Riconoscimento e recupero dell’informazione per bioinformatica

Classificazione: validazione

Manuele Bicego

Corso di Laurea in Bioinformatica

Dipartimento di Informatica - Università di Verona

Introduzione

- Validazione del classificatore:
  - Capire se il sistema di classificazione disegnato rappresenta una buona scelta
  - In questo caso, tipicamente, si va a misurare quanto “buono” è un sistema in termini di capacità di classificare
    - Non si considerano in questa fase altri fattori, come usabilità, efficienza, velocità, portabilità, accettabilità, etc etc
- Spesso utilizzato per comparare diverse possibili scelte
  - Es. In questo problema, è meglio Parzen Windows o KNN?
- Ci sono diversi modi per validare un classificatore, ma l'obiettivo principale è misurare la “capacità di generalizzazione”

Capacità di generalizzazione

DEFINIZIONE di capacità di generalizzazione: capacità di classificare correttamente anche oggetti sconosciuti (non presenti nel training set)

- Dipende da come si è costruito il sistema di classificazione
  - Scelta del modello e dei suoi parametri, ampiezza e completezza del training set, metodo di addestramento, etc etc

Capacità di generalizzazione

- NOTA: non è detto che classificare bene gli oggetti presenti nel training set implichi una buona capacità di generalizzazione

MEMORIZZARE vs APPRENDERE

- Overtraining: il sistema ha imparato talmente bene i pattern del training set che non è più in grado di generalizzare (ha memorizzato, non ha appreso)

Capacità di generalizzazione

- Quindi, è buona norma, per testare la capacità di generalizzazione, avere a disposizione un altro insieme, chiamato TESTING SET (non utilizzato per costruire il classificatore):
  - Insieme che contiene oggetti del problema
  - Oggetti per cui si conosce la classe “vera”

Il testing set è utilizzato per “testare” le capacità discriminative del classificatore costruito

…. ma cosa vuol dire “testare”?

Testing

Diversi approcci, il più semplice implica di “contare gli errori” sul testing set

- Si prende ogni oggetto del testing set
- Lo si classifica con il classificatore appena costruito
- Si confronta la classe assegnata dal classificatore con la classe vera (che si conosce a priori)
  - Se non coincidono si ha un errore
- Si determina la percentuale di errore su tutto il testing set

Testing

Esempio: dieci oggetti nel testing set

Classe Vera	Classe Assegnata	Errore?
1	1	Corretto
1	1	Corretto
1	1	Corretto
1	2	ERRORE
1	1	Corretto
2	1	ERRORE
2	2	Corretto
2	2	Corretto
2	2	Corretto
2	1	ERRORE

Errore di classificazione

3/10 = 0.3 (30%)

Come ottenere il Testing Set

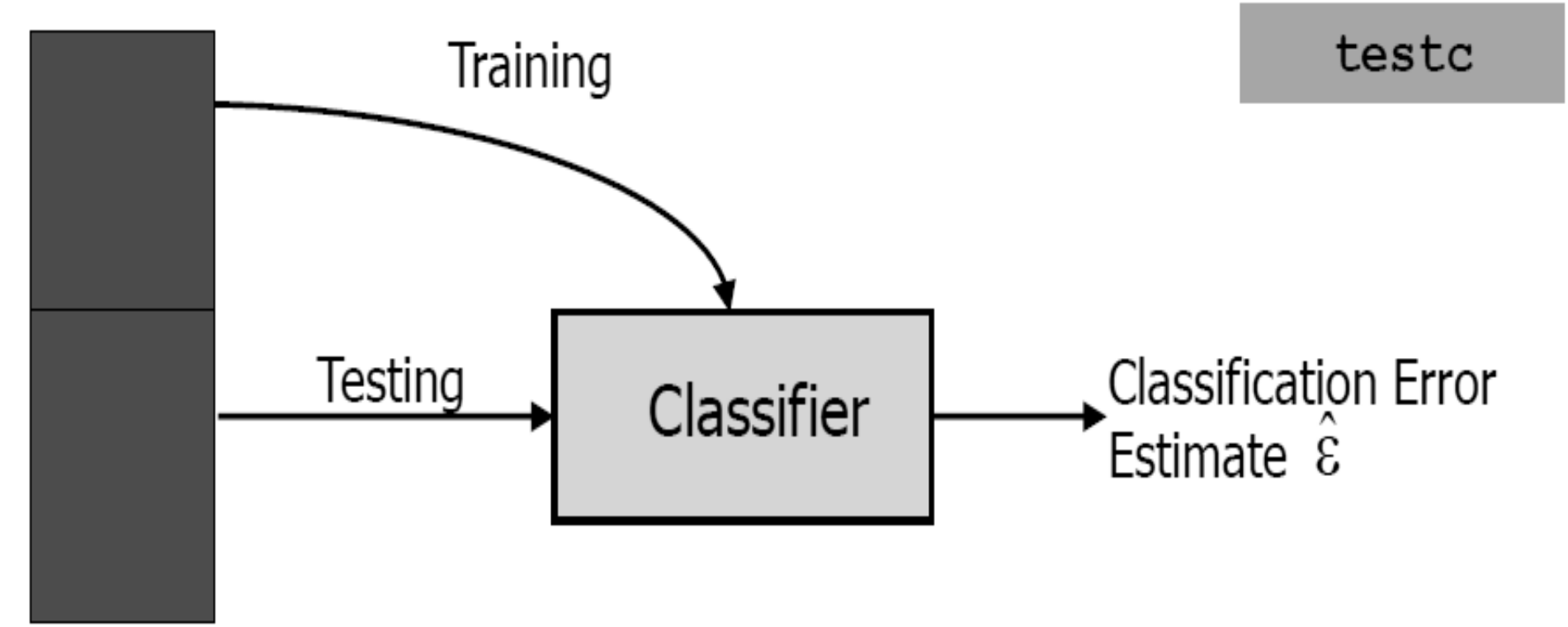
SOLUZIONE IDEALE:

- riestrarre altri esempi dal problema (nuovo campionamento) e utilizzarli per testare il sistema (o meglio, mettere il sistema “in funzione”)
  - purtroppo questo può risultare troppo dispendioso o non fattibile

SOLUZIONE ADOTTATA IN PRATICA:

- suddividere l'insieme a disposizione in due parti
  - utilizzare una parte per costruire (addestrare) il sistema di classificazione
  - utilizzare l’altra parte per testare
  - (In questo modo il sistema è testato su oggetti che non ha mai visto)

Come ottenere il Testing Set



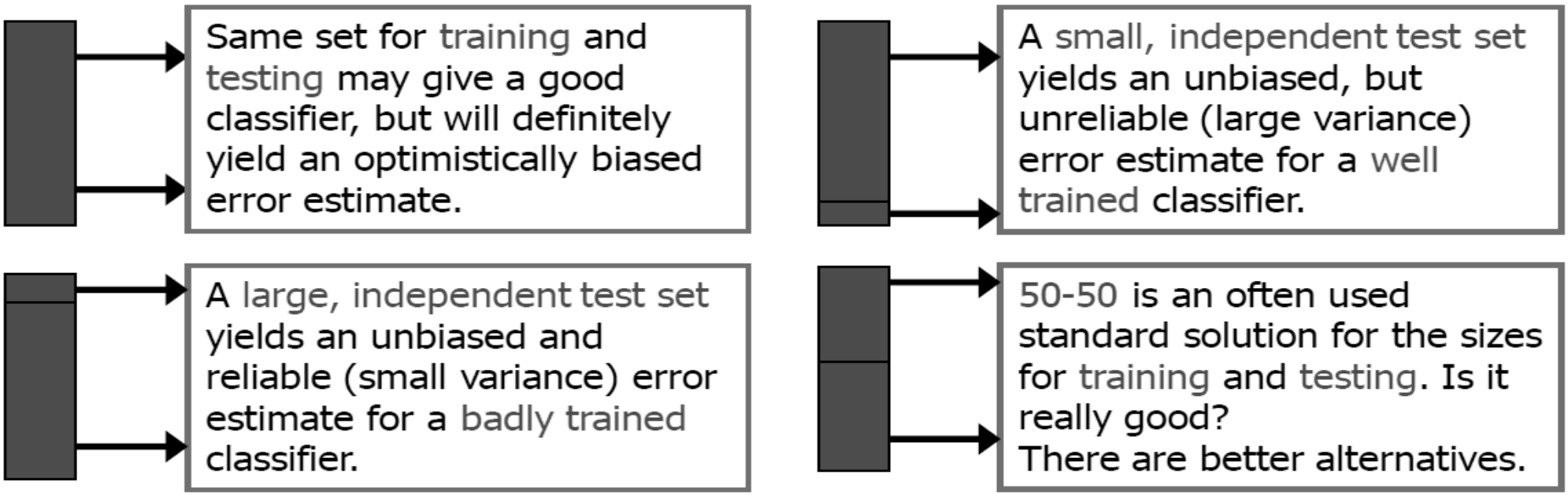
La Cross Validation

- Metodo più utilizzato per la suddivisione tra training e testing

- Tipicamente utilizzato anche per fare confronti tra diverse soluzioni:
  - Confronto tra diversi classificatori
    - Esempio: (KNN) vs (SVM)
  - Confronto tra diverse versioni di un classificatore
    - Esempio: (SVM con kernel rbf) vs (SVM con kernel lineare)
  - Confronto tra diverse parametrizzazioni
    - Esempio: (KNN con K=1) vs (KNN con K=3)

La Cross Validation

Come effettuare la suddivisione



⇨ Occorre trovare un compromesso tra dimensione del training (efficacia dell'addestramento) e dimensione del testing (significatività della stima dell'errore)

La Cross Validation

- Diverse varianti:
  - Holdout
  - Averaged Holdout
  - Leave One-Out
  - Leave K-Out

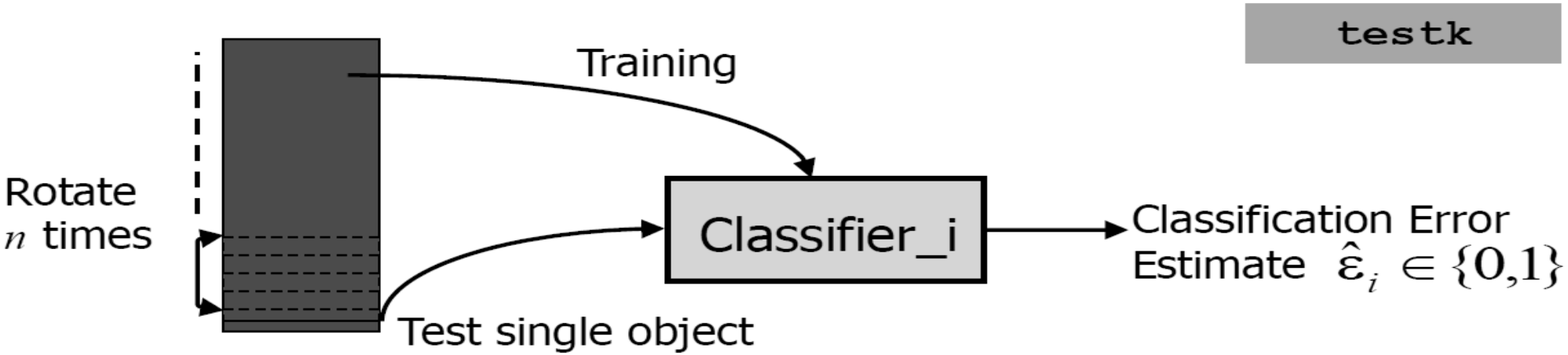
- *Holdout*
  - L'insieme dei dati viene partizionato casualmente in due sottoinsiemi disgiunti di eguale dimensione;
  - uno dei due sottoinsiemi viene utilizzato come Training e l'altro come Testing;

- *Averaged Holdout*
  - per rendere il risultato meno dipendente dalla partizione scelta, si mediano i risultati calcolati su più partizioni holdout;
  - le partizioni sono costruite casualmente, oppure in modo esaustivo;

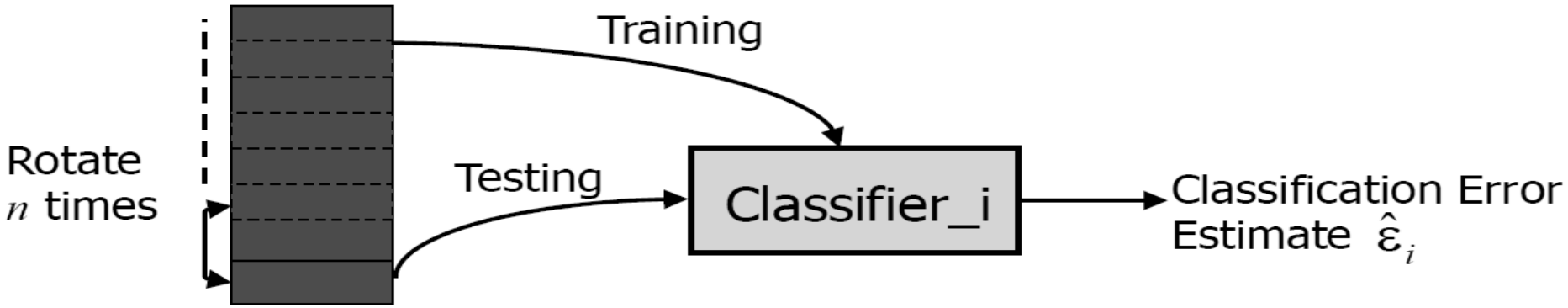
Leave One-Out: dato un insieme di dati X (che contiene N elementi):

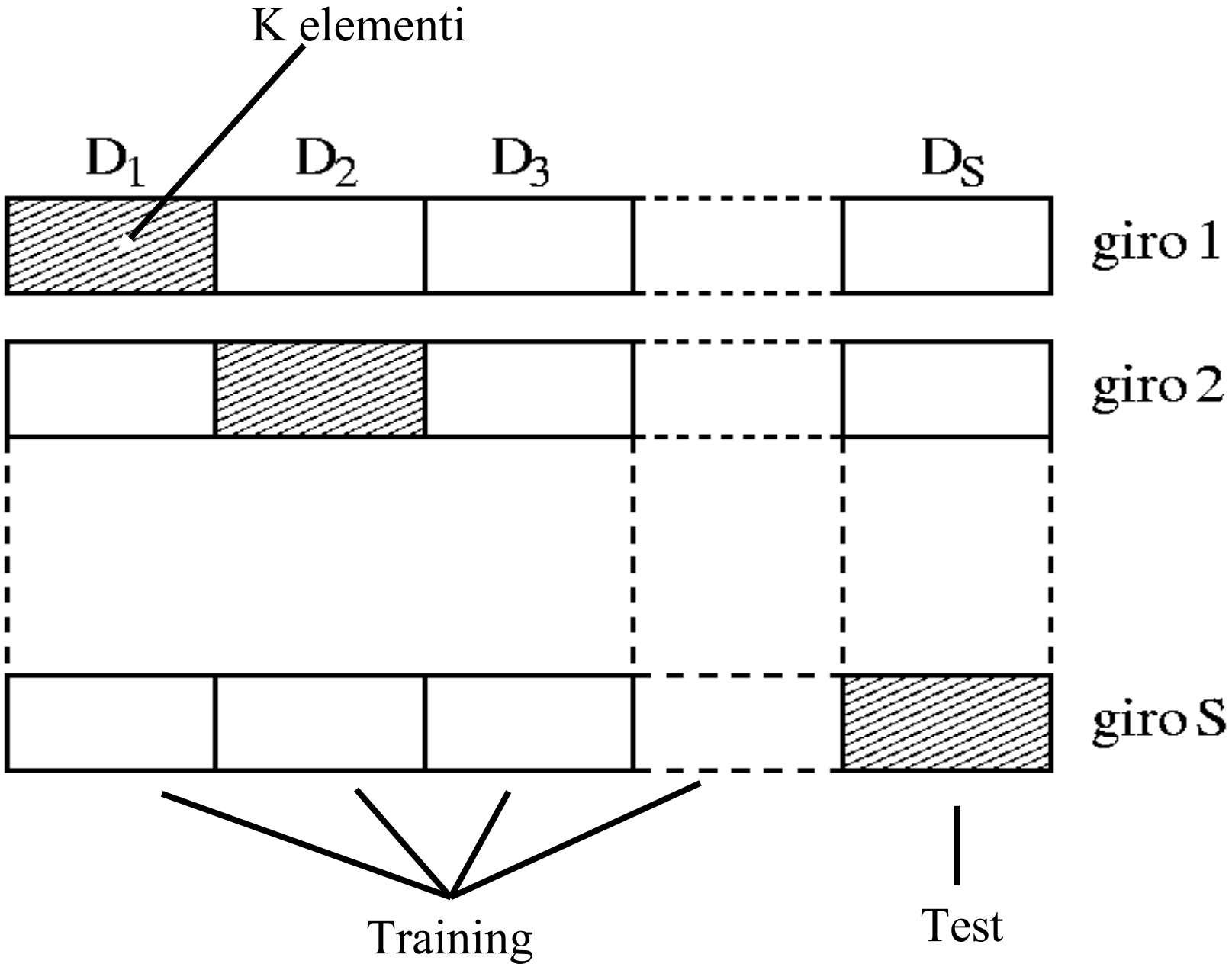
- Si sceglie un oggetto xi
- Si utilizza X\{xi} (tutti gli elementi tranne xi) per costruire il classificatore
- Si testa il classificatore con xi

- Si ripete l'operazione per tutti gli xi possibili e si media il risultato finale



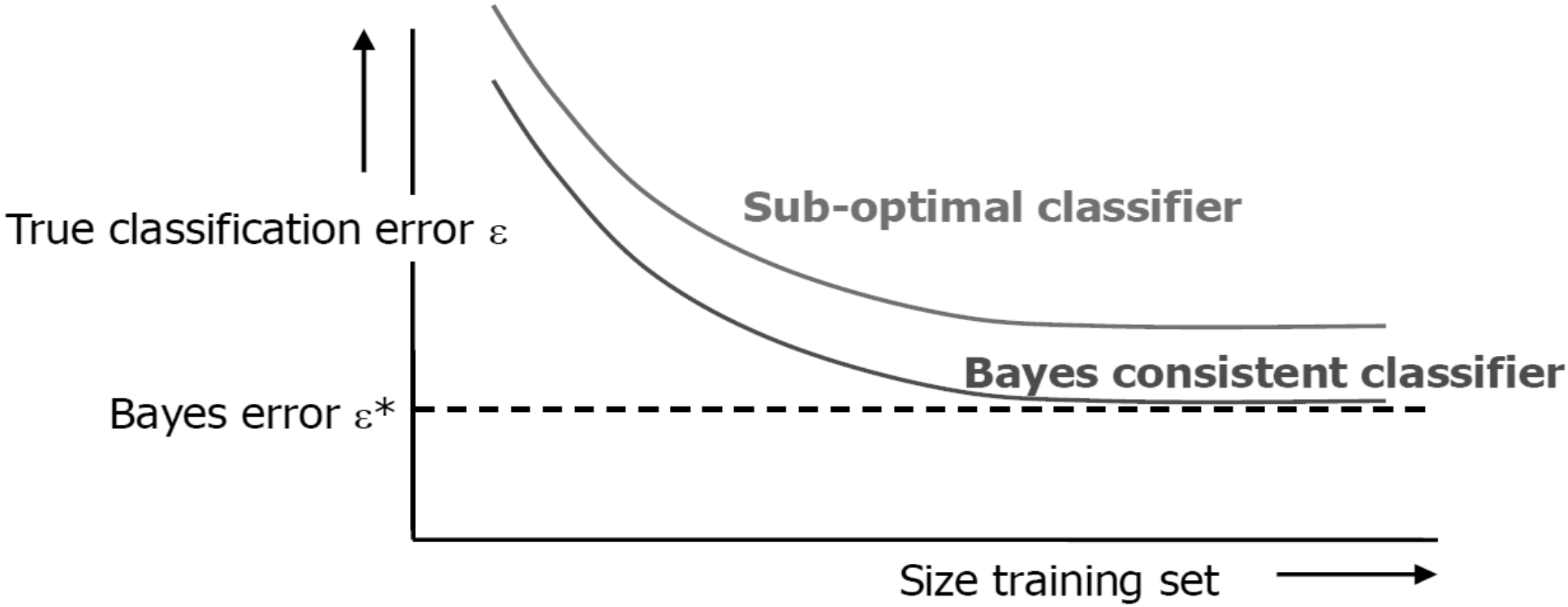
- *Leave K-Out (o S-fold cross validation)*: generalizzazione della tecnica precedente;
  - l'idea è quella di suddividere l'insieme dei dati in S segmenti distinti e casuali;
  - si realizza il classificatore utilizzando S-1 segmenti, mentre lo si testa utilizzando il segmento rimanente;
  - questa operazione viene effettuata S volte, variando a turno il segmento del Test Set;
  - infine la capacità di generalizzazione viene mediata tra gli S risultati.




K elementi
D1
D2
D3
DS
giro 1
giro 2
giro S
Training
Test

Altri Strumenti: learning curves

⇨ Variazione dell'errore al variare della dimensione del training set



Altri strumenti

- Nota: l'errore di classificazione non sempre ci permette di capire o confrontare completamente due classificatori

ESEMPIO

Vera	Asse-gnata
1	1
1	1
1	1
1	1
1	1
1	1
1	1
2	1
2	1
2	1

Classificatore 1: assegno un oggetto sempre alla prima classe

ERRORE: 3/10 = 0.3 (30%)

Vera	Asse-gnata
1	1
1	2
1	2
1	1
1	1
1	1
1	1
2	2
2	2
2	1

Classificatore 2

ERRORE: 3/10 = 0.3 (30%)

L'errore è uguale, ma i classificatori sono molto diversi!

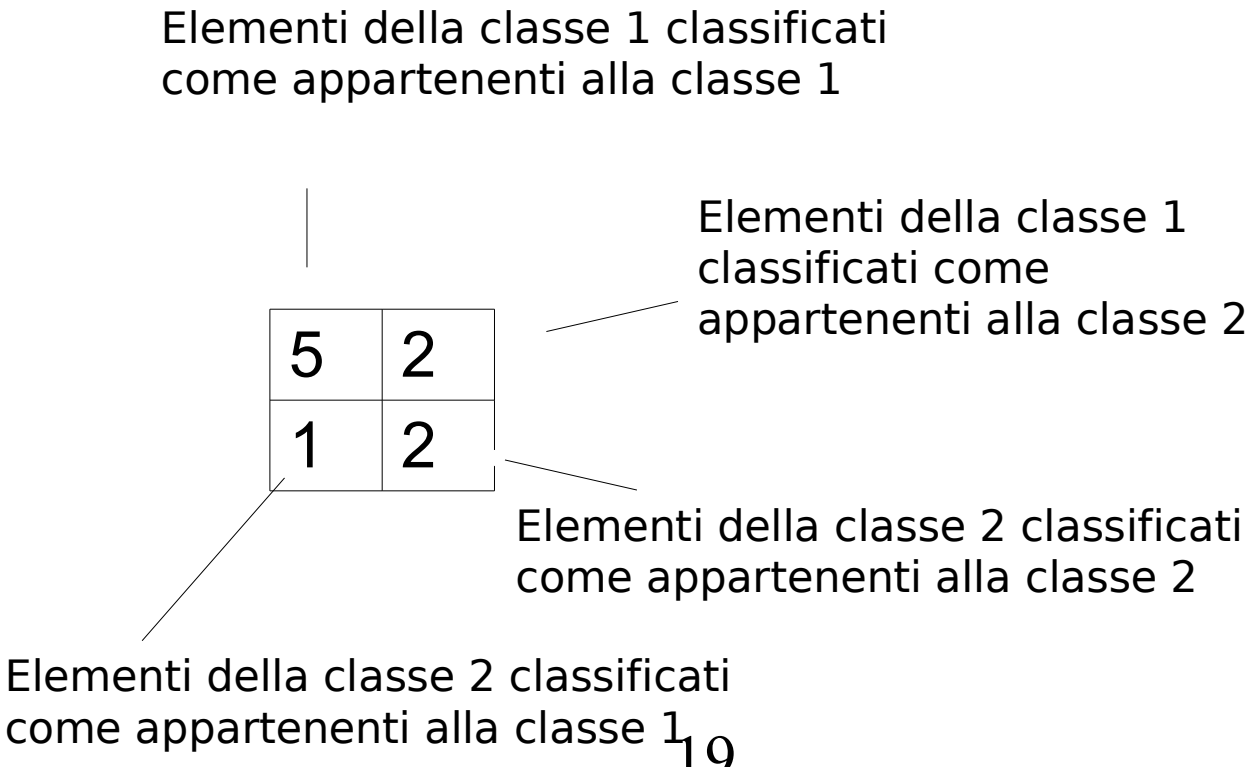
Matrici di confusione

- Matrici che ci dicono come un classificatore funziona rispetto alle diverse classi

  A(i,j) = numero di elementi della classe *i* classificati come elementi della classe *j*

ESEMPIO

Vera	Asse-gnata
1	1
1	2
1	2
1	1
1	1
1	1
1	1
2	2
2	2
2	1



Matrici di confusione

- L'errore di classificazione può essere calcolato facilmente dalla matrice di confusione
  - La somma di tutti gli elementi fuori dalla diagonale
  - O, meglio, può essere calcolato come “1-accuratezza”
  - Accuratezza: somma degli elementi della diagonale/numero elementi totali

Matrici di confusione per problemi a 2 classi

- Nel caso di problema a due classi la matrice di confusione assume una forma particolare (2 classi, positivi vs negativi)

True Positive (TP)	False Negative (FN)
False Positive (FP)	True Negative (TN)

ESEMPIO: classificazione tra malati (positivi) e sani (negativi)

- CLASSIFICAZIONE CORRETTA:
  - Veri positivi: pazienti malati classificati come malati
  - Veri negativi: pazienti sani classificati come sani
- CLASSIFICAZIONE ERRATA:
  - Falsi positivi: pazienti sani classificati come malati
  - Falsi negativi: pazienti malati classificati come sani

Indici

- Dalla matrice di confusione possono essere calcolati diversi indici

Indice	Formula	Intuizione
Accuracy	$\frac{TP+TN}{TP+FP+TN+FN}$	Percentuale di classificazioni corrette
Precision	$\frac{TP}{TP+FP}$	Percentuale di classificazioni positive che sono corrette
Recall (Sensitivity)	$\frac{TP}{TP+FN}$	Percentuale di elementi positivi del testing set che sono stati classificati come positivi
Specificity	$\frac{TN}{TN+FP}$	Percentuale di elementi negativi del testing set che sono stati classificati come negativi

Precision: se dico “positivo”, faccio giusto?

Recall: riesco a trovare tutti i positivi del testing set?

Indice	Formula
Accuracy	$\frac{TP + TN}{TP + FP + TN + FN}$
Precision	$\frac{TP}{TP + FP}$
Recall (Sensitivity)	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FP}$

Vera	Assegnata
1	1
1	1
1	1
1	1
1	1
1	1
1	1
2	1
2	1
2	1

Matrice di confusione

TP: 7	FN: 0
FP: 3	TN: 0

Accuracy: 7/10 (0.7)

Precision: 7/10 (0.7)

Recall: 7/7 (1)

Specificity: 0/3 (0)

Vera	Assegnata
1	1
1	2
1	2
1	1
1	1
1	1
1	1
2	2
2	2
2	1

Matrice di confusione

TP: 5	FN: 2
FP: 1	TN: 2

Accuracy: 7/10 (0.7)

Precision: 5/6 (0.83)

Recall: 5/7 (0.71)

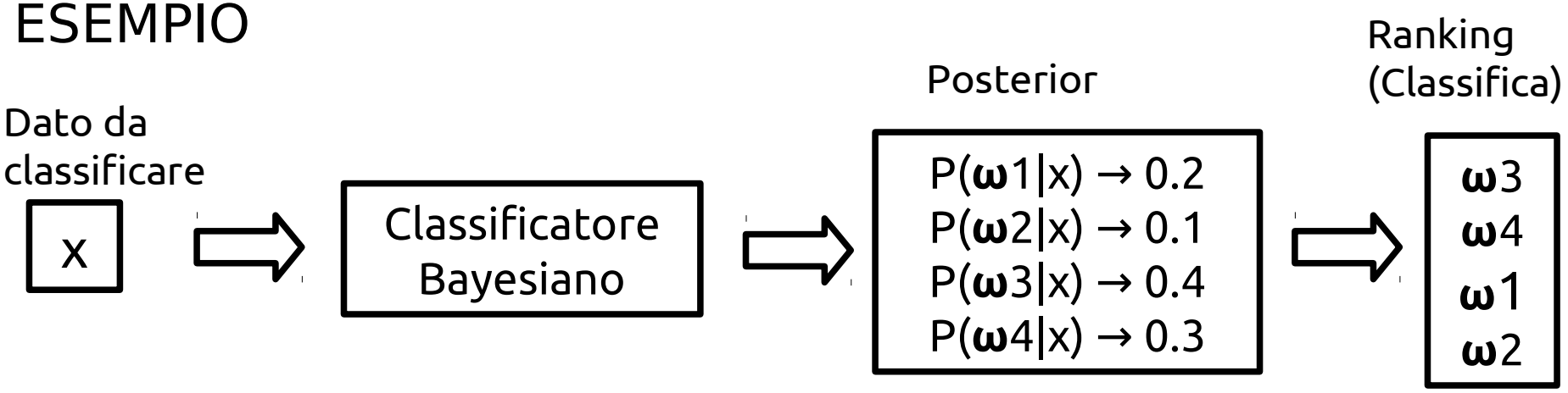
Specificity: 2/3 (0.66)

La curva CMC

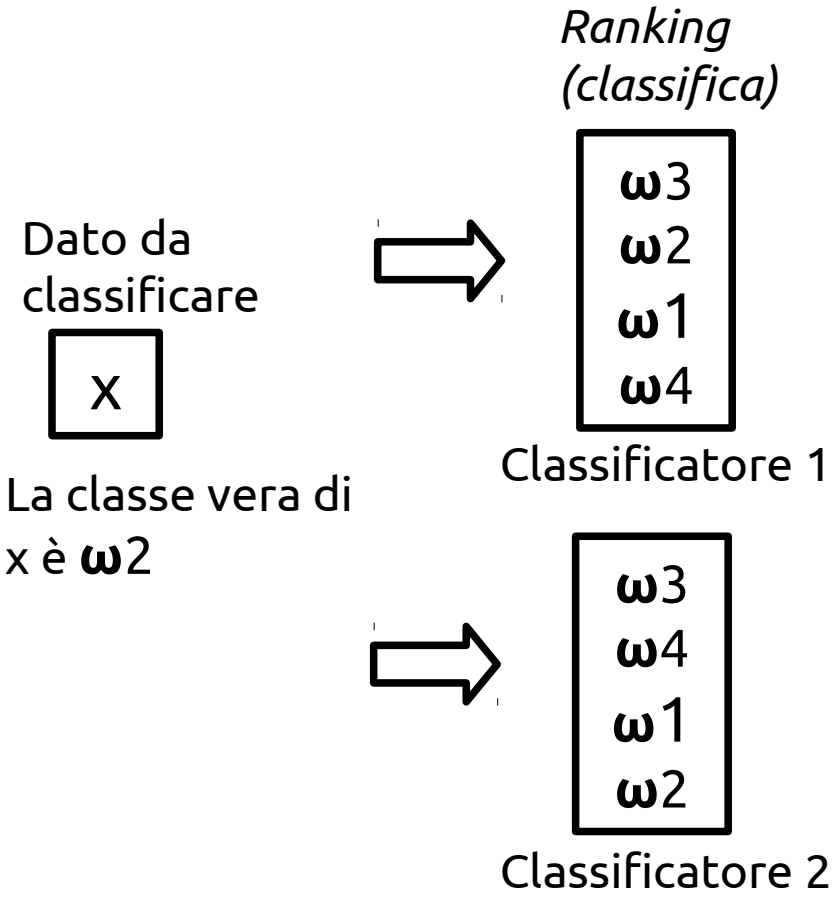
NOTA: con le tecniche viste finora non viene fornita nessuna informazione sulla “gravità” degli errori (“errore lieve” oppure “errore grave”?)

⇨ Informazione cruciale!

ESEMPIO



La curva CMC



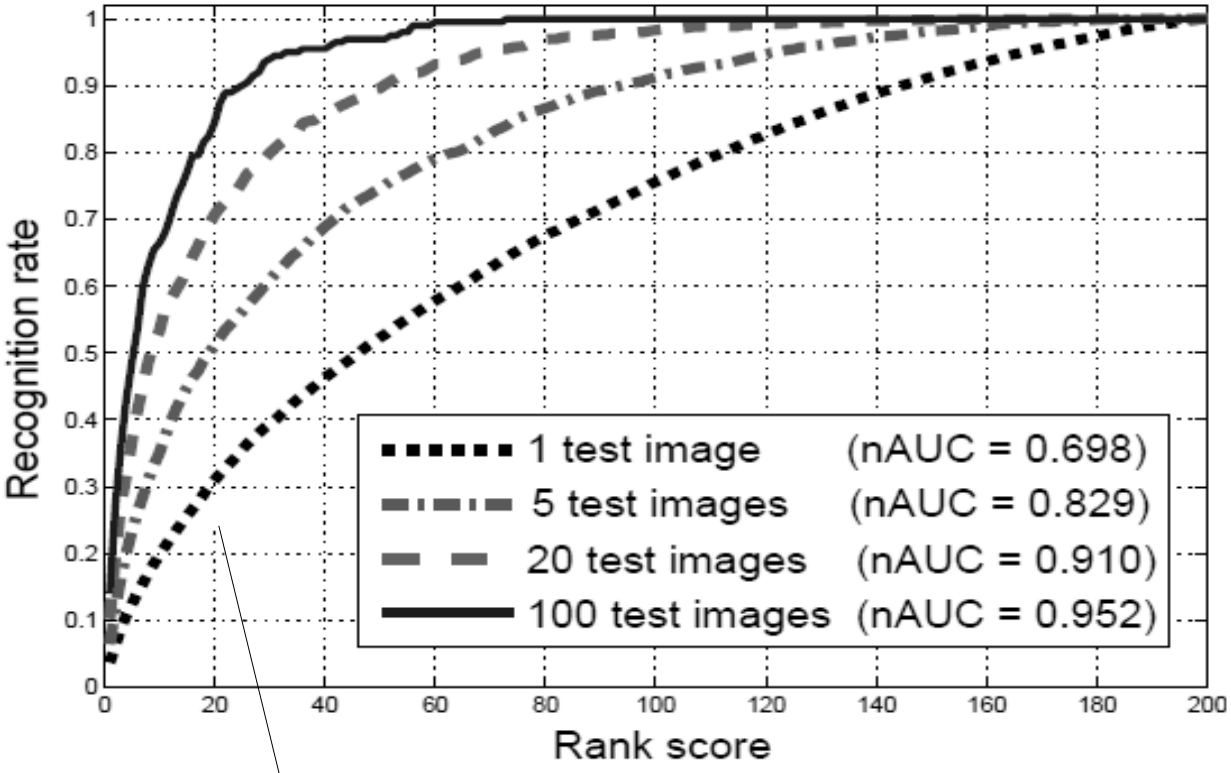
Nota: la classe vera di x è 2, quindi entrambi i classificatori sbagliano.

Ma il classificatore 1 sbaglia di poco (ha messo la classe 2 al secondo posto nella sua classifica), mentre il classificatore 2 sbaglia di molto (mette la classe 2 all'ultimo posto)

Il sistema di “contare gli errori” non permette di effettuare questa analisi → si usa la curva CMC

La curva CMC

Curva che in posizione **n** conta la frequenza con cui la classe corretta viene trovata entro le prime **n** posizioni del ranking

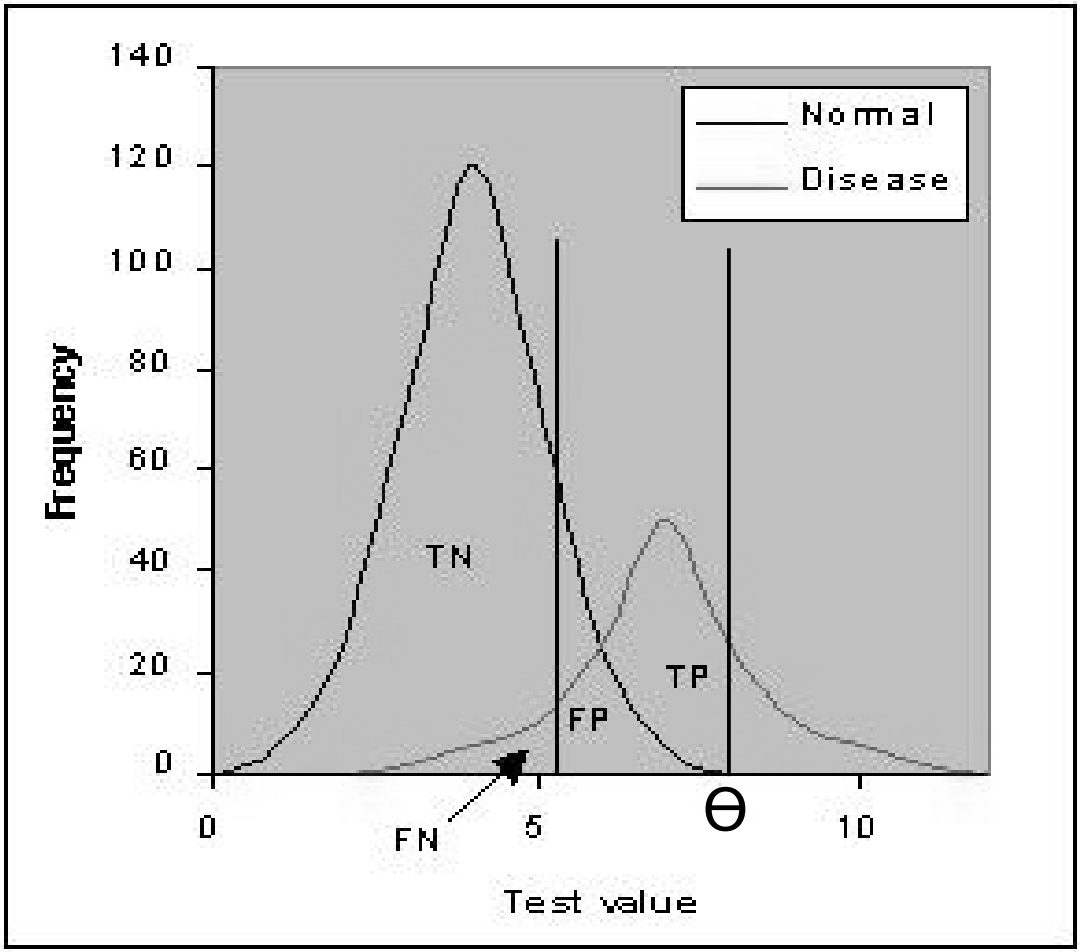


In prima posizione c'è l'errore di classificazione (numero di volte che la classe corretta è stata trovata al primo posto)

Percentuale di volte che il classificatore ha trovato la classe corretta nelle prime 20 del suo ranking

Un altro strumento: la curva ROC

⇨ Sistema molto utilizzato per valutare un classificatore binario basato su soglia

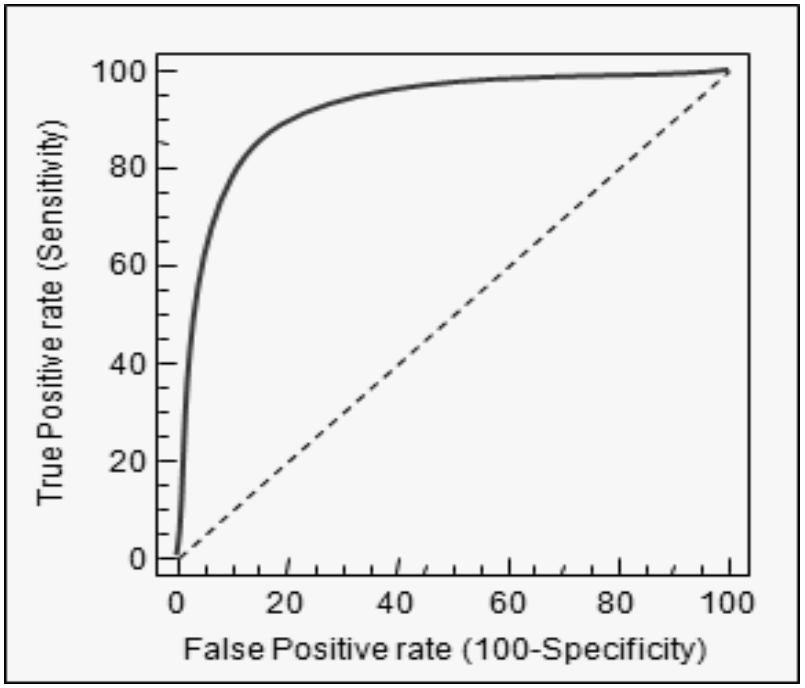


Variando la soglia Θ si ottengono diversi valori di TP, TN, FP e FN

Esempio: con il valore di Θ verde i Falsi Positivi sono a zero

La curva ROC

- La curva ROC mette in relazione la specificity con la sensitivity al variare della soglia
  - Fissata una soglia, quanti sono i veri positivi rispetto ai falsi positivi?

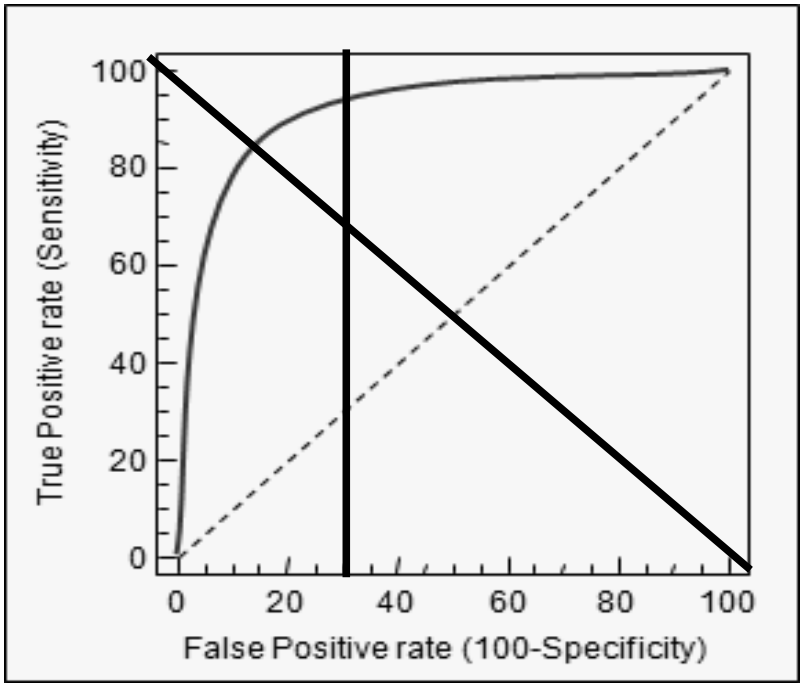


Come si calcola:

Si fa variare la soglia calcolando i corrispondenti veri positivi e falsi positivi, che rappresentano un punto della curva

Il valore minimo/massimo della soglia è quello per cui sono tutti falsi positivi o tutti veri positivi

A seconda di come si vuole operare si sceglie la soglia

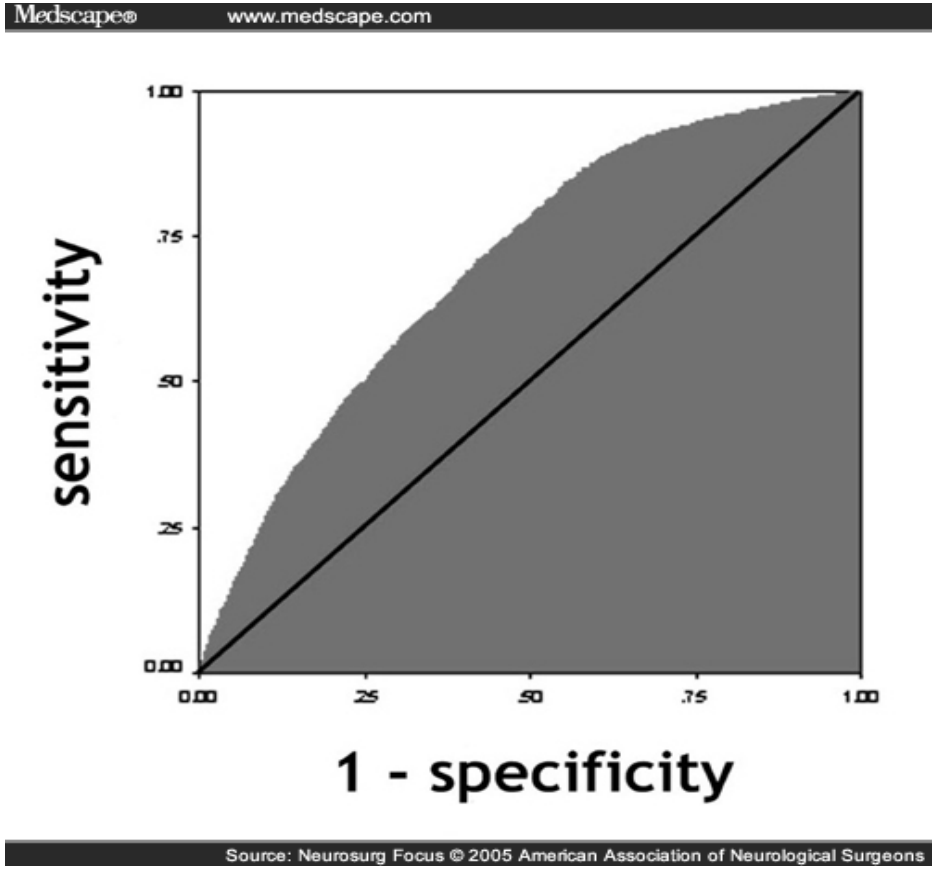


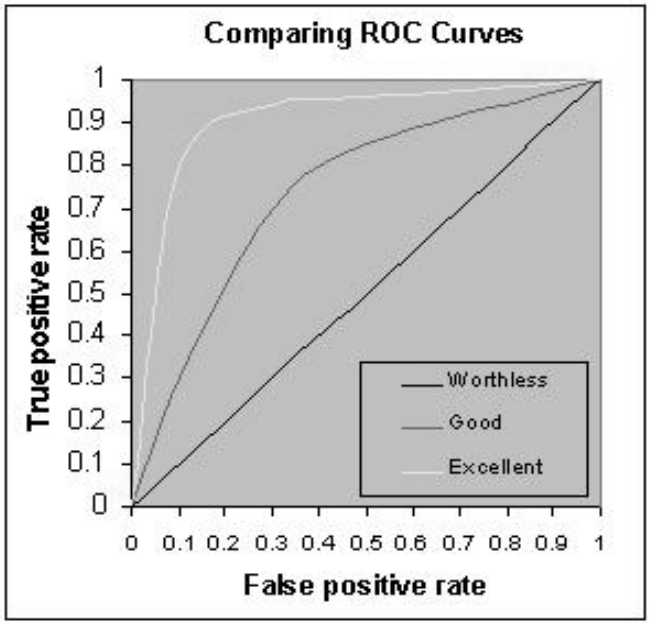
Esempio 1: Sensitivity al 95%, si ottiene un corrispondente valore di Specificity (70%)

Esempio 2: Sensitivity = 100 - Specificity, si chiama Equal Error Rate

L'area sotto la curva ROC

⇨ Si possono confrontare curve ROC calcolando l'area sotto la curva (AUC – Area Under the Curve)





Un AUC più grande implica un classificatore migliore

Significatività statistica

Esempio:

- ⇨ Si hanno due classificatori: KNN e SVM
- ⇨ Si calcola l'errore con Averaged Holdout (30 ripetizioni)
- ⇨ KNN ha un errore medio (sulle 30 ripetizioni) di 0.15, SVM ha 0.12

Possiamo dire che SVM è meglio di KNN?

No, senza un'analisi della significatività statistica!

(mi dice se questi risultati sono sensati o sono dettati dal caso)

Significatività statistica

- Diversi metodi per determinare la significatività statistica
  - T-test, anova, standard error of the mean, ...

- Il più semplice: la deviazione standard (radice quadrata della media)

Se si assume che l'errore calcolato sia gaussiano, allora la deviazione standard ci fornisce dei limiti di significatività

ESEMPIO:
KNN: errore medio 0.15, σ = 0.005
SVM: errore medio 0.12, σ = 0.001

Sono significativamente diversi!