

# Introduzione al Data Mining e applicazioni al dominio del Contact Management

Andrea Brunello

Università degli Studi di Udine

28 Ottobre 2015



*In collaborazione con dott. Enrico Marzano, CIO Gap srl  
progetto Active Contact System*

Introduzione al  
Data Mining

Andrea Brunello

Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

Tipologie di  
Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

Modelli principali

Supervised Learning

Unsupervised Learning

Valutazione dei  
modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

Combinazione di  
più modelli

Bagging

Randomization

Boosting

Applicazioni

Performance agenti

Estensioni

Riferimenti

## Contenuti presentati nel corso di 3 incontri:

- ▶ Incontro I:
  - ▶ cos'è il Data Mining e come si svolge;
  - ▶ classificazione delle tipologie di Learning;
  - ▶ principali modelli utilizzati.
- ▶ Incontro II:
  - ▶ valutazione dei modelli;
  - ▶ combinazione di più modelli;
- ▶ Incontro III:
  - ▶ punto della situazione presso Gap;
  - ▶ casi pratici di applicazione, nel contesto della valutazione degli agenti.

### Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

### Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

### Modelli principali

Supervised Learning  
Unsupervised Learning

### Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

### Combinazione di più modelli

Bagging  
Randomization  
Boosting

### Applicazioni

Performance agenti  
Estensioni

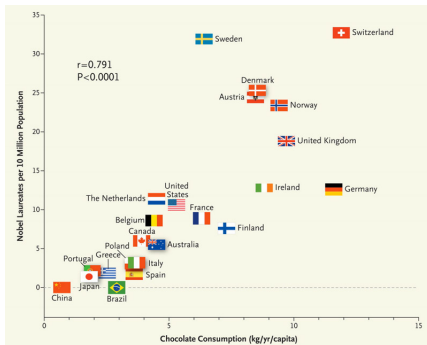
### Riferimenti

- ▶ **Dati**  $\approx$  **fatti** memorizzati, registrati;
- ▶ L'**informazione** è costituita dall'insieme dei **concetti**, delle regolarità, degli schemi che si trovano "nascosti" fra i dati;
- ▶ Il **Data Mining** si occupa dell'**estrazione** e della presentazione di **informazione** utile, precedentemente sconosciuta, ed implicitamente contenuta in una (grande) mole di dati.
  - ▶ E' un processo di astrazione (generazione di un modello).
- ▶ Il **Machine Learning** costituisce la "base tecnica" del Data Mining.

- ▶ Utilizzare i modelli/pattern appresi per:
  - ▶ **conoscere**: comprendere che determinate fasce di popolazione sono più propense ad acquistare un determinato bene;
  - ▶ **inferire**: probabile guasto ad un macchinario, da un insieme di sintomi;
  - ▶ **predire**: stabilire se e quale variazione nelle vendite risulterà da un aumento del budget pubblicitario.
- ▶ tali fini possono mescolarsi, si pensi alla ricerca di un modello che fornisca la valutazione di un'abitazione sulla base di diversi valori in input.

# Cos'è il Data Mining (3)

- ▶ spesso i pattern scoperti risulteranno banali, frutto di correlazioni casuali, o non completamente corretti.
- ▶ ciò può essere dovuto a:
  - ▶ errori nei dati;
  - ▶ caratteristiche del dominio (es. esistenza di dipendenze funzionali)



## Riassumendo:

- ▶ Il Data Mining sfrutta tecniche di Machine Learning
- ▶ per estrarre semi-automaticamente
- ▶ da (grandi) quantità di dati
- ▶ informazioni, pattern utili

## Input del processo:

- ▶ istanze, esempi dei concetti che si vogliono apprendere

## Output del processo:

- ▶ predizioni/classificazioni
- ▶ modelli

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

# Il processo di Data Mining

In genere, il processo di Data Mining si articola come segue:

- ▶ il tutto inizia con il porsi una **domanda**, ben chiara e specifica;
- ▶ in seguito, si passa alla raccolta dei **dati** da utilizzare come input;
- ▶ viene selezionato un insieme di **caratteristiche** (features) ritenute importanti su tali dati, e per il fine che si vuole ottenere;
- ▶ si applica un **algoritmo** di machine learning sul dataset così definito, in modo da “addestrare” un modello;
- ▶ dopo un’eventuale fase di tuning, il modello prodotto dall’algoritmo viene **valutato**, ed è infine pronto per essere utilizzato.

# Un primo esempio: SPAM vs HAM

Ripercorriamo ora il processo di Data Mining con un esempio focalizzato sulla predizione.

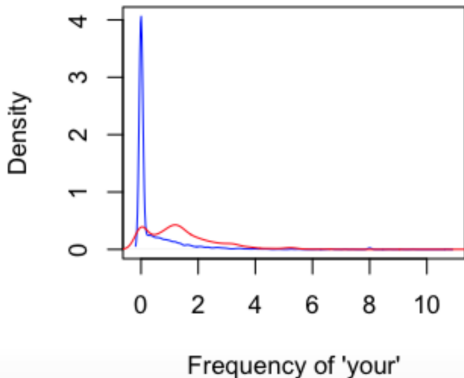
- ▶ **Domanda:** è possibile distinguere automaticamente fra i messaggi email che sono indesiderati (SPAM) e quelli legittimi (HAM)?
- ▶ **Dati:** insieme di 4601 istanze di email già classificate, ed ognuna avente 57 caratteristiche indicanti la frequenza di determinate parole e caratteri nel corpo del messaggio;
  - ▶ [www.inside-r.org/packages/cran/kernlab/docs/spam](http://www.inside-r.org/packages/cran/kernlab/docs/spam)



# Un primo esempio: SPAM vs HAM (2)

## Selezione delle **caratteristiche**:

- ▶ processi di *attribute selection*;
- ▶ esplorazione e selezione manuale.



## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

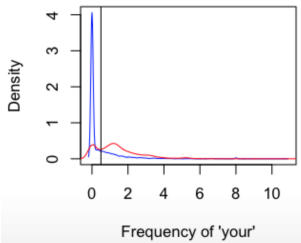
Estensioni

## Riferimenti

# Un primo esempio: SPAM vs HAM (3)

**Modello:** utilizzo di un valore di **cutoff**

- ▶ se frequenza della parola “your” nel testo  $> 0.5$ , allora l'email è SPAM



- ▶ Otteniamo la **tabella di contingenza** (valori su insieme di training):

<i><b>Predizione \ Classe</b></i>	<b>nonspam</b>	<b>spam</b>
<b>nonspam</b>	0.4590	0.10017
<b>spam</b>	0.1469	0.2923

Entrando nel dettaglio del processo di Data Mining, abbiamo che il suo input è costituito da:

- ▶ **concetti**
- ▶ **istanze**
- ▶ **attributi:**
  - ▶ *numerici VS nominali*
  - ▶ *feature VS target* (supervised learning)

Rivestono grande importanza l'integrazione, pulizia, e **trasformazione** dei dati.

↪ raramente le istanze in input copriranno tutti i casi possibili per il dominio!

# L'input del processo (2)

Condizioni	Temp.	Umidità	Vento	Si_gioca?
soleggiato	calda	alta	falso	no
soleggiato	calda	alta	vero	no
nuvoloso	calda	alta	falso	si
pioggia	mite	alta	falso	si
pioggia	fredda	normale	falso	si
pioggia	fredda	normale	vero	no
nuvoloso	fredda	normale	vero	si
soleggiato	mite	alta	falso	no
soleggiato	fredda	normale	falso	si
pioggia	mite	normale	falso	si
soleggiato	mite	normale	vero	si
nuvoloso	mite	alta	vero	si
nuvoloso	calda	normale	falso	si
pioggia	mite	alta	vero	no

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

# Supervised vs Unsupervised Learning

Distinzione fondamentale, a seconda dell'obiettivo del processo:

- ▶ **Supervised Learning:** all'algoritmo di learning viene fornito un risultato noto per ciascuna istanza di training e si punta a determinare il valore per nuove istanze (attributo obiettivo).
  - ▶ *Classificazione con alberi, regressione lineare, ...*
- ▶ **Unsupervised Learning:** non si cerca di prevedere il valore di uno specifico attributo, ma viene ricercata ogni possibile associazione/correlazione fra gli attributi.
  - ▶ *Clustering, regole di associazione, ...*

# Problemi di regressione e di classificazione

Nel *Supervised Learning*, a seconda della tipologia dell'attributo obiettivo, distinguiamo problemi di

- ▶ **classificazione:** (o predizione). Si vuole assegnare a ciascuna istanza uno di un insieme finito di valori (*cl. discreti*; in altri casi viene restituita la probabilità di appartenere a ciascuna classe, o un ranking delle istanze: *cl. continui*);
- ▶ **regressione:** si vuole assegnare a ciascuna istanza un valore numerico.

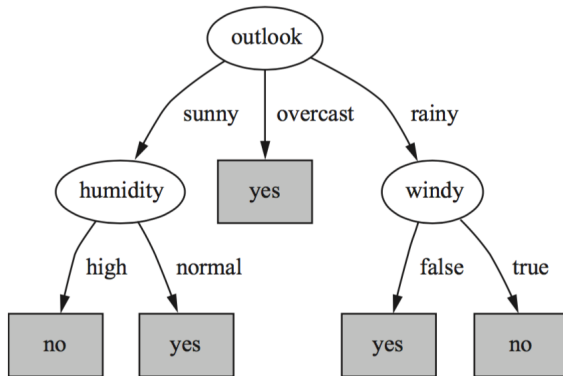
Alcune famiglie di algoritmi di learning si adattano ad entrambi i problemi (alberi di classificazione e di regressione, regressione lineare e logistica, ...).

Metodo semplice e compatto per rappresentare l'output di un processo di **classificazione**.

- ▶ ciascun nodo interno corrisponde alla valutazione di determinato attributo;
- ▶ la classificazione di un'istanza avviene partendo dalla radice, e scendendo via via sino a giungere ad una foglia etichettata con una determinata classe (o insieme di classi, distribuzione di probabilità);
- ▶ variante per problemi di regressione: *alberi di regressione*.

## Alberi di decisione (2)

L'albero di decisione classifica correttamente tutte le istanze del *Weather Problem*.



Osserviamo che non viene mai presa in considerazione la *temperatura*.



Presentiamo una metodologia generale per la creazione di un albero di decisione, simile a quella adottata da *ID.3*:

- ▶ costruzione dell'albero ricorsiva, partendo dalla radice;
- ▶ ad ogni passo si sceglie l'attributo su cui effettuare lo split, intuitivamente quello che porta al maggior *information gain* (alberi più bassi, metodo *greedy*);
- ▶ sino alla generazione di una foglia, in corrispondenza di un caso base:
  - ▶ gli elementi nel nodo hanno stessa classe (attenzione all'overfitting, *i.e.* modello costruito ad-hoc sui dati, che non generalizza), oppure
  - ▶ si è raggiunto un sufficiente grado di purezza del nodo .

# Costruzione di un albero di decisione (2)

Come misurare il guadagno di informazione dato dallo split su un attributo?

- ▶ insieme  $T$  di istanze partizionate nelle classi  $C = \{C_1, \dots, C_k\}$  dall'attributo obiettivo;

- ▶ distribuzione di probabilità associata a  $T$ :

$$P = (|C_1|/|T|, |C_2|/|T|, \dots, |C_k|/|T|)$$

- ▶ definiamo l'informazione necessaria ad identificare la classe di un elemento di  $T$  come:

$$Info(T) = H(P) = - \sum_{i=1}^k p_i * \log(p_i)$$

- ▶ intuitivamente, se quasi tutti gli elementi fanno parte di una stessa classe,  $Info(T)$  è bassa, ed aumenta all'aumentare della "confusione" (entropia).

# Costruzione di un albero di decisione (3)

Cosa succede partizionando l'insieme di istanze sulla base di un attributo?

- ▶ supponiamo di suddividere  $T$  in sottoinsiemi  $T_1, \dots, T_n$  sulla base del valore di una delle *feature*, diciamo  $X$ ;
- ▶ l'informazione necessaria ad identificare la classe di un elemento di  $T$  è la media pesata dell'informazione necessaria ad identificare la classe dell'elemento all'interno di ciascun sottoinsieme:

$$Info(X, T) = \sum_{i=1}^n (|T_i|/|T|) * Info(T_i)$$

# Costruzione di un albero di decisione (4)

Diamo infine la definizione di *information gain*:

$$Gain(X, T) = Info(T) - Info(X, T)$$

- ▶ rappresenta la differenza fra l'informazione necessaria ad identificare la classe di un elemento di  $T$  e l'informazione necessaria dopo la suddivisione di  $T$  in sottoinsiemi attraverso l'attributo  $X$ ;
- ▶ in altre parole, il guadagno d'informazione dovuto all'attributo  $X$  (alto è meglio).

# Costruzione di un albero di decisione (5)

## Considerazioni finali:

- ▶ utilizzare l'information gain può portare a preferire la scelta di attributi con un gran numero di valori (*highly branching attributes*);
- ▶ il che porta spesso a sua volta a fenomeni di overfitting;
- ▶ l'*information gain ratio* considera anche il numero e la dimensione dei vari sottoinsiemi che verrebbero generati.

Oltre all'entropia, esistono altre misure di impurità dei nodi.

- ▶ indice di impurità di Gini;
- ▶ se l'attributo da predire è numerico, si può utilizzare *RMSE* rispetto alla media del nodo;
- ▶ ...

## Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

## Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning  
Unsupervised Learning

## Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

## Combinazione di più modelli

Bagging  
Randomization  
Boosting

## Applicazioni

Performance agenti  
Estensioni

## Riferimenti

Diversi algoritmi per la costruzione degli alberi:

- ▶ *ID.3, Iterative Dichotomizer 3* (attributi nominali);
- ▶ *C4.5* (implementazione Weka *J48*);
- ▶ *C5.0*;
- ▶ *CART, Classification And Regression Trees*;
- ▶ ...

Alternativa rispetto agli alberi di decisione.

*Condizioni = soleggiato  $\wedge$  Umidita = alta  $\rightarrow$  Si\_gioca = no*

*Condizioni = pioggia  $\wedge$  Vento = vero  $\rightarrow$  Si\_gioca = no*

*Condizioni = nuvoloso  $\rightarrow$  Si\_gioca = si*

*Umidita = normale  $\rightarrow$  Si\_gioca = si*

*Else Si\_gioca = si*

- ▶ parte a sinistra: precondizioni
- ▶ parte a destra: classe dell'attributo su cui si sta facendo predizione
- ▶ tipicamente intese per essere considerate in ordine
- ▶ derivabili da un albero di decisione, o costruzione ad-hoc es. tramite il *metodo delle coperture*

A differenza che nei metodi precedenti non vi è una fase iniziale di “training”, l’idea è la seguente:

- ▶ si ha a disposizione un insieme di istanze delle quali è nota la classe;
- ▶ data una nuova istanza, essa viene ricondotta ad uno o più dei casi noti:
  - ▶ *nearest neighbour*
  - ▶ *k-nearest neighbour*
- ▶ diverse nozioni di “distanza”;
- ▶ l’output può essere qualitativo o quantitativo.



Output numerico (*variabile dipendente*), sulla base di attributi in input numerici (*variabili indipendenti*).

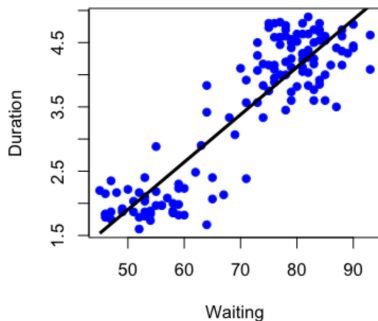
- ▶ si vuole esprimere la variabile dipendente come combinazione di una o più variabili indipendenti:

$$X = w_0 + w_1 a_1 + w_2 a_2 + \dots + w_k a_k$$

- ▶ metodo semplice e di intuitiva interpretazione, tuttavia ha delle limitazioni:
  - ▶ la relazione fra variabile dipendente ed indipendenti deve essere lineare;
  - ▶ le variabili indipendenti non devono essere “correlate” fra loro (no *multicollinearità*);
  - ▶ altre condizioni di applicabilità: [https://en.wikipedia.org/wiki/Linear\\_regression](https://en.wikipedia.org/wiki/Linear_regression).

# Modelli di regressione lineare (2)

Durata eruzione	Tempo attesa
2.883	55
1.883	54
2.167	52
1.600	52
1.750	47
1.967	55



## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

Esistono varianti del modello di regressione (ed ancora più algoritmi che le implementano):

- ▶ *regressione polinomiale*: per relazioni non lineari
- ▶ *regressione multivariata*: per predire il valore di più di una variabile
- ▶ *regressione logistica*: per classificazione binaria
- ▶ *regressione logistica multinomiale e analisi discriminante lineare*: per classificazione multiclasse

I Naive Bayes sono una famiglia di classificatori probabilistici, basati sul teorema di Bayes della probabilità condizionata:

- ▶ l'idea alla base è che ciascuna feature nel dataset contribuisca in modo indipendente, e con lo stesso peso, alla determinazione della classe dell'istanza;
- ▶ l'assunzione è molto forte;
- ▶ tuttavia, il metodo Naive Bayes funziona sorprendentemente bene nei casi reali.

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

## Tabella con dati riassuntivi sul *Weather Problem*:

**Table 4.2** Weather Data with Counts and Probabilities

Outlook			Temperature			Humidity			Windy			Play	
	yes	no		yes	no		yes	no		yes	no	yes	no
sunny	2	3	hot	2	2	high	3	4	false	6	2	9	5
overcast	4	0	mild	4	2	normal	6	1	true	3	3		
rainy	3	2	cool	3	1								
sunny	2/9	3/5	hot	2/9	2/5	high	3/9	4/5	false	6/9	2/5	9/14	5/14
overcast	4/9	0/5	mild	4/9	2/5	normal	6/9	1/5	true	3/9	3/5		
rainy	3/9	2/5	cool	3/9	1/5								

Supponiamo ora di voler classificare una nuova istanza:

Condizioni	Temp.	Umidità	Vento	Si_gioca
soleggiato	fredda	alta	vero	?

### Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

### Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

### Modelli principali

Supervised Learning

Unsupervised Learning

### Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

### Combinazione di più modelli

Bagging

Randomization

Boosting

### Applicazioni

Performance agenti

Estensioni

### Riferimenti

Come classificare la nuova istanza?

- ▶ otteniamo la “tendenza” a giocare moltiplicando le frazioni corrispondenti:

$$(2/9) * (3/9) * (3/9) * (3/9) * (9/14) = 0.0053$$

- ▶ allo stesso modo otteniamo la “tendenza” a non giocare:

$$(3/5) * (1/5) * (4/5) * (3/5) * (5/14) = 0.0206$$

- ▶ dunque, la tendenza a non giocare è circa 4 volte superiore a quella di farlo;
- ▶ possiamo trasformare i numeri in probabilità:
  - ▶ a favore:  $0.0053 / (0.0053 + 0.0206) = 20.5\%$
  - ▶ contro:  $0.0206 / (0.0053 + 0.0206) = 79.5\%$

Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

Tipologie di  
Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

Modelli principali

Supervised Learning

Unsupervised Learning

Valutazione dei  
modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

Combinazione di  
più modelli

Bagging

Randomization

Boosting

Applicazioni

Performance agenti

Estensioni

Riferimenti

Metodo basato sul Teorema di Bayes:

$$Pr[H|E] = (Pr[E|H] * Pr[H]) / Pr[E]$$

Nel nostro caso:

- ▶  $H$ : corrisponde al fatto di giocare o meno;
- ▶  $E$ : corrisponde alla combinazione dei valori delle 4 feature  $E_1, E_2, E_3, E_4$

Sostituendo nella formula otteniamo, facendo uso dell'ipotesi di indipendenza delle variabili:

$$\begin{aligned} & Pr[s_i|E] \\ &= (Pr[E_1|s_i] * Pr[E_2|s_i] * Pr[E_3|s_i] * Pr[E_4|s_i] * Pr[s_i]) / Pr[E] \\ &= [(2/9) * (3/9) * (3/9) * (3/9) * (9/14)] / Pr[E] \end{aligned}$$

Il denominatore scompare effettuando la normalizzazione dei valori, ottenendo lo stesso risultato di prima.

Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

Tipologie di  
Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

Modelli principali

Supervised Learning

Unsupervised Learning

Valutazione dei  
modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

Combinazione di  
più modelli

Bagging

Randomization

Boosting

Applicazioni

Performance agenti

Estensioni

Riferimenti

Principale metodologia di *Unsupervised Learning*:

- ▶ Cerchiamo di raggruppare fra loro istanze simili, senza considerare un determinato attributo come obiettivo;
- ▶ i gruppi individuati possono essere:
  - ▶ *esclusivi (hard clustering)*
  - ▶ *con overlap (fuzzy clustering)*
  - ▶ *probabilistici (fuzzy clustering)*
  - ▶ *gerarchici*
- ▶ il processo di clustering può essere seguito da uno di classificazione:
  - ▶ come “conferma” dei cluster;
  - ▶ per assegnare ai gruppi nuove istanze.

Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

Tipologie di  
Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

Modelli principali

Supervised Learning

Unsupervised Learning

Valutazione dei  
modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

Combinazione di  
più modelli

Bagging

Randomization

Boosting

Applicazioni

Performance agenti

Estensioni

Riferimenti



Diverse famiglie di algoritmi di clustering:

- ▶ gerarchici;
- ▶ basati su centroidi: *k-means(++), k-medoids, FCM*
- ▶ basati sulla distribuzione: *Expectation-Maximization*
- ▶ basati sulla densità: *DBSCAN, OPTICS*
- ▶ ...

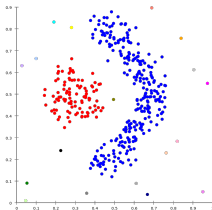
La scelta della tipologia dell'algoritmo riveste grande importanza per quanto riguarda il risultato finale, e dipende anche dalla distribuzione delle istanze.

# Clustering gerarchico

Idea - un oggetto è più simile agli oggetti ad esso vicini, rispetto a quelli lontani:

- ▶ si parte da un nr. di cluster pari al nr. delle istanze;
- ▶ i due cluster più simili fra loro vengono collegati;
- ▶ si procede iterativamente collegando di volta in volta fra loro i due cluster più simili (diverse metodologie);
- ▶ sino a collegare tutte le istanze in un unico cluster;

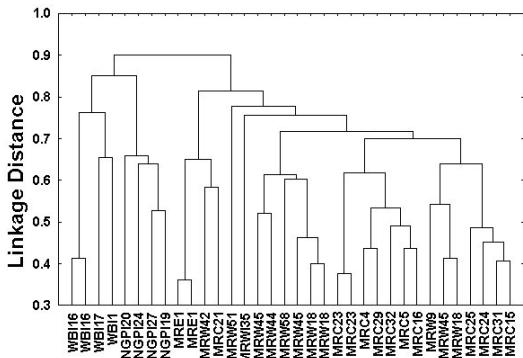
⇒ problema relativo agli outliers/rumore.



# Clustering gerarchico (2)

L'output è rappresentato da un *dendrogramma*:

- ▶ **asse x**: istanze;
- ▶ **asse y**: distanza;
- ▶ date due istanze, quanto più il loro primo punto di fusione è basso, tanto più sono simili;
- ▶ la similarità **NON** dipende dalla vicinanza lungo l'asse delle x!



## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

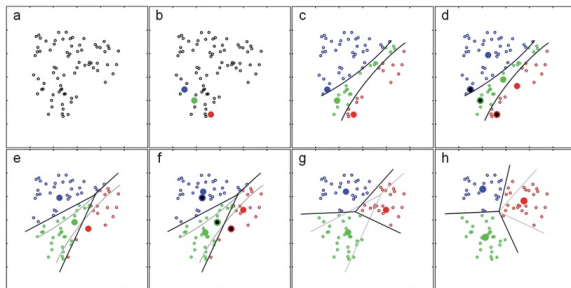
## Riferimenti

- ▶ ciascun cluster viene rappresentato da un *centroide*, elemento che può o meno appartenere al dataset iniziale;
- ▶ per gli algoritmi della famiglia *k-means*, il numero di cluster da ricercare deve essere fissato a priori;
- ▶ si cercano  $k$  centri di altrettanti cluster, e si assegnano ad essi le istanze, in modo tale da minimizzare la somma dei quadrati delle distanze;
- ▶ problema *NP-hard*, si cercano soluzioni approssimate.

Uno dei più diffusi algoritmi per il clustering *partizionale*.  
Funzionamento:

1. seleziona il numero di cluster da ricercare,  $k$ ;
2. scegli casualmente  $k$  istanze, centri di altrettanti cluster;
3. assegna tutte le istanze-non-centro al punto, fra i  $k$  centri, più vicino;
4. calcola sulle istanze di ciascun cluster la media dei diversi attributi, e poni il punto così ottenuto come nuovo centro del cluster;
5. se, alla luce dei nuovi centri, almeno un'istanza cambierebbe cluster di appartenenza, ripeti dal punto 3, altrimenti la situazione è stabile e l'algoritmo termina.

# L'algoritmo *k-means* (2)



- ▶ l'algoritmo non garantisce l'ottimalità del raggruppamento;
- ▶ la variante *k-means++* è migliore per velocità ed accuratezza;
- ▶ sono da tenere in considerazione aspetti riguardanti la standardizzazione degli attributi;
- ▶ è necessario impostare il valore di  $k$ , a differenza che in altre metodologie di clustering (es. gerarchico).

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

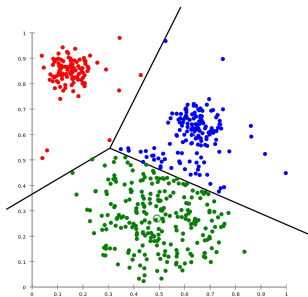
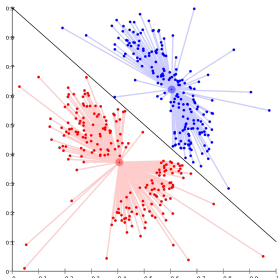
Performance agenti

Estensioni

## Riferimenti

# Casi di cattiva clusterizzazione

Esempi di cattiva clusterizzazione:



- ▶ *prima immagine*: incorretta definizione dei confini fra i cluster;
- ▶ *seconda immagine*: impossibilità di catturare la forma dei due cluster (densità);

## Generalizzazione dell'approccio *k-means*:

- ▶ ciascun cluster è rappresentato da una diversa distribuzione di probabilità (consideriamo Gaussiana), ognuna delle quali descrive la distribuzione dei valori per i membri di quel cluster;
- ▶ ogni distribuzione fornisce la probabilità che una istanza abbia certi valori per i suoi attributi, supponendo che sia noto a quale cluster appartiene;
- ▶ dunque, l'ipotesi di fondo è che le istanze del dataset siano generabili a partire da un mix di modelli probabilistici (*mixture*).

### Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

### Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

### Modelli principali

Supervised Learning  
Unsupervised Learning

### Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

### Combinazione di più modelli

Bagging  
Randomization  
Boosting

### Applicazioni

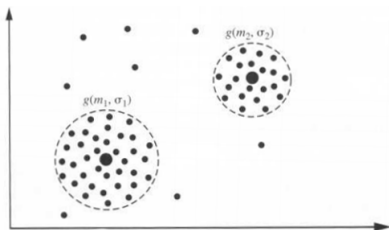
Performance agenti  
Estensioni

### Riferimenti



# Clustering basato sulla distribuzione (2)

Esempio di un mixture model. In esso sono presenti due cluster, ciascuno dei quali segue una distribuzione Gaussiana con la sua media e la sua deviazione standard:

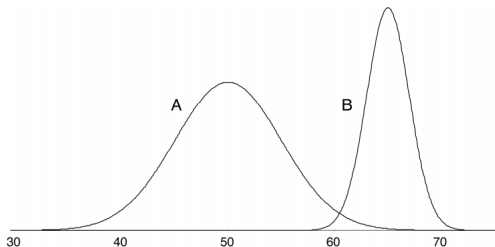


↪ L'obiettivo è, partendo dall'insieme di istanze, trovare le distribuzioni corrispondenti ai cluster, più le probabilità di appartenenza delle istanze ad essi (modello).

# Clustering basato sulla distribuzione (3)

Esempio con singola variabile numerica e due distribuzioni gaussiane. Dati etichettati dalle “classi” reali (per comodità, sopra) e modello (sotto):

A	51	B	62	B	64	A	48	A	39	A	51
A	43	A	47	A	51	B	64	B	62	A	48
B	62	A	52	A	52	A	51	B	64	B	64
B	64	B	64	B	62	B	63	A	52	A	42
A	45	A	51	A	49	A	43	B	63	A	48
A	42	B	65	A	48	B	65	B	64	A	41
A	46	A	48	B	62	B	66	A	48		
A	45	A	49	A	43	B	65	B	64		
A	45	A	46	A	40	A	46	A	48		
model											



# Clustering basato sulla distribuzione (4)

- ▶ il modello è dato da due cluster  $A, B$  con medie  $\mu_A = 50, \mu_B = 65$  e deviazioni standard  $\sigma_A = 5, \sigma_B = 2$  (le due *mixture* sono qui formate da un'unica gaussiana);
- ▶ le istanze appartengono ad  $A$  con probabilità  $p_A = 0.6$  ed a  $B$  con probabilità  $p_B = 1 - p_A$ .

⇒ date le istanze (senza conoscenza di  $A$  e  $B$ ), vogliamo trovare i 5 parametri:  $\mu_A, \mu_B, \sigma_A, \sigma_B, p_A$ .

Problema: non conosciamo né i 5 parametri, né l'effettiva appartenenza delle istanze ai cluster.

- ▶ *EM* è un algoritmo di raffinamento iterativo che può essere usato per trovare le stime dei parametri;
- ▶ può essere visto come un'estensione di *k – means* operante *fuzzy clustering*;
- ▶ esistono varianti in cui il numero di cluster viene determinato automaticamente;
- ▶ non viene garantita l'ottimalità del risultato.

Funzionamento:

1. imposta casualmente i valori per  $\mu_A, \mu_B, \sigma_A, \sigma_B, p_A$ ;

2. itera:

**2.1 Expectation step** - calcola la (densità di) probabilità di appartenenza ai cluster  $A, B$  per ogni istanza  $x_i$ , cioè  $Pr(x_i \in A)$  e  $Pr(x_i \in B)$ , a partire dai parametri (assegnamento “soft” delle istanze ai cluster);

**2.2 Maximization step** - stima, in maniera pesata in base alle probabilità di appartenenza delle istanze ai cluster, i parametri.

⇒ dunque, ad ogni passo *E-M* riassegna gli oggetti tenendo conto del modello dato dai parametri; gli oggetti assegnati vengono quindi usati per generare nuove stime dei parametri.

## Algoritmo E-M (3): dettagli

Possiamo calcolare  $Pr(x_i \in A)$  a partire dai parametri:

$$Pr(x_i \in A) = Pr(A|x_i) = \frac{Pr(x_i|A) * p_A}{Pr(x_i)}$$

dove:

$$Pr(x_i|A) = \frac{1}{\sigma_A \sqrt{2\pi}} e^{-\frac{(x_i - \mu_A)^2}{2\sigma_A^2}}$$

ossia la funzione di densità di probabilità nel caso della distribuzione normale.

Non conosciamo  $Pr(x_i)$  (*probabilità di un'istanza dati i cluster*); tuttavia,  $Pr(A|x_i) + Pr(B|x_i) = 1$ , dunque:

$$\frac{Pr(x_i|A) * p_A + Pr(x_i|B) * p_B}{Pr(x_i)} = 1$$

# Algoritmo E-M (4): dettagli

I parametri possono essere invece ricalcolati a partire dalle (densità di) probabilità come segue:

$$\mu_a = \frac{\sum_i Pr(x_i \in A) * x_i}{\sum_i Pr(x_i \in A)}$$

$$\sigma_a = \sqrt{\frac{\sum_i Pr(x_i \in A) * (x_i - \mu_a)^2}{\sum_i Pr(x_i \in A)}}$$

$$p_A = \frac{\sum_i Pr(x_i \in A)}{\text{numero\_totale\_istanze}}$$

## Quando terminare?

- ▶ *EM* converge verso un punto fisso;
- ▶ concetto di verosimiglianza globale (*likelihood*), calcolata ad ogni iterazione:
  - ▶ misura della bontà del clustering;
  - ▶ aumenta ad ogni iterazione, fino ad un massimo locale;
  - ▶ intuitivamente, quanto è verosimile che il modello descriva/generi il dataset originario.
- ▶ l'algoritmo si arresta quando la *likelihood* rimane invariata (o miglioramento trascurabile).

## Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

## Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning  
Unsupervised Learning

## Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

## Combinazione di più modelli

Bagging  
Randomization  
Boosting

## Applicazioni

Performance agenti  
Estensioni

## Riferimenti



## Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

## Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning  
Unsupervised Learning

## Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

## Combinazione di più modelli

Bagging  
Randomization  
Boosting

## Applicazioni

Performance agenti  
Estensioni

## Riferimenti

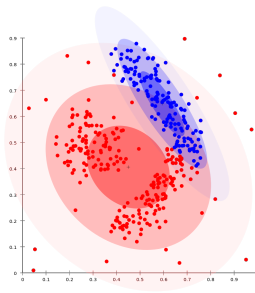
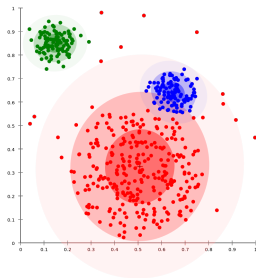
Come calcolare la *likelihood*?

$$\mathcal{L} = \prod_i Pr(x_i) = \prod_i (Pr(x_i|A) * p_A + Pr(x_i|B) * p_B)$$

In pratica si calcola il logaritmo della verosimiglianza.

↪ la *likelihood* può essere anche utilizzata per  
confrontare la bontà di diversi risultati di clustering.

## Esempi di clusterizzazione con *EM*:



- ▶ *prima immagine*: buona clusterizzazione su dati con distribuzione gaussiana;
- ▶ *seconda immagine*: impossibilità di catturare la forma dei due cluster (densità).

### Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

### Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

### Modelli principali

Supervised Learning  
Unsupervised Learning

### Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

### Combinazione di più modelli

Bagging  
Randomization  
Boosting

### Applicazioni

Performance agenti  
Estensioni

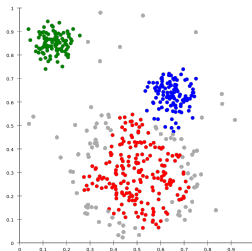
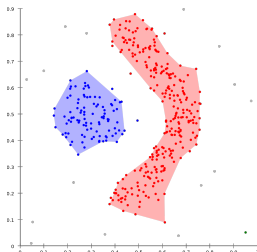
### Riferimenti

# Clustering basato sulla densità

- ▶ Un cluster è inteso come una regione densa di punti, separata da regioni a bassa densità dalle altre regioni a elevata densità;
- ▶ i punti nelle regioni a bassa densità sono considerati rumore o punti di confine.

Due algoritmi principali:

- ▶ *DBSCAN*: assume cluster aventi densità simile;
- ▶ *OPTICS*: variante in grado di trattare densità diverse.



Simili alle classification rules, con le differenze:

- ▶ si vogliono mettere in luce generiche relazioni fra gli attributi;
- ▶ la loro parte destra può fare riferimento ad uno o più attributi;
- ▶ ciascuna regola opera in maniera indipendente.

*Temperatura = fredda*

→ *Umidita = normale;*

*Umidita = normale*  $\wedge$  *Vento = false*

→ *Si\_gioca = si;*

*Condizioni = soleggiato*  $\wedge$  *Si\_gioca = no*

→ *Umidita = alta;*

*Vento = false*  $\wedge$  *Si\_gioca = no*

→ *Condizioni = soleggiato*  $\wedge$  *Umidita = alta;*

La valutazione dei modelli ha un'importanza fondamentale nel processo di Data Mining:

- ▶ *supervised learning*: dato l'attributo obiettivo, è possibile valutare l'accuratezza delle predizioni, ad esempio:
  - ▶ numero di predizioni corrette nel caso di classificazione;
  - ▶ MSE o RMSE nel caso di regressione.
- ▶ *unsupervised learning*: non essendoci un attributo obiettivo, la validazione è principalmente affidata ad un esperto del dominio (significatività del risultato);
- ▶ dimensione del modello, tempo di calcolo, ...

⇒ concentriamoci sul caso del *supervised learning*.

# Errore *in sample* VS errore *out of sample*

Nel valutare i modelli, dobbiamo tenere conto di due fondamentali categorie d'errore:

- ▶ **in sample:** errore che si ottiene applicando il modello sullo stesso dataset sul quale è stato costruito (detto anche errore di risostituzione);
- ▶ **out of sample:** errore che si ottiene applicando il modello su un diverso (ed indipendente) dataset, rispetto a quello su cui è stato costruito.

L'errore *in sample* sarà sempre ottimistico e  $\leq$  rispetto a quello *out of sample*, in quanto gli algoritmi tendono ad effettuare un “tuning” sul dataset di training (**overfitting**, caratteristica indesiderata).

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

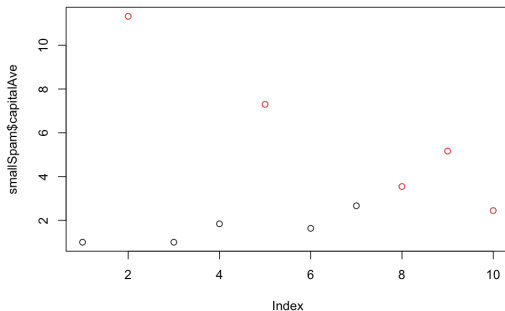
Performance agenti

Estensioni

## Riferimenti

# SPAM vs HAM e overfitting

Numero di lettere maiuscole in 10 esemplari del dataset  
(rosso=SPAM, nero=HAM):



## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

### Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

# SPAM vs HAM e overfitting (2)

Modello di predizione con accuratezza del 100% sul dataset ristretto:

*CapitalAve* > 2.7 → spam

*CapitalAve* < 2.4 → nonspam

*CapitalAve* between 2.4 and 2.45 → spam

*CapitalAve* between 2.45 and 2.7 → nonspam

Performance sull'intero dataset:

<b>Predizione/Classe</b>	<b>nonspam</b>	<b>spam</b>
nonspam	2141	588
spam	647	1225

⇒ 588 + 647 = 1235 errori di classificazione



# SPAM vs HAM e overfitting (3)

Modello di predizione alternativo (e più semplice):

$$\textit{CapitalAve} > 2.4 \rightarrow \textit{spam}$$

$$\textit{CapitalAve} \leq 2.4 \rightarrow \textit{nonspam}$$

Performance sull'intero dataset:

<b>Predizione/Classe</b>	<b>nonspam</b>	<b>spam</b>
nonspam	2224	642
spam	564	1171

$$\rightsquigarrow 642 + 564 = 1206 \text{ errori di classificazione}$$

Ciò è dovuto al fenomeno dell'*overfitting*:

- ▶ il modello è “tarato” troppo specificamente sul dataset di training.

In particolare, all'interno dei dataset distinguiamo:

- ▶ **segnale**: la parte che vogliamo utilizzare per addestrare un modello per effettuare predizioni;
- ▶ **rumore** variazioni random nel dataset, inutili se non dannose per i nostri fini.

Il goal principale in fase di training è considerare il *segnale* escludendo, per quanto possibile, il *rumore*.

Per la regressione è possibile utilizzare misure quali **MSE** o **RMSE**.

Nel caso della classificazione:

- ▶ problema di classificazione binario, ad esempio stabilire gli individui affetti da una patologia;
- ▶ insieme  $I$  di istanze;
- ▶ ciascuna istanza mappata nell'insieme  $\{p, n\}$  di risultati positivi e negativi (reali);
- ▶ il modello di classificazione permette di ottenere un mapping fra  $I$  e l'insieme  $\{Y, N\}$  delle classi previste.

# Metriche per il calcolo dell'errore (2)

Dato un classificatore ed un'istanza, vi sono 4 possibili tipologie di risultato:

		<u>True class</u>			
		<b>p</b>	<b>n</b>		
<u>Hypothesized class</u>	<b>Y</b>	True Positives	False Positives	$fp\ rate = \frac{FP}{N}$	$tp\ rate = \frac{TP}{P}$
	<b>N</b>	False Negatives	True Negatives	$precision = \frac{TP}{TP+FP}$	$recall = \frac{TP}{P}$
Column totals:		<b>P</b>	<b>N</b>	$accuracy = \frac{TP+TN}{P+N}$	

specificity =  $TN / N = 1 - FP\ Rate$

La *confusion matrix* (tabella di contingenza) si estende al caso della classificazione multiclasse.

Definito il concetto di errore, vediamo come calcolarlo il più accuratamente possibile.

Suddividiamo i dati a disposizione in 2 (o 3) sottoinsiemi indipendenti:

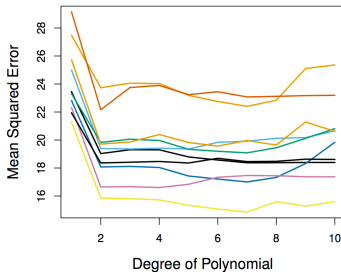
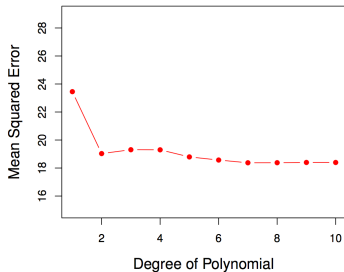
- ▶ **training**: addestramento del modello;
- ▶ **test**: singola prova (errore OOS), o tuning del modello se vi è l'insieme di *validazione*;
- ▶ **validazione** (opzionale): singola prova del modello per stimare l'errore OOS.

- ▶ ciascun insieme dovrebbe rispecchiare la frequenza delle classi tipica dell'intera popolazione (*stratificazione*);
- ▶ ad esempio, generazione con selezione casuale senza reinserimento;
- ▶ il minimo numero necessario di istanze varia da problema a problema;
- ▶ empiricamente, a seconda della mole di dati a disposizione, si possono stabilire i seguenti rapporti fra gli insiemi:
  - ▶ grande: 60% training, 20% test, 20% validazione
  - ▶ media: 60% training, 40% test
  - ▶ piccola: utilizzo di *cross-validazione* o *bootstrap*

# Generazione degli insiemi (2)

Al decrescere della mole di dati a disposizione riveste sempre maggiore importanza la selezione delle istanze per ciascun insieme, ai fini dell'addestramento del modello e della stima dell'errore out of sample.

Esempio con dataset di 392 elementi, diviso in due sottoinsiemi (training+test):



Metodologia per stimare l'errore out of sample quando non è disponibile un vero e proprio insieme di test.

*K-fold cross-validation:*

1. si parte da un singolo dataset;
2. suddividi casualmente il dataset in  $k$  partizioni;
3. ripeti per  $k$  volte i seguenti passi:
  - 3.1 seleziona, a rotazione, una delle  $k$  partizioni come insieme di test, tenendo le altre  $k - 1$  come insieme di training;
  - 3.2 genera un modello utilizzando l'insieme di training;
  - 3.3 valuta il modello utilizzando l'insieme di test;
  - 3.4 registra la stima dell'errore calcolata;
4. calcola la media delle  $k$  stime d'errore;
- V. costruisci un modello utilizzando l'**intero dataset**;
- VI. restituisci il modello generato al passo precedente, assieme alla stima dell'errore.



# Cross-validazione (2)

- ▶ in generale, eseguendo più volte la cross-validazione, si ottengono stime diverse dell'errore;
- ▶ tipicamente, utilizzare  $k = 10$  dà buoni risultati;
- ▶ se si desiderano risultati estremamente affidabili, è possibile ripetere la *10-fold cross-validation* per 10 volte.

Oltre alla *k-fold*, esistono altre tipologie di cross validazione:

- ▶ **random subsampling**: per  $n$  volte, seleziona casualmente senza reinserimento un insieme di istanze di training, e con le rimanenti forma l'insieme di test;
- ▶ **leave one out**: per  $|dataset|$  volte, seleziona a rotazione un'istanza che verrà utilizzata per il test, utilizzando le altre  $1 - |dataset|$  istanze per il training.

# Bootstrap (0.632 Bootstrap)

Utile per la stima dell'errore nel caso di dataset molto piccoli, si basa sulla selezione **con reinserimento** delle istanze.

- ▶ è dato un dataset con  $n$  istanze;
- ▶ si selezionano con reinserimento da esso  $n$  istanze, ottenendo un multinsieme;
- ▶ il multinsieme di istanze selezionate costituirà il *training set*;
- ▶ le istanze mai selezionate (ciascuna con probabilità  $(1 - 1/n)^n$ ) costituiranno l'insieme di test.

Si osservi che  $(1 - 1/n)^n \approx e^{-1} \approx 0.368$ . Dunque, per dataset ragionevolmente grandi, l'insieme di training sarà composto da circa il 63.2% delle istanze (contate in modo distinto).

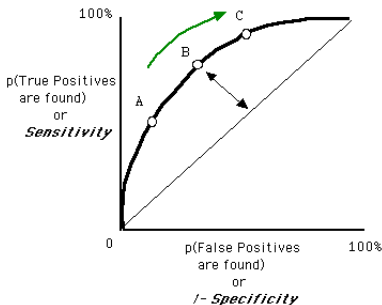
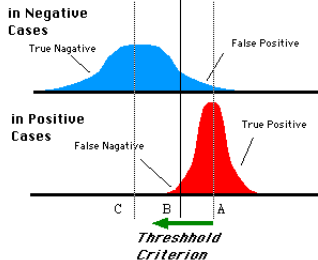
- ▶ intuitivamente, utilizzare una sottoparte del dataset originario per la generazione del modello porta ad una **sovrastima** dell'errore *out of sample*, rispetto al modello finale;
- ▶ tale sovrastima viene compensata attraverso l'errore di sostituzione (*in-sample*):

$$err_{stimato} = 0.632 * err_{out\_of\_sample} + 0.368 * err_{in\_sample}$$

- ▶ in situazioni di forte overfitting, l'intero addendo destro della formula tende a scomparire, portando ad una sottostima dell'errore;
- ▶ la variante *0.632+ Bootstrap* aumenta, al crescere dell'overfitting, il peso dato all'errore *out of sample*, diminuendo l'altro.

- ▶ acronimo per: *Receiver Operating Characteristic*;
- ▶ utilizzate per la prima volta durante la seconda guerra mondiale, con l'obiettivo di individuare i nemici utilizzando il radar;

Distributions of the Observed signal strength



## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

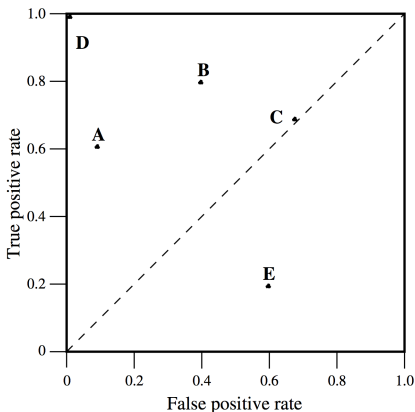
Estensioni

## Riferimenti

- ▶ intuitivamente la curva ROC mostra l'andamento del tradeoff fra benefici (*TP Rate*) e costi (*FP Rate*);
- ▶ esso riveste particolare importanza nei casi in cui gli errori commessi da un algoritmo di classificazione possono avere gravità diverse;
- ▶ le curve ROC sono inoltre utili per confrontare le prestazioni di diversi classificatori;
- ▶ osserviamo che, essendo basate su *FP Rate* e *TP Rate*, esse sono insensibili alla distribuzione dei valori della classe su cui si vuole fare predizione.
- ▶ trattiamo le curve ROC nel caso di classificatori binari, ma esistono tecniche per la loro applicazione al caso multiclasse.

# Curve ROC e classificatori discreti

Un classificatore discreto produce una singola confusion matrix, dunque un'unica coppia (*FP Rate*, *TP Rate*).



⇒ Casi: esame medico VS concessione di un prestito

- ▶ L'output di alcuni algoritmi di classificazione non è costituito da una risposta "unica", ma dalla "probabilità" che l'istanza classificata appartenga ad una determinata classe;
- ▶ stabilendo una soglia di cutoff, si può dunque decidere di assegnare tutti i valori sopra una certa soglia ad una classe piuttosto che all'altra;
- ▶ al variare della soglia di cutoff, cambierà l'output fornito dall'algoritmo, ed in particolare i suoi valori di TP e FP Rate;
- ▶ dunque, variando la threshold è possibile disegnare la curva dell'algoritmo nello spazio ROC.

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

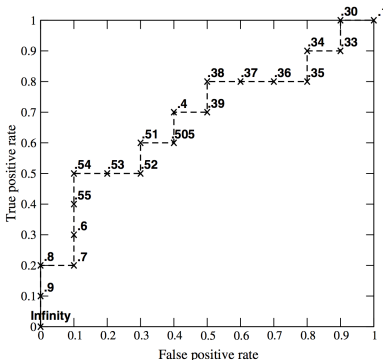
## Riferimenti

# Curve ROC e classificatori continui (2)

Introduzione al  
Data Mining

Andrea Brunello

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



Il punto alle coordinate (0.1, 0.5) produce la più alta accuratezza (70%, 14 su 20 istanze sono correttamente classificate).

## Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

## Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning  
Unsupervised Learning

## Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione

## Curve ROC

## Combinazione di più modelli

Bagging  
Randomization  
Boosting

## Applicazioni

Performance agenti  
Estensioni

## Riferimenti

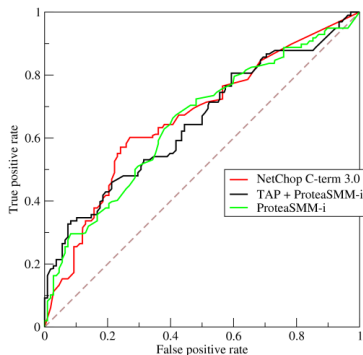


# Curve ROC e classificatori continui (3)

Introduzione al  
Data Mining

Andrea Brunello

Esempio di confronto fra diversi classificatori utilizzando le curve ROC:



Tipicamente si valuta il valore dell'area sotto la curva ( $> 0.8 \approx$  buono).

Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

Tipologie di  
Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

Modelli principali

Supervised Learning

Unsupervised Learning

Valutazione dei  
modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

Combinazione di  
più modelli

Bagging

Randomization

Boosting

Applicazioni

Performance agenti

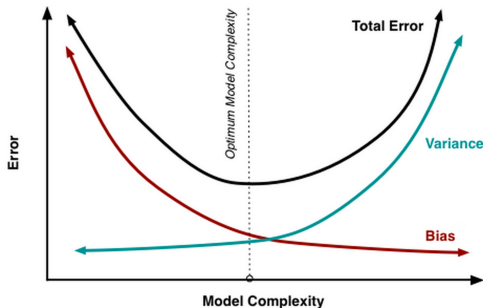
Estensioni

Riferimenti

# Bias-Variance Trade-Off

Quanto è possibile ridurre l'errore del modello?

- ▶ Due componenti intuitive d'errore:
  - ▶ **bias**: parte fissa e non eliminabile dell'errore, dovuta ad esempio alla capacità del modello di “matchare” il problema.
  - ▶ **variance**: parte variabile dell'errore, dipende dal particolare training set usato per costruire il modello.



## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

Oltre a quanto presentato, possiamo considerare:

- ▶ grandezza del modello;
- ▶ interpretabilità del modello;
- ▶ tempo di costruzione del modello;
- ▶ altre tipologie di curve, oltre alla ROC;
- ▶ ...

- ▶ Spesso il combinare più modelli porta ad un significativo aumento delle performance di classificazione;
- ▶ il prezzo da pagare riguarda la difficile interpretabilità del “supermodello” ottenuto;
- ▶ come combinare il risultato di più modelli?
  - ▶ nel caso di output qualitativo, es. votazione
  - ▶ nel caso di output quantitativo, es. media

Idea che sfrutta l'instabilità del metodo di costruzione del modello (es. alberi di decisione):

- ▶ supponiamo di avere un insieme di dataset, tutti della stessa dimensione, rappresentativi per il problema che si vuole affrontare;
- ▶ costruiamo un albero di decisione a partire da ciascuno di essi;
- ▶ in generale, ciascun albero sarà diverso, e classificherà in maniera corretta alcune istanze;
- ▶ si combinano i risultati per fornire l'output finale.

Risultati tendenzialmente migliori rispetto all'uso di un unico classificatore.

↪ riduce la *variance*, errore dovuto al particolare training set utilizzato; aiuta ad evitare l'*overfitting*

Problema: ottenere molti dataset di training distinti può essere difficile. La soluzione è una “approssimazione” dell’idea:

- ▶ partiamo da un unico insieme di  $n$  esemplari;
- ▶ generiamo multinsiemi di cardinalità  $n$ , selezionando casualmente con reinserimento;
- ▶ in generale è possibile costruire un qualsiasi modello su essi, il cui processo di training sia instabile;
- ▶ Bagging  $\approx$  Bootstrap Aggregating.

Il bagging costruisce diverse varianti di un modello introducendo casualità nel processo di generazione dell'input.

- ▶ è tuttavia applicabile solo a metodologie di training instabili.

Idea → introdurre casualità nel processo di generazione del modello:

- ▶ es. negli alberi di ricerca, ad ogni nodo scegliere il miglior attributo su cui effettuare lo split da un sottoinsieme casuale degli attributi.

Importante determinare il giusto *trade-off* fra varietà dei modelli generati ed accuratezza del singolo.

Due implementazioni largamente utilizzate:

- ▶ **Random Forest:** stessa procedura del bagging, con la differenza che per la generazione di ciascun albero viene utilizzato l'approccio randomization.
- ▶ **Rotation Forest:** miglioramento dell'approccio *Random Forest*, utilizza *random subspaces* (applicazione di randomization allo instance-based learning), *PCA* e *bagging*.



## Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

## Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning  
Unsupervised Learning

## Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

## Combinazione di più modelli

Bagging  
Randomization  
**Boosting**

## Applicazioni

Performance agenti  
Estensioni

## Riferimenti

Si propone di migliorare la tecnica di bagging:

- ▶ nel bagging i singoli modelli componenti vengono generati separatamente;
- ▶ nel boosting la costruzione di un nuovo modello è influenzata dai precedenti;
  - ▶ intuitivamente si cerca di generare un insieme di modelli complementari
- ▶ inoltre, nel boosting in sede di voto si dà maggior peso al risultato fornito dai modelli ritenuti più “affidabili”.

Implementazione ampiamente usata: **AdaBoost**  
(Adaptive Boosting).

Sperimentalmente si osserva che:

- ▶ boosting tende a produrre classificatori più performanti rispetto a bagging;
- ▶ a differenza che nel bagging, c'è però un rischio di fallimento su casi reali, tendenzialmente indice di *overfitting*.

- ▶ grande mole di dati a disposizione, riguardante il lavoro degli agenti (*call center representatives*) e le sessioni (telefoniche) da essi gestite;
- ▶ diversi obiettivi:
  - ▶ previsione dell'esito di una sessione;
  - ▶ previsione della presa in carico o meno di una telefonata da parte di un'agente;
  - ▶ valutazione delle performance lavorative degli agenti;
- ▶ scenario ideale per l'applicazione delle tecniche di Data Mining:
  - ▶ attribute selection;
  - ▶ clusterizzazione;
  - ▶ classificazione.

# Funzione di calcolo performance agenti

Idea → strumento per la valutazione automatica degli agenti.

Funzione analitica che considera diversi parametri:

- ▶ dati tecnici:
  - ▶ tempo di conversazione medio;
  - ▶ tempo di postcall medio;
- ▶ dati di servizio:
  - ▶ INB, note redatte su totale sessioni;
  - ▶ INB, nominativi registrati su totale sessioni;
  - ▶ INB, indirizzi registrati su totale sessioni;
  - ▶ INB, numeri di telefono registrati su totale sessioni;
  - ▶ INB, esiti != da nullo su totale sessioni;
  - ▶ INB, scopi != da sconosciuto su totale sessioni;
  - ▶ OUTB, *Redemption* (positivi su lavorabili);
  - ▶ OUTB, *DB\_Burn* (negativi su chiusi);
  - ▶ OUTB, resa oraria (positivi su tempo di lavoro).

# Funzione di calcolo performance agenti (2)

- ▶ la performance è un valore compreso fra 0 e 1;
- ▶ ciascun agente parte da una valutazione di 0.5;
- ▶ si aggiunge/toglie punteggio a seconda dell'abilità mostrata dall'agente secondo ciascun parametro (risp. mediana della commessa sul lungo periodo);
- ▶ otteniamo un ranking degli agenti.

## Estensioni:

- ▶ valutazione delle note redatte;
- ▶ indice ICC e bonus;
- ▶ rilevamento operatori "anomali";
- ▶ in futuro:
  - ▶ modellazione su base statistica;
  - ▶ valutazione della conversazione.

# Funzione di calcolo performance agenti (3)

## Interfaccia grafica (Alberto Dal Santo):

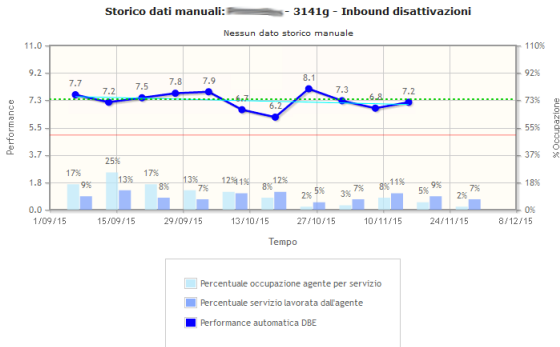
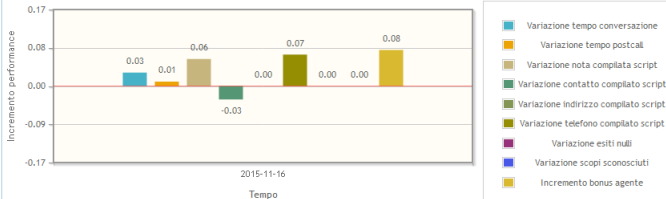


Grafico dettaglio componenti performance del: 2015-11-16



## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

# Classificazione delle note degli agenti

Metriche calcolate per ogni nota (script  $R$ ):

- ▶ numero di parole e di caratteri;
- ▶ articoli su parole;
- ▶ congiunzioni su parole;
- ▶ verbi su parole;
- ▶ aggettivi su parole;
- ▶ avverbi su parole;
- ▶ preposizioni su parole;
- ▶ quantificatori su parole;
- ▶ sostantivi su parole;
- ▶ pronomi su parole;
- ▶ codici numerici su parole;
- ▶ indice *Gulpease* di leggibilità;
- ▶ termini trovati nel dizionario italiano sul totale atteso;
- ▶ termini abbreviati su parole;
- ▶ termini non riconosciuti su parole.

# Classificazione delle note degli agenti (2)

- ▶ clusterizzazione su 1000 note selezionate casualmente (*k-means++*, in seguito *E-M*);
- ▶ individuazione di 5 cluster:
  - ▶ note composte da molte abbreviazioni (20%);
  - ▶ note articolate, ben scritte (26%);
  - ▶ note non articolate, ma comunque ben scritte e comprensibili (20%);
  - ▶ note con molti termini non riconosciuti, es. codici di dominio od errori di battitura (23%);
  - ▶ note “ibride” (11%).
- ▶ arricchimento del dataset con etichetta del cluster, e raffinamento degli assegnamenti ai cluster.

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti



# Classificazione delle note degli agenti (3)

Sul dataset generato effettuiamo classificazione:

- ▶ generazione e tuning di un albero di decisione (*J48*), obiettivo il cluster (94.5% accuratezza);
- ▶ albero implementato come funzione SQL per classificare note nuove o non considerate nella fase di clustering.

```
riconosciuti_abbr_su_parole <= 0.666667
| riconosciuti_dict_su_parole <= 0.666667
| | non_riconosciuti_su_parole <= 0.0625: c_ibride
| | non_riconosciuti_su_parole > 0.0625
| | | preposizioni_su_parole <= 0.05
| | | congiunzioni_su_parole <= 0.083333: c_nonsense
| | | congiunzioni_su_parole > 0.083333
| | | | riconosciuti_abbr_su_parole <= 0.083333: c_articolate
| | | | riconosciuti_abbr_su_parole > 0.083333: c_ibride
| | | preposizioni_su_parole > 0.05
| | | non_riconosciuti_su_parole <= 0.545455: c_articolate
| | | non_riconosciuti_su_parole > 0.545455: c_nonsense
| | riconosciuti_dict_su_parole > 0.666667
| | | n_parole <= 3
| | | preposizioni_su_parole <= 0.166667: c_brevi_ok
| | | preposizioni_su_parole > 0.166667: c_articolate
| | | n_parole > 3
| | | | aggettivi_su_parole <= 0.230769: c_articolate
| | | | aggettivi_su_parole > 0.230769
| | | | n_parole <= 5: c_brevi_ok
| | | | n_parole > 5: c_articolate
riconosciuti_abbr_su_parole > 0.666667: c_abbreviate
```

# Classificazione delle note degli agenti (4)

Come utilizzare la conoscenza del cluster di appartenenza delle note per valutare gli operatori?

Idea:

- ▶ stabilire le occorrenze dei diversi cluster per le note di ciascuna commessa;
- ▶ confrontare con le occorrenze dei cluster per le note scritte dall'agente.

Inoltre:

- ▶ il cluster delle note abbreviate ha una connotazione negativa;
- ▶ il cluster delle note articolate è concettualmente simile a quello delle non articolate;
- ▶ il cluster delle note "nonsense" è categoria a parte, diversa dagli ultimi due.

# Classificazione delle note degli agenti (5)

Introduzione al  
Data Mining

Andrea Brunello

## Esempio:

	valore text	classificazione text
1	info pagam boll	cluster abbreviate
2	disatt serv	cluster abbreviate
3	pagam boll	cluster abbreviate
4	Chiede info su pagamenti da effettuare a seguito rate e solleciti	cluster articolate
5	dopo aver scelto i posti e aver messo "si" sul web checkin dà errore	cluster articolate
6	chiede se abbiamo ricevuto la mail con il contratto di locazione.	cluster articolate
7	chiede rimborso	cluster brevi ok
8	pronto intervento	cluster brevi ok
9	DISATTIVATO SERVIZIO	cluster brevi ok
10	verif bollette	cluster ibride
11	ALTRO NUMERO - info disatt	cluster ibride
12	nessun serv attivo	cluster ibride
13	Primo PNR: PJ7VY	cluster nonsense
14	disattivato best game klub	cluster nonsense
15	cheng no pnltty within 24h	cluster nonsense

## Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

## Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning  
Unsupervised Learning

## Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

## Combinazione di più modelli

Bagging  
Randomization  
Boosting

## Applicazioni

Performance agenti  
Estensioni

## Riferimenti

Si vogliono premiare gli agenti in grado di gestire più tipologie di servizio, e/o servizi complessi.

Parametri presi in considerazione:

- ▶ numero di tipologie di servizio distinte gestite dall'agente ( $\propto$ );
- ▶ impatto della commessa in esame ( $\propto$ );
- ▶ peso della commessa presa in esame ( $\propto^{-1}$ );
- ▶ peso delle altre commesse gestite dall'agente nel periodo ( $\propto$ ).

Scostamento massimo di  $\pm 0.1$  della valutazione.

Problema: l'impatto della commessa è un valore arbitrario.

→ sostituzione con indice *ICC/IO* per commessa:

► **importanza:**

- numero di agenti distinti assegnati;
- numero di sessioni su periodo di attività.

► **complessità:**

- numero di servizi e di attività;
- tempo medio di gestione (conversazione + postcall).

► **criticità:**

- presenza di lavoro notturno;
- presenza di lavoro festivo;
- varianza nel flusso di chiamate (picchi);
- sessioni in fasce critiche su totale.

Indice di ranking in/outbound ricalcolato periodicamente.

### Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

### Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

### Modelli principali

Supervised Learning

Unsupervised Learning

### Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

### Combinazione di più modelli

Bagging

Randomization

Boosting

### Applicazioni

Performance agenti

Estensioni

### Riferimenti

Dati i valori degli attributi, come giungere ad un indice di sintesi?

Idea:

- ▶ clusterizzazione delle commesse a seconda del valore degli attributi, alla ricerca di un buon raggruppamento;
- ▶ intuitivamente, se si ha successo significa che gli attributi sono rilevanti;
- ▶ costruzione di una funzione sugli attributi il cui output numerico “ricalchi” l’assegnamento ai cluster.

### Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

### Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

### Modelli principali

Supervised Learning  
Unsupervised Learning

### Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

### Combinazione di più modelli

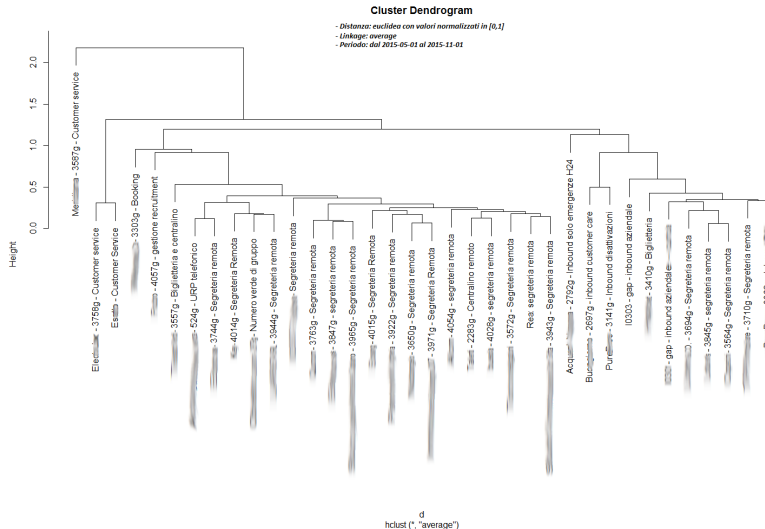
Bagging  
Randomization  
Boosting

### Applicazioni

Performance agenti  
Estensioni

### Riferimenti

## Clustering gerarchico, distanza *euclidea* e fusione *average*, esempio per le commesse inbound.



### Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

### Tipologie di Learning

Supervised vs  
Unsupervised Learning

Problemi di regressione e di  
classificazione

### Modelli principali

Supervised Learning

Unsupervised Learning

### Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

### Combinazione di più modelli

Bagging

Randomization

Boosting

### Applicazioni

Performance agenti

Estensioni

### Riferimenti

## Risultato della funzione sullo stesso periodo:

	descrizione character varying(255)	importanza numeric	complessita numeric	criticita numeric	icc numeric
1	Me - 3587g - Customer service	1.00	1.00	1.00	1.73
2	Ele - 3758g - Customer service	0.52	0.85	0.26	1.03
3	Es - Customer Service	0.48	0.82	0.22	0.97
4	Acq - 2792g - Inbound solo emergenze H24	0.27	0.55	0.73	0.95
5	Buc - 2697g - inbound customer care	0.48	0.39	0.61	0.87
6	Pur - 3141g - Inbound disattivazioni	0.56	0.10	0.64	0.86
7	I0303 - gap - inbound aziendale	0.24	0.47	0.44	0.69
8	- 3679g - Customer service	0.04	0.63	0.01	0.63
9	- 4057g - gestione recruitment	0.48	0.41	0.00	0.63
10	- 3710g - Segreteria remota	0.20	0.13	0.55	0.60
11	- 3410g - Biglietteria	0.14	0.31	0.47	0.58
12	- gap - inbound aziendale -	0.17	0.22	0.48	0.55
13	- 3694g - Segreteria remota	0.15	0.25	0.46	0.54
14	- 3368g - Inbound RAI	0.10	0.16	0.48	0.52
15	- 3303g - Booking	0.26	0.04	0.43	0.51
16	- 3552g - Biglietteria e centralino	0.20	0.22	0.21	0.40

### Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

### Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di classificazione

### Modelli principali

Supervised Learning

Unsupervised Learning

### Valutazione dei modelli

Concetto di errore

Insiemi di training, test, validazione

Curve ROC

### Combinazione di più modelli

Bagging

Randomization

Boosting

### Applicazioni

Performance agenti

Estensioni

### Riferimenti



Alcuni operatori talvolta inseriscono esiti di sessione non corrispondenti alla realtà:

- ▶ inconsciamente, a causa di errori;
- ▶ consciamente, ad esempio simulando sondaggi non realmente effettuati.

Possibilità di rilevare tali operatori, valutando:

- ▶ distribuzione degli esiti;
- ▶ tempi di dial, conversazione e postcall.

⇒ focalizzazione su singolo servizio, *Wenatex Sondaggi* (Outbound)

# Rilevamento degli operatori anomali (2)

Raggruppamento degli esiti di sessione in 5 categorie:

- ▶ *occupato o inesistente;*
- ▶ *segreteria o fax;*
- ▶ *nessuna risposta;*
- ▶ *parlato negativo;*
- ▶ *sondaggio effettuato.*

Due fronti di analisi:

- ▶ valutazione dello scostamento della distribuzione degli esiti dell'agente rispetto ai valori del servizio (in termini di media e deviazione standard);
- ▶ generazione di un albero di decisione in grado di prevedere il macroesito di ciascuna sessione sulla base dei tempi di dial, conversazione e postcall.

# Classificazione macroesito della sessione

Vogliamo generare un training set costituito da sessioni con etichettatura corretta dell'esito:

- ▶ selezione di un sottoinsieme di agenti ritenuti affidabili;
- ▶ ulteriore pulizia dalle tuple affette sicuramente da *class noise*, e non da *attribute noise*:
  - ▶ *sondaggi* con  $< 40$  sec di conversazione;
  - ▶ *segreteria o fax* senza conversazione;
  - ▶ parlato negativo con  $< 5$  sec di conversazione;
  - ▶ *occupato o inesistente* con conversazione?  $\rightsquigarrow$  **NO!**

Otteniamo un albero che cerca di prevedere il più probabile esito della sessione, dedotto sulla base dei tempi (accuratezza  $\approx 93\%$ , possibile valore più elevato ma indizi di overfitting).

# Classificazione macroesito della sessione (2)

## Albero di decisione ottenuto:

```
tempo_di_conversazione <= 7
| tempo_di_conversazione <= 0
| | tempo_di_dial <= 30
| | | tempo_di_dial <= 11: occupato_o_inesistente
| | | tempo_di_dial > 11
| | | | tempo_di_dial <= 14: occupato_o_inesistente
| | | | tempo_di_dial > 14: nessuna_risposta
| | tempo_di_dial > 30: nessuna_risposta
| tempo_di_conversazione > 0
| | tempo_di_postcall <= 1
| | | tempo_di_dial <= 29: segreteria_o_fax
| | | tempo_di_dial > 29
| | | | tempo_di_conversazione <= 1: nessuna_risposta
| | | | tempo_di_conversazione > 1: segreteria_o_fax
| | tempo_di_postcall > 1
| | | tempo_di_conversazione <= 4: segreteria_o_fax
| | | tempo_di_conversazione > 4: parlato_negativo
tempo_di_conversazione > 7
| tempo_di_conversazione <= 76
| | tempo_di_conversazione <= 11
| | | tempo_di_postcall <= 1
| | | | tempo_di_conversazione <= 9
| | | | | tempo_di_dial <= 22
| | | | | tempo_di_conversazione <= 8: segreteria_o_fax
| | | | | tempo_di_conversazione > 8: parlato_negativo
| | | | tempo_di_dial > 22: segreteria_o_fax
| | | | tempo_di_conversazione > 9: parlato_negativo
| | | tempo_di_postcall > 1: parlato_negativo
| | tempo_di_conversazione > 11: parlato_negativo
tempo_di_conversazione > 76
| tempo_di_conversazione <= 87
| | tempo_di_postcall <= 0: parlato_negativo
| | | tempo_di_postcall > 0: sondaggio_effettuato
| tempo_di_conversazione > 87: sondaggio_effettuato
```

## Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

## Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning

Unsupervised Learning

## Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

## Combinazione di più modelli

Bagging

Randomization

Boosting

## Applicazioni

Performance agenti

Estensioni

## Riferimenti

# Classificazione macroesito della sessione (3)

Calcolo, sul periodo di valutazione, per ciascun macroesito, di:

- ▶ rapporto dato dal numero delle sessioni con esito classificato in modo diverso dall'albero, su sessioni totali;
- ▶ deviazione standard (calcolata sui rapporti relativi ai diversi operatori);

Per ciascun operatore: confronto del rapporto fra il numero delle sessioni mal classificate sul totale, con i due valori calcolati relativi al servizio in generale.

# Report agenti anomali

Le informazioni vengono sintetizzate in un report:

## Report anomalie di lavorazione agenti Wenatex Sondaggi

Dal (inclusivo): 20151001 Al (esclusivo): 20151101

Il presente documento è strettamente riservato al destinatario.

La divulgazione e la riproduzione anche parziale del presente materiale è vietata.



DATI AGENTE, DIVISIONE IN GRUPPI PER 2 DI WARNING	Distribuzione macrosesti delle sessioni					Percentuale errori di classificazione macrosesti delle sessioni				
	Parlato negativo	Nessuna risposta	Segreteria/Fax	Occupato/Inesistente	Sondaggio	Parlato negativo	Nessuna risposta	Segreteria/Fax	Occupato/Inesistente	Sondaggio
Numero di warning agenti del gruppo: 5										
Nominativo: GHO # sessioni: 543	Agente: 0.25 Media: 0.49 Deviaz: 0.11	Agente: 0.48 Media: 0.24 Deviaz: 0.11	Agente: 0.12 Media: 0.13 Deviaz: 0.04	Agente: 8.13 Media: 0.08 Deviaz: 0.02	Agente: 0.03 Media: 0.07 Deviaz: 0.02	Agente: 0.04 Media: 0.04 Deviaz: 0.02	Agente: 8.89 Media: 0.38 Deviaz: 0.21	Agente: 0.12 Media: 0.28 Deviaz: 0.12	Agente: 0.13 Media: 0.13 Deviaz: 0.09	Agente: 0.03 Media: 0.05 Deviaz: 0.05
Nominativo: GIANVIA # sessioni: 5421	Agente: 0.14 Media: 0.49 Deviaz: 0.11	Agente: 0.69 Media: 0.22 Deviaz: 0.18	Agente: 0.19 Media: 0.13 Deviaz: 0.04	Agente: 0.03 Media: 0.07 Deviaz: 0.02	Agente: 0.03 Media: 0.07 Deviaz: 0.02	Agente: 0.05 Media: 0.04 Deviaz: 0.02	Agente: 8.48 Media: 0.32 Deviaz: 0.23	Agente: 0.78 Media: 0.28 Deviaz: 0.14	Agente: 0.13 Media: 0.13 Deviaz: 0.09	Agente: 0.03 Media: 0.05 Deviaz: 0.05
Numero di warning agenti del gruppo: 4										
Nominativo: SUSANNA # sessioni: 4251	Agente: 0.25 Media: 0.49 Deviaz: 0.11	Agente: 0.48 Media: 0.24 Deviaz: 0.11	Agente: 0.15 Media: 0.13 Deviaz: 0.04	Agente: 0.06 Media: 0.06 Deviaz: 0.02	Agente: 0.05 Media: 0.07 Deviaz: 0.02	Agente: 0.08 Media: 0.04 Deviaz: 0.02	Agente: 0.87 Media: 0.38 Deviaz: 0.23	Agente: 0.17 Media: 0.28 Deviaz: 0.12	Agente: 0.13 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: PNO # sessioni: 7566	Agente: 0.21 Media: 0.49 Deviaz: 0.11	Agente: 0.58 Media: 0.24 Deviaz: 0.18	Agente: 0.11 Media: 0.13 Deviaz: 0.04	Agente: 0.07 Media: 0.06 Deviaz: 0.02	Agente: 0.06 Media: 0.07 Deviaz: 0.02	Agente: 0.03 Media: 0.04 Deviaz: 0.02	Agente: 0.70 Media: 0.32 Deviaz: 0.23	Agente: 0.15 Media: 0.28 Deviaz: 0.14	Agente: 0.11 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Numero di warning agenti del gruppo: 3										
Nominativo: PERGIORGIO # sessioni: 1332	Agente: 0.22 Media: 0.49 Deviaz: 0.11	Agente: 0.58 Media: 0.24 Deviaz: 0.18	Agente: 0.09 Media: 0.13 Deviaz: 0.04	Agente: 0.04 Media: 0.06 Deviaz: 0.02	Agente: 0.06 Media: 0.07 Deviaz: 0.02	Agente: 0.03 Media: 0.04 Deviaz: 0.02	Agente: 0.78 Media: 0.32 Deviaz: 0.23	Agente: 0.26 Media: 0.28 Deviaz: 0.12	Agente: 0.13 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: MARIO # sessioni: 1275	Agente: 0.25 Media: 0.49 Deviaz: 0.11	Agente: 0.58 Media: 0.24 Deviaz: 0.18	Agente: 0.11 Media: 0.13 Deviaz: 0.04	Agente: 0.09 Media: 0.06 Deviaz: 0.02	Agente: 0.08 Media: 0.07 Deviaz: 0.02	Agente: 0.03 Media: 0.04 Deviaz: 0.02	Agente: 0.64 Media: 0.32 Deviaz: 0.23	Agente: 0.15 Media: 0.28 Deviaz: 0.12	Agente: 0.13 Media: 0.13 Deviaz: 0.09	Agente: 0.02 Media: 0.05 Deviaz: 0.05
Numero di warning agenti del gruppo: 2										
Nominativo: GIGI # sessioni: 4453	Agente: 0.58 Media: 0.49 Deviaz: 0.11	Agente: 0.06 Media: 0.24 Deviaz: 0.11	Agente: 0.26 Media: 0.13 Deviaz: 0.04	Agente: 0.05 Media: 0.06 Deviaz: 0.02	Agente: 0.07 Media: 0.07 Deviaz: 0.02	Agente: 0.02 Media: 0.04 Deviaz: 0.02	Agente: 0.05 Media: 0.32 Deviaz: 0.23	Agente: 0.68 Media: 0.28 Deviaz: 0.12	Agente: 0.08 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: DINO # sessioni: 5258	Agente: 0.36 Media: 0.49 Deviaz: 0.11	Agente: 0.26 Media: 0.24 Deviaz: 0.13	Agente: 0.10 Media: 0.13 Deviaz: 0.04	Agente: 0.07 Media: 0.06 Deviaz: 0.02	Agente: 0.08 Media: 0.07 Deviaz: 0.02	Agente: 0.05 Media: 0.04 Deviaz: 0.02	Agente: 0.68 Media: 0.32 Deviaz: 0.23	Agente: 0.15 Media: 0.28 Deviaz: 0.12	Agente: 0.14 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: FRANCO # sessioni: 3979	Agente: 0.55 Media: 0.49 Deviaz: 0.11	Agente: 0.18 Media: 0.24 Deviaz: 0.13	Agente: 0.11 Media: 0.13 Deviaz: 0.04	Agente: 0.05 Media: 0.06 Deviaz: 0.02	Agente: 0.13 Media: 0.07 Deviaz: 0.02	Agente: 0.08 Media: 0.04 Deviaz: 0.02	Agente: 0.03 Media: 0.32 Deviaz: 0.23	Agente: 0.12 Media: 0.28 Deviaz: 0.12	Agente: 0.11 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: FRANCA # sessioni: 3323	Agente: 0.36 Media: 0.49 Deviaz: 0.11	Agente: 0.48 Media: 0.24 Deviaz: 0.18	Agente: 0.06 Media: 0.13 Deviaz: 0.04	Agente: 0.06 Media: 0.06 Deviaz: 0.02	Agente: 0.03 Media: 0.07 Deviaz: 0.02	Agente: 0.05 Media: 0.04 Deviaz: 0.02	Agente: 0.67 Media: 0.32 Deviaz: 0.23	Agente: 0.20 Media: 0.28 Deviaz: 0.12	Agente: 0.17 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: LUISA # sessioni: 1382	Agente: 0.49 Media: 0.49 Deviaz: 0.11	Agente: 0.17 Media: 0.24 Deviaz: 0.13	Agente: 0.17 Media: 0.13 Deviaz: 0.04	Agente: 0.08 Media: 0.06 Deviaz: 0.02	Agente: 0.04 Media: 0.07 Deviaz: 0.02	Agente: 0.02 Media: 0.04 Deviaz: 0.02	Agente: 0.14 Media: 0.32 Deviaz: 0.23	Agente: 0.08 Media: 0.28 Deviaz: 0.12	Agente: 0.06 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: ROSALIA # sessioni: 9264	Agente: 0.48 Media: 0.49 Deviaz: 0.11	Agente: 0.18 Media: 0.24 Deviaz: 0.13	Agente: 0.04 Media: 0.13 Deviaz: 0.04	Agente: 0.07 Media: 0.06 Deviaz: 0.02	Agente: 0.06 Media: 0.07 Deviaz: 0.02	Agente: 0.04 Media: 0.04 Deviaz: 0.02	Agente: 0.08 Media: 0.32 Deviaz: 0.23	Agente: 0.11 Media: 0.28 Deviaz: 0.12	Agente: 0.11 Media: 0.13 Deviaz: 0.09	Agente: 0.05 Media: 0.05 Deviaz: 0.05
Nominativo: ALBA # sessioni: 4742	Agente: 0.48 Media: 0.49 Deviaz: 0.11	Agente: 0.23 Media: 0.24 Deviaz: 0.13	Agente: 0.10 Media: 0.13 Deviaz: 0.04	Agente: 0.12 Media: 0.06 Deviaz: 0.02	Agente: 0.06 Media: 0.07 Deviaz: 0.02	Agente: 0.03 Media: 0.04 Deviaz: 0.02	Agente: 0.23 Media: 0.32 Deviaz: 0.23	Agente: 0.27 Media: 0.28 Deviaz: 0.12	Agente: 0.17 Media: 0.13 Deviaz: 0.09	Agente: 0.04 Media: 0.05 Deviaz: 0.05

Introduzione al  
Data Mining

Andrea Brunello

Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

Tipologie di  
Learning

Supervised vs  
Unsupervised Learning

Problemi di regressione e di  
classificazione

Modelli principali

Supervised Learning

Unsupervised Learning

Valutazione dei  
modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

Combinazione di  
più modelli

Bagging

Randomization

Boosting

Applicazioni

Performance agenti

Estensioni

Riferimenti

Le diverse applicazioni hanno permesso di affrontare praticamente questioni riguardanti:

- ▶ clustering: gerarchico, *k-means++*, *EM*, sia per l'esplorazione del dominio che come base per classificazione (caso delle note);
- ▶ classificazione con *J48*, e tuning dei parametri di costruzione dell'albero (es. per prevenire overfitting o per aumentare la leggibilità);
- ▶ trattamento di *class* ed *attribute noise*;
- ▶ definizione di funzioni analitiche (caso ICC e performance agenti).

- ▶ modellazione ed inquadramento statistico della funzione performance agenti;
- ▶ proseguimento studio riguardante il flusso per presa in carico o meno di una telefonata inbound;
- ▶ aggiunta di un dizionario con termini del dominio per arricchire la valutazione delle note;
- ▶ approccio data mining per stabilire il livello di soddisfazione di un agente;
- ▶ clustering degli agenti;
- ▶ analisi del flusso audio.

### Introduzione

Cos'è il Data Mining

Il processo di Data Mining

Terminologia

### Tipologie di Learning

Supervised vs

Unsupervised Learning

Problemi di regressione e di  
classificazione

### Modelli principali

Supervised Learning

Unsupervised Learning

### Valutazione dei modelli

Concetto di errore

Insiemi di training, test,  
validazione

Curve ROC

### Combinazione di più modelli

Bagging

Randomization

Boosting

### Applicazioni

Performance agenti

Estensioni

### Riferimenti



## Introduzione

Cos'è il Data Mining  
Il processo di Data Mining  
Terminologia

## Tipologie di Learning

Supervised vs  
Unsupervised Learning  
Problemi di regressione e di  
classificazione

## Modelli principali

Supervised Learning  
Unsupervised Learning

## Valutazione dei modelli

Concetto di errore  
Insiemi di training, test,  
validazione  
Curve ROC

## Combinazione di più modelli


Bagging  
Randomization  
Boosting


## Applicazioni

Performance agenti  
Estensioni

## Riferimenti

 I. H. Witten, E. Frank, M. A. Hall: *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 3rd edition, 2011.

 G. James, D. Witten, T. Hastie, R. Tibshirani: *An Introduction to Statistical Learning with Applications in R*. Springer, 2013.

 T. Fawcett: *An introduction to ROC analysis*. Pattern Recognition Letters 27, pp. 861-874, 2006.