SVILUPPO INFORMATICO
GRUPPO TECNOINVESTIMENTI

Introduzione all'Intelligenza Artificiale

Concetti base ed esempi di applicazione

Prof. Francesco Trovò

12/05/2018

unibg
UNIVERSITÀ DEGLI STUDI DI BERGAMO
POLITECNICO
MILANO 1863

Chi sono

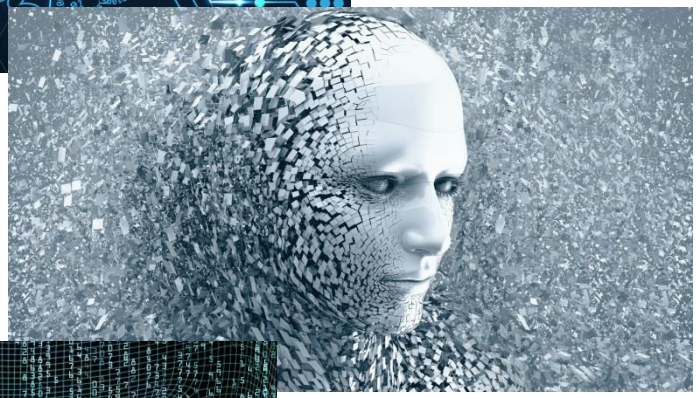
Francesco Trovò

- Laureato @Polimi 2011
- Laureato @Aalto 2011
- Dottorato @Polimi 2015

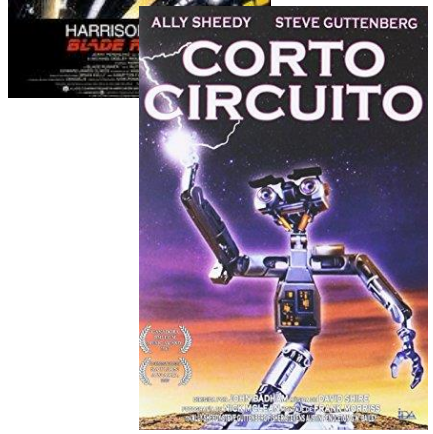
Assegnista di Ricerca presso il Dipartimento di Elettronica, Informazioni e Bioingegneria del Politecnico di Milano

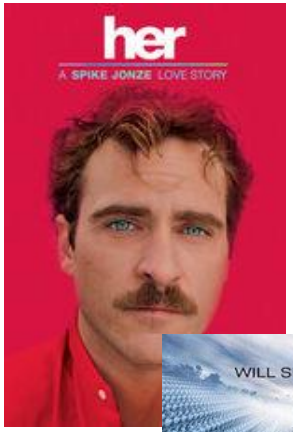
Titolare del corso di Intelligenza Artificiale all'Università degli Studi di Bergamo

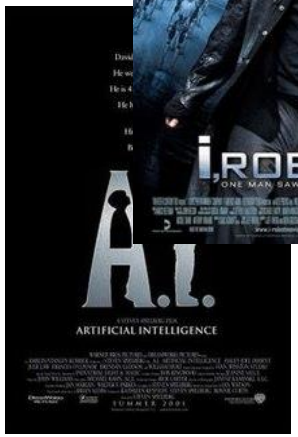
Come viene percepita l’Intelligenza Artificiale



L’Intelligenza Artificiale di Hollywood







Outline

- Definizione di intelligenza
- Applicazioni dell’intelligenza artificiale nel mondo reale

- Machine Learning – Supervised e Unsupervised Learning
- Esempi pratici

Intelligenza Artificiale

Definizioni di Intelligenza Artificiale

Domande:

Cos'è l'intelligenza? Cos'è il pensiero?

Come possiamo costruire entità **intelligenti**?

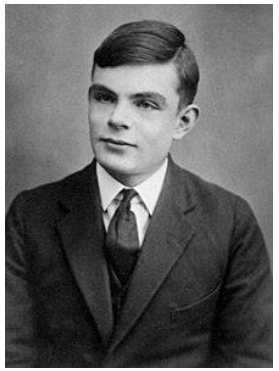
Defiizione di intelligenza:

- **Agire umanamente**
- Pensare umanamente
- Pensare razionalmente
- **Agire razionalmente**

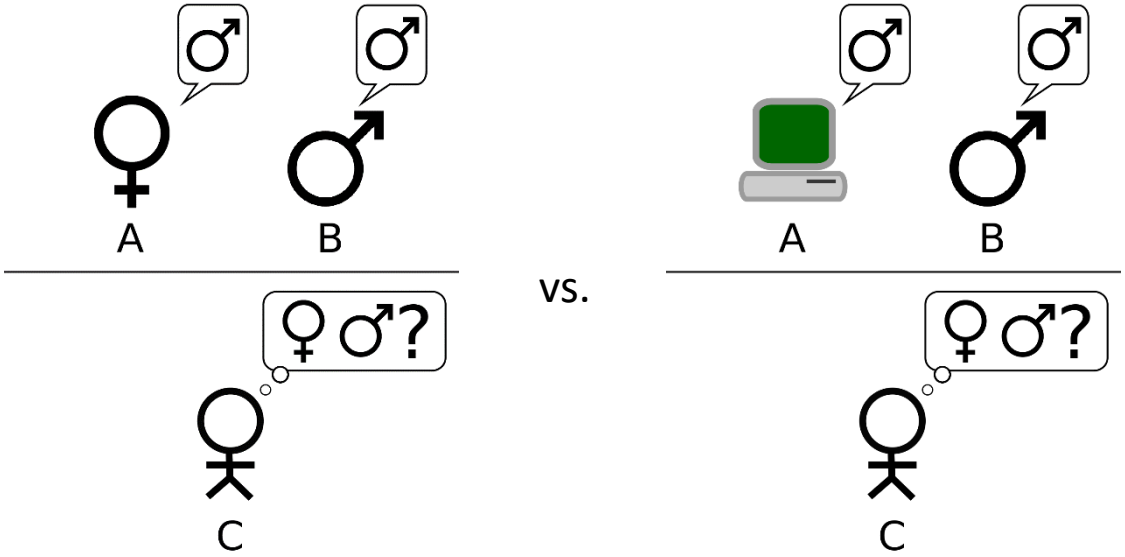
Agire umanamente

Alan Turing (1912 - 1954)

- Formalizzazione algoritmo, macchina di turing
- Primo crittoanalista (macchina Enigma)
- Padre dell'informatica e dell'IA



Test di Turing:

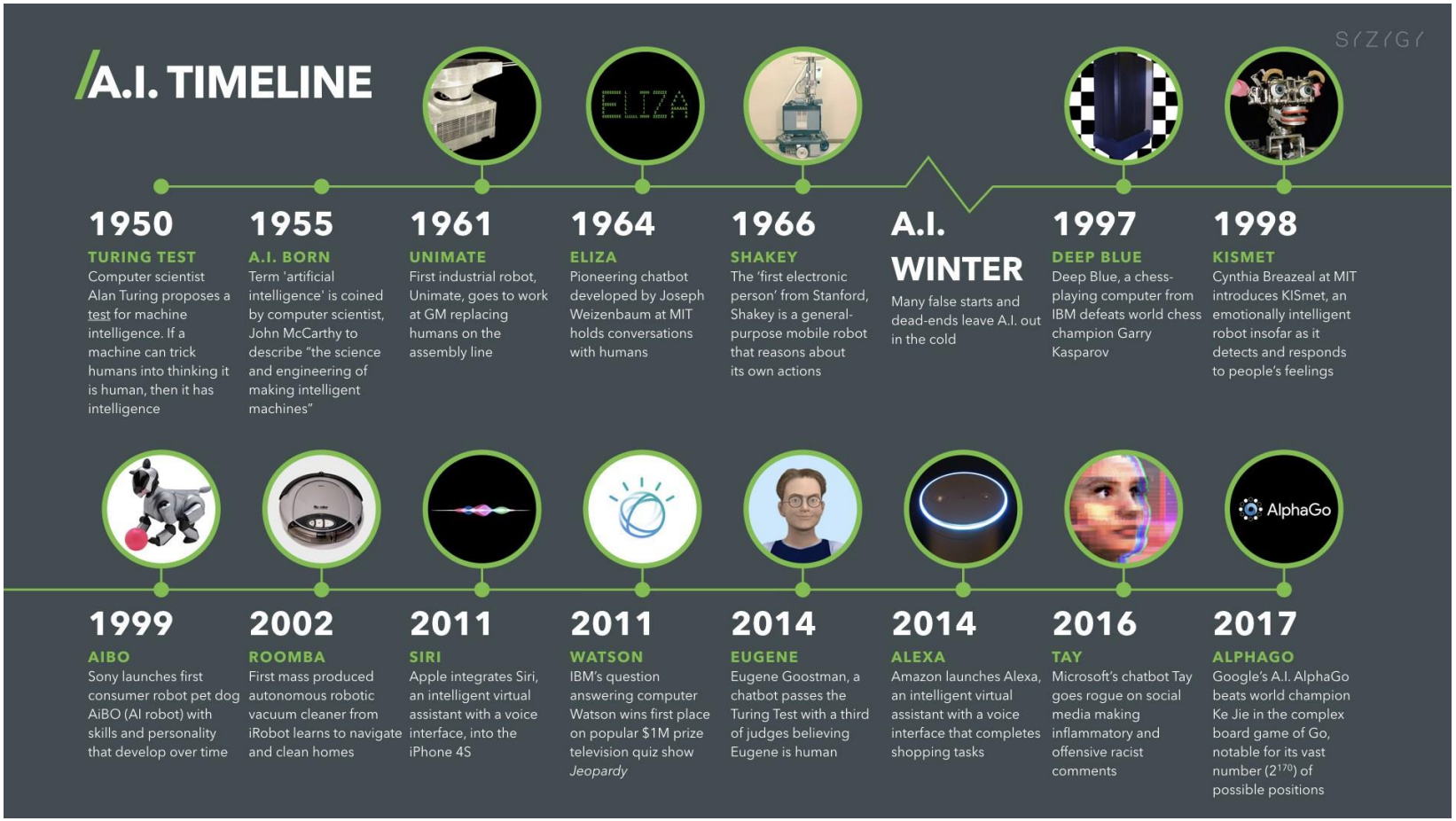


Agire razionalmente

Non sempre dobbiamo imitare le azioni degli umani

Contest: far camminare un agente bipede

I passi importanti per lo sviluppo dell'IA



Il nuovo paradgma sono i dati

Negli anni il volume dei **dati** generati è cresciuto a ritmi impressionanti

Prima cercavamo di formalizzare un problema astraendone le leggi caratterizzanti

Oggi le tecniche di apprendimento si sostituiscono al processo manuale di categorizzazione della conoscenza

Compentenze base richieste:

- Gestione basi di dati
- Statistica
- Programmazione

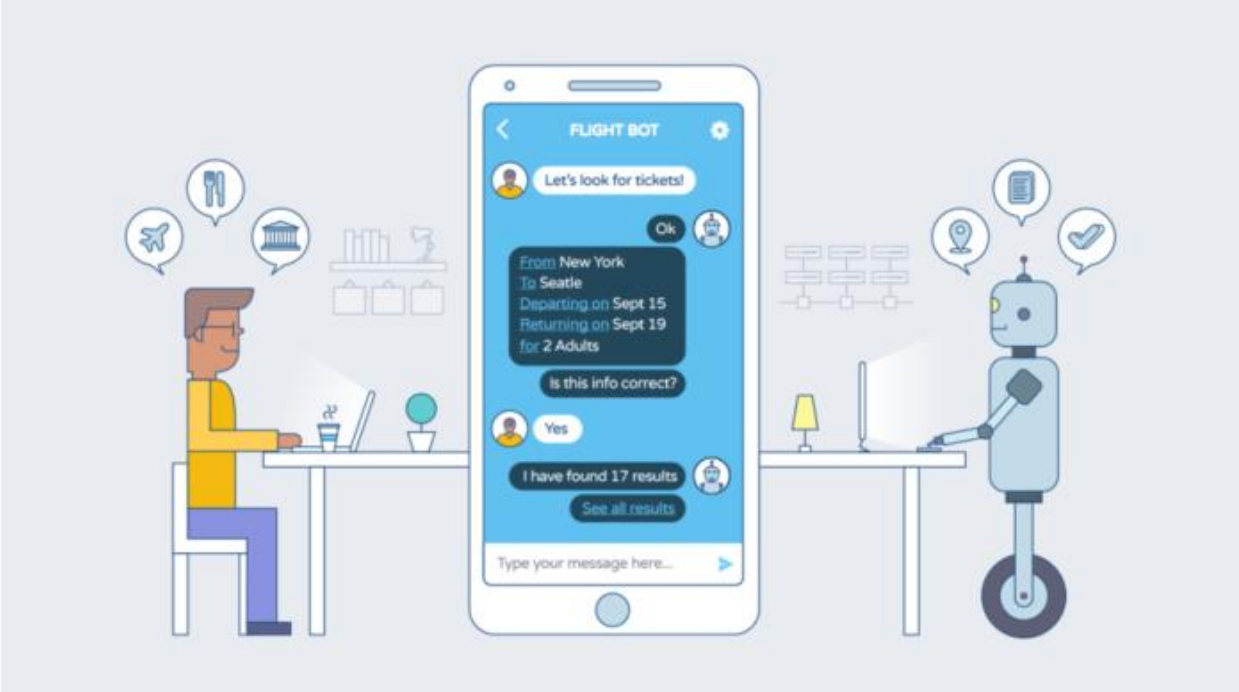
Machine Learning

Esempi di problemi risolti da metodi di IA

Veicoli autonomi – DARPA urban challenge, TESLA

Esempi di problemi risolti da metodi di IA

Chatbot – call center automatizzati



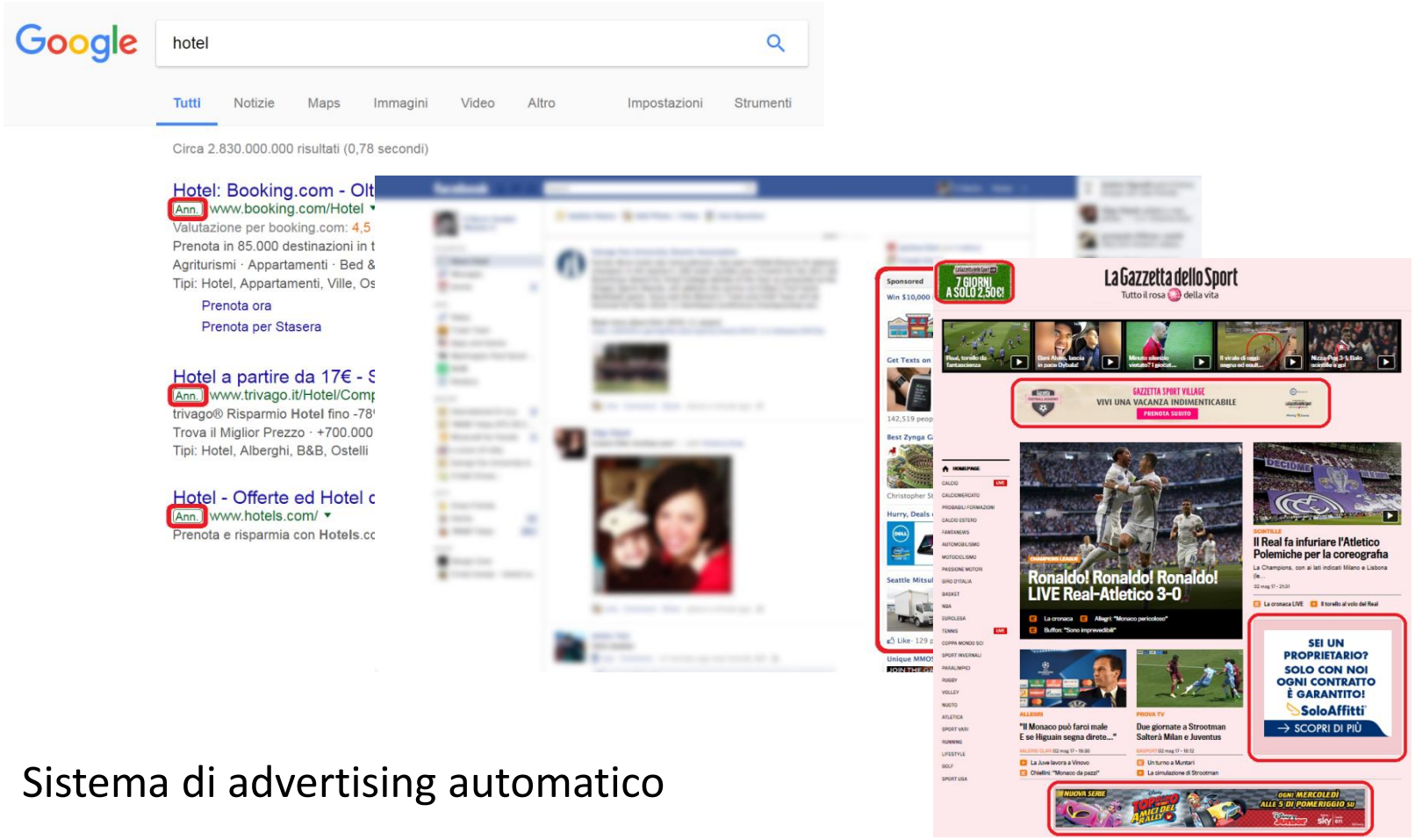
Esempi di problemi risolti da metodi di IA

Giochi – scacchi DEEPBLUE, go ALPHA GO, poker LIBRATUS

Esempi di problemi risolti da metodi di IA

Kidney exchange problem

Applicazioni di tecniche di AI: esperienze locali

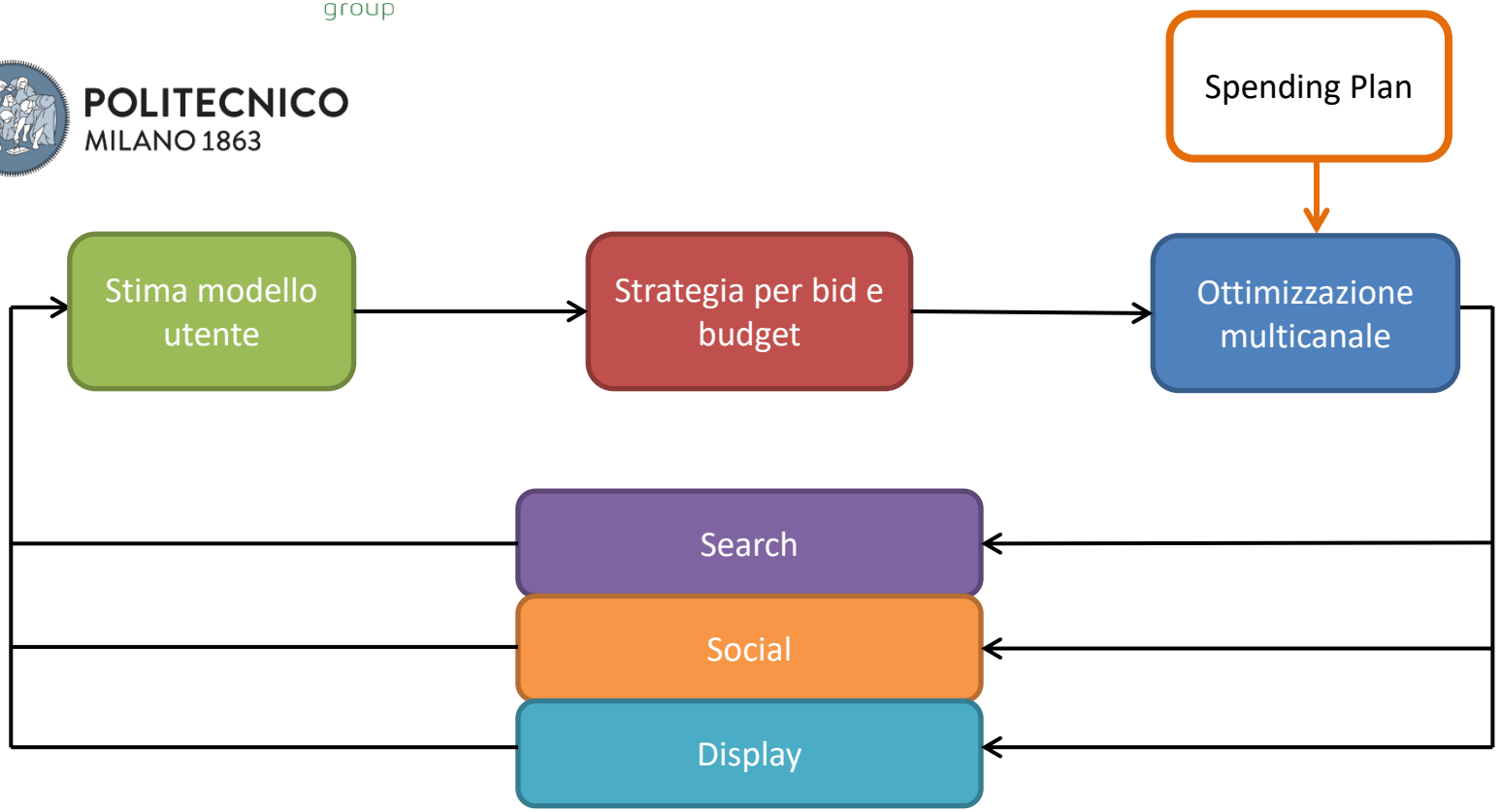


Sistema di advertising automatico

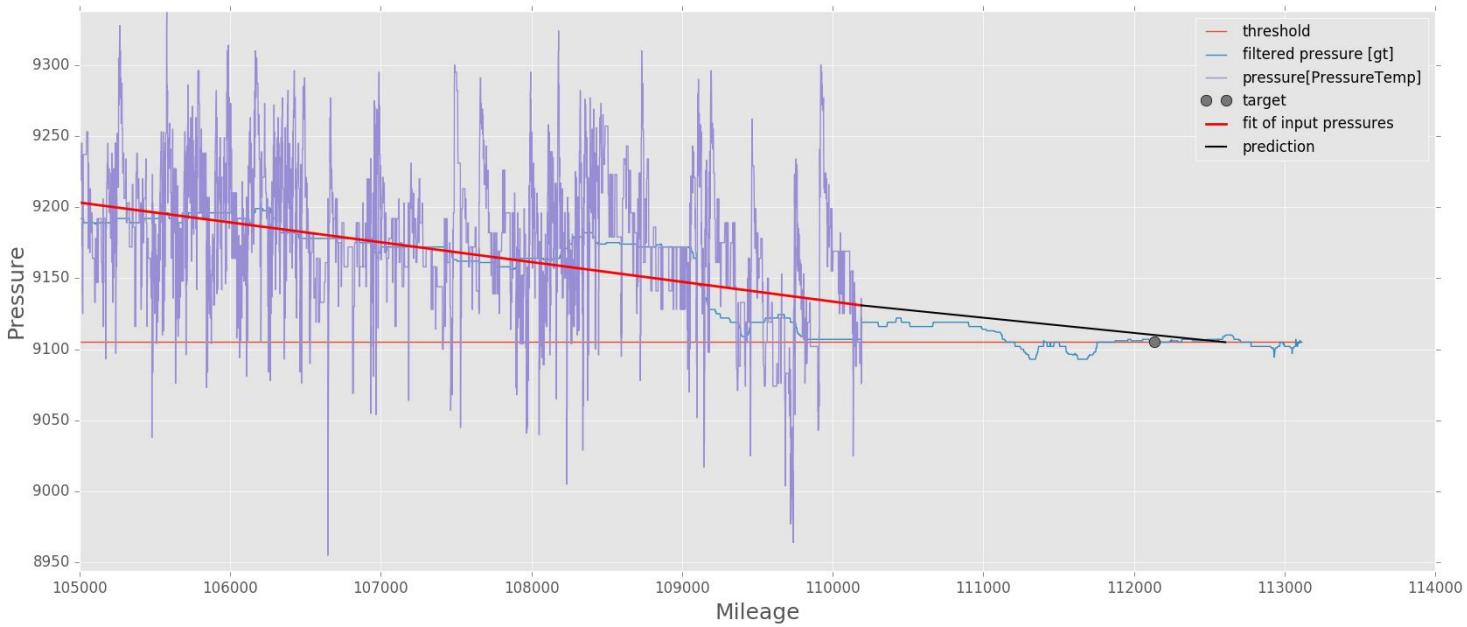
Applicazioni di tecniche di AI: esperienze locali

MEDIAMATIC mmm group

POLITECNICO MILANO 1863



Applicazioni di tecniche di AI: esperienze locali



PIRELLI™

POLITECNICO
MILANO 1863

Machine Learning

Definizione di Machine Learning

Mitchell (1997):

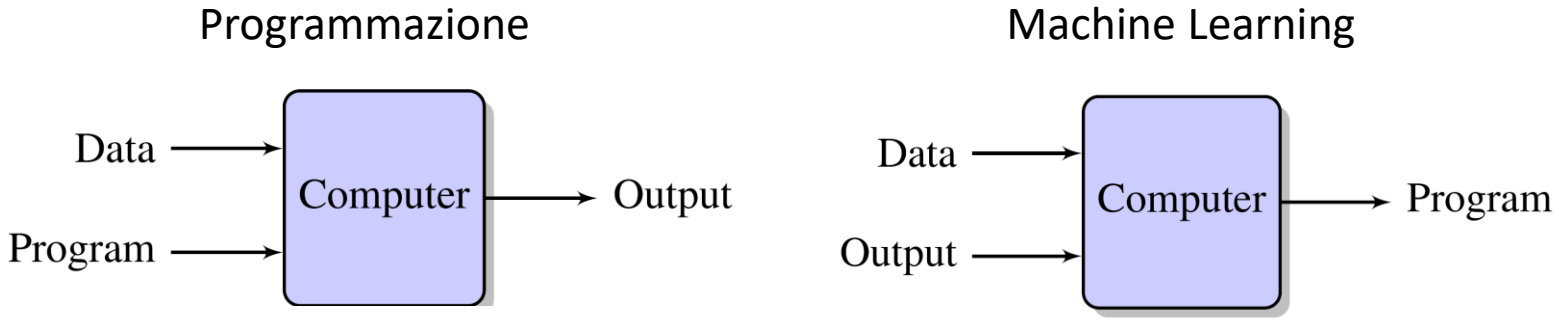
*"A computer program is said to learn from **experience E** with respect to some **class of tasks T** and **performance measure P** if its performance at tasks in T, as measured by P, improves with experience E".*

Prima di proporre una soluzione dobbiamo definire

- Experience: dati disponibili
- Task: problema definito
- Performance: metodo di valutazione

Utilità delle tecniche di ML

- Principale task del ML: predire cosa accadrà a ad un fenomeno nel futuro



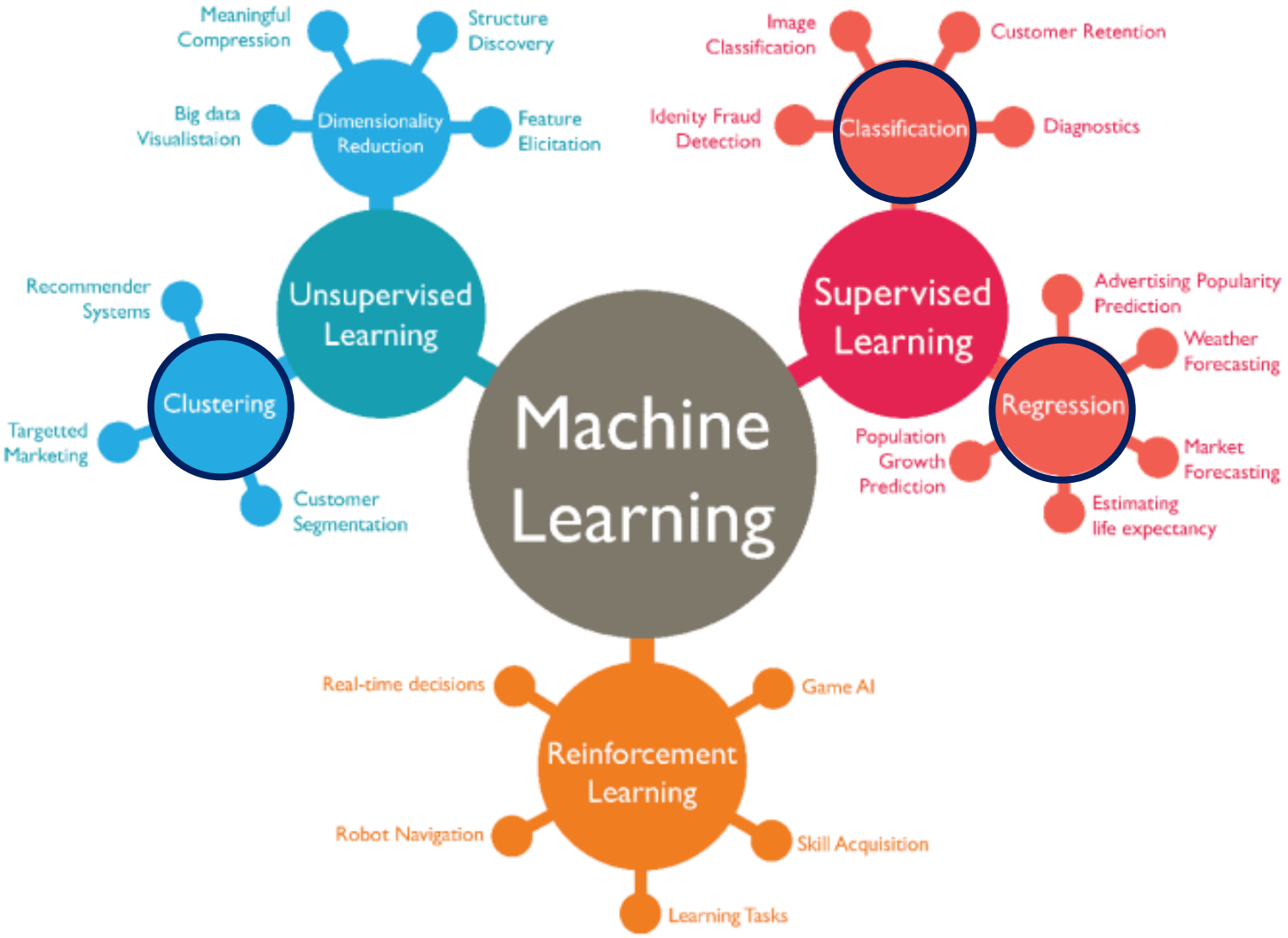
- Le tecniche di ML estraggono tali modelli dai dati disponibili e li utilizzano per fare predizioni su nuovi dati

Hanno detto riguardo al ML

- “A breakthrough in machine learning would be worth ten Microsofts”. (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet”. (Tony Tether, Director, DARPA)
- “Machine learning is the hot new things”. (John Hennessy, President, Stanford)
- ”Machine learning today is one of the hottest aspects of computer science“. (Steve Ballmer, CEO, Microsoft)

- “All models are wrong, but some models are useful”. (George Box, Statistician, Princeton)
- “We are drowning in information and starving for knowledge”. (John Naisbitt, Writer)

Topologia dei problemi risolti tramite ML



Quando usare delle tecniche di ML

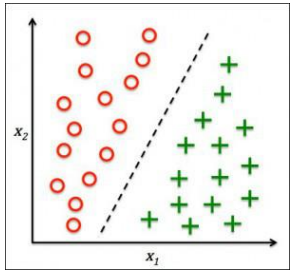
1. Ho dati storici relativi al problema
2. Inoltre:
   i. Non esistono degli esperti di dominio
   ii. É complesso formalizzare il problema
   iii. La funzione da approssimare cambia frequentemente
   iv. Ogni utente richiede la stima di una differente funzione

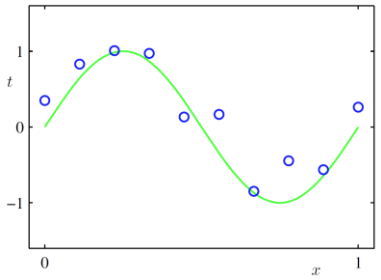
Supervised Learning

Dati: un training set $D = \{(\boldsymbol{x}_i, t_i)\}_{i=1}^{N}$ generato da $f(\boldsymbol{x}) = t$

Obbiettivo: apprendere da D un'approssimazione $y(\cdot)$ di $f(\cdot)$

- I vettori $\boldsymbol{x}_i$ sono anche detti input, predittori, attributi, covariate
- I valori t_i sono anchedetti target, risposte, etichette (labels)
  - se il target è categorico → **classificazione**
  - se il target è ordinale → **regressione**





Esempi di Supervised Learning

Iris dataset

Setosa

Versicolor

Virginica

Dati relativi a differenti fiori

- Dimensioni petalo
- Dimensioni sepalo
- Specie

Esempio di training set (Classificazione)

Dataset D

	Row ID	D sepal_length	D sepal_width	D petal_length	D petal_width	S species	
x_{45}	Row45	4.8	3	1.4	0.3	setosa	t_{45}
	Row46	5.1	3.8	1.6	0.2	setosa	
	Row47	4.6	3.2	1.4	0.2	setosa	
	Row48	5.3	3.7	1.5	0.2	setosa	
	Row49	5	3.3	1.4	0.2	setosa	
	Row50	7	3.2	4.7	1.4	versicolor	
⋮	Row51	6.4	3.2	4.5	1.5	versicolor	⋮
	Row52	6.9	3.1	4.9	1.5	versicolor	
	Row53	5.5	2.3	4	1.3	versicolor	
	Row54	6.5	2.8	4.6	1.5	versicolor	
	Row55	5.7	2.8	4.5	1.3	versicolor	
	Row56	6.3	3.3	4.7	1.6	versicolor	
	Row57	4.9	2.4	3.3	1	versicolor	
	Row58	6.6	2.9	4.6	1.3	versicolor	
	Row59	5.2	2.7	3.9	1.4	versicolor	
x_{60}	Row60	5	2	3.5	1	versicolor	t_{60}

Matrice degli input X

Vettore degli output $\boldsymbol{t}$

Esempio di training set (Regressione)

Dataset D

	Row ID	sepal_length	sepal_width	petal_length	petal_width	
x_{45}	Row45	4.8	3	1.4	0.3	t_{45}
	Row46	5.1	3.8	1.6	0.2	
	Row47	4.6	3.2	1.4	0.2	
	Row48	5.3	3.7	1.5	0.2	
	Row49	5	3.3	1.4	0.2	
	Row50	7	3.2	4.7	1.4	
	Row51	6.4	3.2	4.5	1.5	
⋮	Row52	6.9	3.1	4.9	1.5	⋮
	Row53	5.5	2.3	4	1.3	
	Row54	6.5	2.8	4.6	1.5	
	Row55	5.7	2.8	4.5	1.3	
	Row56	6.3	3.3	4.7	1.6	
	Row57	4.9	2.4	3.3	1	
	Row58	6.6	2.9	4.6	1.3	
	Row59	5.2	2.7	3.9	1.4	
x_{60}	Row60	5	2	3.5	1	t_{60}

Matrice degli input X

Vettore degli output $\boldsymbol{t}$

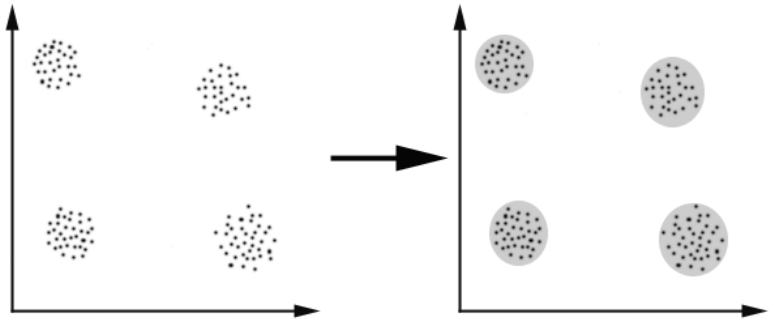
Unsupervised Learning

Dati: un training set composto da soli input $D = \{\boldsymbol{x}_i\}_{i=1}^{N}$

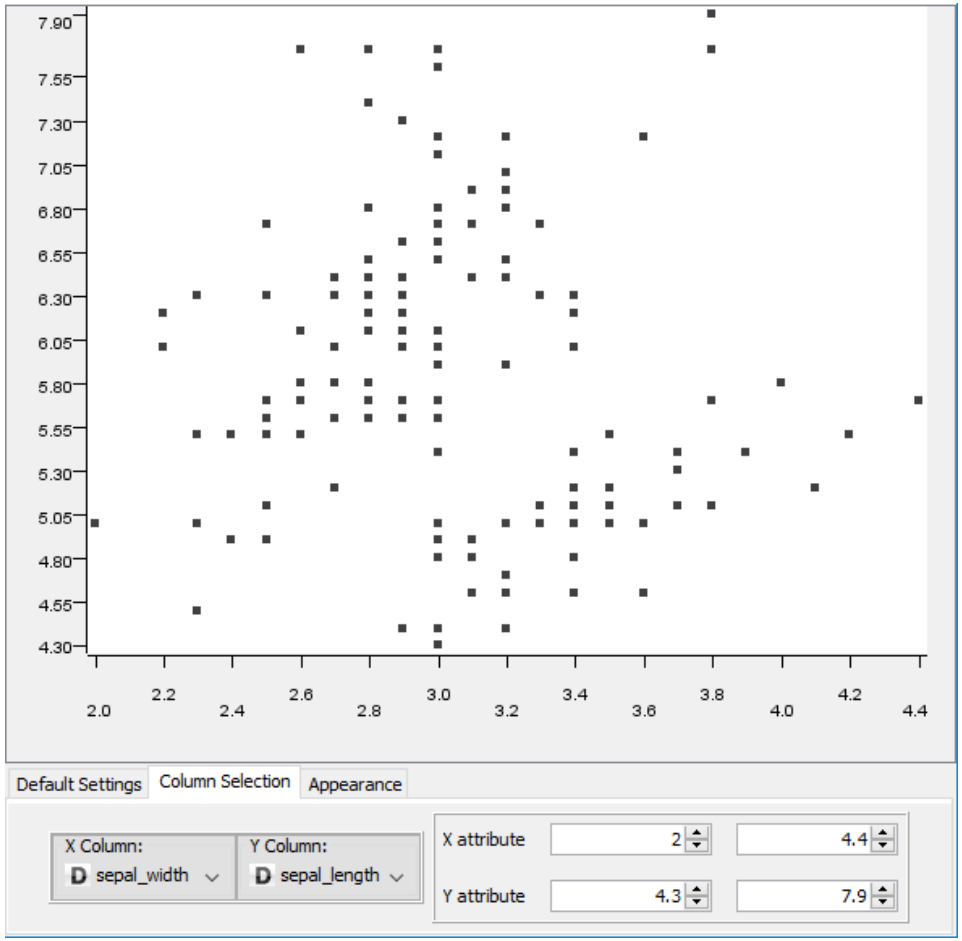
Obbiettivo: un modo efficace per descrivere i dati in maniera concisa

Tecniche differenti:

- **Clustering**
- Dimensionality reduction
- Data visualization



Esempio di training set (Clustering)

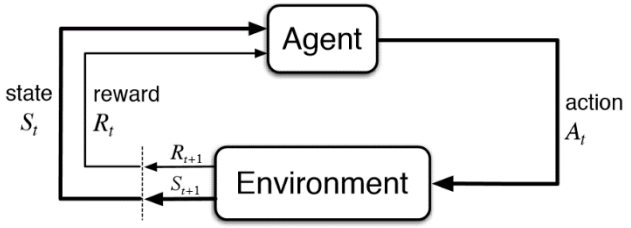


Reinforcement Learning

Dati: un training set di episodi $D = \{(\boldsymbol{x}, a, \boldsymbol{x}', r)\}$

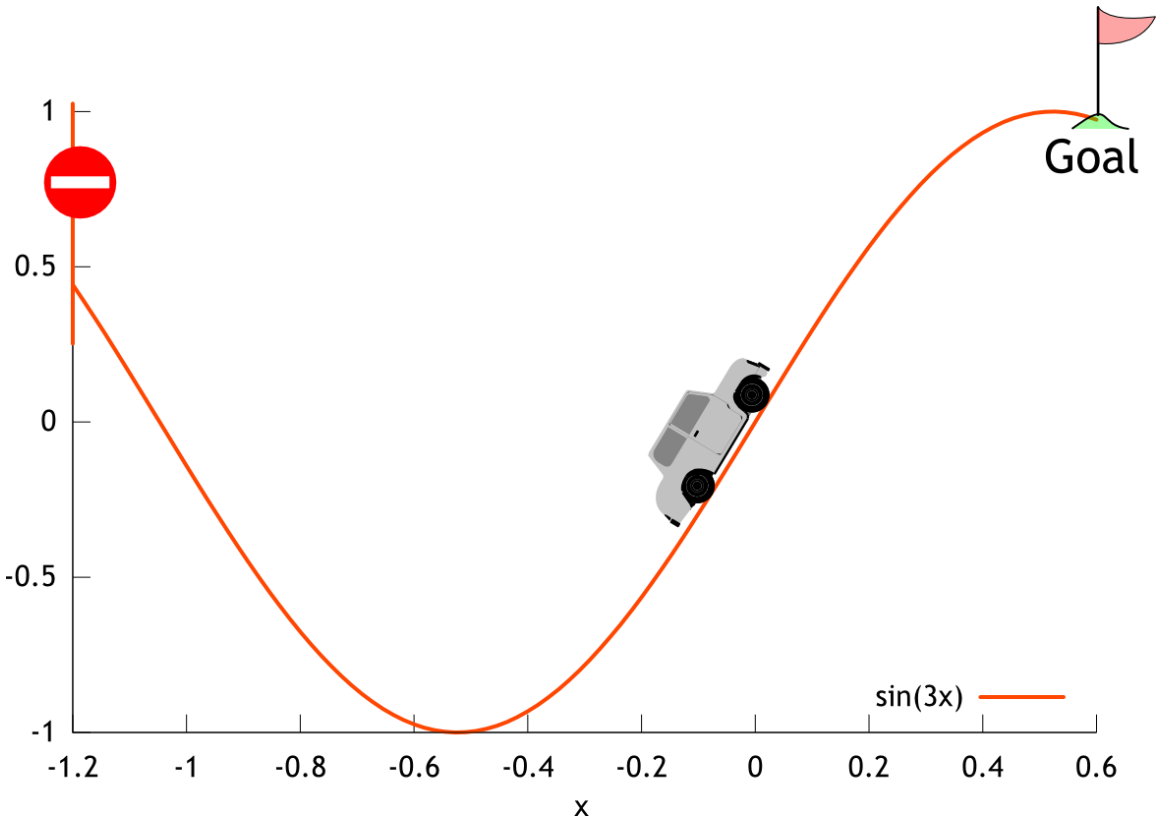
Obbiettivo: una politica che massimizzi il reward atteso collezionato nel tempo

- Markov Decision Process (MDP)
- Partially Observable MDP
- Stochastic Games



Esempi di Reinforcement Learning

Mountain car



Esempi di Reinforcement Learning

Atari Games

Ruolo dell'esperto di ML

No Free Lunch Theorem
«Non esiste un algoritmo di ML che sia in grado di avere una buona performance su ogni problema»

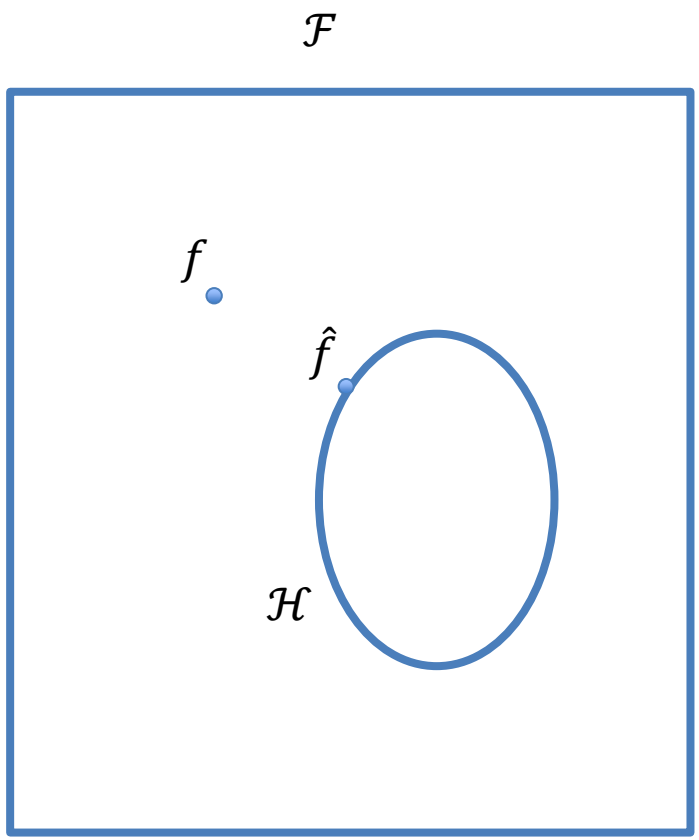
- Esistono quindi molte tecniche ognuna adatta ad un particolare sottoinsieme di problemi
- Sta nell'abilità di chi sceglie la tecnica la selezione di quella adatta al problema specifico

Elementi fondamentali del learning

Vorremmo approssimare la funzione f appartenente ad uno spazio $\mathcal{F}$ utilizzando i dati appartenenti al training set D

Dobbiamo scegliere

- Un hypothesis space $\mathcal{H}$
- Una loss $l(\hat{f})$
- Un metodo di ottimizzazione



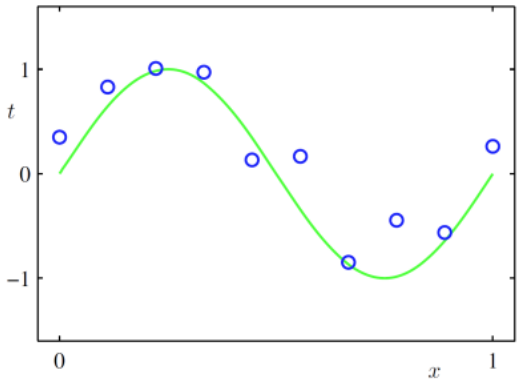
Supervised Learning - Regressione

Regressione

Vogliamo apprendere una relazione tra un input $\boldsymbol{x}$ e un output (target) t che assumiamo essere continuo ed ordinato

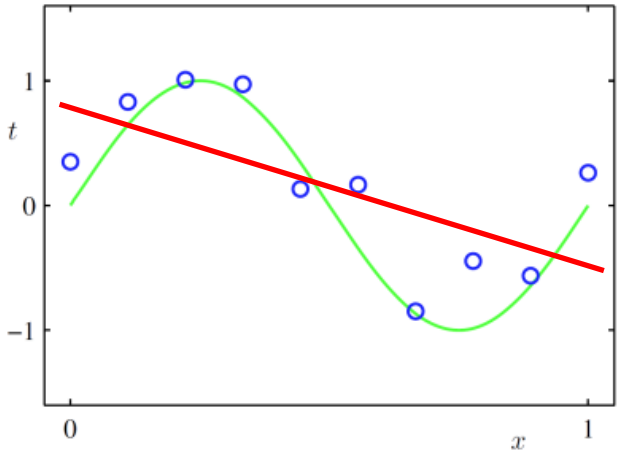
Esempi:

- Predizione dell'andamento di un'azione
- Predizione dell'età di un utente
- Predire l'effetto di un'azione di un robot
- Predire il valore di una casa



Regressione Lineare

Tutti i processi reali possono essere approssimati tramite funzioni lineari



Pro:

- Ha una soluzione analitica
- È utile per interpretare il fenomeno
- Può essere estesa per considerare relazioni nonlineari

Contro:

- Non tutti i processi sono lineari

Hypothesis space: funzioni lineari

Approssimimamo la funzione di input/output $f(\cdot)$ con una funzione lineare rispetto ad un vettore di parametri $\boldsymbol{w}$:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{D-1} w_j x_j = \mathbf{w}^T \mathbf{x}$$

Offset

Parametro

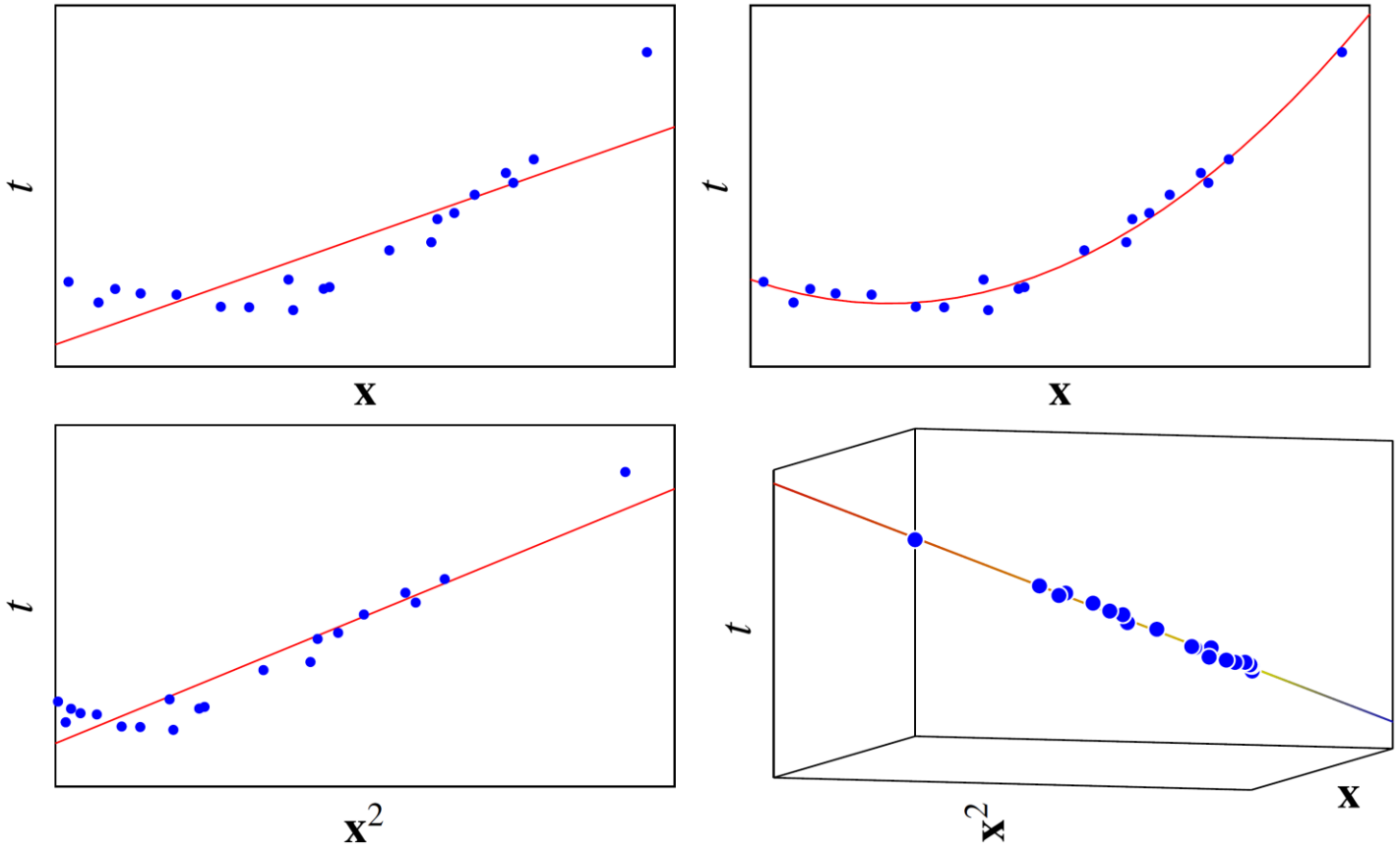
Possiamo anche utilizzare delle funzioni di base per rendere il modello più espressivo:

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

$$\boldsymbol{\phi}(\mathbf{x}) = (1, \phi_1(\mathbf{x}), \ldots, \phi_{M-1}(\mathbf{x}))^T$$

Esempio di base quadratica

$$\boldsymbol{\phi}(x) = (1, x, x^2)^T$$



Loss function: errore quadratico

Come possibile funzione di loss possiamo considerare l'errore medio rispetto alla distribuzione degli input e degli output:

$$\mathbb{E}[L] = \int \int L(t, y(\mathbf{x}))p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t$$

Una scelta usuale è la loss quadratica ($q = 2$)

$$\mathbb{E}[L] = \int \int (t - y(\mathbf{x}))^q p(\mathbf{x}, t)\mathrm{d}\mathbf{x}\mathrm{d}t$$

???

Purtroppo non conosciamo la forma della distribuzione congiunta delle coppie input/output

Loss sul training Set

Quello che posso fare è minimizzare la loss sui dati che ho a disposizione

$$L(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(y(x_n, \mathbf{w}) - t_n\right)^2$$

Infatti, essi sono sample della distribuzione congiunta $p(x, y)$

É anche (metà del) la somma dei residui, ovvero di quanto errore faccio sulle predizioni

$$RSS(\mathbf{w}) = \|\boldsymbol{\epsilon}\|_2^2 = \sum_{n=1}^{N}\epsilon_i^2$$

Differenti soluzioni per l'ottimizzazione

Per minimizzare la loss possiamo utilizzare differenti metodi

- **Ordinary least square**
- Stochastic gradient descent
- Maximum likelihood estimation
- Bayesian linear regression

Ottimizzazione: OLS

Voglio minimizzare la somma dei residui

$$L(\mathbf{w}) = \frac{1}{2} RSS(\mathbf{w}) = \frac{1}{2} (\mathbf{t} - \mathbf{\Phi w})^T (\mathbf{t} - \mathbf{\Phi w})$$

Controllo che esista un punto di minimo

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = -\mathbf{\Phi}^T (\mathbf{t} - \mathbf{\Phi w}) ; \quad \frac{\partial^2 L(\mathbf{w})}{\partial \mathbf{w} \partial \mathbf{w}^T} = \mathbf{\Phi}^T \mathbf{\Phi}$$

Il parametro ottimale è:

$$\hat{\mathbf{w}}_{OLS} = \left(\mathbf{\Phi}^T \mathbf{\Phi} \right)^{-1} \mathbf{\Phi}^T \mathbf{t}$$

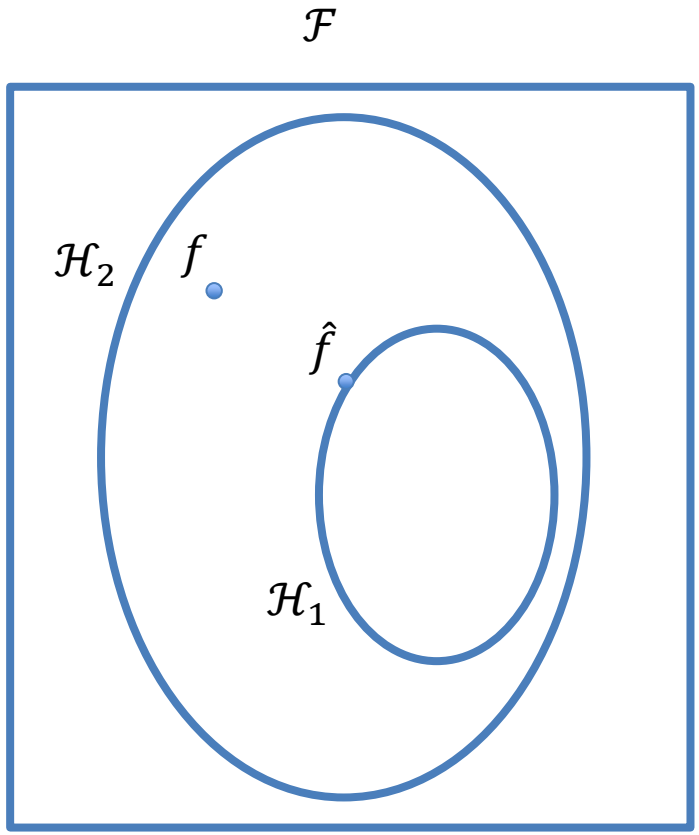
Sessione pratica (1)

Problema: voglio predire la larghezza dei petali dei fiori usando la loro lunghezza

Secelta del modello

Idea: scegliendo un hypothesis space più grande arriveremmo ad approssimare la funzione vera

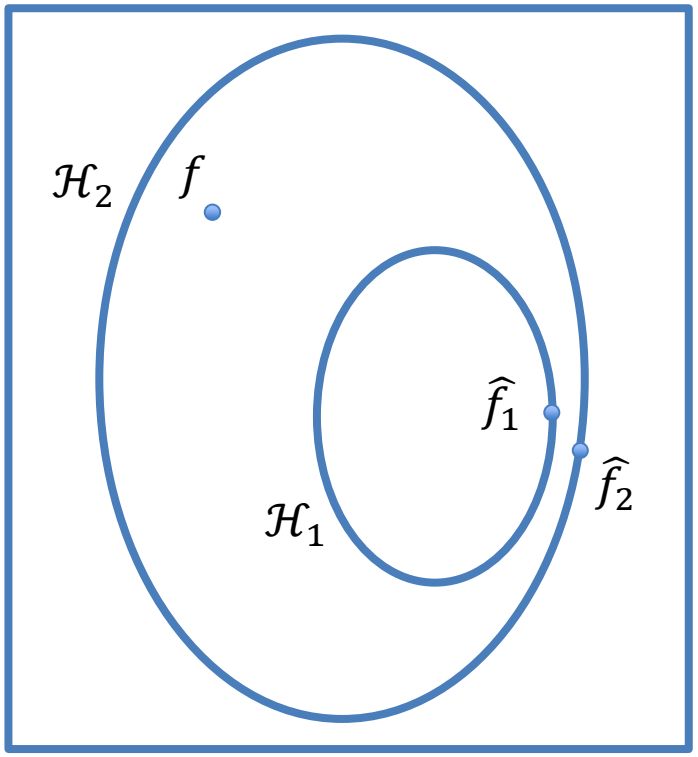
Questo accade in un mondo ideale, ovvero se avessimo un numero di dati infinito



Problemi

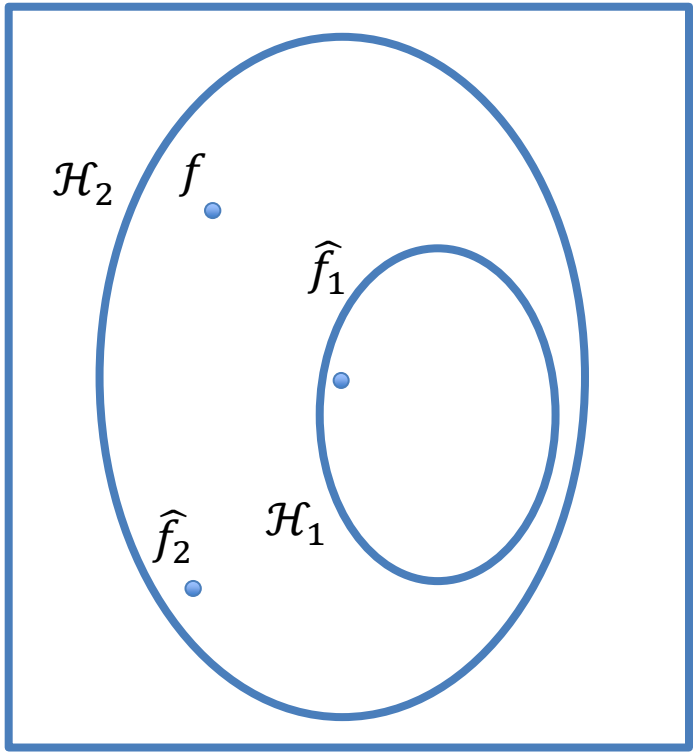
Scegliamo una funzione di loss scorretta

$\mathcal{F}$

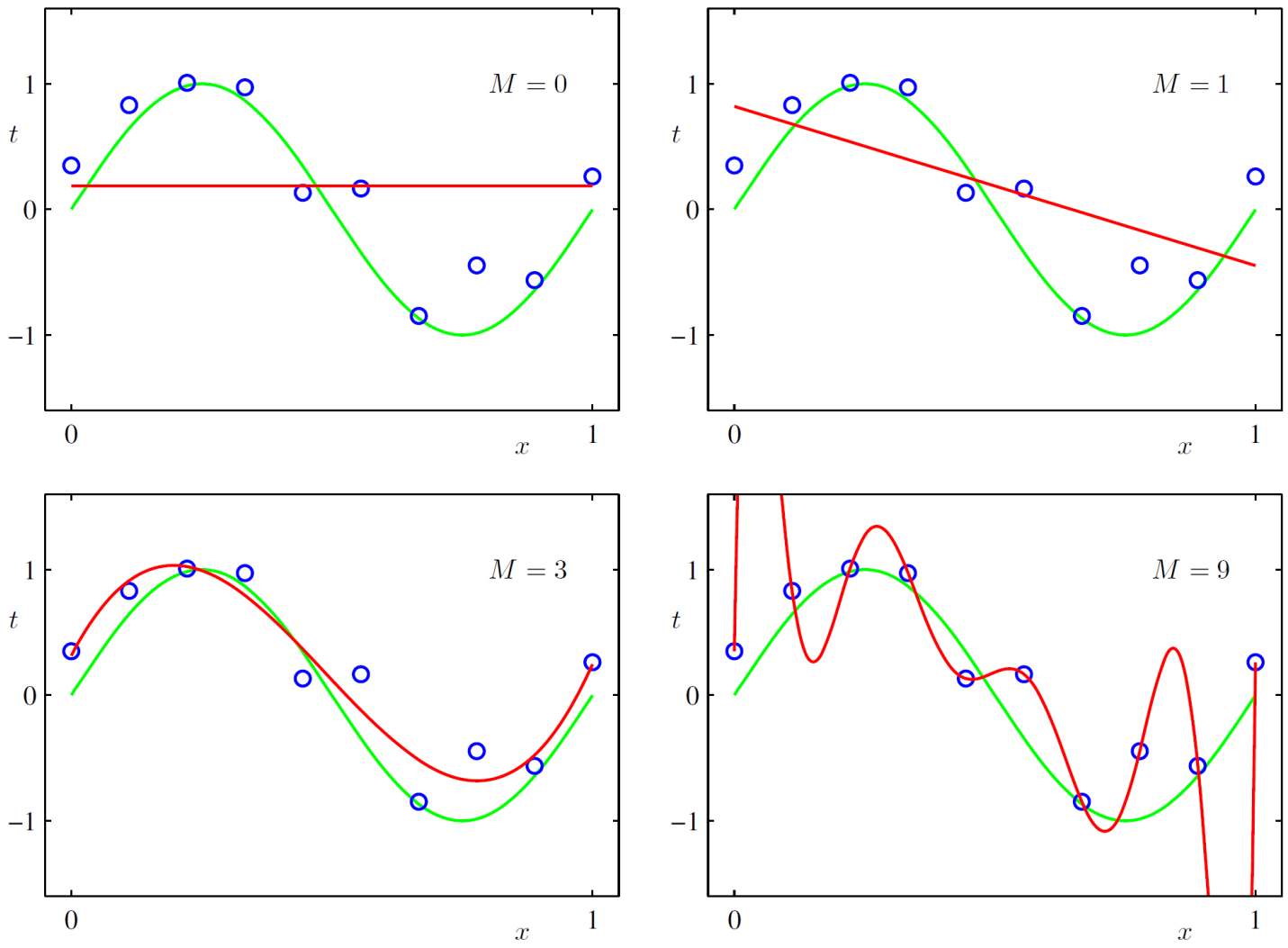


Il metodo di ottimizzazione è approssimato

$\mathcal{F}$



Modelli troppo semplici o troppo complessi



Underfitting e overfitting

Possiamo sbagliare scegliendo:

- Modelli troppo semplici → **underfitting**
- Modelli troppo complessi → **overfitting**

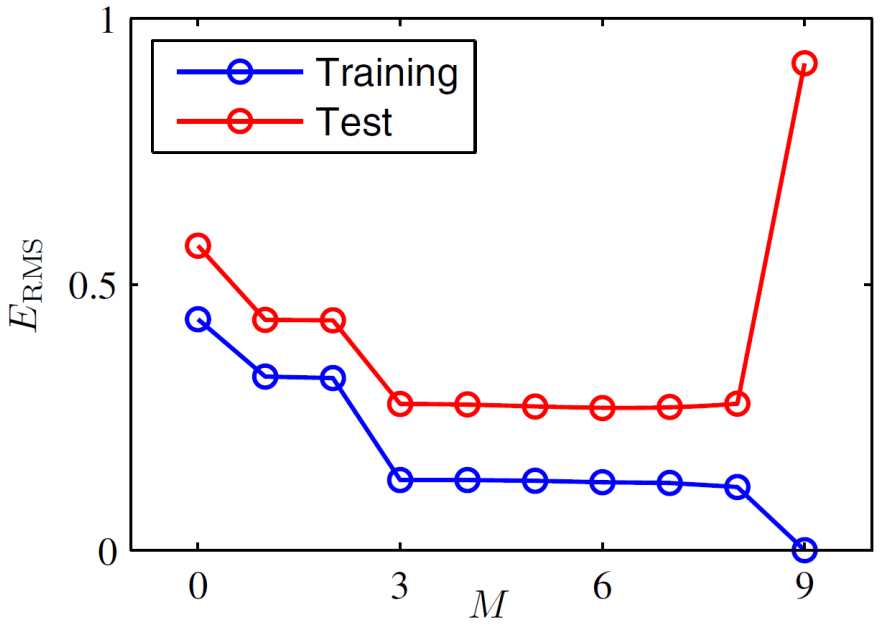
Il nostro scopo è avere buoni risultati di approssimazione su dati mai visti

Possibile soluzione: valuto il modello su un nuovo dataset (indipendente dal training set) e vedo come performa

Errore su un nuovo dataset

Valuto:

$$E_{RMS} = \sqrt{\frac{2RSS(\boldsymbol{w}^*)}{N}}$$



Il modello generalizza in maniera diversa per differenti basi scelte

Soluzioni:

- Aumentiamo il numero di dati nel training set
- Studiamo un modo per contenere il numero dei parametri

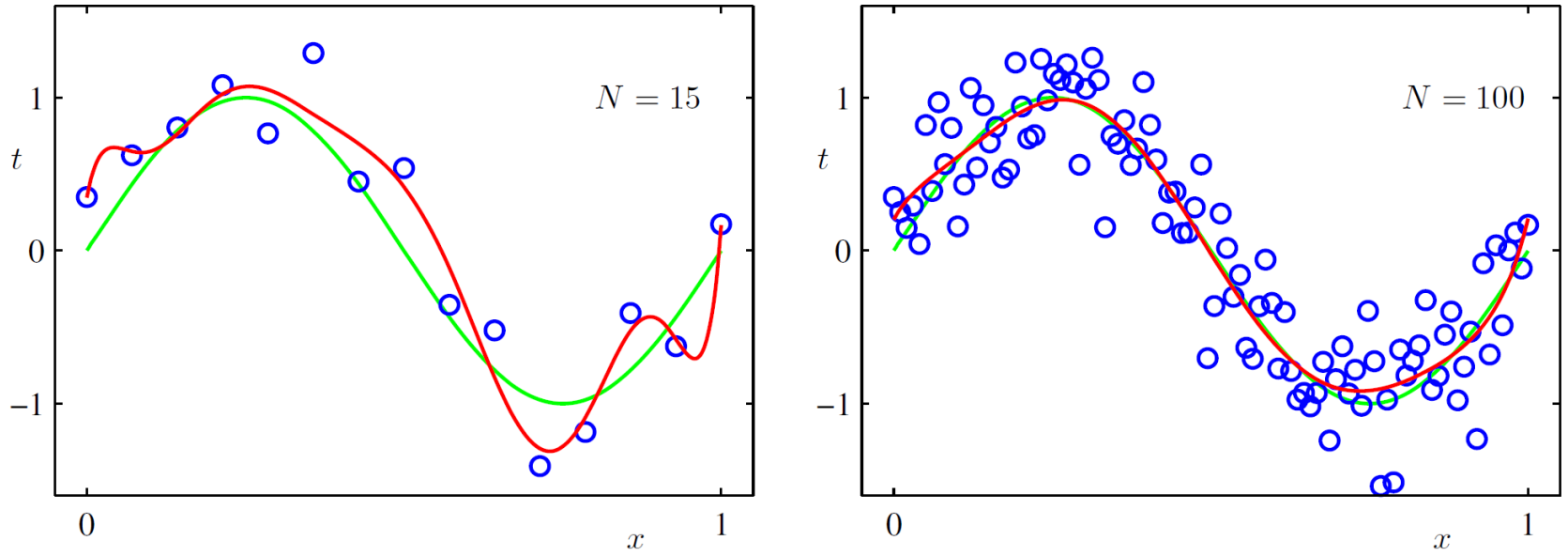
Sessione pratica (2)

Valutiamo il modello di regressione su un dataset indipendente

Training/Test split



Aumentare le dimensioni del training set



La «rule of thumb» è che si dovrebbero avere dati pari a circa 5-10 volte il numero dei parametri

Di solito i dati sono **dati** (non ne posso aggiungere altri)

Tecniche per la selezione del modello

Per risolvere il dilemma tra underfitting e overfitting esistono differenti tecniche:

- Model selection
  - **Feature selection**: prendo in considerazione solo un subset degli input che è significativo per il problema
  - **Adjustment techniques**: cerco di considerare contemporaneamente la complessità del modello e il suo errore
  - Dimensionality reduction: proietto gli input in uno spazio dimensionalmente più piccolo di quello originale (es. PCA)

- Ensemble models
  - Bagging
  - Boosting

Selezione dei modelli

Il modello che usa tutte le feature disponibili tendenzialmente avrà l'errore sul training più piccolo

Dobbiamo stimare l'errore sul test set

- Validazione
- Adjustment techniques

Attenzione!!!
Non possiamo usare il test set per decidere il modello altrimenti la stima che otteniamo sul test set sarebbe dipendente dal training

Validation

Divido ulteriormente il training set in due parti, una delle due la uso per scegliere il modello

Prima riordino casualmente i dati

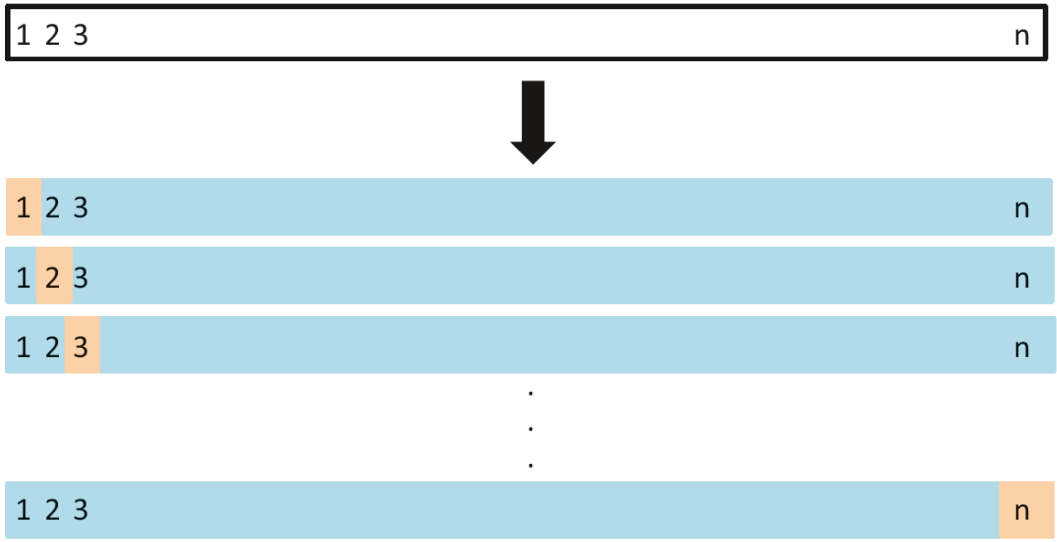
Scelgo il modello con errore minore sul **validation set**

Pro: ogni modello di differente complessità dovrà essere addestrato una sola volta

Contro: potremmo avere fenomeni di overfitting sul validation set

Leave One Out

Scelgo un validation set con un solo dato e ripeto la procedura per ogni sample nel training set ed uso il valor medio dell'errore come stima dell'errore sul test set

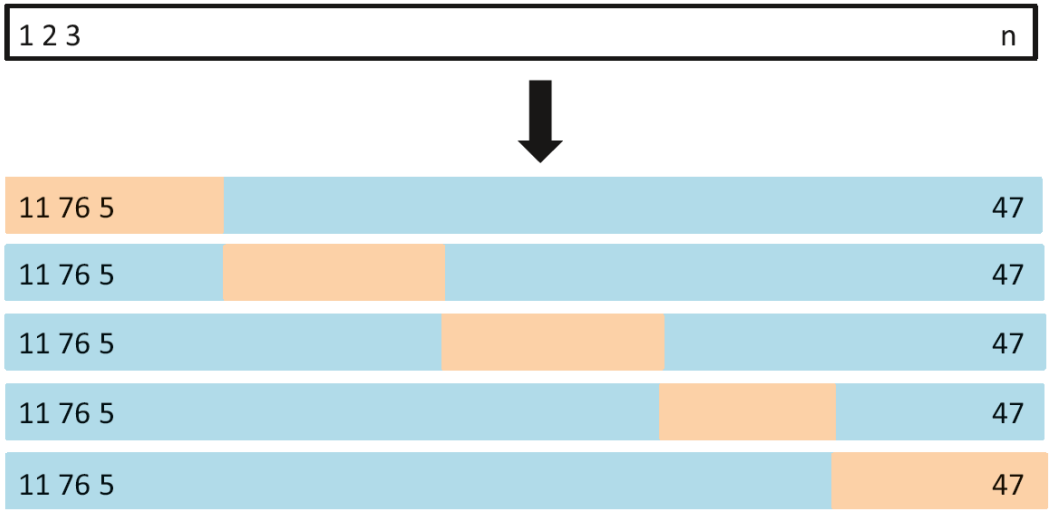


Pro: errore leggermente più pessimistico del reale, ma quasi unbiased

Contro: computazionalmente oneroso

K-fold crossvalidation

Invece di validare su di un punto alla volta ne considere un piccolo set e ripeto l'operazione con tutti i subset indipendenti di quella dimensione presenti nel training set



Mitiga i problemi ed i vantaggi dei due metodi precedenti

Adjustment Techniques

Mallow's C_p:

$$C_p = \frac{1}{N}(RSS + 2d\tilde{\sigma}^2)$$

Akaike Information Criterion:

$$AIC = -2\log L + 2d$$

Bayesian Information Criterion:

$$BIC = \frac{1}{N}(RSS + \log(N)d\tilde{\sigma}^2)$$

Adjusted R^2

$$AdjustedR^2 = 1 - \frac{RSS/(N-d-1)}{TSS/(N-1)}$$

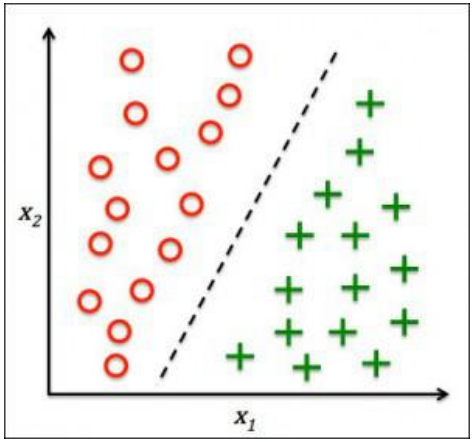
Supervised Learning - Classificazione

Classificazione

Vogliamo apprendere una relazione tra un input $\boldsymbol{x}$ e un output (target) t che assumiamo essere appartenente ad un insieme non ordinato

Esempi:

- Dato un paziente dire se è malato
- Dato un pezzo meccanico dire se è difettoso
- Data un'immagine dire che oggetto contiene
- Data un'immagine dire che numero contiene



Classificazione lineare

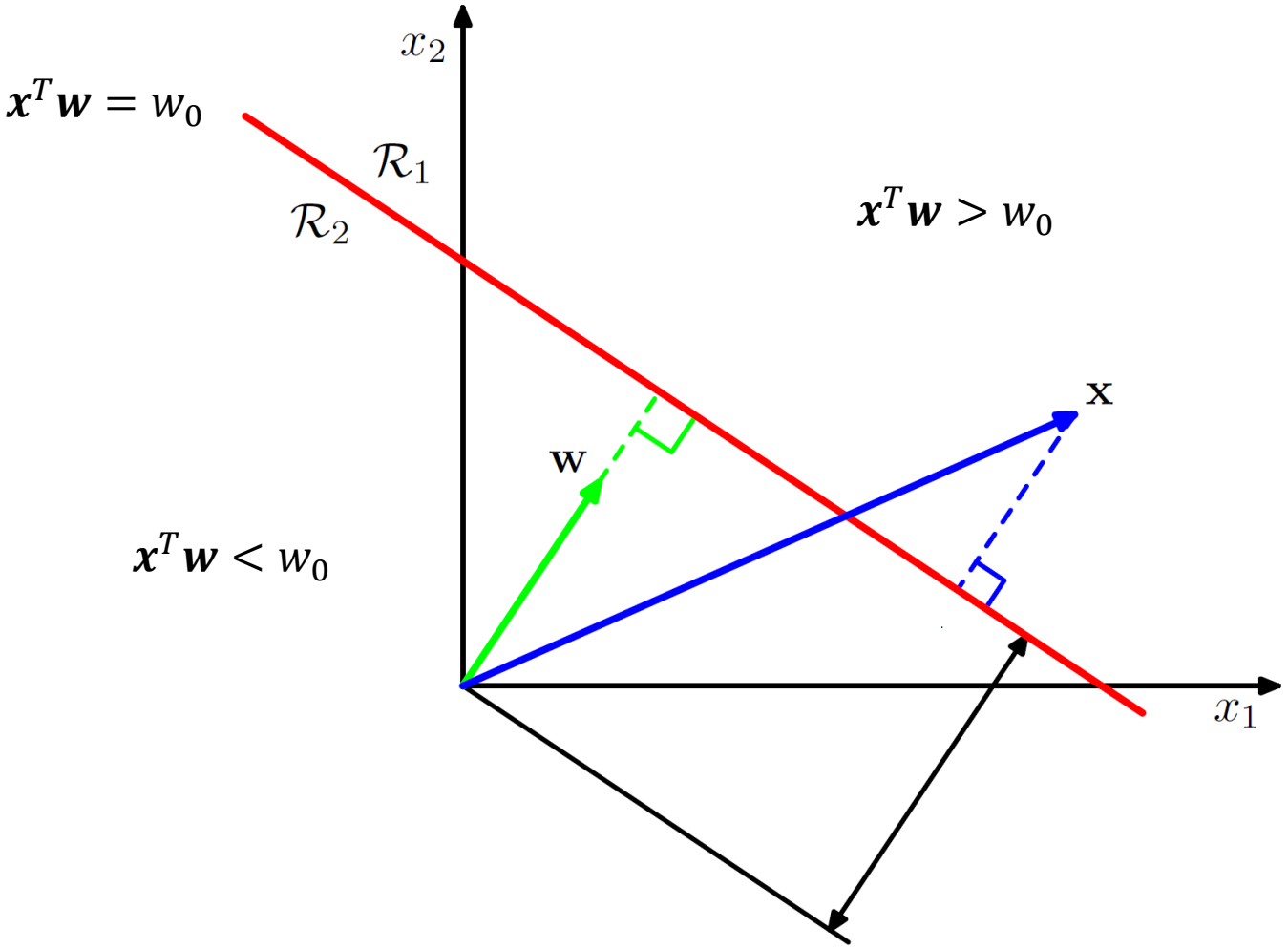
Vogliamo dividere lo spazio degli input in regioni corrispondenti alle classi

I confini di tali regioni sono detti **decision boundaries** o **decision surfaces**

Qui ci limiteremo a considerare dei modelli che scelgono come decision boundaries delle superfici lineari (rette, piani o iperpiani)

$$y(\mathbf{x}, \mathbf{w}) = f(\mathbf{x}^T \mathbf{w} + w_0)$$

Decision boundary

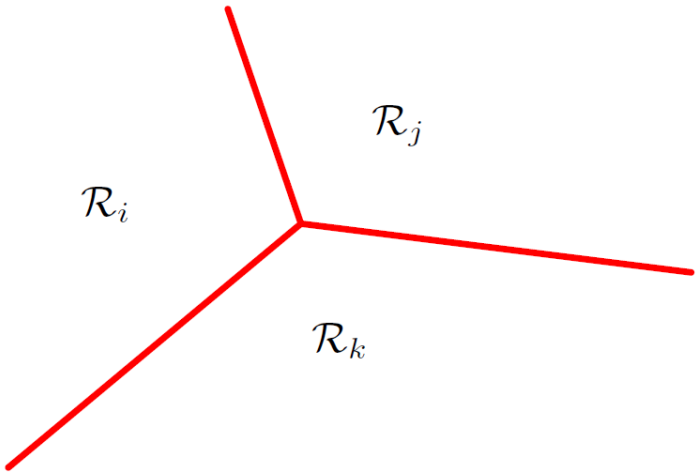


Soluzione multiclasse: K vs. all

Addestro K classificatori in cui separo una classe da tutte le altre

$$y_k(\mathbf{x}) = \mathbf{w}_k^{\mathrm{T}}\mathbf{x} + w_{k0}$$

E assegno il punto alla classe $y_k(\mathbf{x}) > y_j(\mathbf{x})$ ıi $j \neq k$



Valutazione di un metodo di classificazione

Ci sono differenti metriche per valutare le prestazioni di un metodo di classificazione (nel caso in cui abbiamo due sole classi)

	Actual Class: 1	Actual Class: 0
Predicted Class: 1	tp	fp
Predicted Class: 0	fn	tn

Accuracy	$Acc = \frac{tp+tn}{N};$
Precision	$Pre = \frac{tp}{tp+fp};$
Recall	$Rec = \frac{tp}{tp+fn};$
F1 score	$F1 = \frac{2 \cdot Pre \cdot Rec}{Pre+Rec};$

Con più classi:

$$Acc = \frac{\sum_{k \in C_1,\ldots,C_K} tc_k}{N}$$

Metodi classici di classificazione

Metodi con boundaries lineari

- Perceptron
- **Logistic regression**
- SVM

Metodi Bayesiani

- Naive Bayes

Metodi non parametrici

- K-nearest neigbour
- SVM

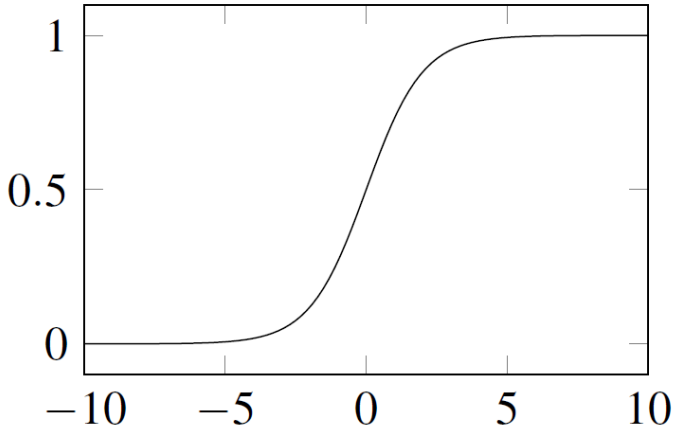
Hypothesis space: Logistic Regression

Idea: non modellizziamo l'output direttamente, ma la probabilità di appartenere ad una delle due classi, usando una funzione sigmoidale

Se ho due classi

$$p(C_1|\phi) = \frac{1}{1+\exp(-\mathbf{w}^T\phi)} = \sigma(\mathbf{w}^T\phi)$$

$$p(C_2|\phi) = 1 - p(C_1|\phi)$$



Loss: likelihood

Voglio massimizzare la probabilità di classificare correttamente il dataset

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n}, \quad y_n = \sigma(\mathbf{w}^T \boldsymbol{\phi}_n)$$

Definiamo

$$L(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = -\sum_{n=1}^{N} (t_n \ln y_n + (1 - t_n) \ln(1 - y_n)) = \sum_{n=1}^{N} L_n$$

e

$$\frac{\partial L_n}{\partial y_n} = \frac{y_n - t_n}{y_n(1 - y_n)}, \quad \frac{\partial y_n}{\partial \mathbf{w}} = y_n(1-y_n)\boldsymbol{\phi}_i,$$

$$\frac{\partial L_n}{\partial \mathbf{w}} = \frac{\partial L_n}{\partial y_n}\frac{\partial y_n}{\partial \mathbf{w}} = (y_n - t_n)\boldsymbol{\phi}_n$$

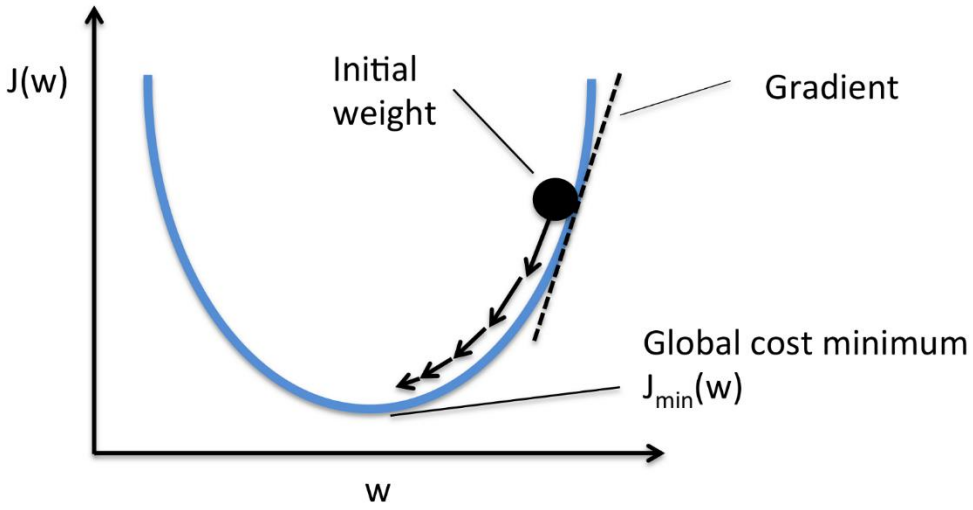
Ottimizzazione: gradient descend

Possiamo qundi calcolare il gradiente della loss $\nabla L(\boldsymbol{w})$

$$\nabla L(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n$$

In questo caso non abbiamo una forma chiusa per la sua minimizzazione

Possiamo usare l'algoritmo gradient descend in quanto il problema è convesso



Sessione pratica (3)

Classifichiamo i fiori dell’iris dataset

Unsupervised Learning - Clustering

Quando usare tecniche di unsupervised learning

Abbiamo a disposizione un set di dati molto grande

Non riusciamo a gestirli tutti per prendere decisioni

Soluzione: riduciamone il numero

- Riduciamo il numero di sample (clustering)
- Riduciamo la dimensionalità di ognuno dei dati (dimensionality reduction)

Clustering

Voglio estrarre dal dataset un sottoinsieme di sample che siano rappresentativi dell'intero dataset

Divido il dataset in sottonsiemi simili di sample

Domande:

- Come valuto se due sample sono simili tra loro?
- Come valuto il risultato ottenuto? (non esiste una risposta condivisa)

K-means clustering

Idea: uso un vettore medio come rappresentativo di ogni cluster

Scelgo un numero di cluster (gruppi) K e minimizzo:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|_2^2$$

Appartenenza del sample n al cluster k

Rappresentante del cluster k (media)

Algoritmo K-means

Una procedura analitica per la minimizzazione non esiste

Procedura iterativa:

- Parto con dei rappresentanti μ_k^0 scelti a caso
- Ripeto fino a convergenza (nessun punto si sposta più)
  - Assegno ogni sample $\boldsymbol{x}_n$ ad un cluster
  - Calcolo i nuovi rappresentanti μ_k^i come media dei sample appartenenti al cluster

Poiché la procedura diminuisce il valore della funzione obbiettivo J ad ogni iterazione, la procedura converge

Potrebbe convergere ad un minimo locale

Più in dettaglio

Ogni sample viene assegnato ad un cluster se

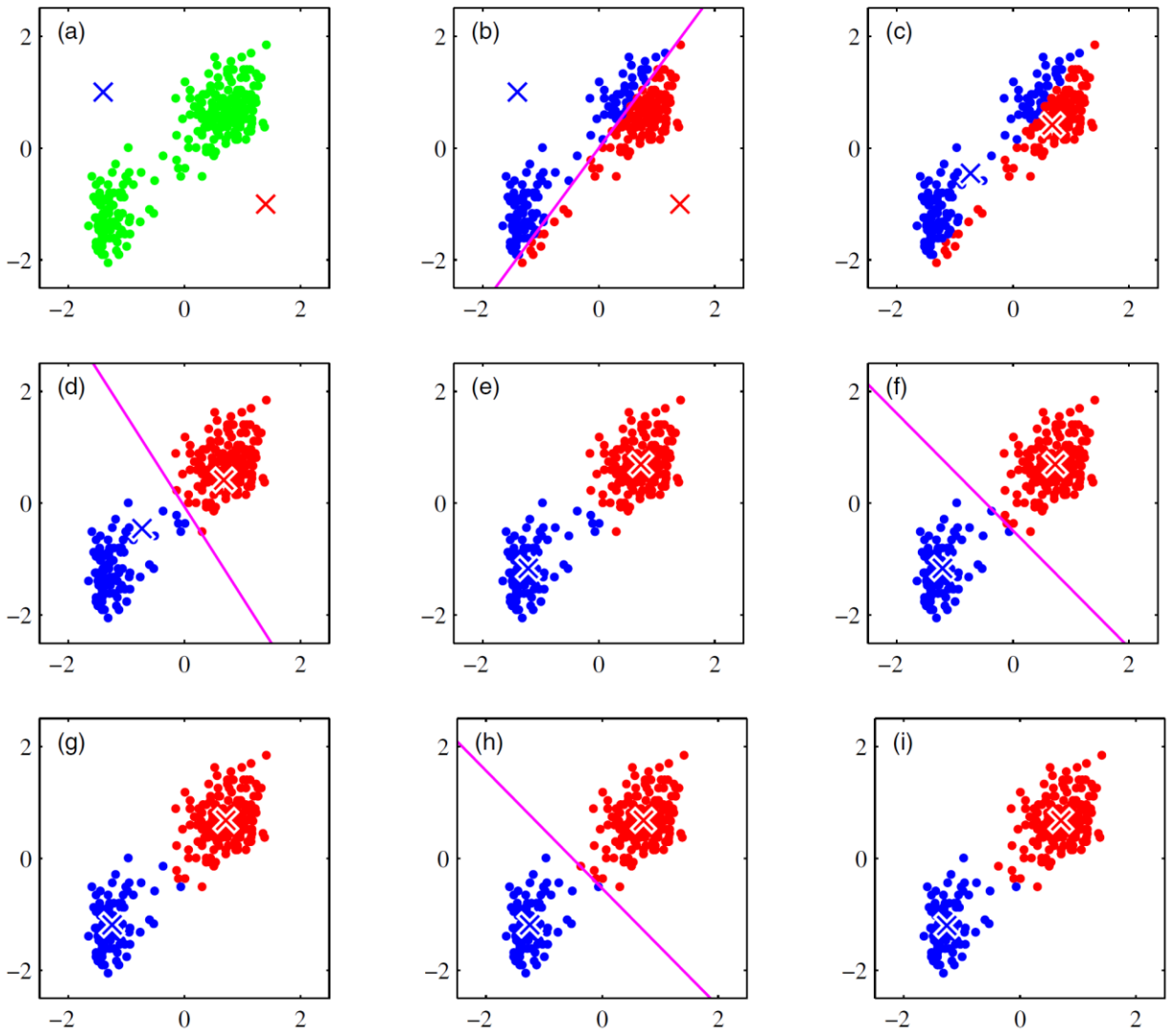
$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

Dopodichè minimizzo la funzione di costo J, ponendo la sua derivata a zero

$$2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \quad \Longrightarrow \quad \boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

ovvero stiamo utilizzando la media dei punti nel cluster come rappresentante

Esempio di K-means



Sessione pratica (4)

Clusterizziamo l’iris dataset

Vediamo quanto l’informazione di specie è inclusa negli input del dataset

Problemi del K-means

Utilizzando come misura di dissimilarità la distanza euclidea:

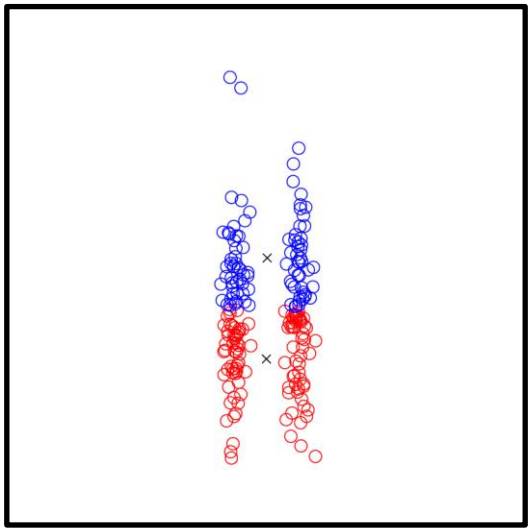
- Non è robusto agli outliers
- Non possiamo applicarlo a dati categorici (non esistono i rappresentanti medi)

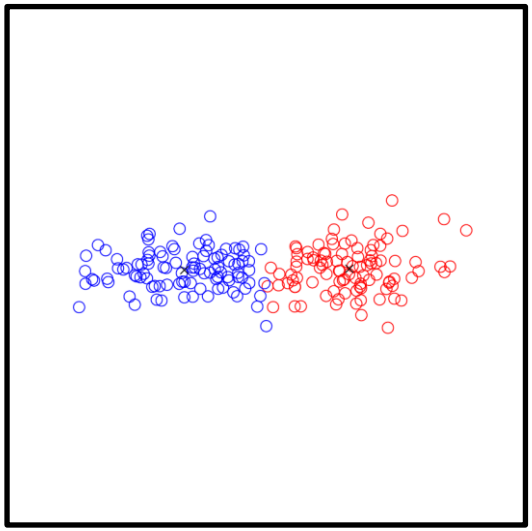
Il K-means assume che i cluster debbano:

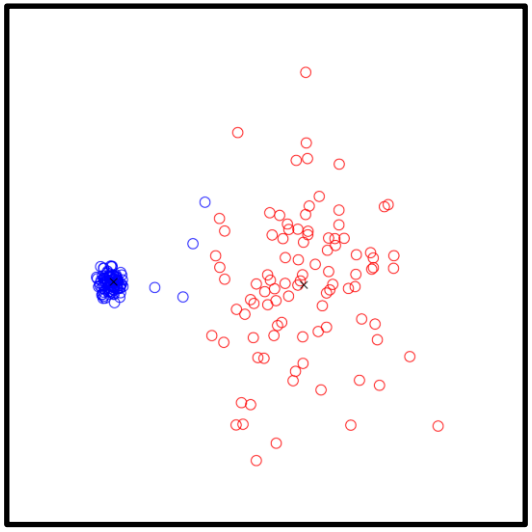
- essere sferici
- essere ben separati
- avere lo stesso volume
- avere lo stesso numero di punti

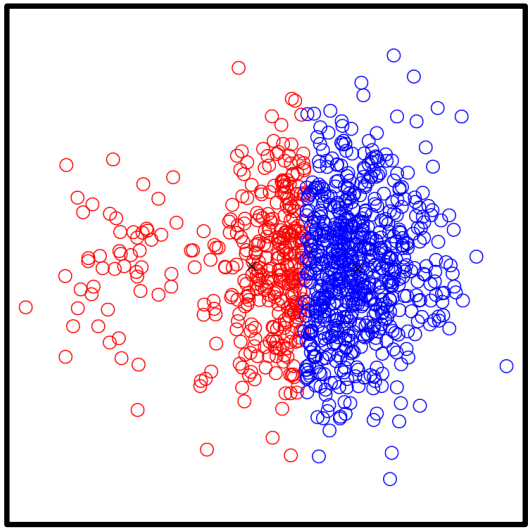
In caso contrario potrebbe portare a risultati non ragionevoli

Esempi di clustering finito male





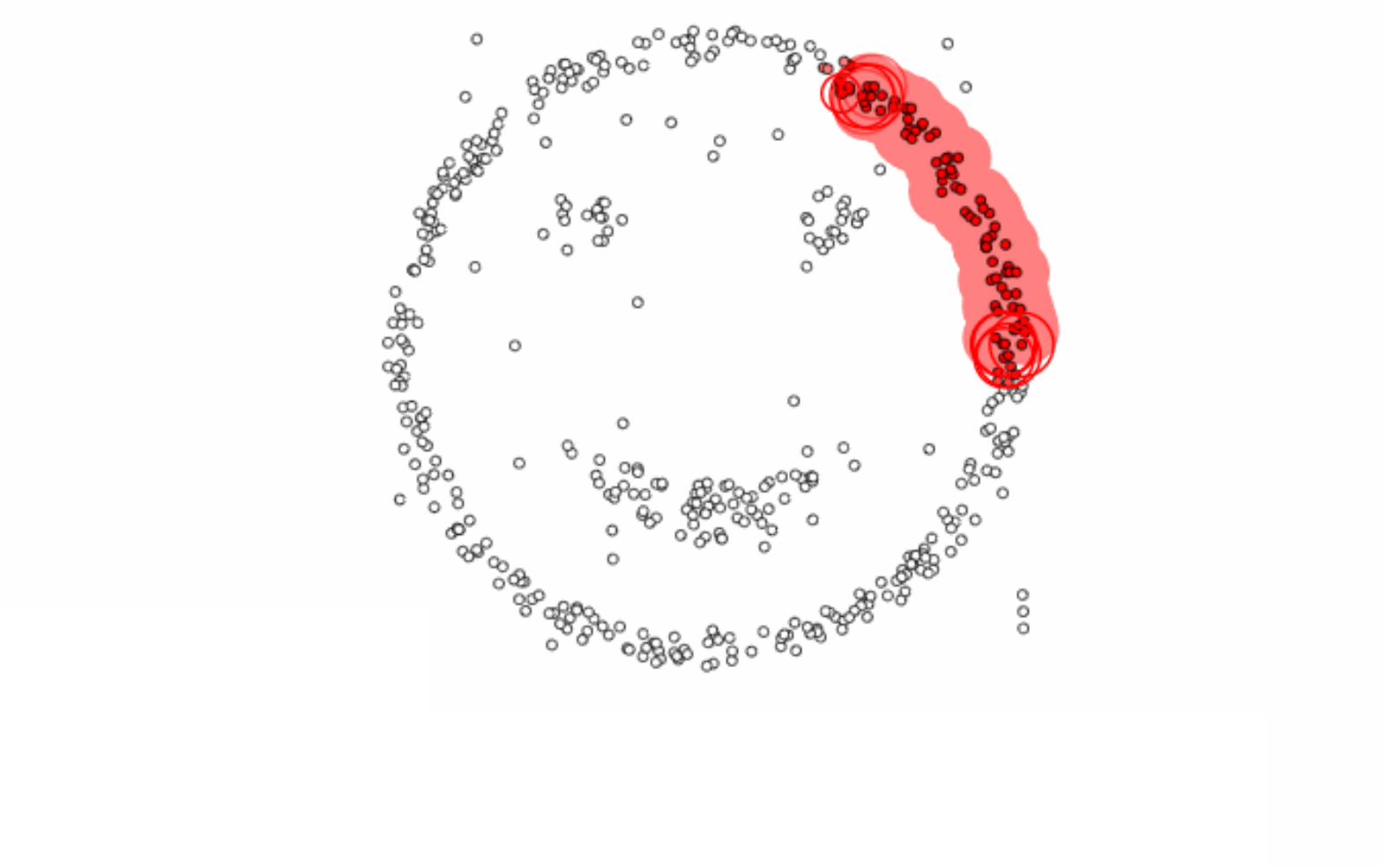




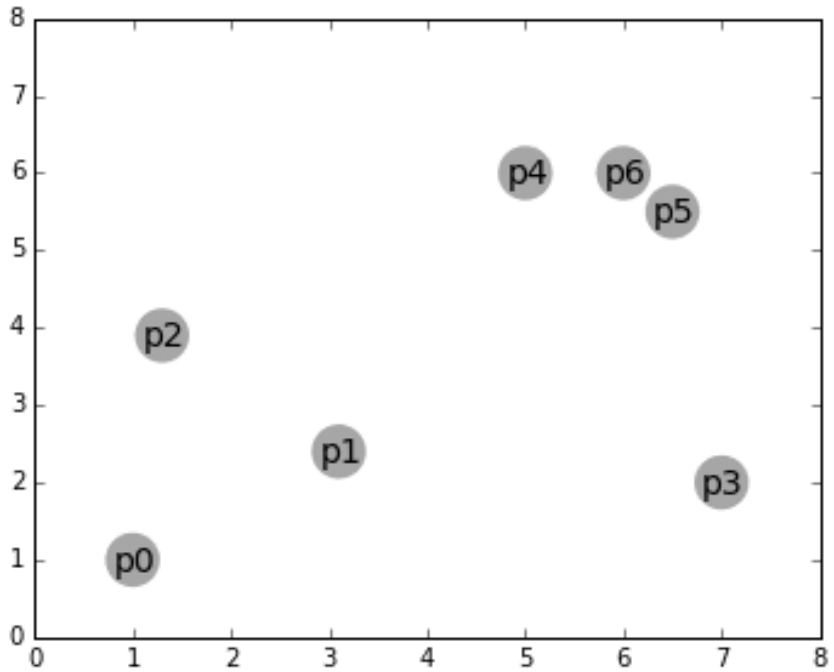
Altri metodi di clustering

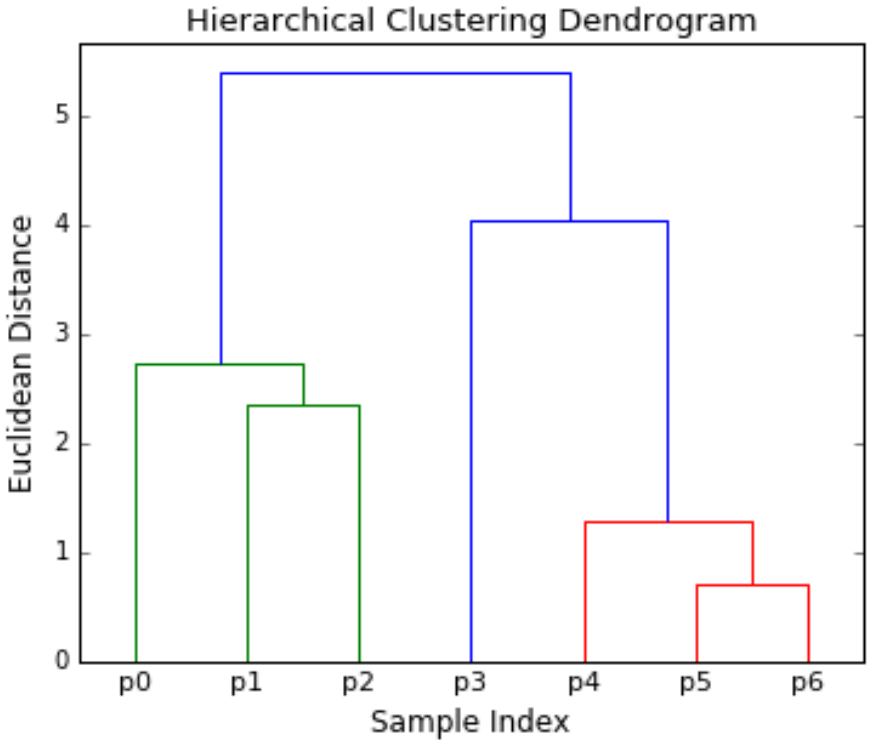
- X-means: sceglie in maniera automatica il numero di cluster che vogliamo identificare
- Mixture models: approssimano i dati con una mistura di distribuzioni
- **DB-scan**: considera l'esistenza di alcuni punti non assegnati ad alcun cluster. Non richiede di specificare il numero di cluster
- **Clustering gerarchico**: raggruppa per primi i punti più vicini tra di loro, dopodichè continua fino a creare gruppi con distanza sempre più alta tra i punti
- Affinity propagation: basato sul concetto di message passing. Non richiede di specificare il numero di cluster
- Clustering spettrale: utilizza degli indici calcolati sulla matrice di dissimilarità tra i sample per raggruppare poi i dati

Esempio: DBScan



Esempio: Clustering gerarchico





Conclusioni

Conclusioni

Oggi abbiamo visto:

- Visione delle potenzialità delle tecniche di Intelligenza Artificiale
- Definizione e topologia delle tecniche di Machine Learning
  - Regressione (lineare)
  - Classificazione (Logistic Regression)
  - Clustering (K-means)

Dove trovare fonti di ML

Riviste scientifiche

- Journal of Machine Learning Research (JMLR)
- Machine Learning Journal (MLJ)
- Journal of Artificial Intelligence Research (JAIR)

Conferenze internazionali

- International Conference on Machine Learning (ICML)
- Neural Information and Processing Systems (NIPS)
- American Association on Artificial Intelligence (AAAI)

MOOC

- Statistical Learning, Stanford Online (In R)
- Machine Learning, Coursera (In Matlab)

Bibliografia

Libri di testo

- Alpaydin, E. *Introduction to machine learning*
- Bishop, C. M. *Pattern recognition and machine learning*
- James, G., Witten, D., Hastie, T., Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*
- Sutton, R.S., Barto, A. G. *Reinforcement learning: An introduction*