Tecniche di riconoscimento statistico

Teoria e applicazioni industriali

Parte 3 – Costruzione di un classificatore

Ennio Ottaviani
On AIR srl
ennio.ottaviani@onairweb.com
http://www.onairweb.com/corsoPR

A.A. 2018-2019

Introduzione

- Con l'approccio bayesiano si potrebbe realizzare un classificatore ottimo a patto di conoscere:
  - le probabilità a priori $P(\omega_i)$
  - le probabilità condizionate $P(X| \omega_i)$

- Queste non sono quasi mai disponibili nelle applicazioni reali !

- L'unica alternativa praticabile è raccogliere degli esempi (*training set*) e stimare da questi le probabilità richieste
  - facile e di solito sempre possibile
  - il numero di esempi necessari può essere molto elevato
  - la rappresentatività del campione non è garantita

Approcci possibili

- Per ottenere una stima delle distribuzioni di probabilità delle varie classi ci sono due approcci generali

  – Approccio parametrico: si assume che le distribuzioni abbiano una forma nota, dipendente da un numero finito di parametri (es. misture gaussiane)

  – Approccio non parametrico: non si assume nessuna forma esplicita delle distribuzioni, queste vengono definite implicitamente dai dati di training

- La scelta del metodo ottimale dipende dalla presenza di informazioni a priori che giustifichino l'uso di certi modelli

Stima parametrica

- Si assume nota la forma della distribuzione di probabilità dei vettori di ogni classe (es. somma di gaussiane)
- Nel caso più semplice (singola gaussiana), occorre stimare solo il vettore medio e la matrice di covarianza di ogni classe
- La tecnica più usata è quella del Maximum Likelyhood (ML)
- Essa suppone che i parametri da stimare siano costanti ma sconosciuti. Ne desume quindi il valore massimizzando la probabilità di osservare i dati di training
- Si ipotizza l'indipendenza statistica tra le classi

Maximum likelyhood

- Sia θ l'insieme dei parametri da stimare per una singola classe, e sia D l'insieme dei dati di training ($X_1 \ldots X_n$). La probabilità di ottenere D dato θ vale

$$P(D \mid \theta) = \prod_{k=1}^{n} P(x_k \mid \theta)$$

- La stima ML di θ è quella che massimizza P(D| θ)

$$\hat{\theta} = \arg\max_{\theta} \left[p\left(D \middle| \theta\right) \right] = \arg\max_{\theta} \left[\log p\left(D \middle| \theta\right) \right]$$

- Operando con prodotti e con gaussiane conviene usare i logaritmi

$$\hat{\theta} = \arg\max_{\theta} \left[\log\left(\prod_{k=1}^{n} p\left(x_k \middle| \theta\right) \right) \right] = \arg\max_{\theta} \left[\sum_{k=1}^{n} \log\left(p\left(x_k \middle| \theta\right) \right) \right]$$

Soluzione ML

- Come ci si poteva aspettare, la soluzione ML della stima parametrica nel caso gaussiano scalare porta a

$$\hat{\mu} = \hat{\theta}_1 = \frac{1}{n}\sum_{k=1}^{n} x_k \qquad \hat{\sigma}^2 = \hat{\theta}_2 = \frac{1}{n}\sum_{k=1}^{n} (x_k - \hat{\mu})^2$$

- La generalizzazione al caso gaussiano N-dimensionale è banale

- Dato il *training set* D per una certa classe, si ottiene una stima ML della distribuzione gaussiana utilizzando come media e covarianza le quantità stimate sul training set stesso (medie campionarie).

- Se il numero di esempi è sufficientemente elevato, questo basta ad ottenere stime stabili

Approccio non parametrico

- Nell'approccio parametrico tradizionale tutte le distribuzioni di probabilità delle classi sono monomodali
- Sono possibili estensioni multimodali tramite misture (di stima non banale), ma la forma generale delle distribuzioni resta vincolata
- Nell'approccio non parametrico è possibile invece generare distribuzioni di forma arbitraria
- Sono possibili due meccanismi:
  - stima delle distribuzioni di probabilità condizionate $P(X|\omega_i)$
  - stima diretta delle probabilità a posteriori $P(\omega_i|X)$

Metodo della finestra di Parzen

- Si basa sull'assunzione che p(X|ω) è elevata nei punti X vicino ai quali il training contiene molti esempi della classe ω

- Dato un dominio di volume V centrato su X (es. un ipercubo di lato h a d dimensioni) possiamo scrivere il numero di campioni k contenuti in V sommando n termini del tipo

$$k = \sum_{i=1}^{n} \varphi\left(\frac{x - x_i}{h}\right) \qquad \varphi(u) = \begin{cases} 1 & |u_j| \le \frac{1}{2} \quad j = 1, \ldots, d \\ 0 & \text{altrimenti} \end{cases}$$

- La stima della densità di probabilità è quindi data da

$$p(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V} \varphi\left(\frac{x - x_i}{h}\right)$$

Metodo della finestra di Parzen

- La funzione φ può essere di forma generale, purchè sia ovunque positiva e ad integrale unitario. Per ragioni pratiche conviene anche che sia monotona decrescente in funzione della distanza dal punto di applicazione

- Con questo metodo P(X) non è mai nulla dove è presente almeno un campione, ed il suo valore dipende da quanti campioni sono presenti entro un distanza h

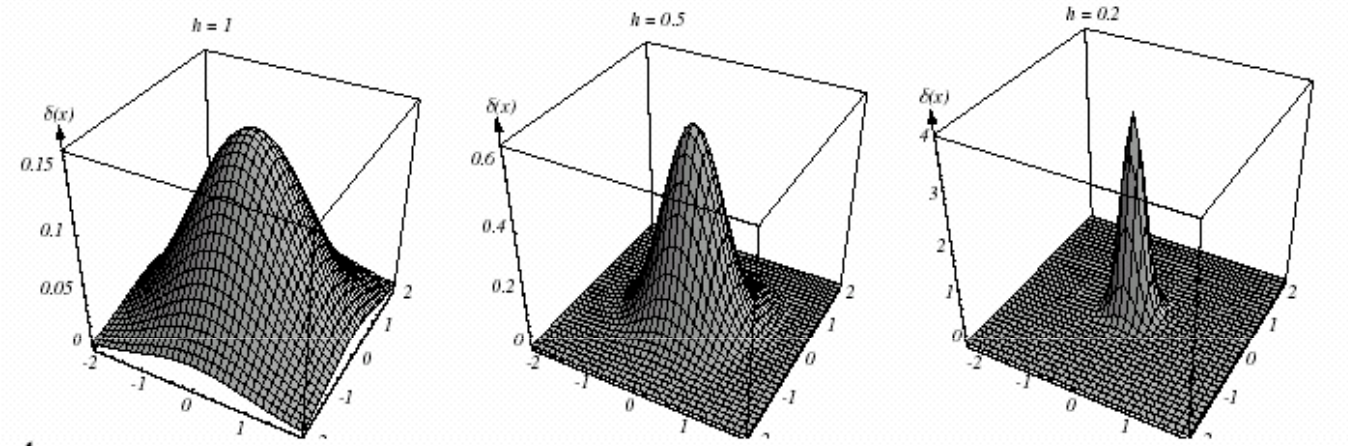
- Spesso φ è gaussiana, per cui si ha

$$\varphi(\mathbf{u}) = \frac{1}{\left(\sqrt{2\pi}\right)^d} \exp\left(-\frac{\|\mathbf{u}\|^2}{2}\right) \qquad p(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\left(\sqrt{2\pi}\right)^d h^d} \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2h^2}\right)$$

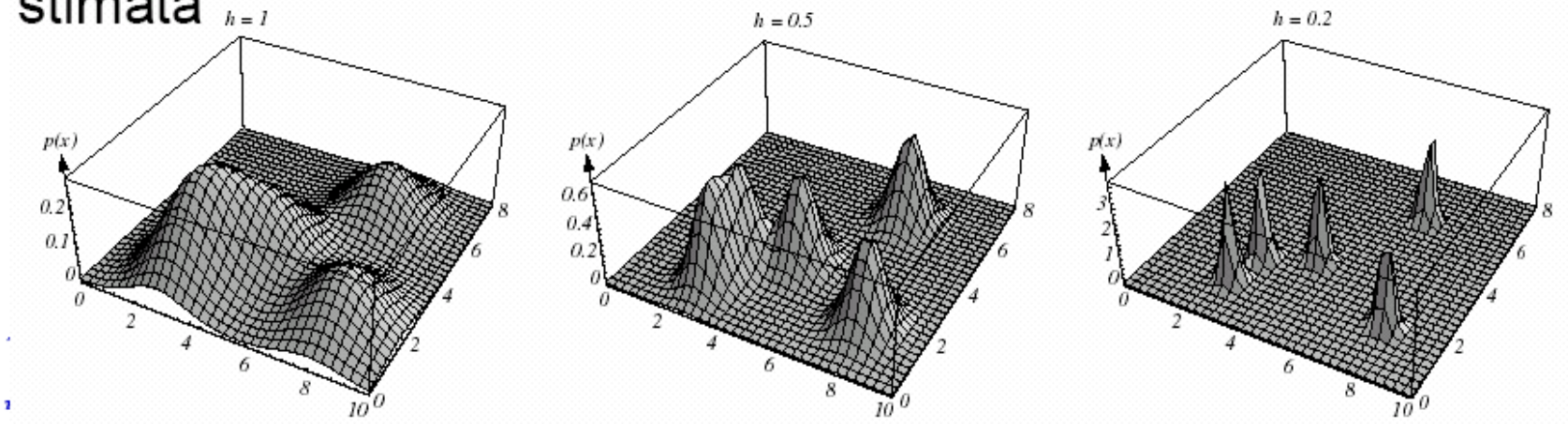
Esempio di stima

- A parità di campioni n, la stima dipende molto dalla forma di φ e dal valore di h

φ(u)

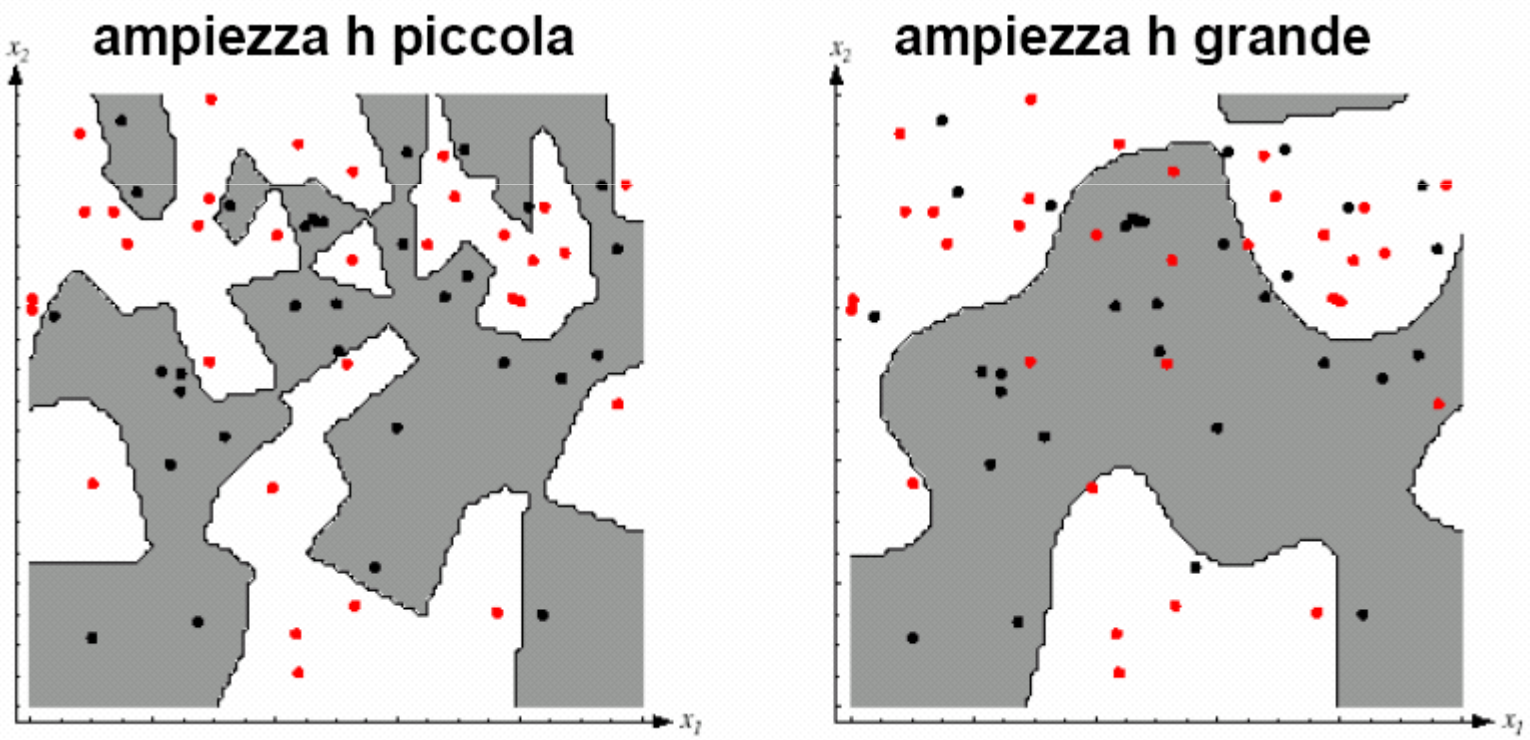


p(x) stimata



Esempio di classificazione

- La classificazione si esegue normalmente massimizzando la probabilità a posteriori. Il valore di h impatta sulla forma delle superfici di decisione ed è quindi molto critico



Problemi aperti

- Il valore di h ottimale dipende dal problema in esame e può essere definito solo attraverso test statistici della capacità di generalizzazione (es. cross-validation)
- Tutta la trattazione prevede una distanza di tipo euclideo, valida solo per features omogenee ed indipendenti
- Nei casi generali è utile normalizzare e standardizzare i dati prima di applicare il metodo della finestra
- Box-Cox è la normalizzazione più usata, scegliendo un λ opportuno

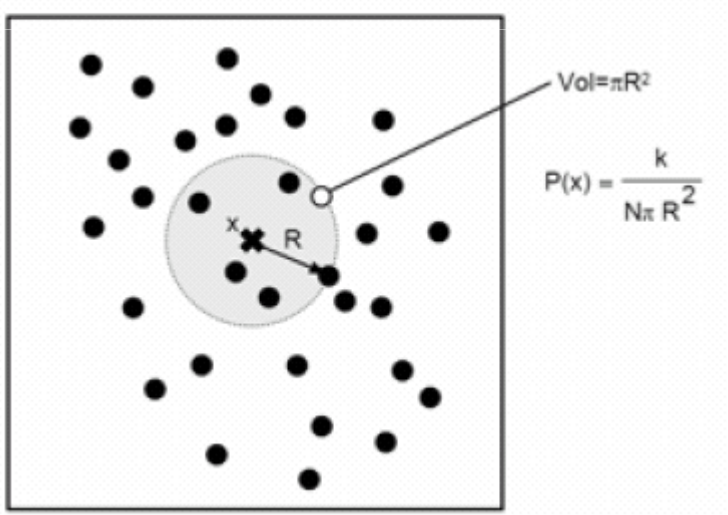
$$x'_\lambda = \frac{x^\lambda - 1}{\lambda}.$$

Stima a k vicini

- Nella stima a k vicini si considera un intorno del punto X tale da contenere almeno k campioni

- Si otterrebbe una stima della densità di probabilità come

$$p(x) = \frac{k}{n \cdot V(x)}$$



Stima a k vicini

- Il metodo si presta a stimare direttamente la probabilità a posteriori $P(\omega_i|X)$ nel caso in cui X non sia un elemento del *training set*

- Se ki è il numero di k vicini appartenenti alla classe i, che ha ni campioni su n totali, possiamo scrivere

$$p(\mathbf{x} \mid \omega_i) = \frac{k_i}{n_i \cdot V} \qquad p(\mathbf{x}) = \frac{k}{n \cdot V} \qquad P(\omega_i) = \frac{n_i}{n}$$

- Da cui

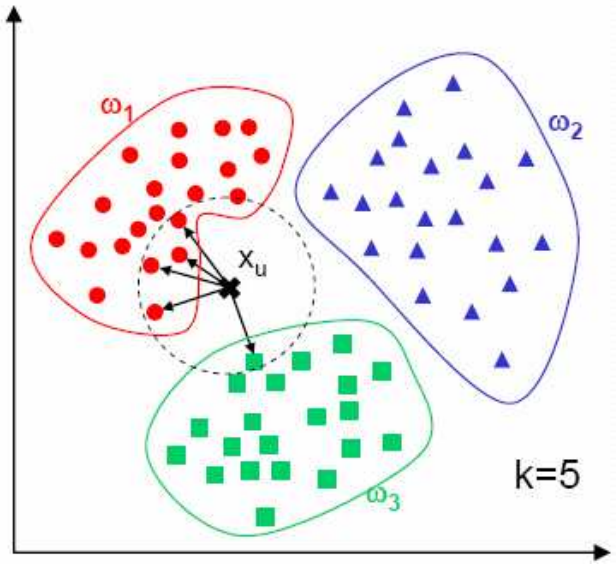
$$P(\omega_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \omega_i) P(\omega_i)}{p(\mathbf{x})} \cong \frac{k_i}{n_i \cdot V} \frac{n_i}{n} \frac{n \cdot V}{k} = \frac{k_i}{k}$$

Classificatore kNN

- Con il principio dei k vicini è facile realizzare un semplicissimo classificatore, detto kNN, in grado di rappresentare superfici di decisione qualunque

$$\alpha(\mathbf{x}) = \omega_i \qquad \omega_i = \arg\max_i \frac{k_i(\mathbf{x})}{k(\mathbf{x})}$$

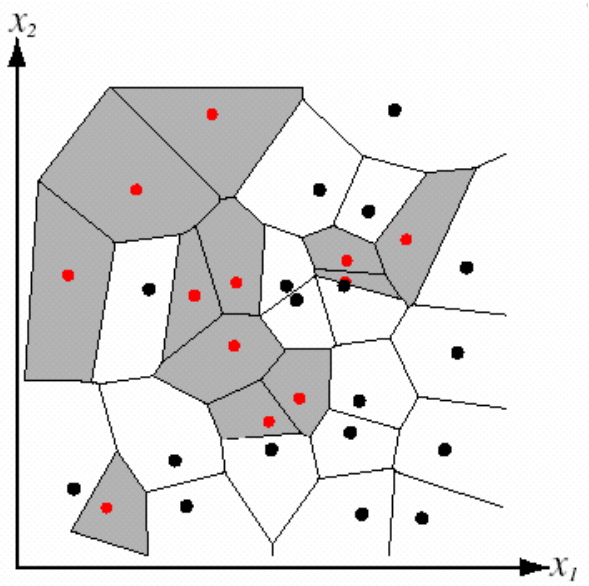
- Per realizzarlo è sufficiente:
  - Scegliere un valore di k
  - Raccogliere gli esempi
  - Classificarli
  - Definire una metrica



Classificatore NN

- Nei casi in cui occorre determinare rapidamente una classificazione di X, viene usato il classificatore kNN con k=1

- La classe di X è quella del campione di training più vicino. Questo induce una particolare tassellazione dello spazio delle *features*, detta tassellazione di Voronoi



Ottimizzazione kNN

- Nonostante l'apparente semplicità, realizzare classificatori non parametrici può essere computazionalmente molto oneroso, se il training set ha un numero di campioni elevato

- Esistono due approcci per ridurre drasticamente la complessità:

  – Considerare solo i campioni di frontiera tra classi diverse, cioè quelli che hanno tra i k vicini almeno un campione di classe diversa

  – Sostituire la popolazione di una classe con una sua versione quantizzata vettorialmente, con un numero di elementi fissato

Funzioni discriminanti

- Un approccio alternativo a quelli probabilistici prevede la costruzione diretta delle funzioni discriminanti a partire dai dati

- Si tratta sempre di un metodo parametrico, ma spesso la scelta (e la stima) del modello per tali funzioni è più semplice della scelta (e della stima) del modello probabilistico

- Particolare rilevanza hanno le funzioni discriminanti lineari
  - Semplicità analitica
  - Basso costo computazionale
  - Possibile combinazione in architetture complesse

Funzioni discriminanti lineari

- Una funzione discriminante lineare ha la forma

$$g(\mathbf{x}) = \mathbf{w}^t \mathbf{x} + w_0$$

- Le componenti del vettore w sono dette pesi

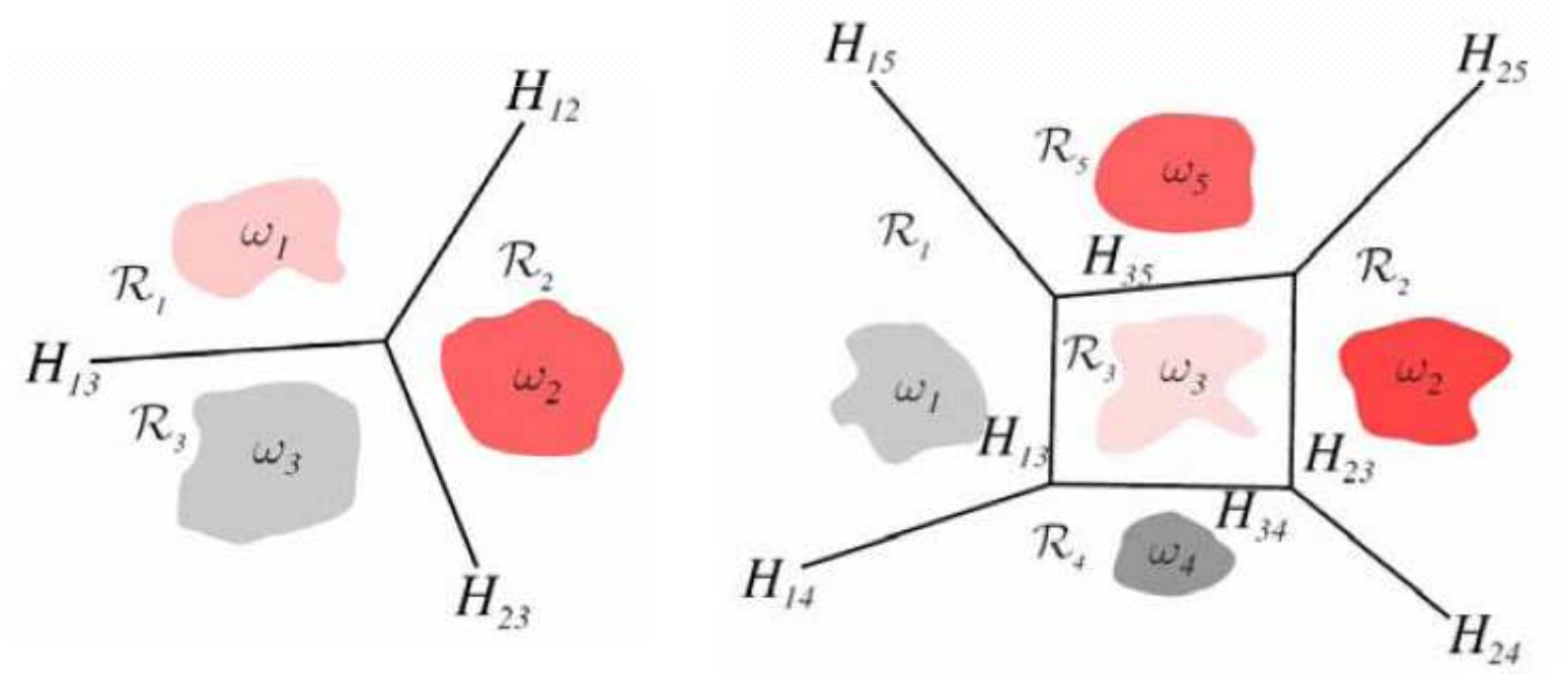
- La stima dei pesi viene condotta minimizzando una opportuna funzione di costo sul training set (es. errore o rischio)

- L'obiettivo è sempre quello di ottenere una regola generale valida non solo sul *training set* ma soprattutto nei test successivi

Funzioni discriminanti multiclasse

- Una sola funzione lineare è adatta per problemi a due classi.

- Nel caso multiclasse occorrono in generale tante funzioni quante sono le coppie di classi. Alcune possono essere poi ridondanti



Costruzione di una f.d.l.

- La stima di w viene svolta minimizzando una funzione di costo J(w)
- Se la funzione J è semplice, è sufficiente cercare un w_0 tale che

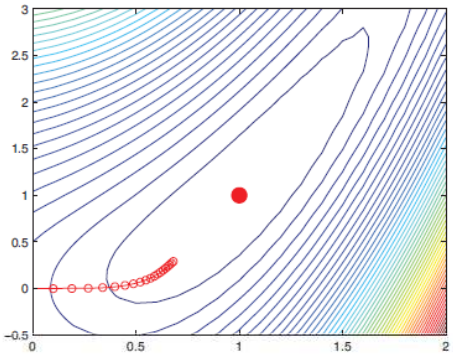
$$\nabla_{\mathbf{w}} J(\mathbf{w})\Big|_{\mathbf{w}_0} = 0$$

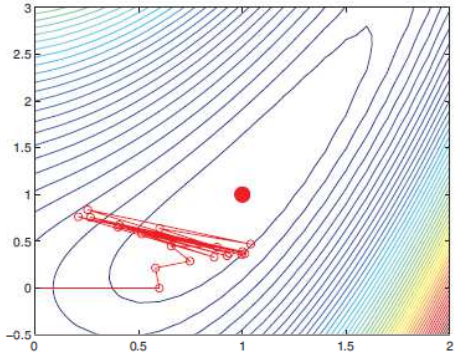
- Nel caso generale, la minimizzazione di ottiene tramite metodi di discesa del gradiente. In essi, a partire da un valore iniziale di w, ci si sposta nella direzione che porta alla maggiore diminuzione di J
- Alla k-esima iterazione si ha

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta \cdot \nabla J\big(\mathbf{w}(k)\big)$$

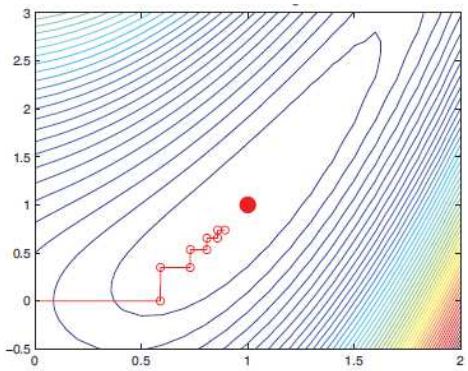
Problemi di convergenza

- Il valore dello stepsize η è molto critico





- Migliori risultati si ottengono scegliendo η variabile in base ad uno sviluppo in serie di potenze della funzione J, in base al quale si determina il valore ottimale ad ogni passo
  - primo ordine (line search)
  - secondo ordine (metodo di Newton)



Apprendimento

- Il parametro η (*learning rate*) regola la convergenza del metodo e la rapidità con cui viene ottenuta

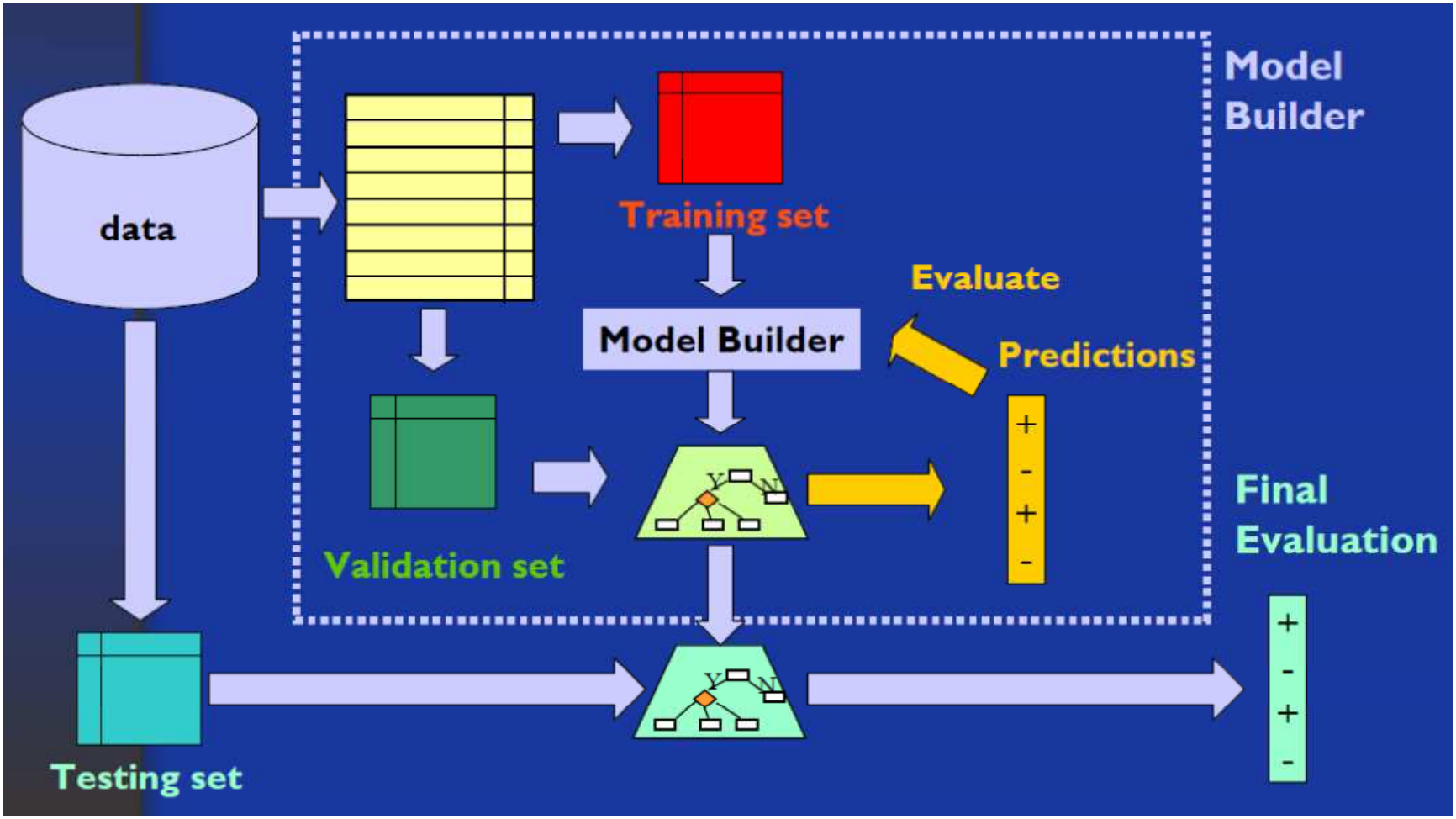
- Sulla base della forma di J(w) e del metodo di minimizzazione utilizzato, si realizzano diversi algoritmi di calcolo dei pesi ottimi (apprendimento)

- Gli algoritmi di apprendimento sono la componente fondamentale delle architetture di classificazione complesse basate su f.d.l.
  - reti neurali
  - support vector machines (SVM)

Valutazione delle prestazioni

- Quanto sono credibili le prestazioni dichiarate da un sistema di PR?
- Le % di errore valutata sul training set non è un buon indicatore delle prestazioni future (troppo ottimismo, rischio overfitting)
- Il semplice impiego di un test set diverso dal training set (per semplice suddivisione dei dati) non basta. Occorre definire anche degli intervalli di confidenza sulla % di errore
- E' anche utile, in fase di selezione del modello, usare un verification set per controllare la stabilità dei risultati durante l'addestramento e scegliere il modello più appropriato

Ciclo di costruzione/valutazione



Bilanciamento

- Spesso le classi hanno frequenze molto diverse, e questo crea problemi durante l'addestramento, e rende spesso illusorie le stime della % di errore. Casi tipici: malati/sani, transazioni illegali/legali, terroristi/viaggiatori

- Esempio: se c'è solo 1 malato ogni 1000 sani, un sistema di PR che risponde sempre "sano" ha in media il 99.9% di risposte corrette

- Non ci si deve far ingannare dai numeri. Un sistema di riconoscimento che dichiara 0.1% di falsi allarmi, su 1000000 di casi produce comunque 1000 falsi positivi da gestire!

- E' sempre utile "bilanciare" le popolazioni prima di iniziare l'addestramento, per evitare che classi rare siano ignorate. In alternativa, si deve usare uno schema con funzioni di rischio adeguate invece delle sole frequenze.

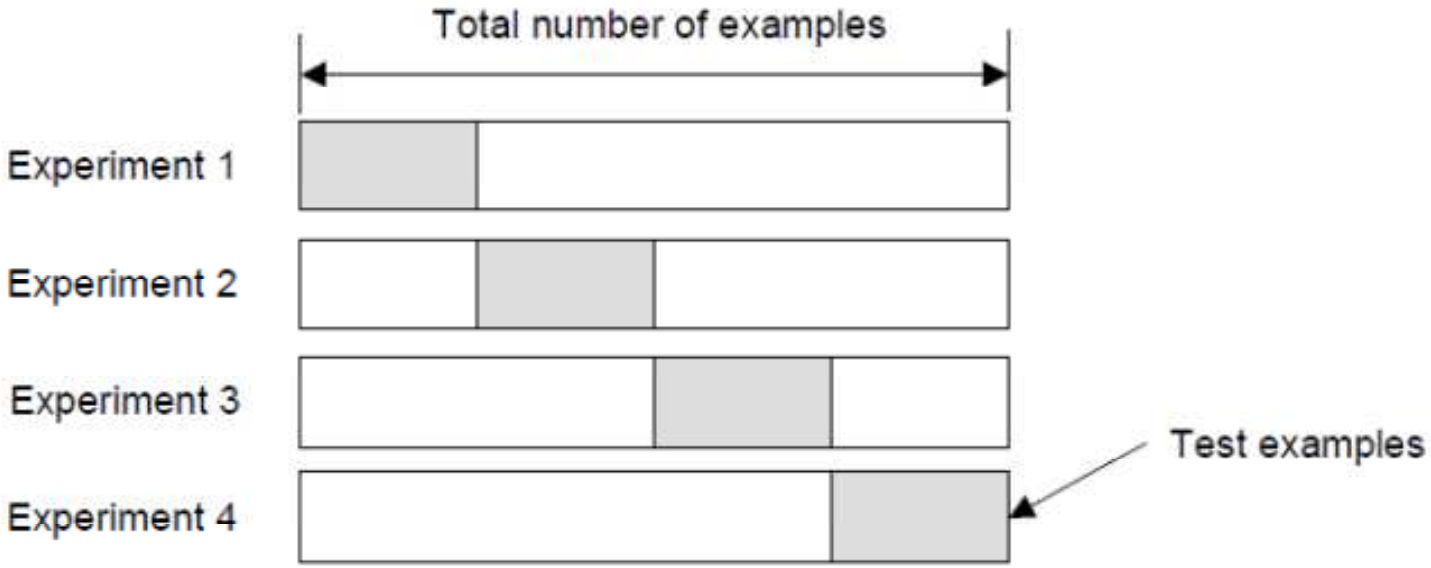
Intervalli di confidenza

- Dato un sistema di PR (binario) con *S* successi su *N* casi, una semplice stima di confidenza è fornita dalla statistica di Bernoulli
- La frequenza dei successi $f = S/N$ è distribuita con media p e varianza $p(1-p)/N$
- Per grandi *N*, la distribuzione è gaussiana e si possono definire degli intervalli di confidenza per la variabile standardizzata $z=(f-p)/\sigma$
- Esempio: N=100, f=75%, confidenza 80% fornisce $0.69<p<0.80$; con N=1000 invece $0.73<p<0.77$

Cross-validation

- E’ una tecnica per definire sperimentalmente intervalli di confidenza di un sistema di PR
- I dati vengono suddivisi in *K* parti uguali (spesso *K=10*). Una parte viene usata come test set, e le altre *K-1* come training set
- Vengono addestrati e valutati *K* classificatori. La % di successo media e l’intervallo di confidenza sono ottenuti analizzando la distribuzione delle *K* stime % ottenute
- Tutti i dati vengono quindi usati come training e come test ma mai simultaneamente

Cross-validation



$$E = \frac{1}{K}\sum_{i=1}^{K} E_i$$

- Caso estremo: K=N (Leave-One-Out CV)

Comparazione di classificatori

- La comparazione oggettiva di classificatori dovrebbe essere basata sulla rilevazione delle curve ROC (molto onerosa)

- E' più semplice sfruttare la cross-validation per produrre due insiemi di stime di successo $x_1 \ldots x_K$ e $y_1 \ldots y_K$ e chiedersi se queste possono o meno essere generate dalla stessa distribuzione

- Le quantità x e y, ed anche la loro differenza $m=x-y$, per K bassi seguono la distribuzione di Student. Scelto quindi un livello di significatività % si determina l'intervallo di confidenza per *m* medio

- Se l'intervallo non include lo zero, i classificatori sono diversi in modo significativo