

Pattern Recognition un'introduzione



By the time they are five years old, most children can recognize digits and letters. Small characters, large characters, handwritten, machine printed, or rotated—all are easily recognized by the young. The characters may be written on a cluttered background, on crumpled paper or may even be partially occluded. We take this ability for granted until we face the task of teaching a machine how to do the same. Pattern recognition is the study of how machines can observe the environment, learn to distinguish patterns of interest from their background, and make sound and reasonable decisions about the categories of the patterns. In spite of almost 50 years of research, design of a general purpose machine pattern recognizer remains an elusive goal.



A.K. Jain, R.P. Duin, J. Mao
“Statistical Pattern Recognition: a Review”



TABLE 1
Examples of Pattern Recognition Applications

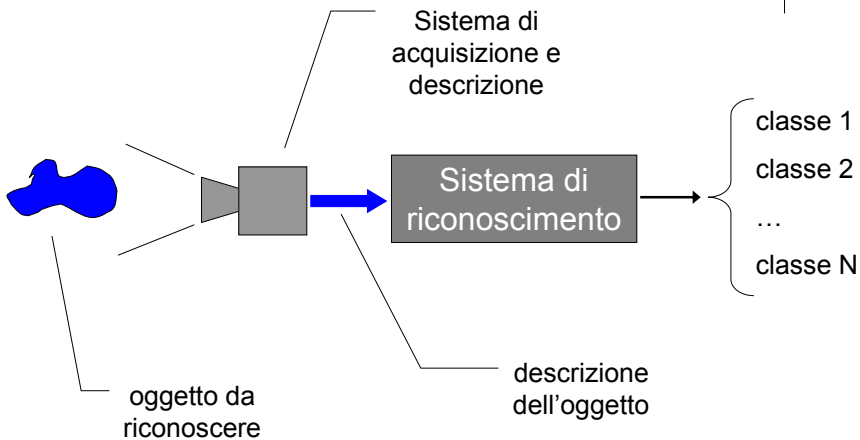
Problem Domain	Application	Input Pattern	Pattern Classes
Bioinformatics	Sequence analysis	DNA/Protein sequence	Known types of genes/ patterns
Data mining	Searching for meaningful patterns	Points in multi- dimensional space	Compact and well- separated clusters
Document classification	Internet search	Text document	Semantic categories (e.g., business, sports, etc.)
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters, words
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective / non-defective nature of product
Multimedia database retrieval	Internet search	Video clip	Video genres (e.g., action, dialogue, etc.)
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories, growth pattern of crops
Speech recognition	Telephone directory enquiry without operator assistance	Speech waveform	Spoken words

Riconoscimento automatico di oggetti (Pattern Recognition)



- Scopo: definire un sistema per riconoscere automaticamente un oggetto
- data la descrizione di un oggetto che può appartenere ad una tra N classi possibili, compito del sistema è ascrivere l'oggetto ad una classe, utilizzando una base di conoscenza precedentemente costruita

Schema ideale di un sistema di riconoscimento

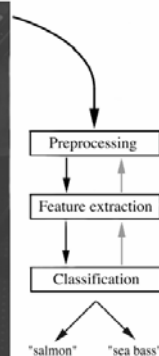


Un esempio ittico (Duda, Hart, Stork)



Problema:

distinguere il tipo di pesce (salmone o spigola) che transita su un nastro trasportatore



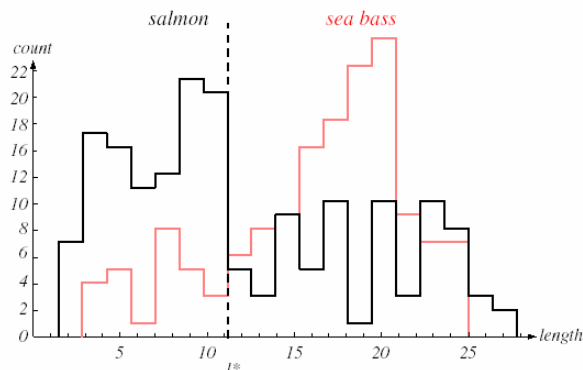


Scelta delle features

- Sulla base di informazioni ottenute sul problema si sceglie come feature iniziale la lunghezza.
- Per valutare l'efficacia di tale scelta e come questa possa essere usata è necessario considerare un insieme di campioni su cui effettuare la misura e fare le valutazioni. Si parla di *insieme di addestramento (training set)*.

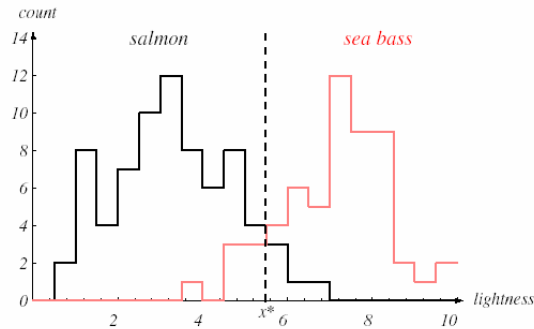


Scelta delle features



La feature scelta non è molto discriminante. C'è una certa differenza, in media, ma non tale da separare nettamente le due classi.

Scelta delle features



La nuova feature scelta permette una distinzione migliore tra le due classi. La soglia x^* è scelta in modo da minimizzare l'errore totale.

Costi degli errori



- Nel decidere la soglia si considerano due tipi di errore:
 - spigola \rightarrow salmone
 - salmone \rightarrow spigola
- La soglia è scelta ritenendo uguale il costo dei due tipi di errore. Ipotesi spesso non valida.
- Nell'esempio corrente quali sono le conseguenze dei due tipi di errore ?



Costo di classificazione

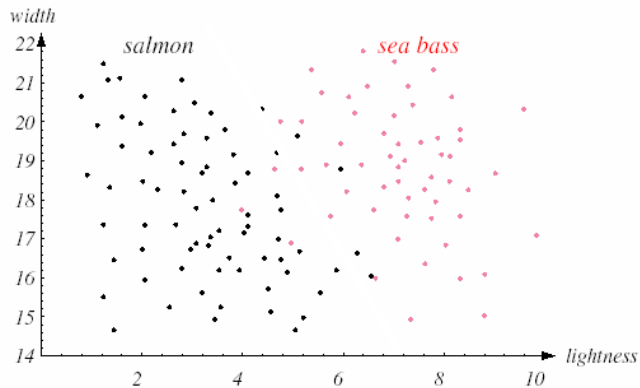
- Al risultato della decisione è quindi associato un costo complessivo (*costo di classificazione*).
- L'obiettivo è quindi quello di realizzare una regola di decisione che minimizzi tale costo (*cost-sensitive classification*).
- Questo è l'argomento centrale della *teoria della decisione (decision theory)*, un capitolo fondamentale del Pattern Recognition.



Scelta delle features

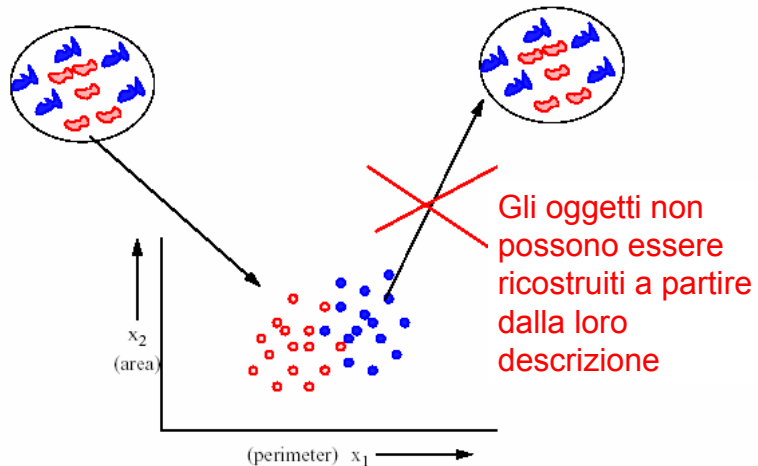
- La feature scelta, sebbene migliorativa rispetto al primo caso, potrebbe non portare ad una prestazione soddisfacente.
- Che cosa fare per migliorare ancora ? La soluzione più efficace è quella di scegliere un insieme di features da usare insieme.
- In questo caso, il singolo campione è descritto non più da un unico valore, ma da un insieme di valori.
- Formalmente ogni campione è descritto da un *vettore di features (feature vector)* a N componenti che individua un punto in uno spazio N-dimensionale (*spazio delle features o feature space*).

Spazio delle features



In questo caso si sono scelte due features. Lo spazio delle features è un piano ed ogni campione corrisponde a un feature vector $[x_1 \ x_2]^T$.

Rappresentazione e descrizione



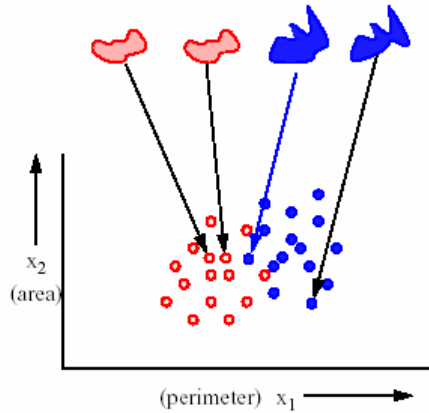
L'ipotesi di compattezza



Oggetti simili sono vicini nello spazio delle features

Achtung!

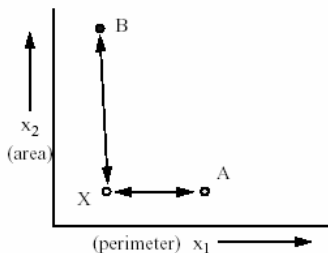
Oggetti diversi non sono necessariamente distanti



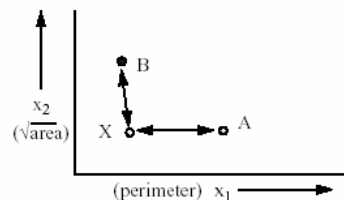
Normalizzazione delle features



In caso di features omogenee è opportuna una normalizzazione dimensionale



Prima della normalizzazione:
 $d(X,B) > d(X,A)$

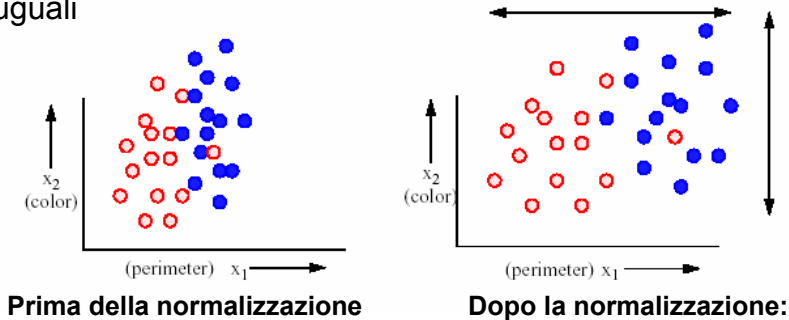


Dopo la normalizzazione:
 $d(X,B) < d(X,A)$

Normalizzazione delle features



Anche se le features non sono omogenee potrebbe essere opportuna una normalizzazione che renda le varianze uguali



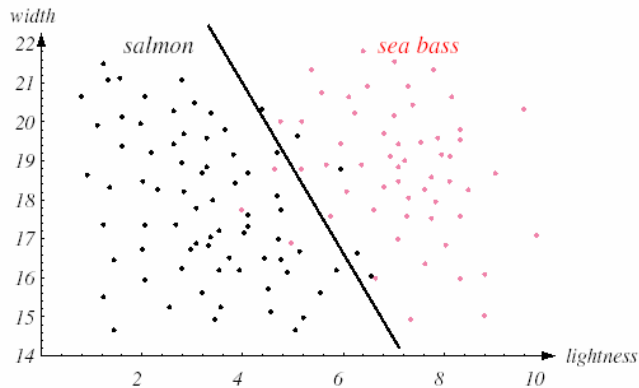
$$x' = \frac{x}{\sqrt{\text{var } x}} \quad y' = \frac{y}{\sqrt{\text{var } y}}$$

Regioni di decisione



- Il problema ora è dividere lo spazio delle features in regioni, ognuna delle quali sia ascrivibile ad una delle classi note.
- Si identificano così delle *regioni di decisione* (*decision regions*), separate da una frontiera (*decision boundary*).
- In questo modo è possibile decidere a quale classe assegnare il campione sulla base della posizione del punto nel feature space.

Regioni di decisione



La scelta più immediata è quella di una frontiera semplice, lineare. Gli errori complessivi sono minori rispetto al caso di una sola feature, ma sono comunque presenti.

Si può dare di più ?



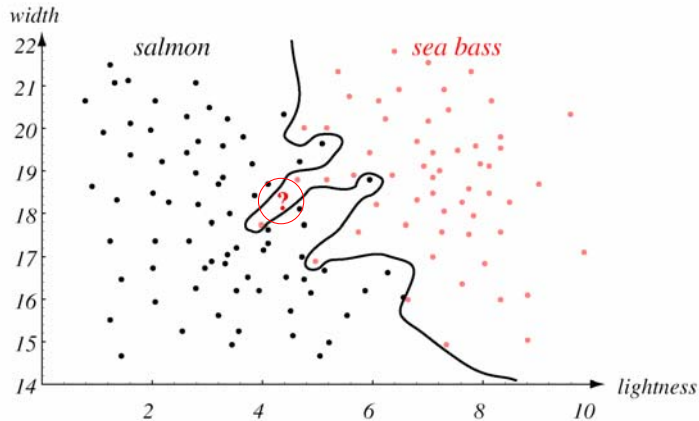
- Le regioni definite in base alla frontiera lineare implicano ancora degli errori.
- Sarebbe possibile eliminare del tutto gli errori con una frontiera meno semplice ?
- Ricordiamo che la frontiera di decisione è generata dal sistema di classificazione; quindi

frontiera meno semplice



classificatore più complesso

Si può dare di più ?



Una frontiera di decisione più complessa che annulla gli errori sul training set. Come sarà classificato il nuovo campione ?

Il problema della generalizzazione



- Con una frontiera complessa è possibile annullare l'errore sul training set.
- Il problema è che in questo modo non si garantisce una buona prestazione del sistema sui campioni che bisognerà classificare in fase operativa.
- Questo aspetto (*generalizzazione*) è fondamentale nella progettazione dell'intero sistema.

Il problema della generalizzazione



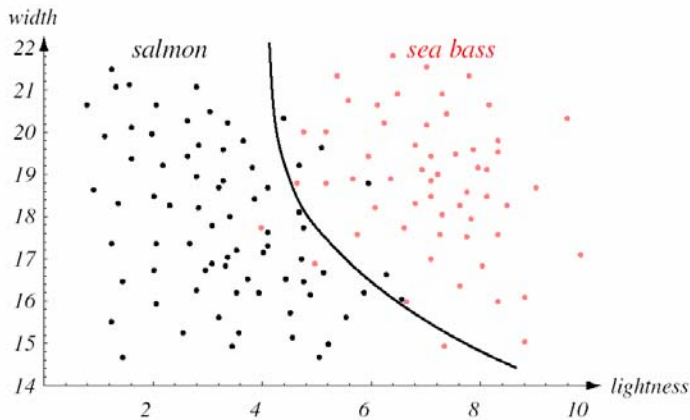
- E' improbabile che un classificatore estremamente complesso garantisca buone capacità di generalizzazione in quanto costruito strettamente sulle caratteristiche dei campioni del particolare training set (e del particolare rumore che si portano dietro).
- Un classificatore efficiente dovrebbe invece essere costruito su caratteristiche più generali che siano valide anche per campioni non appartenenti al training set.

Il problema della generalizzazione



- Si impone quindi di stabilire un compromesso tra:
 - prestazioni del classificatore sul training set
 - capacità di generalizzazione del classificatore (legata alla sua "semplicità")
- Di conseguenza, è preferibile tollerare qualche errore sul training set se questo porta ad una migliore generalizzazione del classificatore.

Il problema della generalizzazione



Una frontiera di decisione più complessa della frontiera lineare. Sebbene gli errori sul training set siano ancora presenti, il classificatore sembra garantire una buona capacità di generalizzazione.

Come si costruisce il classificatore ?



- Appurate le caratteristiche che dovrebbe esibire un classificatore efficiente, si pone il problema della costruzione del classificatore.
- **E' possibile una soluzione algoritmica ?**
E' cioè possibile definire un algoritmo per classificare caratteri, parlato, immagini,... ?
- Dopo 40 anni di sforzi in questo senso la risposta è chiaramente negativa.
- Perché ?
- Ci sono alternative ?

L'apprendimento



- L'unica alternativa percorribile è quella di apprendere a risolvere i problemi a partire da esempi (*learning by examples*).
- *Apprendimento (learning)*: ogni metodo che, nella costruzione di un classificatore, combina informazioni empiriche provenienti dall'ambiente e conoscenza a priori del contesto del problema.
- Le informazioni empiriche sono di solito nella forma vista di campioni di esempio (training set).
- Conoscenza a priori: invarianti, correlazioni, ...

Paradigmi di apprendimento



Esistono diversi paradigmi di apprendimento:

- *Apprendimento supervisionato (supervised)*:
per ogni campione del training set è provvista la classe di appartenenza. Obiettivo dell'apprendimento è quello di minimizzare gli errori (o il costo di classificazione).
- *Apprendimento non supervisionato (unsupervised)*:
non sono fornite esplicite informazioni sulla classe dei campioni del training set. Obiettivo dell'apprendimento è quello di formare dei *raggruppamenti (clusters)* dei campioni generalmente sulla base di una distanza. Spesso è definito dall'esterno il numero dei clusters da produrre. Questo paradigma viene definito anche *clustering*.