

Tesina di Intelligenza Artificiale

Il Classificatore NAIVE BAYES

Studente

Stefano Tortora

Professore

Filippo Sciarrone

Indice

Indice.....	2
-------------	---

Parte seconda:

1	Background Teorico	3
1.1.	Il Machine Learning	3
1.2.	La Probabilità: accenni e visione Bayesiana	9
1.3.	I Classificatori	11
1.3.1	Classificazione di Testi	11
1.3.2	Training, Test e Validation.....	14
1.3.3.	Valutazione dei classificatori	15
1.3.4.	Costruzione di un Classificatore	21
1.3.5.	Modelli di Classificatori	22

Parte seconda:

2	IL Classificatore Naive Bayes.....	23
2.1.	Cenni storici sull'autore.....	23
2.2.	Introduzione al Teorema di Bayes	24
2.3.	Il Teorema di Bayes.....	26
2.4.	Esempi di applicazione del teorema di Bayes	28
2.5.	L' Apprendimento bayesiano.....	30
2.6.	Le reti Bayesiane	32
2.7.	Il Classificatore Naive Bayes	35
2.8.	Esempi di applicazione dell' Algoritmo Naive Bayes	44

Parte prima:

1 Background Teorico

In questo primo capitolo viene fatta un'introduzione sulla teoria necessaria per poter comprendere e studiare al meglio l'algoritmo *Naive Bayes*.

Questo capito è quindi suddiviso in tre rami principali, nel primo viene presentato il Machine Learning e le sue principali caratteristiche, nel secondo sono presenti i principi fondamentali della teoria della Probabilità ed il particolare viene introdotto l'approccio bayesiano alla probabilità ed alla statistica, mettendo in risalto le differenze con la visione classica; nel terzo invece viene fatta inizialmente una breve introduzione nella quale viene spiegato cosa effettivamente è un classificatore, ed in seguito vengono presentati concetti fondamentali per poter studiare un classificatore nel dettaglio.

1.1. Il Machine Learning

Il Machine Learning (o Apprendimento Automatico) è il settore della Computer Science che si occupa di realizzare dispositivi artificiali capaci di emulare le modalità di ragionare tipiche dell'uomo: riconoscere, decidere, scegliere, . . . Seguendo la definizione di Simon diremo che il ML è:

*...ogni cambiamento in un sistema che gli permette di migliorarsi
nell'esecuzione dello stesso compito o di un altro
dello stesso tipo. [Sim83]*

Da questa definizione si deduce che il ML rappresenta una forma di adattamento all'ambiente attraverso l'esperienza, analogamente a quanto

avviene per gli organismi biologici o, in senso lato e in tempi più lunghi, per le specie viventi che vi si adattano geneticamente. L'obiettivo del ML è far migliorare un sistema senza bisogno del continuo intervento umano. Per fare questo tali dispositivi devono essere in grado di apprendere, ovvero di estrarre informazioni su un determinato problema esaminando una serie di esempi ad esso relativi.

La messa a punto di dispositivi artificiali capaci di apprendere riveste una tale importanza che costituisce uno degli obiettivi primari di diversi settori scientifici: la Statistica, l'Intelligenza Artificiale, le Scienze Cognitive, ecc.

Nel Machine Learning per introdurre uno dei concetti fondamentali è bene introdurre i concetti di procedimento deduttivo ed induttivo:

- Il ragionamento deduttivo passa dalle regole agli esempi: è tipico dei computer che devono essere programmati (= inserire delle regole) per produrre un risultato (eseguire un'operazione, calcolare limiti e derivate).
- Il ragionamento induttivo passa dagli esempi alle regole generali: è tipico dell'uomo (impara dall'esperienza). È ciò che avviene quando un bambino impara a parlare o a leggere. Questo processo viene detto **apprendimento**.

Ed è proprio l'apprendimento il concetto su cui ruota il Machine Learning. Apprendere consiste pertanto nell'effettuare un ragionamento di tipo induttivo in maniera implicita (comprensione del linguaggio) o esplicita (costruzione di una strategia di gioco).

Il passaggio dagli esempi alle regole può essere visto come un processo di estrazione di informazione da un insieme di esempi. In particolare, supponiamo di avere n immagini di volti, alcune delle quali appartengono ad una data persona. L'obiettivo del processo di apprendimento potrebbe consistere nel costruire un dispositivo automatico che riconosca la persona prescelta indipendentemente dall'espressione che assume nella immagine, dall'orientazione del viso, dalle zone di penombra. Possiamo affermare che la

costruzione del dispositivo richiede di estrarre dalle n immagini a disposizione una quantità di informazione sufficiente a comprendere come il cervello umano avrebbe eseguito il processo di riconoscimento.

Per estrarre l'informazione consideriamo un sistema fisico S , caratterizzato da una variabile (vettoriale) z , in cui desideriamo estrarre da n osservazioni z_1, z_2, \dots, z_n la maggior quantità possibile di informazione, così da comprendere (o al più prevedere) il comportamento del sistema S . Il vettore z può contenere soltanto uscite o anche ingressi al sistema S , posti o no sotto il nostro controllo. L'estrazione dell'informazione sul sistema S a partire da esempi ha quale obiettivo finale la realizzazione di un dispositivo artificiale (modello) D che emuli al meglio il comportamento del sistema S . Sarebbe molto desiderabile comprendere il funzionamento del modello D , in quanto ciò potrebbe far luce sulle motivazioni del comportamento di S . A questo punto ci possiamo trovare in tre diverse situazioni:

- **Scatola bianca o trasparente:** il comportamento del sistema S può essere completamente determinato svolgendo calcoli di tipo algebrico e/o differenziale;
- **Scatola grigia:** il valore dei parametri ignoti può essere stimato a partire dagli n esempi a disposizione, impiegando un'opportuna tecnica di ottimizzazione che minimizzi una misura dell'errore commesso dal nostro modello sui dati a disposizione.
- **Scatola nera:** occorre dapprima definire il modello da impiegare in modo sufficientemente generale e successivamente determinare la sua realizzazione specifica per il sistema S a partire dagli n esempi a disposizione.

Il problema dell'estrazione dell'informazione da esempi (nel caso “scatola nera”) è quindi il problema centrale del **machine learning** o dell'

apprendimento, che definiremo **automatico** per distinguerlo da quello naturale dell'uomo: desideriamo ottenere un modello D del sistema fisico S partendo da un insieme finito di n osservazioni z_1, z_2, \dots, z_n (training set). In generale, tali osservazioni sono influenzate da errori di acquisizione e da altri tipi di incertezze (rumore), per cui la soluzione del problema di apprendimento automatico richiede un approccio di tipo statistico. Si suppone pertanto la presenza di una densità di probabilità $p(z)$ incognita (caratteristica del sistema S), attraverso la quale sono stati generati i campioni z_i .

I principali impieghi dell'apprendimento Automatico sono: *Data Mining* (estrazione di conoscenza): scoprire regolarità e patterns in dati multi-dimensionali e complessi (es. cartelle cliniche); *Miglioramento delle performance*: macchine che migliorano le loro capacità (es. movimenti di un robot); *Software adattabili*: programmi che si adattano alle esigenze dell'utente (es. Letizia).

Le scelte che devono essere effettuate nella definizione di un sistema di apprendimento sono:

1. Componenti Migliorabili: decidere quali sono le componenti sulle quali “lavorare”
2. Scelta e Rappresentazione di una “funzione obiettivo” che rappresenti il compito
3. Scelta di un algoritmo di apprendimento
4. Modalità di training: supervisionato, guidato da obiettivi

Esclusa la prima parte per ovvi motivi passeremo ora ad analizzare i successivi punti:

2. Funzione obiettivo:

è l'espressione formale della conoscenza appresa e viene usata per determinare le prestazioni del programma di apprendimento. Difficilmente un sistema riesce ad apprendere perfettamente una funzione obiettivo f ; in genere viene appresa una funzione, avente la forma prescelta (polinomio, regole,..), che rappresenta una stima, o **IPOTESI** (indicata con h), per la funzione obiettivo. L'obiettivo è di apprendere una h che approssimi f "il meglio possibile".

3. Algoritmo di apprendimento:

La scelta dell'algoritmo e della funzione obiettivo sono ovviamente correlate; i principali algoritmi, di cui in seguito parleremo meglio sono:

- Metodi statistici: Bayesian learning, Markov models
- Metodi algebrici: Gradient descent, Support Vector Machines
- Metodi knowledge-based: Alberi di decisione, Regole di associazione

4. Modalità di training:

Sono basate naturalmente sull'informazione che si ha a disposizione:

- Apprendimento non supervisionato: non sono individuabili ingressi (pilotabili o meno) al sistema S . L'unica attività rilevabile è la generazione dell'uscita z . L'obiettivo dell'apprendimento è individuare una qualche legge che metta in relazione tra loro i campioni z_i (*clustering*).

- **Apprendimento supervisionato:** il sistema S riceve un ingresso x da un generatore G (che può essere o no sotto il nostro controllo) e produce un uscita y . Pertanto, le osservazioni z_i sono costituite da coppie ingresso/uscita $(x_i; y_i)$.
- Con Rinforzo: L'apprendista interagisce con l'ambiente e riceve una ricompensa (numerica) positiva o negativa (es. se un robot che fa goal il "peso" della sequenza di azioni che lo ha portato a fare goal viene aumentato)

Oppure sul ruolo dell'apprendista (learner)

- Apprendimento passivo (apprende da esempi a-priori) l'apprendista può apprendere solo dai dati che vengono messi a disposizione (E)
- Apprendimento attivo (apprende anche durante il funzionamento) l'apprendista può fare domande ed esperimenti

Per quanto riguarda l'Apprendimento supervisionato tre sono le classi di problemi che possono essere ulteriormente caratterizzati, a seconda dei valori assunti dalla uscita y del sistema:

- Problemi di riconoscimento, se l'uscita è binaria ($y \in \{0,1\}$). Es. diagnosi medica di una determinata patologia.
- **Problemi di classificazione**, se y varia entro un insieme finito di valori non ordinati ($y \in \{1, 2, \dots, m\}$). Es. riconoscimento di caratteri manoscritti.
- Problemi di regressione, se l'uscita è continua ($y \in \mathbb{R}$). Es. previsione dell'andamento di serie temporali.

1.2. La Probabilità: accenni e visione Bayesiana

Per definire il concetto di probabilità nel corso degli anni vi sono state innumerevoli discussioni; sono infatti nate nel tempo, due scuole di pensiero completamente diverse: la prima ha dato una definizione nota come: *la visione classica della probabilità* mentre la seconda ha dato una detta visione soggettiva della probabilità o meglio *visione bayesiana della probabilità*

La concezione classica afferma che la probabilità è una proprietà fisica del mondo e per essere valutata è necessario ripetere varie volte un esperimento. Più precisamente, siano w_1, \dots, w_n n eventi esaustivi e mutuamente esclusivi e si effettuino N esperimenti osservandone il risultato, sia N_i il numero di volte in cui si è verificato l'evento w_i e $\eta \equiv N_i / N$ il rapporto tra il numero di volte in cui si è verificato l'evento w_i e il numero totale degli esperimenti. Se N è sufficientemente grande, le frequenze η tendono a stabilizzarsi e si può quindi definire la probabilità dell'evento w_i nel seguente modo:

$$P(\omega_i) \equiv \lim_{N \rightarrow +\infty} \eta \equiv \lim_{N \rightarrow +\infty} \frac{N_i}{N}$$

Figura 1: Definizione classica della probabilità

D'altra parte, la visione bayesiana afferma che la probabilità di un evento w_i è il grado di credenza (*degree of belief*) di una persona in quell'evento, piuttosto che una qualche proprietà fisica del mondo: quindi la probabilità è **il grado soggettivo di aspettativa che una persona assegna al verificarsi di un evento incerto**. In modo formale, La probabilità di un evento E , secondo l'opinione di un dato individuo, è il prezzo $P(E)$ che egli giudica "equo" pagare per riscuotere un importo unitario nel caso in cui E si verifichi. Si può dimostrare che, se non esiste una strategia di scommessa per cui una persona perde sicuramente, allora la probabilità così definita soddisfa gli assiomi della

probabilità. Secondo la definizione assiomatica, la probabilità $P(w_i)$ di un evento w_i è un numero che soddisfa i seguenti assiomi:

1. $0 \leq P(w_i) \leq 1$;
2. $P(w_i) = 1$ se e solo se w_i è certo
 $P(w_i) = 0$ se e solo se w_i è impossibile;
3. se w_i e w_j sono due eventi, allora la probabilità della loro disgiunzione vale $P(w_i \cup w_j) = P(w_i) + P(w_j) - P(w_i \cap w_j)$.

I primi due assiomi servono per definire la scala di probabilità e il terzo si ricorda facilmente legandolo al diagramma di Venn per la teoria degli insiemi. Da questi assiomi, si possono derivare tutte le regole della probabilità. Per cercare di rendere il concetto di probabilità bayesiana più formale, molti ricercatori hanno suggerito vari insiemi di proprietà che dovrebbero essere soddisfatti dai gradi di credenza ed è stato visto che ogni insieme di proprietà porta alle stesse regole che non sono altro che le regole della probabilità. Una importante differenza tra la concezione classica e quella bayesiana della probabilità è che per misurare quest'ultima non c'è bisogno di ripetere degli esperimenti, quindi tenendo conto di questa differenza, si può assegnare a certi eventi un valore di probabilità che la visione frequentista non sa attribuire. Ad esempio, volendo valutare la probabilità che una certa squadra possa vincere il campionato il prossimo anno, la concezione classica direbbe di ripetere N volte il campionato, con N molto grande, e di contare il numero di volte che la squadra in questione ha vinto il campionato. Questo è certamente un approccio non praticabile, per cui si giustifica l'introduzione di una diversa visione della probabilità.

1.3. I Classificatori

Un sistema di classificazione o di riconoscimento, considerato in senso ampio, ha il compito di fornire ad un utente (uomo o calcolatore) una valutazione della realtà fisica osservata e tale valutazione si avvale di una suddivisione della realtà (costituita da oggetti detti campioni o “pattern”) in insiemi, aventi caratteristiche omogenee, detti classi.

Nell'analisi di classificazione è importante valutare l'affidabilità del modello per fini predittivi, ciò viene definito mediante il calcolo dell'accuratezza del classificatore attraverso vari metodi. L'obiettivo dell'analisi di classificazione è la verifica dell'esistenza di differenze tra le classi in funzione delle variabili considerate e la formulazione di un modello che sia in grado di assegnare ciascun campione alla classe cui esso appartiene.

1.3.1. Classificazione di Testi

Per classificazione di testi si intende l'archiviazione di documenti sotto particolari categorie di interesse chiamate classi. Ogni classe rappresenterà un particolare argomento, e dire che un documento appartiene ad una determinata categoria significa, in pratica, affermare che quel documento tratta dell'argomento rappresentato da tale classe. Negli anni si è sviluppato il concetto di classificazione automatica di testi; questo consiste sostanzialmente nel considerare un processo che, in maniera automatica, riconosce l'argomento trattato da un documento e lo archivia sotto la rispettiva categoria di interesse. Tali categorie sono delle etichette simboliche, e non sono valide informazioni addizionali sul loro significato. Non sono accettate conoscenze esogene, cioè non vengono aggiunte informazioni dall'esterno, la base per la classificazione viene invece fornita

dalle conoscenze endogene. In pratica ad ogni coppia $\langle d_j, c_i \rangle$ viene assegnato un valore booleano true se il documento d_j appartiene alla classe c_i , altrimenti viene assegnato un valore false.

Più formalmente viene approssimata la funzione sconosciuta Φ chiamata target, definita come:

$$\Phi: D \times C \rightarrow \{T, F\}$$

con una funzione Φ^* chiamata classificatore così che Φ e Φ^* coincidano approssimativamente.

Con D intendiamo l'insieme dei documenti, e con C l'insieme delle categorie di appartenenza. Possono essere definite varie caratteristiche sui classificatori:

- Per ogni intero k , esattamente k elementi di C possono essere assegnati ad ogni $d_j \in D$;
- Esattamente una categoria può essere assegnata ad ogni $d_j \in D$, in questo caso si ha una single label;
- Per ogni $d_j \in D$ può essere assegnata la categoria c_i o la sua opposta c_i negato, in questo caso parliamo di classificazione binaria;
- Per ogni $d_j \in D$ possono essere assegnate un numero di categorie da 0 a $|C|$, in questo caso si parla di multilabel.

Ci sono diversi approcci alla classificazione automatica, alcuni esempi sono:

- Classificazione completamente automatizzata in cui, per ogni coppia documento-categoria $\langle d_j, c_i \rangle$, ci deve essere una decisione T o F, a seconda che il documento d_j appartenga o no alla classe c_i ;

- Considerato un documento d_j potrebbe essere più utile dare una lista di categorie correlate a tale documento, l'utente poi sceglierà quella che più gli interessa;
- Considerata una categoria c_i può essere utile considerare una lista di possibili documenti appartenenti a tale classe, lasciando ancora all'utente la scelta di quale sia il migliore a seconda del suo interesse.

Realizzare un processo che riconosce il contenuto dei documenti significa, in sostanza, realizzare un classificatore che, dato in ingresso un insieme di documenti, restituisce la rispettiva categoria ed archivia tale documento sotto di essa.

Negli anni 80, per implementare la classificazione automatica, venivano usate tecniche di Knowledge Engineering. Questo approccio veniva realizzato tramite l'implementazione di regole logiche definite manualmente, del tipo:

if <DNF formula> then <categoria>

Una DNF formula è una disgiunzione di clausole congiuntive, in pratica un documento viene classificato sotto la categoria <categoria> se e solo se soddisfa la formula, ovvero se e solo se soddisfa una delle disgiunzioni. Le regole logiche dovevano essere definite da uno specialista umano e, se veniva aggiunta una nuova categoria, queste dovevano essere ridefinite.

Negli anni 90 si è andato affermando un nuovo modo di costruzione del classificatore chiamato approccio Machine Learning. Questo metodo viene attuato mediante un processo induttivo chiamato learner, che costruisce automaticamente il classificatore per una categoria c_i , prendendo in considerazione un insieme di documenti scelti manualmente e già classificati sotto le categorie c_i e c_i negato. Intuitivamente accade che al learner vengono passati dei documenti di esempio per "addestrare" tale

classificatore a riconoscere i successivi dati che poi verranno presentati. Gli esempi sono stati già classificati manualmente.

1.3.2. Training, Test e Validation

L'insieme dei dati usati per costruire un classificatore viene diviso in insieme di training ed insieme di test. Il primo viene usato per addestrare il classificatore a riconoscere i successivi file di prova, il secondo serve per fare degli esperimenti sul classificatore appena identificato.

Sia dato un insieme $\varphi = \langle d_1, \dots, d_\varphi \rangle \subset D$ di documenti preclassificati sotto categorie $C = \langle c_1, \dots, c_k \rangle$, e sia conosciuta la funzione

$$\Phi^* : D \times C \rightarrow \{T, F\} \text{ per ogni } \langle d_j, c_i \rangle$$

appartenente a $\varphi \times C$. Un documento d_j è un esempio positivo di c_i se $\Phi^*(d_j, c_i) = \text{true}$, mentre è un esempio negativo se $\Phi^*(d_j, c_i) = \text{false}$.

Per costruire un classificatore il set di dati φ viene diviso in due insiemi:

- Training Set $T_1 = \langle d_1, \dots, d_n \rangle$, il classificatore Φ^* per categorie C viene costruito induttivamente osservando le caratteristiche dei documenti;
- Test Set $T_2 = \langle d_{n+1}, \dots, d_\varphi \rangle$, usato per testare il classificatore. Ogni d_j appartenente a T_2 è passato al classificatore, e il target $\Phi(d_j, c_i)$ è comparato con $\Phi^*(d_j, c_i)$, allo scopo di identificarne le differenze.

Il training set $T_1 = \langle d_1, \dots, d_n \rangle$ può essere ulteriormente diviso in due insiemi, il primo $T_r = \langle d_1, \dots, d_a \rangle$ con cui viene costruito il classificatore, e il secondo $V_a = \langle d_{a+1}, \dots, d_n \rangle$ su cui viene provato tale classificatore, al fine di valutarne la correttezza.

Dato l'insieme di documenti $\varphi = \langle d_1, \dots, d_n \rangle$ è possibile anche definire una funzione $g_\varphi(c_i)$, che identifica la percentuale di documenti appartenenti a c_i , come:

$$g_\varphi(c_i) = \frac{|\{d_j \in \varphi \mid \Phi^*(d_j, c_i) = T\}|}{|\varphi|}$$

1.3.3. Valutazione dei classificatori

In questo paragrafo tratteremo di come valutare un classificatore, ossia come riuscire a capire se il classificatore in questione ha prodotto dei buoni risultati oppure no. I principali metodi utilizzati per dare un'effettiva valutazione di un classificatore sono l'accuratezza e l'efficacia; descriveremo quindi nel dettaglio questi due metodi.

I Misura dell'Accuratezza

In generale l'accuratezza da un'idea della qualità di uno strumento in esame. Nel caso specifico dei classificatori è definita come il numero di campioni correttamente classificati rispetto al numero totale di campioni classificati. Due possibili metodi per determinare l'accuratezza di un classificatore, basati entrambi sulla Cross Validation sono:

- *Dataset train e validation (two folder validation):*

Consiste nell'utilizzare, per i classificatori, un set di addestramento ("*training-set*") costituito da campioni di cui si conosce a priori la classe di appartenenza, curandosi del fatto che tale insieme sia *significativo e completo*, cioè con un numero sufficiente di

campioni rappresentativi di tutte le classi. Per la verifica del metodo di riconoscimento ci si avvale di un set (“*validation-set*”), anch’esso costituito da campioni la cui classe è nota, usato per controllare la generalizzazione dei risultati; esso è costituito da un insieme di campioni diversi rispetto a quelli del training-set. In seguito a questa operazione possiamo calcolare l’errore sia di training che di validazione:

$$E_{training} = \frac{\text{Campioni - sbagliati - in - training}}{N. - \text{campioni - training}}$$

$$E_{validation} = \frac{\text{Campioni - sbagliati - in - validation}}{N. - \text{campioni - validation}}$$

L’errore di validazione ci dà informazioni su quanto bene ha imparato il classificatore. Per definire un buon classificatore però non basta basarsi su questi due errori, bisogna anche considerare il tipo di classificazione che si sta facendo. Volendo esaminare l’accuratezza del metodo su esempi reali, si fornisce al classificatore un set di test (detto “*testing-set*”) le cui classi non sono note. In seguito le notizie sulle classi di questo insieme vengono utilizzate, dopo la classificazione, per determinare gli errori e quindi l’accuratezza reale del classificatore.

- *N-fold validation (Leave One Out)*

Per conoscere le capacità predittive del modello si può utilizzare il metodo della validazione incrociata (cross-validation), consiste nel calcolare il modello con l'esclusione di un oggetto alla volta (metodo leave-one-out) o di un oggetto ogni N oggetti (leave-

more-out), predicendo i valori della risposta per tutti gli oggetti esclusi dal modello. Per esempio, se si hanno 10 campioni e si decide di

utilizzare per metodo leave-one-out per validare il modello di classificazione ottenuto, dovremo procedere come segue:

- eliminare il campione N.1;
- ricalcolare il modello utilizzando solo i 9 campioni rimasti;
- riassegnare il campione N.1 ad una classe in base al modello ricalcolato sui 9 campioni rimasti;
- eliminare il campione N.2;
- ricalcolare nuovamente il modello sui 9 campioni rimasti e procedere alla riassegnazione del campione N.2;
- proseguire eliminando uno alla volta tutti i campioni fino al N.10 e riassegnandoli di volta in volta in base al modello ricalcolato senza il campione eliminato.

Al termine del procedimento avremo così ricalcolato il modello 10 volte, ottenendo una nuova serie di parametri detti "cross-validati" (CV) per la valutazione della bontà del modello di classificazione in predizione. Come risultato del procedimento di validazione si otterranno quindi, oltre all'ER, al NER ed al MR, anche i valori ERCV, NERCV e MRCV. Questi nuovi parametri forniscono di norma stime meno ottimistiche dei precedenti ma, pur fornendo valori peggiori di quelli relativi ai parametri di valutazione non cross-validati, sono più realistici per valutare le prestazioni del modello ottenuto anche per finalità predittive. Tale metodo viene utilizzato soprattutto quando non si ha un numero di dati sufficienti per formare dei training e validation set adeguati.

II Misura dell'Efficacia

La valutazione dei classificatori di testi è effettuata sperimentalmente piuttosto che analiticamente, la classificazione di testi non può essere infatti formalizzata (a causa della sua natura soggettiva) e quindi non può essere valutata analiticamente. La valutazione sperimentale di un classificatore solitamente misura la sua efficacia ossia : l'abilità di prendere la giusta decisione di classificazione. È quindi l'efficacia l'altro parametro utilizzato per la valutazione di un classificatore. Questa viene misurata in termini di precisione (π), e recall (ρ). La precisione rispetto alla classe c_i (π_i) è definita come la probabilità condizionata :

$$P(\Phi^*(d_x, c_i) = T \mid \Phi(d_x, c_i) = T)$$

che è la probabilità che un documento preso a caso, chiamato d_x , sia classificato sotto c_i , e che questa decisione sia corretta, in altre parole che indica il grado di correttezza di un classificatore rispetto alla categoria c_i . Analogamente il recall rispetto a c_i (ρ_i) è definito come la probabilità condizionata:

$$P(\Phi(d_x, c_i) = T \mid \Phi^*(d_x, c_i) = T)$$

che è la probabilità che un documento preso a caso, chiamato d_x , necessiti di essere classificato sotto c_i e che questo sia fatto, ossia indica il grado di completezza di un classificatore rispetto alla categoria c_i . Sia FP_i (falsi positivi) il numero di documenti classificati incorrettamente sotto la classe c_i , TP_i (veri positivi) il numero di documenti classificati correttamente sotto c_i , FN_i (falsi negativi) il numero di documenti non classificati, erroneamente, sotto c_i , e TN_i (veri negativi) il numero di

documenti non classificati, correttamente, sotto c_i , le stime della precisione e del recall possono essere ottenute come:

$$\hat{\pi}_i = \frac{TP_i}{TP_i + FP_i} \quad \hat{\rho}_i = \frac{TP_i}{TP_i + FN_i}$$

Per ottenere π e ρ , possiamo procedere in due modi:

- Micromedia dove π e ρ sono ottenute sommando tutte le decisioni individuali: (dove μ indica la micromedia.)

$$\hat{\pi}^{\mu} = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)}$$

$$\hat{\rho}^{\mu} = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)}$$

- Macromedia dove la precisione ed il recall sono prima valutate localmente per ogni categoria e poi globalmente dalla media dei risultati. Le formule sono: (dove M indica la macromedia)

$$\hat{\pi}^M = \frac{\sum_{i=1}^{|C|} \pi_i}{|C|} \quad \hat{\rho}^M = \frac{\sum_{i=1}^{|C|} \rho_i}{|C|}$$

Le misure π e ρ prese singolarmente non bastano per esprimere l'efficacia. Il classificatore che classifica tutti i documenti sotto c_i ha

$$\rho = 1 \text{ (non ci sono falsi negativi)}$$

Il classificatore che classifica tutti i documenti sotto $\neg c_i$ ha:

$$\pi = 1 \text{ (non ci sono falsi positivi)}$$

Possiamo quindi dedurre che π e ρ sono inversamente proporzionali quindi, per valutare l'efficacia di un classificatore, si deve trovare la giusta combinazione di queste due misure: anche per questo scopo sono state elaborate numerose funzioni di combinazione.

Queste due medie possono dare risultati molto diversi. Quale delle due deve essere usata dipende dalle richieste dell'applicazione. Ci sono misure alternative all'efficacia per valutare un classificatore costruito automaticamente:

- Accuratezza definita come:

$$\hat{A} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Errore definito come:

$$\hat{E} = \frac{FP + FN}{TP + TN + FP + FN} = 1 - \hat{A}$$

Queste misure non sono molto usate in per via della grandezza del loro denominatore, che le rende meno sensibili alle variazioni del numero delle decisioni corrette rispetto a π e a ρ .

1.3.4. Costruzione di un Classificatore

Ci sono vari approcci induttivi per costruire un classificatore automatico di testi, i principali sono essenzialmente due: ranking (semiautomatico) e hard (completamente automatico). Il primo metodo è così attuato: data una categoria $c_i \in C$ viene definita una funzione $CSV_i: D \rightarrow [0,1]$ che riceve in input un documento d_j , e restituisce un numero compreso tra 0 e 1, che rappresenta il grado di appartenenza di d_j a c_i . Questo viene fatto per tutte le classi; il CSV più grande rappresenterà la classe di appartenenza di d_j .

Per costruire un classificatore di tipo hard viene definita la funzione CSV come nel caso precedente; in seguito è presa in considerazione una soglia (threshold) τ_i tale che un risultato di $CSV_i(d_j) \geq \tau_i$ viene interpretato come appartenente alla classe in esame, mentre un risultato $CSV_i(d_j) < \tau_i$ viene considerato non appartenente.

1.3.5. Modelli di Classificatori

Esistono vari tipi di modelli di classificazione e differiscono per il formalismo utilizzato per rappresentare la funzione di classificazione (Linguaggio delle ipotesi).

Elenco di alcuni modelli di classificazione:

- *Basati sugli esempi:* (es Nearest neighbor)
memorizzano tutti gli esempi del training set ed assegnano la classe ad un oggetto valutando la “somiglianza” con gli esempi memorizzati (la cui classe è nota)

- *Matematici* (es. Reti Neurali Artificiali, SVM)
la funzione di classificazione è una funzione matematica, di cui si memorizzano i vari parametri

- *Statistici* (es **Naive Bayes**)
memorizzano i parametri delle varie distribuzioni di probabilità relative alle classi ed agli attributi → per classificare un generico oggetto si possono stimare le probabilità di appartenenza alle varie classi

- *Logici* (es Alberi e Regole di Decisione)
la funzione di classificazione è espressa mediante condizioni logiche sui valori degli attributi

Parte seconda:

2 IL Classificatore Naive Bayes

In questo secondo capitolo viene trattato nel dettaglio l'argomento principe della tesina, ossia l'algoritmo vero e proprio: il *Classificatore Naive Bayes*.

Il capitolo è suddiviso in due parti principali, la prima, dopo una breve introduzione storica sull'autore, introduce il Teorema di Bayes, su cui l'algoritmo si basa, lo esamina dettagliatamente e vengono infine esposti alcuni esempi mirati ad una maggiore comprensione del teorema stesso. La seconda parte invece dopo aver introdotto i concetti di apprendimento Bayesiano e le reti Bayesiane, parla dell'algoritmo ed inseguito ne viene presentato un tipico esempio.

2.1. Cenni storici sull'autore

Thomas Bayes nacque a Londra nel 1702 e morì il 17 aprile 1761 a Tunbridge Wells, Kent. È stato un matematico nonché pastore presbiteriano. È noto in statistica per il suo Teorema di Bayes sulla probabilità condizionata, pubblicato postumo nel 1763. Di lui sono note le seguenti pubblicazioni:



Rev. Thomas Bayes
b. 1702, London
d. 1761, Tunbridge Wells,
Kent

- *Divine Benevolence, or an Attempt to Prove That the Principal End of the Divine Providence and Government is the Happiness of His Creatures* (1731)
- *An Introduction to the Doctrine of Fluxions, and a Defence of the Mathematicians Against the Objections of the Author of the Analyst* (1736)
(in difesa della nuova teoria del calcolo infinitesimale)

- *Essay Towards Solving a Problem in the Doctrine of Chances* (1763, pubblicato postumo in *Philosophical Transactions of the Royal Society of London*)

Pur senza aver mai ricoperto cariche accademiche e pubblicato lavori a suo nome, Bayes fu eletto Fellow della Royal Society nel 1742

È sepolto nel cimitero Bunhill Fields di Londra.

2.2. Introduzione al Teorema di Bayes

Il teorema di Bayes venne presentato nel 1763 nell'articolo *Essay Towards Solving a Problem in the Doctrine of Chances* di Thomas Bayes, pubblicato postumo in *Philosophical Transactions of the Royal Society of London*.

Alcuni anni dopo (nel 1774) viene formulato da Pierre Simon Laplace che probabilmente non era a conoscenza del lavoro di Bayes.

L'importanza di questo teorema per la statistica è tale che la divisione tra le due scuole (statistica Bayesiana e statistica frequentista) nasce dall'interpretazione che si dà al teorema stesso

Le applicazioni del teorema sono innumerevoli, come ad esempio nella realizzazione di sistemi di filtraggio impiegati nella lotta contro lo spam.

Il **teorema di Bayes** deriva da tre teoremi fondamentali delle probabilità: il teorema della probabilità condizionata, il teorema della probabilità composta ed il teorema della probabilità assoluta.

Questi tre teoremi dicono rispettivamente che:

- teorema della probabilità condizionata:

$$P(A | B) = \frac{P(A \cap B)}{P(B)}$$

la probabilità di A condizionata da B è definita come la probabilità che si verifichi l'evento A a condizione che si verifichi pure l'evento B , entrambi eventi dello spazio S di cui B tale che $P(B) > 0$.

- teorema della probabilità composta:

$$P(A \cap B) = P(B) P(A | B) = P(A) P(B | A)$$

per cui la probabilità che due eventi si verifichino contemporaneamente è pari alla probabilità di uno dei due eventi moltiplicato con la probabilità dell'altro evento condizionato dal verificarsi del primo .

- Il teorema della probabilità assoluta:

afferma che se A_1, \dots, A_n formano una partizione dello spazio di tutti gli eventi possibili Ω (ossia $A_i \cap A_j = \emptyset \forall i, j$ e $\bigcup_{i=1}^n A_i = \Omega$) e B è un qualsiasi evento, allora:

$$P(B) = \sum_{i=1}^n P(A_i) P(B | A_i)$$

2.3. Il Teorema di Bayes

il Teorema di Bayes nella sua forma più semplice può essere scritto nel seguente modo:

$$P(B | A) = \frac{P(A | B)P(B)}{P(A)}$$

Gli eventi A e B possono essere eventi qualunque, contemporanei oppure aventi luogo in tempi diversi. Per fissare le idee è però estremamente utile utilizzare come esempio il caso di due eventi legati da una relazione di causa-effetto: ci chiederemo quindi qual è dato un certo effetto, la probabilità che sia prodotto da una data causa. Con l'uso delle lettere D e H per indicare i dati e l'ipotesi in esame il teorema diventa

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

Nel caso in cui non sia disponibile direttamente P(D) si ricorre alla regola delle alternative, per esempio nella forma

$$P(D) = P(D \cap H) + P(D \cap \bar{H})$$

o nella forma

$$P(D) = P(D | H)P(H) + P(D | \bar{H})P(\bar{H})$$

Nel primo caso il teorema di Bayes assume la forma

$$P(H | D) = \frac{P(D | H) \cdot P(H)}{P(D \cap H) + P(D \cap \bar{H})}$$

nel secondo invece

$$P(H | D) = \frac{P(D | H) \cdot P(H)}{P(D | H)P(H) + P(D | \overline{H})P(\overline{H})}$$

Preso il teorema di Bayes nella sua forma più classica può risultare molto importante discutere l'interpretazione di ciascun termine del teorema di Bayes .

- $P(H)$ è il nostro grado di fiducia nell'affermazione H prima di prendere in considerazione i dati ed è nota come probabilità a priori per H , o grado di fiducia a priori in H . Per brevità è detta prior.
- $P(H|D)$ riflette invece la fiducia a posteriori, aggiornata alla luce dei dati e calcolata mediante l'espressione del teorema di Bayes. Per brevità è detta posterior.
- La parte destra dell'espressione del teorema di Bayes include la prior, così abbastanza naturalmente la posterior è proporzionale alla prior. La parte destra dell'equazione include anche $P(D|H)$, che è nota come la verosimiglianza dei dati, o *likelihood*, e rappresenta la probabilità che si verifichino i dati sotto l'ipotesi che H sia vera. Per calcolare la *likelihood* dobbiamo disporre di un modello probabilistico che connetta la proposizione H cui siamo interessati con i dati osservati D , questo è il cuore dell'apprendimento probabilistico.
- Il denominatore della frazione è un termine di normalizzazione e ci garantisce di tenere in conto adeguatamente l'ipotesi alternativa (non H), infatti se prima di osservare i dati dobbiamo garantire che $P(H)+P(\text{non } H)=1$, dopo aver visto i dati dobbiamo garantire che $P(H|D)+P(\text{non } H|D)=1$.

Il fatto che $P(D) = P(D|H)P(H) + P(D|\text{non } H)P(\text{non } H)$ fa sì che di fatto la probabilità a posteriori $P(H|D)$ dipenda dal peso relativo delle due ipotesi concorrenti H e $(\text{non } H)$.

Riscrivendo il teorema nel modo seguente (abbiamo diviso sopra e sotto per $P(D|H)P(H)$)

$$P(H | D) = \frac{1}{1 + \frac{P(D | \bar{H})P(\bar{H})}{P(D | H)P(H)}}$$

vediamo ancora più chiaramente che più grande è $P(D|H)P(H)$ rispetto al termine concorrente $P(D|\text{non } H)P(\text{non } H)$ e più grande sarà la posterior.

2.4. Esempi di applicazione del teorema di Bayes

In questa sezione vengono presentati dei semplici esempi che servono per far intuire immediatamente l'importanza dei concetti appena descritti:

ES 1) Individuare la probabilità di una meningite

- Conoscenza pregressa:

Un dottore sa che la meningite causa rigidità del collo per il 50% dei casi $\rightarrow P(\text{rigidità del collo} | \text{meningite}) = 1/2$

La probabilità incondizionata che un paziente possa avere la meningite è $\rightarrow P(\text{meningite}) = 1/50000 = 0,00002$

La probabilità incondizionata che un paziente possa avere rigidità del collo è $\rightarrow P(\text{rigidità del collo}) = 1/20 = 0,05$

La domanda che quindi viene fatta è la seguente:

- Se un paziente ha rigidità del collo, qual è la probabilità che egli abbia la meningite?

Dal teorema di Bayes sappiamo che:

$$P(C | A) = \frac{P(A | C)P(C)}{P(A)}$$

Quindi sostituendo otteniamo:

$$P(M | R) = \frac{P(R | M)P(M)}{P(R)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

La Probabilità che quindi un paziente che ha rigidità del collo abbia la meningite è uguale a 0,0002.

ES 2) Individuare la probabilità che l'esame svolto sia quello d'Inglese

- Conoscenza pregressa:

In una certa facoltà universitaria, è obbligatorio sostenere un esame di Lingua Straniera. Ogni studente può scegliere fra: Inglese, Francese, Spagnolo, Tedesco. Le statistiche dicono che le probabilità di scelta sono rispettivamente:

$$0,4 \quad 0,3 \quad 0,2 \quad 0,1$$

D'altra parte, per la diversa difficoltà dei corsi e severità degli insegnanti, le probabilità di riportare la massima votazione (30/30) variano da lingua a lingua e sono rispettivamente:

$$0,1 \quad 0,2 \quad 0,3 \quad 0,9$$

Supponiamo di sapere che un certo studente ha riportato 30/30 nell'esame di Lingua.

La domanda che quindi viene fatta è la seguente:

- Che probabilità c'è che la materia d'esame sia stata Inglese?

$$\begin{aligned}
 p(\text{Inglese}/"30") &= \\
 &= \frac{p(I) \cdot p("30"/I)}{p(I) \cdot p("30"/I) + p(F) \cdot p("30"/F) + p(S) \cdot p("30"/S) + p(T) \cdot p("30"/T)} = \\
 &= \frac{0,4 \cdot 0,1}{0,4 \cdot 0,1 + 0,3 \cdot 0,2 + 0,2 \cdot 0,3 + 0,1 \cdot 0,9} = \frac{0,04}{0,04 + 0,06 + 0,06 + 0,09} = \frac{0,04}{0,25} = \frac{4}{25} = 0,16
 \end{aligned}$$

La Probabilità che quindi la materia d'esame sia inglese è uguale a 0,16

2.5. L' Apprendimento bayesiano

L'apprendimento bayesiano si pone come obiettivo il problema di fare delle previsioni e ritiene il problema della formulazione di ipotesi a partire dai dati, come un suo sottoproblema. Un modo per specificare che cosa intendiamo per la migliore ipotesi è quello di affermare che la migliore ipotesi è quella più probabile, avendo a disposizione dei dati ed una certa conoscenza iniziale delle probabilità a priori delle varie ipotesi. Le ipotesi elaborate dai dati e combinate in modo opportuno, portano alla formulazione di una previsione. I coefficienti con cui vengono pesate le ipotesi non sono altro che la loro verosimiglianza con i dati. Il metodo bayesiano non sceglie tra un insieme di ipotesi, ma le combina in base alla loro capacità di rappresentare i dati. In modo più formale, sia D un insieme di dati, sia X una quantità ignota su cui vogliamo fare delle previsioni e siano H_1, \dots, H_n delle ipotesi; l'apprendimento bayesiano assume la forma:

$$P(X|D) = \sum_{i=1}^n P(X|D, H_i)P(H_i|D) = \sum_{i=1}^n P(X|H_i)P(H_i|D)$$

Si può dimostrare che nessun altro metodo di predizione che fa uso dello stesso spazio di ipotesi e della stessa conoscenza a priori, ha in media delle prestazioni migliori. Un aspetto curioso dell'apprendimento bayesiano è che le predizioni fatte possono non corrispondere a nessuna ipotesi in particolare, in quanto la previsione è una combinazione lineare delle ipotesi formulate. Purtroppo l'apprendimento bayesiano richiede il calcolo di

$$P(H_i|D)$$

per tutte le H_i e nella maggior parte dei casi questo è un problema intrattabile. Si fanno allora delle ipotesi semplificative tra cui la più comune è quella di considerare solo l'ipotesi più probabile, ovvero di considerare solo l'ipotesi H_i che massimizza $P(H_i|D)$. Questa ipotesi viene chiamata massima a posteriori o MAP (maximum a posteriori) e si indica con H_{MAP} . Quindi:

$$P(X|D) \approx P(X|H_{MAP})$$

Per calcolare H_{MAP} si ricorre al teorema di Bayes che fornisce un metodo diretto per calcolare la probabilità di una ipotesi partendo dalla sua probabilità a priori, dalla probabilità di osservare certi dati data l'ipotesi stessa e dai dati osservati:

$$H_{MAP} = \arg \max_{H_i} P(H_i|D) = \arg \max_{H_i} \frac{P(D|H_i)P(H_i)}{P(D)}$$

Da notare che la probabilità a posteriori $P(H_i|D)$ riflette l'influenza dei dati, in contrasto con la probabilità a priori $P(H_i)$ che ne è indipendente. Dunque

$$H_{MAP} = \arg \max_{H_i} P(D | H_i) P(H_i)$$

Ci sono state molte discussioni su come scegliere la probabilità a priori di un modello: un metodo è quello di privilegiare le ipotesi più semplici rispetto a quelle più complesse (rasoio di Ockham) e di fare in modo di rispettare il vincolo di normalizzazione sull'intero spazio delle ipotesi. La preferenza per le ipotesi più semplici garantisce una certa immunità al rumore e al sovraddestramento, ma tale preferenza non deve essere troppo spinta per evitare di andare incontro ad un sottoaddestramento, cioè ad una situazione in cui molti dati vengono ignorati. Molto spesso, per mancanza di informazioni o per semplicità di calcolo, si assume un prior uniforme sulle ipotesi e si sceglie quindi l'ipotesi H_i che massimizza la $P(D | H_i)$: tale ipotesi viene chiamata a massima verosimiglianza o ML (maximum likelihood) e indicata con HML:

$$H_{ML} = \arg \max_{H_i} P(D | H_i)$$

2.6. Le reti Bayesiane

In questo paragrafo viene fatta una breve introduzione al concetto di rete Bayesiana che verrà poi sfruttato nel paragrafo successivo.

Una rete bayesiana è un modello grafico probabilistico per rappresentare le relazioni tra un insieme molto vasto di variabili aleatorie che codifica in modo molto efficiente la distribuzione congiunta di probabilità, sfruttando le relazioni di indipendenza condizionale tra le variabili. Sia $X = \{ X_1, \dots, X_n \}$ l'insieme delle variabili aleatorie che descrivono il dominio di interesse. In modo formale, una rete bayesiana è un grafo per cui valgono le seguenti proprietà:

- il grafo $G = (V, \vec{E})$ è un grafo diretto aciclico o DAG³;
- ciascun nodo del grafo rappresenta una variabile aleatoria X_i ;
- ogni variabile aleatoria può assumere un numero finito di stati mutuamente esclusivi;
- ciascun arco orientato che connette due nodi X_i, X_j del grafo con verso entrante in X_j , indica un'influenza diretta di X_i su X_j ;
- la mancanza di un arco tra due nodi X_i, X_j codifica l'indipendenza condizionale delle variabili associate ai due nodi;
- a ciascuna variabile X_i con genitori⁴ $Pa_i = \{Y_1, \dots, Y_m\}$, è associata una tabella di probabilità condizionali

$$P(X_i | Y_1, \dots, Y_m) = P(X_i | Pa_i)$$

che codifica gli effetti che i genitori hanno su quel nodo;

- ciascun nodo è condizionatamente indipendente dai suoi predecessori (esclusi i genitori), dati i suoi parent.

Per poter valutare ogni probabilità di interesse, è sufficiente conoscere la distribuzione congiunta di probabilità dell'insieme di variabili aleatorie $X = \{X_1, \dots, X_n\}$ che descrivono il dominio di interesse. Usando la regola del prodotto (chain rule) per cui $P(X, Y) = P(X | Y) P(Y)$, si ottiene che:

$$P(\mathbf{x}) = P(x_1, \dots, x_n) = P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1}, \dots, x_1)$$

Applicando la stessa regola al termine $P(x_{n-1}, \dots, x_1)$, si fattorizza la probabilità congiunta di tutte le variabili:

$$\begin{aligned} P(\mathbf{x}) &= P(x_1, \dots, x_n) \\ &= P(x_n | x_{n-1}, \dots, x_1) P(x_{n-1} | x_{n-2}, \dots, x_1) \cdots P(x_2 | x_1) P(x_1) \end{aligned}$$

In forma compatta:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1)$$

Le condizioni di indipendenza condizionale ci consentono di semplificare la probabilità congiunta $P(x_1, \dots, x_n)$. Supponendo che i nodi del grafo e quindi le variabili associate siano numerati secondo l'ordinamento topologico indotto dalla struttura del grafo orientato, si ha che

$$x_i \perp \{x_{i-1}, \dots, x_1\} | \text{pa}_i$$

Sostituendo l'equazione precedente in questa, si giunge alla fattorizzazione della probabilità congiunta:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{pa}_i)$$

Questa relazione è di fondamentale importanza in quanto permette di esprimere la probabilità congiunta di tutte le variabili, come il prodotto delle probabilità condizionali locali limitate alle sole variabili che hanno un'influenza diretta sulla variabile in esame. In un sistema localmente strutturato, cioè in un grafo sparso, ogni sottocomponente interagisce direttamente solo con un numero limitato di altri componenti, indipendentemente dal numero totale dei componenti. La struttura locale è solitamente associata a crescite lineari invece che esponenziali della complessità. La distribuzione congiunta assume quindi una forma compatta che evita una crescita esponenziale della dimensione della tabella di probabilità associata.

Per costruire una rete bayesiana di un dominio di interesse, si può procedere nel seguente modo:

- si individua un insieme di variabili aleatorie $X = \{ X_1, \dots, X_n \}$ che descrivono il dominio;
- si sceglie un ordinamento per le variabili in X ;
- finché ci sono rimaste delle variabili:
 - (a) si prende una variabile X_i e si aggiunge un nodo alla rete;
 - (b) si pone P_{X_i} come un qualche insieme minimo di nodi già presenti nella rete, tale che la proprietà di indipendenza condizionale sia soddisfatta;
 - (c) si definisce la tabella delle probabilità condizionate per X_i .

Qualunque ordinamento delle variabili garantisce la rappresentazione del dominio. Tuttavia l'ordinamento delle variabili è cruciale, perchè può indurre a ricercare condizioni di indipendenza non facilmente esplicitabili. Inoltre la topologia finale della rete dipende da tale ordinamento. Per questo, l'ordine corretto per aggiungere i nodi è quello che prevede prima l'inserimento delle cause alla radice, quindi delle variabili che influenzano e così via fino ad arrivare alle foglie che non hanno nessuna influenza causale sulle altre variabili. Per prime devono dunque essere scelte quelle variabili con una probabilità a priori nota. Inoltre, poiché ogni nodo è connesso solo ai nodi precedenti, questo metodo di costruzione garantisce che la rete sia aciclica.

2.7. Il Classificatore Naive Bayes

Di recente sono stati studiati vari classificatori bayesiani con dei risultati piuttosto sorprendenti: nonostante la loro estrema semplicità, riescono a raggiungere delle prestazioni addirittura superiori rispetto ai più noti algoritmi di apprendimento. Queste metodologie di tipo probabilistico fanno delle forti assunzioni sulla generazione dei dati ed utilizzano un modello probabilistico che ingloba queste assunzioni. Avvalendosi di un insieme di dati etichettati per

l'addestramento, si stimano i parametri del modello generativo e si classificano le nuove istanze utilizzando il teorema di Bayes e selezionando la classe o categoria che ha la probabilità più alta di aver generato l'esempio.

Il classificatore Naive Bayes è un metodo di apprendimento bayesiano che si è rivelato utile in molte applicazioni, tra cui la classificazione di documenti testuali. E' uno tra i più semplici di questi modelli, come dice il nome stesso ed è basato sull'assunzione semplificativa che tutti gli attributi che descrivono una certa istanza sono tra loro condizionatamente indipendenti data la categoria a cui appartiene l'istanza. Questa affermazione viene detta assunzione del Naive Bayes. Quando questa ipotesi è verificata, il Naive Bayes esegue una classificazione di tipo *MAP*. Nonostante questa assunzione sia violata nella maggior parte dei problemi reali come, ad esempio, nella categorizzazione del testo, il Naive Bayes si comporta molto bene e risulta essere molto efficace. L'assunzione di indipendenza permette di apprendere separatamente i parametri di ogni attributo, semplificando molto l'apprendimento, specialmente in quelle situazioni in cui il numero di attributi è molto elevato ed in cui i dati a disposizione non sono molto numerosi. La classificazione di documenti è uno di quei domini con un grande numero di attributi. Una scelta comune associa gli attributi con le parole del documento e si capisce quindi che il loro numero può diventare molto elevato. Alcune tecniche di apprendimento riducono drasticamente la taglia del vocabolario, in modo da avere pochi attributi da gestire. Purtroppo nel campo della categorizzazione del testo solo un esiguo numero di termini è irrilevante ai fini della classificazione e quindi un'eccessiva diminuzione porta ad un deterioramento delle prestazioni. La soluzione consiste nell'impiegare delle tecniche in grado di elaborare un numero elevato di attributi.

Il dominio applicativo del classificatore Naive Bayes riguarda la classificazione di istanze che possono essere descritte mediante un insieme di attributi di cardinalità anche molto elevata. Innanzitutto si devono stimare i parametri del modello utilizzando i dati a disposizione. La stima di questi parametri

corrisponde ad aver identificato un modello tra tutti quelli presenti nello spazio delle ipotesi. A differenza di altri algoritmi di apprendimento, non c'è un'esplicita ricerca nello spazio delle possibili ipotesi, ma l'ipotesi viene definita semplicemente contando la frequenza degli attributi negli esempi di addestramento. Il modello così definito permette di classificare le nuove istanze che gli vengono presentate. Sia $A = \{A_1, \dots, A_n\}$ un insieme di variabili che modellano gli attributi delle istanze da classificare e sia $C = \{C_1, \dots, C_{|C|}\}$ una variabile i cui stati rappresentano le categorie a cui appartengono le istanze. Si tratta di stimare una funzione che, applicata ad ogni istanza, fornisca la classe di appartenenza dell'istanza stessa, partendo da un insieme di dati di addestramento D in cui ogni elemento $d_i = \{a_1, \dots, a_n\}$ è descritto tramite i valori dei suoi attributi e tramite la sua classe c_i . Vediamo innanzitutto come può essere classificata una nuova istanza partendo dalla sua descrizione in termini di attributi. Secondo l'approccio bayesiano, si calcolano le probabilità a posteriori $P(c_j | d_i)$ con $j = 1, \dots, |C|$ e si determina la categoria c_j che massimizza tale probabilità:

$$c_{MAP} = \arg \max_{c_j \in C} P(c_j | d_i) = \arg \max_{c_j \in C} P(c_j | a_1, \dots, a_n)$$

Il teorema di Bayes permette di esprimere la probabilità a posteriori in funzione della verosimiglianza e della probabilità a priori:

$$\begin{aligned} c_{MAP} &= \arg \max_{c_j \in C} \frac{P(d_i | c_j) P(c_j)}{P(d_i)} \\ &= \arg \max_{c_j \in C} \frac{P(a_1, \dots, a_n | c_j) P(c_j)}{P(a_1, \dots, a_n)} \\ &= \arg \max_{c_j \in C} P(a_1, \dots, a_n | c_j) P(c_j) \end{aligned}$$

Abbiamo tralasciato il termine $P(d_i) = P(a_1, \dots, a_n)$ che rappresenta una costante di normalizzazione ed è quindi ininfluenza ai fini della massimizzazione.

Supponiamo per semplicità che tutti gli attributi $A_i \in A$ che descrivono un'istanza siano realizzazioni diverse della stessa variabile aleatoria caratterizzata da m stati distinti e cerchiamo di stimare i due termini dell'equazione precedente. Il termine $P(c_j)$ può essere stimato facilmente contando il numero di volte che la classe c_j compare tra i dati. La stima del termine $P(a_1, \dots, a_n | c_j)$ è più difficoltosa, perché il numero di configurazioni che può assumere è $m^n |C|$, dove n è il numero degli attributi. Per avere una stima attendibile, servirebbe un numero di dati che è esponenziale nel numero di attributi. Per ridurre il numero dei parametri da stimare e corrispondentemente il numero di dati necessari, si fa l'ipotesi semplificativa che gli attributi siano mutuamente indipendenti data la classe dell'istanza:

$$P(a_1, \dots, a_n | c_j) = P(a_1 | c_j) \cdots P(a_n | c_j) = \prod_{i=1}^n P(a_i | c_j)$$

Sostituendo quest'ultima equazione alla precedente, si ottiene:

$$c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_{i=1}^n P(a_i | c_j)$$

Se è soddisfatta l'ipotesi di indipendenza degli attributi, il classificatore Naive Bayes esegue una stima MAP, quindi $c_{NB} = c_{MAP}$. Il numero dei termini da stimare si riduce a $n |C|$, che è lineare nel numero di attributi. Il numero totale dei parametri da stimare è $|C| + n |C| = (n + 1) |C|$.

Poiché difficilmente nei casi reali vale l'ipotesi di indipendenza tra gli attributi, alcuni studi hanno rilassato l'ipotesi, aggiungendo la presenza di archi tra gli attributi. Per stimare le probabilità di interesse, si ricorre ancora una volta all'approccio bayesiano che permette di combinare la probabilità a priori con i dati, utilizzando l' m -stima di probabilità:

$$\frac{n_c + mp}{n + m}$$

dove i termini n_c e n permettono di stimare la quantità di interesse come numero di occorrenze tra i dati, p è la nostra stima a priori della probabilità che vogliamo determinare ed m è una costante chiamata taglia del campione equivalente che determina il peso relativo di p e dei dati osservati. Il nome di m è dovuto al fatto che l'equazione può essere interpretata come l'aver aggiunto m dati virtuali distribuiti secondo p agli n dati reali, pesando in modo relativo la conoscenza a priori e l'informazione presente nei dati. Se non disponiamo di nessuna informazione a priori, p può essere scelto in modo uniforme: se un attributo assume k valori, allora $p = 1/k$. Se $m = 0$, allora l' m -stima coincide con n_c/n ed utilizza solo l'informazione presente nei dati, mentre se m e n sono entrambi diversi da zero, l'informazione a priori p e la stima n_c/n ricavata dai dati sono combinate in accordo ad m .

Il classificatore Naive Bayes e più in generale il problema della classificazione, possono essere associati ad una rete bayesiana che codifica le relazioni di indipendenza presenti tra le variabili che descrivono il dominio. Nel caso del Naive Bayes, le variabili di interesse sono gli attributi $A = \{A_1, \dots, A_n\}$ che descrivono le istanze da classificare e la categoria $C = \{c_1, \dots, c_{|C|}\}$ i cui stati rappresentano le classi delle istanze. Sia quindi $U = \{A \cup C\}$ l'universo di variabili della rete bayesiana. La distribuzione congiunta globale $P(U)$ viene codificata nella topologia della rete, mediante la presenza o la mancanza di archi orientati tra i nodi del grafo. Innanzitutto, come si può vedere dall'equazione,

$$c_{MAP} = \arg \max_{c_j \in C} P(a_1, \dots, a_n | c_j) P(c_j)$$

la variabile che rappresenta la categoria ha un'influenza diretta sull'istanza: supponendo di modellare l'istanza con una variabile X che condensa tutti gli attributi, la topologia della rete assume la forma:



Figura 2 Modellazione della Relazione tra Classe e Istanza.

La direzione dell'arco, che va dalla classe verso l'istanza, indica che l'istanza è l'effetto di appartenere ad una certa categoria e quindi la categoria ha un'influenza diretta sulla generazione dell'istanza.

Le tabelle di probabilità $P(C)$ e $P(X | C)$ che rappresentano i parametri della rete, non sono altro che i termini che compaiono nell'equazione precedente. Se l'istanza viene descritta mediante un insieme $A = \{A_1, \dots, A_n\}$ di attributi che sono istanze della stessa variabile, la rete può essere rappresentata come in figura:

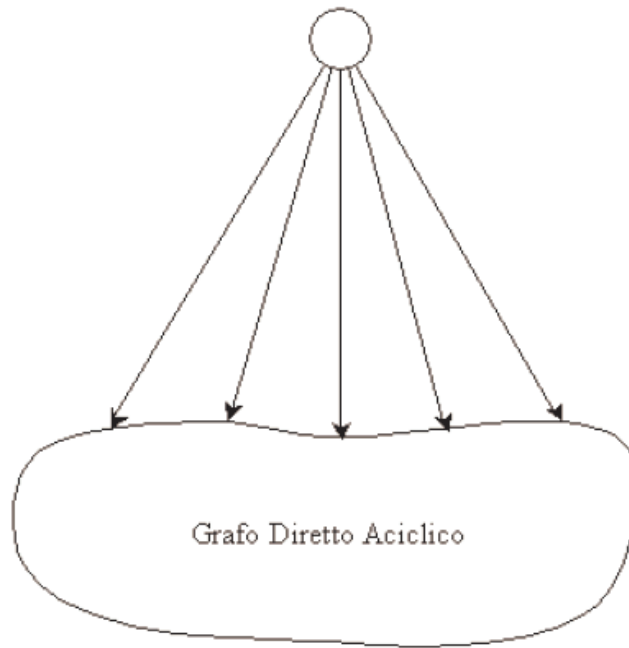


Figura 3 Rete Bayesiana come Modello Generativo per i Dati.

Dove la curva chiusa contiene un grafo diretto aciclico i cui vertici sono gli attributi e in cui non viene fatta nessuna ipotesi riguardo alla connettività: al limite, si può trattare di un grafo completo. Infine, se si fa l'ipotesi semplificativa del Naive Bayes per cui gli attributi sono tra loro indipendenti data la classe, si giunge alla **rete bayesiana del classificatore Naive Bayes** riportata in figura:

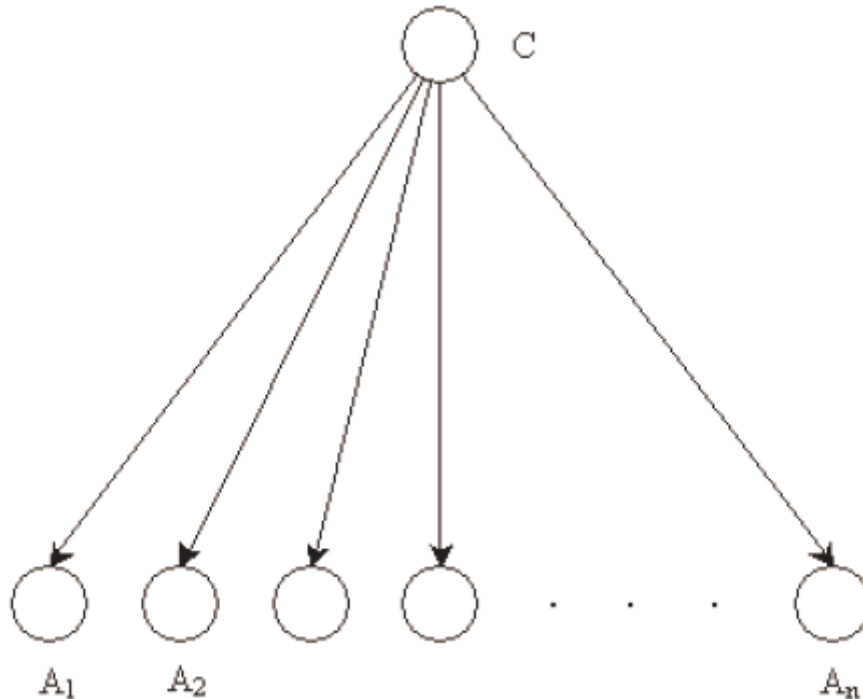


Figura 4 Rete Bayesiana del Classificatore Naive Bayes.

La direzione degli archi, che va dalla classe verso gli attributi, indica che le caratteristiche di ciascuna istanza sono l'effetto di appartenere ad una certa categoria, per cui gli attributi corrispondenti e quindi tutta l'istanza, sono generati da una certa classe.

La semantica della rete bayesiana relativa all'indipendenza condizionale impone che le variabili che hanno cause in comune siano indipendenti data la classe. Da notare che non vi sono archi tra gli attributi come conseguenza dell'ipotesi semplificativa. L'apprendimento e la classificazione del Naive Bayes possono essere interpretati alla luce della corrispondente rete bayesiana. I parametri della rete da apprendere sono le probabilità $P(c_j)$ e $P(a_i | c_j)$. Infatti i parametri $P(a_i | c_j)$ sono comuni a tutti gli attributi: è come stimare la probabilità che lanciando una moneta esca testa o croce, avendo a disposizione più monete identiche che vengono ripetutamente lanciate. Una generica istanza descritta dai valori dei suoi attributi e dalla corrispondente classe, può essere vista come l'assegnazione dell'evidenza a tutte le variabili presenti nella rete e costituisce uno dei dati completi che fanno parte dell'insieme di addestramento

L'apprendimento non è quindi altro che la stima dei valori delle tabelle di probabilità associate con i nodi della rete.

Il problema della classificazione può invece essere interpretato come un caso particolare di inferenza nella rete. In questa circostanza si tratta di assegnare l'evidenza agli attributi che rappresentano l'istanza (nodi in grigio) e di propagarla verso il nodo che modella la classe: si calcola quindi la probabilità di ogni categoria, data una particolare realizzazione dell'istanza, mediante la procedura di inferenza nella rete.

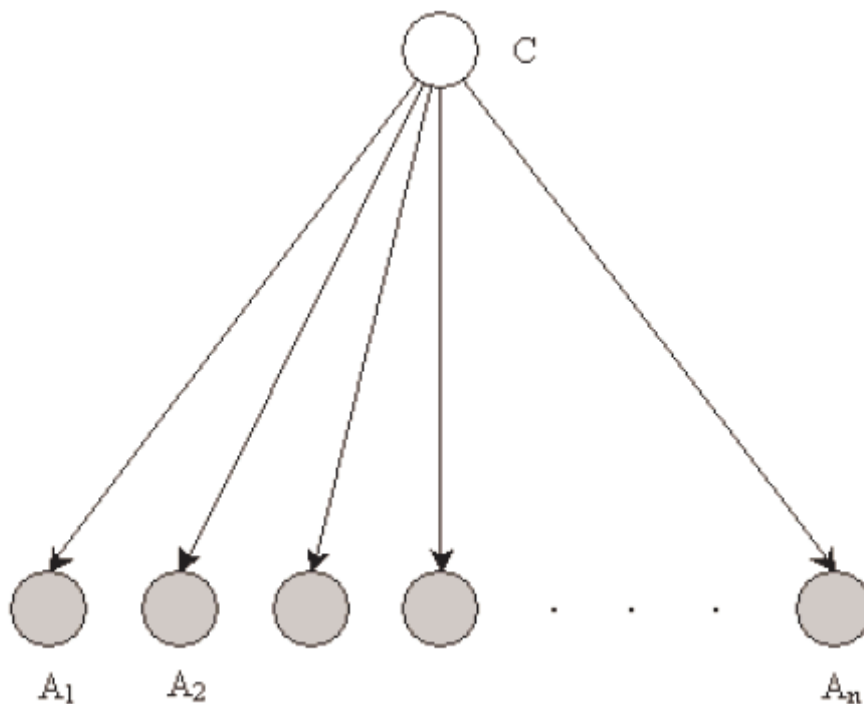


Figura 5 Classificatore Naive Bayes con Evidenza sugli Attributi.

In accordo alla topologia della rete della figura precedente, la distribuzione di probabilità congiunta $P(\mathbf{U})$ può essere fattorizzata nel seguente modo:

$$P(\mathbf{U}) = P(C) \prod_{i=1}^n P(A_i | C)$$

2.8. Esempi di applicazione dell' Algoritmo Naive Bayes

In questa ultima sezione viene presentato un esempio sull'uso dell'algoritmo Naive Bayes; lo scopo di questo esempio è quello di fornire un'applicazione pratica a tutti i concetti fin'ora illustrati.

L'obiettivo di questo esempio è quello di determinare se è possibile o meno giocare una partita di tennis ogni sabato mattina, in base ai dati meteorologici del giorno stesso. Nella seguente tabella sono presenti una serie di dati raccolti nei giorni precedenti:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figura 6 tabella con i dati meteorologici

Ogni giorno è descritto dai seguenti attributi: *Outlook* (tempo), *Temperature* (temperatura), *Humidity* (umidità), e *Wind* (vento).

Utilizzeremo quindi il classificatore Naive Bayes e l'insieme di dati in ingresso di questa tabella per classificare la seguente istanza:

< *Outlook=sunny, Temperature = cool, Humidity=high, Wind=strong* >

L'obiettivo è quello di predire il valore finale (yes or no) ossia determinare se è possibile o meno giocare una partita di tennis per la suddetta istanza.

Sappiamo dalla teoria che :

$$v_{NB} = \operatorname{argmax}_{v_j \in \{yes, no\}} P(v_j) \prod_i P(a_i | v_j)$$

E quindi:

$$= \operatorname{argmax}_{v_j \in \{yes, no\}} P(v_j)$$

$$P(\text{Outlook} = \text{sunny} | v_j) P(\text{Temperature} = \text{cool} | v_j)$$

$$P(\text{Humidity} = \text{high} | v_j) P(\text{Wind} = \text{strong} | v_j)$$

È importante notare che a_i è stato istanziato usando il valore specifico dell'attributo. Per calcolare v_{NB} noi ora dobbiamo calcolare le 10 probabilità che possono essere stimate partendo dai dati in ingresso; per prima cosa si devono trovare le probabilità dei differenti risultati finali basandoci sulla loro frequenza negli esempi:

$$P(\text{PlayTennis} = \text{yes}) = 9/14 = 0.64$$

$$P(\text{PlayTennis} = \text{no}) = 5/14 = 0.36$$

Nello stesso modo si possono calcolare le probabilità condizionali, ad esempio per $\text{Wind} = \text{Strong}$ sono:

$$P(\text{Wind} = \text{Strong} | \text{PlayTennis} = \text{yes}) = 3/9 = 0.33$$

$$P(\text{Wind} = \text{Strong} | \text{PlayTennis} = \text{no}) = 3/5 = 0.60$$

Utilizzando queste probabilità appena calcolate, e tutte le rimanenti che si calcolano esattamente del medesimo modo, riusciamo a calcolare v_{BN} :

$$P(\text{yes}) P(\text{ sunny } | \text{ yes }) P (\text{ cool } | \text{ yes }) P (\text{ high } | \text{ yes }) P (\text{ strong } | \text{ yes }) = 0.0053$$

$$P(\text{no}) P(\text{ sunny } | \text{ no }) P (\text{ cool } | \text{ no }) P (\text{ high } | \text{ no }) P (\text{ strong } | \text{ no }) = 0.0206$$

Normalizzando le sopraindicate quantità sommate una per una possiamo calcolare la probabilità condizionale che il risultato finale sia: “no” dati i risultati dei dati osservati.

Quindi per l’esempio corrente la probabilità è:

$$\frac{.0206}{.0206+.0053} = .795$$

A questo punto abbiamo quindi calcolato la probabilità utilizzando la frazione dei tempi dell’evento osservato in particolare: n_c / n dove $n=5$ è il numero totale degli esempi di “training” per i quali $PlayTennis = no$ e $n_c = 3$ è il numero di queste per i quali $Wind = strong$. Questa frazione da buoni risultati in molti casi ma se n_c è molto piccolo i risultati non sono soddisfacenti. Per superare questo inconveniente è utilizzato l’approccio Bayesiano, illustrato nei paragrafi precedenti, per stimare la probabilità, usando la *m-stima* così calcolata:

$$\frac{n_c + mp}{n + m}$$

Dove n e n_c sono definite come sopra e p è la stima *prior* della probabilità che vogliamo determinare e m è una costante chiamata *quantità semplice*

equivalente. Un tipico metodo per la scelta di p in assenza di altre informazioni è la quella di assumere uniforme *prior*; così se un attributo ha k possibili valori si setta $p = 1/k$.

Quindi nell'esempio precedente nello stimare

$$P(Wind = Strong | PlayTennis = no)$$

Si nota che l'attributo *Wind* ha due possibili valori, le uniformi *priors* corrisponderebbe allo scegliere $p = 0.5$.

È bene osservare che nel caso in cui $m = 0$, la *m-stima* è equivalente alla semplice frazione n_c / n e che se sia m che n sono diversi da zero allora la frazione n_c / n e *prior* p possono essere combinate ad il peso m .