Data Mining and Machine Learning

Sequence namic Program

Peter Jančovič

Objectives

- To consider data mining for sequential data
- To understand Dynamic Programming (DP)
- Using DP to compute distance between sequences
- To understa
  - An alignment path
  - The DP recurrence equation
  - The distance matrix
  - The accumulated distance matrix
  - The optimal path

Sequences

- Sequences are common in real applications:
  - DNA analysis in bioinformatics and forensic science
    - Sequences of the letters A, G, C and T
  - Signature r
  - Words and
    - Spelling and grammar c or verification,..
  - Speech, music and audio
    - Speech/speaker recognition, speech coding and synthesis
    - Electronic music
  - Radar signature recognition…

Mining sequential data

- Sequences may not be amenable to human interpretation (complexity, dimension, quantity)
- Need for automated sequential data retrieval/mining
- For clusterin… …undamental requirement is for a measu… …istance between two sequences

Basic definitions

- In a typical sequence analysis application we have a basic alphabet consisting of N symbols

$$\mathrm{A} = \{\alpha_1, \ldots, \alpha_n, \ldots, \alpha_N\}$$

- Examples:
  - In text **A** is the set of letters plus punctuation plus 'white space'
  - Bioinformatics $\boldsymbol{A} = \{A, G, C, T\}$ (elements of DNA sequences)

Sequences of continuous variables

- In some applications, elements of a discrete sequence are taken from a continuous vector space, rather than a finite set.
- Sequences o can be dealt with in two ways:
  - Directly
  - Vector quantization (VQ):
    - Represent space as a set of *K* centroids:
    - Replace each data point by its closest centroid

Distance between sequences (1)

- Sequences from the alphabet **A** = {A, B, C, D}
- How similar are the sequences:
  - S_1 = ABC
  - S_2 = ABD
- Intuitively S_2 is obtained from S_1 by deleting C
- Alternatively S_2 is obtained from S_1 by substituting D for C and then deleting D

Distance between sequences (2)

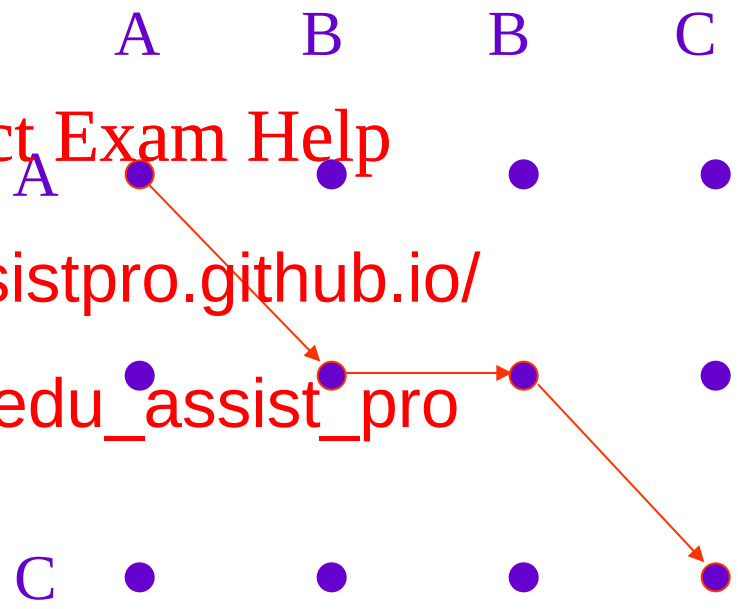
- Or S_2 was obtained from S_1 be deleting ABCD and inserting ABC
- …
- First explana rect', final explanation i Why?
- We favour the simplest ex involving the minimum number of insertions, deletions and substitutions
- ...but maybe not always

Distance between sequences (3)

- Consider:
  - S_1 = AABC
  - S_2 = SABC
  - S_3 = PABC
  - S_4 = ASCB
- If these sequ $_2$ is closer to S_1 than S_3 is, bec $_2$ are adjacent on a keyboard
- Similarly S_4 is close to S_2 because letter-swapping (SA $\rightarrow$AS etc) is a common typographical error

Alignments

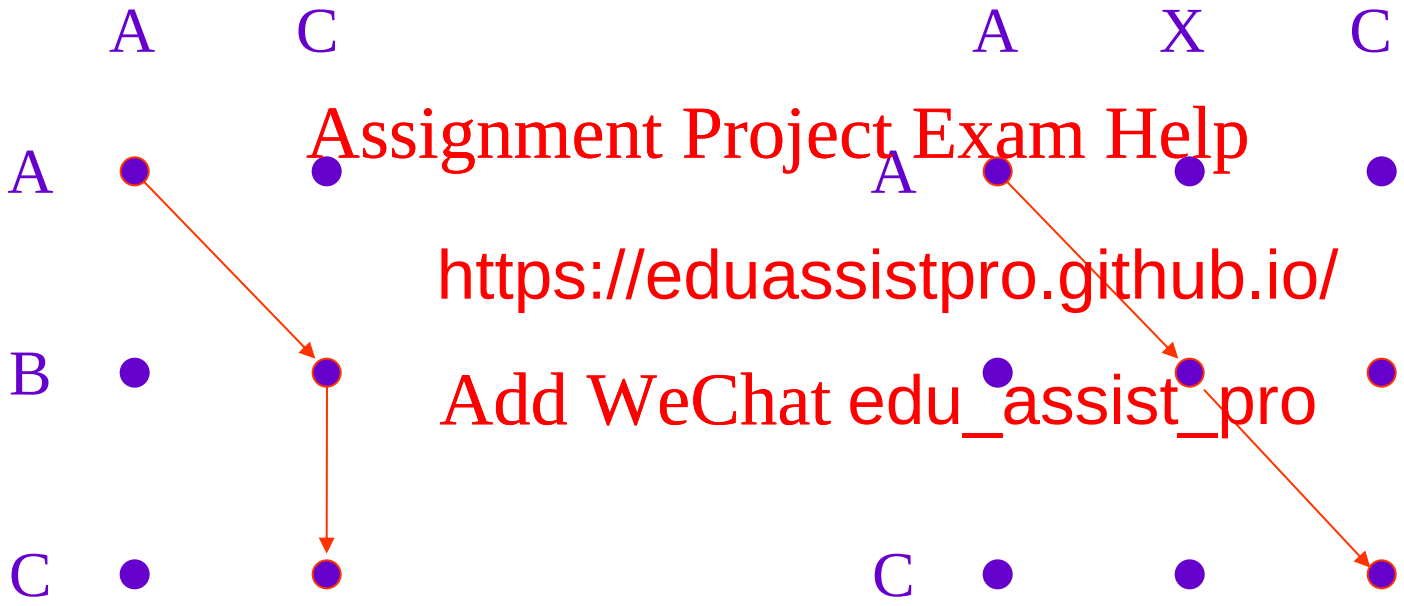
- Relationship between two sequences can be expressed as an alignment be their elemen
- Insertion (w.r.t. ABC) is a horizontal step



Alignment: deletion and substitution

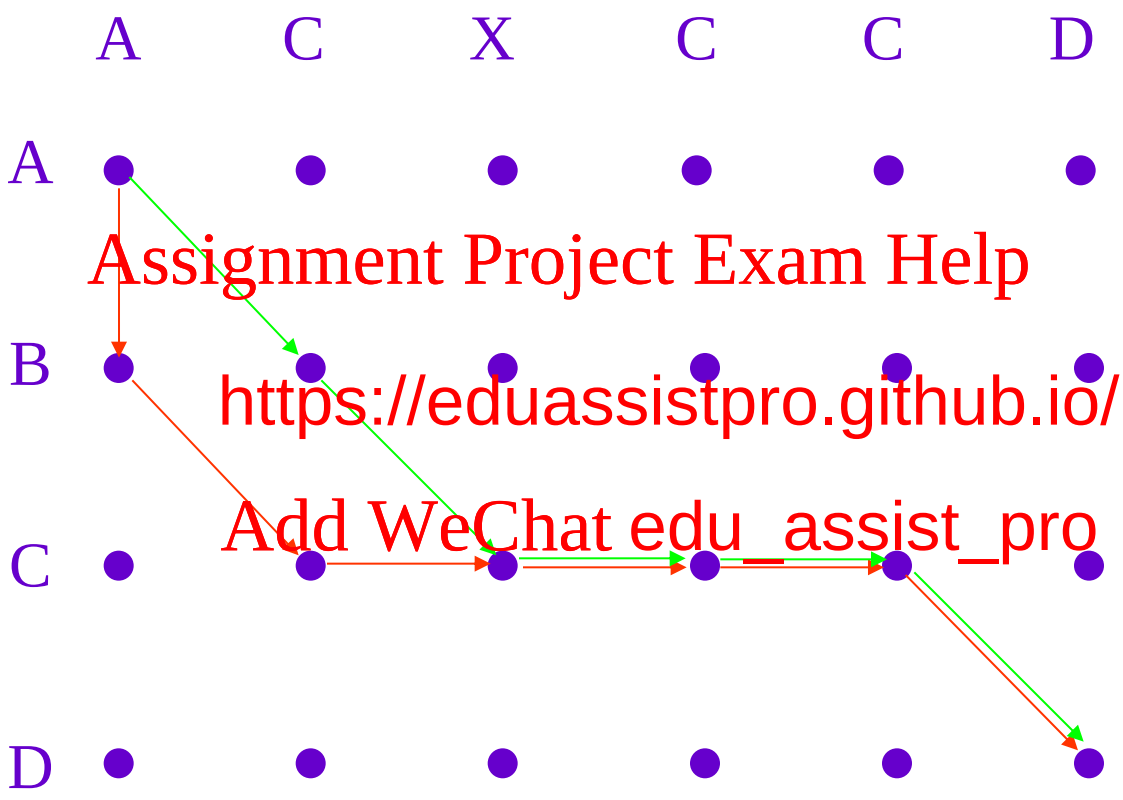
Deletion is a vertical step

Substitution or perfect alignment are diagonal steps



N.B: Edits described relative to vertical string

Alternative alignment paths



Which alignment path is best?

The Distance Matrix

- Let *d* be a metric, so *d*(A,B) is the distance between the alphabet symbols A and B
- Examples:
  - *d*(A,B) = 0 *d*(A,B) = 1
  - In typing, *d*(A,B) is how unlikely it is that A would be mist
  - For continuous valued sequences *d* could be Euclidean distance, or City Block distance, or L_∞ distance

Notation

- Suppose we have an alphabet:

$$\mathrm{A} = \{\alpha_1, \ldots, \alpha_n, \ldots, \alpha_N\}$$

- The distance N matrix

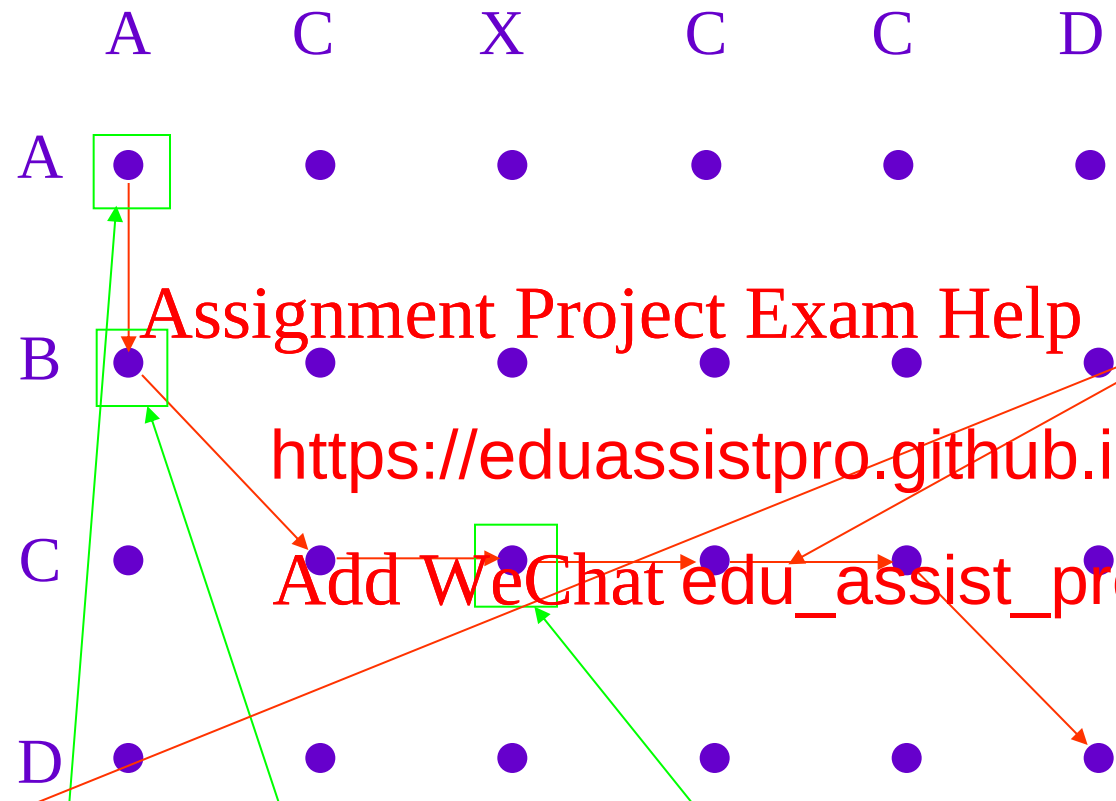
$$D = [D_{m,n}], \; 1 \le m, n \le N$$

where $D_{m,n} = d(\alpha_m, \alpha_n)$ is the distance between the m^{th} and n^{th} alphabet symbols

The Accumulated Distance

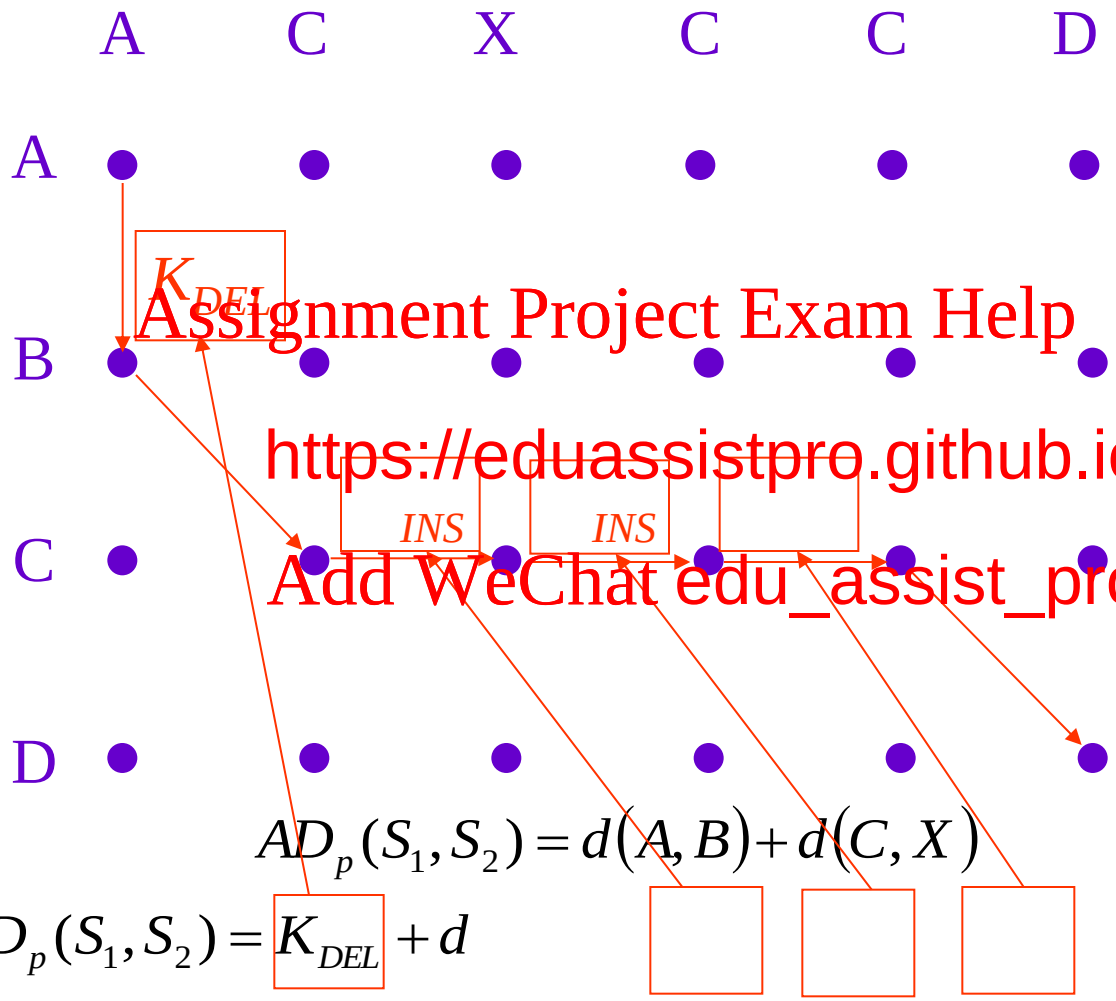
- Consider two sequences:
  - S_1 = ABCD
  - S_2 = ACX
- For an align $_1$ and S_2 the accumulated distance betw S_2, denoted by $AD_p(S_1,S_2)$, is the sum ove des of p of the corresponding distances between elements of S_1 and S_2

Accumulated distance along *p*



$$AD_p(S_1, S_2) = d(A,A) + d(A,B) + d(C,C) + d(C,X) + d(C,C) + d(C,C) + d(D,D)$$

Accumulated distance (continued)



$$AD_p(S_1, S_2) = d(A, B) + d(C, X)$$

$$AD_p(S_1, S_2) = K_{DEL} + d \qquad (C, X)$$

Optimal path and DP distance

- Optimal path is path with minimum accumulated distance
- Formally the optimal path is $\hat{p}$

  where:

$$\hat{p} = \arg\min_{p} AD_p(S_1, S_2), \text{ or } AD_{\hat{p}}(S_1, S_2) = \min_{p} AD_p(S_1, S_2)$$

- The DP distance, or accumulated distance $AD(S_1,S_2)$ between S_1 and S_2 is given by:

$$AD(S_1, S_2) = AD_{\hat{p}}(S_1, S_2)$$

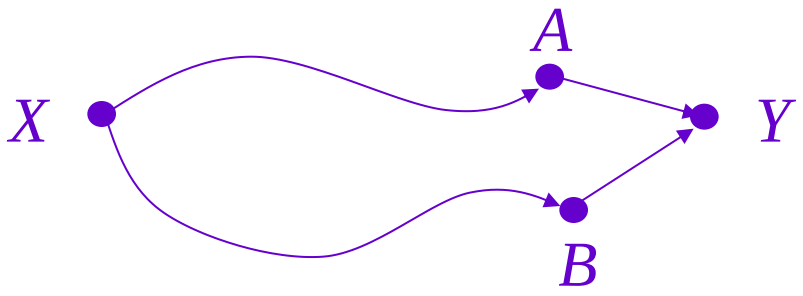
Calculating the optimal path

- Given
  - the distance matrix D,
  - the insertion penalty K_{INS}, and
  - the deleti
- How can we compute the th between two (potentially very long) seq and S_2?

*If K_{DEL} and K_{INS} are not defined you should assume that they are zero

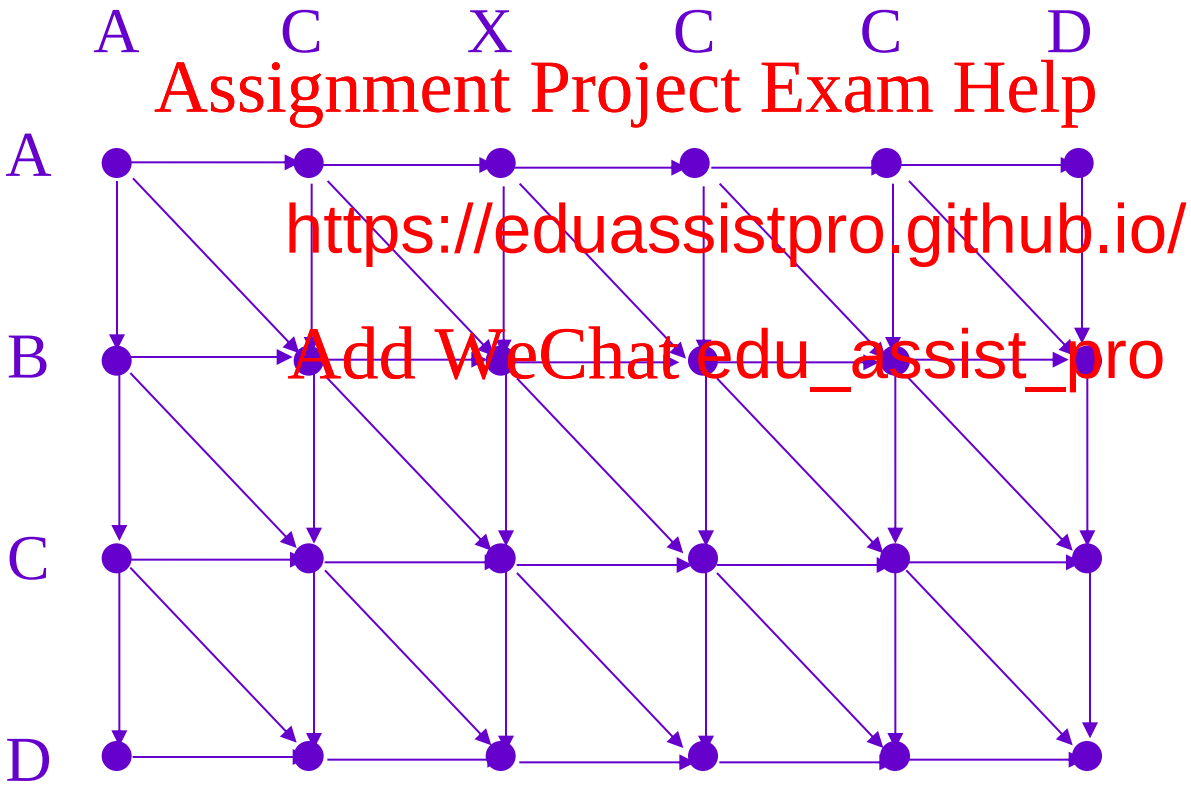
Dynamic Programming (DP)

- Optimal path calculated using Dynamic Programming (DP), based on principle of optimality
- If paths from X to Y go through A or B immediately before Y, optimal path is best of:
  - Best path from X to A plus cost to go from A to Y
  - Best path from X to B plus cost to go from B to Y



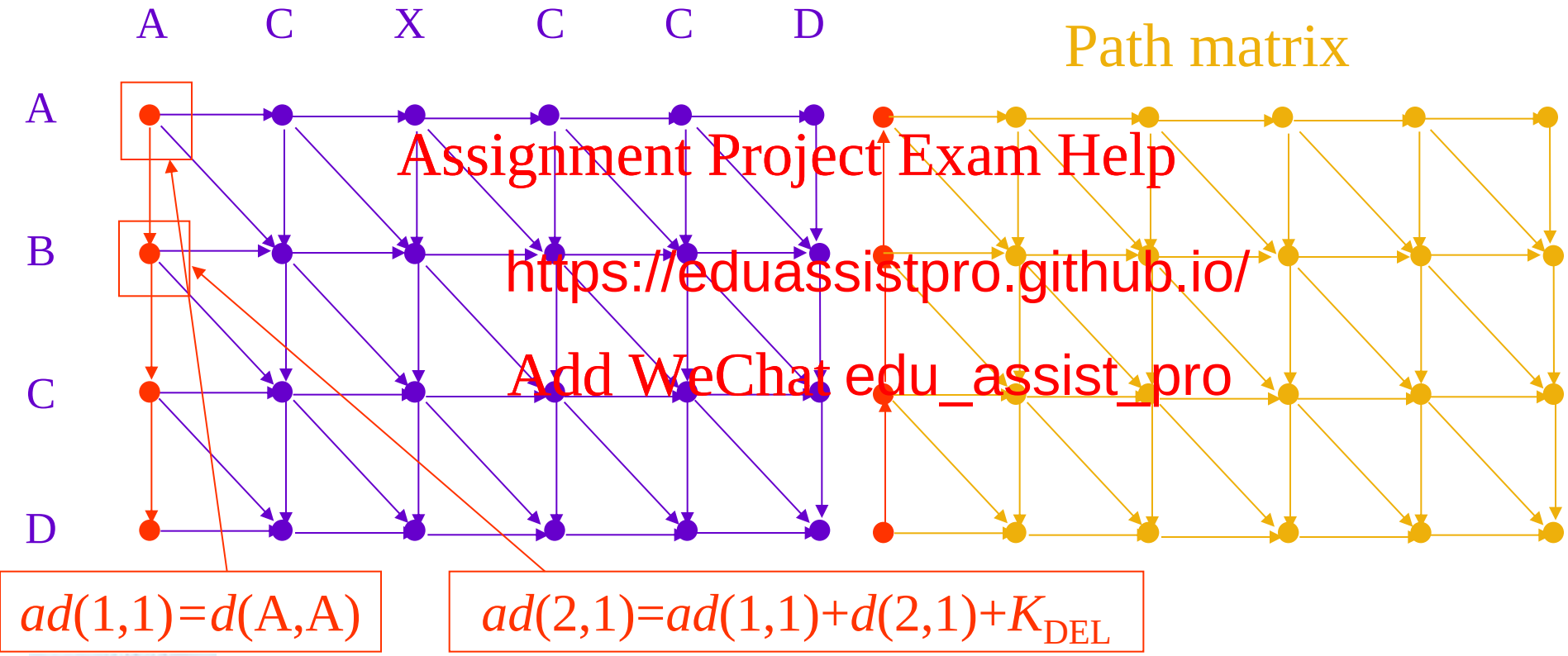
DP – step 1

- Step 1: draw the trellis of all possible paths



DP – forward pass – initialisation

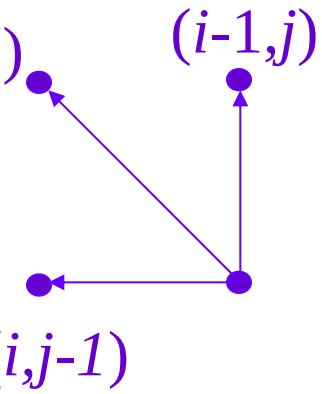
Accumulated distance matrix



$ad(i,j)$

- $ad(i,j)$ is the sum of distances along the best (partial) path from (1,1) to (i,j)
- Calculated using the principle of optimality

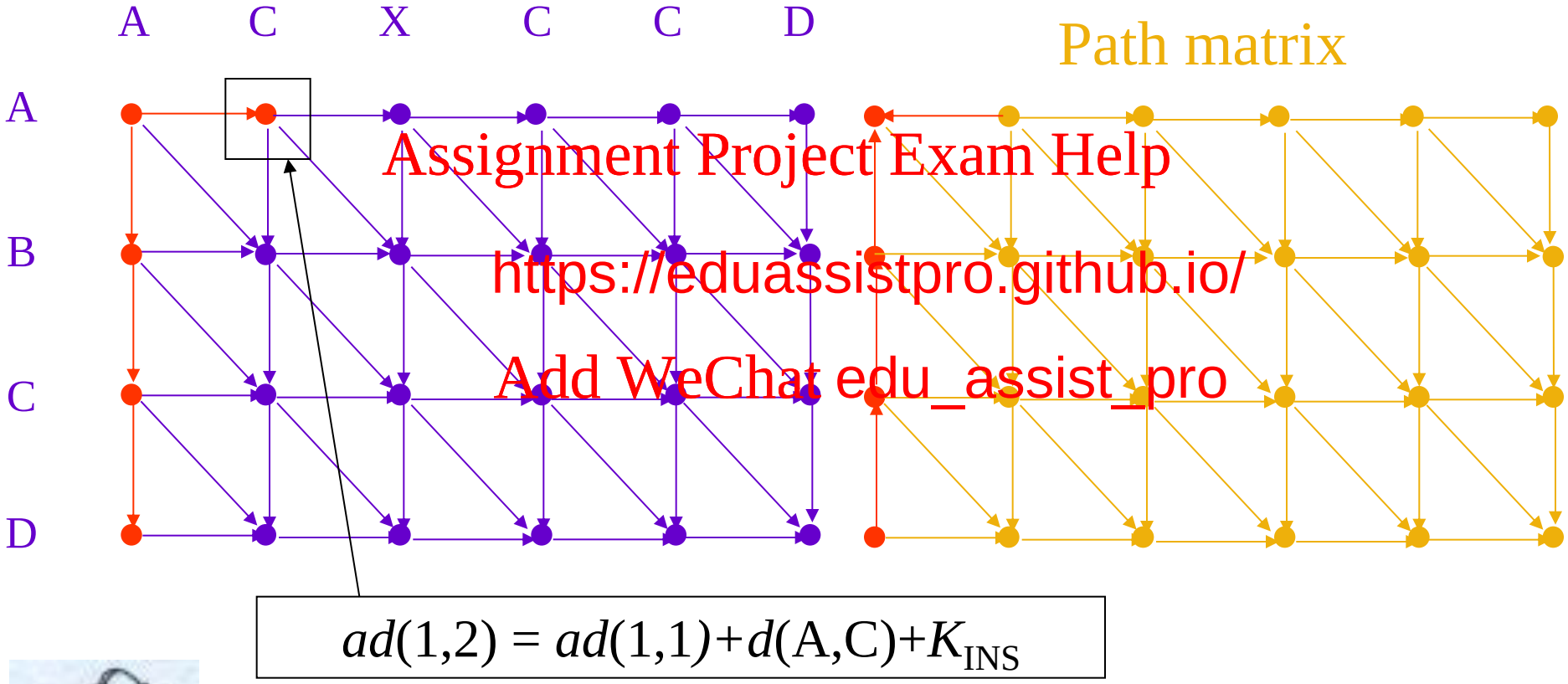
$$ad(i,j) = \min \begin{cases} ad(i-1,j) + K_{DEL} + d(i,j) \\ ad(i-1,j-1) + d(i,j) \\ ad(\end{cases}$$



- Forward path matrix records local optimal paths

DP – forward pass – continued

Accumulated distance matrix



$$ad(1,2) = ad(1,1)+d(\mathrm{A},\mathrm{C})+K_{\mathrm{INS}}$$

DP – forward pass – continued

Accumulated distance matrix

A C X C C D

Path matrix

A

B

C

D

$$ad(3,3) = \min \begin{cases} ad(2,3) + K_{DEL} + d(C, X) \\ ad(2,2) + d(C, X) \\ ad(3,2) + K_{INS} + d(C, X) \end{cases}$$

DP – forward pass – continued

Accumulated distance matrix

A C X C C D

Path matrix

A

B

C

D

$$AD(S_1, S_2) = ad(4,6) = \min \begin{cases} ad(3,6) + K_{DEL} + d(D,D) \\ ad(3,5) + d(D,D) \\ ad(4,5) + K_{INS} + d(D,D) \end{cases}$$

DP – forward pass – continued

Accumulated distance matrix

A C X C C D

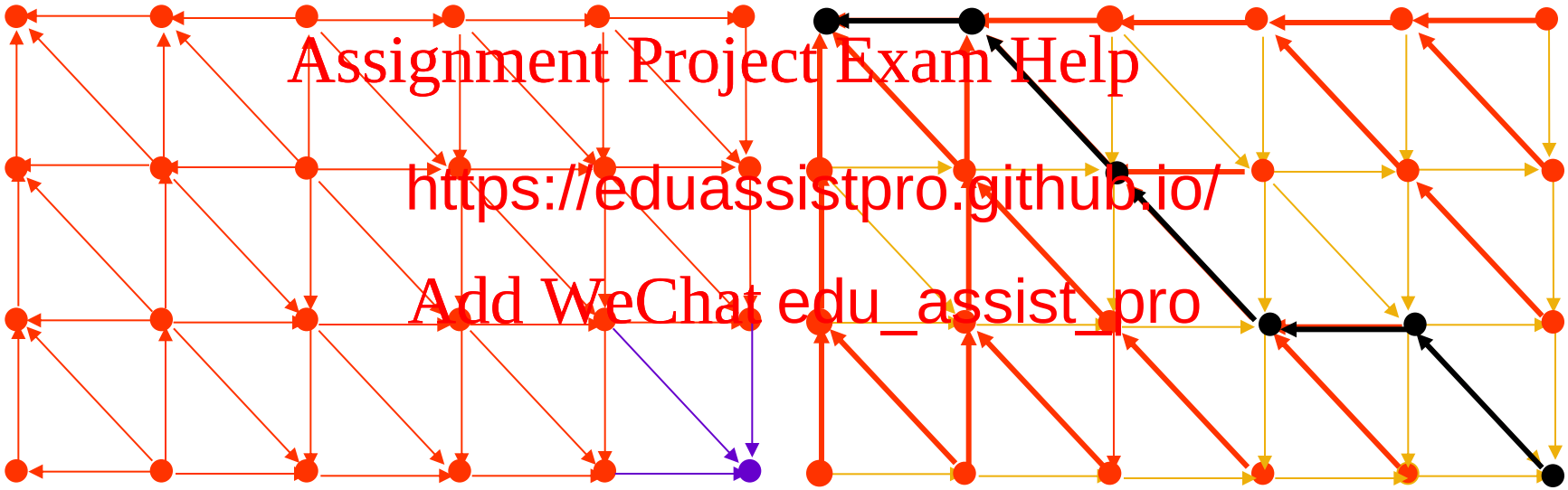
Path matrix

A

B

C

D



Optimal path obtained by tracing back through path matrix, starting at the bottom right-hand corner

Summary

- Introduction to sequence analysis
- Dynamic Programming (DP) and the principle of optimality
- Computing t ce using DP
  - Distance matrix, Accu tance matrix, Path matrix, and Optimal path
- Recovering the optimal path