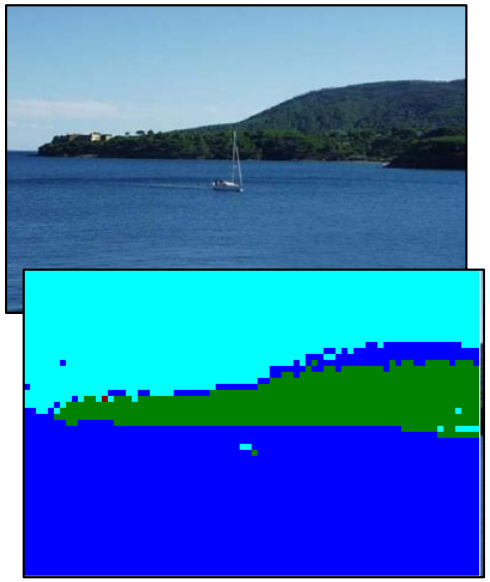
Tecniche di Classificazione

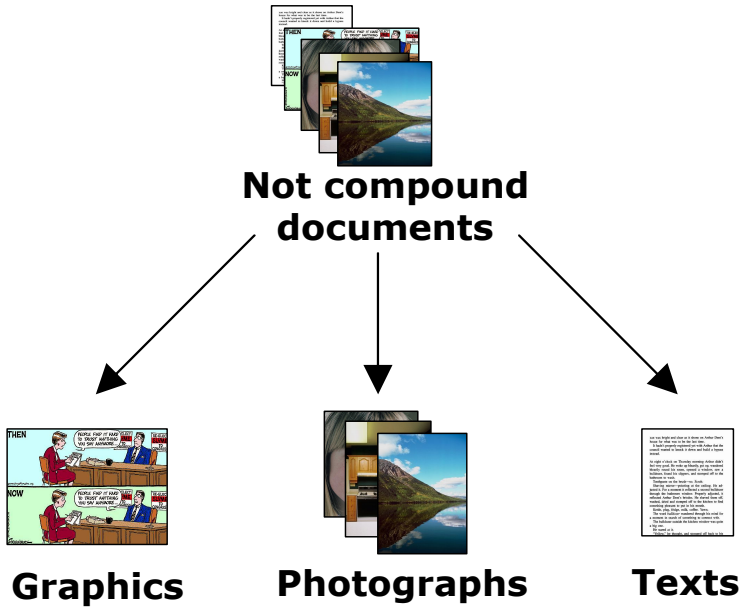
Elaborazione delle Immagini - Complementi

Gianluigi Ciocca

Tecniche di Classificazione

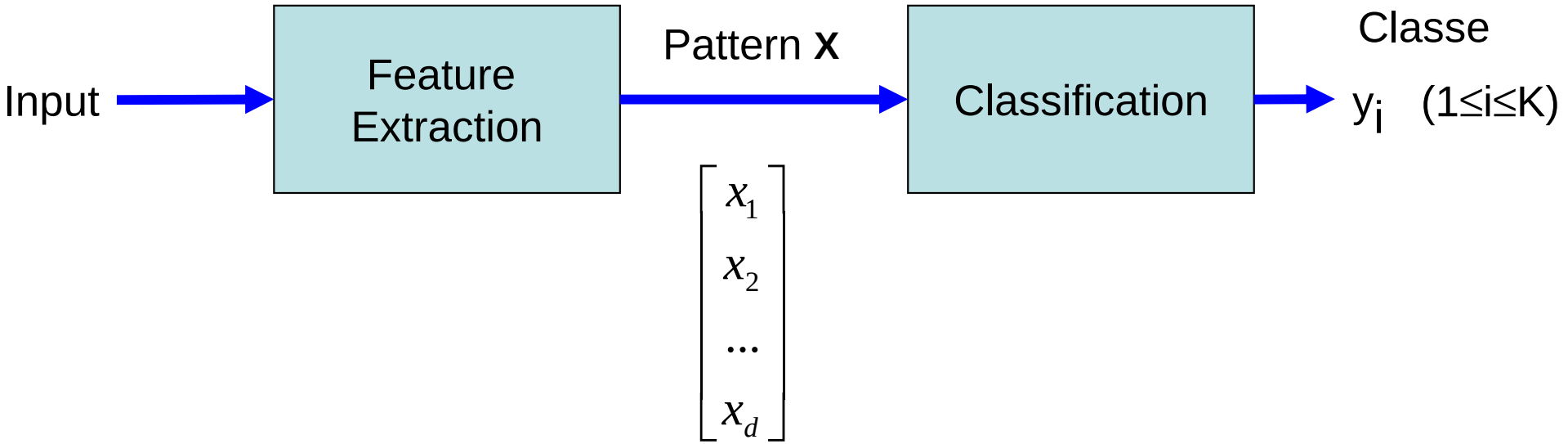
- Tutte quelle metodologie ed algoritmi che permettono di inferire informazioni a partire da un insieme di dati.
- In particolare, dato un insieme di classi indicare a quale classe appartiene un nuovo elemento





Classificazione

- L'oggetto da classificare deve essere rappresentato da un ***pattern***
  - una disposizione di descrittori (feature), cioè un insieme di valori, che descrivono l'input, organizzati in una qualche struttura.



Classificazione

- Il classificatore può essere visto come una funzione *f*:

$$f: X \rightarrow Y$$

- **X** è lo spazio dei pattern o ***spazio delle feature***.
- **Y** è lo spazio delle classi: **Y**={1,…,k}

- La funzione f si può trovare mediante tecniche di ***apprendimento supervisionato***
  - Apprendimento di regole o funzioni basate sull'analisi di un insieme di pattern di classe nota (***training set***).

Classificazione

- Dato un training set:

$$T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\} \qquad \mathbf{x}_i \in X, y_i \in Y$$

- di coppie estratte secondo una qualche distribuzione di probabilità non nota su $X \times Y$.

- si desidera trovare una funzione $f: X \rightarrow Y$ che sia in grado di generalizzare l'associazione della classe corretta anche ai pattern non presenti nel training set.

Classificazione

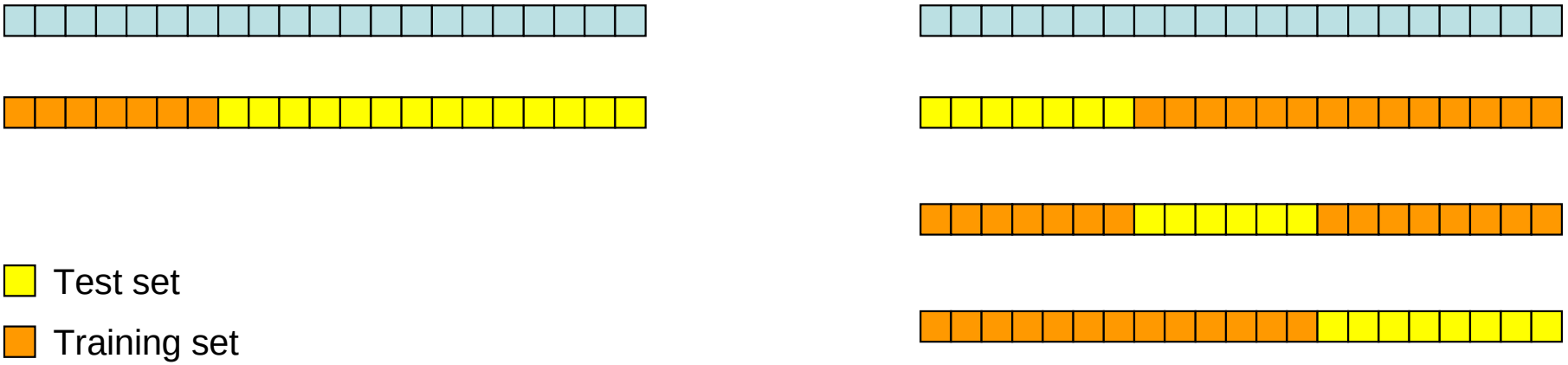
- Definizione del training set
  - definizione delle classi, selezione dei dati e annotazione dei casi con le rispettive classi.
- Scelta della strategia di validazione
  - test del classificatore su nuovi dati
- Scelta dei descrittori
  - si usa la conoscenza a priori del problema di classificazione per selezionare le feature più adatte.
- Scelta della strategia di classificazione
  - in base alle conoscenze del problema,
  - può dipendere anche da vincoli implementativi (requisiti di tempo e/o spazio) o dalla tipologia di descrittori utilizzati.

Scelta del training set

- Definizione del data set
  - Definizione delle classi, selezione dei dati e annotazione dei casi con le rispettive classi.
- Training set
  - Porzione del data set usato per costruire il classificatore
  - Rappresentativo dei dati e delle classi
- Test set
  - Porzione del data set usato per validare il classificatore

Scelta della strategia di Validazione

- Validazione
  - Applicare il classificatore sui nuovi dati
  - **single test set**
  - **cross validation**



Scelta delle feature

- Le feature possono riguardare diversi aspetti di un'immagine (o di una sua porzione):
  - Colore
  - Forma degli oggetti rappresentati (edge)
  - Caratteristiche delle regioni (segmentazione)
  - Tessitura
  - …
- L'intera immagine è anch'essa una feature.

Scelta delle feature

- La scelta delle feature deve tenere in considerazione le seguenti proprietà:
  - **Discriminazione**: i valori di una feature dovrebbero essere significativamente diversi per oggetti appartenenti a classi diverse.
  - **Affidabilità**: le feature dovrebbero assumere valori simili per oggetti appartenenti alla stessa classe.
  - **Indipendenza**: dovrebbero essere indipendenti l'una dall'altra.
  - **Cardinalità**: un numero eccessivo di feature può rendere difficoltoso l'apprendimento oltre ad essere poco efficiente.

Approccio statistico

- Il problema di classificazione viene espresso tramite un modello statistico
- Un training set viene usato per stimare i parametri del modello
- E’ necessario trovare la regola ottimale che permette di stabilire se un campione appartiene o meno al modello

Approccio statistico

- Si indica con $P(y=i)$ la probabilità con cui si presenta la classe i: ***probabilità a priori***

$$\sum_{i=1}^{k} P(y=i) = 1$$

- Si indica con con $P(y=i \mid \mathbf{x})$ la probabilità che un campione **x** sia di classe i: ***probabilità a posteriori***

$$\sum_{i=1}^{k} P(y=i \mid \mathbf{x}) = 1$$

Approccio statistico

- Intuitivamente, la regola di decisione più razionale è:

$$f(\mathbf{x}) = i \Leftrightarrow P(y = i \mid \mathbf{x}) \geq P(y = j \mid \mathbf{x}), \forall j$$

- Chiamato anche Maximum a Posteriori (MAP)

- Questa regola minimizza l'errore di classificazione per un dato pattern **x**.

Regola di Bayes

- Le probabilità a posteriori non sono direttamente ricavabili
- Per ottenere una loro stima conviene riscriverle applicando la **regola di Bayes**:

$$p(y=i,\mathbf{x}) = P(y=i\,|\,\mathbf{x})\,p(\mathbf{x}) = p(\mathbf{x}\,|\,y=i)P(y=i)$$

- $p(\mathbf{x} \mid y=i)$ è detta **verosimiglianza** ed è la distribuzione dei pattern di classe *i*
- $p(\mathbf{x})$ è chiamata **evidenza**, e vale:

$$p(\mathbf{x}) = \sum_{j=1}^{k} p(\mathbf{x}\,|\,y=j)P(y=j)$$

Regola di Bayes

- Le probabilità a posteriori sono quindi esprimibili come:

$$P(y=i\,|\,\mathbf{x})=\frac{p(\mathbf{x}\,|\,y=i)P(y=i)}{p(\mathbf{x})}$$

- Dato che il denominatore è indipendente dalla classe, si può definire la seguente **regola di decisione di Bayes**:

$$f(\mathbf{x})=i \Leftrightarrow p(y=i\,|\,\mathbf{x}) \geq p(y=j\,|\,\mathbf{x}), \forall j$$

$$f(\mathbf{x})=i \Leftrightarrow p(\mathbf{x}\,|\,y=i)P(y=i) \geq p(\mathbf{x}\,|\,y=j)P(y=j), \forall j$$

Regola di Bayes

$$f(\mathbf{x}) = i \Leftrightarrow p(\mathbf{x} \mid y = i)P(y = i) \geq p(\mathbf{x} \mid y = j)P(y = j), \forall j$$

- La verosimiglianza e le probabilità a priori possono essere
  - Note
  - Ipotizzabili
  - Stimabili da un training set

Funzioni discriminanti

- I classificatori sono spesso definiti in termini di **funzioni discriminanti**:

$$f(\mathbf{x}) = i \Leftrightarrow g_i(\mathbf{x}) \geq g_j(\mathbf{x}), \forall j$$

- Per la regola di decisione di Bayes:

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid y = i) P(y = i)$$

- Applicando una funzione strettamente crescente è possibile ottenere delle funzioni discriminanti equivalenti. Es:

$$g'_i(\mathbf{x}) = \log(p(\mathbf{x} \mid y = i)) + \log(P(y = i))$$

Funzioni discriminanti

- Le regioni dei punti in cui la funzione discriminante che assume valore massimo non è unica, vengono chiamate **superfici di decisione**.
  - Sono le regioni di confine (separazione) tra le classi

Classificatore Bayesiano

- Segue la regola di decisione definita tramite le funzioni di discriminazione:

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid y = i)P(y = i)$$

- Le probabilità a priori sono (di solito) facilmente stimabili
- La verosimiglianza può essere stimata se facciamo l'ipotesi che segua una qualche distribuzione di probabilità nota:

$$p(\mathbf{x} \mid y = i) \sim D(\mathbf{x}; \boldsymbol{\alpha})$$

- I parametri α vengono stimati sulla base del training set

La distribuzione Normale

- La distribuzione normale (o Gaussiana) viene usata molto spesso per approssimare la verosimiglianza:
  - Molti fenomeni naturali seguono la distribuzione normale
  - In molti problemi si può considerare una classe come un insieme di pattern ottenuti tramite perturbazioni di un ***prototipo***

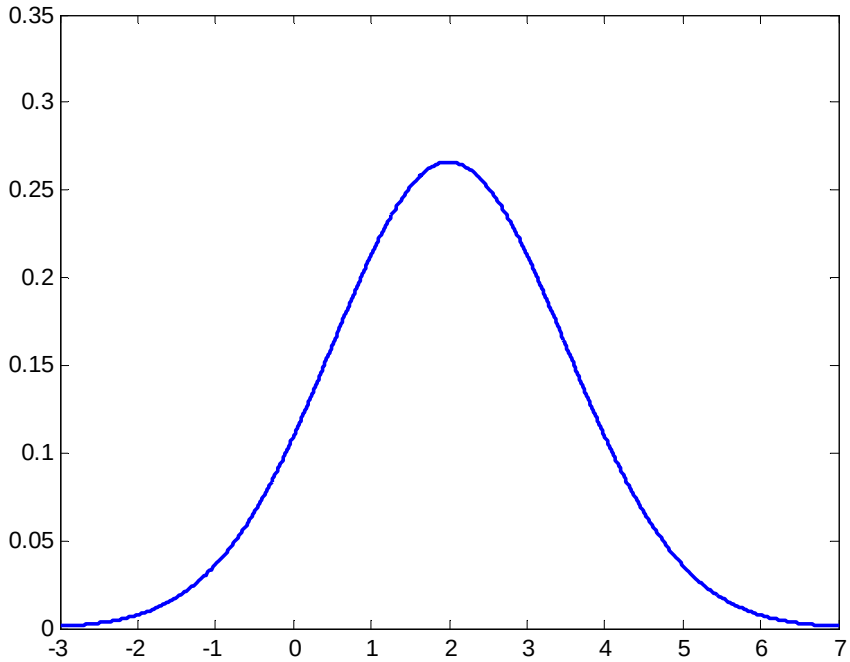
La distribuzione Normale

- Caso monodimensionale:

$$N(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2} \right)$$

μ valore atteso

σ^2 varianza



La distribuzione Normale

- Caso multivariato (n-dimensionale):

$$N(\mathbf{x};\boldsymbol{\mu},\Sigma)=\frac{1}{\det(\Sigma)^{1/2}(2\pi)^{d/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{T}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

μ valore atteso

Σ matrice di covarianza

d=dimensione del vettore di features

$$\Sigma_{ij}=\mathrm{Cov}[x_i,x_j]=E[(x_i-\mu_i)(x_j-\mu_j)] \qquad \Sigma_{ii}=Var[x_i]$$

Classificatore Bayesiano

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid y = i) P(y = i)$$

- Si supponga che le verosimiglianze seguano delle distribuzioni normali:

$$p(\mathbf{x} \mid y = i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$$

- I parametri delle distribuzioni (condizionati dalla classe) possono essere stimati sul training set:

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{N_i} \sum_{j:y_j=i} \mathbf{x}_j \qquad \hat{\Sigma}_i = \frac{1}{N_i} \sum_{j:y_j=i} (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_i)(\mathbf{x}_j - \hat{\boldsymbol{\mu}}_i)^T \qquad N_i = \sum_{j:y_j=i} 1$$

Classificatore Bayesiano
Features indipendenti con varianze uguali

- Introduciamo alcune semplificazioni:
  - Le features sono tra loro indipendenti (=covarianza nulla)
  - Le features sono distribuite con uguale varianza all'interno delle classi.

$$\Sigma_i = \mathbf{I}\sigma^2$$

$$\det(\Sigma_i) = \sigma^{2d}$$

$$\Sigma^{-1} = \frac{\mathbf{I}}{\sigma^2}$$

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid y = 1)P(y = i)$$

$$= \frac{1}{\sigma^d (2\pi)^{d/2}} \exp\left(-\frac{1}{2} \frac{(\mathbf{x} - \boldsymbol{\mu}_i)^T (\mathbf{x} - \boldsymbol{\mu}_i)}{\sigma^2} \right) P(y = i)$$

Classificatore Bayesiano

Features indipendenti con varianze uguali

$$g_i(\mathbf{x}) = \frac{1}{\sigma^d (2\pi)^{d/2}} \exp\left(-\frac{1}{2} \frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T (\mathbf{x}-\boldsymbol{\mu}_i)}{\sigma^2} \right) P(y=i)$$

- Applichiamo una trasformazione logaritmica (crescente) a tutte le funzioni discriminanti:

$$g'_i(\mathbf{x}) = \log(g_i(\mathbf{x})) =$$

$$= -d \log \sigma - \frac{d}{2} \log(2\pi) - \frac{1}{2} \frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T (\mathbf{x}-\boldsymbol{\mu}_i)}{\sigma^2} + \log(P(y=i))$$

Classificatore Bayesiano
Features indipendenti con varianze uguali

- Poiché sono interessato solo all'ordine relativo dei valori delle funzioni discriminanti, elimino i termini indipendenti dalla classe *i:*

$$-d\log\sigma - \frac{d}{2}\log(2\pi) - \frac{1}{2}\frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T(\mathbf{x}-\boldsymbol{\mu}_i)}{\sigma^2} + \log(P(y=i))$$

- Se supponiamo che le probabilità a priori siano tutte uguali (P(y=i)=1/k), possiamo eliminare anche l'ultimo termine:

$$g'_i(\mathbf{x}) = -(\mathbf{x}-\boldsymbol{\mu}_i)^T(\mathbf{x}-\boldsymbol{\mu}_i) = -\left\|\mathbf{x}-\boldsymbol{\mu}_i\right\|^2$$

La funzione discriminante è la distanza Euclidea del campione dal vettore delle medie!

Classificatore a minima distanza

- Dato un campione **x**, si calcolano le distanze del campione dalle medie (prototipi delle classi) μ_i

- Il classificatore sceglie la classe che ha dato distanza minore

$$g'_c(\mathbf{x}) \geq g'_j(\mathbf{x}) \quad \forall j \neq c$$

$$-\left\|\mathbf{x}-\boldsymbol{\mu}_c\right\|^2 \geq -\left\|\mathbf{x}-\boldsymbol{\mu}_j\right\|^2 \quad \forall j \neq c$$

$$\left\|\mathbf{x}-\boldsymbol{\mu}_c\right\|^2 \leq \left\|\mathbf{x}-\boldsymbol{\mu}_j\right\|^2 \quad \forall j \neq c$$

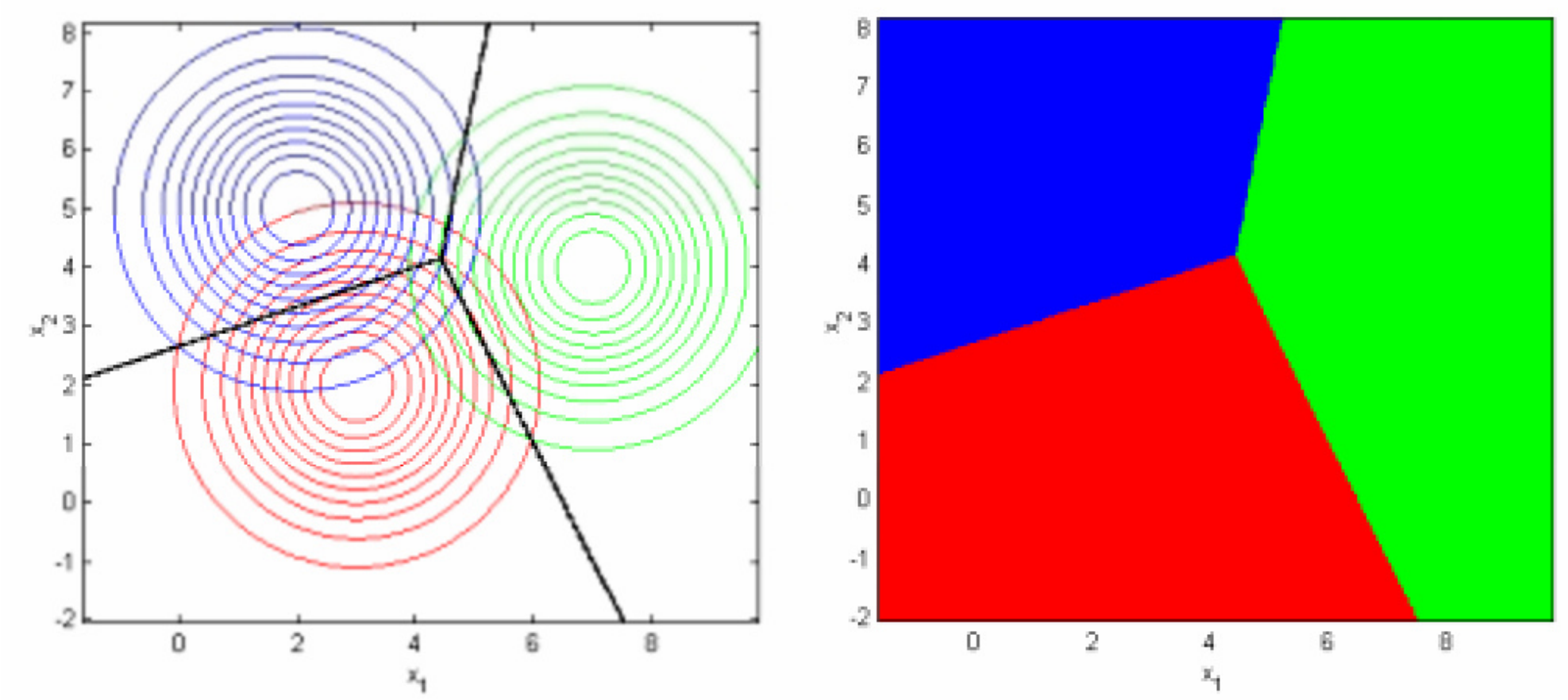
Classificatore a minima distanza

- Riassumendo, il classificatore a minima distanza è il classificatore Bayesiano sotto le seguenti ipotesi:
  - Features distribuite con uguale varianza all'interno delle classi.
  - Features statisticamente indipendenti.
  - Probabilità a priori uguali per tutte le classi.

- Nota: le funzioni discriminanti sono lineari

$$g'_i(\mathbf{x}) = -\|\mathbf{x}-\boldsymbol{\mu}_i\|^2 = -\|\mathbf{x}\|^2 - \|\boldsymbol{\mu}_i\|^2 + 2\mathbf{x}^{\mathrm{T}}\boldsymbol{\mu}_i$$

Classificatore a minima distanza



Classificatore a minima distanza

- Quando i range delle features sono diversi tra loro, la distanza Euclidea non rappresenta efficacemente le differenze tra i pattern

- Le features possono essere normalizzate affinché i range di variabilità siano uniformi:
  - Normalizzazione nell'intervallo [0,1]

$$\mathbf{x}_i' = \frac{\mathbf{x}_i - \min_i}{\max_i - \min_i}$$

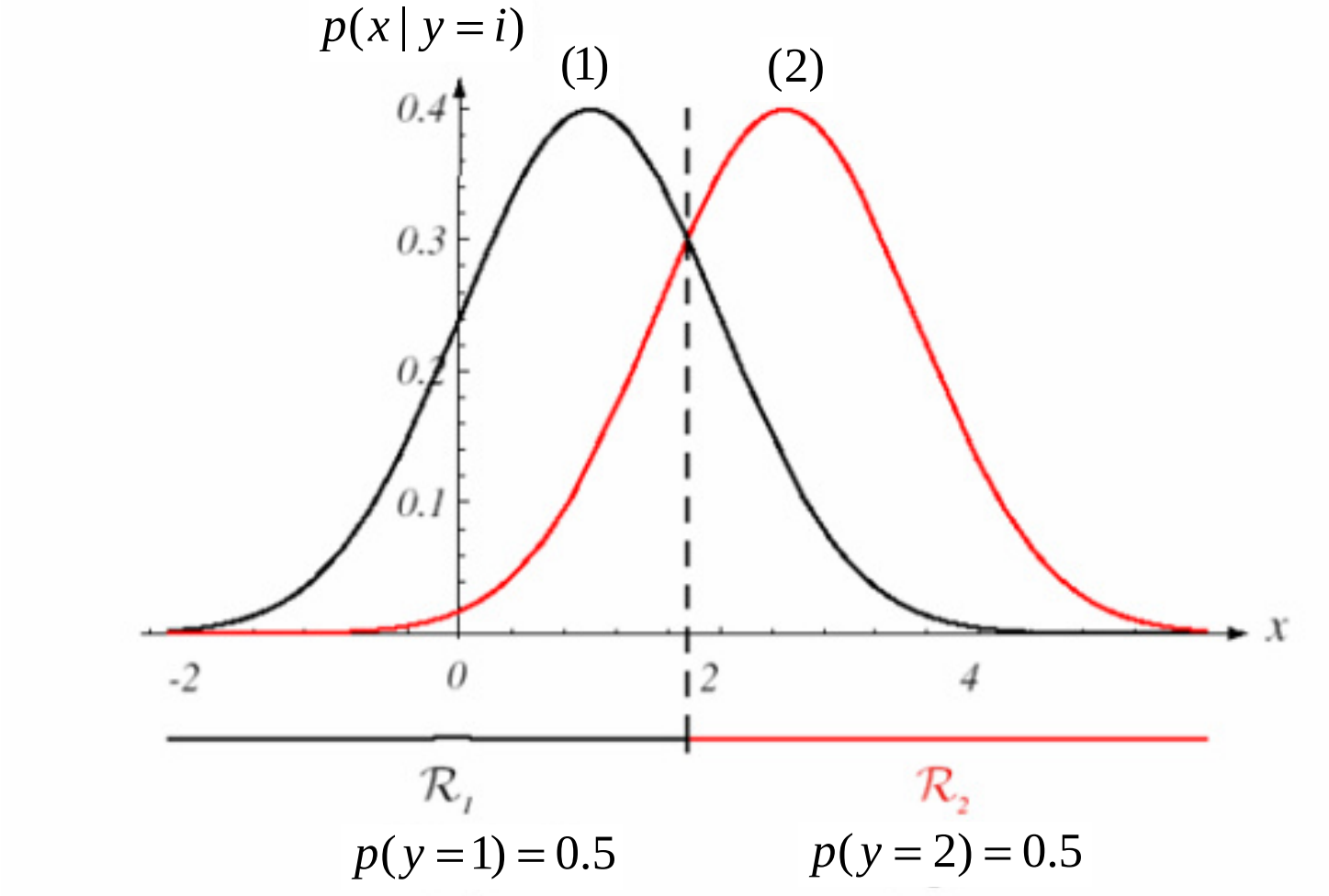
  - Normalizzazione di media e varianze

$$\mathbf{x}_i' = \frac{\mathbf{x}_i - \mu_i}{\sigma_i}$$

- I parametri (minimo e massimo, media e deviazione) devono essere stimati su un training set

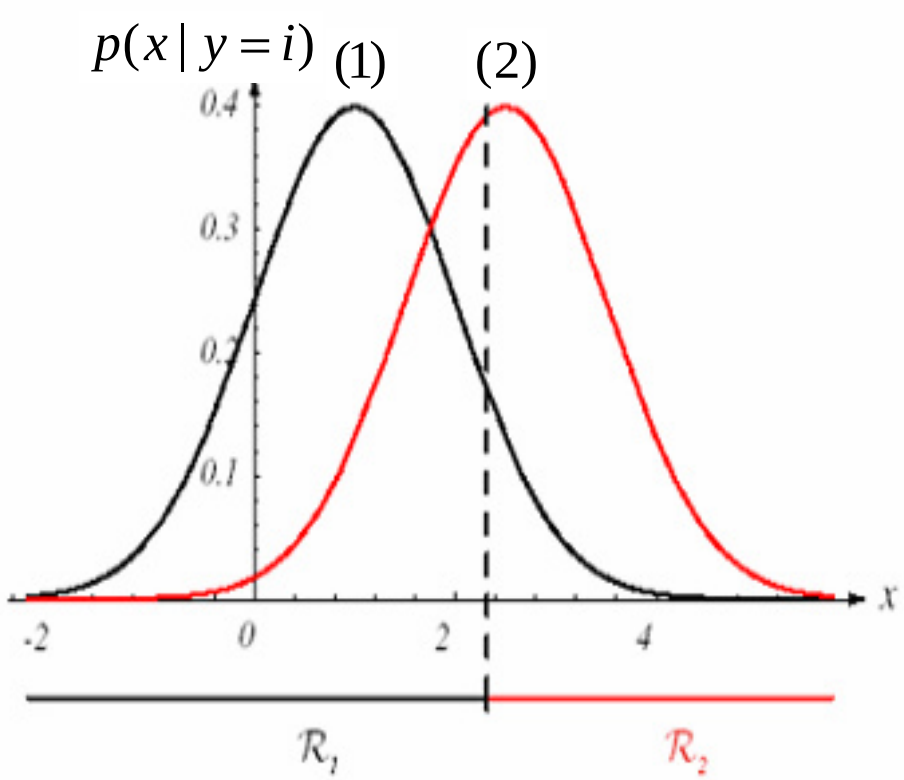
Distribuzione normale:

Features indipendenti con varianze uguali, P(y=i) uguali

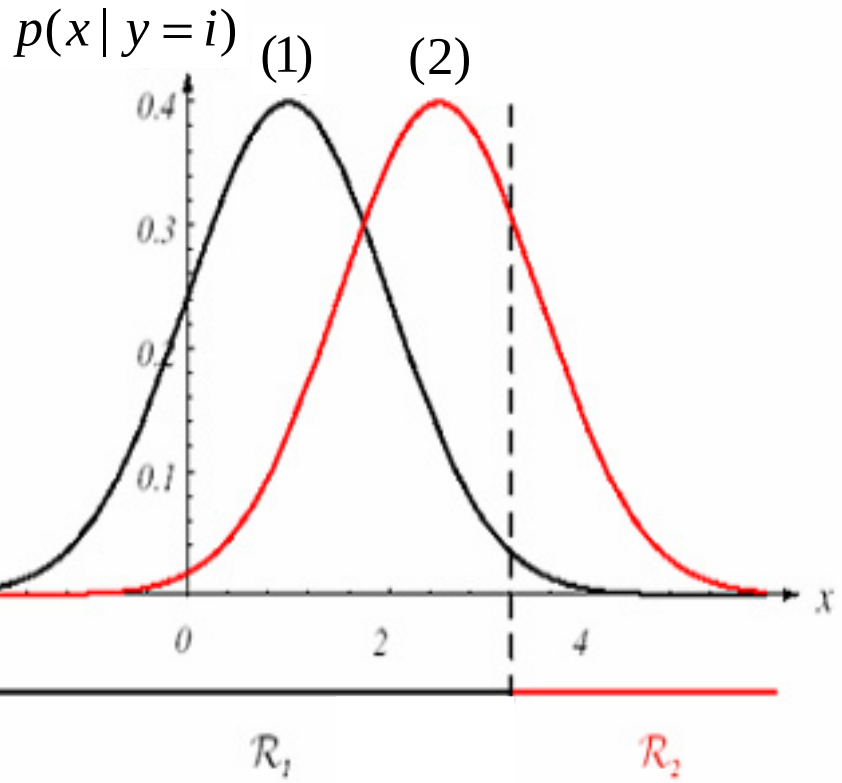


Distribuzione normale:

Features indipendenti con varianze uguali, P(y=i) diverse



$p(y=1)=0.7$ $p(y=2)=0.3$



$p(y=1)=0.9$ $p(y=2)=0.1$

Distribuzione normale:
Feature non indipendenti, covarianze uguali

- Le matrici di covarianza sono indipendenti dalla classe:

$$\Sigma_i = \Sigma$$

- Le funzioni discriminanti sono:

$$g_i(\mathbf{x}) = -\frac{1}{2}\log(\det(\Sigma)) - \frac{d}{2}\log(2\pi) - \\ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \log(P(y=i))$$

Distribuzione normale:
Feature non indipendenti, covarianze uguali, P(y=i) uguali

- Se le probabilità a priori sono identiche, si ha un classificatore a minima distanza in cui la distanza Euclidea è sostituita dalla ***distanza di Mahalanobis*****:**

$$-(\mathbf{x}-\boldsymbol{\mu}_i)^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)$$

- La distanza di Mahalanobis è una distanza “normalizzata” rispetto alle varianze delle features e alle loro correlazioni

Distribuzione normale:

Feature non indipendenti, covarianze uguali, P(y=i) uguali

- Espandendo la distanza di Mahalanobis:

$$-(\mathbf{x}-\boldsymbol{\mu}_i)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) = -\left(\cancel{\mathbf{x}^T\Sigma^{-1}\mathbf{x}} - 2\boldsymbol{\mu}_i{}^T\Sigma^{-1}\mathbf{x} + \boldsymbol{\mu}_i{}^T\Sigma^{-1}\boldsymbol{\mu}_i\right)$$

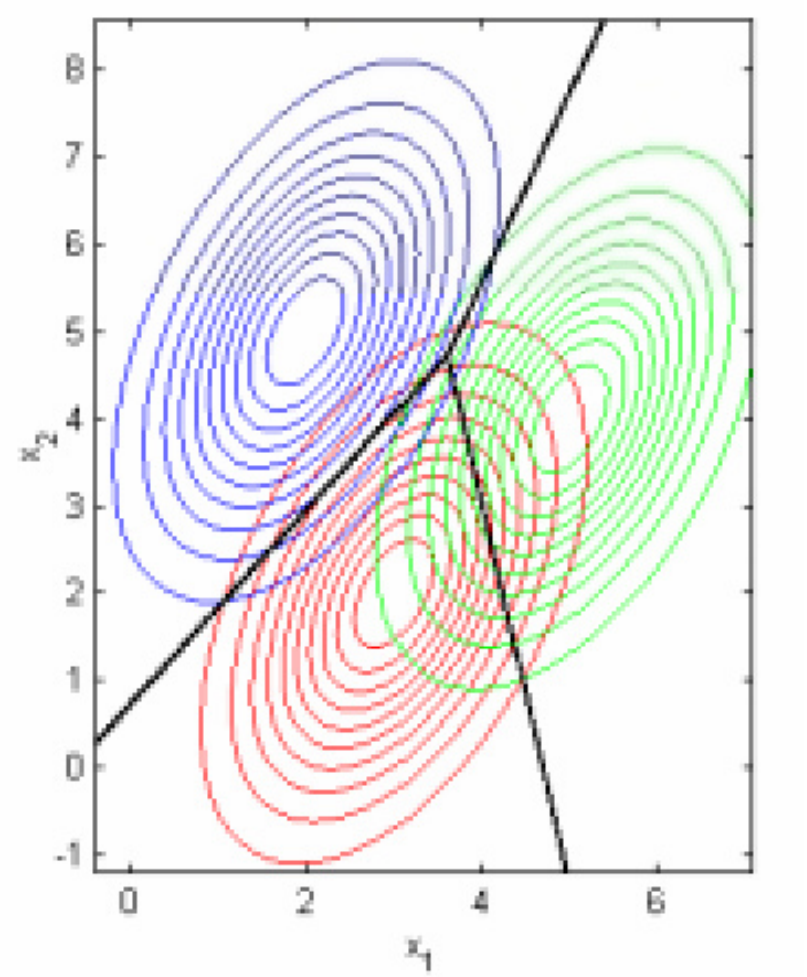
- Possiamo derivare le funzioni di discriminazione:

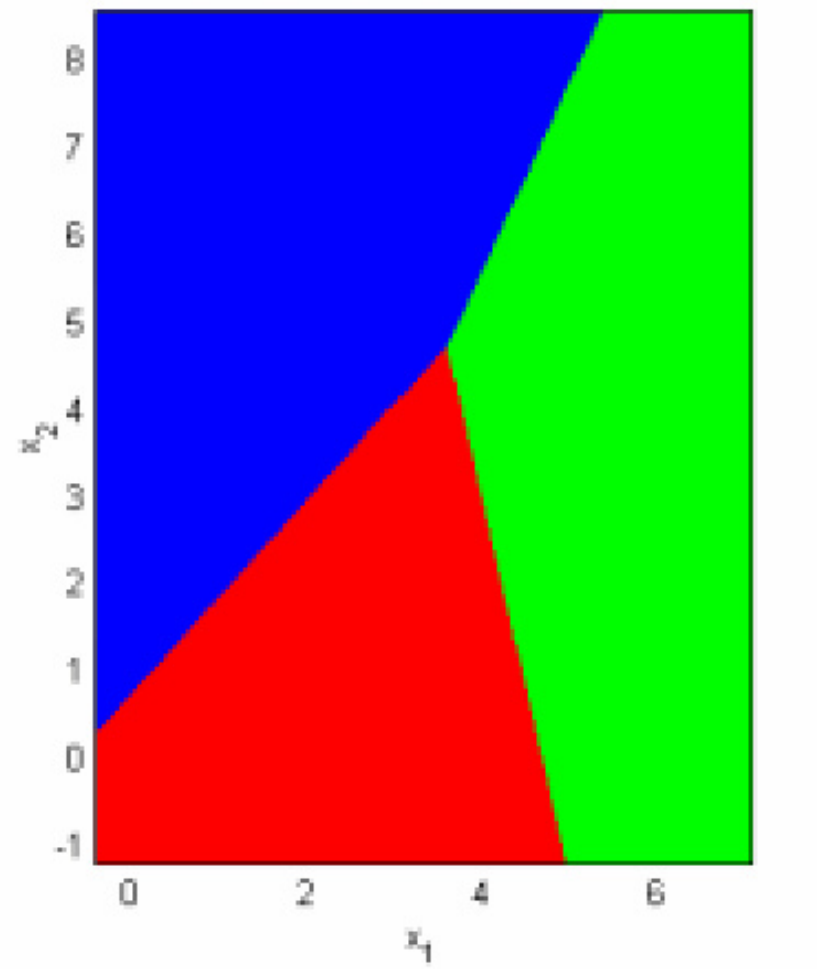
$$g_i(\mathbf{x}) = -(\boldsymbol{\mu}_i{}^T\Sigma^{-1})\mathbf{x} + \frac{\boldsymbol{\mu}_i{}^T\Sigma^{-1}\boldsymbol{\mu}_i}{2}$$

- che sono lineari

Distribuzione normale:

Feature non indipendenti, covarianze uguali, P(y=i) uguali





Distribuzione normale:
Feature non indipendenti, covarianze diverse

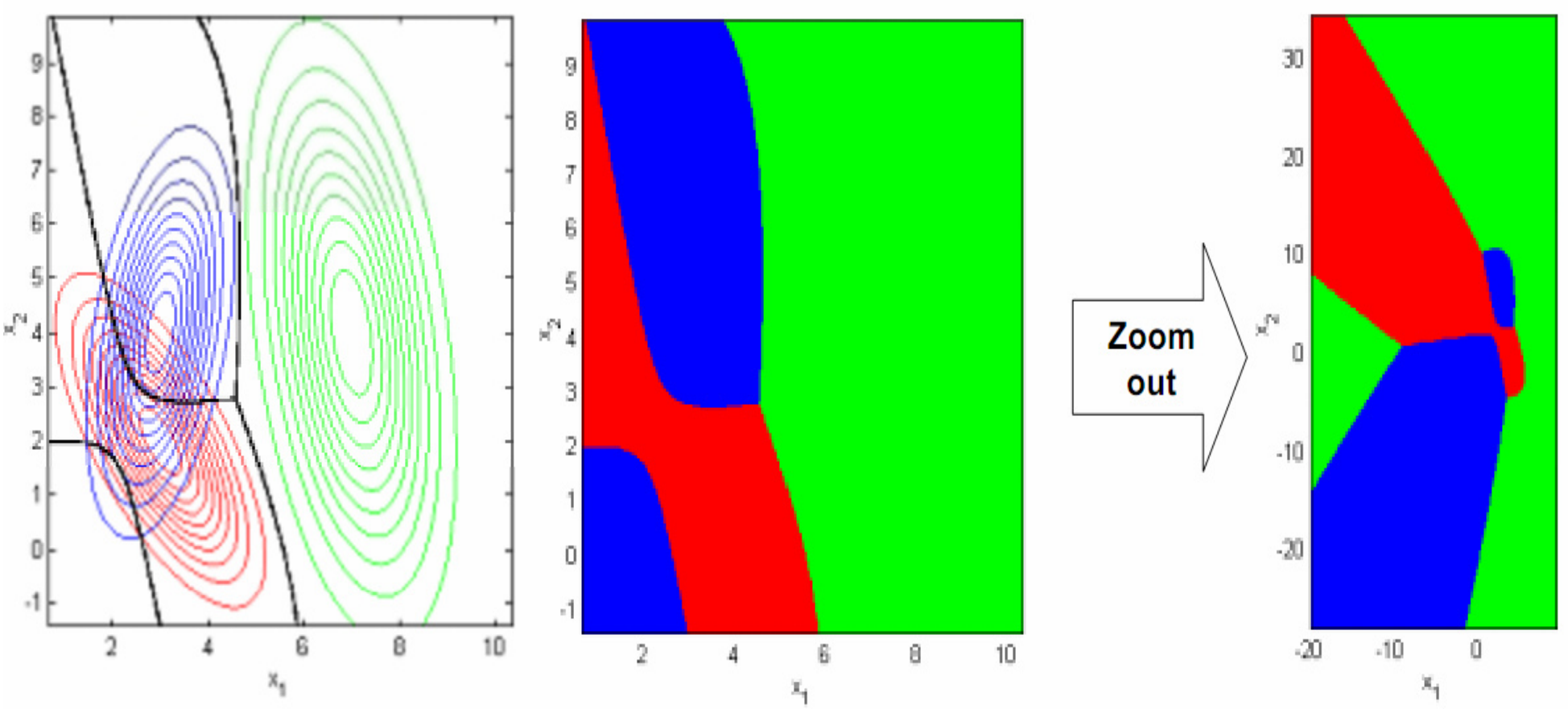
- E’ il caso generale

- In questo caso si può eliminare solo un termine:

$$g_i(\mathbf{x}) = -\frac{1}{2}\log\det(\Sigma_i) - \cancel{\frac{d}{2}\log(2\pi)} - \\ -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \log(P(y=i))$$

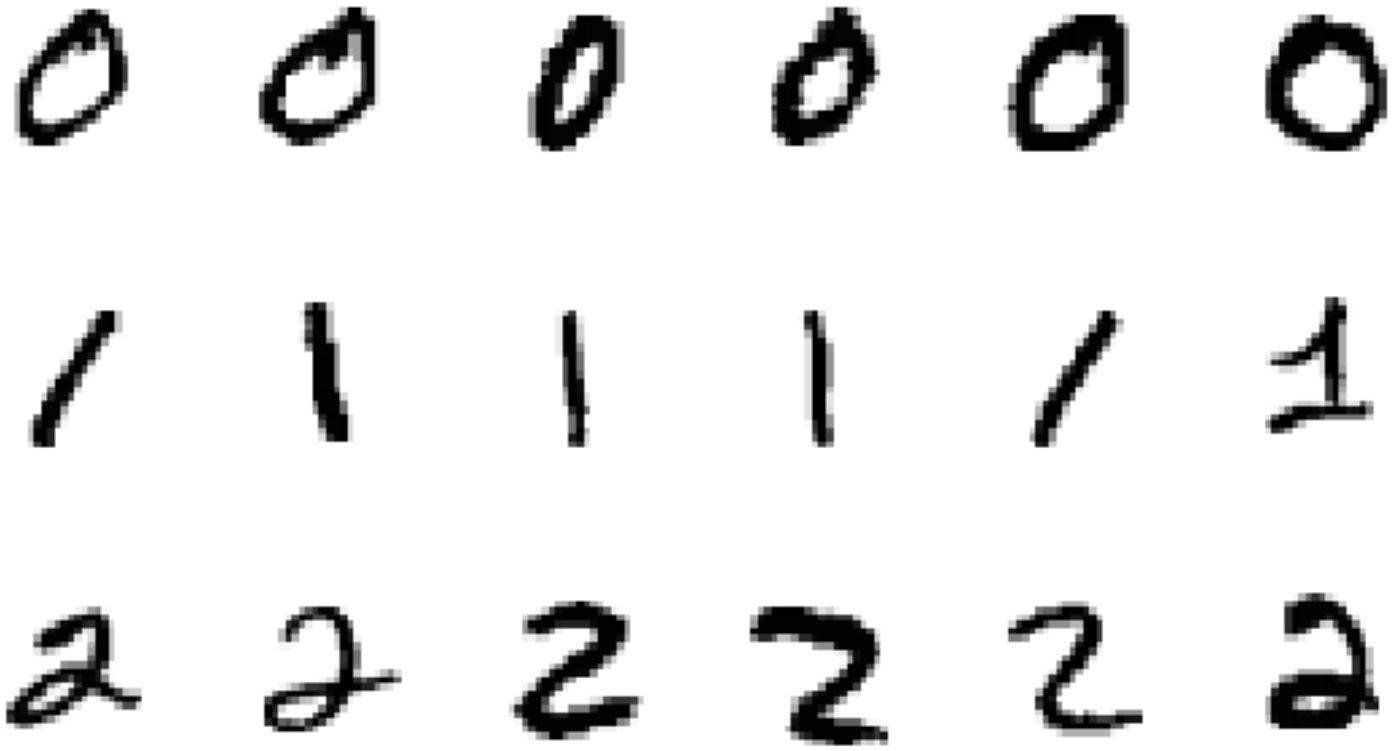
- Le funzioni discriminanti in questo caso sono quadratiche.

Distribuzione normale:
Feature non indipendenti, covarianze diverse



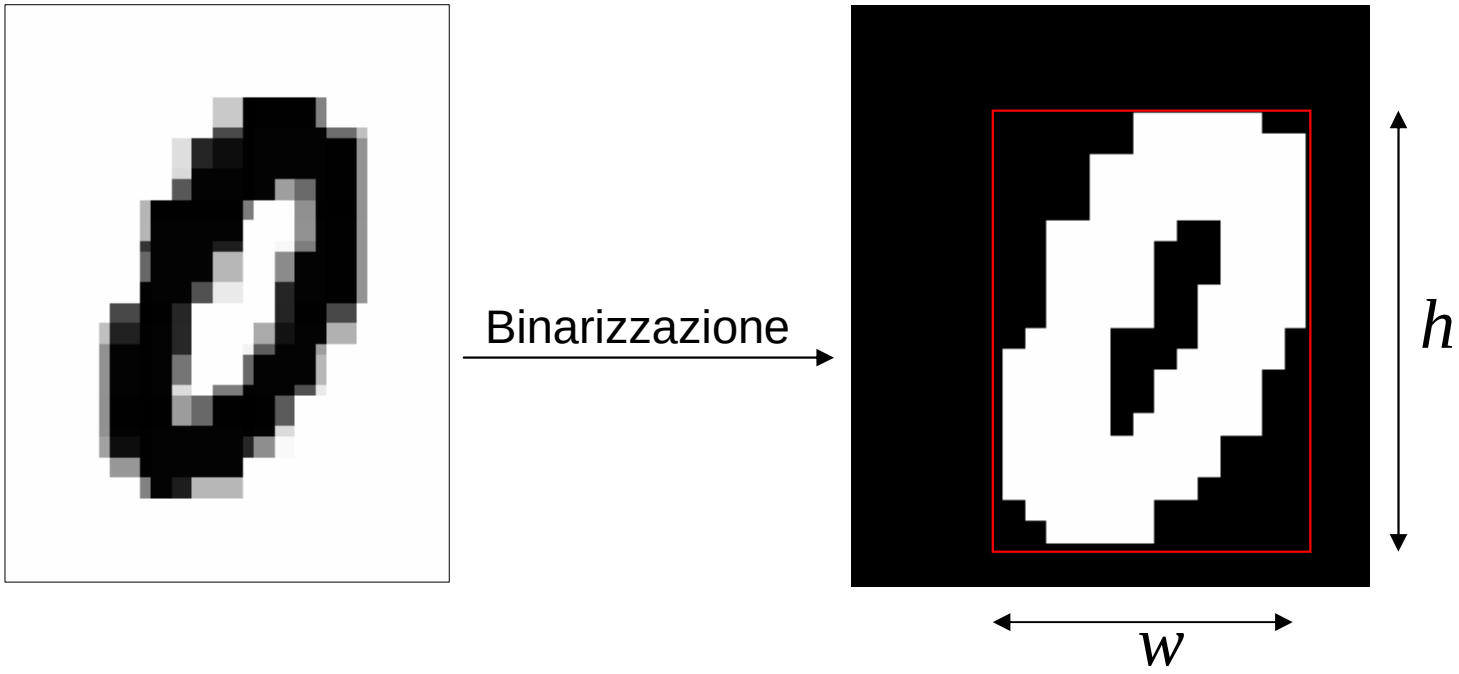
Riconoscimento di cifre

- Determinare se l'immagine di una cifra scritta a mano corrisponde ai numeri '0', '1' o '2' (classi equidistribuite)



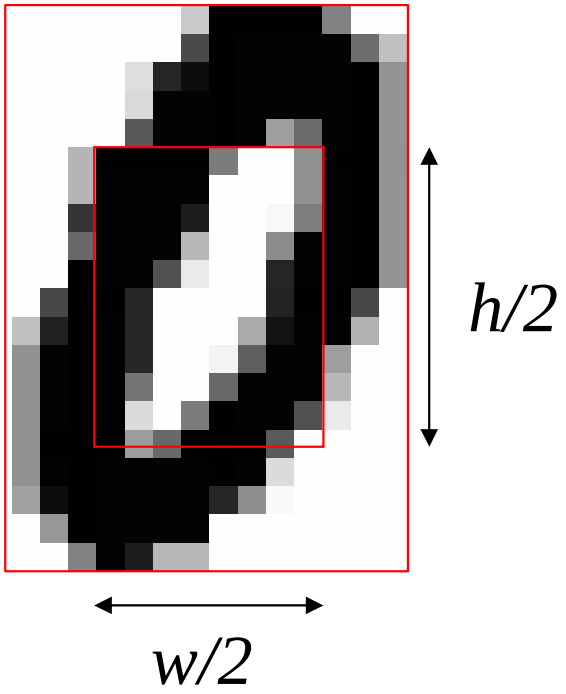
Riconoscimento di cifre

- La prima feature considerata è il rapporto tra l'altezza del bounding box e il suo semiperimetro
- Aspect ratio = *h/(w+h)*



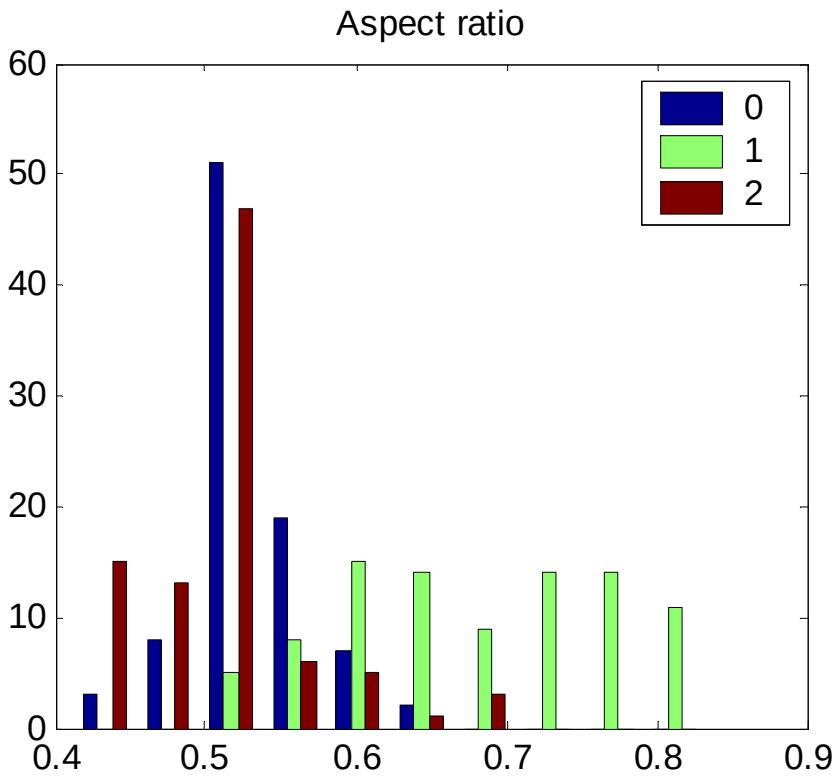
Riconoscimento di cifre

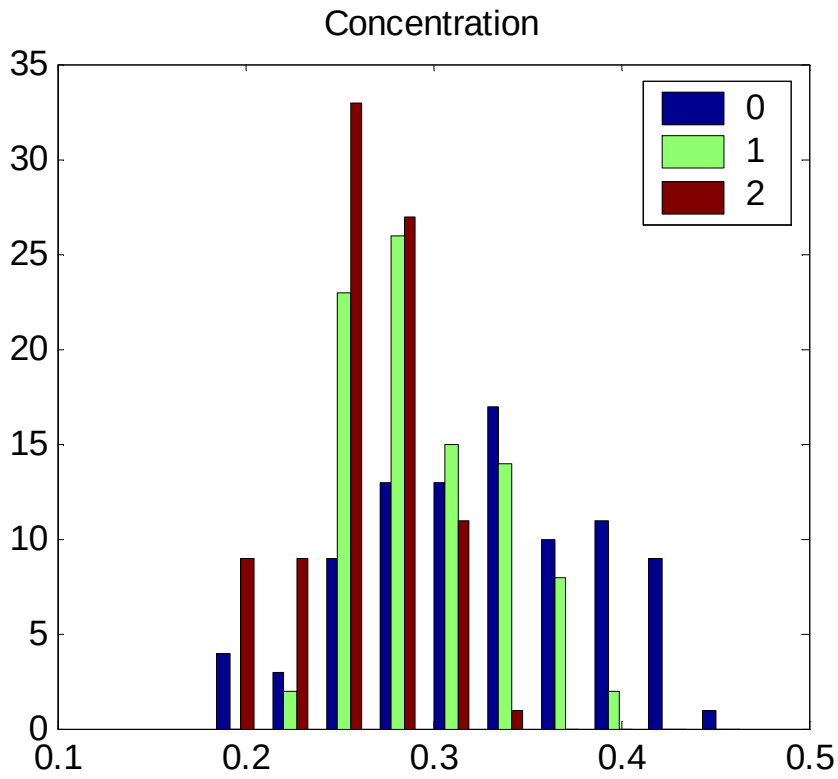
- La seconda feature rappresenta la frazione di intensità nell'immagine dei pixels nel centro del carattere
- Concentration = somma delle intensità nell'area centrale / somma delle intensità nel bounding box



Riconoscimento di cifre

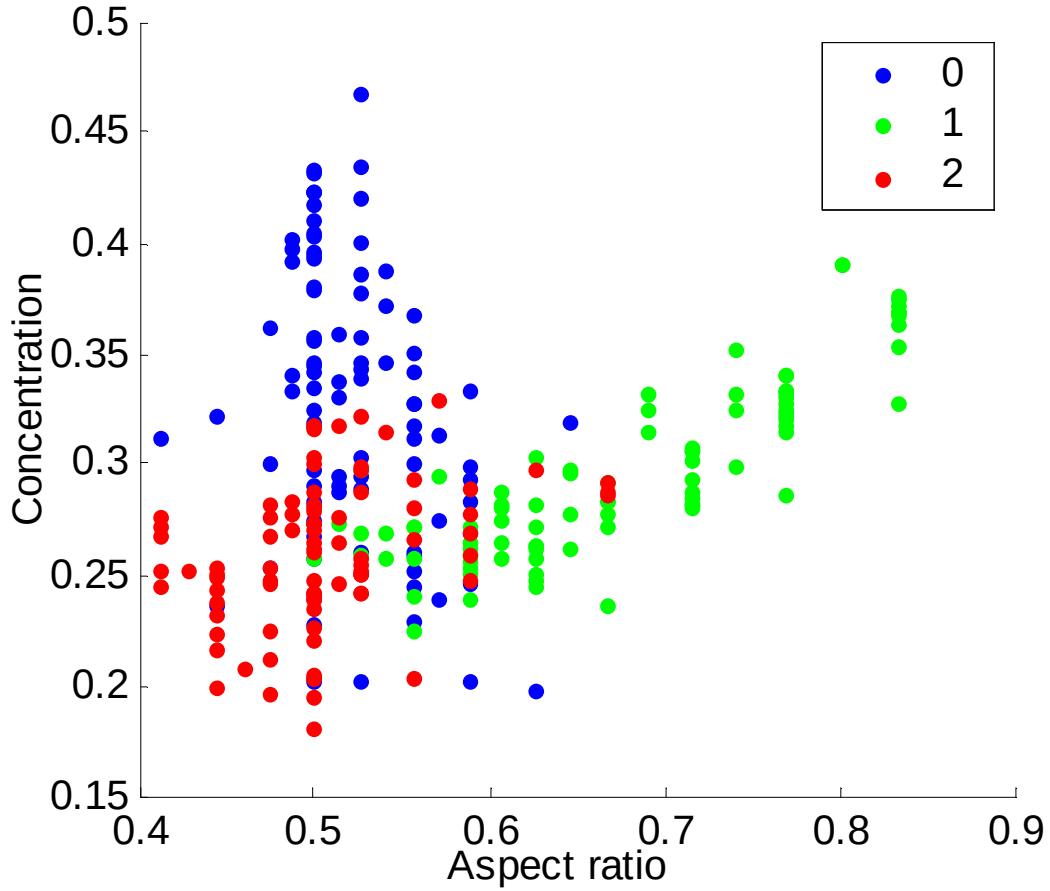
- Distribuzione delle features nelle tre classi



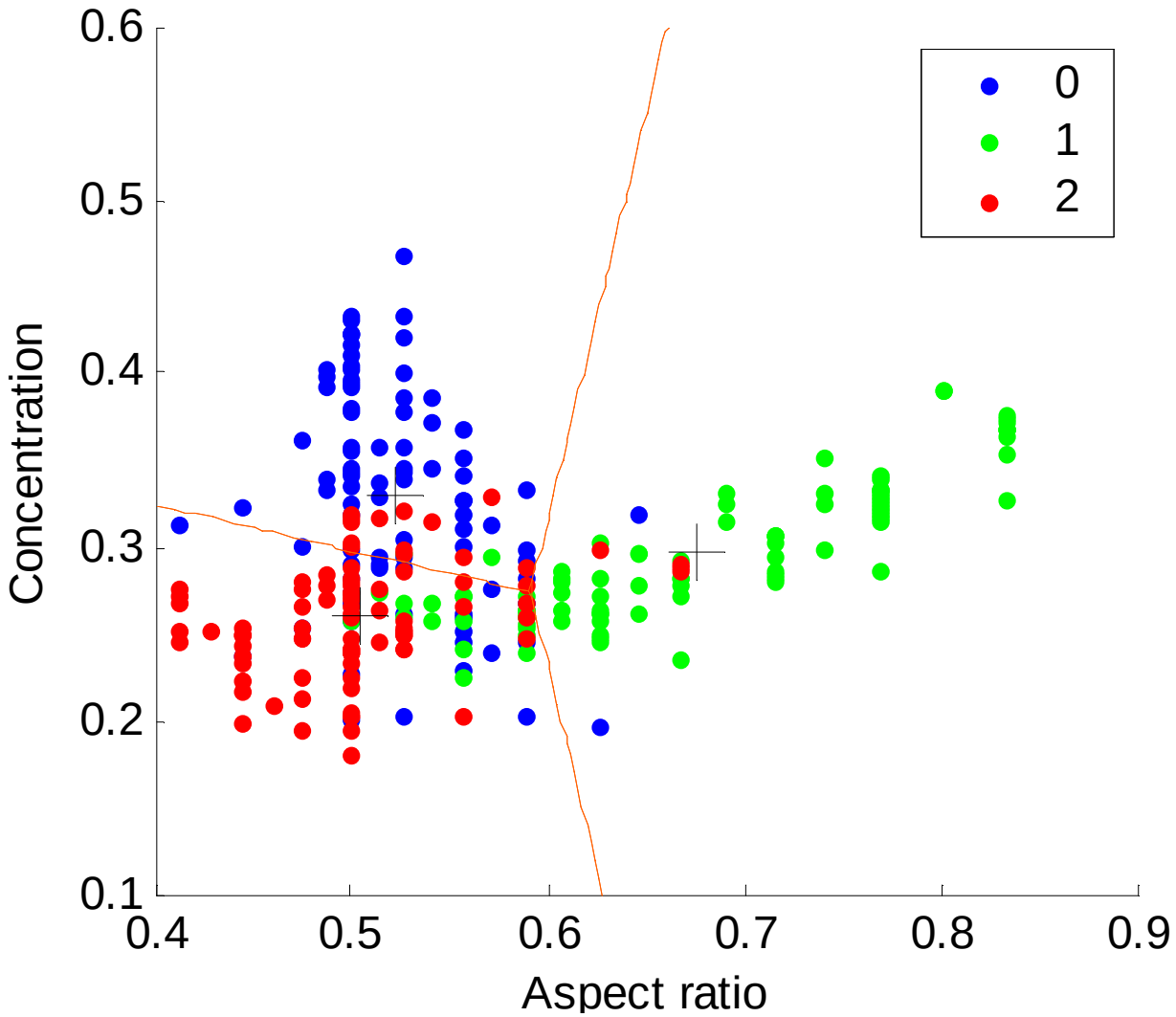


Distribuzione delle features

- Rappresentazione nello spazio delle features



Ipotesi 1
Classificatore a minima distanza



Feature indipendenti
Varianze uguali
Prob. priori uguali

Errori:

classe 0: 31%

classe 1: 25%

classe 2: 22%

err. medio: 26%

Ipotesi 2
Distanza di Mahalanobis

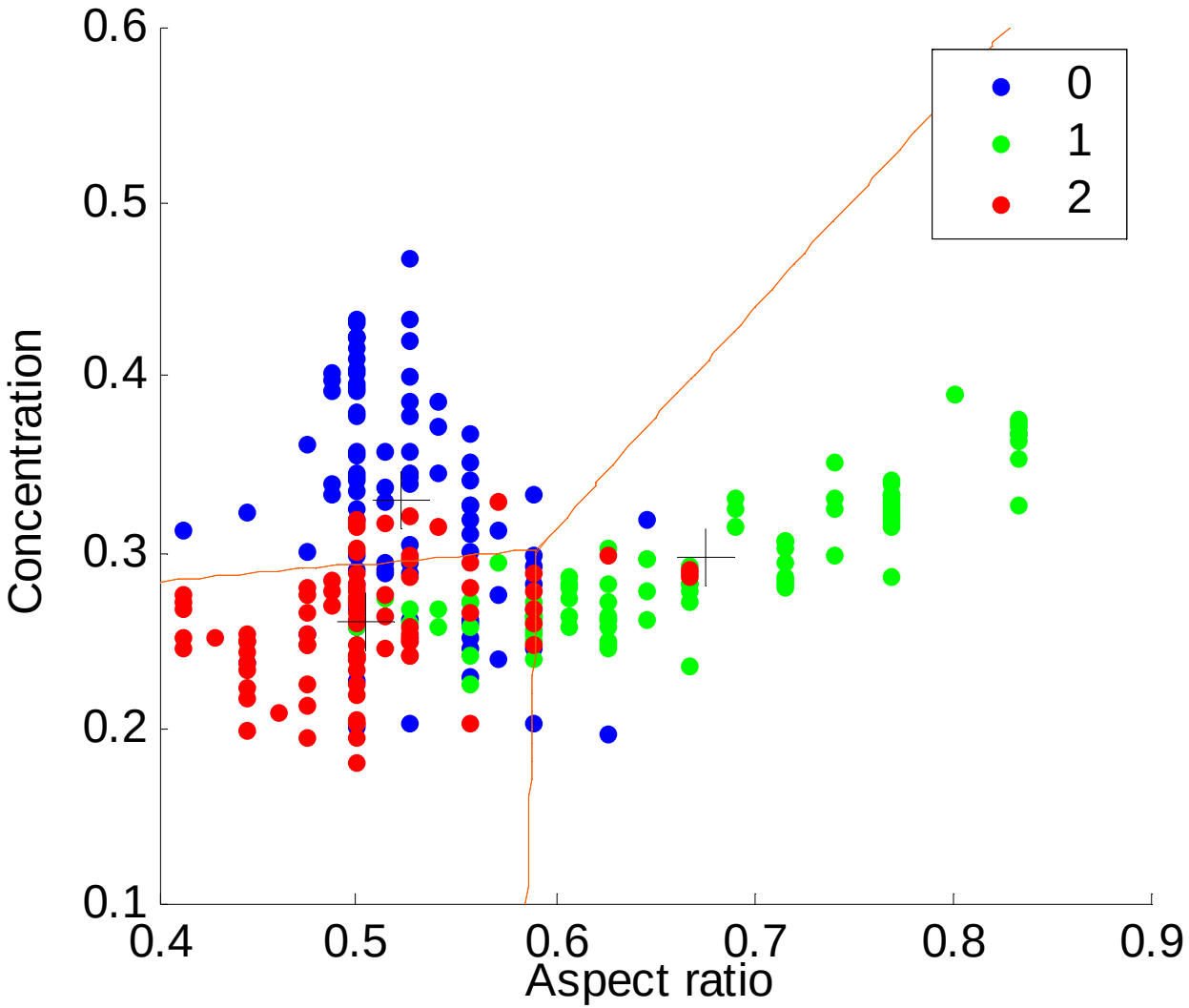
- Le due feature mostrano un certo grado di correlazione, come mostrato dalla matrice di covarianza e dal coefficiente di correlazione:

$$\Sigma = \begin{bmatrix} 0.0043 & 0.0010 \\ 0.0010 & 0.0022 \end{bmatrix}$$

$$corr = 0.3251$$

- L'Aspect ratio ha una varianza maggiore della concentrazione
  - differenze in concentrazione sono più significative

Ipotesi 2
Distanza di Mahalanobis



Feature dipendenti
Covarianze uguali
Prob. priori uguali

Errori:

classe 0: 33%

classe 1: 25%

classe 2: 18%

err. medio: 25%

Ipotesi 3
Caso generale

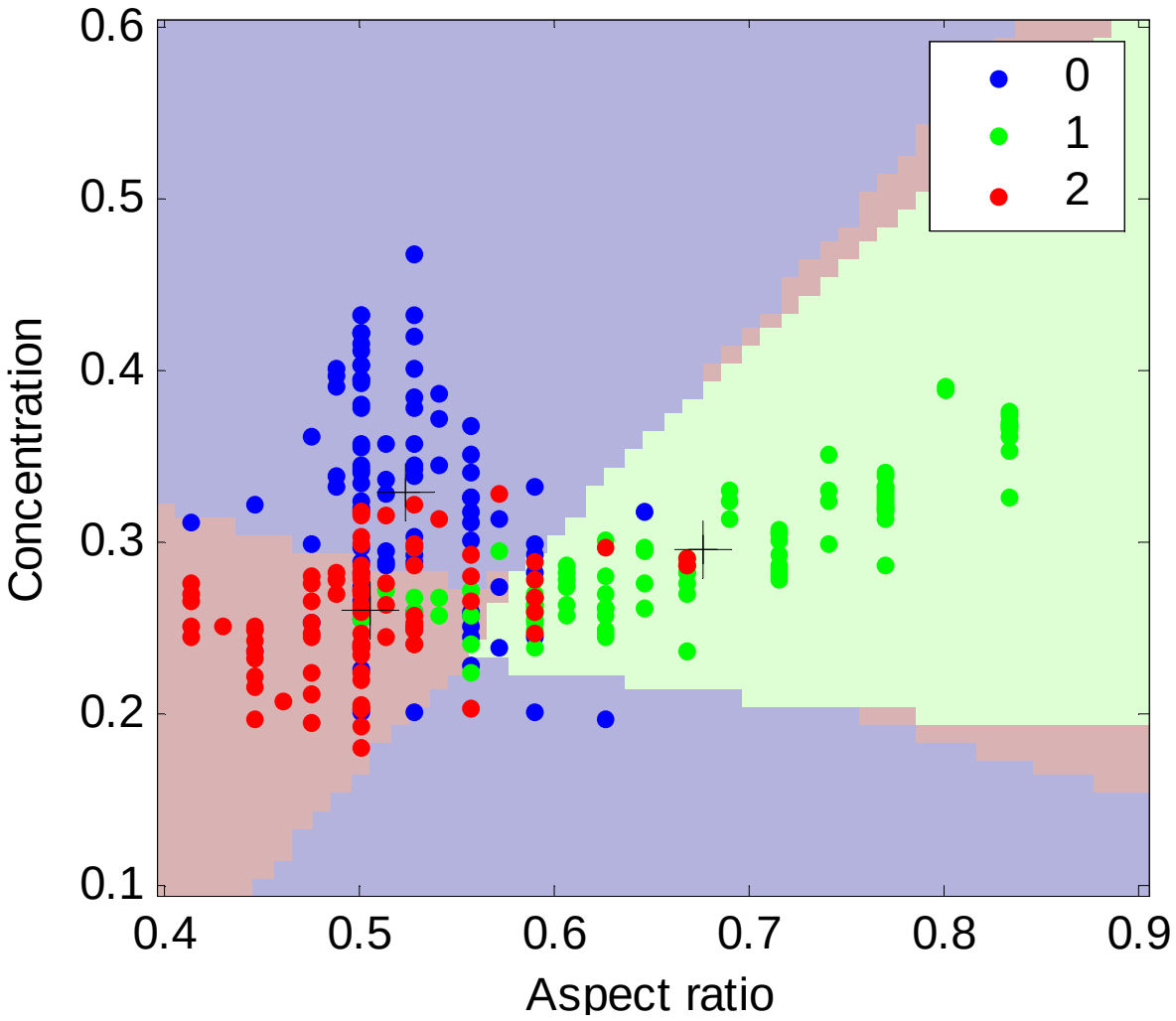
- Le tre classi sono distribuite in modo eterogeneo nello spazio delle features:

$$\Sigma_0 = \begin{bmatrix} 0.0015 & -0.0007 \\ -0.0007 & 0.0039 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 0.0088 & 0.0031 \\ 0.0031 & 0.0015 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} 0.0027 & 0.0006 \\ 0.0006 & 0.0011 \end{bmatrix}$$

Ipotesi 3
Caso generale



Feature dipendenti
Covarianze diverse
Prob. priori uguali

Errori:

classe 0: 25%

classe 1: 15%

classe 2: 28%

err. medio: 22%

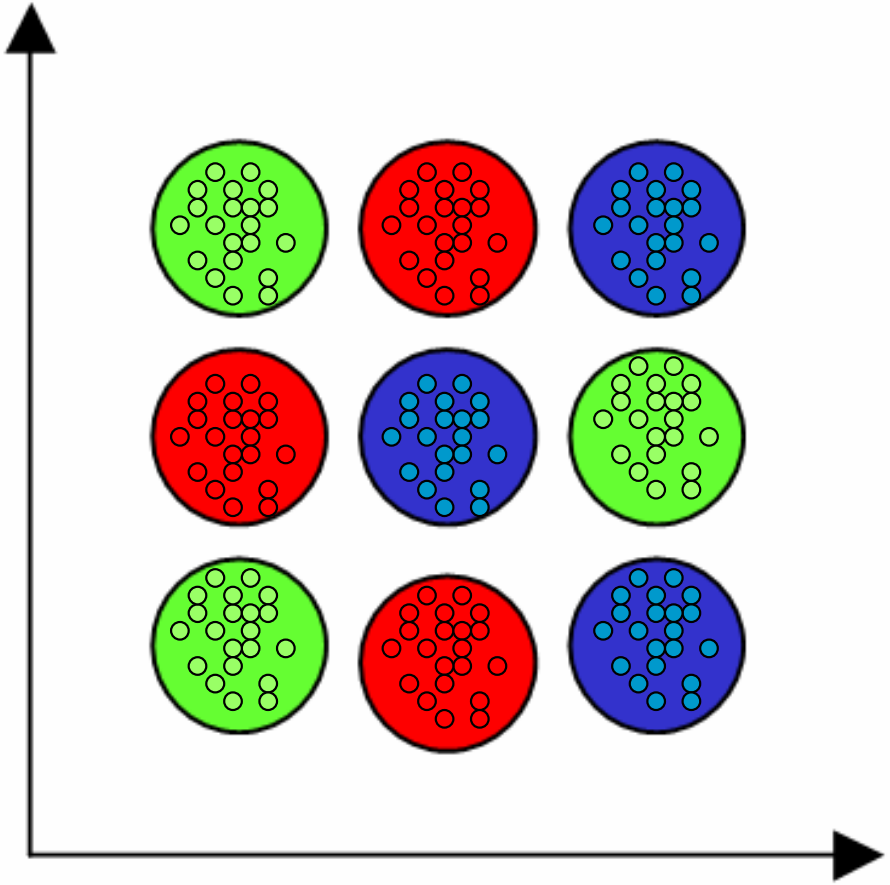
Apprendimento statistico

- Finora si è proceduto cercando di determinare una regola di decisione razionale tramite la stima delle probabilità a posteriori $P(y=i|\mathbf{x})$

- Abbiamo ipotizzato la forma della distribuzione di probabilità e di conseguenza abbiamo stimato i suoi parametri
  - **Modello Parametrico**

$$p(x) = N(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Apprendimento statistico



- Come si comporta un classificatore a minima distanza su questi dati?

Modelli Parametrici

- In generale i modelli parametrici sono utilizzati per descrivere distribuzioni unimodali (hanno una unica “forma”)
  - Di norma però, i processi generano dati che sono descritti da distribuzioni multimodali

+	Ci sono pochi parametri da apprendere
+	Il modello è compatto (uso efficiente della memoria / CPU)
+	Se il modello ipotizzato è adeguato la descrizione dei dati è eccellente
-	Se il modello ipotizzato è sbagliato la descrizione dei dati è pessima
-	Necessaria conoscenza a priori (ipotesi del modello)
-	Il modello reale può essere molto complesso da ipotizzare…

Modelli Parametrici

- Il problema della stima di una distribuzione di densità di probabilità (PDF) con modello parametrico può essere molto "più difficile" del problema della classificazione!!!

- Si possono usare delle tecniche di classificazione che stimano una PDF con **Modelli Non Parametrici**

Modelli Non Parametrici

- Non fanno nessuna assunzione esplicita sulla forma della funzione di densità di probabilità
  - L'unica assunzione (non rigida) è che la PDF abbia un andamento "smooth"

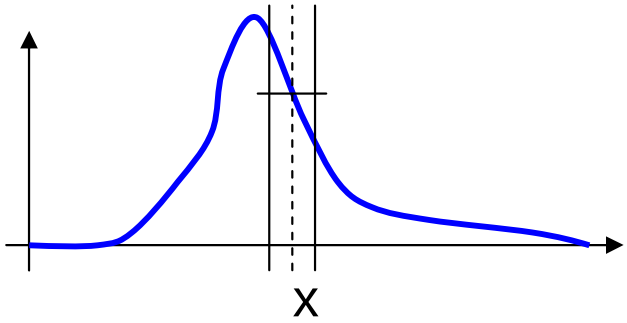
+	Pochi o nessun parametro da apprendere
+	Possono modellare qualunque funzione di densità di probabilità
+	Praticamente nessuna conoscenza a priori necessaria
-	Sono computazionalmente pesanti
-	Non è possibile includere conoscenza

Modelli Non Parametrici: Teoria

- Sia p(x) la PDF da stimare.
- La probabilità *P* che un campione *x* estratto da *p(x)* cada in una regione *R* dello spazio dei campioni è data da:

$$P = \int_R p(u)\, du$$

- Se *R* è sufficientemente piccolo e tale che *p(x)* non cambia significativamente all'interno di *R* si ha anche che:



$$P \cong p(\mathbf{x})\ V$$

V è il volume della regione *R*

Modelli Non Parametrici: Teoria

- Dati *N* campioni la probabilità che la regione R contenga *k* di questi campioni è data dalla distribuzione binomiale:

$$P(k) = \binom{k}{N} P^k (1-P)^{N-k}$$

- Dalle proprietà della distribuzione binomiale, se N$\rightarrow\infty$, si può dimostrare che una stima di P è data da:

$$P \cong \frac{k}{N}$$

Modelli Non Parametrici: Teoria

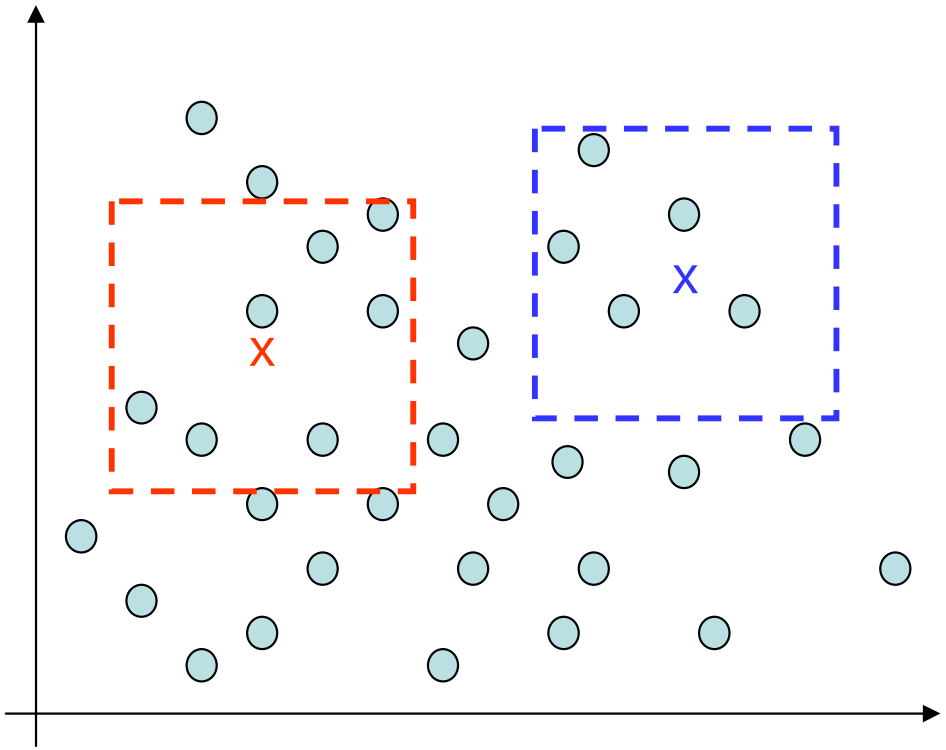
- Mettendo insieme le due formule si ottiene la stima di *p(x)*:

$$p(\mathbf{x}) \cong \frac{k}{NV} = \hat{p}(\mathbf{x})$$

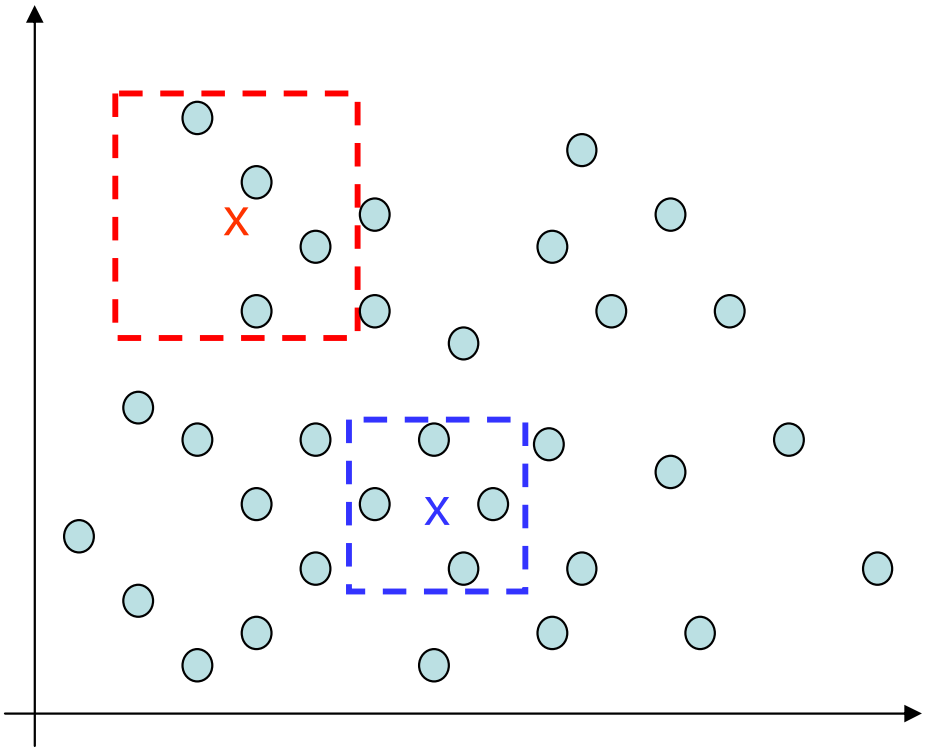
  - *N* è il numero dei campioni disponibili
  - *k* è il numero di campioni che cadono dentro *R*
  - *V* è il volume della regione *R*
  - NOTA: *R* contiene x

- Possiamo usare questa formula per stimare la PDF p(x) sull'intero spazio a partire dai campioni disponibili
  - k e V sono interdipendenti. Due approcci equivalenti possibili.

Modelli Non Parametrici: Teoria



Approccio 1:
Fissare R (V) e ricavare k



Approccio 2:
Fissare k e ricavare R (V)

k Nearest Neighbor Classifier

- Usiamo la stima della PDF per determinare le varie probabilità per il problema di classificazione

$$\hat{p}(\mathbf{x}) = k/NV$$

- Le verosimiglianze (distribuzioni dei pattern di classe i) come:

$$\hat{p}(\mathbf{x} \mid y = i) = \frac{k_i}{N_i V}$$

- Le probabilità a priori sono stimate come:

$$\hat{p}(y = i) = \frac{N_i}{N}$$

k Nearest Neighbor Classifier

- Le probabilità a posteriori diventano:

$$\hat{p}(y=i\,|\,\mathbf{x}) = \frac{\hat{p}(\mathbf{x}\,|\,y=i)\,\hat{p}(y=i)}{\hat{p}(\mathbf{x})} = \left(\frac{k_i}{N_i V}\frac{N_i}{N}\right) \Big/ \frac{k}{NV} = \frac{k_i}{k}$$

- Le funzioni discriminanti risultanti sono quindi:

$$g_i(\mathbf{x}) = \frac{k_i}{k}$$

- Tradotto:
  - Un elemento **x** appartiene alla classe i-esima sse

    la maggioranza dei *k* elementi racchiusi dalla regione R (che contiene anche **x**) sono di classe i

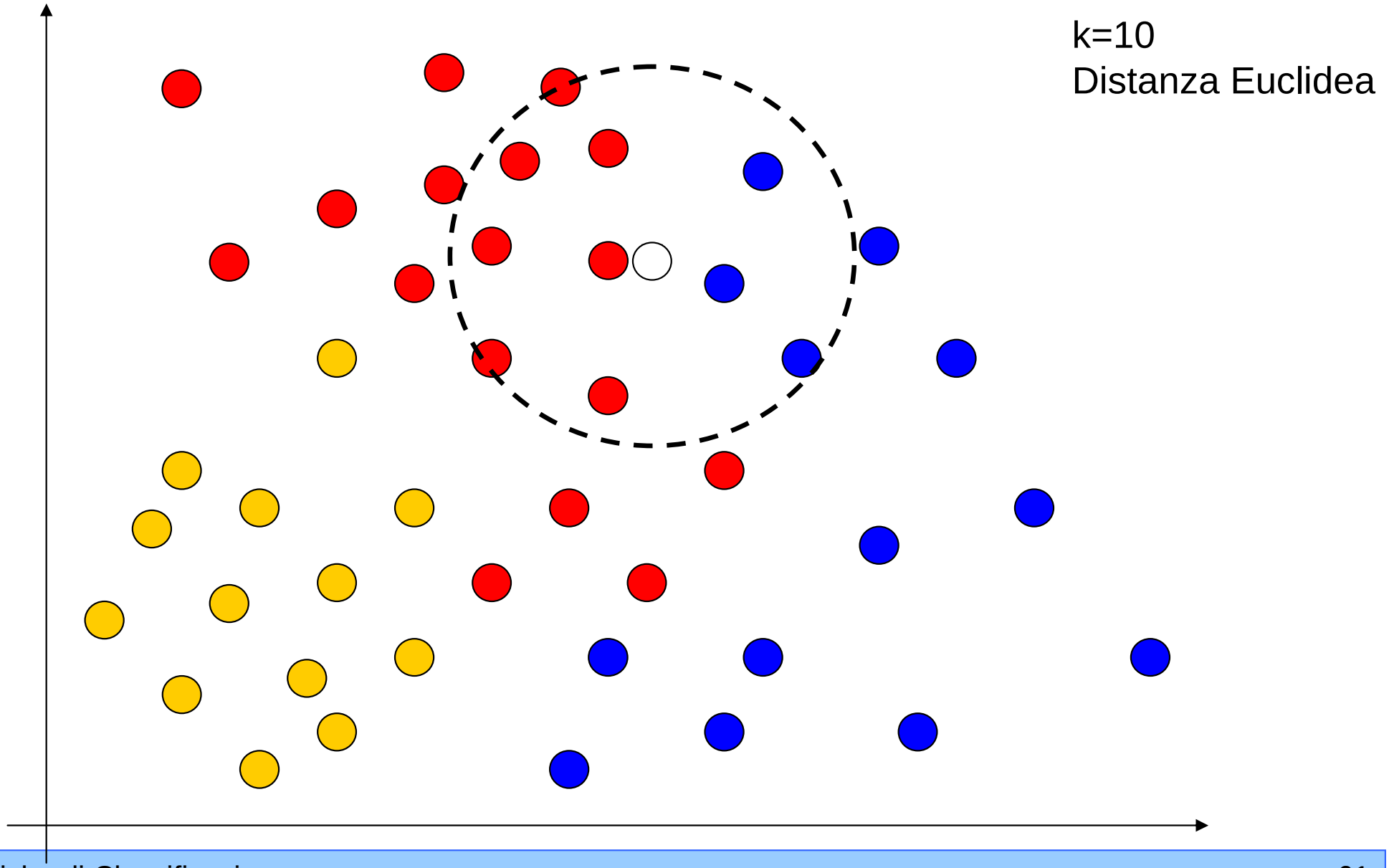
k Nearest Neighbor Classifier

- In pratica l'algoritmo kNN è:
  - Dato un campione **x**
  - Cercare nel training set i suoi k vicini (*)
  - Contare sui k vicini il numero di elementi di ogni classe
  - Assegnare ad **x** la classe più occorrente

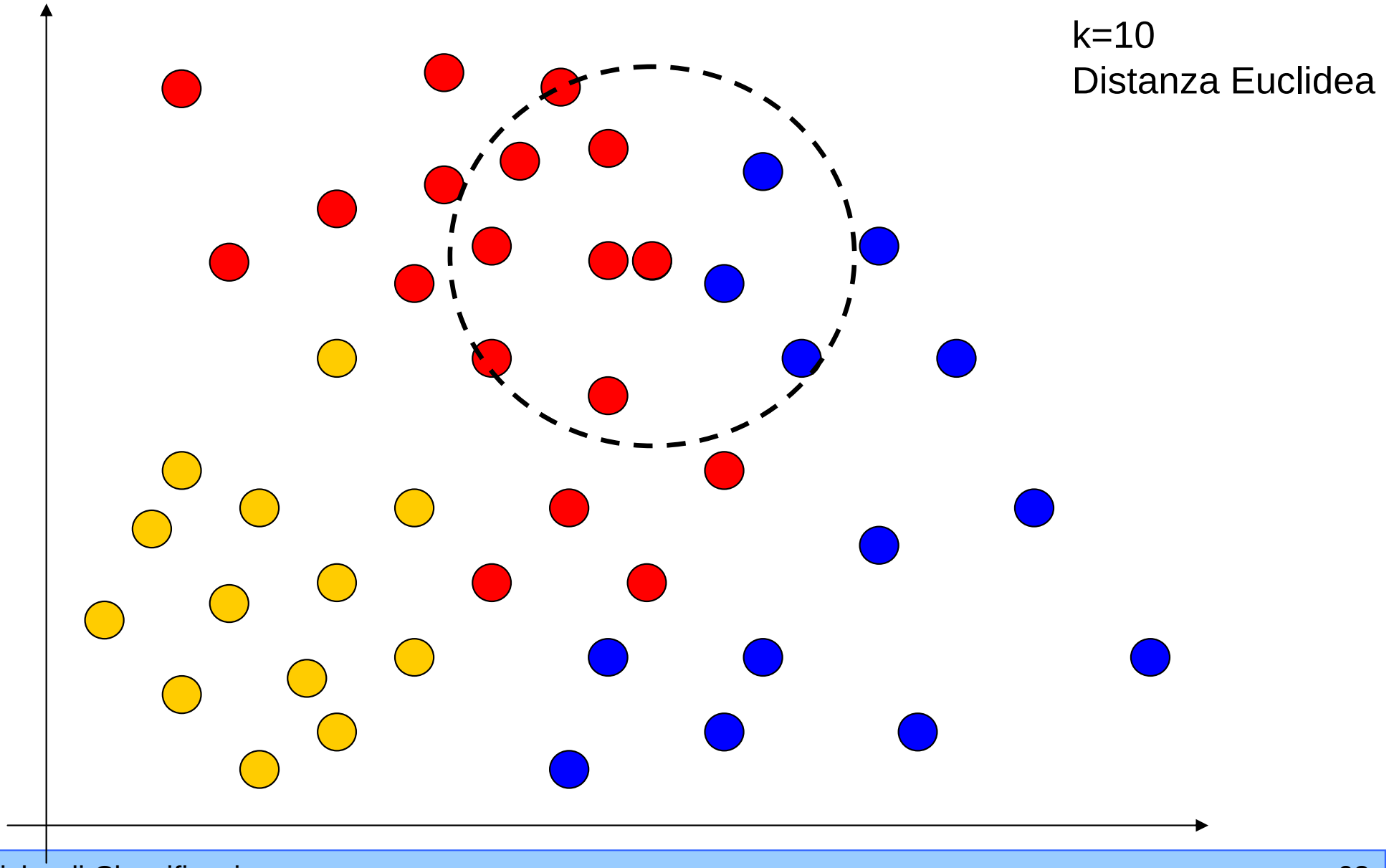
(*) si usa l'approccio 2 per la stima di p(x) per assicurarsi di avere k campioni da usare per la classificazione

- Il classificatore kNN richiede la conoscenza solamente di:
  - Il valore k
  - Un insieme di campioni con associata la classe (training set)
  - Una misura di vicinanza

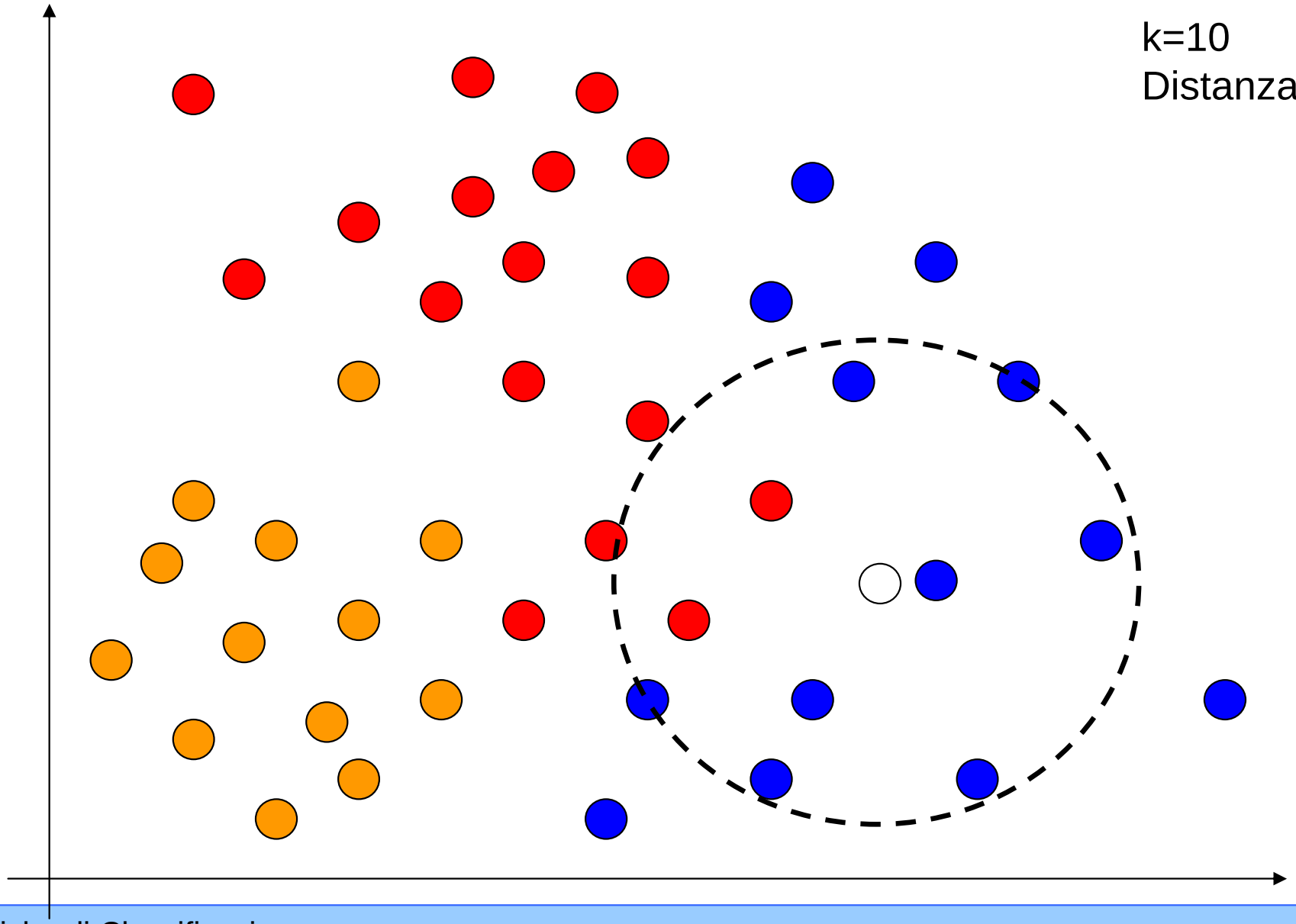
k Nearest Neighbor Classifier



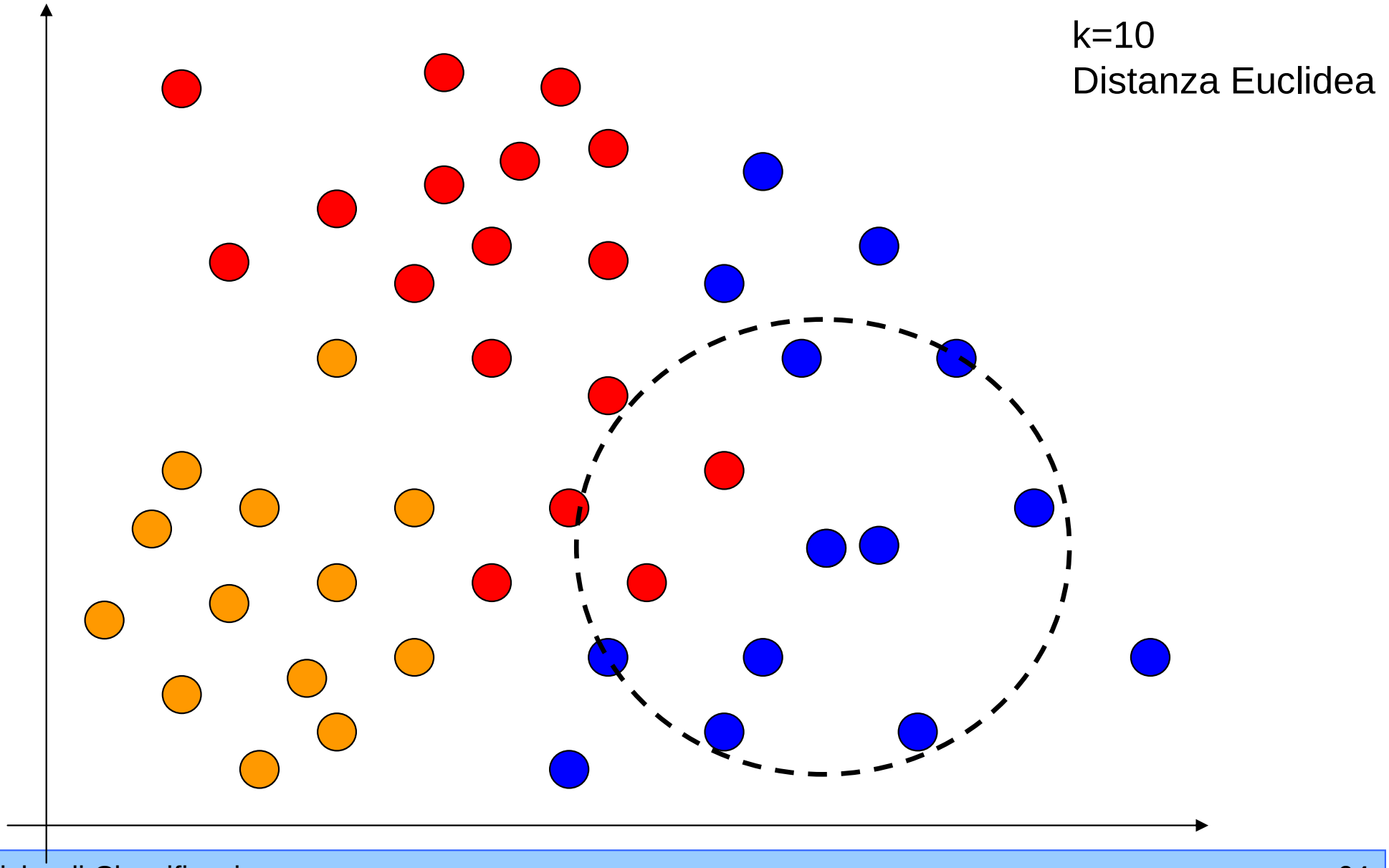
k Nearest Neighbor Classifier



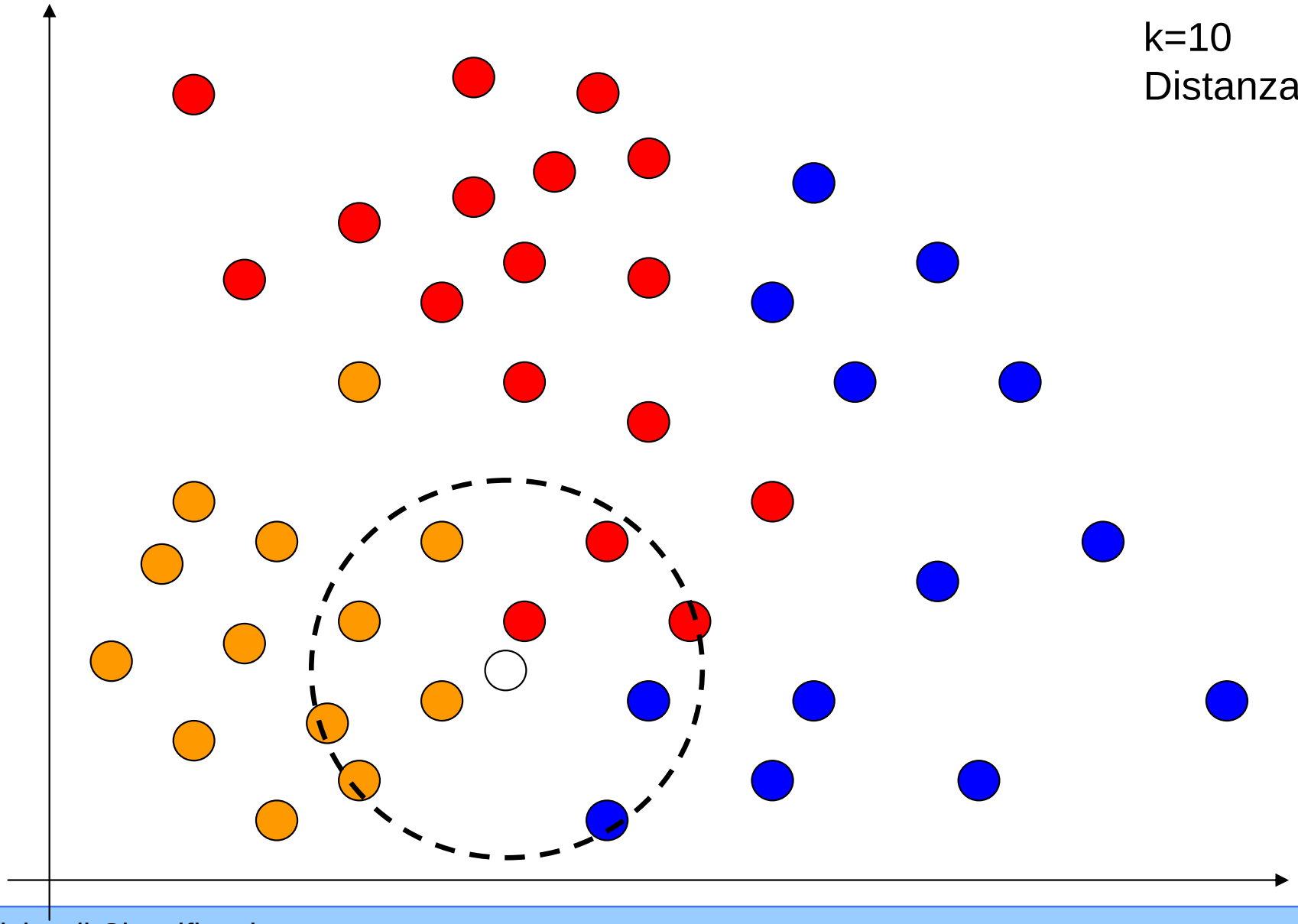
k Nearest Neighbor Classifier



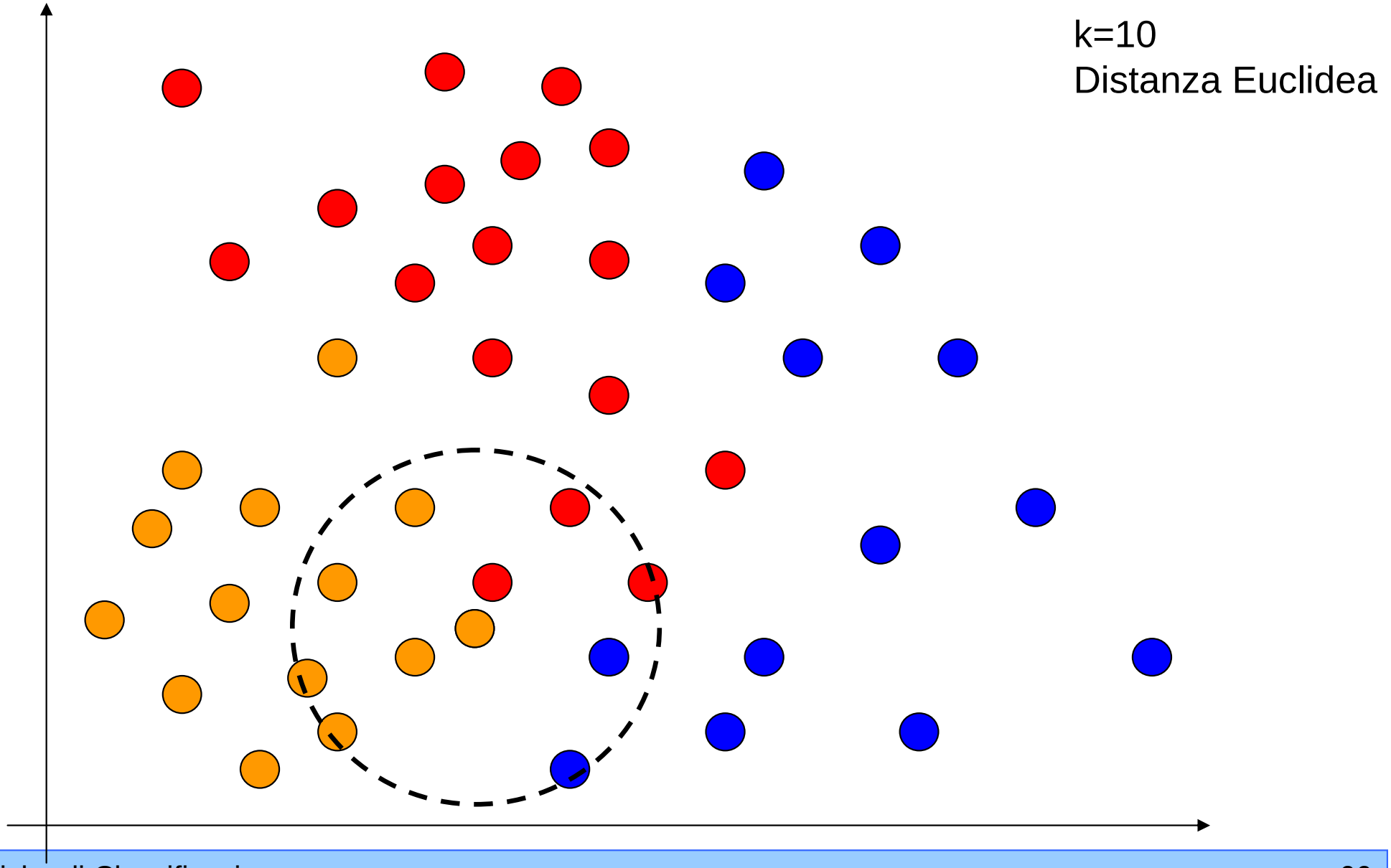
k Nearest Neighbor Classifier



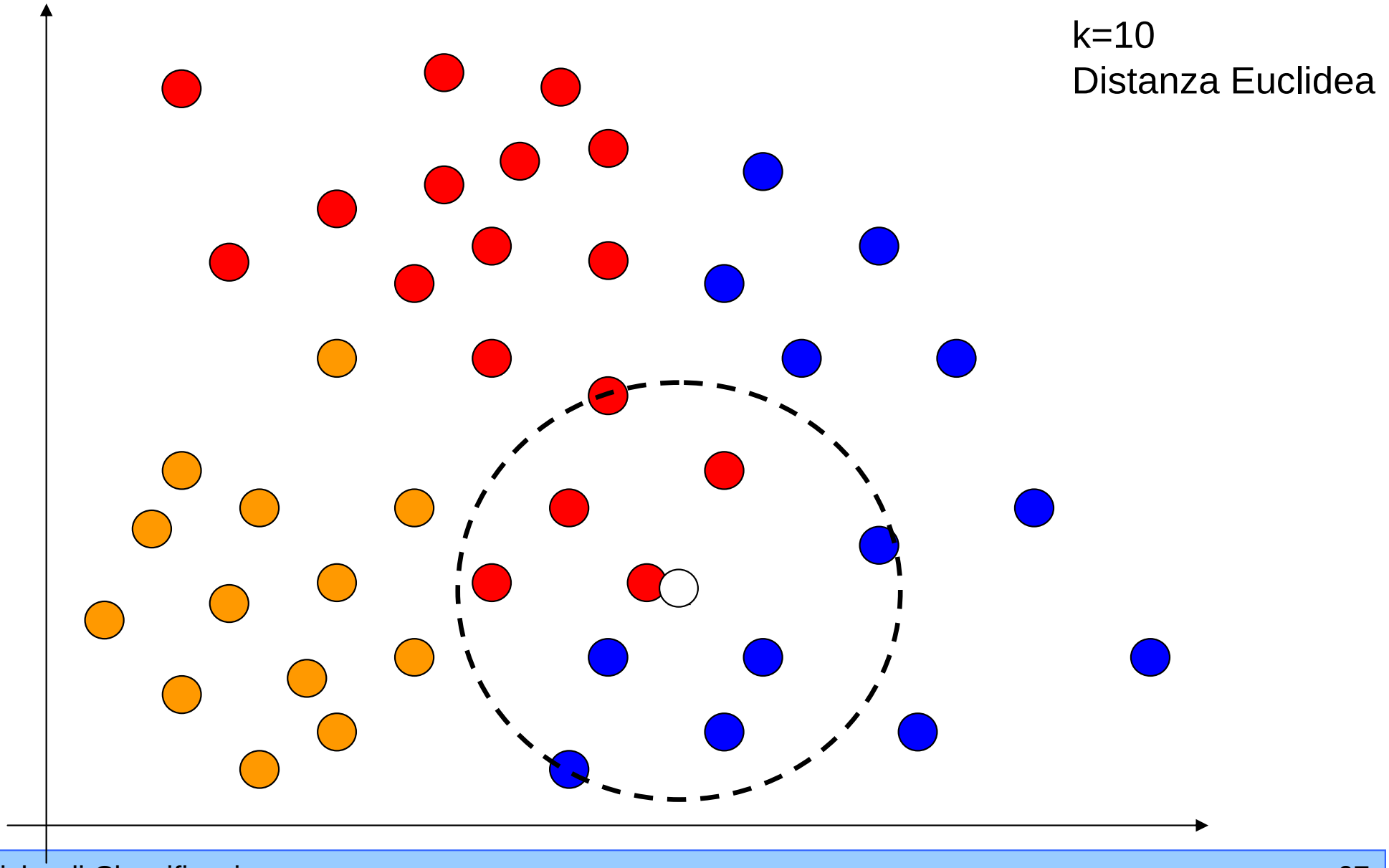
k Nearest Neighbor Classifier



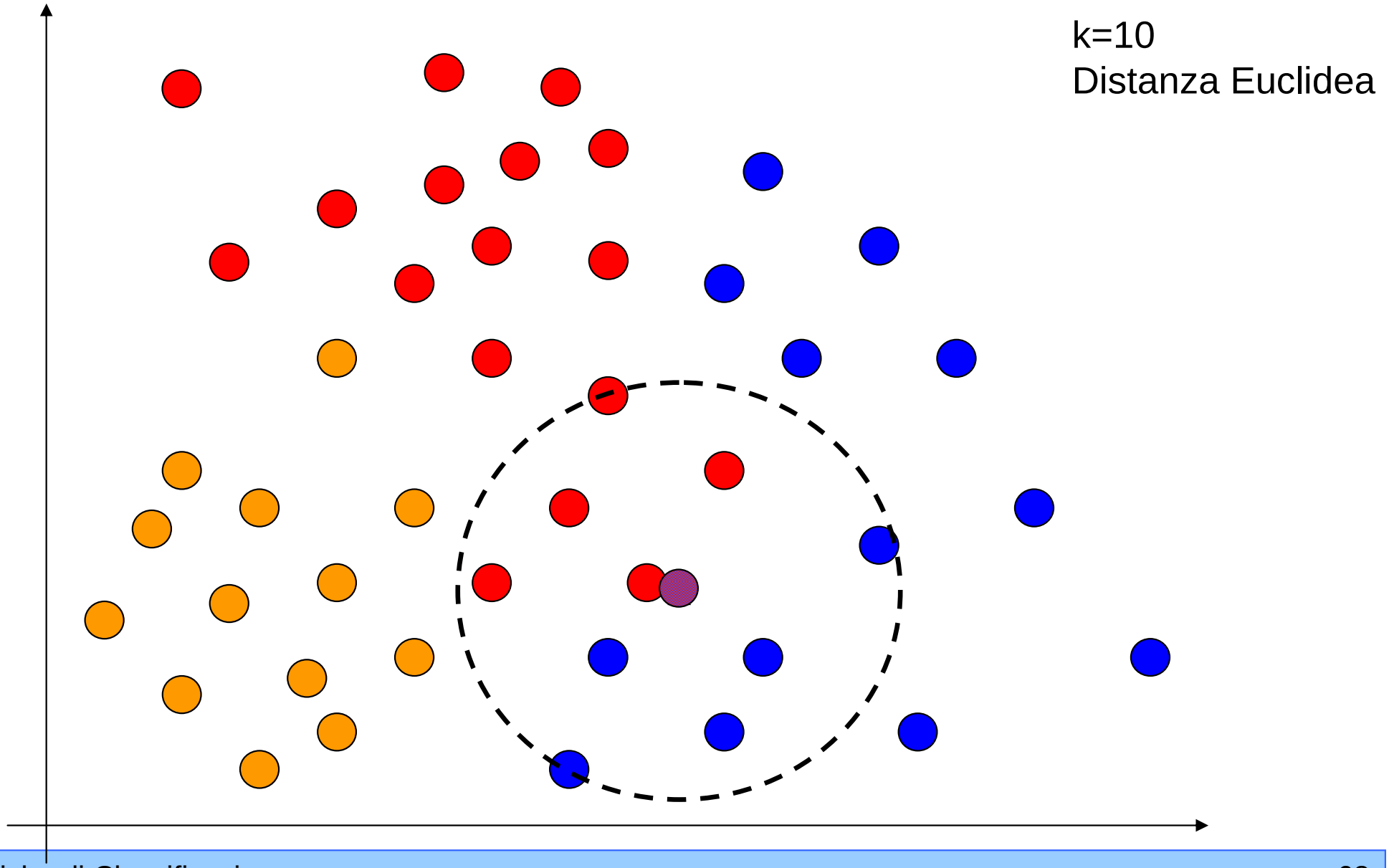
k Nearest Neighbor Classifier



k Nearest Neighbor Classifier

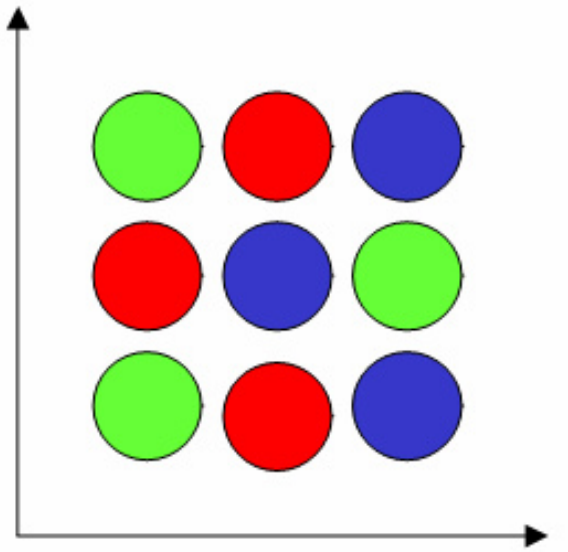


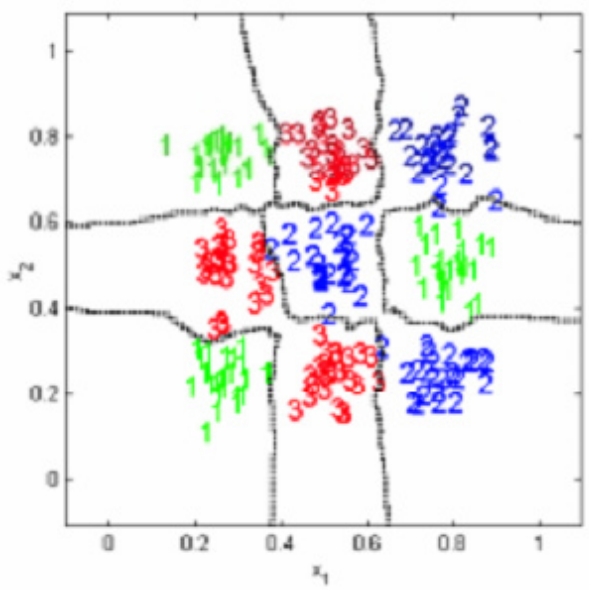
k Nearest Neighbor Classifier

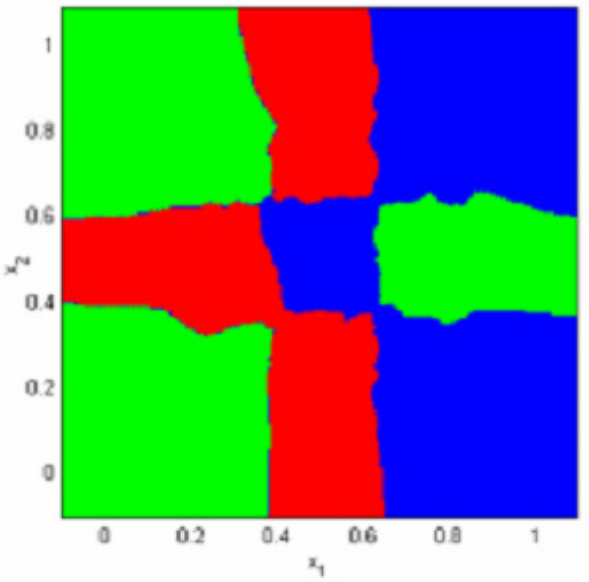


kNN Classifier - Esempio

k=5
Distanza Euclidea

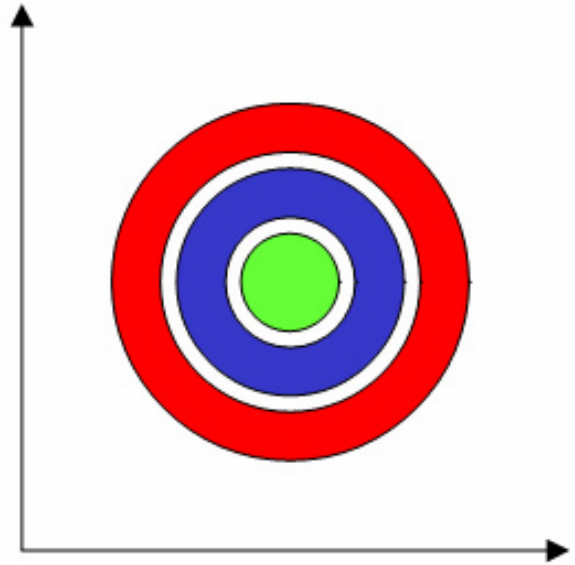


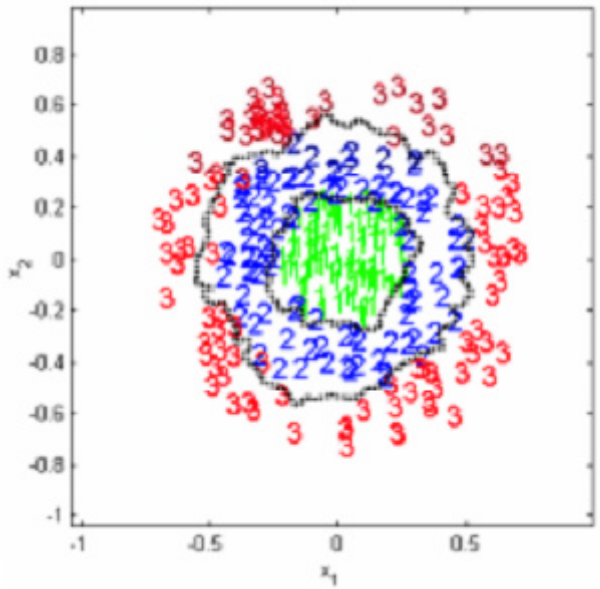


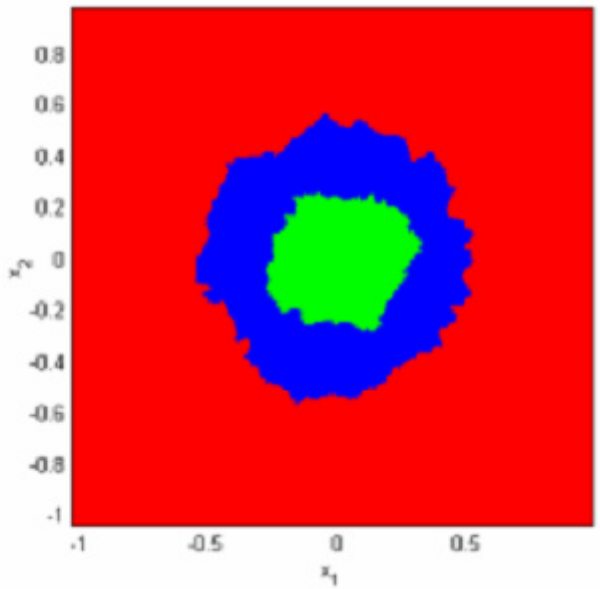


kNN Classifier - Esempio

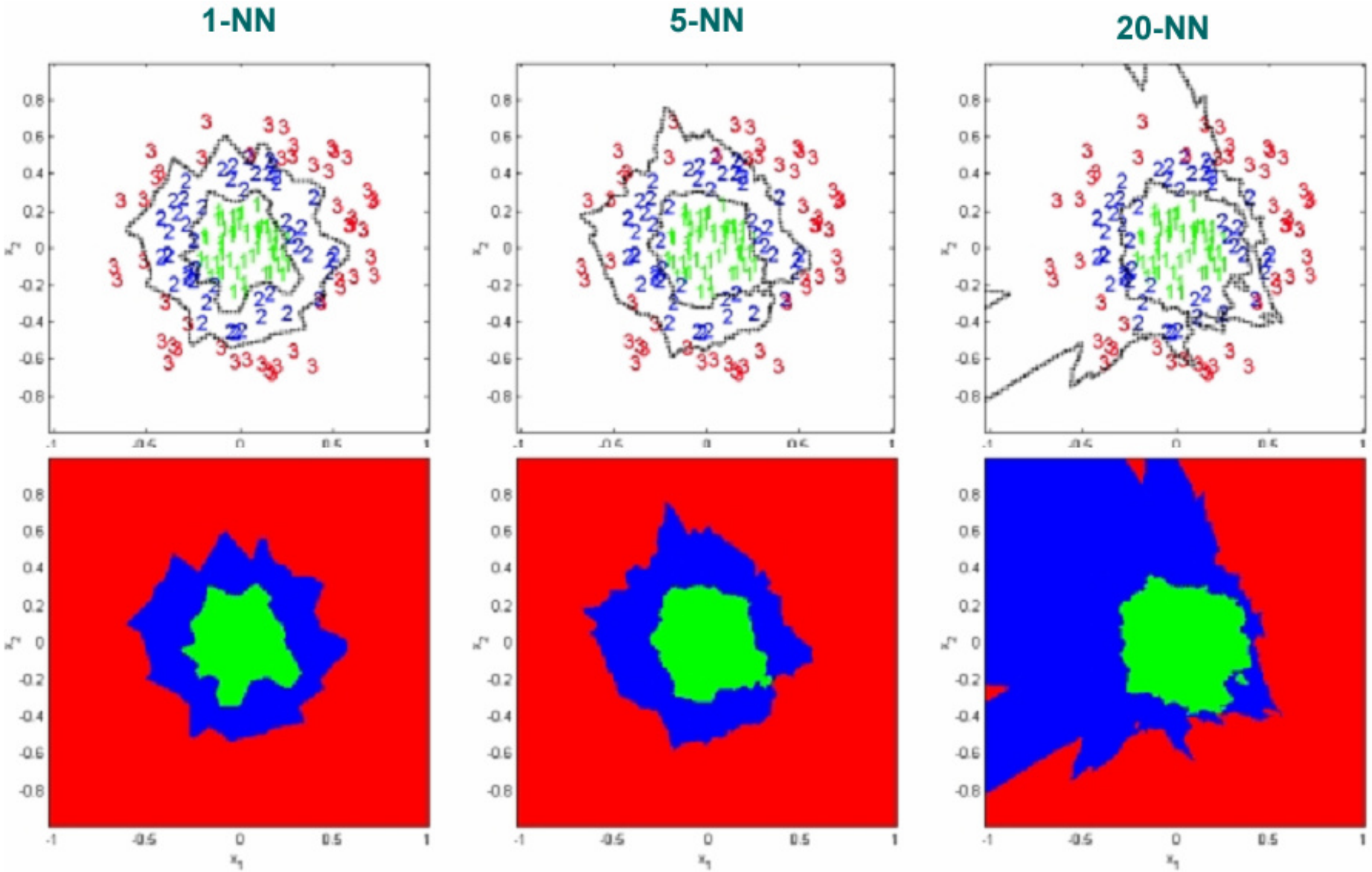
k=5
Distanza Euclidea







kNN Classifier – La scelta di k



kNN Classifier

- Vantaggi
  - Funzionamento intuitivo
  - Utilizza informazioni locali (forte adattività)
  - Classificatore quasi ottimale per valori elevati di N
- Svantaggi
  - Computazionalmente pesante
  - Necessarie strutture dati particolari per la ricerca efficiente dei vicini
- La scelta di k
  - k grandi danno
    - superfici di decisione smussate e stime più accurate
  - k troppo grandi
    - rompono la località delle informazioni e rendono la classificazione computazionalmente ancora più onerosa

Modelli Non Parametrici: Teoria

- Perchè $P \cong \frac{k}{N}$?

- La probabilità che su N campioni estratti da une distribuzione p(x), k cadano in una regione R è data dalla distribuzione binomiale:

$$P(k) = \binom{k}{N} P^k (1-P)^{N-k}$$

- Se consideriamo la media e la varianza della distribuzione binomiale sulla variabile aleatoria k/N si ha:

$$E\left[\frac{k}{N}\right] = P \qquad Var\left[\frac{k}{N}\right] = \frac{P(1-P)}{N}$$

- Per N$\rightarrow\infty$, la varianza della distribuzione tende a zero e quindi il valore atteso può essere usato come stima di P

Tecniche di Classificazione

Analisi delle componenti principali

Elaborazione delle Immagini - Complementi

Gianluigi Ciocca

Analisi delle componenti principali

- E’ un metodo molto utilizzato per ridurre il numero delle features usate in un problema di classificazione o riconoscimento

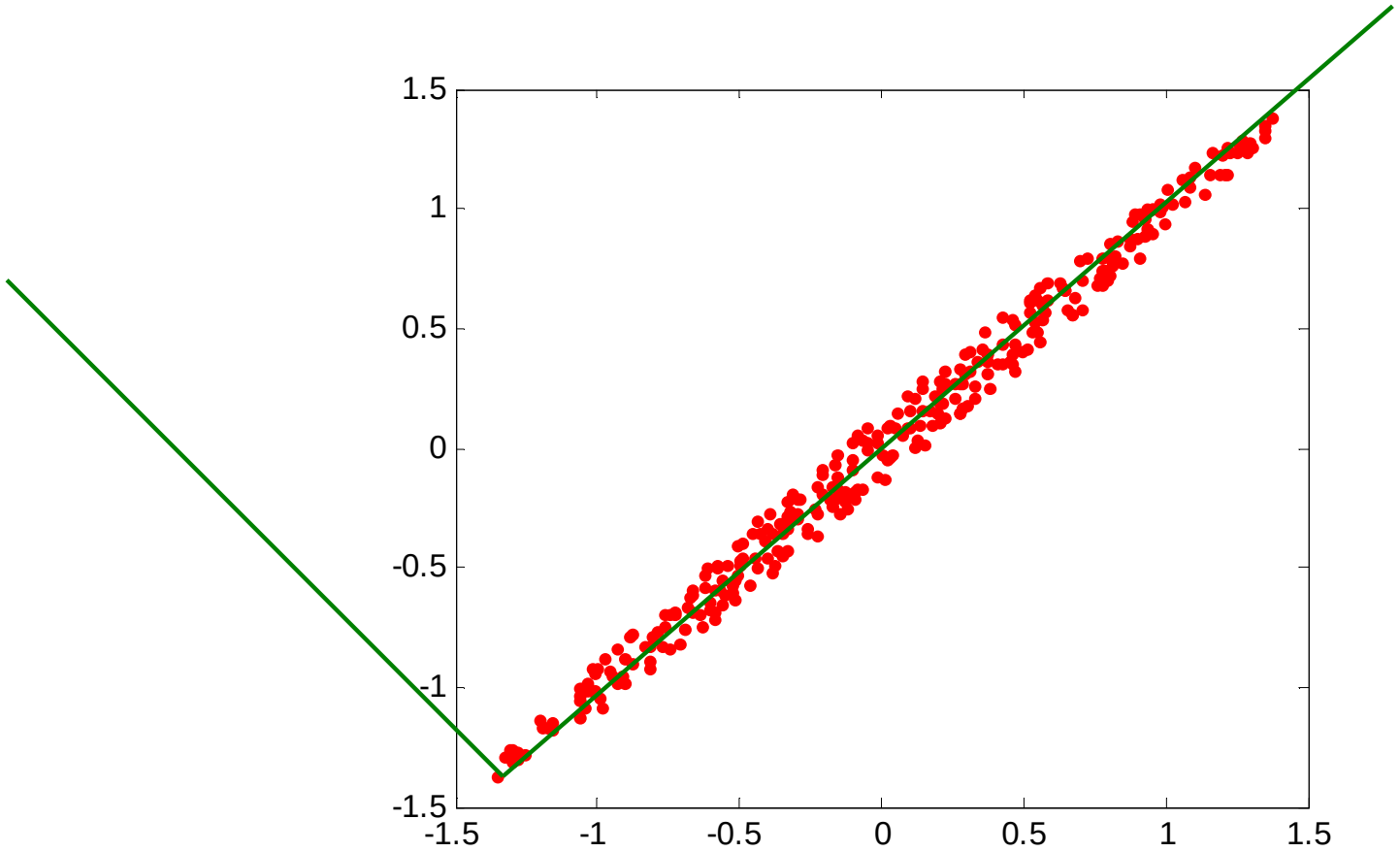
- Dato un insieme di *N* pattern in $\Re^d$:

$$T = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$$

- Si cerca una rappresentazione degli stessi dati che conservi il più possibile la variabilità dei dati in ingresso in uno spazio di dimensione $k < d$.

Analisi delle componenti principali

- Una feature è (quasi) ridondante



Analisi delle componenti principali

- Determina le direzioni del nuovo spazio lungo le quali la varianza dei dati è massima (nel training set).

- La varianza della proiezione dei vettori **x** lungo una direzione **v** (**v**^{T}**v**=1) è:

$$Var[\mathbf{X}\mathbf{v}] = \mathbf{v}^T\mathbf{C}\mathbf{v}$$

- Dove **X** è la matrice N $\times$ d che ha per *i*-esima riga l'*i*-esimo pattern e **m** è il pattern medio, **C** è la matrice di covarianza:

$$\mathbf{C}_{ij} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{X}_{ki} - \mathbf{m}_i)(\mathbf{X}_{kj} - \mathbf{m}_j)$$

Analisi delle componenti principali

- Per massimizzare tale varianza bisogna risolvere il problema quadratico:

$$\max \mathbf{v}^T \mathbf{C}\mathbf{v} \qquad \text{con il vincolo} \qquad \mathbf{v}^T\mathbf{v} = 1$$

- Per trovare i massimi (e i minimi) occorre trovare il punto di sella della funzione Lagrangiana:

$$L(\mathbf{v}, \lambda) = \mathbf{v}^T \mathbf{C}\mathbf{v} - \lambda(\mathbf{v}^T\mathbf{v} - 1)$$

$$\frac{\partial L(\mathbf{v}, \lambda)}{\partial \mathbf{v}} = 2(\mathbf{C}\mathbf{v} - \lambda\mathbf{v}) = 0 \Rightarrow \mathbf{C}\mathbf{v} = \lambda\mathbf{v}$$

Analisi delle componenti principali

- La soluzione

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$$

- Corrisponde a trovare gli autovalori (λ) e autovettori (**v**) della matrice di covarianza **C**

- Essendo la matrice di covarianza simmetrica, e definita positiva, gli autovalori sono reali e gli autovettori sono ortonormali

Analisi delle componenti principali

- Gli autovettori della matrice di covarianza definiscono gli assi (una base) di un nuovo spazio di feature.

- L'entità della varianza delle feature sugli assi è data dai rispettivi autovalori:

$$\mathbf{v}^T \mathbf{C}\mathbf{v} = \mathbf{v}^T (\lambda \mathbf{v}) = \lambda (\mathbf{v}^T \mathbf{v}) = \lambda$$

- L'insieme degli autovettori forma la matrice di trasformazione **D=[v_1 v_2 v_3 …]**

Algoritmo per il calcolo della PCA

- La proiezione dei vettori di ottiene semplicemente:

$$\mathbf{y} = \mathbf{D}^T(\mathbf{x} - \mathbf{m})$$

- E la trasformazione inversa è:

$$\mathbf{x} = \mathbf{D}\mathbf{y} + \mathbf{m}$$

- Un elemento nello spazio originario può essere ricostruito con una combinazione lineare degli autovettori

- L'idea è quella di usare solo i k autovettori più significativi (componenti principali) per la trasformazione **D**

Numero di componenti principali

- Si tengono i k autovettori corrispondenti ai k autovalori più grandi

- Il valore di *k* è tale che la retained variance *r*(*k*) sia maggiore o uguale ad una certa soglia (es. 90%):

$$r(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq S$$

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$$

Proprietà della PCA

- Alcune proprietà delle componenti principali:
  - La media delle componenti $\mathbf{y}_i$ è nulla.
  - La varianza delle componenti $\mathbf{y}_i$ è pari agli autovalori λ_i.
  - La covarianza di due componenti $\mathbf{y}_i$, $\mathbf{y}_j$ è nulla (decorrelazione)

Algoritmo per il calcolo della PCA

- Algoritmo
  - Dati N vettori $\{\mathbf{x}_i\}$ n-dimensionali, metterli per riga in una matrice $\mathbf{A}=[x_{ij}]$
  - Calcolare il vettore delle medie colonna per colonna $\mathbf{m}=[m_j]^T$
  - Calcolare la matrice $\mathbf{B}=[x_{ij}-m_j]$
  - Calcolare la matrice di covarianza $\mathbf{C}=\mathbf{B}\mathbf{B}^T$
  - Calcolare gli autovettori e autovalori di $\mathbf{C}$
  - Ordinare gli autovettori in ordine decrescente rispetto l'autovalore corrispondente
  - Mettere i primi k autovettori (v_k), in una matrice $\mathbf{D}$
  - Proiettare i vettori in $\mathbf{B}$ nel nuovo spazio con $\mathbf{y}_i=\mathbf{D}^T[\mathbf{x}_i-\mathbf{m}]$
  - $\{\mathbf{y}_i\}$ sono i nuovi vettori nello spazio a componenti ridotte

Algoritmo per il calcolo della PCA

- PCA – Principal Component Analysis
  - Esempio

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \; x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; x_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} \qquad m = \begin{bmatrix} \frac{2}{3} & 1 \end{bmatrix}^T \qquad B = \begin{bmatrix} 1/3 & 1 \\ 1/3 & 0 \\ -2/3 & -1 \end{bmatrix}$$

$$C = BB^T = \begin{bmatrix} 2/3 & 1 \\ 1 & 2 \end{bmatrix}$$

Algoritmo per il calcolo della PCA

- PCA – Principal Component Analysis
  - Esempio

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \; x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \; x_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} \qquad m = \begin{bmatrix} \frac{2}{3} & 1 \end{bmatrix}^T$$

$$B = \begin{bmatrix} 1/3 & 1 \\ 1/3 & 0 \\ -2/3 & -1 \end{bmatrix} \qquad C = BB^T = \begin{bmatrix} 2/3 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\lambda_1 = 2.53, \quad v_1 = \begin{bmatrix} 0.47 \\ 0.88 \end{bmatrix}$$

$$\lambda_2 = 0.13, \quad v_2 = \begin{bmatrix} -0.88 \\ 0.47 \end{bmatrix}$$

Algoritmo per il calcolo della PCA

- PCA – Principal Component Analysis
  - Esempio

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix},\ x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ x_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 0 & 0 \end{bmatrix} \qquad m = \begin{bmatrix} \frac{2}{3} & 1 \end{bmatrix}^T$$

$$B = \begin{bmatrix} 1/3 & 1 \\ 1/3 & 0 \\ -2/3 & -1 \end{bmatrix} \qquad C = BB^T = \begin{bmatrix} 2/3 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\lambda_1 = 2.53, \quad v_1 = \begin{bmatrix} 0.47 \\ 0.88 \end{bmatrix}$$

$$\lambda_2 = 0.13, \quad v_2 = \begin{bmatrix} -0.88 \\ 0.47 \end{bmatrix}$$

$$D = \begin{bmatrix} 0.47 \\ 0.88 \end{bmatrix}$$

$$y_1 = D^T[x_1 - m] = 1.0367$$

$$y_2 = D^T[x_2 - m] = 0.1567$$

$$y_3 = D^T[x_3 - m] = -1.1933$$

Algoritmo per il calcolo della PCA

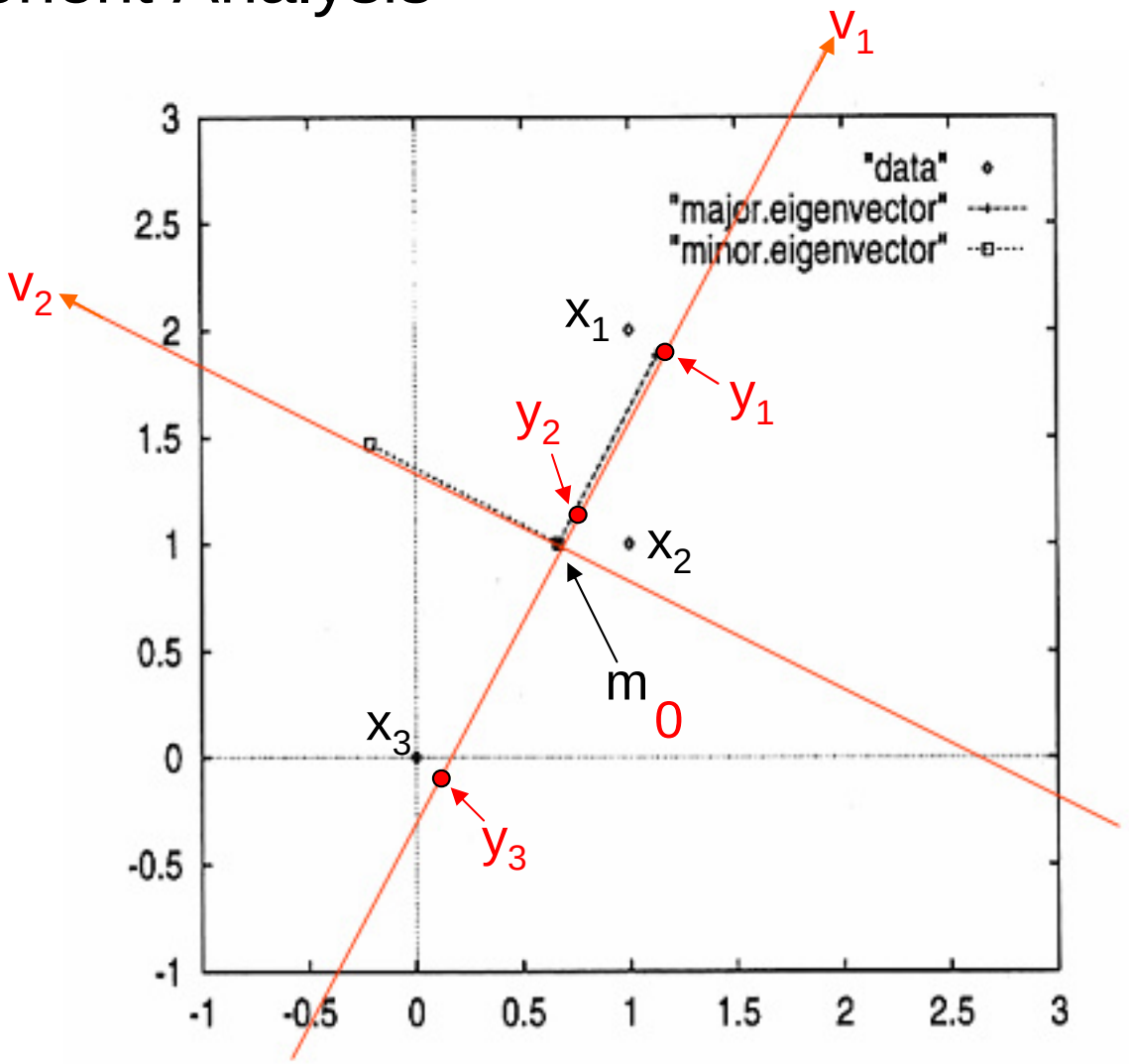
- PCA – Principal Component Analysis
  - Esempio

$$x_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix},\ x_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ x_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$y_1 = 1.0367$$

$$y_2 = 0.1567$$

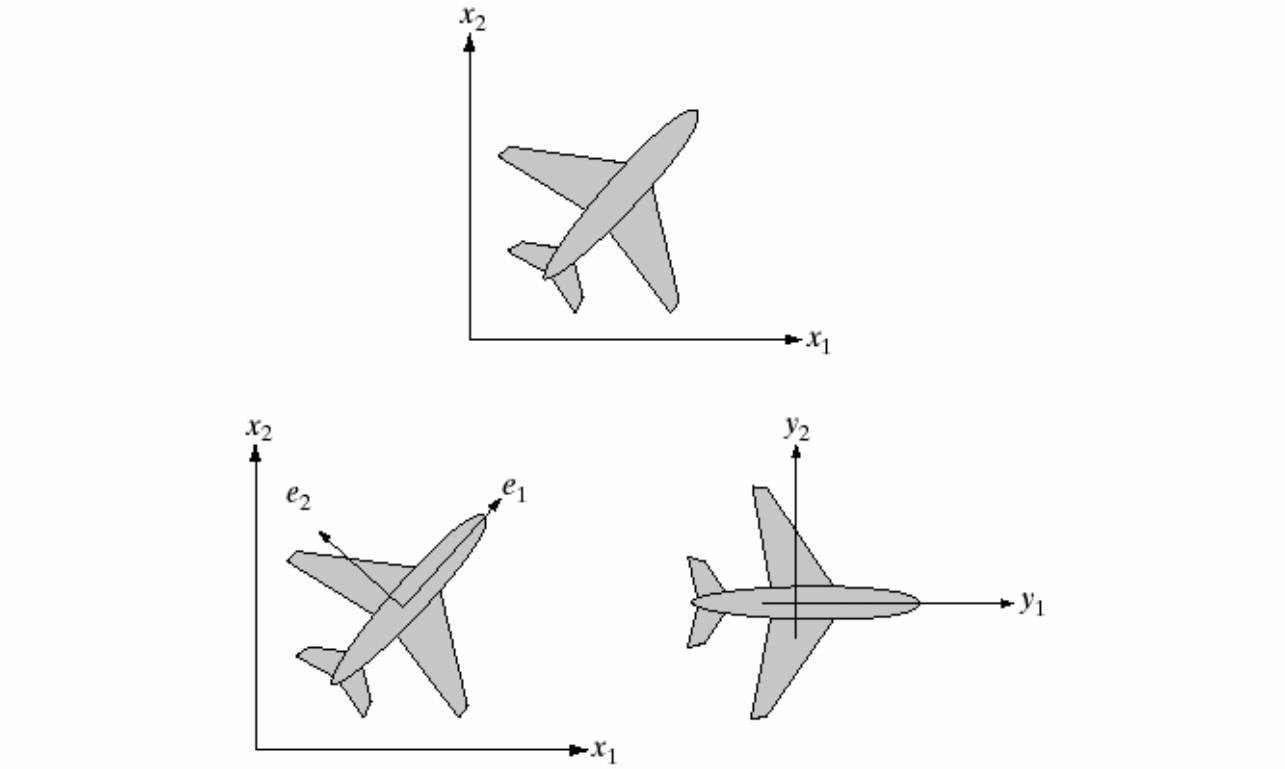
$$y_3 = -1.1933$$



Analisi delle componenti principali

- Possibili utilizzi:
  - Riduzione della dimensionalità dello spazio delle feature.
  - Metrica per il Template matching normalizzata, tramite l'individuazione di un prototipo nello spazio proiettato.
  - Individuazione dell'angolo di rotazione (e della scala) degli oggetti individuati in un'immagine.
  - Analisi della distribuzione del colore in un'immagine.
  - …

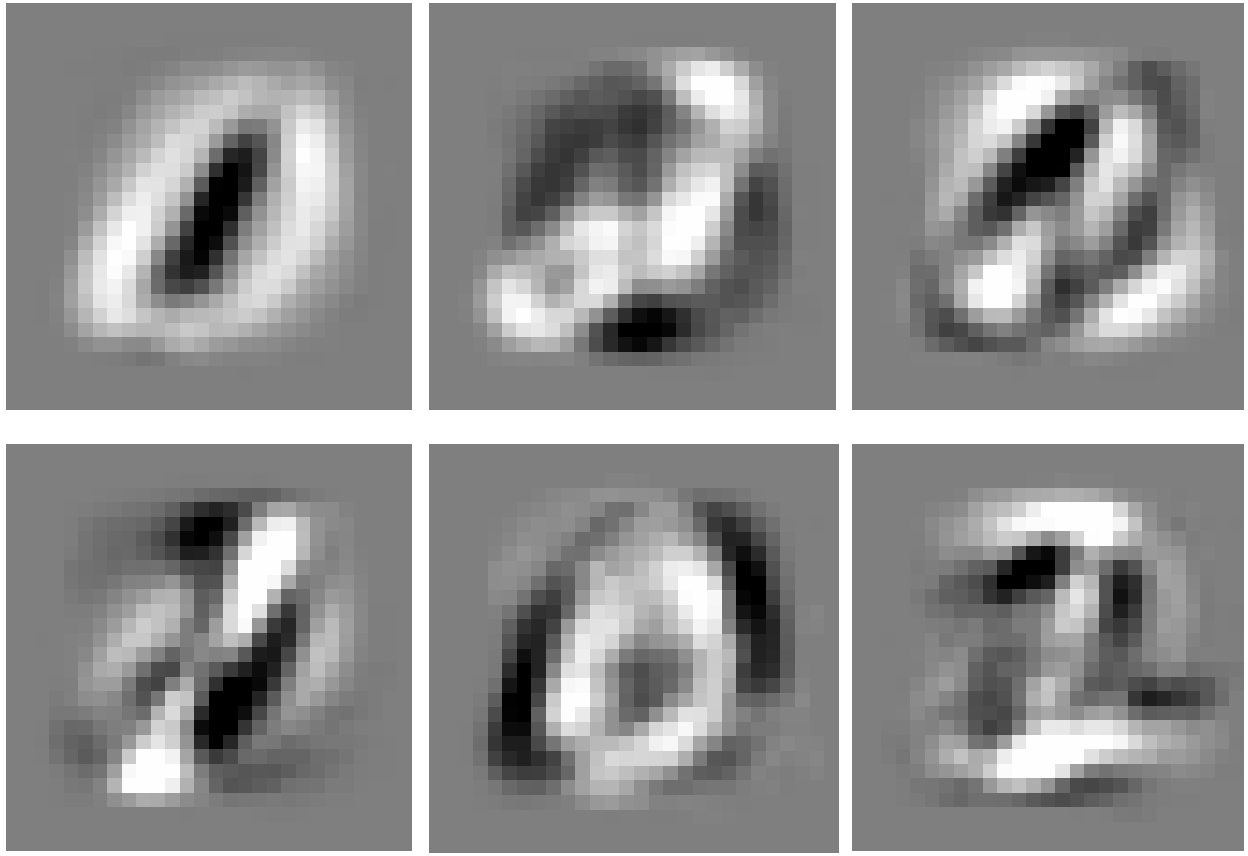
Analisi delle componenti principali



a
b c

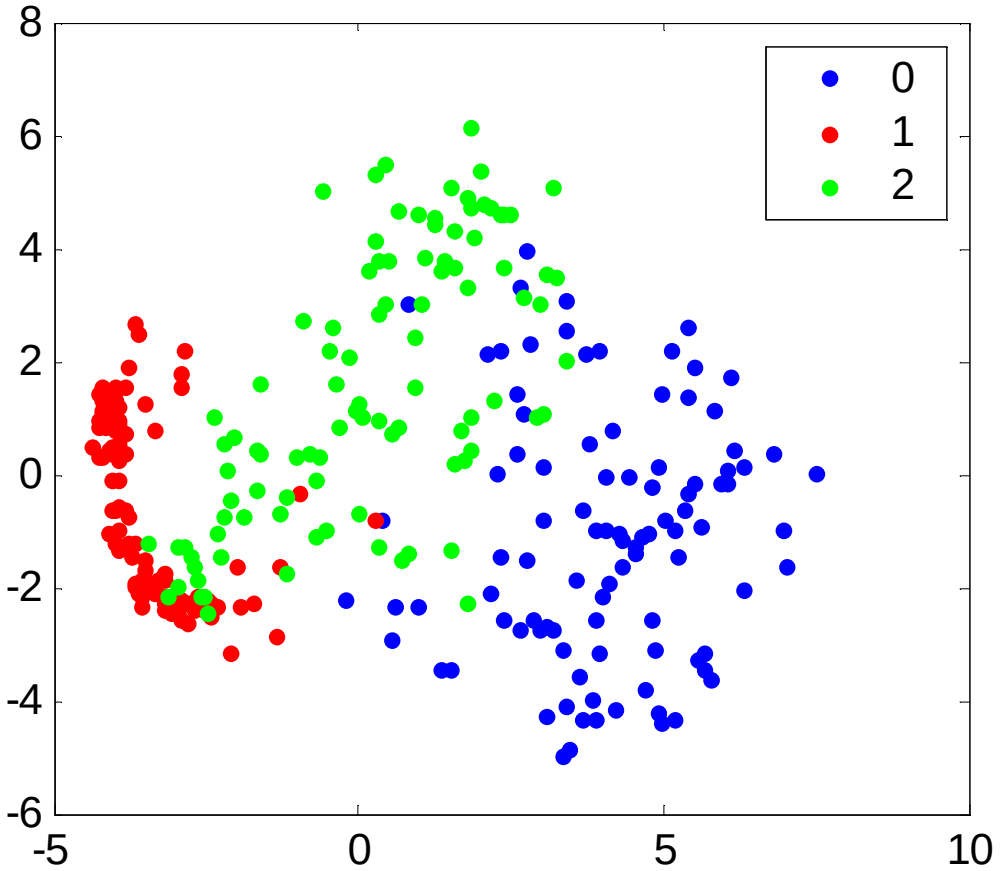
FIGURE 11.29 (a) An object. (b) Eigenvectors. (c) Object rotated by using Eq. (11.4-6). The net effect is to align the object along its eigen axes.

Riconoscimento di cifre



Vettore di feature = insieme di tutti i pixel dell'immagine (N^2 componenti)

Riconoscimento di cifre



- Errore classificatore a minima distanza (6 componenti principali): 9.6%