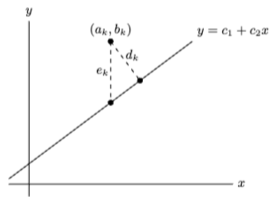
ECE 133A - Homework 2

Sunday, October 15, 2023 6:34 PM

1.8 *Orthogonal distance regression.* We use the same notation as in exercise 1.7: a, b are non-constant n-vectors, with means m_a, m_b, standard deviations s_a, s_b, and correlation coefficient ρ.



For each point (a_k, b_k), the vertical deviation from the straight line defined by $y = c_1 + c_2 x$ is given by

$$e_k = |c_1 + c_2 a_k - b_k|.$$

The least squares regression method of the lecture minimizes the sum $\sum_k e_k^2$ of the squared vertical deviations. The orthogonal (shortest) distance of (a_k, b_k) to the line is

$$d_k = \frac{|c_1 + c_2 a_k - b_k|}{\sqrt{1 + c_2^2}}.$$

As an alternative to the least squares method, we can find the straight line that minimizes the sum of the squared orthogonal distances $\sum_k d_k^2$. Define

$$J = \frac{1}{n} \sum_{k=1}^{n} d_k^2 = \frac{\|c_1 \mathbf{1} + c_2 a - b\|^2}{n(1 + c_2^2)}.$$

(a) Show that the optimal value of c_1 is $c_1 = m_b - m_a c_2$, as for the least squares fit.

$$\frac{d}{dc_1} J = \frac{d}{dc_1}\left(\frac{\|c_1 \mathbf{1} + c_2 a - b\|^2}{n(1+c_2^2)} = \frac{(c_1 + c_2 a_1 - b_1)^2 + \ldots + (c_1 + c_2 a_n - b_n)^2}{n(1+c_2^2)} \right)$$

$$= \frac{2(c_1 + c_2 a_1 - b_1) + \ldots + 2(c_1 + c_2 a_n - b_n)}{n(1+c_2^2)}$$

$$0 = \frac{2nc_1 + 2\sum_{i=1}^{n} c_2 a_i - b_i}{n(1+c_2^2)}$$

$$\to 2nc_1 = 2\sum_{i=1}^{n} b_i - c_2 a_i \to c_1 = \frac{\sum_{i=1}^{n} b_i - c_2 a_i}{n} = \sum_{i=1}^{n} \frac{b_i}{n} - c_2 \sum_{i=1}^{n} \frac{a_i}{n}$$

$$\Rightarrow \boxed{c_1 = m_b - c_2 m_a}$$

(b) If we substitute $c_1 = m_b - m_a c_2$ in the expression for J, we obtain

$$J = \frac{\|c_2(a - m_a\mathbf{1}) - (b - m_b\mathbf{1})\|^2}{n(1+c_2^2)}.$$

Simplify this expression and show that it is equal to

$$J = \frac{s_a^2 c_2^2 + s_b^2 - 2\rho s_a s_b c_2}{1 + c_2^2}.$$

Set the derivative of J with respect to c_2 to zero, to derive a quadratic equation for c_2:

$$\rho c_2^2 + \left(\frac{s_a}{s_b} - \frac{s_b}{s_a}\right) c_2 - \rho = 0.$$

If $\rho = 0$ and $s_a = s_b$, any value of c_2 is optimal. If $\rho = 0$ and $s_a \neq s_b$ the quadratic equation has a unique solution $c_2 = 0$. If $\rho \neq 0$, the quadratic equation has a positive and a negative root. Show that the solution that minimizes J is the root c_2 with the same sign as ρ.

$$s_a = \frac{1}{\sqrt{n}} \|a - m_a\mathbf{1}\|$$

$$\rho = \frac{1}{n} \frac{(a - m_a\mathbf{1})^T(b - m_b\mathbf{1})}{s_a s_b}$$

$$J = \frac{(c_2(a_1 - m_{a_1}) - (b_1 - m_{b_1}))^2 + \dots + (c_2(a_n - m_{a_n}) - (b_n - m_{b_n}))^2}{n(1+c_2^2)}$$

$$(c_2(a_1 - m_{a_1}) - (b_1 - m_{b_1}))^2 = c_2^2(a_1 - m_{a_1})^2 - 2c_2(a_1 - m_{a_1})(b_1 - m_{b_1}) + (b_1 - m_{b_1})^2$$

$$\Rightarrow J = \frac{c_2^2\left[(a_1 - m_{a_1})^2 + \dots + (a_n - m_{a_n})^2\right] - 2c_2\left[(a_1 - m_{a_1})(b_1 - m_{b_1}) + \dots + (a_n - m_{a_n})(b_n - m_{b_n})\right] + \left[(b_1 - m_{b_1})^2 + \dots + (b_n - m_{b_n})^2\right]}{n(1+c_2^2)}$$

$$= \frac{c_2^2 \|a - m_a\mathbf{1}\|^2 - 2c_2(a - m_a\mathbf{1})^T(b - m_b\mathbf{1}) + \|b - m_b\mathbf{1}\|^2}{n(1+c_2^2)}$$

$$= \frac{c_2^2 \|a - m_a\mathbf{1}\|^2}{n(1+c_2^2)} - \frac{2c_2(a - m_a\mathbf{1})^T(b - m_b\mathbf{1})}{n(1+c_2^2)} + \frac{\|b - m_b\mathbf{1}\|^2}{n(1+c_2^2)}$$

$$= \frac{c_2^2 s_a^2}{1+c_2^2} - \frac{2c_2\rho s_a s_b}{1+c_2^2} + \frac{s_b^2}{1+c_2^2}$$

$$\Rightarrow \boxed{J = \frac{c_2^2 s_a^2 - 2c_2\rho s_a s_b + s_b^2}{1+c_2^2}}$$

$$\frac{dJ}{dc_2} = \frac{d}{dc_2}\left(\frac{c_2^2 s_a^2 - 2c_2\rho s_a s_b + s_b^2}{1+c_2^2}\right)$$

$$\Rightarrow 0 = \frac{(1+c_2^2)(2c_2 s_a^2 - 2\rho s_a s_b) - (c_2^2 s_a^2 - 2c_2 \rho s_a s_b + s_b^2)(2c_2)}{(1+c_2^2)^2}$$

$$\Rightarrow 0 = 2c_2 s_a^2 + \cancel{2c_2^3 s_a^2} - 2\rho s_a s_b - 2c_2^2 s_a s_b - \cancel{2c_2^3 s_a^2} + 4c_2^2 \rho s_a s_b - 2c_2 s_b^2$$

$$= 2c_2 s_a^2 - 2\rho s_a s_b + 2c_2^2 \rho s_a s_b - 2c_2 s_b^2$$

$$= c_2 s_a^2 - \rho s_a s_b + c_2^2 \rho s_a s_b - c_2 s_b^2$$

$$= \rho c_2^2 s_a s_b + (s_a^2 - s_b^2)c_2 - \rho s_a s_b$$

$$= \rho c_2^2 + \frac{(s_a^2 - s_b^2)}{s_a s_b} c_2 - \rho$$

$$\Rightarrow \boxed{0 = \rho c_2^2 + \left(\frac{s_a}{s_b} - \frac{s_b}{s_a}\right) c_2 - \rho}$$

$$c_2 = \frac{(s_b^2 - s_a^2) \pm \sqrt{s_a^4 + 2 s_a^2 s_b^2 (2\rho^2 - 1) + s_b^4}}{2 s_a s_b \rho}$$

by having ρ on the bottom, if we do not have the same sign for c_2 & ρ $\rightarrow$ $\frac{dJ}{dc_2} = \rho c_2^2 + \left(\frac{s_a}{s_b} - \frac{s_b}{s_a}\right)c_2 - \rho$ will end up adding $\rho c_2^2 + \rho$

instead of $\rho c_2^2 - \rho$

(c) Download the file **orthregdata.m** and execute it in MATLAB to create two arrays **a**, **b** of length 100. Fit a straight line to the data points (a_k, b_k) using orthogonal distance regression and compare with the least squares solution. Make a MATLAB plot of the two lines and the data points. Julia users can import the data using the command `include("orthregdata.m")`.

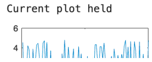
Homework 2

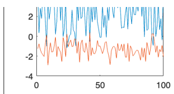
A1.8c

```
m_a = mean(a);
m_b = mean(b);
s_a = std(a);
s_b = std(b);|
p = 1/100 * transpose(a - m_a) * (b - m_b) / (s_a *s_b);
c2 = 0
c1 = m_b - m_a * c2
J = norm(c2*(a-m_a) - (b-m_b),2)^2 / (100 * (1 + c2^2))
plot(a);
hold;
plot(b);
hold;
plot(J);
```

```
c2 = 0
c1 = -1.2789
J = 0.5136

Current plot held
```

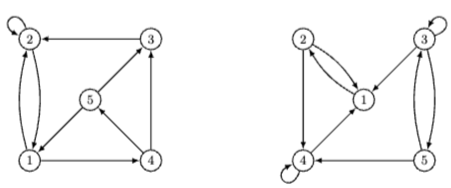





```
Current plot released
```

2.4 Let A be an $n \times n$ matrix with nonnegative elements ($A_{ij} \geq 0$ for all i, j). We define a directed graph G_A with vertices (nodes) $1, \ldots, n$, and an arc (directed edge) from vertex j to vertex i if only if $A_{ij} > 0$. The figure shows the graphs for the matrices

$$A_1 = \begin{bmatrix} 0 & 3 & 0 & 0 & 5 \\ 2 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 3 & 2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 0 & 3 & 1 & 2 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 2 & 0 & 1 & 4 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$



The matrix A is called *irreducible* if all the elements of the matrix $(I + A)^{n-1}$ (the $(n-1)$st power of $I + A$) are positive. Show that A is irreducible if and only if the graph G_A is *strongly connected*, *i.e.*, for every vertex i and every vertex j there is a directed path from vertex i to vertex j.

The matrix A_1 is an irreducible matrix. The matrix A_2 is not (for example, there is no directed path from vertex 1 to 3, or from 2 to 5).

iff $[(I + A)^{n-1}]_{ij} > 0$ then G_A is SC

//

$B = I + A \quad \therefore \quad \begin{cases} B_{ii} > 0 \\ B_{ij_{i \neq j}} = A_{ij} \end{cases}$

$[B^2]_{ii} = B_{i1}B_{1i} + \dots + B_{in}B_{ni} = \underset{>0}{B_{ii}^2} + \underset{\geq 0}{\sum_{k \neq i} B_{ik}B_{ki}} \quad \therefore \; > 0$

$[B^2]_{ij} = B_{i1}B_{1j} + \dots B_{in}B_{nj}$

define k: $B_{ik}B_{kj} > 0$ for off-diagonal elements $B = A$

$\underset{\substack{\text{path from} \\ k \text{ to } i}}{A_{ik} > 0} \;\; \& \;\; \underset{\substack{\text{path from} \\ j \text{ to } k}}{A_{kj} > 0}$

$\therefore \; \exists$ a path of $2 > x$

$i \leftarrow k \leftarrow j$

$[B^n] = B_{11} \dots B_{n1} + \dots + B_{1n} \dots B_{nn}$ diagonals are > 0 & $\exists$ a path $m \leq$ $\therefore$ prove

10.11 *Trace of matrix-matrix product.* The sum of the diagonal entries of a square matrix is called the *trace* of the matrix, denoted $\mathbf{tr}(A)$.

(a) Suppose A and B are $m \times n$ matrices. Show that

$$\mathbf{tr}(A^T B) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ij}.$$

What is the complexity of calculating $\mathbf{tr}(A^T B)$?

$C = \underset{n \times n}{A^T} \underset{m \times n}{B} \rightarrow \begin{bmatrix} A_{11} & \cdots & A_{m1} \\ \vdots & \ddots & \vdots \\ A_{1n} & \cdots & A_{mn} \end{bmatrix} \begin{bmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & \ddots & \vdots \\ B_{m1} & \cdots & B_{mn} \end{bmatrix} = \begin{bmatrix} (A_{11}B_{11} + \dots + A_{m1}B_{m1}) & & \\ & \ddots & \\ & & (A_{1n}B_{1n} + \dots + A_{mn}B_{mn}) \end{bmatrix}$

$C_{11} = A_{11}B_{11} + \ldots + A_{m1}B_{n1}$ $\qquad \sum_{j=1}^{n} A_{ij}B_{ij}$

$C_{22} = A_{12}B_{12} + \ldots + A_{m2}B_{n2}$ $\qquad$ 1 to m

$\vdots$

$C_{nn} = A_{1n}B_{1n} + \ldots A_{mn}B_{mn}$ $\qquad \sum_{j=1}^{n} A_{nj}B_{nj}$

$$\boxed{tr(C) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}B_{ij}}$$

(b) The number $\mathbf{tr}(A^T B)$ is sometimes referred to as the inner product of the matrices A and B. (This allows us to extend concepts like angle to matrices.) Show that $\mathbf{tr}(A^T B) = \mathbf{tr}(B^T A)$.

from part A: $\sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}B_{ij}$

$$D = B^T A = \begin{bmatrix} B_{11} & \cdots & B_{m1} \\ \vdots & \ddots & \vdots \\ B_{1n} & \cdots & B_{mn} \end{bmatrix} \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix} = \begin{bmatrix} (B_{11}A_{11} + \ldots + B_{m1}A_{m1}) & & \\ & \ddots & \\ & & (B_{1n}A_{1n} + \ldots + B_{mn}A_{mn}) \end{bmatrix}$$

$D_{11} = B_{11}A_{11} + \ldots + B_{m1}A_{m1}$

$\vdots$

$D_{nn} = B_{1n}A_{1n} + \ldots + B_{mn}A_{mn}$

$$\underbrace{\sum_{j=1}^{n} B_{1j}A_{1j} \quad \cdots \quad \sum_{j=1}^{n} B_{mj}A_{mj}}_{1 \text{ to } m} = \sum_{j=1}^{n} A_{nj}B_{nj}$$

$$\boxed{tr(C) = tr(D) \Rightarrow tr(A^T B) = tr(B^T A)}$$

(d) Show that $\mathbf{tr}(A^T B) = \mathbf{tr}(BA^T)$, even though in general $A^T B$ and BA^T can have different dimensions, and even when they have the same dimensions, they need not be equal.

from part A: $tr(A^T B) = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}b_{ij}$

$$E = BA^T = \begin{bmatrix} B_{11} & \cdots & B_{1n} \\ \vdots & & \vdots \\ B_{m1} & \cdots & B_{mn} \end{bmatrix} \begin{bmatrix} A_{11} & \cdots & A_{m1} \\ \vdots & & \vdots \\ A_{1n} & \cdots & A_{mn} \end{bmatrix} = \begin{bmatrix} (B_{11}A_{11} + \ldots B_{1n}A_{1n}) & & \\ & \ddots & \\ & & (B_{m1}A_{m1} + \ldots + B_{mn}A_{mn}) \end{bmatrix}$$

$m \times n$ $\qquad$ $n \times m$

$$E_{00} = B_{11}A_{11} + \dots + B_{1n}A_{1n} \rightarrow \sum_{j=1}^{n} B_{1j}A_{1j}$$

$$\vdots \qquad \Big\} \; 1 \text{ to } m$$

$$E_{mn} = B_{m1}A_{m1} + \dots + B_{mn}A_{mn} \rightarrow \sum_{j=1}^{n} B_{mj}A_{mj}$$

$$\boxed{tr(E) = \sum_{i=1}^{n}\sum_{j=1}^{n} B_{ij}A_{ij} = tr(BA^T) = tr(A^TB)}$$

2.8 The *Kronecker product* of two $n \times n$ matrices A and B is the $n^2 \times n^2$ matrix

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & & \vdots \\ A_{n1}B & A_{n2}B & \cdots & A_{nn}B \end{bmatrix}.$$

A_{11} A_{12}
A_{21} A_{22}

For example,

$$\begin{bmatrix} 1 & 3 \\ 2 & -1 \end{bmatrix} \otimes \begin{bmatrix} 3 & 4 \\ -5 & 6 \end{bmatrix} = \begin{bmatrix} 3 & 4 & 9 & 12 \\ -5 & 6 & -15 & 18 \\ 6 & 8 & -3 & -4 \\ -10 & 12 & 5 & -6 \end{bmatrix}.$$

Suppose the $n \times n$ matrices A and B, and an n^2-vector x are given. Describe an efficient method for the matrix-vector multiplication

$$y = (A \otimes B)x = \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & & \vdots \\ A_{n1}B & A_{n2}B & \cdots & A_{nn}B \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

(On the right we partitioned x in subvectors x_i of size n.) What is the complexity of your method? How much more efficient is it than a general matrix-vector multiplication of an $n^2 \times n^2$ matrix and an n^2-vector?

General Method:

$$(n^2 \times n^2)\begin{pmatrix} n^2 \\ \times \\ 1 \end{pmatrix} \Rightarrow \left((n^2 \text{ mult}) + (n^2 - 1 \text{ addition})\right) \times n^2$$

$$\boxed{\therefore O(2n^4)}$$

Kronecker Method: $\quad A \otimes B \Rightarrow O(n^3)$

$$A_{11}Bx_1 + \dots + A_{1n}Bx_n$$
$(n \times n)(n \times 1)$

$\Downarrow$

$n(2n-1)$

$\Rightarrow 2n^2 - n + n - 1 \Rightarrow O(2n^2)$

$\uparrow$

do this n times

$$\therefore \boxed{y = (A * B)x \Rightarrow O(2n^3) \quad \text{n times more efficient}}$$

2.10 Consider a product of m matrices

$$A_1 A_2 \cdots A_m, \tag{1}$$

where A_i has size $n_{i-1} \times n_i$. The total number of flops required to compute the result depends on the order in which we evaluate the matrix-matrix products. For example, for $m = 4$, we have the five possibilities

$$A_1(A_2(A_3A_4)) = A_1((A_2A_3)A_4) = (A_1A_2)(A_3A_4) = (A_1(A_2A_3))A_4 = ((A_1A_2)A_3)A_4.$$

In this problem we develop an efficient method for determining the optimal order for the matrix product in (1) and the number of flops if we use the optimal order.

We denote by c_{ij} the cost (optimal number of flops) of computing $A_iA_{i+1} \cdots A_j$, where $1 \le i \le j \le m$. Clearly $c_{11} = c_{22} = \cdots = c_{mm} = 0$, and

$$c_{12} = 2n_0n_1n_2, \qquad c_{23} = 2n_1n_2n_3, \qquad \ldots.$$

(We use the simplification $(2q-1)pr \approx 2pqr$ for a product of a $p \times q$ and a $q \times r$ matrix.) We are interested in computing c_{1m}, the optimal number of flops for the entire product $A_1A_2 \cdots A_m$.

(a) Explain why

$$c_{ij} = \min_{k=i,i+1,\ldots,j-1} (c_{ik} + c_{k+1,j} + 2n_{i-1}n_kn_j).$$

The minimum is over all values of k that satisfy $i \le k < j$. For example, if $i = 1$ and $j = 4$,

$$c_{14} = \min\{c_{11} + c_{24} + 2n_0n_1n_4,\ c_{12} + c_{34} + 2n_0n_2n_4,\ c_{13} + c_{44} + 2n_0n_3n_4\}.$$

C_{ij} is the min cost of multiplying matrices 1 to j. If $C_{1,k}$ is the min cost of multiplying A_1 to A_k & $C_{k+1,j}$ is the min cost of multiplying A_{k+1} to A_j with the cost of multiplying $(A_{1 \ldots k})(A_{k+1 \ldots j}) \Rightarrow$ then we know that is the entire cost of that split. By choosing the min amongst the set, we find the best partition that minimizes the set.

(b) The formula in part (a) suggests a recursive method for computing all the values of c_{ij}. We write the coefficients in a triangular table

$$\begin{array}{cccccc}
c_{11} & c_{12} & c_{13} & \cdots & c_{1,m-1} & c_{1m} \\
 & c_{22} & c_{23} & \cdots & c_{2,m-1} & c_{2m} \\
 & & c_{33} & \cdots & c_{3,m-1} & c_{3m} \\
 & & & \ddots & \vdots & \vdots \\
 & & & & c_{m-1,m-1} & c_{m-1,m} \\
 & & & & & c_{mm}
\end{array}$$

and compute the entries diagonal by diagonal, as follows:

Define $c_{11} = \cdots = c_{mm} = 0$.
for $l = 1, \ldots, m-1$ **do**
  for $i = 1, \ldots, m-l$ **do**
    Compute

$$c_{i,i+l} = \min_{k=i,i+1,\ldots,i+l-1} (c_{ik} + c_{k+1,i+l} + 2n_{i-1}n_k n_{i+l}) \tag{2}$$

  end for
end for

An optimal order can be found by recording for each entry in the table a value of k that attains the

minimum in (2).

Apply this algorithm to find the optimal order and number of flops for the product of four matrices $A_1A_2A_3A_4$ with dimensions

$$100 \times 5000 \times 10000 \times 1000 \times 10.$$

Also compare with the optimal order and the number flops required for the product $A_1A_2A_3$ of the first three matrices.

```python
def main():
    m = 4
    n = [100, 5000, 10000, 1000, 10]
    tab = [[0] * m] * m

    for l in range(m - 1):
        for i in range(m - l):
            currMin = None

            for k in range(i, i + l - 1):
                curr = tab[i][k] + tab[k + 1][i + l] + (2 * n[i - 1] * n[k] * n[i + l])
                if currMin is None:
                    currMin = curr
                else:
                    currMin = min(currMin, curr)

            tab[i][i + l] = currMin

    print(tab)


if __name__ == '__main__':
    main()
```