Lezione 2

Status	Done
Due date	@02/20/2025
Attach file	2_PreprocessingDati.pdf

Fasi del preprocessing

Preparazione dei dati

Prima di applicare qualsiasi algoritmo di mining sui dati, occorre processarli correttamente poichè alcuni dati potrebbero:

- essere errati o incompleti
- non essere direttamente utilizzabili e richiedere una corretta estrazione dei feature
- potrebbero provenire da sorgenti diverse (e bisogna quindi armonizzarli)
- contenere troppe informazioni, alcune delle quali poco utili o che potrebbero generare rumore

Le fasi del preprocessing sono quindi le seguenti:

1. Estrazione dei feature: estrarre da dati grezzi (raw data) feature che siano significativi e facili da interpretare. Se i dati vengono da più sorgenti, bisogna integrarli in un unico dataset
2. Portabilità dei dati: se un algoritmo è in grado di lavorare con uno specifico tipo di dato, convertirlo opportunamente per un altro tipo di dato
3. Cleaning dei dati: rimuovere o trattare opportunamente dati mancanti, errati o inconsistenti
4. Riduzione dei dati: ridurre la dimensionalità dei dati mediante tecniche di selezione di dati, riduzione di feature o trasformazione dei dati.

In base alle caratteristiche dei dati, potrebbero essere neccessarie solo alcune delle fasi sopra. Inoltre, alcuni algoritmi di data mining (come classificazione o clustering) potrebbero incorporare tecniche di trasformazione dei dati o estrazione dei feature.

1 - Estrazione di feature

L'estrazione dei feature dai dati consiste nel creare un set di feature più **adatte** al problema da risolvere. Dipende fortemente dall'obiettivo che ci si prefissa e dalla tipologia di dati. In caso di errata estrazione, l'analisi sarà penalizzata. Quali sono le sorgenti solite? Di seguito le più frequenti:

- Dati sensoriali: segnali a basso livello utilizzati in questa forma o trasformati in feature tramite Fourier (che trasforma gli impulsi elettrici dei sensori in dati fruibili). I dati si raccolgono dai sensori (principalmente nell'ambito dell'energia rinnovabile).

- Immagini: rappresentati da matrici di pixel (b/n o a colori) o istogrammi di colore
- Web logs: log di accesso al web solitamente rappresentati come testo formattato
- Dati di traffico di rete: pacchetti trasferiti in rete, da cui possono essere estratti feature di vario tipo
- Documenti: dati raw non strutturati, relazioni linguistiche tra entità diverse. Si utilizza una rappresentazione tramite bag-of-words.

2 - Portabilità dei dati

La portabilità consente di trasformare un tipo di dato (ad es. stringa) in un altro (ad es.
numerico). La trasformazione può essere lossy o lossless. Le conversioni possono avvenire in diversi modi, come di seguito.

Da dati numerici a categoriali

La conversione da dati numerici a categoriali è detta discretizzazione, che divide l'insieme dei valori numerici in intervalli. Ad ogni valore numerico è associato uno dei intervalli. Ad esempio intervalli di età $[0, 9], [10, 19], \dots$ a cui è possibile associare valori interi ($1, 2, \dots$).

Discretizzazione

La conversione può avvenire tramite discretizzazione. Ne consideriamo 3 tipologie:

Equi width ranges: sotto-intervalli costruiti in modo che abbiano la stessa ampiezza

Equi-log ranges: ogni sotto-intervallo $[a, b]$ è costruito in modo che $log_x a + log_x b$ abbia sempre lo stesso valore

Equi-depth ranges: sotto intervalli costruiti in modo che abbiano lo stesso numero di record

Da dati categoriali a numerici

La conversione da dati catergoriali a numerici avviene semplicemente associando ad ogni categoria un numero.

Alcuni modelli (come le reti neurali) richiedono che i valori numerici associati alle categorie non siano ordinali, ovvero che non presuppongano un ordine tra le classi.

In questi casi si adotta lo schema one-hot encoding: ogni catergoria è rappresentato da un vettore binario di 0, tranne i bit di categoria.

Sample	Category	Numerical
1	Human	1
2	Human	1
3	Penguin	2
4	Octopus	3
5	Alien	4
6	Octopus	3
7	Alien	4

Sample	Human	Penguin	Octopus	Alien
1	1	0	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	1	0
5	0	0	0	1
6	0	0	1	0
7	0	0	0	1

Da testo a dati numerici

La conversione da testo a dati numerici, ad esempio per la conversione di una collezione di documenti, può essere rappresentata tramite una matrice dove:

- le righe sono le parole
- le colonne sono i documenti

I valori della matrice è un intero che rappresenta la frequenza con cui la parola i si presenta nel documento j.

3 - Cleaning dei dati

Alcuni errori, mancanze o incosistenze possono verificarsi durante il processo di collezionamento dei dati. Bisogna quindi:

a - Gestire dati mancanti (imputazione dei dati)

Si può gestire in vari modi:

- eliminando record contenenti uno o più valori mancanti

    - poco pratico se il record ha molti valori mancanti
- stima dei valori mancanti tramite interpolazione
    - rischioso poichè stime errate possono condizionare i risultati
- lasciare all'algoritmo il compito di gestire i dati mancanti
    - non tutti gli algoritmi sono in grado

b - Gestire dati errati o inconsistenti (outlier)

Bisogna essere in grado di rilevare inconsistenze (valori chiaramente fuori scala) o duplicati: solitamente questo passaggio è necessario se si integrano dati provenienti da sorgenti diverse.

La risoluzione può avvenire anche tramite un esperto nel dominio in questione (USA↔Roma).

Infine si ricorre all'utilizzo di tecniche statistiche per il rilevamento degli outlier.

Uno di questi metodi statistici di basa sui quantili:

Un quantile di ordine x, con $0 \leq x \leq 1$ è un valore q_x che divide un insieme di valori S in due parti, corrispondenti a $x\%$ valori di S minori di q_x e $(100 - x)\%$ elementi maggiori o uguali a q_x.

L'analisi dell'outlier avviene sfruttando i quartili.

sfruttando l'interquantile range, ovvero la differenza tra terzo e primo quartile, si riesce a scovare gli (eventuali) outlier:

- se la differenza tra 1° e un valore x (freccia verde) è pari ad almeno 1.5 volte l'interquantile range, allora x è un outlier (valore troppo piccolo)

- allo stesso modo si ragiona con il 3° quartile e un valore x′ (seconda freccia verde).

c - Scala e normalizzazione

Scalare e normalizzare opportunamente i dati è fondamentale per evitare che in fase di analisi alcune feature siano pesate troppo.

- In molti casi gli attributi hanno scale diverse e quindi non risultano confrontabili (ETA↔SALARIO)
- Qualsiasi funzione di aggregazione applicata ai dati (ad es. distanza euclidea)
  risulterà essere dominata dall'attributo con valori più alti.

Ho due possibili metodi per la correzione del problema:

▼ Standardizzazione

Avendo una matrice, per ogni colonna (vettore) calcolo media e deviazione standard. Poi prendo ogni valore, gli sottraggo la media e divido per la deviazione. Così ottengo dei dati standardizzati.

Per la media:

$$\mu_j = \frac{\sum_{i=1}^{N} x_i^j}{N}$$

Per la varianza:

$$\sigma_j = \frac{\sum_{i=1}^{N} (x_i^j - \mu_j)^2}{N}$$

La standardizzazione consiste nel trasformare ogni valore nel nuovo valore standardizzato usando la seguente formula:

$$z_p^j = \frac{x_p^j - \mu_j}{\sigma_j}$$

▼ Min-max scaling

Per ogni vettore trovo min e max.

Il min-max scaling trasforma ogni calore x in un nuovo valore y usando la seguente formula:

$$y = \frac{x - min}{max - min}$$

Questo scaling produce valori tra 0 e 1.

A differenza della standardizzazione, quando il max e il min sono valori molto estremi (ovvero in presenza di outlier), questo scaling è meno robusto.

4 - Riduzione dei dati

La riduzione della dimensionalità dei dati consiste nel rappresentare i dati in maniera più compatta, in modo da facilitare l'utilizzo di algoritmi di data mining. Ho di base 4 tecniche possibili:

▼ Sampling dei dati

Consiste nel effettuare un campionamento dei dati per creare un dataset più piccolo.

Il sampling può essere:

- Biased: considero le porzioni di dati più rilevanti (ad esempio le più recenti a livello temporale)
- Stratificato: il dataset viene partizionato in strati e per un sottinsieme di campioni viene estratto da ogni strato.

▼ Selezione di feature

Consiste nel scartare dai dati attributi che sono irrilevanti per l'analisi. La rilevanza delle feature dipende dal dominio del problema.

La selezione può essere:

- non supervisionata: non sono esperto il problema e ottimizzo senza sapere cosa perdo
- supervisionata: selezionole feature che riesco a predire, eliminando quelle feature che, in base alla mia esperienza, non sono necessarie. In seguito controllo l'accuracy: se dopo riaddrestamento con eliminazione delle feature l'accuracy scende, allora ho eliminato una feature rilevante.

Nel caso io abbia tante feature e mi sia impossibile controllarle a mano, uso delle tecniche di riduzione.

a- Riduzione con rotazione degli assi (PCA, SVD, LSA)

I dati reali sono correlazioni tra attributi legati a vincoli o regole, oppure possono essere impliciti e possono quindi sfuggire ad una prima analisi. Una possibile tecnica è individuare una rotazione degli assi che permetta di rimuovere uno o più dimensioni a bassa varianza (di poco peso). Le principali tecniche utilizzate sono:

- Principal Component Analysis (PCA);
- Singular Value Decomposition (SVD);
- Latent Semantic Analysis (LSA).

a.1 - Principal Component Analysis (PCA):

La PCA serve a **ruotare i dati** in un nuovo sistema di coordinate, in modo che la maggior parte della **varianza** sia concentrata in poche dimensioni. Così:

- si riducono le dimensioni del problema
- si eliminano ridondanze tra variabili fortemente correlate
- si mantiene la maggior parte dell'**informazione**.

Il primo passo è costruire una matrice di covarianza degli attributi.

La varianza dei dati lungo una direzione si descrive tramite la **matrice di covarianza**.

- Dati due vettori X e Y, la **covarianza** misura come variano insieme:
  - Se X grande → anche Y grande ⇒ covarianza positiva.
  - Se X grande → Y piccolo ⇒ covarianza negativa.
  - Se non hanno relazione ⇒ covarianza vicina a zero.

Con **n vettori k-dimensionali**, la **matrice di covarianza** è una matrice C con elementi:

Le principali proprietà sono:

- La covarianza di n vettore con se stesso è la **varianza**.
- La matrice di covarianza è **simmetrica**:

  Cov(X,Y)=Cov(Y,X)\text{Cov}(X,Y) = \text{Cov}(Y,X)Cov(X,Y)=Cov(Y,X).
- Il **segno della covarianza** indica la correlazione:
  - Positiva → correlati.
  - Negativa → anti-correlati.

Per ottenere le componenti principali:

A partire da un dataset D (M righe = record, N colonne = attributi):

1. Si calcola la matrice di covarianza.
2. Si ricavano le **componenti principali** = **nuove variabili** ottenute come

Le componenti **non sono correlate** tra loro e catturano gran parte della **varianza** in poche dimensioni.

👉 Se gli attributi sono molto correlati, bastano poche componenti principali per descrivere quasi tutta l'informazione.

⚠️ Ma attenzione: queste nuove componenti **non hanno significato reale diretto** → sono combinazioni matematiche.

I passi principali della PCA sono:

1. **Standardizzazione dei dati**
   - Portare ogni attributo a media = 0 e varianza = 1.
   - Necessaria perché la PCA è sensibile alla scala delle variabili.
   - Se una variabile ha varianza molto più alta, "domina" le altre.
2. **Calcolo della matrice di covarianza**.
3. **Calcolo delle componenti principali**
   a. Si decomposizione C come $C = P\Lambda P^T$ dove:
      i. P = matrice degli **autovettori** (ortonormali: direzioni principali).

ii. Λ = matrice diagonale degli **autovalori** (quanta varianza spiega ciascuna componente).

4. **Creazione del vettore delle nuove feature**
   - Si scelgono i primi P autovettori (quelli con autovalori più grandi).
5. **Trasformazione dei dati**
   - Se D è il dataset originale (M x N) e P_p è la matrice (N x P) dei primi autovettori: $D' = DP_p$
   - D' è il nuovo dataset nello spazio delle componenti principali.

a.2 - Singular Value Decomposition (SVD)

La Singular Value Decomposition (SVD) decompone una matrice M di dimensione $m \cdot d$ nel prodotto di tre matrici:

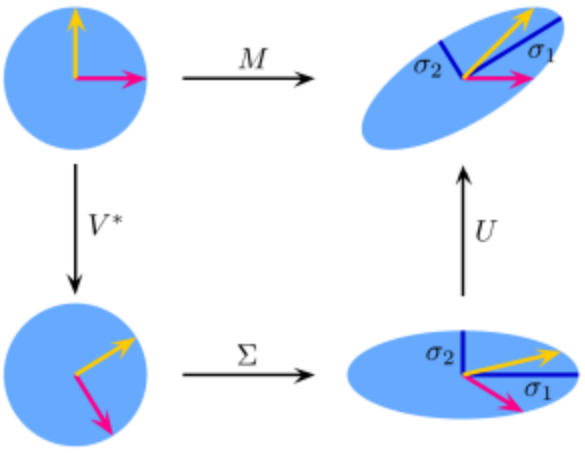
$$M = U\Sigma V^T$$

dove:

- U è una matrice n×n le cui colonne sono vettori ortonormali (vettori singolari sinistri), e corrispondono agli autovettori di MM^T.
- Σ è una matrice diagonale n×d, i cui elementi sono chiamati valori singolari. Il numero di valori singolari non nulli corrisponde al rango di M.
- V è una matrice d×d le cui colonne sono vettori ortonormali (vettori singolari destri), e corrispondono agli autovettori di M^TM.

L'interpretazione geometrica è la seguente:

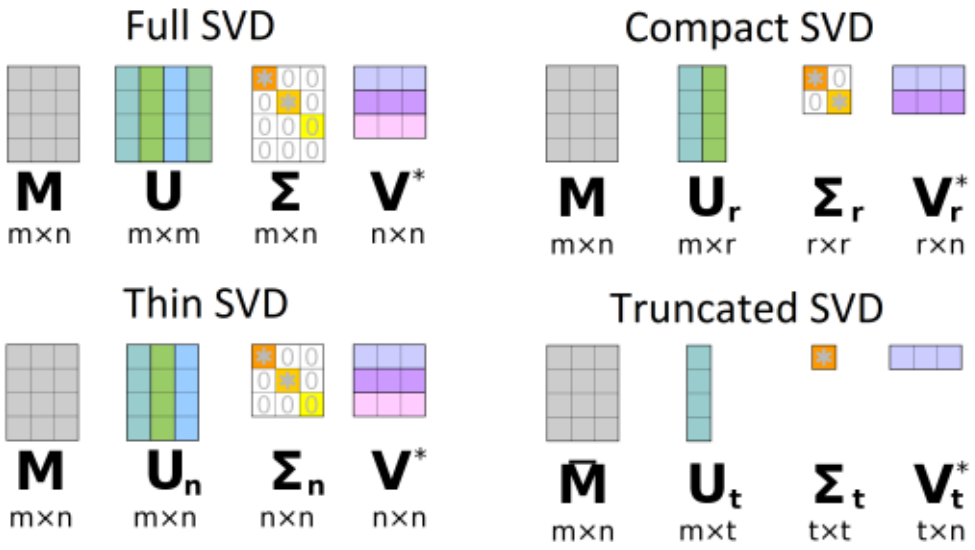
Data una matrice M 2×2. In $\mathbb{R}^2$ considero un cerchio di raggio unitari (con i due vettori unitari). La SVD ruota e trasforma il cerchio in un ellisse in cui i due semiassi hanno lunghezze pari ai valori singolari non nulli di Σ . Per una matrice 3D, quindi nello spazio, si lavora con una sfera che viene ruotata e trasformata in un ellissoide.



$$M = U \cdot \Sigma \cdot V^*$$

In genere si usano versioni ridotte della SVD:

- Thin SVD: rimuove le colonne di U e le righe Σ in eccesso rispetto al numero di righe di V^T : questo assicura la decomposizione esatta
- Compact SVD: rimuove le righe di Σ che contengono valori singolari nulli e di conseguenza anche le colonne di U e le righe di V^T in eccesso rispetto al nuovo numero di righe di Σ: anche questa assicura una decomposizion esatta
- Truncated SVD: mantiene solo le righe di Σ che contengono i più alti valori singolari e rimuove le corrispondenti colonne di U e righe di VT: la decomposizione non è esatta.



a.3 - PCA vs SVD

La SVD è più generale della PCA poiché produce due set di autovettori anziché uno
solo, uno per le righe e uno per le colonne del dataset.

- SVD corrisponde alla PCA nel caso in cui i dati sono centrati attorno allo zero, ovvero la media dei valori di ogni attributo è 0;
- La PCA cattura quanto più varianza possibile nei dati, la SVD cattura quanta più
  distanza euclidea al quadrato rispetto all'origine possibile

a.4 - Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) è un'applicazione della SVD al text mining.
La matrice di partenza è una matrice n x d di n documenti e d termini, contenente le
frequenze normalizzate delle parole in ciascun documento.
La matrice dei documenti e dei termini è molto sparsa:

- I risultati ottenuti con LSA sono molto simili alla PCA;
- La riduzione di dimensionalità che si ottiene con LSA è drastica.

b - Riduzione con trasformazione dei dati

Si tratta di metodi che uniscono alla riduzione di dimensionalità la trasformazione dei dati in tipi di dati meno complessi da analizzare.
Esempi:

- Serie temporali: trasformata di Fourier, Haar wavelet transform;
- Grafi: tecniche di embedding di grafi pesati in spazi multidimensionali che preservano la similarità tra grafi o le distanze tra i nodi (ad es. multidimensional scaling, metodi spettrali).