Appunti di Data Mining

Luciano Quercia
Simone Rutigliano

Aggiornato: 14 marzo 2014

Indice

Capitolo 1

Introduzione

1.1 Definizioni preliminari

Dato Ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione.

Informazione Il risultato di un processo di elaborazione e interpretazione di dati, che porta materia precedentemente sconosciuta al ricevente.

Conoscenza Familiarità, consapevolezza, o comprensione raggiunte mediante studio, esperienza e apprendimento.

1.2 Paradigmi

Teoria

1. Caratterizzare gli oggetti dello studio (definizioni)
2. Ipotizzare possibili relazioni fra loro (enunciazioni)
3. Determinare se le relazioni sono vere (dimostrazioni)
4. Interpretare i risultati (conclusioni)

Sperimentazione

1. Formare un'ipotesi
2. Costruire un modello e fare una previsione
3. Disegnare un esperimento e raccogliere i dati
4. Analizzare i risultati

Computational simulation

1. Astrai le componenti di un sistema complesso e le interazioni fra essi e parametrizzale
2. Descrivi le componenti e le interazioni mediante un programma
3. Genera dei dati al variare dei parametri
4. Osserva le macro proprietà e interpretale

Data discovery

1. Catturare i dati
2. Curare i dati (gestiti e manutenuti)
3. Analizzare i dati (algoritmi di DM e KD)
4. Pubblicare i risultati

Capitolo 2

KDD Process - CRISP-DM

Definizione La knowledge discovery è l'estrazione *non banale* di informazioni *implicite*, precedentemente *sconosciute* e potenzialmente *utili* dai dati.

2.1 Business Understanding

- Capire il dominio
- Fissare gli obiettivi di business
- Fissare il criterio di successo (in termini di business)
- Scegliere il task appropriato
- Definire il criterio di successo del processo di DM
- Produrre un piano di progetto

2.1.1 I task

Predizione

- Classificazione
- Regressione

Descrizione

- Regole di associazione
- Clustering
- Summarization
- Dependency modeling

- Change and deviation detection

2.2 Data Understanding

- Raccogliere i dati
- Descrivere i dati
- Verificare la qualità dei dati (accuratezza, completezza, consistenza e aggiornamento)
- Esplorare i dati (strumenti statistici)

2.3 Data Preparation

2.3.1 Campionamento

Campionamento casuale semplice Estraggo casualmente i dati dal dataset. Posso farlo con reimmissione e senza reimmissione.

Campionamento casuale stratificato Divido il dataset in strati ed estraggo casualmente i dati da ogni strato. Può garantire la rappresentazione delle minoranze.

Cluster Sampling Divido il dataset in cluster e estraggo alcuni cluster per intero.

Block Sampling Variante del Cluster Sampling utilizzata per ottimizzare le letture da disco. Leggo per intero alcuni blocchi in rappresentanza dell'intero dataset.

Two-stage Sampling Scelgo casualmente dei cluster e poi campiono all'interno di ogni cluster.

Sistematic Sampling Estraggo un elemento ogni k.

Two-phase Sampling Effettuo un primo campionamento per facilitare le decisioni sulle strategie di campionamento da compiere successivamente.

2.3.2 Feature Selection

Embedded Methods

L'algoritmo di Data Mining è in grado di ignorare gli attributi inutili.

Feature Subset Generation

Per i due successivi metodi di Feature Selection è necessario generare i vari subset da valutare. È possibile farlo:

- Enumerando tutti i possibili subset (2^N)
- Prendendo casualmente alcuni di questi subset
- Generando i subset in maniera greedy
  - Forward selection: parto da un insieme vuoto e aggiungo una feature alla volta.
  - Backward selection: parto dall'insieme pieno e rimuovo una feature alla volta.

Wrapper Models

Viene utilizzato l'algoritmo di DM per determinare il subset di feature ottimale. Sceglie il subset di feature che massimizza l'accuratezza predittiva dell'algoritmo sul training set.

Filter Models

È indipendente dall'algoritmo di DM. I diversi subset di feature sono valutati usando una qualche misura (distanza e informazione).

Information Gain Rappresenta la differenza fra l'incertezza delle classi prima e dopo aver preso in considerazione la feature X.

$$IG(X) = \sum_i U\left(P(c_i)\right) - E\left[\sum_i U\left(P(c_i|X)\right)\right]$$

dove $P(c_i)$ sono le probabilità a priori di ogni classe, $U()$ è l'incertezza. Se $U(x) = -x \log x$ allora $\sum_i U(P(c_i))$ è una misura di *entropia*.

Distance Measures Solo se si tratta di un task di classificazione, è possibile utilizzare una misura di distanza.

Dobbiamo trovare il sottoinsieme di feature che più sono in grado di aumentare la distanza fra le classi.

Kullback-Leibler divergence

$$m_{KL}(P, Q) = \sum_{v \in V} q(v) \log \frac{q(v)}{p(v)}$$

Misura la perdita di informazioni se prendiamo P invece di Q. Misura la differenza fra 2 distribuzioni di probabilità. Non è simmetrica.

χ^2-divergence

$$m_{\chi^2}(P,Q) = \sum_{y \in Y} \frac{|p(y) - q(y)|^2}{p(y)}$$

Manhattan distance

$$m_1(P,Q) = \sum_{y \in Y} |p(y) - q(y)|$$

Euclidean distance

$$m_2(P,Q) = \sum_{y \in Y} |p(y) - q(y)|^2$$

Minkowski's distance

$$m_\mu(P,Q) = \sum_{y \in Y} |p(y) - q(y)|^\mu$$

Misure di dipendenza Misura quanto una feature è associata alla classe.

Misure di inconsistenza Cercano il minimo numero di feature che mantenga la consistenza con i dati. Mirano a ottenere:

$$P(C|FullSet) = P(C|SubSet)$$

2.3.3 Pulizia dei dati

I due problemi maggiori nei dati sono:

Noisy data

Possiamo trovare errori umani o outlier. Dobbiamo decidere se eliminarli o sostituirli.

Missing values

I valori mancanti possono derivare da errori umani, da informazioni non disponibili o da un errato incrocio tra sorgenti di dati eterogenee.

Diverse tecniche:

- eliminare le istanze che contengono missing values
- eliminare l'attributo che contiene missing values
- rimpiazzare il missing value con:

  - media o mediana in caso di variabili quantitative
  - moda o valore 'unknown' in caso di variabili qualitative
  - un valore predetto da modelli predittivi

2.3.4 Costruzione dei dati

I dati sono tipicamente raffinati per venire incontro ai requisiti dell'algoritmo di DM.

Possiamo:

- convertire un attributo in un altro formato (e.g. data)
- calcolare un nuovo attributo da altri (e.g. età da data di nascita)
- aggregare i dati (e.g. media o somma di un periodo di tempo)
- normalizzare o scalare:
  - min-max: porta ad un nuovo range di valori min e max
  - z-score: utilizza media e deviazione standard
  - decimal scaling: porta al range $[-1, 1]$
- discretizzazione:
  - equal width: i bin sono tutti di ugual misura
  - equal depth: i bin contengono tutti lo stesso numero di istanze
- One-of-N: converte una variabile categorica in numerica
- Factor Analysis: crea nuovi attributi rappresentanti di un insieme di attributi

2.3.5 Integrazione dei dati

È necessario integrare i dati provenienti da differenti tabelle in un'unica tabella, eventualmente effettuando operazioni di aggregazione.

Unità di analisi

È l'entità più importante che verrà analizzata nello studio.

Unità di osservazione

È l'unità sulla quale sono raccolti i dati.

Fallacia ecologica

È errato fare delle conclusioni sugli individui se le unità di analisi erano aggregazione di individui.

2.3.6 Formato dei dati

CSV, ARFF.

2.4 Modeling

- Selezione della tecnica di Modeling in base alle caratteristiche degli algoritmi e alle necessità
- Valutazione del modello (coss-validation)
- Ricerca (di parametri, del modello, di entrambi)
- Applicazione la tecnica ai dati
- Valutazione dei risultati del modello (accuratezza, errori di classificazione, MSE per la regressione)

2.5 Evaluation

Valutiamo i risultati del processo in relazione agli obiettivi di business.

2.6 Deployment

Mettere in pratica i risultati ottenuti.

Capitolo 3

Apprendimento di regole

Obiettivo Determinare una ipotesi (congiunzione di vincoli) che coincida con il concetto target sull'intero insieme delle istanze.

Definizioni

Ipotesi Congiunzione di vincoli.

Copertura Un'ipotesi h *copre* un esempio positivo se correttamente classifica tale esempio come positivo.

$$h(x) = c(x) = 1$$

Soddisfacibilità Un esempio $\langle x, c(x)\rangle$ *soddisfa* un'ipotesi h se $h(x) = 1$ indipendentemente da $c(x)$.

Consistenza Un'ipotesi h è *consistente* con un esempio $\langle x, c(x)\rangle$ se $h(x) = c(x)$ indipendentemente dal valore della classe.
Un'ipotesi consistente con un dataset D se è consistente $\forall d \in D$.

Version Space Il Version Space è il sottoinsieme delle ipotesi in H consistenti con gli esempi in D.

$$VS_{H,D} \equiv \{\ h \in H \ \mid \ Consistent(h, D)\ \}$$

3.1 Generate-and-test

Generare tutte le possibili ipotesi e testare sul dataset. Computazionalmente impossibile.

3.2 Find-S

Si basa sull'ordinamento nello spazio delle ipotesi.

Partendo dall'ipotesi nulla, per ogni esempio positivo, generalizza l'ipotesi in modo da coprire l'esempio. Alla fine avrò l'ipotesi più specifica che riesce a coprire tutti gli esempi positivi. Non prende mai in considerazione i negativi. Potrebbe non essere consistente.

3.3 List-then-eliminate

Partendo dall'insieme H, genero il $VS_{H,D}$ eliminando tutte le ipotesi non consistenti con D. Computazionalmente improbabile.

3.4 Candidate Elimination

General Boundary G è il sottoinsieme di H che contiene tutte le ipotesi più generali consistenti con D.

Specific Boundary S è il sottoinsieme di H che contiene tutte le ipotesi più specifiche consistenti con D.

Algoritmo

Per ogni esempio d di training, aggiorna S e G in modo da eliminare le ipotesi non consistenti con d e, se positivo (negativo), generalizza (specializza) S (G).

Caratteristiche

Se non ci sono errori nel dataset D e se $c \in H$ (inductive bias), se S e G convergono in una singola ipotesi h, $h = c$.

Se l'algoritmo restituisce un insieme vuoto, allora ci potrebbero essere errori in D oppure in H non c'era c.

3.5 Inductive Bias

> "Un algoritmo che non fa nessuna assunzione a priori sull'identità del concetto target, non ha nessuna base razionale per classificare qualsiasi nuova istanza."

Il bias induttivo è un insieme minimo di asserzioni che permettono la classificazione di nuove istanze mai viste.

3.6 Sequential Covering (or Separate-and-conquer)

Esegue iterativamente LEARN-ONE-RULE fino a quando non vengano coperti tutti e solo gli esempi positivi.

Ad ogni iterazione trova le ipotesi che coprono il maggior numero di esempi positivi, purché conservi la consistenza anche con gli esempi negativi.

3.6.1 Sequential Covering + Candidate Elimination

Implementa LEARN-ONE-RULE con il Candidate Elimination.

Bottom-up search. Parte da un esempio positivo (seed) ed elimina le ipotesi inconsistenti da G.

3.6.2 PRISM

Genera delle regole in maniera iterativa, utilizzando un approccio top-down e scegliendo di volta in volta il test che massimizzi una misura di purezza $\frac{p}{t}$ che media fra accuratezza e copertura.

Ogni regola non pura al 100%, viene ulteriormente specializzata, con la congiunzione di un nuovo test, fino a raggiungere la purezza massima. In assenza di errori nel dataset, la regola perfetta verrà sempre raggiunta.

3.6.3 Beam Search

Gli approcci precedenti effettuavano la scelta migliore localmente, rischiando di non raggiungere mai l'ottimo globale. Con la *Beam Search* si prendono in considerazione ad ogni iterazione le k migliori scelte per aumentare la probabilità di raggiungere l'ottimo globale.

3.7 Simultaneous Covering

È possibile generare regole a partire da alberi di decisione. In questo modo le regole trovate non saranno indipendenti tra loro, ma copriranno gli esempi di training in maniera "simultanea".

Il simultaneous covering è generalmente più veloce del sequential.

3.8 Multiple-Concept Learning

Negli approcci precedenti si cercavano regole in grado di classificare gli esempi di una sola classe e si lasciavano tutti gli altri esempi alla classe di default. È possibile generare regole con diversi valori di classe nelle conclusioni, a volte anche non mutualmente esclusivi.

La dipendenza trai concetti può essere un problema.

3.9 Problema di overfitting

In caso di presenza di dati rumorosi, le regole perfette trovate dagli algoritmi saranno altamente adattati ai dati di training e quindi saranno scarsamente utili per predire nuovi dati. È necessario creare regole più semplici anche a costo di ridurre la purezza, in modo che siano più predittive.

Capitolo 4

Alberi di decisione

4.1 Attributi discreti

Ogni nodo-test può avere tanti figli quanti sono i valori possibili per quell'attributo oppure è possibile effettuare test di numero arbitrario partizionando l'insieme dei valori possibili. [1]

4.1.1 Entropia

- S Training
- $C_j \dots C_k$ Classi
- $RF(C_i, S)$ frequenze relative dei $s \in S$ di classe C_i

$$E(S) = -\sum_{i=1}^{k} RF(C_i, S) \log(RF(C_i, S))$$

Misura l'incertezza contenuta in S.

4.1.2 Information Gain

Rappresenta il guadagno di informazione, ovvero la riduzione di incertezza ottenuta grazie al test.

$$G(S, t) = E(S) - \sum_{i} \frac{|S_i|}{|S|} E(S_i)$$

Il test migliore sarà quello in grado di massimizzare $G(S, t)$.

[1] Non tutti i test hanno lo stesso costo.

4.1.3 Problema

L'Information Gain favorisce i test con numerosi valori possibili.

Soluzione 1: utilizzare solo test binari. Computazionalmente oneroso.

Soluzione 2: Pesare l'IG con il rapporto delle istanze in ogni partizione.

$$P(S,t) = -\sum_i \frac{|S_i|}{|S|} \log\left(\frac{|S_i|}{|S|}\right)$$

P è l'informazione potenziale della partizione stessa, senza considerare la classe.

4.1.4 Gain ratio Criterion

$$GainRatio(S,t) = G(S,t)/P(S,t)$$

Il test preferito sarà quello che massimizza il Gain Ratio.

4.2 Attributi continui

Sugli attributi continui si effettuano test del tipo $A_i < \theta$ per suddividere in 2 sottoinsiemi. [2]

4.2.1 Information Gain

Calcoliamo l'IG sui valori θ candidati.

I candidati saranno i valori medi degli intervalli esistenti fra 2 coppie di valori di classi diverse.

4.3 Missing Value Problem

- Off-line: Processo Kdd → Data transformation
- On-line: l'algoritmo è in grado di gestirli

I missing value generano 3 tipi di problemi:

- Valutazione di un test
- Partizionamento del training set
- Classificazione di un nuovo esempio

[2] Non tutti i test hanno lo stesso costo.

4.3.1 Valutazione di un test

Pesiamo l'Information Gain di un test t su un attributo A_j con il rapporto tra esempi senza missing value in A_j e il totale degli esempi.

Nel calcolo di $P(S,t)$ dobbiamo considerare il missing value come ulteriore valore di A_j.

4.3.2 Partizionamento del training set

Le tuple con missing value in A_j, in un eventuale partizionamento con test su A_j verranno inserite in tutti i sottoinsiemi S_i, pesate con la probabilità di appartenere a quel sottoinsieme.

$$w_{sunny} = \frac{\#tuple\ sunny}{\#tuple\ senza\ missing\ value\ in\ A_j}$$

Alle foglie degli alberi avremo un doppio valore. `Don't Play(3.4/0.4)` dove il primo indica il numero di esempi che ricadono in quella foglia (anche parziali) e il secondo il numero di errori commessi.

4.3.3 Classificazione di un nuovo esempio

Dato un nuovo esempio con missing value in A_j, quando raggiungo un test su A_j, devo esplorare tutti i sottoalberi, pesando con la probabilità che l'esempio ricada in quel sottoalbero e combinando aritmeticamente i risultati.

4.4 Problemi

Gli alberi di decisione non permettono la revisione delle scelte, quindi possono non raggiungere l'ottimalità.

Un errore fatto nella scelta di un nodo di test, si propaga su tutti i suoi discendenti.

Gli alberi di decisione sono limitati ai problemi risolvibili splittando ripetutamente lo spazio delle soluzioni.

Gli alberi troppo profondi possono frammentare troppo lo spazio delle soluzioni e adattarsi eccessivamente ai dati di training (*overfitting*).

Possibili soluzioni:

- Fermare la crescita (difficile)
- Potare

4.5 Pruning

4.5.1 REP - Reduced Error Pruning

Usa un pruning set per stimare l'accuratezza di un nodo intermedio v e confrontarla con quella del suo sottoalbero T.

$$Gain_{REP} = \varepsilon_T - \varepsilon_v$$

dove ε è il numero di errori di classificazione.

Restrizione bottom-up T può essere potato solo se non contiene un sottoalbero con errore inferiore a T.

Algoritmo

Partiamo dall'albero completo. Visitiamo l'albero in post-ordine. Per i nodi intermedi v:

- calcolo l'accuratezza (sul pruning set) dell'albero completo
- calcolo l'accuratezza (sul pruning set) potando a v e sostituendo T con la classe di maggioranza che ha nel growing set

In caso di aumento di accuratezza, poto. In caso di uguaglianza, poto per Occam.

4.5.2 MEP - Minimal Error Pruning

- Non necessita di pruning set.
- Stima gli errori sul growing/training.
- Usa il metodo Bayesiano per la stima delle probabilità.
- Bottom up.
- Pota per minimizzare la stima degli errori di classificazione.

Calcolo la probabilità di misclassification ad un noto v ignorando i suoi sottoalberi figli (utilizzo la classe di maggioranza a v) e poi prendendoli in considerazione.

Se l'errore statico (senza figli) è minore o uguale (Occam) all'errore di backed-up (con i suoi sottoalberi), poto.

L'errore a T sarà il minimo trai 2.

$$E(T) = \min\left(e\left(v\right), \sum_i p_i e\left(T_i\right)\right)$$

$e(v)$ possiamo calcolarlo in 2 modi.

Dati N esempi, n_C numero di esempi di classe di maggioranza C, p_{C_a} probabilità a priori di classe C, m parametro non negativo:

Laplace

$$p_C = \frac{n_C + 1}{N + k}$$

m-estimate

$$p_C = \frac{p_{C_a} \times m + n_c}{N + m}$$

4.5.3 PEP - Pessimistic Error Pruning

Come MEP però top-down. L'errore del nodo è calcolato come segue.

Dati N esempi totali, N_v gli esempi che ricadono nel nodo v, ε_v il numero di errori di classificazione al nodo v :

$$q(v) = \frac{\varepsilon_v + 0.5}{N_v}$$

l'errore all'albero è:

$$q(T) = \frac{\sum_{l \in leafs(T)} (\varepsilon_l + 0.5)}{N_v}$$

4.5.4 EBP - Error-Based Pruning

Miglioramento di PEP. Bottom-up (al contrario di PEP). Aumenta l'errore pessimistico con la regola di pruning 1-SE (1 standard error).

Pota se

$$q(v) \leq q(T) + SE\,(q(T))$$

Permette l'innesto. Qualsiasi nodo interno può essere rimosso e rimpiazzato da uno dei suoi sottoalberi.

4.5.5 CCP - Cost-Complexity Pruning

Considera il tasso di errore sul growing e sul pruning. Considera la dimensione dell'albero. Trova il compromesso per minimizzare l'errore e la complessità.

$$Total\ cost = Error\ cost + Complexity\ cost$$

Con $R()$ il numero di errori di classificazione nel growing set e N_T il numero di foglie in T,

$$Cost(T) = R(T) + \alpha N_T$$

$$Cost(v) = R(v) + \alpha$$

α rappresenta il costo computazionale di ogni foglia.

4.5.6 Rule Pruning

Converte l'albero in regole e le valuta indipendentemente. Rimuove le precondizioni se il risultato è più accurato. Ordina le regole per accuratezza stimata. Applica le regole in questo ordine per classificare.

Capitolo 5

Framework Bayesiano

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

5.1 MAP - Maximum A Posteriori

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

per il teorema di Bayes e per la costanza di $P(D)$, deriva che:

$$h_{MAP} = \arg\max_{h \in H} P(D|h)P(h)$$

Se tutte le ipotesi $h \in H$ sono equiprobabili, $P(H)$ sarà costante $\left(\frac{1}{|H|}\right)$ e avremo la maximum likelihood (ML) hypothesis:

$$h_{ML} = \arg\max_{h \in H} P(D|h)$$

5.2 MDL - Minimum Description Length

Il principio di Minimum Description Length raccomanda la scelta della ipotesi che minimizzi la somma di due lunghezza di descrizione, una che rappresenti l'ipotesi (C_1) e l'altra che rappresenti $D|h$ (C_2).

¡¡¡¡¡¡¡ HEAD

$$h_{MDL} = \arg\min_{h \in H} (L_{C_1}(h) + L_{C_2}(D|h))$$

=======

$$h_{MDL} = \arg\min_{h \in H} (L_{C_1}(h) + L_{C_2}(D|h))$$

¿¿¿¿¿¿¿ 6558c64bb230c5b744c6f99d7406fae405b4f667

Come Occam, raccomanda la scelta dell'ipotesi più semplice (corta). Fornisce un modo per gestire l'overfitting.

h_{MDL} e h_{MAP} sono fortemente correlate. Infatti:

$$\begin{aligned} h_{MAP} = & \arg\max_{h \in H} \quad P(D|h)P(h) \\ h_{MAP} = & \arg\max_{h \in H} \quad (\log_2 P(D|h) + \log_2 P(h)) \\ h_{MAP} = & \arg\min_{h \in H} \quad (-\log_2 P(D|h) - \log_2 P(h)) \end{aligned}$$

$\log_2 P(h)$ rappresenta la lunghezza della codifica ottimale dell'ipotesi h. La chiamiamo $L_{C_H}(h)$. $\log_2 P(D|h)$ rappresenta la lunghezza della codifica ottimale dell'ipotesi $D|h$. La chiamiamo $L_{C_{D|h}}(D|h)$.

Quindi se $C_1 = C_H$ e $C_2 = C_{D|h}$ allora $h_{MDL} = h_{MAP}$.

5.3 Bayesian Optimal Classifier

Possiamo classificare una nuova istanza combinando le predizioni di tutte le ipotesi, pesate con la loro probabilità a posteriori.

$$P(v_j|D) = \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

la nuova istanza verrà classificata in base a

$$\arg\max_{v_j \in V} P(v_j|D)$$

Qualsiasi sistema che classifica in questo modo è chiamato Classificatore Ottimale di Bayes.

Nessun altro metodo, usando lo stesso spazio delle ipotesi e la stessa conoscenza a priori, può superare questo metodo in media.

Poiché è troppo oneroso da applicare, può essere approssimato utilizzando solo h_{MAP}

$$P(v_j|D) \approx P(v_j|h_{MAP})P(h_{MAP}|D)$$

5.4 Gibbs Algorithm

L'algoritmo di Gibbs, invece di utilizzare solo h_{MAP}, sceglie ogni volta una ipotesi $h \in H$ a caso, in accordo alla distribuzione di probabilità a posteriori su H, e la usa per predire la nuova istanza x.

L'errore di classificazione è al più il doppio di quello Ottimale di Bayes.

5.5 Naïve Bayes Classifier

Si applica quando ogni istanza è descritta da una tupla di valori $\langle a_1, \ldots, a_n \rangle$. La nuova istanza è assegnata al più probabile valore di classe v_{MAP} data la tupla $\langle a_1, \ldots, a_n \rangle$.

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j | a_1, \ldots, a_n)$$

Per il teorema di Bayes e per la costanza di $P(a_1, \ldots, a_n)$, deriva che:

$$v_{MAP} = \arg\max_{v_j \in V} P(a_1, \ldots, a_n | v_j) P(v_j)$$

$P(v_j)$ può essere stimato con la frequenza relativa del valore v_j nel training set.

Calcolare $P(a_1, \ldots, a_n | v_j)$ è impraticabile a causa delle troppe probabilità da stimare.

Assumiamo che i valori degli attributi siano condizionalmente indipendenti, quindi

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i=1}^{n} P(a_i | v_j)$$

5.5.1 Realizzazione

Realizzare un NBC significa stimare due insiemi di parametri:

$$\theta_{ijk} = \hat{P}(A_k = a_{ki} \mid v_j)$$

$$\pi_j = \hat{P}(v_j)$$

con J possibili classi, n possibili attributi, I possibili valori discreti per ogni attributo.

Possiamo stimare i valori θ_{ijk} e π_j con la m-estimate

$$\frac{N_c + mp}{N + m}$$

con $\frac{N_c}{N}$ la frequenza relativa corrispondente alla probabilità da stimare, p probabilità a priori che vogliamo stimare e m costante che determina quanto pesare p sui dati osservati. Tipicamente $m = |A_k|$ e $p = \frac{1}{|A_k|}$, cioè la correzione di Laplace. In assenza di dati, parto dall'equiprobabilità invece che da $\frac{0}{0}$.

5.5.2 Attributi continui

Nel caso di attributi continui A_k si assume una distribuzione di probabilità Gaussiana. Quindi per utilizzare ogni A_k continuo nella stima dei parametri dobbiamo calcolare media e varianza di ogni A_k

$$\mu_{kj} = E\left[A_k|v_j\right]$$
$$\sigma_{kj}^2 = E\left[(A_k - \mu_{kj})^2|v_j\right]$$

5.5.3 Proprietà

Incrementale (possiamo aggiungere dinamicamente nuovi esempi di training).

Tira fuori non solo una classificazione ma anche una distribuzione di probabilità sulle classi.

Può essere utilizzato per la meta-classificazione, combinando e pesando diversi classificatori.

Capitolo 6

Regressione

6.1 Ipotesi classiche nella regressione con errore additivo ε

- $\varepsilon_i \perp X_1, \ldots, X_p$ Indipendenza dai regressori.
- $E(\varepsilon_i) = 0$ Valore atteso nullo
- $Var(\varepsilon_i) = \sigma^2$ Omoschedasticità (stessa dispersione)
- $Cov(\varepsilon_i, \varepsilon_j) = 0$ Incorrelazione tra gli errori

6.2 Modello di regressione semplice

Si parla di regressione *semplice* quando si prende in considerazione un solo regressore.

$$Y = f(X, \beta) + \varepsilon$$

Si parla di regressione *lineare* semplice quando f assume la forma di una retta con $\beta = (\beta_0, \beta_1)$ quindi:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

6.2.1 Metodo dei minimi quadrati

Metodo per ottenere β_0 e β_1 che minimizzino la sommatoria dei quadrati degli scarti.

La funzione:

$$D(\beta_0, \beta_1) = \sum_i (y_i - \beta_0 - \beta_1 x_i)^2$$

(parabola) ammette un solo punto di minimo, quindi la soluzione è unica. Passa per il punto di coordinate medie $(\bar{x}, \bar{y})$

Teorema di Gauss-Markov Sotto le ipotesi classiche (6.1), gli stimatori sono lineari, non distorti e a varianza uniformemente minima.

6.2.2 R^2

$$\sum_i (y_i - \bar{y})^2 = \sum_i (y_i - \hat{y}_i)^2 + \sum_i (\hat{y}_i - \bar{y})^2$$

$\sum_i (y_i - \hat{y}_i)^2$ Devianza di dispersione (residua)
$\sum_i (\hat{y}_i - \bar{y})^2$ Devianza di regressione (spiegata)

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

R^2 ($\rightarrow [0, 1]$) calcola il complementare della percentuale tra devianza residua e devianza totale.

Per ottenere funzioni di regressione migliori dobbiamo massimizzare R^2, cioè minimizzare la devianza residua, cioè avvicinare il più possibile le $\hat{y}_i$ (y calcolate da f) ai valori reali y.

Diagramma di *Anscombe*

- Ascisse: valori interpolati
- Ordinate: residui $y_i - \hat{y}_i$
- Obiettivo: Distribuzione casuale dei punti

Diagramma quantile-quantile

- Ascisse: normale standard
- Ordinate: distribuzione $\hat{\varepsilon}_i$ standardizzati e ordinati
- Obiettivo: $y = x$

6.2.3 Regressori qualitativi

Si sfrutta una funzione indicatrice I_E che assume i valori 1 o 0 a seconda che una certa condizione E sia o meno verificata.

$$\hat{\beta}_0 = \bar{y}_B$$
$$\hat{\beta}_1 = \bar{y}_A - \bar{y}_B$$

6.3 Regressione polinomiale semplice

Una variabile, f polinomiale di grado p con $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_p X^p + \varepsilon$$

La variabile X deve essere nota per almeno $p + 1$ punti.

6.4 Regressione lineare multipla

p regressori, f lineare con $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$$

Le variabili X_i devono essere note per almeno $p + 1$ punti.

6.5 Notazione matriciale

È possibile esprimere tutti i modelli visti finora con la notazione matriciale.

$$Y = \underline{\mathbf{X}}^T \underline{\beta} + \varepsilon$$

dove: $\underline{\mathbf{X}}$ è il vettore colonna dei regressori e $\underline{\beta}$ è il vettore colonna dei coefficienti.

6.6 Trasformazione dei dati

Non necessariamente le ipotesi classiche devono essere valide con riferimento alle variabili originali.

È quindi possibile che tali ipotesi valgano con riferimento a loro opportune trasformazioni.

Esempio:

$$ln(Y) = \beta_0 + \beta_1 \ln(X_1) + \beta_2 e^{X_2} + \beta_3 \frac{X_2}{X_1} + \ldots + \varepsilon$$

6.7 Alberi di regressione

Per regredire potremmo utilizzare una funzione approssimante a gradini, cioè una funziona costante a tratti su intervalli.

Per farlo è necessario partizionare il dataset e poi trovare i parametri c_j associati alla partizione.

Se la partizione è nota, possiamo trovare i parametri c_j che minimizzino l'EEP (Errore Empirico di Previsione, media del quadrato degli scarti tra y e $\hat{y}$).

Se la partizione non è nota, dovremmo valutare tutte le possibili partizioni. Ciò sarebbe computazionalmente impossibile quindi utilizziamo una procedura ricorsiva. Costruiamo l'albero di regressione.

Parto dalla partizione con 1 solo elemento (tutto il training) e ad ogni passo: Provo tutti i valori s trai valori del training. Splitto su s, stimo i parametri c della funzione associata a questa partizione, calcolo l'EEP e lo confronto con l'EEP al passo precedente.

Il test finale scelto sarà quello che massimizzerà il decremento di EEP dal passo precedente tra tutte le suddivisioni di tutti gli elementi della partizione.

6.7.1 Overfitting

Criteri d'arresto

- Individuazione di un numero minimo di osservazioni per ciascuna foglia
- Individuazione di una soglia minima per il decremento dell'EEP

Pruning selettivo

6.7.2 Variabili categoriche

In caso di ordinali, i test saranno del tipo $x < s$. In caso di nominali, $x \in M$ con M un sottoinsieme dei possibili valori dell'attributo.

6.8 Alberi di Modelli

Un albero di modello è un albero di regressione che nelle foglie presenta delle funzioni lineari al posto delle funzioni costanti.

6.9 SMoTI - Stepwise Model Tree Induction

Algoritmo per la creazione di alberi di modelli.

Negli alberi di modelli tradizionali, i parametri dei modelli vengono calcolati alle foglie solo sugli esempi che ricadono in quella particolare foglia.

L'idea alla base di SMoTI è quella di sfruttare più esempi nel calcolo di alcuni parametri per offrire una visione più globale del dataset.

Per fare ciò, introduce un nuovo tipo di nodo chiamato nodo di regressione, non foglia, contenente una regressione parziale su un solo regressore e avente un unico figlio. Tutti gli esempi di quel ramo, passando per il nodo di regressione, subiscono una trasformazione lineare per eliminare l'effetto di quel regressore su tutti gli altri attributi.

Il modello finale per ogni foglia sarà dato da una combinazione della foglia con tutti i nodi di regressione incontrati nel ramo percorso.

Capitolo 7

Analisi di associazione

7.1 Correlazione di 2 variabili

Il coefficiente di correlazione di Pearson ρ_{XY} misura la forza di un'associazione lineare fra 2 variabili X e Y. $\rightarrow [-1; 1]$

$$\begin{aligned}
\sigma_X^2 &= E\left[(X - E[X])^2\right] && (7.1)\\
\sigma_Y^2 &= E\left[(Y - E[Y])^2\right] && (7.2)\\
\sigma_{XY} &= E\left[(X - E[X])(Y - E[Y]\right] && (7.3)\\
\sigma_{X+Y}^2 &= \sigma_X^2 + \sigma_Y^2 + 2\sigma_{XY} && (7.4)\\
\rho_{XY} &= \frac{\sigma_{XY}}{\sqrt{\sigma_X^2 \sigma_Y^2}} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} && (7.5)\\
r_{XY} &= \frac{s_{XY}}{\sqrt{s_X^2 s_Y^2}} = \frac{s_{XY}}{s_X s_Y} && (7.6)
\end{aligned}$$

7.2 Correlazione parziale

È una misura di associazione fra due variabili, dopo aver controllato l'effetto di una o più variabili addizionali.

$$r_{xy.z} = \frac{r_{xy} - (r_{xz} \cdot r_{yz})}{\sqrt{(1 - r_{xz}^2) \cdot \left(1 - r_{yz}^2\right)}}$$

7.3 Regressione multipla

Nella regressione multipla, i coefficienti β_i sono legati alla correlazione parziale. In particolare, se varianza 1, $\beta_i = \rho_{Yi.1,2,\ldots,i-1,i+1,\ldots,p}$

7.4 2 Variabili qualitative

Nel caso di variabili nominali, è impossibile usare Pearson. Si ricorre alle tabelle di contingenza.

7.4.1 χ^2

$$\chi^2 = \sum_i \frac{\left(f_o^i - f_e^i\right)^2}{f_e^i}$$

dove f_o^i è la frequenza osservata in ogni cella e f_e^i è la frequenza attesa calcolata come:

$$f_e^i = \left(\frac{c_i r_i}{N}\right)$$

con c_i somma sulla i-esima colonna e r_i somma sulla i-esima riga. χ^2 è una misura simmetrica.

7.4.2 Lambda

Misura la percentuale di miglioramento nella nostra abilità di predire il valore della variabile dipendente, una volta che conosciamo il valore della variabile dipendente.

$$\lambda(y|x) = \frac{\sum_x \max_y f_{xy} - \max_y f_{*y}}{N - \max_y f_{*y}}$$

A parole, misura la percentuale di nuove istanze corrette sul totale di quelle errate senza conoscere queste nuove informazioni.

$$\frac{Tutte\ le\ corrette\ ora\ MENO\ le\ giá\ corrette}{tutto\ MENO\ le\ giá\ corrette}$$

7.4.3 Spearman Rank Correlation Coefficient

Indice di correlazione tra due classifiche di variabili.

È definita:

$$r_S = 1 - \frac{6\left[\sum_i D_i^2\right]}{(n-1)\ n\ (n+1)}$$

con D_i le differenze tra le posizioni e n la posizione massima.

7.5 Regole di associazione

Task di data mining descrittivo. Mira alla scoperta di relazioni fra gli oggetti di interesse nel database.

In generale una regola di associazione ha forma: $\mathbf{X} \rightarrow \mathbf{Y}$ che correla la presenza di un insieme di oggetti $\mathbf{X}$ (antecedente) con un altro insieme di oggetti $\mathbf{Y}$ (conseguente).

$s\%$ è il supporto. Misura la percentuale di presenza di entrambi i set nell'intero DB. $p(\mathbf{X} \cup \mathbf{Y})$

$c\%$ è la confidenza. Misura la percentuale di presenza del conseguente nelle tuple contenenti l'antecedente. $p(\mathbf{Y}|\mathbf{X})$

7.6 Mining di regole di Associazione

La scoperta di regole di associazione si divide in due passi:

1. Cercare gli itemset più frequenti sul DB ($s\% \geq s\%_{\text{min}}$)
2. Generare regole di associazione partendo dagli itemsets ($c\% \geq c\%_{\text{min}}$)

7.6.1 Ricerca di Itemsets

APRIORI

Ricerca di itemsets.

1. Inizialmente scanna il DB per prendere gli itemset di lunghezza 1.
2. Genera itemset candidati di lunghezza $k+1$ a partire dai frequenti di lunghezza k
3. Testa i candidati sul DB.
4. Termina quando non ci sono più nuovi frequenti oppure non possono essere generati altri candidati.

Problemi Numerosi scan del DB sono costosi. Numero di candidati troppo alto.

FPGrowth

Ricerca di itemsets.

Usa lo stesso principio di pruning di Apriori.

Scansiona il DB solo 2 volte. La prima volta cerca gli itemset di lunghezza 1. La seconda volta, costruisce la propria struttura dati chiamata FP-tree.

Algoritmo

1. Creo la tabella degli 1-itemset frequenti ed eventualmente elimino quelli che non superano una soglia.
2. Creo un nuovo database epurato dagli item inutili e con ogni itemset ordinato per frequenza decrescente.
3. Partendo da questo nuovo DB, costruisco l'albero FP
4. Per ogni item, scrivo tutti i possibili percorsi sull'albero che portano a quell'item e la frequenza con cui accade.
5. Genero tutti i possibili itemset frequenti, unendo ogni item con l'insieme delle parti dell'itemset del percorso più frequente.

Vantaggi

- Non genera candidati (direttamente i frequenti)
- Non testa candidati (non ha candidati)
- Scansione del DB solo 2 volte

7.6.2 Creazione di regole di associazione

Algorithm 1 Generazione di regola da itemset frequenti.

$R = \{\}$
for all itemsets frequenti X **do**
  for all itemsets $A \subset X$, $A \neq \emptyset$ **do**
    $B = X - A$
    crea la regola r: $(A \Rightarrow B)$
    calcola la confidenza $c\%_r$
    if $c\%_r \geq c\%_{\min}$ **then**
      $R = R \cup \{r\}$
    end if
  end for
end for
return R

7.6.3 Problemi

Per gestire variabili quantitative, è necessario discretizzare.