Riconoscimento e Recupero dell'Informazione per Bioinformatica

LAB. 3 – Standardizzazione dei dati e PCA

Pietro Lovato

Corso di Laurea in Bioinformatica
Dip. di Informatica – Università di Verona
A.A. 2016/2017

Un dataset “reale”: Iris

https://en.wikipedia.org/wiki/Iris_flower_data_set

- Problema: classificazione della specie del fiore iris
  - Tre classi, corrispondenti alle diverse specie
- 150 pattern/oggetti
- 4 features:
  - Lunghezza e spessore di petali e sepali

Un dataset “vero”: Iris

- La scelta di avere le features sulle righe è arbitraria

```
>> load iris.mat
>> whos
  Name          Size                    Bytes  Class

  data          4x150                    4800  double
  labels        1x150                    1200  double
```

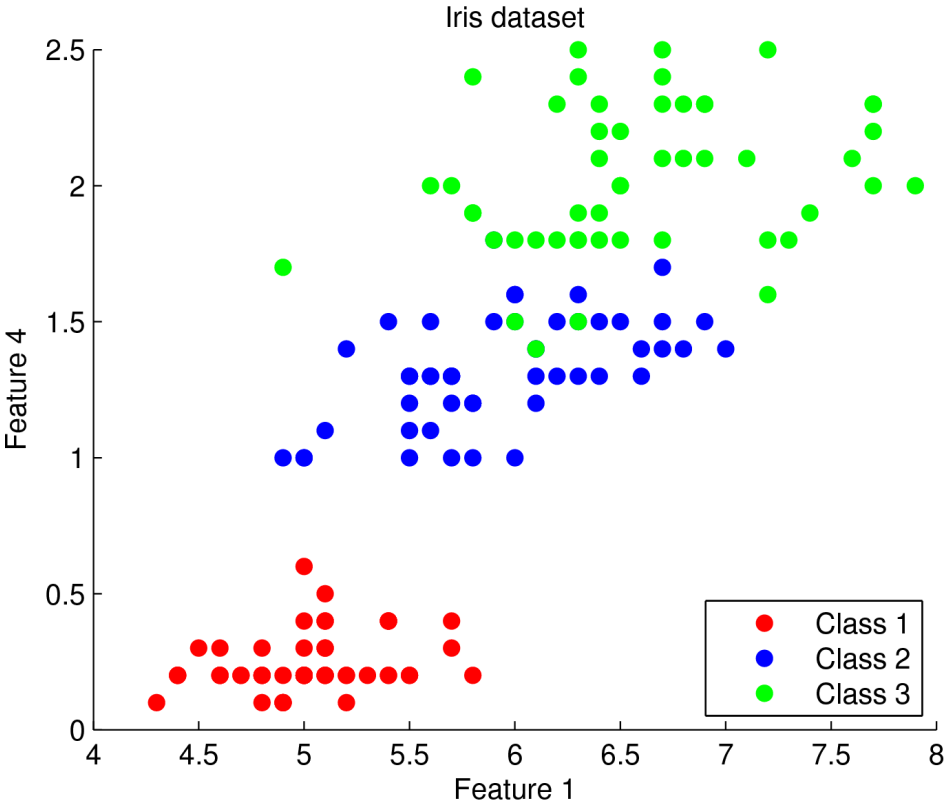
Visualizzazione del dataset

- Per la visualizzazione di un dataset (max. 2 feature) potrebbe essere più utile il comando `scatter`

```
>> scatter(x,y,size,color,'fill')
```

Suggerimenti:

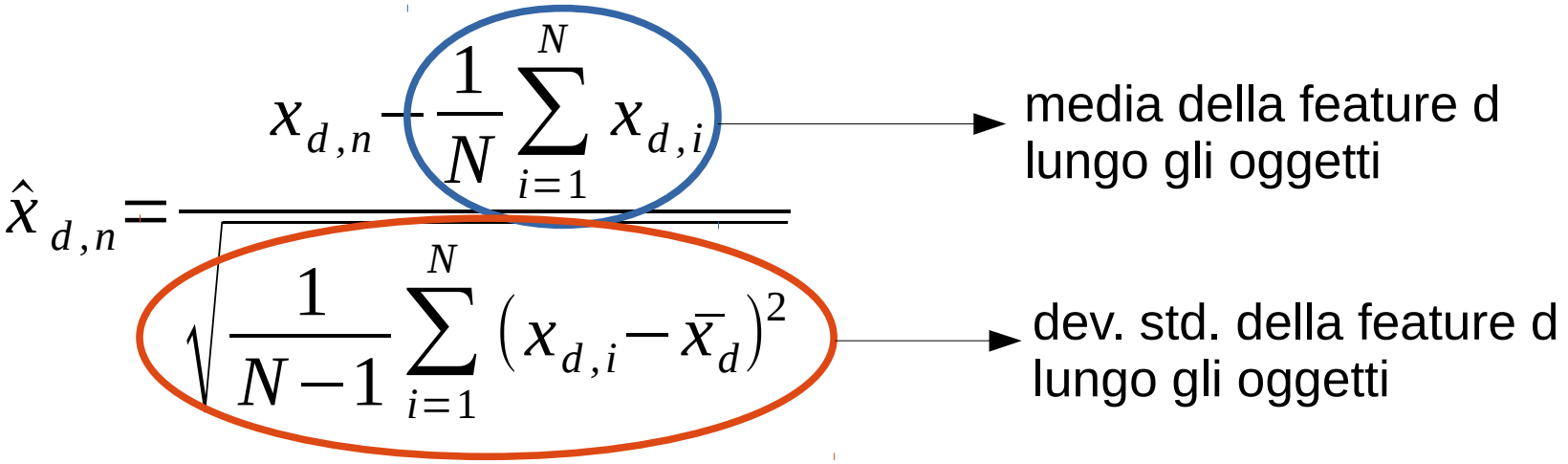
- `color`: si può inserire il vettore `labels`
- Se non si vuole specificare uno dei parametri (e.g. `size`), inserire `[]`



Standardizzazione dei dati

Tipi di standardizzazione:

- z-score:
  - Si centrano i dati (ogni feature avrà media zero lungo tutti gli oggetti)
  - Si divide per la deviazione standard (ogni feature avrà deviazione standard uno lungo tutti gli oggetti)



Standardizzazione dei dati

Tipi di standardizzazione:

- domain (1):
  - Si porta il range dei dati a 1

$$\hat{x}_{d,n} = \frac{x_{d,n}}{(max_n \; x_{d,n}) - (min_n \; x_{d,n})}$$

Minimo valore che assume la feature d lungo gli oggetti

Standardizzazione dei dati

Tipi di standardizzazione:

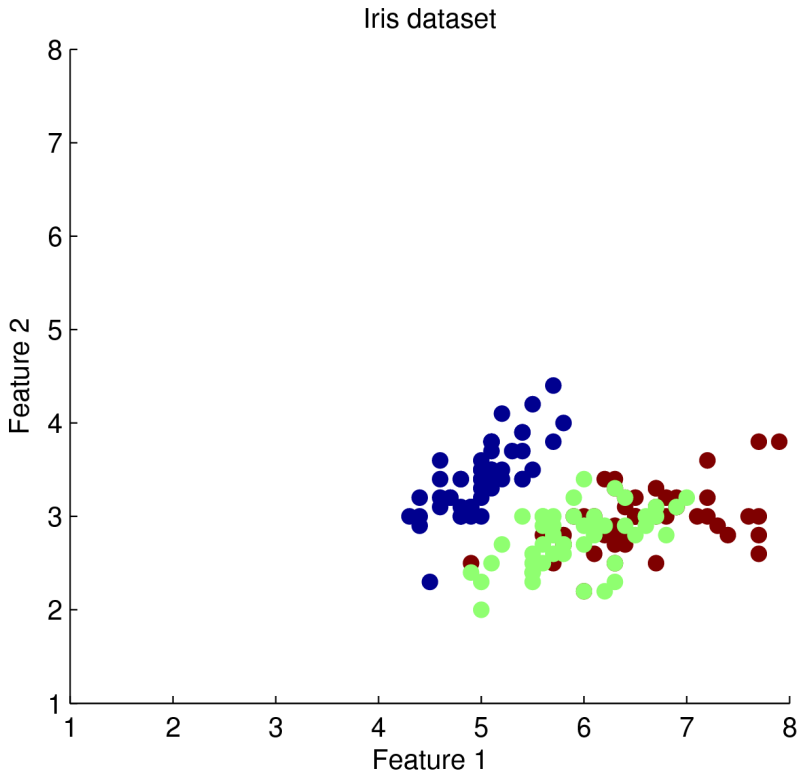
- domain (2):
  - Si porta il dominio dei dati fra 0 e 1

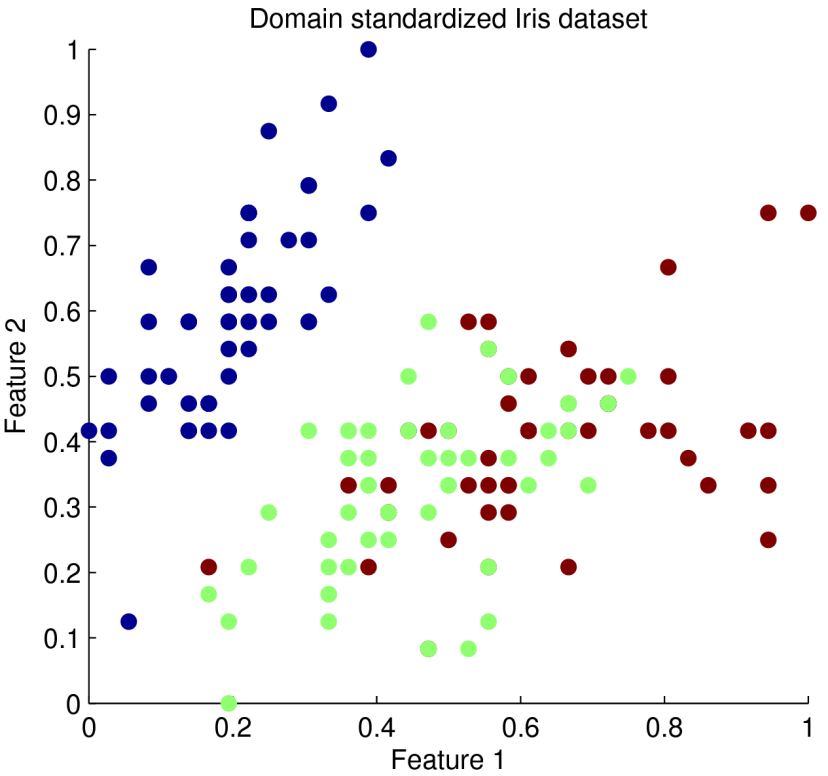
$$\hat{x}_{d,n} = \frac{x_{d,n} - (min_n \; x_{d,n})}{(max_n \; x_{d,n}) - (min_n \; x_{d,n})}$$

Minimo valore che assume la feature d lungo gli oggetti

Esercizio

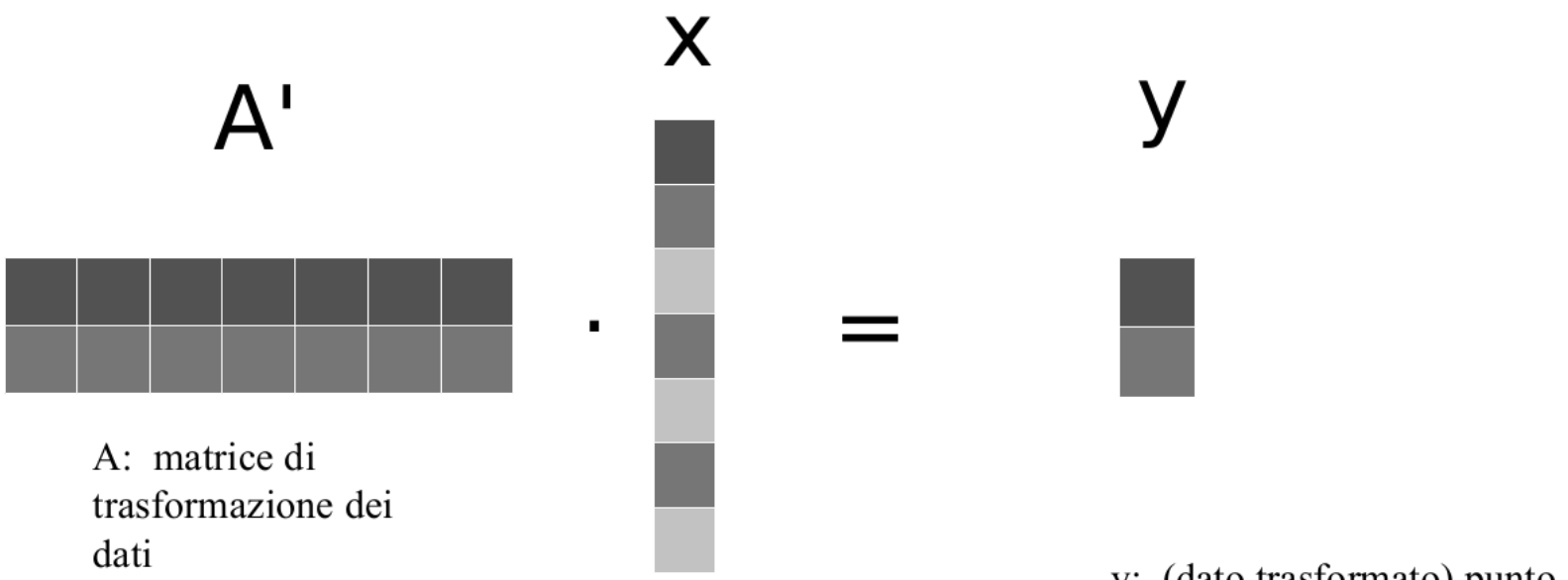
- Capire il contenuto dello script Lezione3Lab_es1.m e provare ad eseguire gli esercizi proposti.





Principal Component Analysis (PCA)

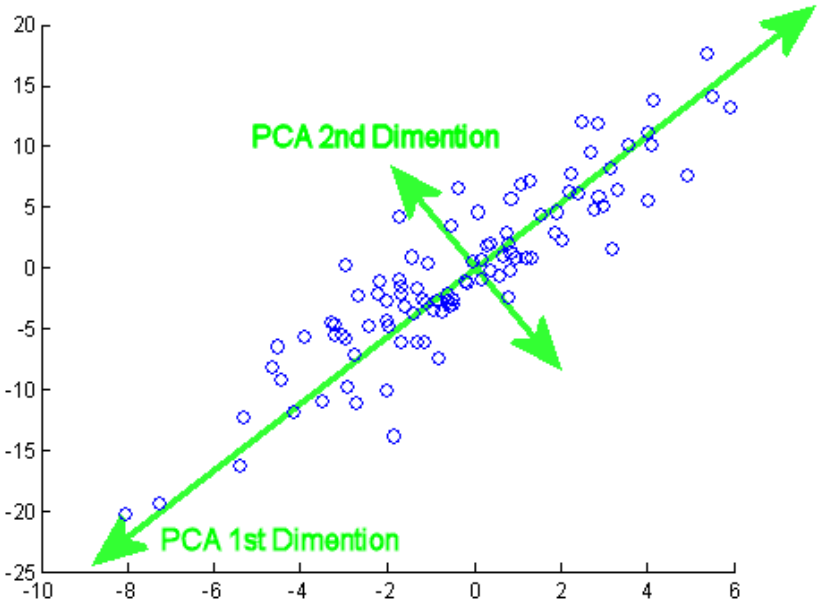
- Idea: si vuole ridurre la dimensionalità dello spazio, mantenendo la maggior quantità di informazione possibile



Principal Component Analysis (PCA)

Cosa fa?

- Proietta i dati in uno spazio tale per cui
  - La prima direzione è quella di massima varianza
  - La seconda direzione è quella di massima varianza, ortogonale alla prima
  - E così via...



Principal Component Analysis (PCA)

Come si fa?

- Calcolo la media e centro i dati $\hat{x}_{d,n} = x_{d,n} - \frac{1}{N}\sum_{i=1}^{N} x_{d,i}$
- Calcolo la matrice di covarianza (DxD)
  - Si può fare direttamente con

```
>> C = 1/(N-1) * datac*datac';
```

- Calcolo autovalori e autovettori della matrice di covarianza

Principal Component Analysis (PCA)

```
>> [V,D] = eig(C);
```

- `V`: matrice in cui ogni **colonna** corrisponde ad un autovettore di `C`
- `D`: matrice in cui sulla **diagonale** sono presenti gli autovalori (per estrarre la diagonale: comando `diag`)

Principal Component Analysis (PCA)

- Ordino gli autovalori dal più grande al più piccolo (comando `sort`)

```
>> [val,idx] = sort(diag(D), 'descend');
```

- `val`: vettore in cui i valori sono riordinati
- `idx`: vettore di indici con cui ordinare il vettore originale

Principal Component Analysis (PCA)

- Gli autovettori corrispondenti agli autovalori più grandi codificano le direzioni principali
  - Devo applicare lo stesso sorting agli autovettori
  - Suggerimento: usare la variabile `idx`
- La matrice di trasformazione A, che useremo per ridurre i dati a L dimensioni, è composta dai primi L autovettori (prime colonne di V), corrispondenti agli L autovalori più grandi.

Principal Component Analysis (PCA)

- Data la matrice A di trasformazione, per applicarla ai dati, proiettandoli nel nuovo spazio, si può eseguire il seguente comando:

```
>> Y = A' * datac;
```

Esercizio

- Implementare la PCA
- Qualche suggerimento nello script Lezione3Lab_es2.m