CLASSIFICAZIONE AUTOMATICA DI TESTI

Seminario del corso di Elaborazione del Linguaggio Naturale A.A. 2002/03

Studente: Filippo Bonchi
Professore:Amedeo Cappelli

Da "Artificial Intelligence. A Modern Approach"

"Le applicazioni di successo che usano il linguaggio naturale condividono due proprieta': si concentrano su di un dominio particolare, piuttosto che permettere la discussione su qualsiasi argomento e si concentrano su di un obbiettivo in particolare, piuttosto che cercare di comprendere il linguaggio nella sua completezza"

S.J. Russell e P. Norvig

Da "Artificial Intelligence. A Modern Approach"

"Vedremo 5 obbiettivi:

1. Traduzione automatica
2. Accesso alle basi di dati
3. Recupero di informazione
4. Classificazione di testi
5. Estrazione di dati

"

S.J. Russell e P. Norvig

Precisazioni…

- Spesso il termine Classificazione di Testi viene usato per indicare task diversi:
  1. L'assegnamento di un documento in una categoria presa da un insieme predefinito (uso proprio del termine)
  2. L'identificazione di un tale insieme di categorie
  3. L'identificazione di tale insieme di categorie e il raggruppamento dei documenti sotto tale insieme (Text Clustering)
  4. Ogni attivita' di inserire documenti di testo dentro gruppi

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificaztori
  3. Valutazione dell'efficacia
- Conclusioni

Una definizione formale

Sia $\boldsymbol{D}$ il dominio dei documenti

Sia $\boldsymbol{C} = \{c_1,\ldots,c_{|C|}\}$ un insieme di categorie predefinite

Il task della classificazione di testi e' di approssimare la funzione target sconosciuta

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

Con una funzione $\Phi : \boldsymbol{D} \times \boldsymbol{C} \rightarrow \{T,F\}$ chiamata classificatore tale che Φ coincida il piu' possibile con Φ^I

Una definizione formale

- Si assume che:
  1. Le categorie sono soltanto etichette simboliche, e non e' disponibile la conoscenza del loro significato per aiutarsi nel costruire il classificatore
  2. Non e' disponibile conoscenza esogena (data di pubblicazione, autore, tipo di documento). Un documento viene classificato soltanto dal suo contenuto.
- Queste assunzioni ci permettono di fare un discorso del tutto generale.
- Chiaramente per implementare un'applicazione e' legittimo usare tutta l'informazione disponibile

Inter-indexer inconsistency

- Visto che la classificazione di un testo si basa sulla semantica, e dato che la semantica di un documento e' una nozione soggettiva ne segue che l'appartenenza di un documento a una categoria non puo' essere decisa deterministicamente
- Questo e' esemplificato dal *Inter-indexer inconsistency*: quando due esperti umani decidono se classificare un documento d_j sotto c_i, si possono trovare in disaccordo, e cio' avviene con una frequenza abbastanza alta

Single Label, Multi Label e Binary

- Al Task della Classificazione si possono aggiungere dei vincoli: per un dato k, esattamente k, (o ≤k,o≥k) elementi di $\boldsymbol{C}$ vengano assegati ad un documento d_j
- Single Label: soltanto una categoria puo' essere assegnata a un documento (k=1)
- Multi Label: 0 o + categorie possono essere assegnate a un documento
- Binary: un documento o appartiene a c_i o appartiene a τc_i

Single Label, Multi Label e Binary

- Un algoritmo per binary puo' essere usato anche per multilabel: si trasforma il problema di classificare sotto le categorie $\{c_1,\dots c_{|C|}\}$ in $|\boldsymbol{C}|$ problemi indipendenti di classificazione binaria sotto le categorie $\{c_i, \tau c_i\}$
- Un classificatore per una categoria c_i e' una funzione $\Phi_i : \boldsymbol{D}\rightarrow\{T,F\}$ che approssima la funzione target sconosciuta

$$\Phi^I_i : \boldsymbol{D}\rightarrow\{T,F\}$$

Classificazione DOCUMENT PIVOTED E CATEGORY PIVOTED

- DOCUMENT PIVOTED CATEGORIZATION: Dato un documento vogliamo trovare tutti le categorie sotto il quale puo' essere classificato
- CATEGORY PIVOTED CLASSIFICATION: Data una categoria vogliamo trovare tutti i documenti che possono essere classificati sotto di essa

Classificazione Hard e Ranking

- Hard: il classificatore Φ_i restituisce un valore booleano
- Ranking: il classificatore Φ_i restituisce un valore [0,1]
- Sistemi di Classificazione semi-automatici: esperti umano che classificano aiutandosi con classificatori ranking

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

APPLICAZIONI

- Indicizzazione automatica per sistemi di IR

Ad ogni documento e' assegnata 1 o piu' parole-chiavi (descriventi il suo contenuto) provenienti da un dizionario controllato. Solitamente questo lavoro era fatto a mano da indicizzatori umani.

- Organizzazione e Archiviazione di documenti

Gli annunci pubblicitari in un giornale di annunci devono essere classificati in delle categorie "vero affare" "macchine usate" "incontri" ecc…

In un giornale serve l'archiviazione degli articoli sotto l'appropriata sezione

APPLICAZIONI

- Filtering di Testi

E' l'attivita' di selezionare una collezione dinamica (uno stream) di testi. Ad esempio una agenzia di stampa invia notizie agli utenti. Il sistema di filtering fa arrivare all'utente soltanto le notizie che gli interessano. E' un'applicazione che risale agli anni '60, ma l'esplosione della disponibilita' di informazione digitale ne ha ingigantito l'importanza. Ad oggi e' usato in moltissimi contesti: la creazione di giornali Web personalizzati, filtraggio di e-mail spazzatura ecc…

Con Filtering adattativo si intende un filtering capace di adattarsi all'esigenze dell'utente che di volta in volta invia una valutazione del filtraggio

APPLICAZIONI

- Word Sense Disambiguation

L'attivita' di trovare, dato un'occorrenza in un testo di una parola ambigua, il senso che tale occorrenza ha.

Se vediamo il contesto della parola come un documento e il significato della parola come la categoria, disambiguare una parola puo' essere visto come classificare un testo (chiaramente Single-Label)

APPLICAZIONI

- Categorizzazione gerarchica di pagine Web

Catalogare le pagine Web sotto categorie organizzate in gerarchie, permette all'utente un'altra via di accesso all'informazione oltre alle query al Web Search Engine.

La classificazione manuale sarebbe infattibile.

Rispetto alle altre applicazioni si deve tenere conto di due peculiarita':

1. La natura ipertestuale delle pagine Web
2. La struttura gerarchica delle categorie

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

Approccio Knowledge Engineering

- Molto popolare negli anni '80
- La creazione di un classificatore di testi automatico consiste nella creazione di un **sistema esperto** capace di prendere decisioni di classificazione.
- Un tale sistema esperto e' un insieme di regole (definite manualmente) del tipo

  if<DNF Formula> ***then*** $<c_i>$ ***else*** $<\tau c_i>$

(DNF=Forma Normale Disgiuntiva)

Approccio Knowledge Engineering

- Le regole venivano definite da un ingegnere della conoscenza con l'aiuto di un esperto del dominio
- L'esempio piu' famoso e' il sistema **CONSTRUE** progettato dal Carnegie Group per l'agenzia di stampa *Reuters*
- SVANTAGGI:
  1. Se si deve modificare l'insieme di categorie e' di nuovo necessario l'aiuto dell'esperto di dominio
  2. Se si vuole cambiare dominio del classificatore si deve chiamare un nuovo esperto di dominio e riniziare il lavoro da capo

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

Approccio Machine Learning

- Si sviluppa a partire dai primi anni '90
- Un processo induttivo costruisce **automaticamente** un classificatore per una categoria c_i osservando le caratteristiche di un insieme di documenti che sono stati precedentemente classificati sotto c_i o $\neg c_i$ da un esperto del dominio. [Dalle caratteristiche osservate il processo induttivo decide quale caratteristiche deve avere un nuovo documento per essere classificato sotto c_i]

Approccio Machine Learning

- Non si costruisce un classificatore, ma un costruttore di classificatori che va bene per ogni dominio
- RISORSA CHIAVE: Documenti classificati manualmente (spesso sono gia' disponibili ma anche se non sono disponibili…)
- E' piu' facile classificare documenti manualmente piuttosto che stabilire delle regole per la classificazione dei documenti perche' e' spesso piu' facile caratterizzare un concetto estensionalmente piuttosto che intensionalmente

Approccio Machine Learning

- Il corpo iniziale dei documenti gia' classificati viene diviso in tre insiemi:

1. Training Set: Insieme dei documenti che vengono usati per costruire il classificatore
2. Validation Set: Una volta costruito il classificatore potrebbe essere necessario aggiustare dei parametri. Per valutare il giusto valore da assegnare ai parametri si fanno test su questo insieme
3. Test Set: Usato per testare l'efficacia del classificatore

- I tre insiemi devono essere assolutamente disgiunti

Approccio Machine Learning

La costruzione di un classificatore si articola in tre fasi:

1. Indicizzazione dei documenti e riduzione dimensionale
2. Induzione del classificatore
3. Valutazione dell'efficacia del classificatore

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

INDICIZZAZIONE DEI DOCUMENTI

- I documenti non possono essere interpretati direttamente da un classificatore
- Per questo si applica una procedura di indicizzazione che mappa un documento in una rappresentazione compatta del suo contenuto
- Un documento d_j viene rappresentato come un vettore di pesi

$$d_j=<w_{1j},\ldots,w_{|T|j}>$$

dove:

T e' l'insieme dei termini

$0<w_{kj}<1$ rappresenta quanto il termine t_k contribuisce alla semantica di d_j

COS'E' UN TERMINE?

- Bag of word: Un termine e' una parola. *T* e' l'insieme di tutte le parole che occorrono in tutti i documenti del Training Set (*Tr*)
- Molti autori hanno provato a usare frasi (piuttosto che parole) come termini, ma i risultati non sono stati soddisfacenti. Nuovi approcci all'indicizzazione con frase sembrano essere efficaci. L'ultima parola deve ancora essere detta.

COME SI CALCOLA IL PESO DI UN TERMINE?

- Pesi Binari: w_{kj}=1 se il termine t_k e' presente in d_j ,0 altrimenti
- Pesi non Binari: Solitamente e' usata la funzione *tf idf* .

$$tf\ idf\ (t_k, d_j) = \#(t_k, d_j)\ \log[|Tr| / \#_{Tr}(t_k)]$$

dove:

$\#(t_k, d_j)$ indica il numero di occorenze di t_k in d_j

$\#_{Tr}(t_k)$ indica il numero di documenti di *Tr* nel quale t_k occorre

IDEA:

1. + un termine appare in un documento, + e' rappresentativo del suo significato
2. + un termine e' comune, - e' discriminante

La funzione tf idf

- SVANTAGGI:
  - La funzione pesa l'importanza di un termine in un documento considerando soltanto le occorrenze del termine. Non viene data nessuna importanza all'ordine in cui i termini occorrono nel documento. La semantica di un documento e' ridotta alla semantica lessicale dei termini che vi occorrono .
  - La funzione non puo' essere utilizzata in quelle applicazioni dove non e' disponibile dall'inizio l'intero *Tr* (Filtering Adattativo)
- Pertanto esistono molti altri tipi di funzioni

Prima di indicizzare…

- Rimozione delle function word (articoli, preposizioni, congiunzioni ecc…): viene quasi sempre effettuata
- Stemming (raggruppare le parole per la loro radice morfologica): ci sono un po' di controversie. Per adesso la tendenza e' quella di adottarlo in quanto riduce:
  1. Lo spazio dei termini
  2. Il livello di dipendenza stocastica tra i termini

RIDUZIONE DIMENSIONALE

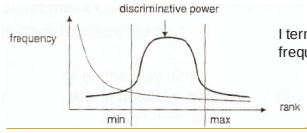
- In TC l'alta dimensione dello spazio dei termini puo' essere problematica: gli algoritmi usati per l'induzione dei classificatori non sono efficienti per alti valori di $|T|$.
- Per questo prima di indurre un classificatore si effettua la riduzione dimensionale: si riduce la dimensione dello spazio vettoriale da $|T|$ a $|T'|$ (insieme dei termini ridotto)

RIDUZIONE DIMENSIONALE

- Rimuovendo dei termini si rischia di rimuovere informazione potenzialmente utile. Pertanto il processo di riduzione deve essere effettuato con cura.
- Esistono due tipi di tecniche usate per la riduzione dimensionale:
  1. Per selezione di termini (T' e' un sottoinsieme di T)
  2. Per estrazione dei termini (i termini in T' non sono dello stesso tipo dei termini in T, ma sono ottenuti dalla combinazione o trasformazione dei termini originali)

RIDUZIONE DIMENSIONALE per selezione di termini

- Fissata un x, si deve determinare il sottoinsieme T' (tale che $|T'|=x$) tale che, quando usato per indicizzare i documenti, produca la piu' alta efficacia.
- Si devono pertanto individuare i termini piu' interessanti per classificare.



I termini piu' interessanti hanno frequenza medio-bassa

RIDUZIONE DIMENSIONALE per selezione di termini

- Metodo di selezione dei termini wrapper:
  Si costruisce un classificatore per ogni T' possibile e si seleziona il T' che ha generato il classificatore piu' efficace. Metodo forza bruta: Costosissimo
- Metodo di selezione dei termini filtering:
  Si prendono i $|T'|$ termini che hanno ricevuto il piu' alto punteggio da una funzione che misura l'importanza di un termine per la classificazione. Sono state definite e studiate molte funzioni per questo proposito.
  La maggior parte di queste sfrutta l'intuizione secondo la quale i termini piu' utili per la classificazione sotto c_i sono quelli che sono distribuiti piu' differentemente negli esempi positivi e negativi di c_i

Funzione document frequency per la selezione di termini

- document frequency $\#_{Tr}(t_k)$ indica il numero di documenti di *Tr* nel quale t_k occorre (si tengono soltanto i termini che occorrono nel piu' alto numero di documenti)
- E' stato dimostrato che si puo' ridurre lo spazio dimensionale di un fattore 10 senza perdita di efficacia
- Facendo cosi' si eliminano i termini con frequenza molto molto bassa e non intacchiamo quelli con frequenza medio-bassa

RIDUZIONE DIMENSIONALE per estrazione di termini

- Fissato un x, si cerca di sintetizzare dall'insieme dei termini *T*, un insieme T' (tale che $|T'|=x$) di nuovi termini che massimizzi l'efficacia
- Uno dei metodi di estrazione dei termini usato per la classificazione di testi e' il Term Clustering: si raggruppano le parole "simili" in cluster, in modo che i cluster possano essere usati (piuttosto che i termini) come dimensioni dello spazio vettoriale.

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

Costruzione induttiva di CLASSIFICATORI DI TESTO

- Il problema della costruzione induttiva di un classificatore di testi e' stato affrontato in svariati modi. Mostreremo i metodi piu' popolari.
- La costruzione induttiva di un classificatore per una categoria ci consiste:
  1. definire una funzione CSV_i: $\boldsymbol{D} \rightarrow [0,1]$
  2. determinare una soglia τ_i

Costruzione induttiva di CLASSIFICATORI DI TESTO

- La funzione CSV_i prende un documento d_j e restituisce un numero che rappresenta quanto d_j dovrebbe essere classificato sotto c_i
- La soglia puo' essere determinata analiticamente o empiricamente (attraverso esperimenti sul validation set)
- Se la classificazione e' Hard allora il nostro classificatore e' $\Phi_i = CSV_i > \tau_i$
- Se la classificazione e' Ranking allora il nostro classificatore e' $\Phi_i = CSV_i$

Costruzione induttiva di CLASSIFICATORI DI TESTO

- CLASSIFICATORI PROBABILISTICI (Naive Bayesiani)
- CLASSIFICATORI SIMBOLICI
  1. Con ALBERI DI DECISIONE
  2. Con REGOLE DI DECISIONE
- METODI CON REGRESSIONE(LLSF)
- METODI LINEARI
  1. METODI ON-LINE(Perceptron)
  2. METODI BATCH (Rocchio)
- CLASSIFICATORI BASATI SUGLI ESEMPI(k-NN)
- SUPPORT VECTOR MACHINE
- COMITATI DI CLASSIFICATORI

CLASSIFICATORI PROBABILISTICI

- Classificatori probabilistici vedono $CSV_i(d_j)$ in termini di $P(c_i|d_j)$, cioe' la probabilita' che un documento, rappresentato da un vettore d_j, appartenga alla categoria c_i e cercano di calcolare tale probabilita' utilizzando il teorema di Bayes:

$$P(c_i|d_j) = P(c_i)P(d_j|c_i)/P(d_j)$$

Dove lo spazio degli eventi e' lo spazio dei documenti:

$P(d_j)$ = probabilita' che un documento preso a caso sia uguale d_j

$P(c_i)$ = probabilita' che un documento preso a caso appartenga a c_i

CLASSIFICATORI PROBABILISTICI

- La stima di $P(d_j|c_i)$ puo' essere problematica dal momento che il numero dei possibili vettori d_j e' troppo alto. Per sorvolare questo problema si assume che tutte le coordinate del vettore del documento siano statisticamente indipendenti

$$P(d_j|c_i) = \prod_{k=1..|T|} P(w_{kj}|c_i)$$

- I Classificatori probabilistici che fanno questa assunzione sono chiamati Naive Bayesiani (Naive perche' l'assunzione non e' mai verificata in pratica)
- Vediamo uno dei piu' famosi: *Binary Independence*

CLASSIFICATORI PROBABILISTICI: Binary Independence

- Si usa solo quando sono stati usati vettori binari per la rappresentazione dei documenti
- Sfruttando il fatto che i documenti sono rappresentati come vettori binari si dimostra che

$$\log[P(c_i|d_j)/(1- P(c_i|d_j))] = \log[P(c_i)/1-P(c_i)] + \Sigma_{k=1\ldots |T|} w_{kj} \log[p_{ki} (1-p_{k\neg i}) / p_{k\neg i} (1-p_{ki})] + \Sigma_{k=1\ldots |T|} \log[1-p_{ki}/1-p_{k\neg i}]$$

Dove:

$p_{ki} = P(w_{kx}=1|c_i)$

$p_{k\neg i} = P(w_{kx}=1|\neg c_i)$

- I termini in verde sono costanti (non dipendono dal documento)
- $P(c_i|d_j)/(1- P(c_i|d_j))$ e' una funzione crescente in $P(c_i|d_j)$ e potrebbe essere usata direttamente come $CSV_i(d_j)$
- Costruire un classificatore significa calcolare (nel modo ovvio) i termini $p_{1i}, p_{1\neg i} \ldots p_{ki}, p_{k\neg i}$

CLASSIFICATORI PROBABILISTICI: Binary Independence

- Il tempo per classificare un documento e' lineare con il numero di termini
- La ricerca si sta' muovendo in queste direzioni per migliorare gli algoritmi Naive-Bayesiani:
  1. Rilassare il vincolo che i vettori di documenti siano binari
  2. Introdurre una forma di normalizzazione dei documenti (documenti + lunghi hanno + probabilita' di essere classificati sotto c_i)
  3. Rilassare l'assunzione di indipendenza dei termini

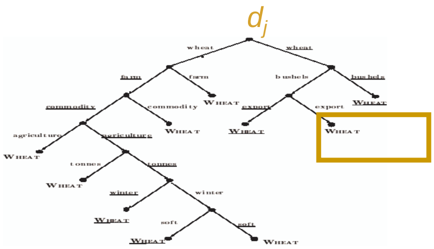
CLASSIFICATORI SIMBOLICI

- I classificatori probabilistici sono di natura essenzialmente numerica, e quindi spesso poco interpretabili direttamente da umani.
- I classificatori simbolici non soffrono di questo problema. Tra questi troviamo:
  1. Alberi di decisione
  2. Regole di Decisione

ALBERI DI DECISIONE

- Un Albero di Decisone classificatore di testi e' un albero tale che:
  1. I nodi interni sono etichettati con termini
  2. I rami che partono dal nodo interno x hanno un test sul peso del termine che etichetta x
  3. Le foglie sono etichettate da categorie
- Un tale classificatore categoriza un documento d_j testando (secondo quanto indicato sui rami) ricorsivamente i pesi che i termini hanno nel vettore d_j, finche' un nodo foglia non e' raggiunto

ALBERI DI DECISIONE



L'albero mostrato in figura classifica un documento sotto la categoria WHEAT (solitamente un albero di decisione puo' classificare tra piu' categorie). Prende in pasto vettori di documenti binari: sui rami testa se un termine vale 1 oppure 0

Mostriamo come quest'albero classifica il documento d_j: **HARD** WHEAT **SOFT** WINTER TONNES COMMODITY **EXPORT** FARM **BUSHEL AGRICOLTURE** (in **giallo** i termini il cui peso e' = 1)

Algoritmi per l'induzione di ALBERI DI DECISIONE

- Strategia "*divide and conquer*"
- Ricorsivamente:
  1. Si testa se tutti i documenti del training set hanno la stessa etichetta
  2. Se no, si seleziona un termine t_k, si partiziona *Tr* in classi di documenti che hanno lo stesso valore di t_k, costruendo cosi' dei nuovi nodi.
- Il processo e' ripetuto finche' ogni foglia cosi' generata contiene documenti del training set tutti di una stessa categoria c_i

Algoritmi per l'induzione di ALBERI DI DECISIONE

- La scelta del termine t_k su cui partizionare e' il passo chiave che distingue i vari algoritmi. Esistono svariate politiche per effettuare questa scelta
- SVANTAGGI: Un albero "troppo allenato" puo' diventare troppo specifico (overfitting). La maggior parte dei metodi per l'induzione di Alberi di Decisione includono un metodo per far crescere l'albero e un metodo per potarlo (per eliminare i rami troppo specifici)

REGOLE DI DECISIONE

- IDEA:Un classificatore per una categoria c_i e' una regola in forma normale disgiuntiva (DNF). Es:
  IF[(¬wheat)∧(bushels ∨ ¬farm) ∧(export)] THEN c_i
- Le Regole di Classificazione sono solitamente classificatori piu' compatti degli Alberi di Classificazione
- Funzionano soltanto con documenti rappresentati come vettori di termini binari

Metodi per l'induzione di REGOLE DI DECISIONE

- I Metodi di induzione delle regole cercono di scoprire tra tutte le possibile regole covering (che classificano correttamente tutti gli esempi del training set) la migliore rispetto a qualche criterio di minimalita'.
- L'induzione di regole avviene in modo BOTTOM-UP: Ogni documento del traing set e' visto come una regola dove le clausole sono i termini del documento e la testa e' la categoria in cui il documento e' inserito. Pertanto i documenti del training set formano un insieme di regole. Queste regole vengono poi generalizzate attraverso una serie di modificazioni.

METODI CON REGRESSIONE

- Mostriamo il *Linear Least Squares Fit* (LLSF) un metodo con regressione proposto da Yang e Chute nel '94.
- Per ogni documento d_j, esiste:
  - Input vector $I(d_j)$: il vettore standard con i termini pesati
  - Output vector $O(d_j)$: un vettore di pesi alle varie categorie (e' binario per i documenti di allenamento)
- Dato un documento d_j e il suo $I(d_j)$ la classificazione consiste nel determinare $O(d_j)$. Costruire un classificatore significa costruire una matrice M^I tale che $M^I I(d_j) = O(d_j)$.

METODI CON REGRESSIONE

- LLSF calcola la matrice dal training set utilizzando questa formula

$$M^I = \text{argmin}_M \|MI - O\|_F$$

dove:

  - $\text{argmin}_M(x)$ e' la M per cui e' minimo x
  - $\|V\|_F$ e' la norma di Frobenio (la somma di tutti gli elementi) della matrice V
  - I e' la matrice le cui colonne sono gli input vector dei training document
  - O e' la matrice le cui colonne sono gli output vector dei training document
- VANTAGGI: risultati sperimentali dimostrano che e' uno dei classificatori piu' efficaci ad oggi
- SVANTAGGI: calcolare la matrice M^I e' costosissimo

CLASSIFICATORI LINEARI

- Un classificatore lineare per una categoria c_i e' un rappresentazione di c_i in termini di un vettore $\underline{c_i} = \langle w_{1i}, \ldots, w_{|T|i} \rangle$ nello spazio |T|-dimensionale (lo stesso dei documenti).
- $CSV_i(d_j)$ e' il dot product tra $\underline{d_j}$ e $\underline{c_i}$
- I metodi per la costruzione induttiva di classificatori lineari sono divisi in due categorie:
  1. On-line Method (Perceptron)
  2. Batch Method (Rocchio)

Metodi On-line per l'induzione di CLASSIFICATORI LINEARI

- IDEA: Si definisce il classificatore dopo aver analizzato il primo training document e con i successivi documenti si raffina il classificatore
- Molto utile:
  1. se inizialmente non e' disponibile interamente il training set
  2. in quelle applicazioni in cui l'utente del classificatore provvede un feedback su come i documenti sono stati classificati (filtering adattativo).

Algoritmo Perceptron per l'induzione di CLASSIFICATORI LINEARI

Algoritmo Perceptron

- Si inizializzano tutti i pesi w_{ki} del classificatore c_i ad uno stesso valore positivo
- Quando esamina un training document d_j, il classificatore cerca di classificarlo e poi esamina il risultato della classificazione
  - Se il risultato e' corretto allora non fa niente
  - Se il risultato e' scorretto modifica i pesi di c_i:
    - Se d_j era un esempio positivo di c_i allora i pesi dei termini attivi (w_{kj}=1) vengono aumentati di α
    - Se d_j era un esempio negativo di c_i allora i pesi dei termini attivi (w_{kj}=1) vengono diminuiti di α

Algoritmo Perceptron per l'induzione di CLASSIFICATORI LINEARI

- Quando il classificatore si e' dimostrato abbastanza efficace, il fatto che un peso w_{ki} e' molto basso significa che il termine t_k ha contribuito negativamente al processo di classificazione, e che quindi puo' essere eliminato dalla rappresentazione. Si puo' dire che l'algoritmo Perceptron permette una "riduzione dello spazio dei termini al volo"
- I Classificatori costruiti con Perceptron hanno dimostrato una buona efficacia

Metodo di Rocchio per l'induzione di CLASSIFICATORI LINEARI

- Il metodo di Rocchio e' un metodo Batch, che tutto in una volta induce un classificatore

$$C_i = \langle w_{1i},\ldots,w_{|T|i}\rangle$$

- I pesi vengono calcolati con la seguente formula:

$$w_{ki} = \alpha \left(\Sigma_{\{dj \in POS_i\}} w_{kj}/|POS_i| \right) - \beta \left(\Sigma_{\{dj \in NEG_i\}} w_{kj}/|NEG_i| \right)$$

- α e β sono due parametri controllati che permettono di valutare l'importanza degli esempi negativi e degli esempi positivi
- $POS_i = \{d_j \in Tr \mid \Phi^I(d_j,c_i)=T\}$
- $NEG_i = \{d_j \in Tr \mid \Phi^I(d_j,c_i)=F\}$

Metodo di Rocchio per l'induzione di CLASSIFICATORI LINEARI

- In generale il classificatore Rocchio guarda la vicinanza del documento da classificare al centroide degli esempi positivi e la lontananza dal centroide degli esempi negativi.
- SVANTAGGI: Il classificatore Rocchio come tutti i Classificatori Lineari ha lo svantaggio che divide lo spazio dei documenti linearmente. Questo comporta gravi perdite di efficacia: la media e' solo parzialmente rappresentativa dell'intero insieme [ill]

CLASSIFICATORI BASATI SU ESEMPI

- IDEA: Non si costruisce una rappresentazione della categoria, ma si confida sui documenti del training set che sono piu' vicini al documento che vogliamo classificare.
- Il primo metodo Example-Based introdotto per la Classificazione di Testi e' del 92 [Creecy et al. 1992; Masand et al. 1992]
- Noi mostreremo un algoritmo implementato da Yang [1994]: *k*-NN

CLASSIFICATORI BASATI SU ESEMPI: *k*-NN (k nearest neighbours)

- IDEA: Per decidere se classificare d_j sotto c_i *k*-NN guarda se i *k* documenti del training set piu' simili a d_j sono stati classificati sotto c_i. Se una parte grande abbastanza e' stata classificata sotto c_i allora il documento viene classificato in c_i altrimenti no .
- La similarita' tra documenti e' calcolata con una qualche funzione RSV(d_i,d_z). Nell'implementazione di Yang si cerca di dare piu' peso ai documenti piu' vicini.

$$CSV_i(d_j) = \sum_{d_z \in Trk(d_j)} RSV(d_i,d_z) \cdot [\Phi^i(d_z,c_i)]$$

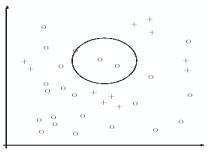
$Trk(d_j)$ = i k documenti del training set piu' simili a d_j secondo la funzione *RSV*

[x] = 1 se x e' TRUE, 0 se x e' FALSE

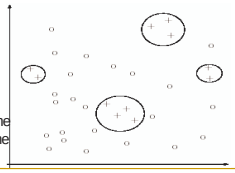
CLASSIFICATORI BASATI SU ESEMPI: *k*-NN (k nearest neighbours)

- La soglia *k* viene determinata empiricamente attraverso test sul validation set. E' stato dimostrato che aumentando di molto *k* non diminuiscono significativamente le performance
- VANTAGGI: *k*-NN, diversamente dai classificatori lineari non suddivide lo spazio dei documenti linearmente. Quindi risulta essere piu' "locale" [ill]
- SVANTAGGI: Inefficienza a tempo di classificazione: *k*-NN deve calcolare la similarita' di tutti i documenti del training set con il documento da classificare
- E' conveniente utilizzarlo per document-pivoted categorization: calcolare la somiglianza dei training document puo' essere fatto una volta per tutte le categorie.

CLASSIFICATORI LINEARI vs BASATI SU ESEMPI



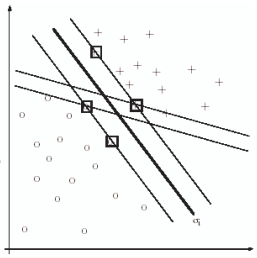
Sulla sinistra e' mostrato il comportamento di un classificatore costruito con Rocchio
In basso un classificatore costruito con *k*-nn



Le croci e i cerchi rappresentano gli esempi positivi e negativi
I cerchi sono l'area di influenza dei classificatori
Per facilitare la comprensione, la similarita' tra documenti e' vista come distanza Euclidea piuttosto che come dot product

SUPPORT VECTOR MACHINE

- IDEA: Tra tutte le superfici σ_1, σ_2 ... nello spazio |T|-dimensionale che separano gli esempi positivi da quelli negativi, si cerca la superficie tale che la propieta' di separazione e' invariante rispetto alla piu' ampia traslazione possibile



SUPPORT VECTOR MACHINE

- VANTAGGI:
  1. Non serve la selezione di termini: e' robusta all'overfitting e scalabile per considerevoli dimensioni
  2. Non serve testare e settare i parametri su un validation set. Esistono valori default di parametri definiti da risultati teorici

COMITATI DI CLASSIFICATORI

- IDEA: un task che richiede conoscenza esperta puo' essere eseguito meglio da k esperti ben coordinati, che da un unico esperto
- Si devono scegliere:
  1. k distinti classificatori
  2. Una funzione di combinazione dei classificatori

COMITATI DI CLASSIFICATORI

1. I k classificatori che formano il comitato dovrebbero essere il piu' indipendenti possibile:
   - In termini di approccio di indicizzazione
   - In termini di metodi induttivi
2. Diverse funzioni di combinazione sono state ideate:
   1. Majority Voting
   2. Weighted Linear Combination
   3. Dinamic Classifier Selection
   4. Adaptive Classifier Combination

COMITATI DI CLASSIFICATORI

- Majority Voting
  - La maggioranza vince
- Weighted Linear Combination
  - Si attribuisce un peso ad ogni classificatore secondo la sua efficacia. Il risultato e' la combinazione lineare dei risultati dei classificatori e dei loro pesi
- Dinamic Classifier Selection
  - Per classificare un documento d_j si sceglie la risposta di quello che si e' dimostrato piu' efficace con gli / esempi piu' vicini a d_j di un validation set
- Adaptive Classifier Combination
  - Si fa la combinazione lineare pesata. I pesi sono dati dinamicamente in base all'efficacia di un classificatore sugli / documenti (di un validation set) piu' vicini al documento da classificare d_j

COMITATI DI CLASSIFICATORI

- Lo svataggio principale di un comitato di classificatori e' l'inefficienza

  Costo Computazionale Totale =
  somma costi dei singoli classificatori
  +
  il costo della funzione di combinazione
- Risultati sperimentali contrastanti riguardo all'efficacia dei Comitati. Sono stati fatti ancora pochi esperimenti e non e' possibile trarre conclusioni

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

VALUTARE UN CLASSIFICATORE DI TESTI

- La valutazione dei classificatori di testi e' effettuata sperimentalmente piuttosto che analiticamente.
- La Classificazione di Testi non puo' essere formalizzata (a causa della sua natura soggettiva) e quindi non puo' essere valutata analiticamente
- La valutazione sperimentale di un classificatore solitamente misura la sua efficacia: l'abilita' di prendere la giusta decisione di classificazione

MISURE DELL'EFFICACIA DI CLASSIFICAZIONE DI TESTI

- π_i [*Precision wrt* c_i] = $P(\Phi^I(d_x,c_i)=T|\Phi(d_x,c_i)=T)$

indica il grado di Correttezza del classificatore rispetto alla categoria c_i

- ρ_i [*Recall wrt* c_i] = $P(\Phi(d_x,c_i)=T|\Phi^I(d_x,c_i)=T)$

indica il grado di Completezza del classificatore rispetto alla categoria c_i

CALCOLO DI $\pi_i = P(\Phi^I(d_x,c_i)=T \mid \Phi(d_x,c_i)=T)$ E $\rho_i = P(\Phi(d_x,c_i)=T \mid \Phi^I(d_x,c_i)=T)$

Le probabilita' vengono stimate sui risultati del classificatore su un test set

$\pi_i = TP_i/(TP_i+FP_i)$ $\rho_i = TP_i/(TP_i+FN_i)$

TP_i = Veri Positivi= #documenti classificati correttamente sotto c_i

FP_i = Falsi Positivi= #documenti classificati incorrettamente sotto c_i

VN_i = Veri Negativi= #documenti non classificati correttamente sotto c_i

FN_i = Falsi Negativi= #documenti non classificati incorrettamente sotto c_i

Category c_i		expert judgments	
		YES	NO
classifier judgments	YES	TP_i	FP_i
	NO	FN_i	TN_i

MISURE DELL'EFFICACIA DI CLASSIFICAZIONE DI TESTI

- π_i e ρ_i sono misure di efficacia relative alla categoria c_i. Vogliamo definire l'efficacia di un Classificatore Globalmente.

  π global precision e ρ global recall

- Si possono calcolare in due metodi distinti:
  - Microaveraging
    - $\pi^\mu = \sum_{i=1...|C|} TP_i / \sum_{i=1...|C|}(TP_i + FP_i)$
    - $\rho^\mu = \sum_{i=1...|C|} TP_i / \sum_{i=1...|C|}(TP_i + FN_i)$
  - Macroaveraging
    - $\pi^M = \sum_{i=1...|C|} \pi_i / |C|$
    - $\rho^M = \sum_{i=1...|C|} \rho_i / |C|$
- I Due metodi danno risultati diversi: non e' ancora chiaro quale sia il migliore da utilizzare.

COMBINARE MISURE DI EFFICACIA

- Le misure π e ρ prese singolarmente non bastano per esprimere l'efficacia:
  - Il classificatore che classifica tutti i documenti sotto c_i ha ρ =1 (non ci sono falsi negativi)
  - Il classificatore che classifica tutti i documenti sotto $\neg c_i$ ha π =1 (non ci sono falsi positivi)
- π e ρ sono inversamente proporzionali, 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

MISURE ALTERNATIVE

- Sono state ideate un enorme quantita' di misure per l'efficacia: accuratezza e errore, misure di successo relativo ecc…
- Sono state proposte misure diverse dall'efficacia per valutare un classificatore: efficienza, utilita' (che tiene conto dei concetti di guadagno e perdita) ecc…

QUALE CLASSIFICATORE E' IL MIGLIORE?

- La risposta e' relativa all'applicazione in cui il classificatore deve essere usato
- Esistono collezioni di documenti standard su cui effettuare il BENCHMARK di un classificatore
- Ma per selezionare il classificatore da usare non possiamo utilizzare i risultati dei benchmark ottenuti in letteratura

Piano del discorso

- Definizione del problema
- Applicazioni
- Approccio Knowledge Engeneering
- Approccio Machine Learning
  1. Indicizzazione e riduzione dimensionale
  2. Induzione di classificatori
  3. Valutazione dell'efficacia
- Conclusioni

CONCLUSIONI

- Dai primi anni 90 ad oggi l'efficacia dei classificatori di testo e' aumentata notevolmente grazie all'impiego di metodi di Machine Learning
- La classificazione automatica di testi e' divenuta un'area molto interessante. A causa di molte ragioni:
  1. I suoi domini di applicazioni sono numerosi e importanti e dato l'aumento di documenti di testo in digitale sono destinati ad aumentare considerevolmente
  2. E' indispensabile in molte applicazioni in cui l'elevato numero di documenti e il breve tempo di risposta richiesto dall'applicazioni, rendono l'alternativa manuale impossibile
  3. Puo' aumentare la produttivita' di un classificatore umano
  4. Si sono raggiunti livelli di efficacia paragonabili a quelli di un classificatore umano

Bibliografia

[1] Russel, Norvig: Artificial intelligence modern approach

[2] J. M. G. Hidalgo: Text Mining and Internet Content Filtering

[3] C.D. Manning, H. Schutze: Foundation of Statistical Natural Language Processing

[4] F. Sebastiani: Machine Learning in Automated Text Categorization