Άνω διδιαγωνοποίηση Golub-Kahan

In [1]:
```
using LinearAlgebra
using Distributions
using Statistics
```

Θεώρημα

Έστω $A \in \mathbb{R}^{m\times n}$, $m \geq n$. Τότε υπάρχουν ορθογώνιοι πίνακες $U \in \mathbb{R}^{m\times m}$ και $V \in \mathbb{R}^{n\times n}$ τέτοιοι ώστε:

$$U^T A V = B = \left[\begin{array}{ccccc} a_1 & \beta_1 & \dots & \dots & 0 \\ 0 & a_2 & \beta_2 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & 0 & a_{n-1} & \beta_{n-1} \\ 0 & \dots & \dots & 0 & a_n \\ \hline & & & & \\ & & \Huge 0 & & \end{array}\right] \tag{1}$$

Κατασκευή πινάκων U, V και B

Έχουμε τις εξής σχέσεις:

$$\begin{gathered} AV = UB \\ A^T U = V B^T \end{gathered}$$

Διαμερίσεις κατά στήλων των U και V:

$$U = [u_1 | \dots | u_m] \quad , \quad V = [v_1 | \dots | v_n]$$

έχουμε:

$$A v_k = a_k u_k + \beta_{k-1} u_{k-1}, \tag{2}$$
$$A^T u_k = a_k v_k + \beta_k v_{k+1} \tag{3}$$

για $k = 1 : n$, με τη σύμβαση ότι $\beta_0 u_0 \equiv 0$ και $\beta_n u_{n+1} \equiv 0$.

Ορίζουμε τα διανύσματα:

$$r_k = A v_k - \beta_{k-1} u_{k-1}, \tag{4}$$
$$p_k = A^T u_k - a_k v_k \tag{5}$$

Από ορθοκανονικότητα των u, έχουμε:

$$\begin{gathered} a_k = \pm ||r_k||_2, \\ u_k = r_k / a_k, \quad (a_k \neq 0) \end{gathered}$$

Αν το $a_k = 0$, τότε ο πίνακας $A(:, 1:k)$ είναι ελλειπούς τάξης.

Παρόμοια συμπεραίνουμε ότι:

$$\begin{gathered} \beta_k = \pm ||p_k||_2, \\ v_{k+1} = p_k / \beta_k, \quad (\beta_k \neq 0) \end{gathered}$$

```
Αλγόριθμος: Άνω διδιαγωνοποίηση Golub-Kahan
```

Είσοδος: $A \in \mathbb{R}^{m \times n}$ με $rank(A) = n$

1. $k = 0$, $p_0 = v_c$, $\beta_0 = 1$, $u_0 = 0$ (όπου, v_c 1η στήλη του V, τυχαίο κανονικοποιημένο διάνυσμα)

2. while $\beta_k \neq 0$

$$\begin{aligned} &v_{k+1} = p_k / \beta_k \\ &k = k + 1 \\ &r_k = A v_k - \beta_{k-1} u_{k-1} \\ &a_k = ||r_k||_2 \\ &u_k = r_k / a_k \\ &p_k = A^T u_k - a_k v_k \\ &\beta_k = ||p_k||_2 \end{aligned}$$

Πρόταση

Έστω $A \in \mathbb{R}^{m \times n}$, $m \geq n$, τέτοιος ώστε:

$$U^T A V = B$$

$U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ ορθογώνιοι και B διδιαγώνιος.

Ισχύουν τα ακόλουθα:

i) $\sigma(B) = \sigma(A)$

ii) $rank(A) = rank(B)$

iii) $||A||_2 = ||B||_2$

iv) $||A||_F = ||B||_F$

Απόδειξη

i) Έχουμε τις σχέσεις:

$$\begin{gathered} AV = UB \\ A^T U = V B^T \end{gathered}$$

Παρατηρούμε ότι:

$$\begin{aligned} V(B^T B)V^T = (VB^T)BV^T &= (A^T U)BV^T \\ &= A^T(UB)V^T = A^T(AV)V^T \\ &= A^T A(VV^T) = A^T A \end{aligned}$$

Άρα οι πίνακες $A^T A$ και $B^T B$ έχουν τις ίδιες ιδιοτιμές.

ii)

$$rank(A) = rank(U^T A V) = rank(B)$$

iii)

$$
\begin{aligned}
||A||_2 &= \sqrt{\max eigenvalues(A^T A)} \\
&= \sqrt{\max eigenvalues(B^T B)} = ||B||_2
\end{aligned}
$$

iv)

$$
\begin{aligned}
||A||_F = ||U^T A V||_F &= ||B||_F \\
&= \sqrt{\sum_{i=1}^{r} b_{i,i}^2 + \sum_{i=1}^{r} b_{i,i+1}^2}
\end{aligned}
$$

Παράδειγμα:

Έστω $X \in \mathbb{R}^{6\times5}$ όπου $x_j \sim N(0,1)$, για $j = 1, \ldots, 4$:

In [2]:
```
function X_gen_wi(r, c, μ, σ)

    X = ones(r, c)
    for j = 2:c
        x = rand(Normal(μ, σ), r)
        while abs.(mean(x)) > 1e-2
            x = rand(Normal(μ, σ), r)
        end
        X[:, j] = x
    end

    return X
end
```

Out[2]:
```
X_gen_wi (generic function with 1 method)
```

In [3]:
```
X = X_gen_wi(6, 5, 0, 1)
```

Out[3]:
```
6×5 Matrix{Float64}:
 1.0   0.80335    0.0414786   0.679752  -0.0793035
 1.0  -0.613232   1.369       2.06523    0.33859
 1.0  -0.335788  -1.50575    -2.10626   -0.85044
 1.0  -0.581439   0.520122   -0.961605  -0.680528
 1.0   0.605846  -0.602642    0.768666   0.910311
 1.0   0.14313    0.130375   -0.457502   0.405988
```

In [4]:
```
function GKUB(A)
    r, c = size(A)

    # Random Unit 2-norm Vector
    x = rand(c, 1)
    p = x ./ norm(x)

    # Initialization
    b = 1
    u = zeros(r, 1)
    a_vec = zeros(1, c)
    b_vec = zeros(1, c)
    U = zeros(r, c)
    V = zeros(c, c)

    for k = 1:c

        # Process
        v = p / b
```

```
        r = A * v - b * u
        a = norm(r)
        u = r / a
        p = A' * u .- a * v
        b = norm(p)


        # Creation and displayof a,b,U,V
        println("\nFor $k-ith iteration we get the following results:")
        a_vec[k] = a
        println("\na value is = $a")
        b_vec[k] = b
        println("b value is = $b")
        U[:, k] = u
        println("\nU matrix is as follows:")
        display(U)
        V[:, k] = v
        println("\nV matrix is as follows:")
        display(V)
    end

    # Upper Bidiagonal
    Bu = Bidiagonal(vec(a_vec), b_vec[1:end-1], :U)
    println("\n And the Bidiagonal matrix is: ")
    display(Bu)

    return Bu, U, V

end
```

Out[4]: GKUB (generic function with 1 method)

In [5]:
```
Bu, U, V = GKUB(X);
```

```
For 1-ith iteration we get the following results:

a value is = 2.3801585004094252
b value is = 0.9121998102057345

U matrix is as follows:
6×5 Matrix{Float64}:
 0.403682  0.0  0.0  0.0  0.0
 0.482101  0.0  0.0  0.0  0.0
 0.182141  0.0  0.0  0.0  0.0
 0.251481  0.0  0.0  0.0  0.0
 0.553768  0.0  0.0  0.0  0.0
 0.448935  0.0  0.0  0.0  0.0
V matrix is as follows:
5×5 Matrix{Float64}:
 0.918412   0.0  0.0  0.0  0.0
 0.0537153  0.0  0.0  0.0  0.0
 0.0314995  0.0  0.0  0.0  0.0
 0.0423078  0.0  0.0  0.0  0.0
 0.388397   0.0  0.0  0.0  0.0
For 2-ith iteration we get the following results:

a value is = 2.516559939239301
b value is = 2.6216972088349486

U matrix is as follows:
6×5 Matrix{Float64}:
 0.403682   0.19013    0.0  0.0  0.0
 0.482101   0.592496   0.0  0.0  0.0
 0.182141  -0.681221   0.0  0.0  0.0
 0.251481  -0.205732   0.0  0.0  0.0
 0.553768  -0.0806924  0.0  0.0  0.0
 0.448935  -0.31607    0.0  0.0  0.0
```

```
V matrix is as follows:
5×5 Matrix{Float64}:
 0.918412    0.149246  0.0  0.0  0.0
 0.0537153   0.102148  0.0  0.0  0.0
 0.0314995   0.20074   0.0  0.0  0.0
 0.0423078   0.837722  0.0  0.0  0.0
 0.388397   -0.47457   0.0  0.0  0.0
For 3-ith iteration we get the following results:

a value is = 1.302008316224737
b value is = 0.6256251412441101

U matrix is as follows:
6×5 Matrix{Float64}:
 0.403682   0.19013    -0.527749  0.0  0.0
 0.482101   0.592496   -0.145216  0.0  0.0
 0.182141  -0.681221   -0.47618   0.0  0.0
 0.251481  -0.205732   -0.265973  0.0  0.0
 0.553768  -0.0806924   0.371018  0.0  0.0
 0.448935  -0.31607     0.515025  0.0  0.0
V matrix is as follows:
5×5 Matrix{Float64}:
 0.918412    0.149246  -0.334393   0.0  0.0
 0.0537153   0.102148  -0.0814049  0.0  0.0
 0.0314995   0.20074    0.472977   0.0  0.0
 0.0423078   0.837722   0.366154   0.0  0.0
 0.388397   -0.47457    0.723725   0.0  0.0
For 4-ith iteration we get the following results:

a value is = 1.8506434847318782
b value is = 0.03517455718054779

U matrix is as follows:
6×5 Matrix{Float64}:
 0.403682   0.19013    -0.527749   0.438129   0.0
 0.482101   0.592496   -0.145216  -0.36314    0.0
 0.182141  -0.681221   -0.47618    0.0583418  0.0
 0.251481  -0.205732   -0.265973  -0.62129    0.0
 0.553768  -0.0806924   0.371018   0.469102   0.0
 0.448935  -0.31607     0.515025  -0.258283   0.0
V matrix is as follows:
5×5 Matrix{Float64}:
 0.918412    0.149246  -0.334393   -0.149757  0.0
 0.0537153   0.102148  -0.0814049   0.613965  0.0
 0.0314995   0.20074    0.472977   -0.662196  0.0
 0.0423078   0.837722   0.366154    0.276371  0.0
 0.388397   -0.47457    0.723725    0.292806  0.0
For 5-ith iteration we get the following results:

a value is = 0.7915057643131803
b value is = 6.070864548156873e-13

U matrix is as follows:
6×5 Matrix{Float64}:
 0.403682   0.19013    -0.527749   0.438129    0.557722
 0.482101   0.592496   -0.145216  -0.36314    -0.459964
 0.182141  -0.681221   -0.47618    0.0583418  -0.458391
 0.251481  -0.205732   -0.265973  -0.62129     0.27466
 0.553768  -0.0806924   0.371018   0.469102   -0.263623
 0.448935  -0.31607     0.515025  -0.258283    0.349745
V matrix is as follows:
5×5 Matrix{Float64}:
 0.918412    0.149246  -0.334393   -0.149757   0.000188264
 0.0537153   0.102148  -0.0814049   0.613965   0.776595
 0.0314995   0.20074    0.472977   -0.662196   0.544519
 0.0423078   0.837722   0.366154    0.276371  -0.293228
 0.388397   -0.47457    0.723725    0.292806  -0.120068
```

```
 And the Bidiagonal matrix is:
5×5 Bidiagonal{Float64, Vector{Float64}}:
 2.38016  0.9122    ⋅        ⋅         ⋅
  ⋅       2.51656  2.6217    ⋅         ⋅
  ⋅        ⋅       1.30201  0.625625   ⋅
  ⋅        ⋅        ⋅       1.85064   0.0351746
  ⋅        ⋅        ⋅        ⋅        0.791506
```

Ελάχιστα Τετράγωνα

Το μοντέλο που γενικά περιγράφει οποιοδήποτε πραγματικό σύνολο δεδομένων είναι της μορφής:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

όπου $\mathbf{y}$ είναι $(n \times 1)$ το διάνυσμα των responses, $\mathbf{X}$ είναι $(n \times p)$ πίνακας σχεδιασμού, $\boldsymbol{\beta}$ είναι $(p \times 1)$ διάνυσμα αγνώστων συντελεστών και $\boldsymbol{\varepsilon}$ είναι $(n \times 1)$ διάνυσμα τυχαίου σφάλματος, όπου $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)'$ και $\varepsilon \sim N(0, \sigma^2 I_n)$.

Έστω $X \in \mathbb{R}^{m \times n}$, $m \geq n, rank(X) = n$. Χρησιμοποιώντας την άνω διδιαγωνοποίησης Golub-Kahan, ο X παραγοντοποιείται ως εξής:

$$X = UBV^T = [U_1 \; U_2] \begin{bmatrix} B_1 \\ 0 \end{bmatrix} V^T = U_1 B_1 V^T$$

όπου $U_1 \in \mathbb{R}^{m \times n}$, $V^T \in \mathbb{R}^{n \times n}$ με ορθοκανονικές στήλες ακαι $B_1 \in \mathbb{R}^{n \times n}$ διαγώνιος.

Λογική επίλυσης:

$$\begin{aligned}
&\begin{cases} y = X\beta = U_1 B_1 V^T \beta \\ \Theta έτουμε \;\; V^T\beta = w \end{cases} \Rightarrow \; y = U_1 B_1 w \Rightarrow \; U_1^T y = B_1 w \Rightarrow \\
&\Theta έτουμε \;\; y' = U_1^T y \\
&Επίλυση\ διδιαγώνιου\ συστήματος\text{: } B_1 w = y' \\
&Υπολογισμός\ των\ συντελεστών\ \beta : \;\; \beta = Vw
\end{aligned}$$

Αλγόριθμος:

1. Εφαρμογή της Golub-Kahan διδιαγωνοποίησης στο X, i.e.:

$$X = U_1 \cdot B_1 \cdot V^T$$

όπου $U_1 \;\; m \times n$, $V \;\; n \times n$ ορθογώνιοι πίνακες και $B_1 \;\; n \times n$ διδιαγώνιος.

2. Θέτουμε $y' = U_1^T y$.

3. Επίλυση διδιαγώνιου συστήματος: $B_1 w = y'$

4. Υπολογισμός των συντελεστών:

$$\hat{\beta} = V \cdot w$$

Παράδειγμα

In [6]:
```
X
```

Out[6]:
```
6×5 Matrix{Float64}:
 1.0   0.80335    0.0414786   0.679752  -0.0793035
 1.0  -0.613232   1.369       2.06523    0.33859
 1.0  -0.335788  -1.50575    -2.10626   -0.85044
 1.0  -0.581439   0.520122   -0.961605  -0.680528
 1.0   0.605846  -0.602642    0.768666   0.910311
 1.0   0.14313    0.130375   -0.457502   0.405988
```

Θεωρώ ένα γραμμικό μοντέλο:

$$y = \beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_4 x_4 = 8x_1 + 3x_2 + \varepsilon,$$

με $\varepsilon \sim N_6(0, \sigma^2 I_6)$

In [7]:
```
# Coefficients

b = [0,8,3,0,0]
```

Out[7]:
```
5-element Vector{Int64}:
 0
 8
 3
 0
 0
```

In [8]:
```
# Error

ε = rand(Normal(0,0.2),size(X,1))
```

Out[8]:
```
6-element Vector{Float64}:
 -0.2768414338869029
 -0.3721912294939491
 -0.18218706581482014
 -0.18944768415687768
 -0.356629488940073
 -0.28327184457013477
```

In [9]:
```
# y with noise

y = X*b + ε
```

Out[9]:
```
6-element Vector{Float64}:
  6.274392633698537
 -1.1710496184398425
 -7.385741474944603
 -3.2805902263477114
  2.682208974660359
  1.252895980581617
```

- Νωρίτερα πραγματοποιήσαμε την διδιαγωνοποίηση, οπότε υπολογίζουμε το $y' = U_1^T y$:

In [10…
```
y_ton = U' * y
```

Out[10]:
```
5-element Vector{Float64}:
 1.8458342383785291
 5.592911617587741
 2.888663007080618
 5.716178250586674
 6.253622652823209
```

- Επίλυση του διδιαγώνιου συστήματος $B_1 w = y'$:

In [11…
```
w = Bu \ y_ton
```

Out[11]:
```
5-element Vector{Float64}:
 0.24580338996333204
 1.3821349184928011
 0.8066096435387833
 2.9385816234784503
 7.900918647446257
```

Υπολογισμός των συντελεστών $\hat{\beta} = V \cdot w$:

In [12…
```
b_hat = V * w
```

Out[12]:
```
5-element Vector{Float64}:
 -0.27628204189778616
  8.028727622328985
  3.022981447559847
 -0.04104504261730346
 -0.06490192488920876
```

- Υπενθύμιση των συντελεστών που ορίσαμε ήταν $\beta_1 = 8$ και $\beta_2 = 3$.
- $||\hat{\beta} - \beta||_2$:

In [13…
```
norm(b_hat - b)
```

Out[13]:
```
0.28910576830913215
```