalg
>>> mtx = sparse.spdiags([[1, 2, 3, 4, 5], [6, 5, 8, 9,
10]], [0, 1], 5, 5)
>>> mtx.todense()
matrix([[ 1, 5, 0, 0, 0],
[ 0, 2, 8, 0, 0],
[ 0, 0, 3, 9, 0],
[ 0, 0, 0, 4, 10],
[ 0, 0, 0, 0, 5]])
>>> rhs = np.array([1, 2, 3, 4, 5], dtype=np.float32)
>>> x= dsolve.spsolve(mtx1, rhs, use_umfpack=False)
```

# СЛАУ, точные методы

## Метод Холецкого

$$A = LL^T$$

$$L = \begin{pmatrix} l_{11} & 0 & \dots & 0 \\ l_{12} & l_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{1n} & l_{2n} & \dots & l_{nn} \end{pmatrix}$$

$$Lv = f, L^T u = v.$$

## Метод QR

$$A = Q \cdot R,$$

$Q$  – ортогональная

$R$  – верхняя треугольная

$$Q^T \cdot Q \cdot R \cdot x = Q^T \cdot b,$$

$$R \cdot x = Q^T \cdot b.$$

# СЛАУ, итерационные методы

## Список методов

- BiConjugate Gradient
- BiConjugate Gradient STABilized
- Conjugate Gradient
- Conjugate Gradient Squared
- Generalized Minimal RESidual(GMRES)
- LGMRES
- MINimum RESidual
- Quasi-Minimal Residual

```
>>> import numpy as np  
>>> import scipy.sparse.linalg as linalg
```

# СЛАУ, предобусловливание

## Общая идея

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x}=\mathbf{M}^{-1}\mathbf{b},$$

$\mathbf{M}$  должна быть по возможности близка к матрице  $\mathbf{A}$ ;

$\mathbf{M}$  должна быть легко вычислима;

$\mathbf{M}$  должна быть легко обратима.

## ILU разложение

$$\mathbf{M}=\mathbf{LU}+\mathbf{R}\approx \mathbf{LU}$$

Функция `spilu()`





# Задание

## Рекомендовать пользователям новые фильмы

userId	movieId	rating	timestamp
1	1	5	847117005
1	2	3	847642142
1	10	3	847641896
1	32	4	847642008
1	34	4	847641956
1	47	3	847641956
1	50	4	847642073
1	62	4	847642105
1	150	4	847116751
1	153	3	847116787
1	160	3	847642008
1	161	4	847641896
1	165	4	847116787
1	185	3	847641919

movieId	title	genres
1	Toy Story (1995)	Adventure   Animation   Children
2	Jumanji (1995)	Adventure   Children   Fantasy
3	Grumpier Old Men (1995)	Comedy   Romance
4	Waiting to Exhale (1995)	Comedy   Drama   Romance
5	Father of the Bride Part II (1995)	Comedy
6	Heat (1995)	Action   Crime   Thriller
7	Sabrina (1995)	Comedy   Romance
8	Tom and Huck (1995)	Adventure   Children
9	Sudden Death (1995)	Action
10	GoldenEye (1995)	Action   Adventure   Thriller
11	American President, The (1995)	Comedy   Drama   Romance
12	Dracula: Dead and Loving It (1995)	Comedy   Horror
13	Balto (1995)	Adventure   Animation   Children
14	Nixon (1995)	Drama

Users (instances)	Movies (features)				
		Movie 1	Movie 2	Movie 3	Movie 4
	User 1	1	3	2	1
	User 2	2	--	--	5
	User 3	5	1	5	3
User 4	4	--	1	4	

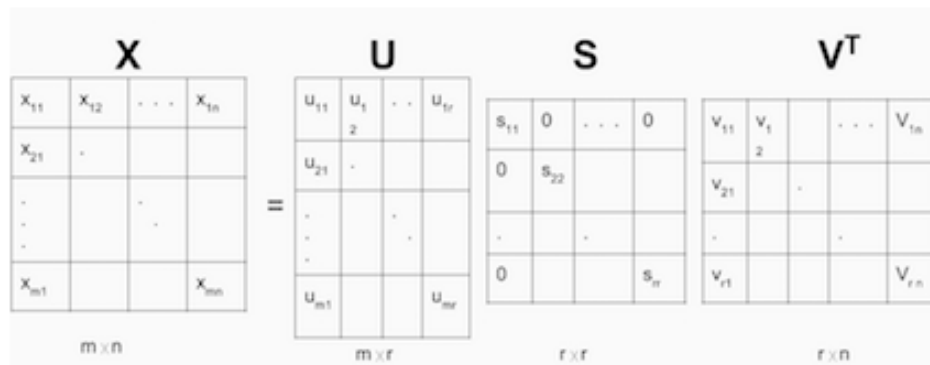
8913 – фильмов

718 – пользователей

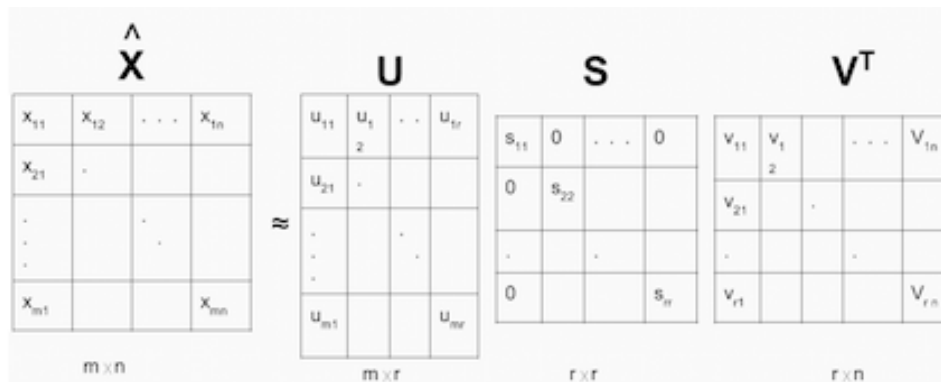
# Задание

## Подход на основе SVD разложения

$$X = U \times S \times V^T$$



← Признаковое представление пользователей и фильмов



← Прогноз рейтингов для новых фильмов