EDIT DISTANCE

Titolo nota 26/02/2016

trovare un <u>ALLINEAMENTO OTTIMO</u> fra due sequenze X e Y

ogni carattere o "spazio" di X si fa corrispondere ad un carattere o spazio in Y.

X = ALBERO Y = LABBRO

-	A	L	B	E	R	O
L	A	B	B	-	R	O
+		+		+		

⇝ distanza 3

A	L	B	E	R	O
L	A	B	B	R	O
+1	+1	0	+1	0	0

⇝ distanza 3

A	L	B	E	R	O	-	-	-	-	-	-
-	-	-	-	-	-	L	A	B	B	R	O
+	+	+	+	+	+	+	+	+	+	+	+

⇝ distanza 12

Allineamento ⟶ distanza (somma delle distanze delle coppie di caratteri/spazi allineati)

Calcolo della distanza tra caratteri/spazi:

"MATCH" (caratteri uguali) $\rightsquigarrow$ distanza $= 0$

"MISMATCH" (caratteri $\neq$ tra loro, e $\neq$ dallo spazio) $\rightsquigarrow$ distanza $= 1$.

"SPACE" (spazio - carattere o viceversa) $\rightsquigarrow$ distanza $= 1$

Problema trovare l'allineamento ottimo, che minimizza la distanza tra le due sequenze

Titolo nota 26/02/2016

Allineamento → sequenza di operazioni che permettono di trasformare X in Y

- sostituzione di un carattere con un altro (costo 1)
- cancellazione di un carattere (costo 1)
- inserimento di un carattere (costo 1)

X: A L - B E R O
Y: - L A B B R O

$\cancel{A}$ L B $\cancel{E}$ R O

- cancella A +1
- mantieni L
- inserisci A +1
- mantieni B
- sostituisci E con B +1
- mantieni R
- mantieni O

costo 3

Titolo nota 26/02/2016

Algoritmo di PD (x EDIT DISTANCE / ALLINEAMENTO OTTIMO)

① Definizione dei sottoproblemi

$X = x_1 x_2 x_3 \dots x_n$

$Y = y_1 y_2 y_3 \dots y_m$

sottoproblema i-j : trovare la edit-distance tra i prefissi X_i e Y_j $0 \le i \le n$, $0 \le j \le m$

$X_i = x_1 x_2 \dots x_i$

$Y_j = y_1 y_2 \dots y_j$

tabella $(n+1) \times (m+1)$

Titolo nota 26/02/2016

$M(i,j)$ = edit distance tra X_i e Y_j $(n+1) \times (m+1)$

↓
distanza minima.

② Sottoproblemi elementari $i=0$, o $j=0$

X_i Y_0

$x_1 \; x_2 \; x_3 \; \dots \; x_i$

$+ \; + \; + \quad +$

$M[i,0] = i \quad 0 \leq i \leq n$

X_0 Y_j

$y_1 \; y_2 \; \dots \; y_j$

$+ \; + \quad +$

costo j $\Rightarrow$ $M[0,j] = j \quad 0 \leq j \leq m$

y

X	Ø	L	A	B	B	R	O
Ø	0	1	2	3	4	5	6
A	1	1	1	2	3	4	5
L	2	1	2	2	3	4	5
B	3	2	2	2	2	3	4
E	4	3	3	3	3	3	4
R	5	4	4	4	4	3	4
O	6	5	5	5	5	4	3

← ↖ ↖ ↖ ↑ ↖ ↖

\- A L B E R O

L A B B - R O

\- A
L A

↑

\- A - - - -
L A B B R O

\+ + + + +

costo 5

$M(n, m) = 3$

edit-distance 3

↖ ↖ ↖ ↖ ↖ ↖

A L B E R O

L A B B R O

(3)

Titolo nota 26/02/2016

(3) Regola ricorsiva

$$P = \begin{cases} 0 & x_i = y_j \\ 1 & x_i \neq y_j \end{cases}$$

X_i Y_j

$x_1 \ldots x_i$ $y_1 y_2 \ldots y_j$

$\underbrace{X_{i-1} \; Y_{j-1}}$ | $x_i \; y_j$

allineamento ottimo tra i prefissi X_{i-1}, Y_{j-1}

$\boxed{M[i-1, j-1] + P}$

$\underbrace{X_{i-1} \; Y_j}$ | x_i —

allineamento ottimo

$\boxed{M[i-1, j] + 1}$

$\underbrace{X_i \; Y_{j-1}}$ | — y_j

all. ottimo

$\boxed{M[i, j-1] + 1}$

SCELGO IL MINIMO

$$M(i,j) = \begin{cases} 0 & i=0,\ j=0 \\ i & i\neq 0,\ j=0 \\ j & i=0,\ j\neq 0 \\ \min\{ M[i-1,j-1]+P,\ M[i-1,j]+1,\ M[i,j-1]+1 \} & i>0,\ j>0 \end{cases}$$

4 Risultato $M(n,m)$ = edit distance $\cancel{X}_n$ e Y_m (X e Y)

Titolo nota 26/02/2016

ED(X, Y) // X e Y array di dim n e m X[1...n] Y[1...m]

(1) M = nuova matrice (n+1) x (m+1) // M[0...n, 0...m]

(2) for i = 0 to n { M[i, 0] = i }
for j = 1 to m { M[0, j] = j }

(3)
```
for i = 1 to n {
    for j = 1 to m {
        if ( X[i] == Y[j] ) p = 0;
        else p = 1;
        M[i, j] = min { M[i-1, j-1] + p, M[i-1, j] + 1, M[i, j-1] + 1 }
    }
}
```

$\Theta(1)$

(4) return M[n, m]

$T(n, m) = \Theta(n \cdot m)$

Titolo nota 26/02/2016

ALLINEA(X, Y, M) // restituisce ALLX e ALLY

ALLX = nuovo array di dimensione m + n

ALLY = " " " " m + n

h = m + n;

i = n

j = m

while (i > 0 || j > 0) {

if (i > 0 && j > 0 &&

(M[i, j] == M[i-1, j-1] && X[i] == Y[j])

||

(M[i, j] == M[i-1, j-1] + 1 && X[i] ≠ Y[j]))

{ ALLX[h] = X[i];

ALLY[h] = Y[j];

i--; j--;

}

x_i

y_j

```
else if ( j>0 && M[i,j] == M[i,j-1]+1 ) {
        ALLX[h] = -
        ALLY[h] = Y[j]
        j--;
    }
    else { ALLX[h] = X[i];
           ALLY[h] = -;
           i--;
    }
    h--;
  }
}
return h, ALLX, ALLY
```

$\overline{y_j}$

$$\boxed{T(n,m) = O(n+m)}$$

Titolo nota 26/02/2016

M → Elenco operazioni per trasformare X in Y

ELENCO (X, Y, M, i, j)

```
if ( i>0 || j>0 ) {
    if ( i>0 && j>0 && M(i,j)==M(i-1, j-1) && X(i)==Y(j)) {
        ELENCO(X, Y, M, i-1, j-1);
        PRINT "MANTIENI X(i)"
    }
    else if ( i>0 && j>0 && M(i,j)==M(i-1, j-1)+1) {
        ELENCO (X, Y, M, i-1, j-1);
        PRINT "SOSTITUISCI X(i) CON Y(j)"
    }
    else if (j>0 && M(i,j)==M(i, j-1)+1) {
```

ELENCO(X, Y, M, i, j-1);

PRINT "INSERISCI Y[j]";

}

else {

ELENCO(X, Y, M, i-1, j);

PRINT "CANCELLA X(i)";

}

}