Intelligenza Artificiale 2

Tecniche di Text Mining per la Biologia Computazionale

Claudio Biancalana
claudio.biancalana@dia.uniroma3.it
www.dia.uniroma3.it/~biancal

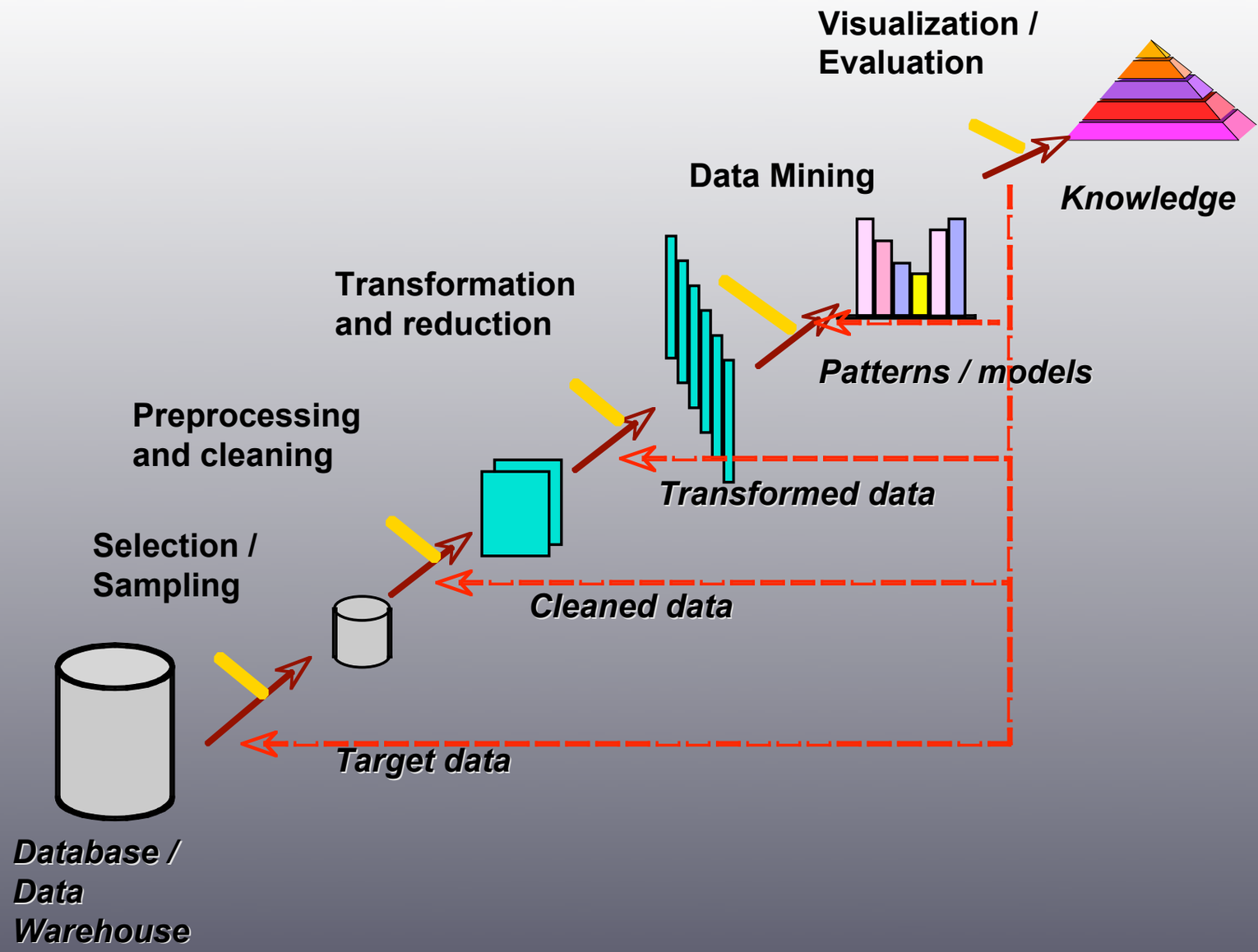
Piano del discorso

- Data mining - Knowledge Discovery in Data
- Splicing - Splice Junction Recognition
- Principi di tecniche di classificazione
- Latent Semantic Analysis
- Hyperspace Analogue to Language
- Il sistema implementato

Data Mining

- Processo di estrazione di conoscenza da banche dati di grandi dimensioni tramite l'applicazione di algoritmi che individuano le associazioni "nascoste" tra le informazioni (pattern) e le rendono visibili.
- I pattern devono essere:
  - Validi
  - Precedentemente sconosciuti
  - Potenzialmente utili
  - Comprensibili

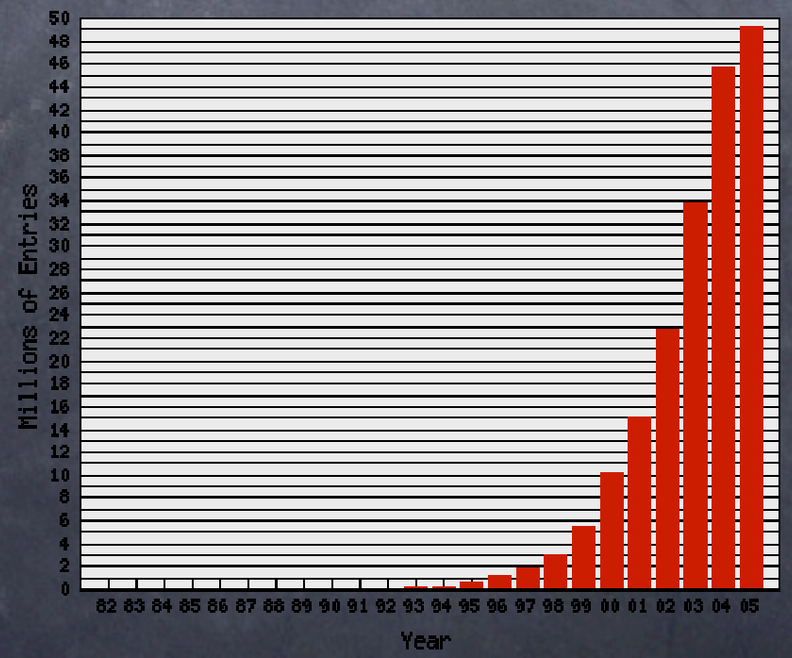
KDD Knowledge Discovery in Data



Perchè utilizzare tecniche di Data Mining?

- La possibilità di sfruttare le potenzialità dei progetti genomici e in generale la mole di informazioni sulle sequenze nucleotidiche è possibile grazie ad Istituzioni che si occpuano di mantenere, gestire e rendere accessibili questi dati alla comunità internazionale.

- Per avere un'idea dell'entità del problema basti considerare che ad oggi la banca EMBL continene circa 49 milioni di sequenze.

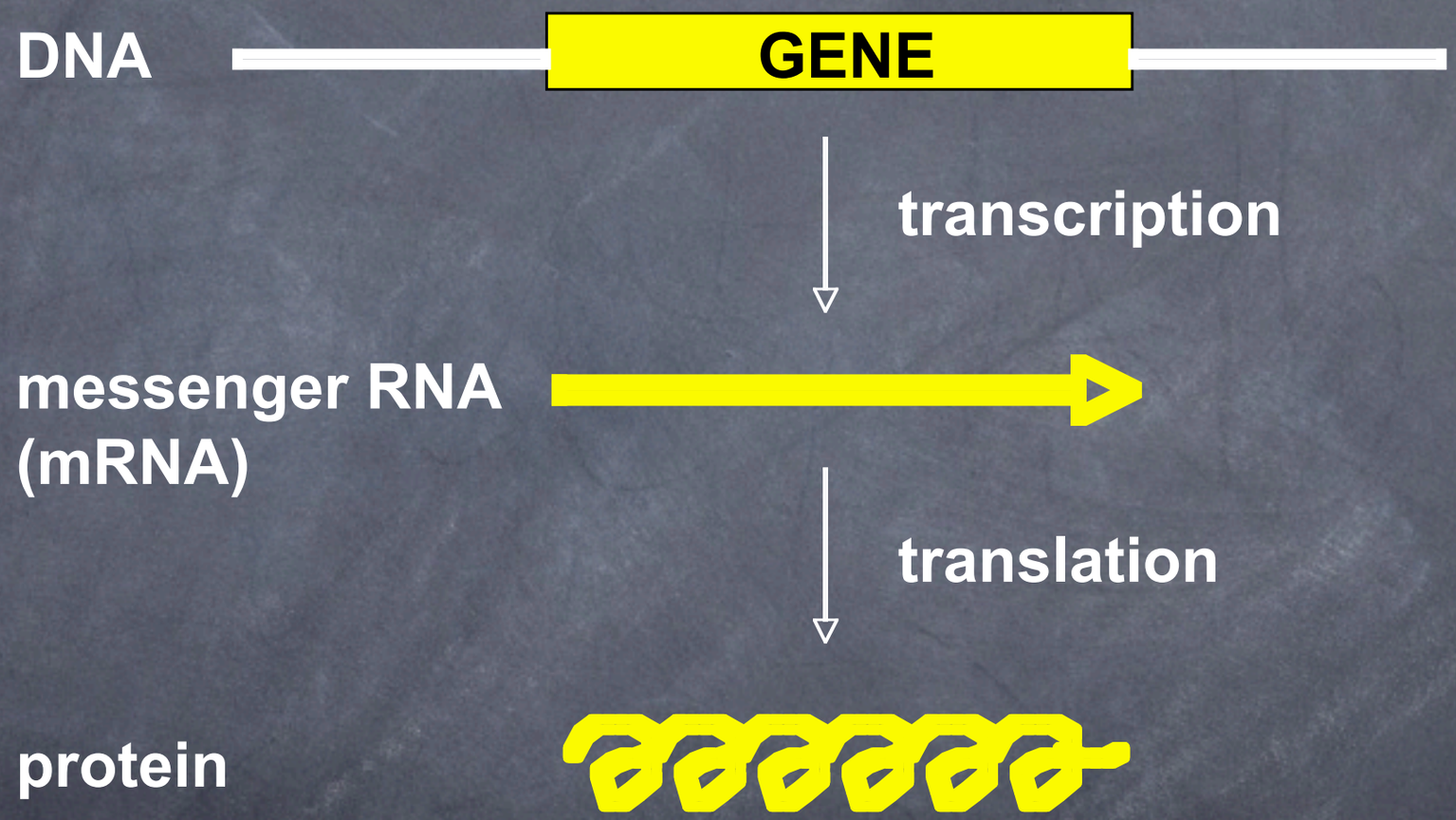


Il problema affrontato... Splice-junction recognition

- Il termine splicing (saldatura) indica, nella lingua inglese, uno dei processi, insieme al capping e alla poliadenilazione, di maturazione del trascritto primario dei geni discontinui.

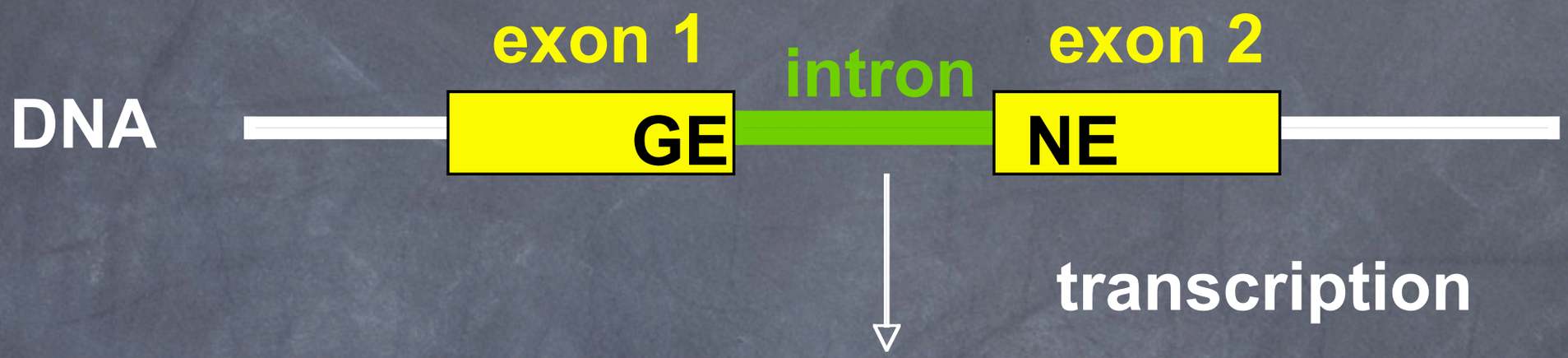
- La maggior parte dei geni eucariotici conta regioni presenti nel mRNA maturo (esoni) e altre non presenti (introni). Alcuni introni sono presenti anche nei geni degli archeobatteri, mentre sono assenti in quelli degli eubatteri. Dopo la trascrizione da parte della RNA polimerasi il trascritto primario va incontro a numerose modificazioni. Prima fra tutte l'eliminazione degli introni, denominata splicing.

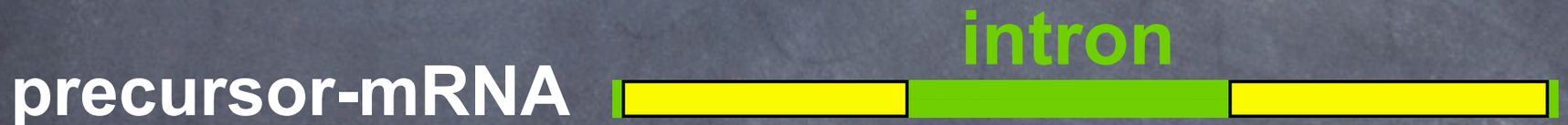
Splicing



Le informazioni genetiche vengono trasferite dai geni alle proteine attraverso l'RNA messaggero

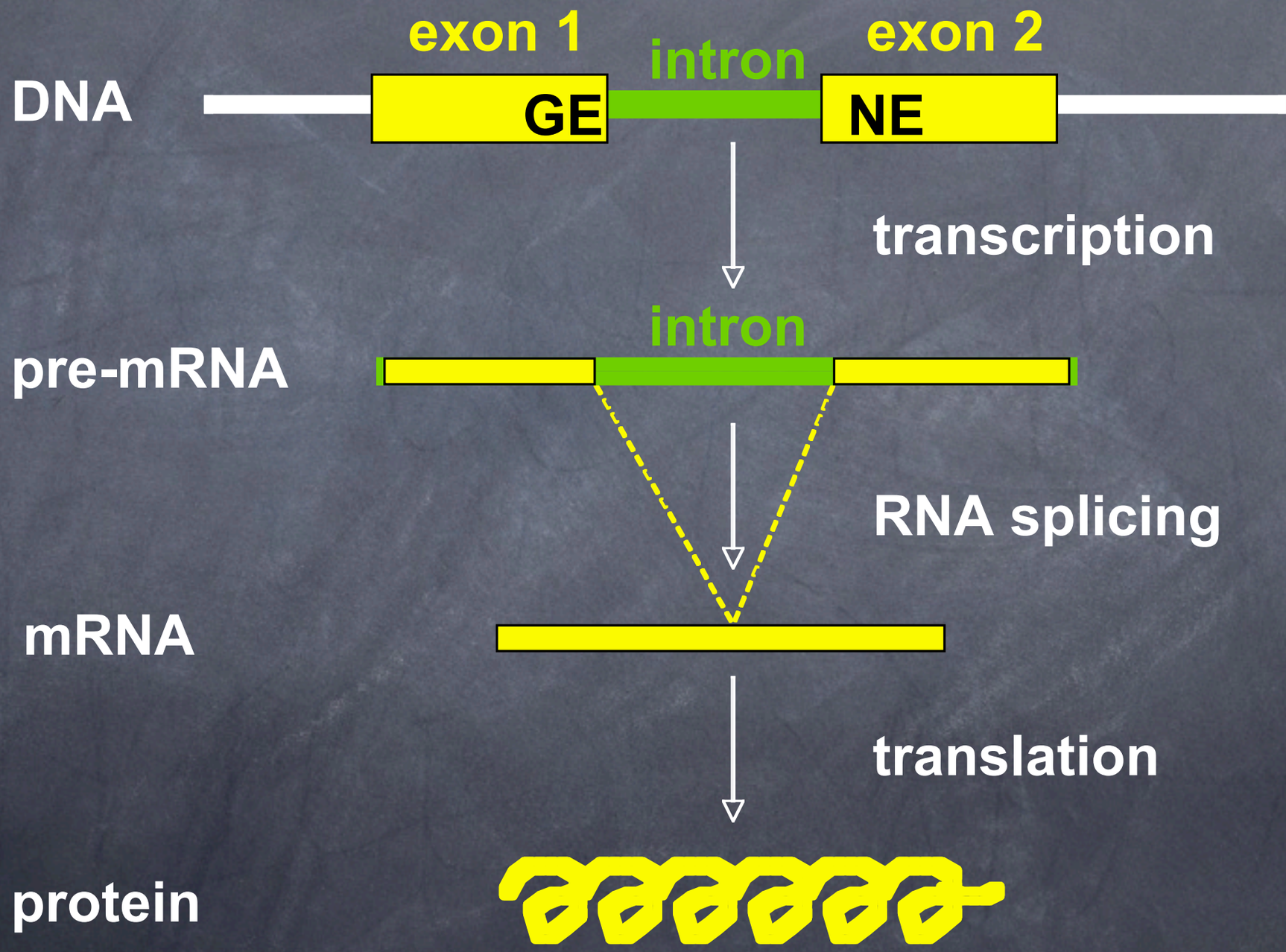
Splicing



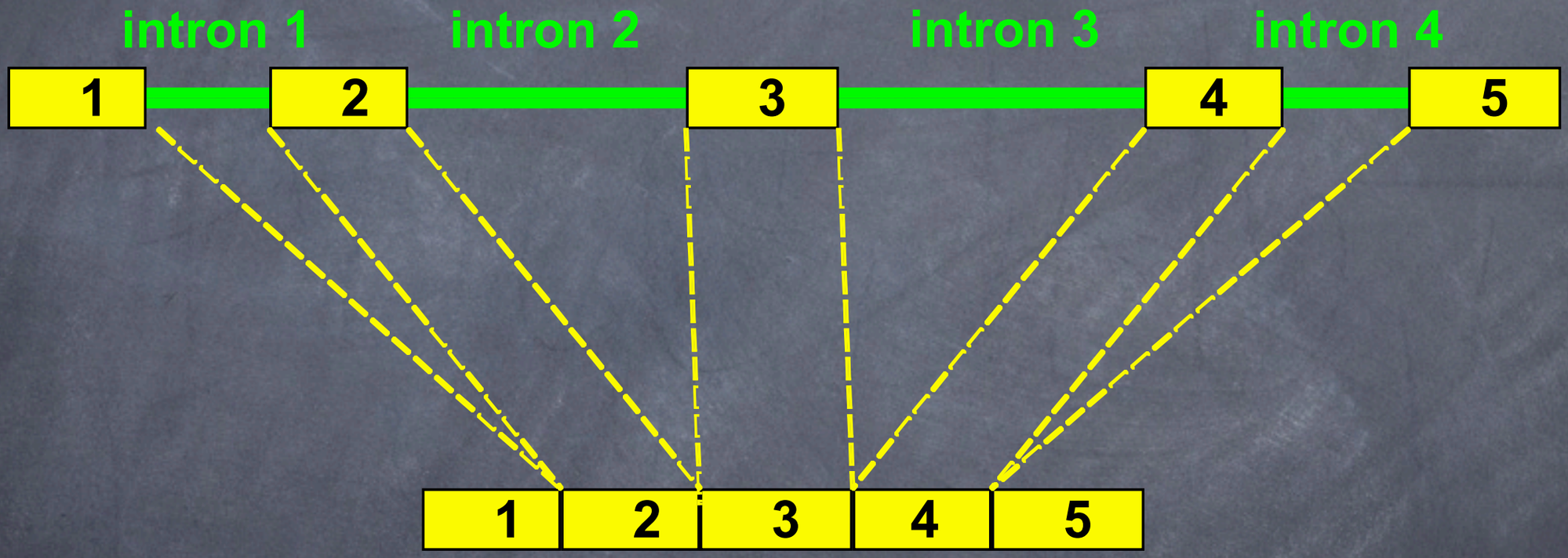


Alcuni geni portano con se informazioni interrotte da sequenze non codificanti chiamati introni (introns). Le sequenze codificanti sono chiamate esoni (exons).

Splicing

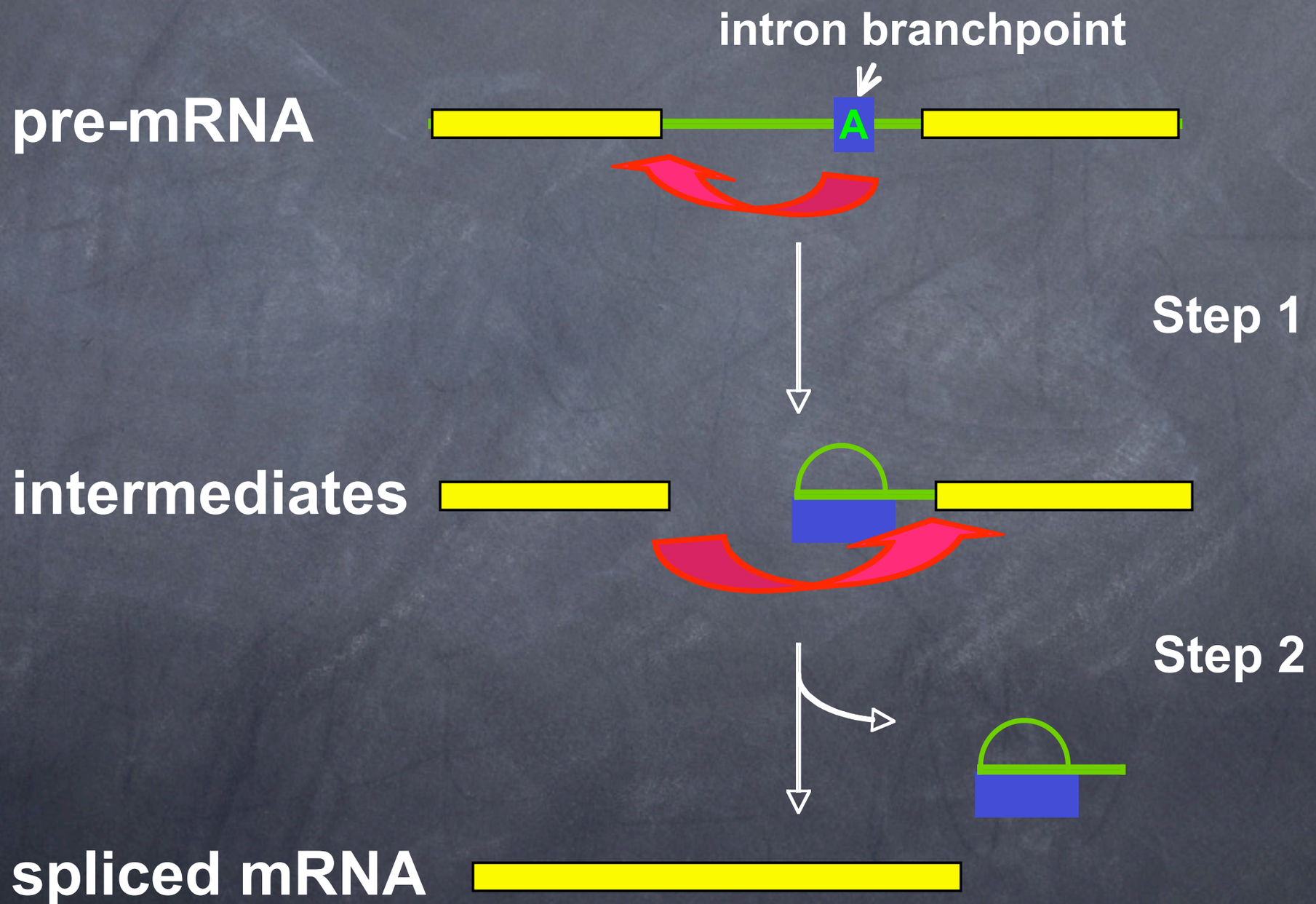

exon 1
intron
exon 2
DNA
GE
NE
transcription
intron
pre-mRNA
RNA splicing
mRNA
translation
protein

Splicing



Nell'uomo, molti geni contengono introni multipli!

Processo FISICO

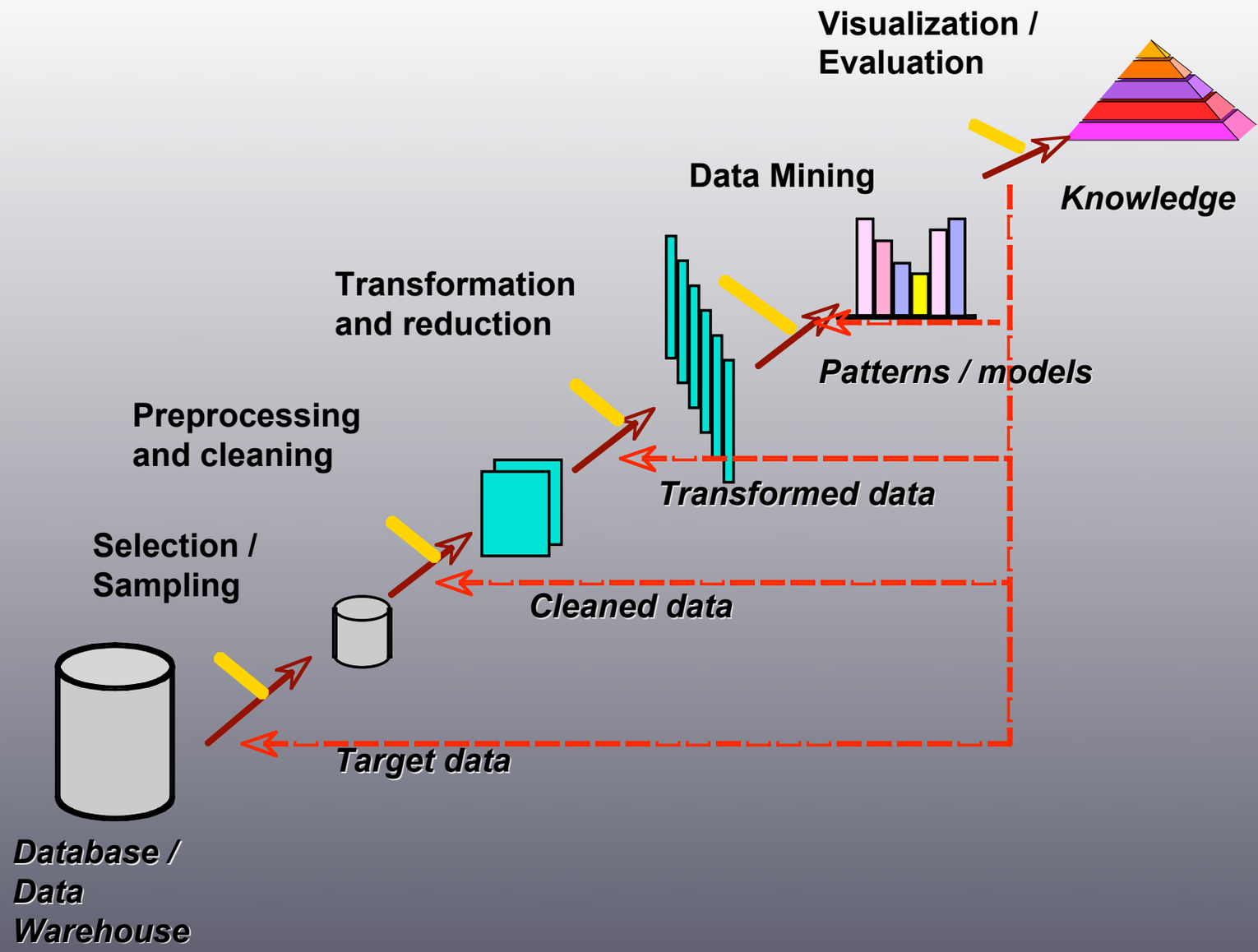


Splicing (saldatura)

EI	ATRINS-DONOR-521	CCAGCTGCATCACAGGAGGCCAGCGAGCAGGTCTGTTCCAAGGGCCTTCGAGCCAGTCTG
EI	ATRINS-DONOR-905	AGACCCGCCGGGAGGCGGAGGACCTGCAGGGTGAGCCCCACCGCCCCTCCGTGCCCCCGC
EI	BABAPOE-DONOR-30	GAGGTGAAGGACGTCCTTCCCCAGGAGCCGGTGAGAAGCGCAGTCGGGGGCACGGGGATG
EI	BABAPOE-DONOR-867	GGGCTGCGTTGCTGGTCACATTCCTGGCAGGTATGGGGCGGGGCTTGCTCGGTTTTCCCC
IE	HUMGHCSA-ACCEPTOR-42394	CTCCCTCTGTTGCCTCCGGTTTCTCCCCAGGCTCCCGGACGTCCCTGCTCCTGGCTTTTG
IE	HUMGHCSA-ACCEPTOR-42765	ATAGACCTTGGTGGGCGGTCCTTCTCCTAGGAAGAAGCCTATATCCTGAAGGAGCAGAAG
IE	HUMGHCSA-ACCEPTOR-42976	GCACAGCCACTGCCGGTCCTTCCCCTGCAGAACCTAGAGCTGCTCCGCATCTCCCTGCTG
IE	HUMGHCSA-ACCEPTOR-43393	GGCCTCTCCTTCTCTTCCTTCACTTTGCAGAGGCTGGAAGATGGCAGCCCCCGGACTGGG

The splice junction recognition problem then involves identifying if a sequence of a fixed size has an intron/exon site (IE), an exon/intron site (EI), or if it does not have a splice site (N).

KDD Knowledge Discovery in Data



Preprocessamento
Allineamento - Edit Distance

A C T G T

A C T T T G T A

$$C_{i,j} = \begin{cases} C_{i-1,j-1} & \text{se } i=0 \text{ o } j=0 \\ C_{i-1,j-1} + 1 & \text{se } a_i = b_j \\ 1 + \text{Max}(C_{i-1,j}, C_{i,j-1}) & \text{altrimenti} \end{cases}$$

Allineamento
Edit Distance

A C T G T **A C T T T G T A**

		A	C	T	T	T	G	T	A
	0	0	0	0	0	0	0	0	0
A	0	1	1	1	1	1	1	1	1
C	0	1	2	2	2	2	2	2	2
T	0	1	2	3	3	3	3	3	3
G	0	1	2	3	3	3	4	4	4
T	0	1	2	3	4	4	4	5	5

Categorizzare...

- Input:
  - Descrizione di una istanza x in X, dove X è l'istanza linguaggio o spazio dell'istanza.
  - Un numero fissato di categorie:
    - C={c1,c2,...,cn}
- Output:
  - La categoria di x: c(x) è una funzione di categorizzazione che ha come dominio X e come codominio C.

Classificazione
Uno contro tutti

```
function classify(sequence) returns sequence class
inputs: sequence as string

if n_classify(sequence) is class N then
      class ← N
else if ei_ie_classify(sequence) is class EI then
      class ← EI
       else
      class ← IE
   end if
end if

return class
```

Valutazione del classificatore

- La valutazione è sperimentale perché il problema non ha una specifica formale che consenta un altro tipo di valutazione
- La valutazione deve essere effettuata su dati di test indipendenti dai dati di training (solitamente insiemi disgiunti di istanze).
- I risultati possono variare in base all'uso di diversi insiemi di training e test.

Valutare un classificatore

Tavola di Contingenza
Matrice di Confusione

	Truth: Yes	**Truth: No**
System: Yes	*a*	*b*
System: No	*c*	*d*

error rate = (b+c)/n
accuracy = 1 - error rate
precision (P) = a/(a+b)
recall (R) = a/(a+c)
break-even = (P+R)/2
F1-measure = 2PR/(P+R)

La matrice di adiacenza

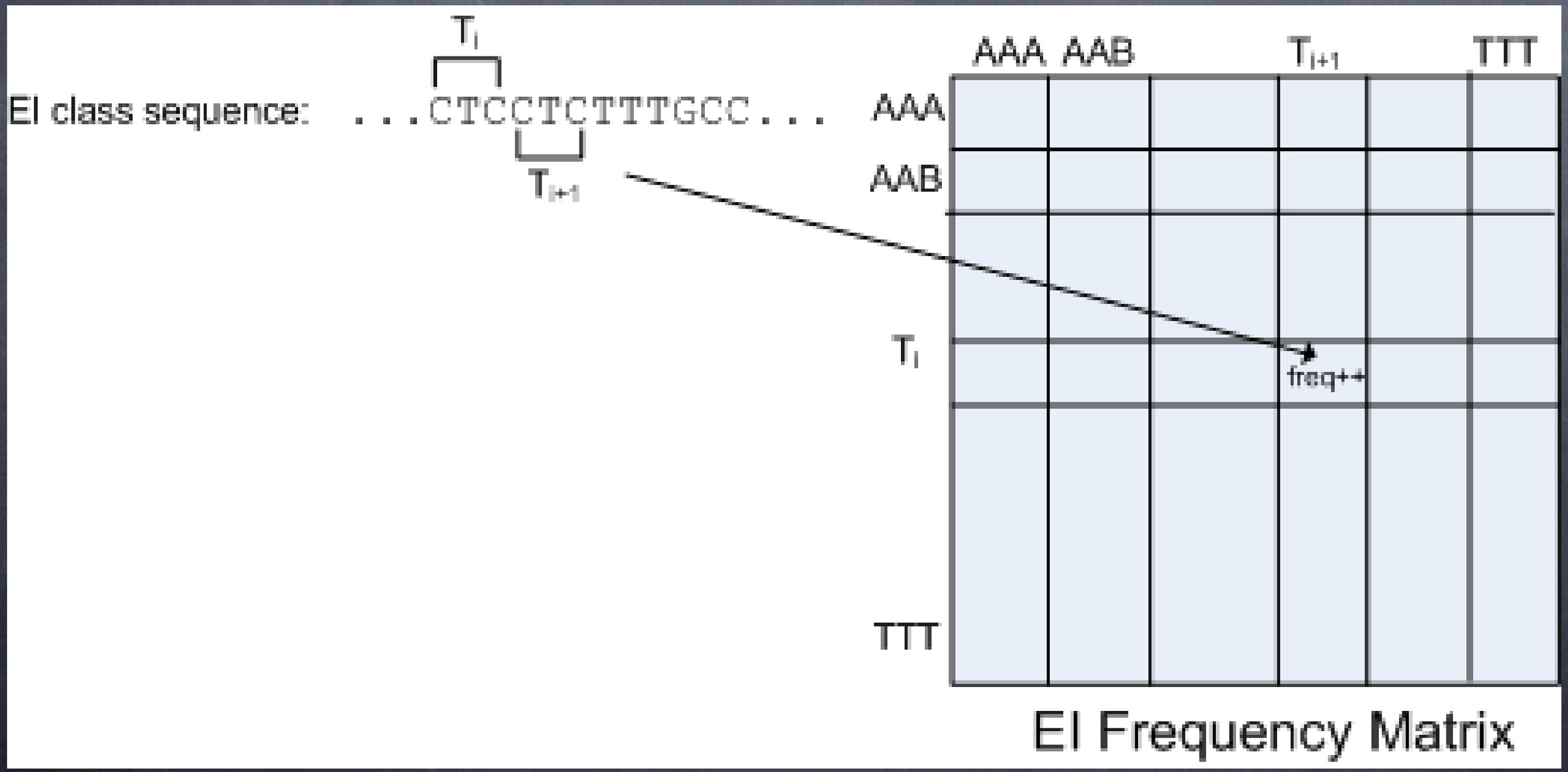
- Dice quali sono i nodi (i soggetti dell'interazione) e con chi sono connessi.
- Permette di costruire il vostro modello.

	A	C	G	T
A	-	1	0	1
C	1	-	0	1
G	0	0	-	1
T	1	1	1	-

Diagonale nulla

Simmetrica

EI Frequency Matrix



Matrice Termine-Documento

Termini	**Documenti**								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	1	1	0	1	0	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	1	0	0	1	0	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

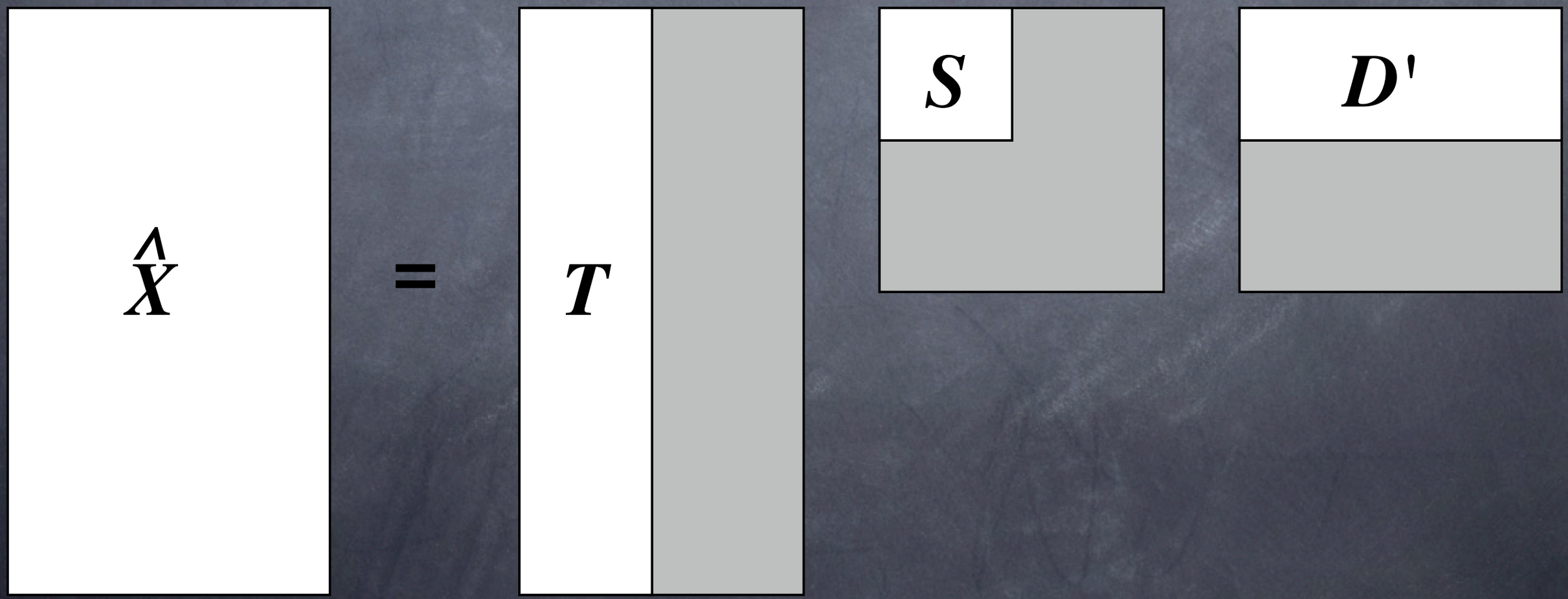
Osservazioni

- La maggioranza delle celle della matrice sono zeri
- La dimensionalità della matrice è elevata (t)
- Con i metodi classici ogni documento è un vettore in uno spazio t-dimensionale
- LSI tenta di proiettare questo spazio un uno spazio di dimensione ridotta, in cui, anziché i termini le dimensioni rappresentano co-occorrenze o domini semantici
- Tutte le possibili co-occorrenze sarebbero assai di più dei termini singoli: ma il metodo della SVD utilizzato da LSI consente di elminare le co-occorrenze non significative

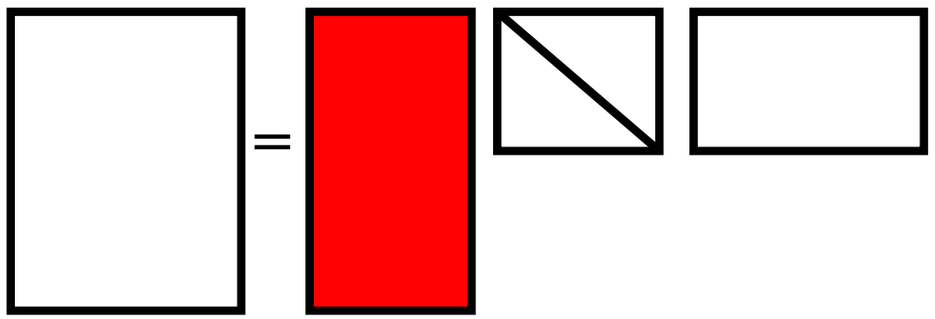
Singular Value Decomposition

- Definisci X coma la matrice, con t righe (numero degli attributi) e d colonne (numero delle istanze).
- Data una qualsiasi matrice txd, esistono matrici T,S e D', tali che: X=TSD'
- T e D sono le matrici dei vettori singolari (eigenvectors) sinistro e destro di X
- Le colonne di T e le righe di D definiscono uno spazio ortonormale
- S è la matrice diagonale dei valori singolari di X

SVD



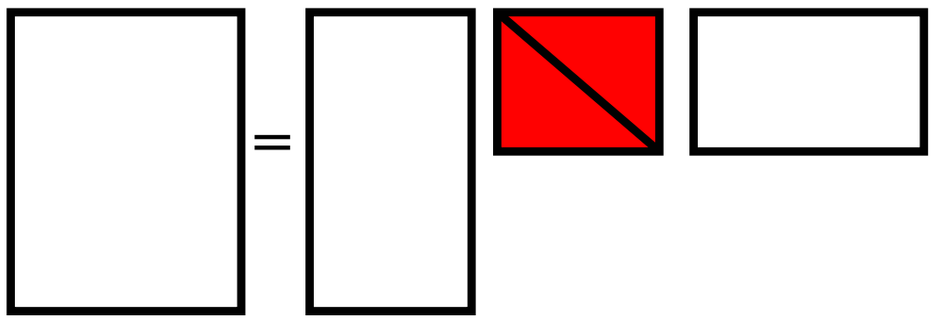
SVD (esempio)



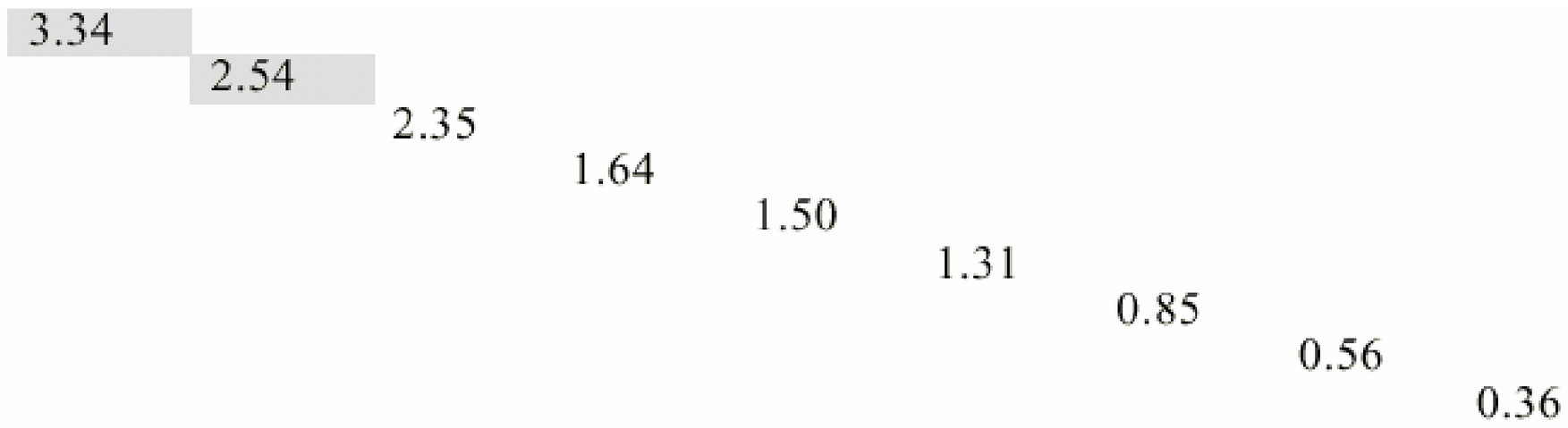
Singular value Decomposition of the words by contexts matrix

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

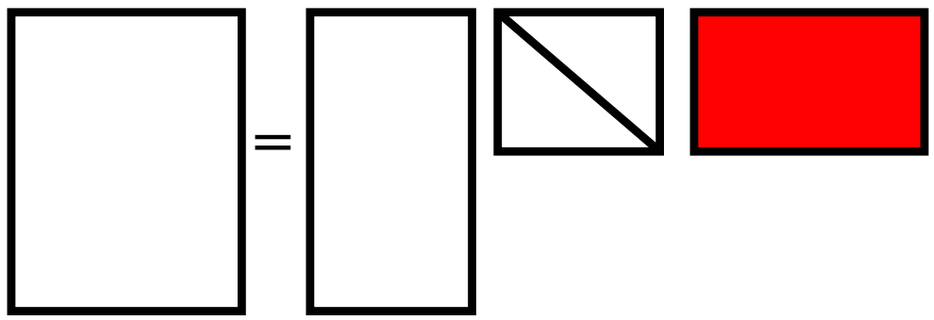
SVD (esempio)



Singular value Decomposition of the words by contexts matrix



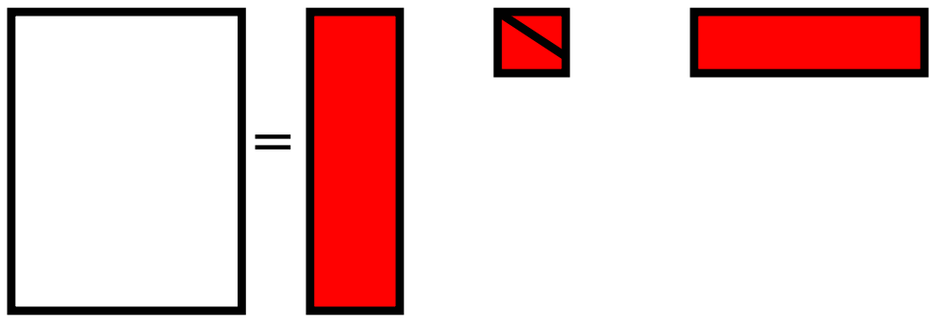
SVD (esempio)



Singular value Decomposition of the words by contexts matrix

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

SVD (esempio)

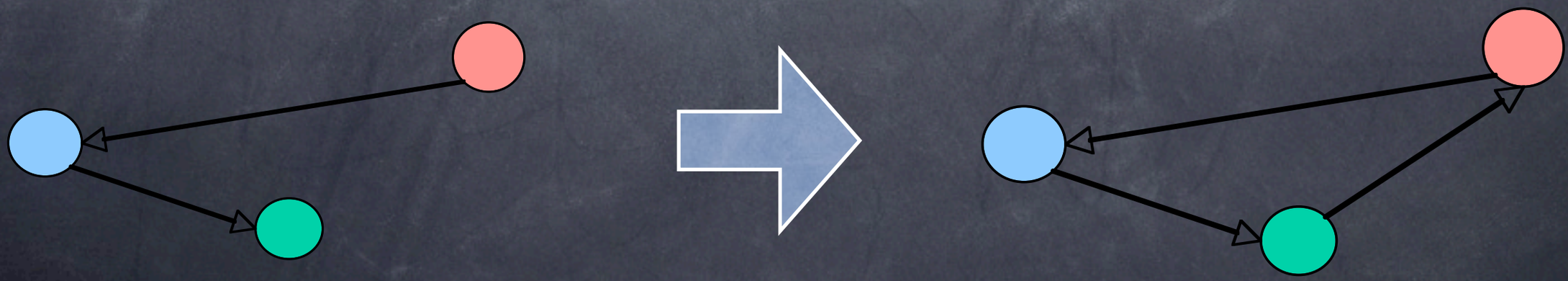


Singular value Decomposition of the words by contexts matrix

Esempio su immagini (matrici bidimensionali)

Scopo SVD

- Trovare relazioni latenti nel grafo
- Esaltare i CONCETTI:
  - Scorporando termini polisemici
  - Accorpando termini sinonimi



HAL

Hyperspace Analogue to Language

- Assunzione: attributi simili compaiono in contesti simili
- Costruzione del modello del contesto con gli attributi che compaiono
- Si usa una dimensione fissata della finestra per analizzare il corpus

HAL

Hyerspace Analogue to Language

- 2 tipi di co-occorrenza: diretta e indiretta
  - Diretta: misura quanto spesso due attributi appaiano insieme
    - "drink" e "wine" sono simili perchè spesso compaiono insieme
  - Indiretta: misura quanto spesso due attributi compaiano insieme ad un terzo
    - "wine" e "beer" sono simili perchè compaiono spesso assieme al termine "drink"

HAL (Hyperspace Analogue Language)

- Denotando il numero di volte che la word w' occorre a distanza dal termine w come:

$$n(w, k, w')$$

- Indicando la grandezza della finestra in cui eseguire tale analisi con K è possibile definire la forza esercitata dalla co-occorrenza tra le due word:

$$W(k) = K - k + 1$$

- La rappresentazione della co-occorrenza sull'intera collezione tra due word w' e w secondo HAL:

$$HAL(w'/w) = \sum_{k=0}^{K} W(k) n(w, k, w')$$

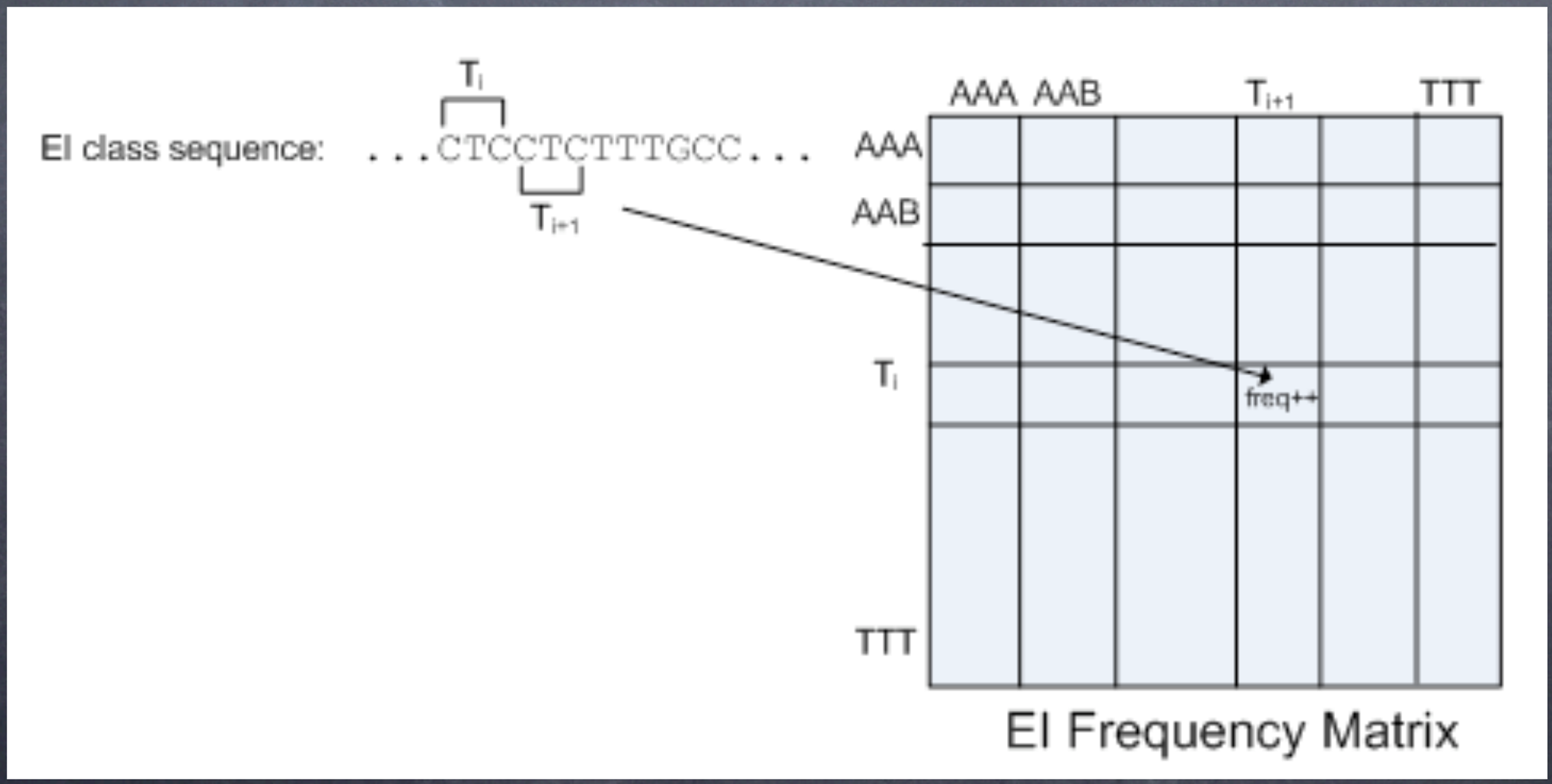
Il sistema implementato...

Raccolta di informazioni...

- Ogni sequenza del training set viene scomposta in un insieme di triplette di basi azotate che chiameremo termini.

- Lo step iniziale nella generazione del classificatore è la costruzione di due matrici quadrate di ordine N=64, chiamate EI Frequency Matrix e IE Frequency Matrix, che rappresentano rispettivamente il numero di occorrenze con cui un termine i e il suo successivo i+1 sono presenti nelle sequenze di classe EI e di classe IE.

Matrice di adiacenza



Algoritmo 1. GAP-RATIO

```
function generateClassifierEI_IE(freqEImatrix, freqIEmatrix)
returns EI Classifier, IE Classifier

inputs: EI Frequency Matrix, IE Frequency Matrix

Odds EI Matrix ← calculateOddsMatrix(freqEImatrix);
Odds IE Matrix ← calculateOddsMatrix(freqIEmatrix);

GapRatio EI Matrix ← calculateGapRatioMatrix
                     (Odds EI Matrix, Odds IE Matrix);
GapRatio IE Matrix ← calculateGapRatioMatrix
                     (Odds IE Matrix, Odds EI Matrix);

EI Classifier ← calculateSVD(GapRatio EI Matrix, k);
IE Classifier ← calculateSVD(GapRatio IE Matrix, k);

return EI Classifier, IE Classifier
```

ODDS - GAP RATIO

$$\text{Matrice degli ODDS} = \frac{a_{ij}}{\sum_i a_{ij} - a_{ij}}$$

$$\text{Matrice GAP RATIO\_IE} = \frac{\text{oddsIE}_{ij}}{\text{oddsEI}_{ij}}$$

per ogni i,j = 1,2…64

Algoritmo 2. HAL

```
function generateClassifierEI_IE(halEImatrix, halIEmatrix)
returns EI Classifier, IE Classifier

inputs: EI HAL Matrix, IE HAL Matrix

pHAL EI Matrix ← calculatepHALMatrix(halEImatrix);
pHAL IE Matrix ← calculatepHALMatrix(halIEmatrix);

EI Classifier ← calculateSVD(pHAL EI Matrix, k);
IE Classifier ← calculateSVD(pHAL IE Matrix, k);

return EI Classifier, IE Classifier
```

$$\text{Matrice pHAL} = \frac{a_{ij}}{\sum_j a_{ij}} \qquad \text{per ogni } i,j = 1,2\ldots64$$

Classificazione...

La sequenza da classificare viene scomposta in termini. La classificazione viene effettuata secondo la formula:

$$\text{Max}\left(\sum_{i=1}^{N-1} ei_i, \quad \sum_{i=1}^{N-1} ie_i \right)$$

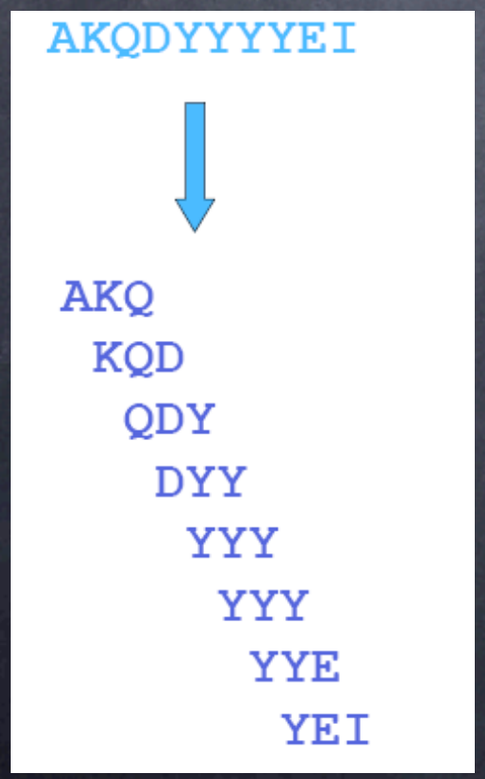
dove ei e ie, sono i valori estratti rispettivamente dalle matrici EI_CLASSIFIER e IE_CLASSIFIER in corrispondenza dei termini T_i e T_{i+1}

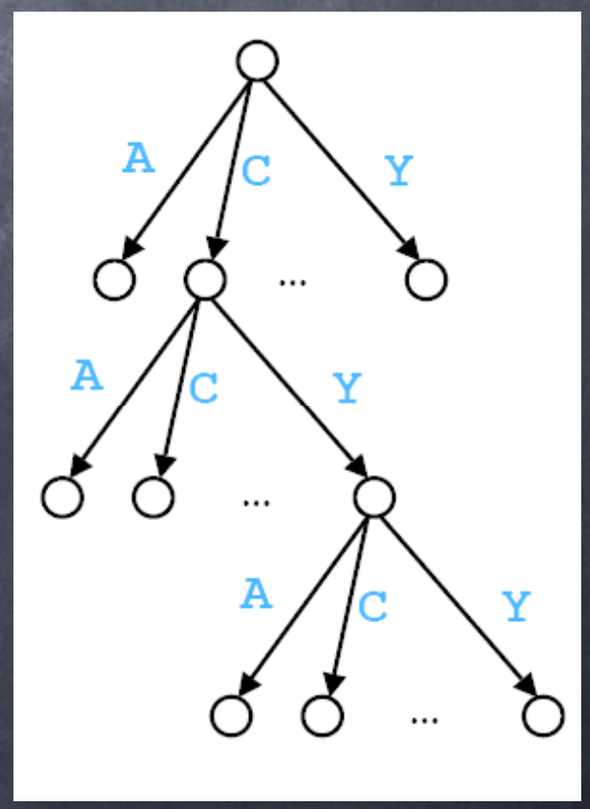
Competitors

- K-Spectrum SVM
  - C.Leslie et Al. "The spectrum Kernel: a string kernel for SVM protein classification" (2002)
- SVM Pairwise
  - L. Liao et Al. "Combining pairwise sequence similarity SVM for detecting remote protein evolutionary and structural relationship" (2003)
- SVM Fisher
  - T. Jaakola et Al. "Using the fisher kernel method to detect remote protein homologies" (1999)

K-Spectrum Feature Map

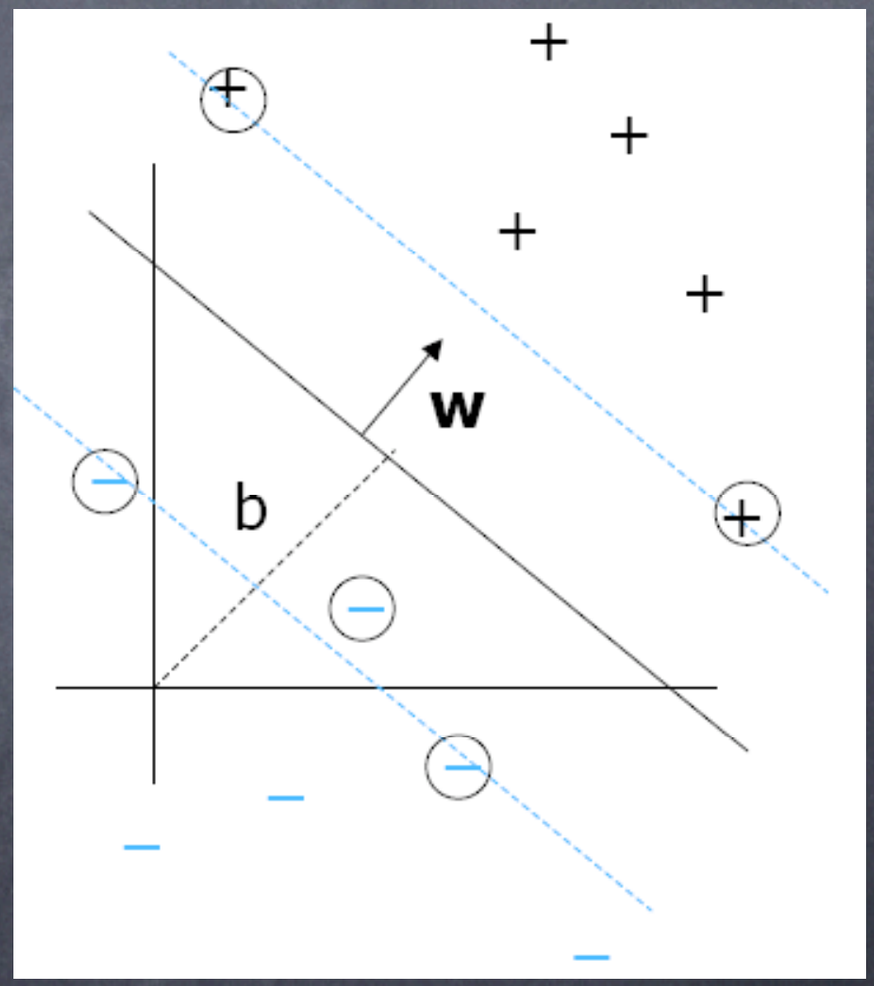
- Idea: mappare gli attributi su una sliding window di sequenze
- Dimensione dello spazio delle istanze: 4^k





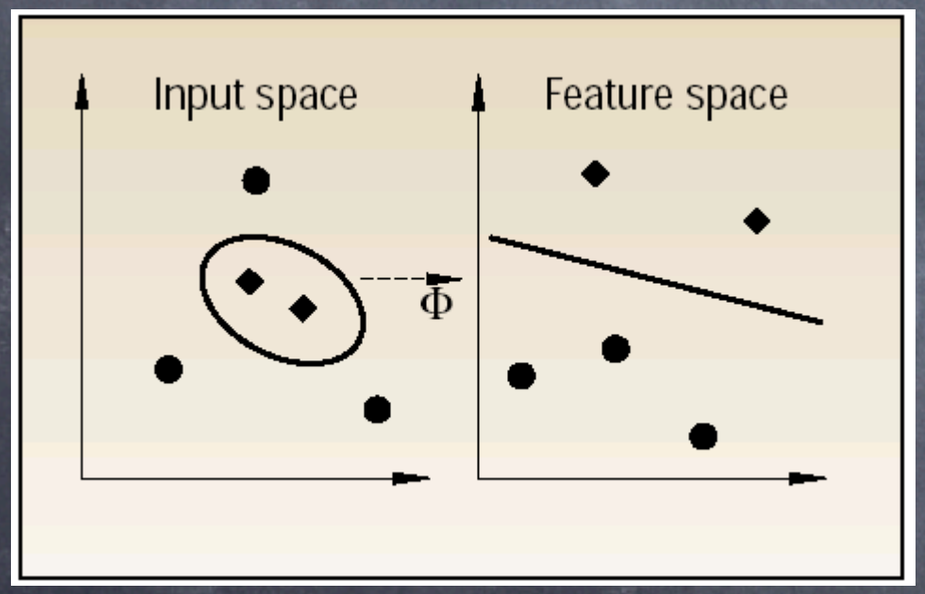
Support Vector Machines

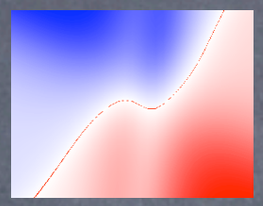
- Gli esempi di training sono mappati all'interno di uno spazio vettoriale ad alta dimensionalità

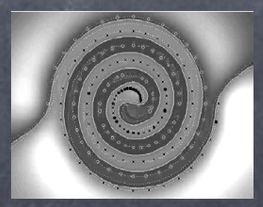


- $(x) = \langle w, F(x) \rangle + b$
- $w = \Sigma_i y_i \alpha_i F(x_i)$

Superare la non linearità







$$k(x,x') = e^{-\|x-x'\|^2}$$

$$e^{-\frac{\|x-x'\|^2}{2\sigma^2}}$$

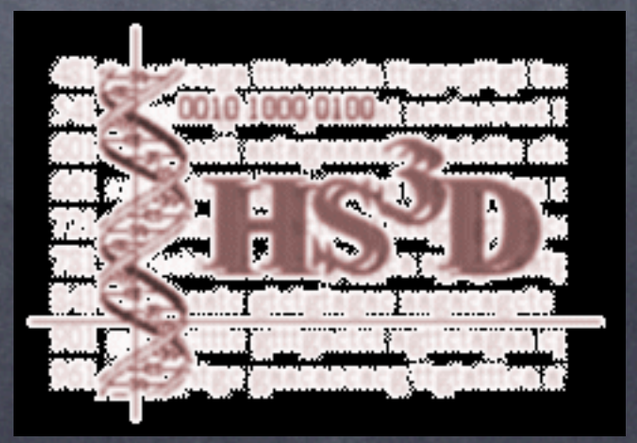
$$(x \cdot y + 1)^d$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$z = x \cdot x' - \theta$$

Benchmark...

- HSSD (Homo Sapiens Splice Sites Dataset) - Polastro and Rampone 2002:
- 5000 esempi (140 attributi) estratti a caso:
  - 25% IE
  - 25% EI
  - 50% N



Risultati - HAL

- HAL - Tuning con LOO: k=17,18,19

```
----------MICRO----------
Accuracy: 0.806
ErrorRate: 0.194
Precision: 0.709
Recall: 0.709
F1Measure: 0.709
BreakEven: 0.709

----------MACRO----------
Accuracy: 0.806
ErrorRate: 0.194
Precision: 0.692
Recall: 0.692
F1Measure: 0.692
BreakEven: 0.692
```

```
----------MICRO----------
Accuracy: 0.806
ErrorRate: 0.194
Precision: 0.709
Recall: 0.709
F1Measure: 0.709
BreakEven: 0.709

----------MACRO----------
Accuracy: 0.806
ErrorRate: 0.194
Precision: 0.692
Recall: 0.692
F1Measure: 0.692
BreakEven: 0.692
```

```
----------MICRO----------
Accuracy: 0.804
ErrorRate: 0.196
Precision: 0.706
Recall: 0.706
F1Measure: 0.707
BreakEven: 0.707

----------MACRO----------
Accuracy: 0.804
ErrorRate: 0.196
Precision: 0.689
Recall: 0.689
F1Measure: 0.689
BreakEven: 0.689
```

Risultati - GAP Ratio

- GAP - Tuning con LOO: k=17,18,19

```
----------MICRO----------
Accuracy: 0.803
ErrorRate: 0.197
Precision: 0.704
Recall: 0.704
F1Measure: 0.704
BreakEven: 0.704

----------MACRO----------
Accuracy: 0.803
ErrorRate: 0.197
Precision: 0.693
Recall: 0.686
F1Measure: 0.685
BreakEven: 0.690
```

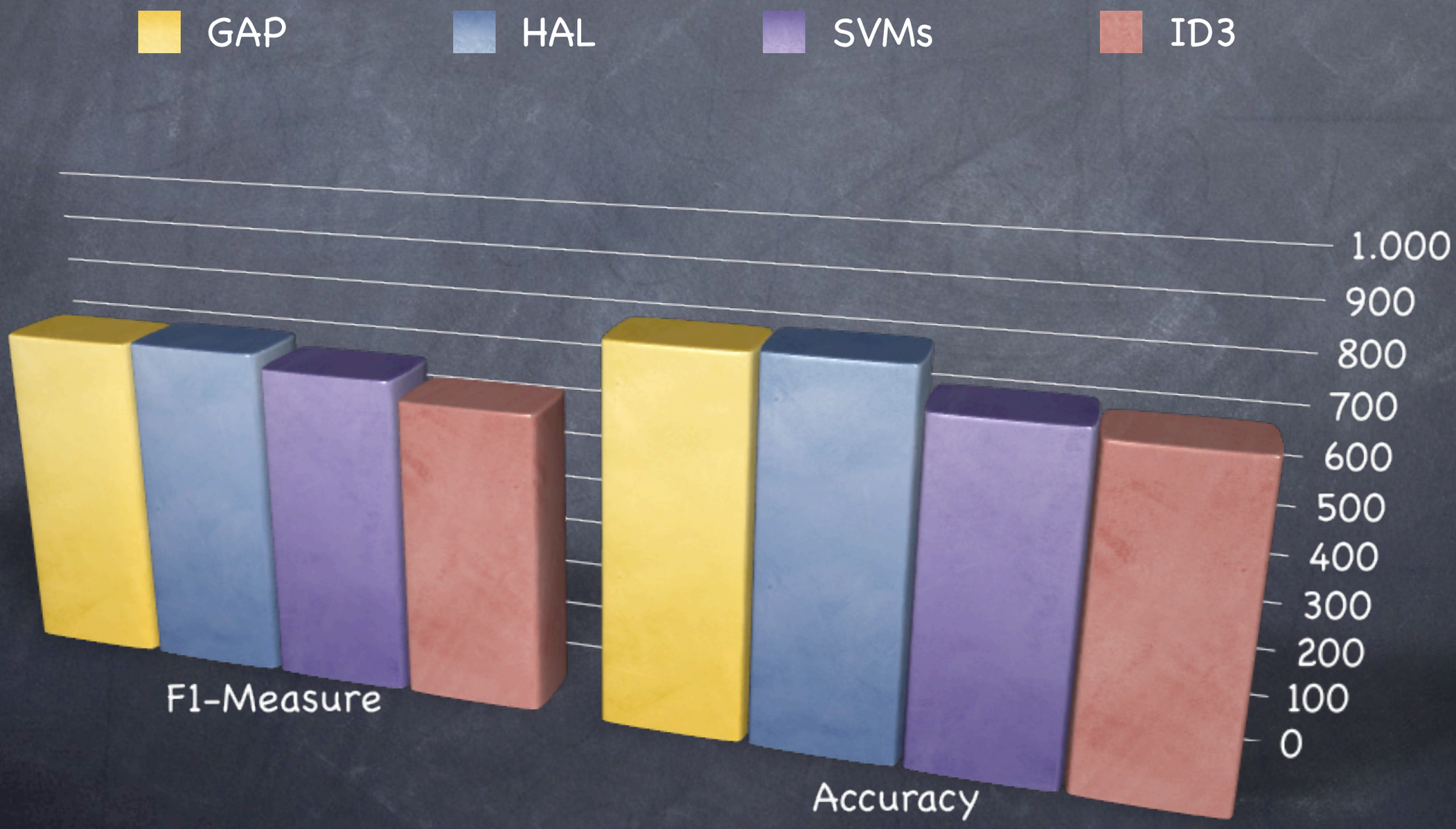
```
----------MICRO----------
Accuracy: 0.802
ErrorRate: 0.198
Precision: 0.703
Recall: 0.703
F1Measure: 0.703
BreakEven: 0.703

----------MACRO----------
Accuracy: 0.802
ErrorRate: 0.198
Precision: 0.692
Recall: 0.685
F1Measure: 0.684
BreakEven: 0.689
```

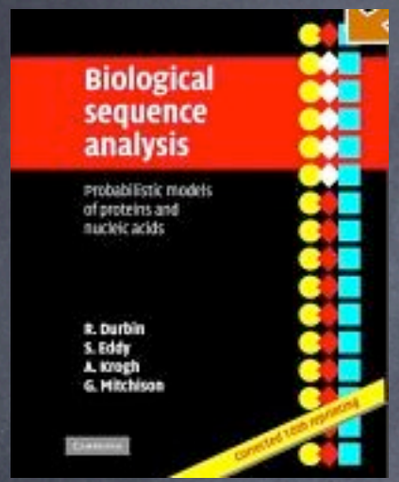
```
----------MICRO----------
Accuracy: 0.803
ErrorRate: 0.197
Precision: 0.704
Recall: 0.704
F1Measure: 0.704
BreakEven: 0.704

----------MACRO----------
Accuracy: 0.803
ErrorRate: 0.197
Precision: 0.694
Recall: 0.689
F1Measure: 0.687
BreakEven: 0.686
```

Confronti con altri algoritmi (WEKA)



Bibliografia (1/3)



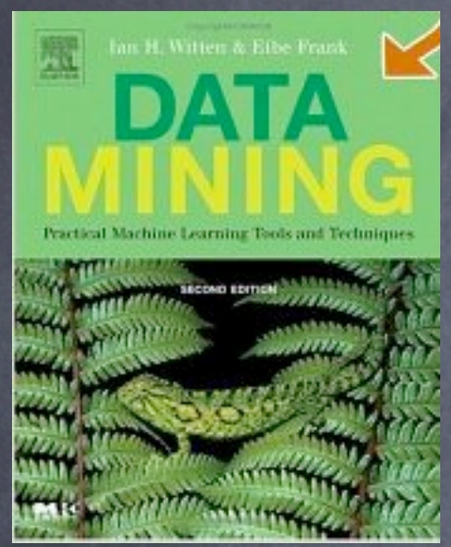
Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids

Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison

Bioinformatics: The Machine Learning Approach, Second Edition (Adaptive Computation and Machine Learning)

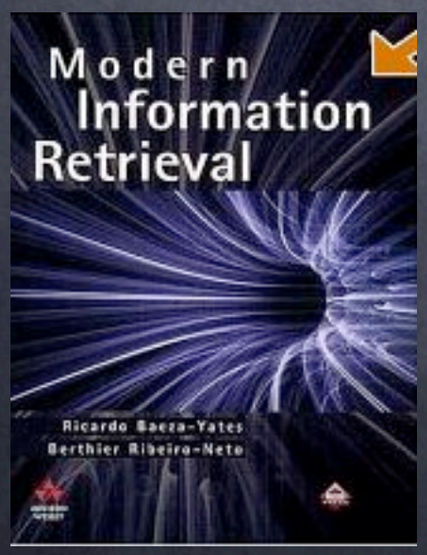
Pierre Baldi and Søren Brunak

Bibliografia (2/3)



Data Mining: Practical Machine Learning Tools and Techniques, Second Edition

Ian H. Witten and Eibe Frank



Modern Information Retrieval

Ricardo Baeza-Yates and Berthier Ribeiro-Neto

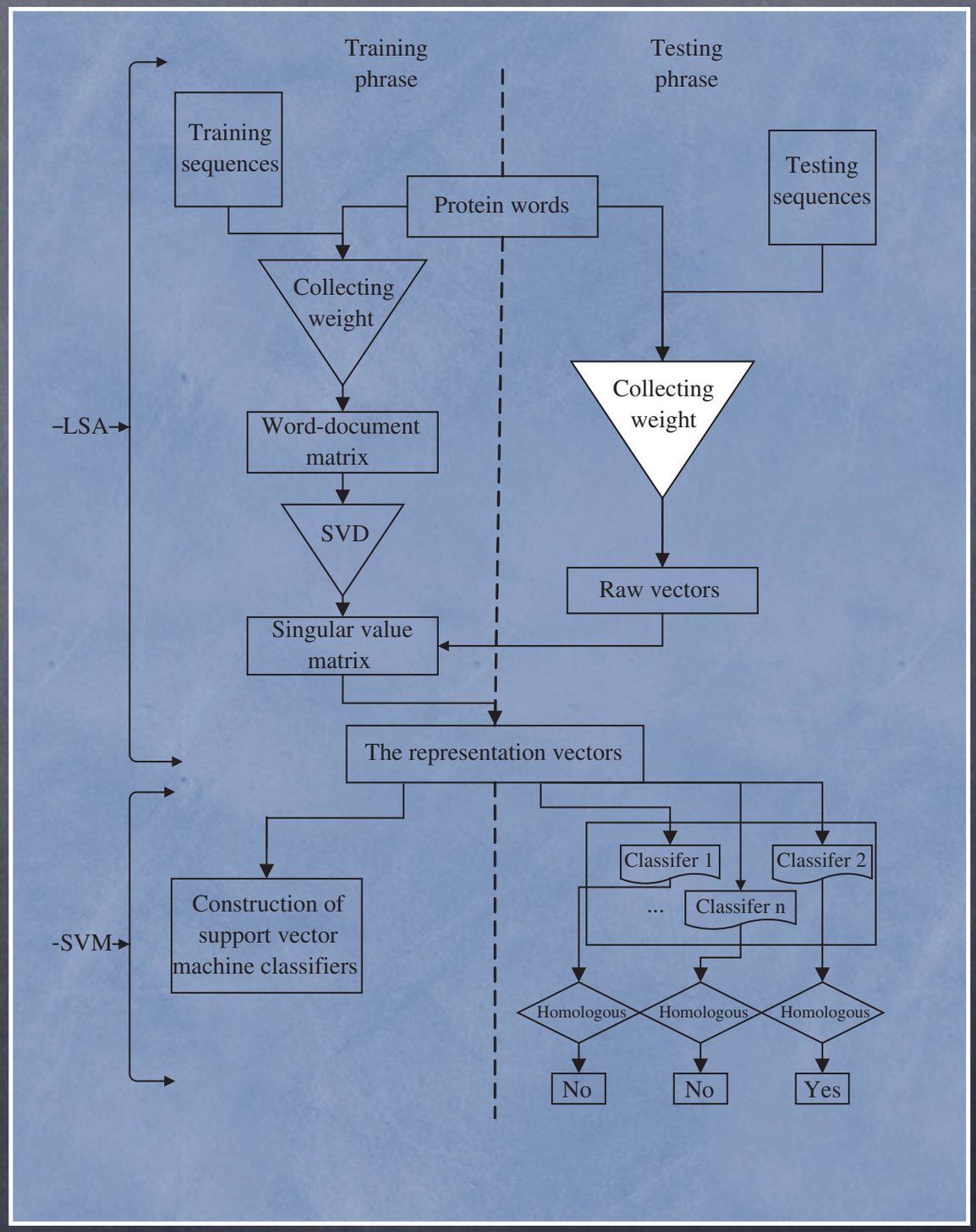
Bibliografia (3/3)

BIOINFORMATICS

Application of latent semantic analysis to protein remote homology detection

Qi-wen Dong, Xiao-long Wang and Lei Lin

School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China



Tecnologia

- Requisiti Non Funzionali
  - Paradigma OO: JAVA 1.5.0
  - Cluster Linux RedHat AS 4
  - Application Server: JBoss 4.2.0GA
  - Integrazione Enterprise Service Bus (Tibco) Architettura SOA
- Requisiti Funzionali
  - WEKA per comparazione con lo stato dell'arte
  - COLT (CERN) per l'elaborazione matriciale (SVD)