# Sequences and Time Series

## Edit Distance

K. Selçuk Candan, Professor of Computer Science and Engineering

CASCADE
ASU Center for Assured and SCAlable Data Engineering

ASU Ira A. Fulton Schools of Engineering
Arizona State University

# Strings, sequences, time series

- A *string* or *sequence*, $S = (c_1, c_2, \ldots, c_N)$, *is a finite sequence of symbols.*

abcbbbaabbaabcbbbaaabbc

- Prefix search:
  - Find all strings that start with “tab”:
    - “table”; “tabular”; “tablet”; ….
- Subsequence search:
  - Find all strings that contain the subsequence “ark”:
    - “marketing”; “spark”; “quark”
  - Find all occurrences of “acd”:
    - “aabacdcdabdcababdacddcab.”
- Sequence similarity:
  - “table” vs. “cable”?
  - “table” vs. “tale”?
  - “table” vs. “tackle”?

# Approximate string match

- Sequence distance/similarity:
  - “table” vs. “cable”?
  - “table” vs. “bale”?

- Edit distance:
  - “table” vs. “cable”: 1 (replace “t” with “c”)
  - “table” vs. “bale”: 3 (delete “t”; replace “a” and “b ”; replace “b” and “a”)
- Common edit operations
  - Replacement:
  - a ->b
  - Deletion:
  - a ->λ
  - Insertion:
  - λ ->a

# Edit cost

- Let E be a sequence of edit operations to convert one string to another
- Let us associate a cost, C, to each edit operation

- Costs of edit operations can be different from each other
  - Type of the operation (replace, delete, insert)
  - Symbols involved in the operation
  - Position of the edit operation

- Given a sequence of edit operations, E

$$C(E) = \sum_{e_i \in E} C(e_i)$$

- Edit Distance:

$$D(String_1, String_2) = \min_{E \text{ takes String}_1 \text{ to String}_2} \{C(E)\}$$

# Edit distance

- Let us be given two strings, P and Q, of lengths N and M
- Let us assume that all edit operations have cost = 1

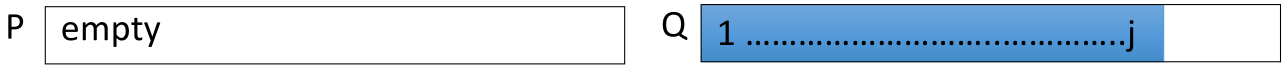
D[i,j] = # of edits from length-i prefix of P to length-j prefix of Q

# Edit distance

- Let us be given two strings, P and Q, of lengths N and M
- Let us assume that all edit operations have cost = 1

D[i,j] = # of edits from length-i prefix of P to length-j prefix of Q

- D[0,j] = j

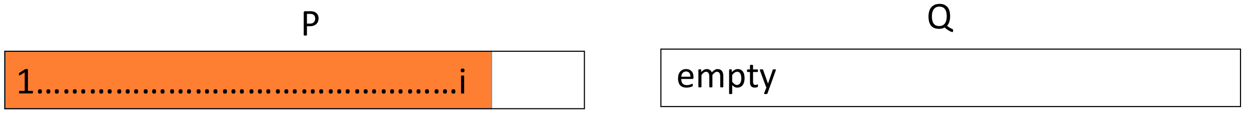


# Edit distance

- Let us be given two strings, P and Q, of lengths N and M
- Let us assume that all edit operations have cost = 1

D[i,j] = # of edits from length-i prefix of P to length-j prefix of Q

- D[0,j] = j
- D[i,0] = i



P

1……………………………………………i

Q

empty

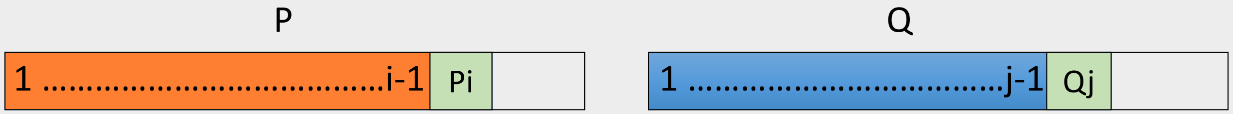
# Edit distance

- Let us be given two strings, P and Q, of lengths N and M
- Let us assume that all edit operations have cost = 1

D[i,j] = # of edits from length-i prefix of P to length-j prefix of Q

- D[0,j] = j
- D[i,0] = i
- if($P_i$=$Q_j$)
  - D[i,j] = D[i-1,j-1]

for all (i<=N) and (j <= M)


P
1 ......................................i-1
Pi
Q
1 ....................................j-1
Qj

# Edit distance

- Let us be given two strings, P and Q, of lengths N and M
- Let us assume that all edit operations have cost = 1

D[i,j] = # of edits from length-i prefix of P to length-j prefix of Q

- D[0,j] = j
- D[i,0] = i
- if($P_i$=$Q_j$) D[i,j] = D[i-1,j-1]

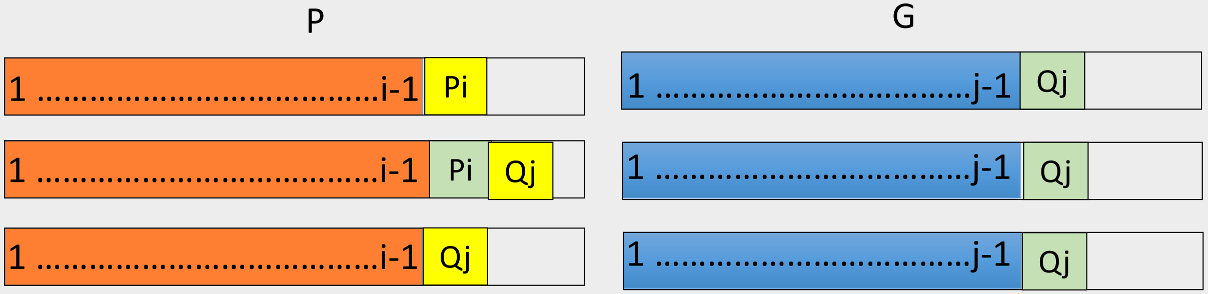
for all (i<=N) and (j <= M)

else D[i,j] = 1 + min{

delete Pi D[i-1,j] ,

insert Qj D[i,j-1] ,

replace Pi with Qj D[i-1,j-1] }


P
G
1 ......................................i-1
Pi
1 .....................................j-1
Qj
1 ......................................i-1
Pi
Qj
1 .....................................j-1
Qj
1 ......................................i-1
Qj
1 .....................................j-1
Qj

# Edit distance

- Let us be given two strings, P and Q, of lengths N and M
- Let us assume that all edit operations have cost = 1

D[i,j] = Cost of edits from length-i prefix of P to length-j prefix of Q

- D[-1,j] = infinity; D[i,-1] = infinity
- D[0,0] = 0
- if($P_i$=$Q_j$) D[i,j] = D[i-1,j-1] for all (i<=N) and (j <= M)

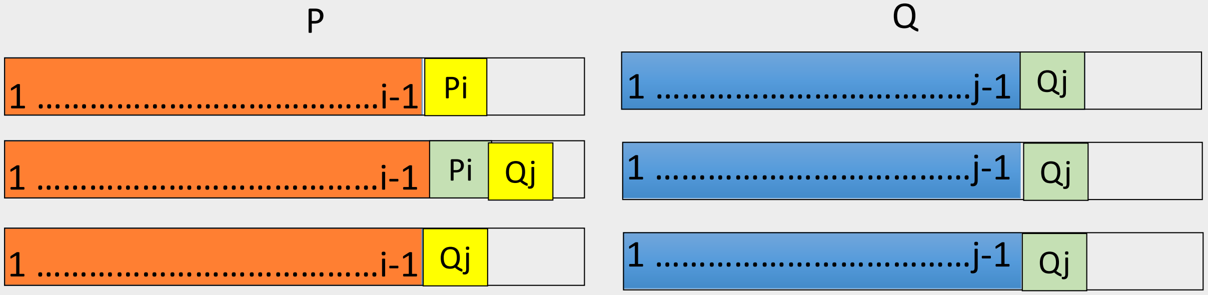
  else D[i,j] = min{

delete Pi $C_{del}(P_i)$ + D[i-1,j] ,

insert Qj $C_{ins}(Q_j)$ + D[i,j-1] ,

replace Pi with Qj $C_{rep}(P_{i,}\ Q_j)$ + D[i-1,j-1]

}


P
Q
1 ....................................i-1
Pi
1 ..................................j-1
Qj
1 ....................................i-1
Pi
Qj
1 ..................................j-1
Qj
1 ....................................i-1
Qj
1 ..................................j-1
Qj

# Edit distance

- Let us be given two strings, P and Q, of lengths N and M
- Let us assume that all edit operations have cost = 1

O(N*M)

D[i,j] = Cost of edits from length-i prefix of P to length-j prefix of Q

- D[-1,j] = infinity; D[i,-1] = infinity
- D[0,0] = 0
- if($P_i$=$Q_j$) D[i,j] = D[i-1,j-1]

  for all (i<=N) and (j <= M)

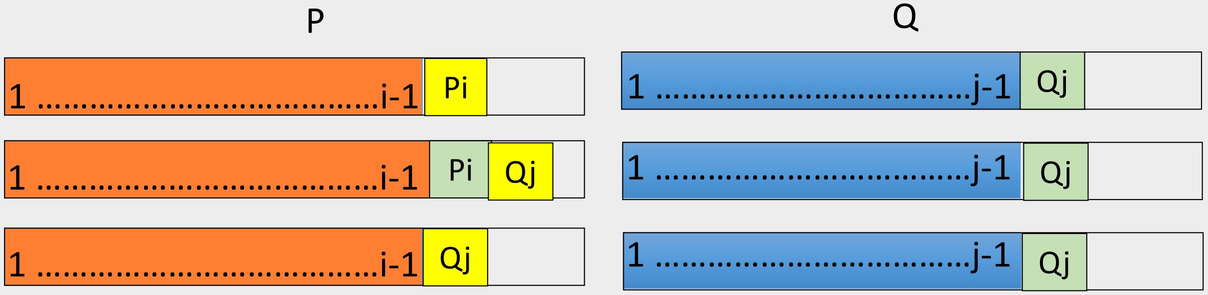
  else D[i,j] = min{

delete Pi $C_{del}(P_i)$ + D[i-1,j] ,

insert Qj $C_{ins}(Q_j)$ + D[i,j-1] ,

replace Pi with Qj $C_{rep}(P_{i,}\ Q_j)$ + D[i-1,j-1]

}


P
Q
1 ......................................i-1
Pi
1 .....................................j-1
Qj
1 ......................................i-1
Pi
Qj
1 .....................................j-1
Qj
1 ......................................i-1
Qj
1 .....................................j-1
Qj

# Summary

- Edit distance can be used to assess how similar or different two strings are

- Problem: Edit distance can be costly for matching long strings.