Classificatori bayesiani

Corso di Data Mining
a.a. 2024-2025

Giorgio Gambosi

Classificatori probabilistici

- In un classificatore probabilistico le classi sono modellate da opportune distribuzioni condizionate $p(\mathbf{x}|C_k)$: è possibile campionare da tali distribuzioni per generare elementi casuali statisticamente equivalenti agli elementi della collezione utilizzata per derivare il modello.
- La regola di Bayes permette di derivare $p(C_k|\mathbf{x})$ dati tali modelli (e le distribuzioni a priori $p(C_k)$ delle classi)
- Possiamo derivare i parametri di $p(\mathbf{x}|C_k)$ e $p(C_k)$ dal dataset, ad esempio attraverso la stima di massima verosimiglianza
- La classificazione viene effettuata confrontando $p(C_k|\mathbf{x})$ per tutte le classi

Esempio nel caso discreto: modelli di linguaggio

Un **modello di linguaggio** è una distribuzione di probabilità (categorica) su un vocabolario di termini (possibilmente, tutte le parole che compaiono in una grande collezione di documenti).

Un modello di linguaggio può essere applicato per prevedere il termine successivo in un testo. La probabilità di occorrenza di un termine è legata al suo contenuto informativo ed è alla base di numerose tecniche di recupero dell'informazione.

Si assume che la probabilità di occorrenza di un termine sia indipendente dai termini precedenti in un testo (modello a **bag of words**).

Dato un modello di linguaggio, è possibile campionare dalla distribuzione per generare documenti casuali statisticamente equivalenti ai documenti della collezione utilizzata per derivare il modello.

- Sia $\mathcal{D} = \{t_1, \ldots, t_n\}$ il dizionario, ovvero l'insieme dei termini che compaiono in un dato dataset, che in questo caso è una collezione $\mathcal{C}$ di documenti
- Consideriamo una rappresentazione di un documento come un vettore di dimensione n, che associa ad ogni termine t_i del dizionario un valore numerico calcolato sul documento, ad esempio il numero di occorrenze del termine nel documento stesso
- Normalmente, i termini considerati sono quelli risultanti dopo la rimozione delle **stop word** (termini comuni e non informativi) e l'applicazione dello **stemming** (riduzione delle parole alla loro forma base).
- Per ogni $i = 1, \ldots, n$ sia inoltre m_i la molteplicità (numero di occorrenze) del termine t_i in $\mathcal{C}$.
- Un modello di linguaggio può essere derivato come una distribuzione categorica associata a un vettore $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)^T$ di probabilità, tale cioè che

$$0 \leq \phi_i \leq 1 \quad i = 1, \ldots, n \qquad\qquad \sum_{i=1}^{n} \phi_i = 1$$

dove $\phi_j = p(t_j|\mathcal{C})$ è la probabilità (stimata) di occorrenza di t_i, condizionata dalla collezione $\mathcal{C}$ da cui l'abbiamo calcolata.

Si noti che il modello di linguaggio considerato qui è molto semplice, in quanto considera la probabilità di occorrenza di t_i indipendente dal carattere precedente: in altri termini, $p(t_i|t_j, \mathcal{C}) = p(t_i|t_k, \mathcal{C})$ per ogni i, j, k. Un modello di linguaggio più ricco (e complesso) è quello in cui sono definite e stimate a partire da $\mathcal{C}$ le probabilità condizionate $p(t_i|t_j, \mathcal{C})$. In questo caso, la rappresentazione di un documento evidentemente varia, assegnando un valore numerico per ogni coppia ordinata i, j di indici, e quindi risulta data da una matrice $n \times n$ di valori.

Si noti che il tutto può essere esteso a considerare le probabilità di occorrenza di t_i condizionata alla sequenza di L caratteri che lo precedono $p(t_i|\langle t_{j_1}, t_{j_2}, \ldots, t_{j_L}\rangle, \mathcal{C})$, e quindi una rappresentazione dei documenti come matrici $L+1$ dimensionali con n valori per gli indici su ogni dimensione.

Apprendimento di un modello di linguaggio tramite ML

Applicando il criterio di massima verosimiglianza per derivare le probabilità dei termini nel modello di linguaggio si ottiene:

$$\phi_j = p(t_j|\mathcal{C}) = \frac{m_j}{\sum_{k=1}^{n} m_k} = \frac{m_j}{N}$$

dove $N = \sum_{i=1}^{n} m_i$ è il numero totale di occorrenze di tutti i termini in $\mathcal{C}$ (dopo rimozione delle stopword e stemming).

Si noti che secondo questa stima, un termine t_i che non è mai comparso in $\mathcal{C}$ ha probabilità zero di essere osservato (paradosso del cigno nero). Questo a causa dell'overfitting del modello ai dati osservati, tipico della stima ML.

Soluzione: assegnare una probabilità piccola, ma non nulla, agli eventi (termini) non ancora osservati. Questo è chiamato **smoothing**. Ad esempio, considerare

$$\phi_j = \frac{m_j + \alpha}{\sum_{k=1}^{n}(m_k + \alpha)} = \frac{m_j + \alpha}{N + n\alpha}$$

dove $\alpha > 0$ è una costante piccola predefinita. Un modo diverso di effettuare lo smoothing è basato su una stima bayesiana (MAP) di $\boldsymbol{\phi}$, che non approfondiamo qui.

Classificatori di documenti

Il modello di linguaggio può essere applicato per costruire classificatori di documenti (binari o multiclasse) nel modo seguente:

- date due classi C_1, C_2, si assume che, per qualsiasi documento d, le probabilità $p(C_1|d)$ e $p(C_2|d)$ siano note: allora, d può essere assegnato (ad esempio) alla classe con probabilità maggiore
- come si può derivare $p(C_k|d)$ per un qualsiasi documento, data una collezione $\mathcal{C}_1$ di documenti noti appartenenti a C_1 e una collezione simile $\mathcal{C}_2$ per C_2? Possiamo applicare la regola di Bayes:

  $$p(C_k|d) \propto p(d|C_k)p(C_k)$$

  l'evidenza $p(d)$ è la stessa per entrambe le classi, e può essere ignorata considerando le collezioni
- rimane ancora il problema di calcolare $p(C_k)$ e $p(d|C_k)$ a partire dalle collezioni $\mathcal{C}_1$ e $\mathcal{C}_2$

Le probabilità a priori $p(C_k)$ $(k = 1, 2)$ possono essere facilmente stimate da $\mathcal{C}_1, \mathcal{C}_2$: ad esempio, applicando il criterio di massima verosimiglianza otteniamo

$$\pi_k = p(C_k) = \frac{|\mathcal{C}_1|}{|\mathcal{C}_1| + |\mathcal{C}_2|}$$

Per quanto riguarda le verosimiglianze $p(d|C_k)$ $(k = 1, 2)$, osserviamo che d può essere visto come la sequenza di $m = |d|$ occorrenze di termini da $\mathcal{C}$*

$$d = \langle t_1, \bar{t}_2, \ldots, \bar{t}_m \rangle$$

Applicando la regola del prodotto, si ottiene

$$p(d|C_k) = p(\langle \bar{t}_1, \ldots, \bar{t}_m \rangle | C_k) = p(\bar{t}_1|C_k)p(\bar{t}_2|\bar{t}_1, C_k) \cdots p(\bar{t}_m|\bar{t}_1, \ldots, \bar{t}_{m-1}, C_k)$$

L'assunzione di naive Bayes

Calcolare $p(d|C_k)$ è molto più semplice se assumiamo che le occorrenze dei termini siano a coppie condizionalmente indipendenti, data la classe C_k, cioè che, per $i, j = 1, \ldots, n_d$ e $k = 1, 2$,

$$p(\bar{t}_i, \bar{t}_j|C_k) = p(\bar{t}_i|C_k)p(\bar{t}_j|C_k)$$

o anche

$$p(\bar{t}_i|\bar{t}_j, C_k) = p(\bar{t}_i|C_k)$$

Come conseguenza di ciò, risulta

$$p(d|C_k) = \prod_{j=1}^{m} p(\bar{t}_j|C_k) = \prod_{r=1}^{n} p(t_r|C_k)^{m_r}$$

dove m_r è la molteplicità (il numero di occorrenze) del termine t_j in d.

Le probabilità $p(t_j|C_k) = \phi_{jk}$ sono disponibili per tutti i termini se sono stati derivati i modelli di linguaggio $\boldsymbol{\phi}_1 = \langle \phi_{11}, \ldots, \phi_{n1} \rangle$ e $\boldsymbol{\phi}_2 = \langle \phi_{12}, \ldots, \phi_{n2} \rangle$ per C_1 e C_2 rispettivamente, dai documenti in $\mathcal{C}_1$ e $\mathcal{C}_2$.

Applicando queste considerazioni, otteniamo un classificatore Naive Bayes che si comporta come segue, per classificare un documento d:

1. siano $\bar{t}_1, \ldots, \bar{t}_m$ le occorrenze di termini in d
2. siano $m_1, \ldots, m_n$ il numero di occorrenze dei termini del dizionario in d
3. per $k = 1, 2$ calcolare, applicando l'assunzione di naive Bayes, $p(d|C_k) = \prod_{r=1}^{n} \phi_{rk}^{m_r}$
4. assegnare d alla classe $\mathcal{C}_s$ tale che $r = \underset{k \in \{1,2\}}{\text{argmax}}\, p(d|C_k)p(C_k) = \underset{k \in \{1,2\}}{\text{argmax}}\, \pi_k \prod_{r=1}^{n} \phi_{rk}^{m_r}$

Si noti che lo stesso approccio può essere applicato alla classificazione di elementi $\mathbf{x} = (x_0, \ldots, x_d)$. In questo caso, l'assunzione Naive Bayes è che le features siano condizionatamente indipendenti data la classe, per cui $p(\mathbf{x}|C_k) = \prod_{i=0}^{d} p(x_i|C_k)$.

Classificatore bayesiano nel caso continuo

In questo caso, le distribuzioni condizionali $p(\mathbf{x}|C_k)$ sono continue, considereremo qui come esempio il caso più semplice in cui assumiamo siano gaussiane. In questo caso le corrispondenti distribuzioni a posteriori $p(C_k|\mathbf{x})$ possono essere facilmente derivate.

*In realtà, visto che assumiamo l'indipendenza di una occorrenza di termine da quelle precedenti, l'ordine delle occorrenze non è rilevante, per cui d può essere visto come un multinsieme $\{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_m\}$

Facciamo inoltre l'ipotesi che tutte le distribuzioni $p(\mathbf{x}|C_k)$ abbiano la stessa matrice di covarianza Σ, di dimensione $d \times d$. Allora,

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)$$

e si può dimostrare che la distribuzione di interesse per la classificazione $p(C_1|\mathbf{x})$ risulta essere

$$p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}+b)$$

dove σ è la funzione sigmoide.

Come si può vedere, il classificatore risulta basato sulla stessa struttura della logistic regression (funzione non lineare sigma applicata a una combinazione lineare dei valori delle features). Mentre però nel caso della logistic regression i valori di $\mathbf{w}, b$ sono calcolati mediante discesa del gradiente rispetto alla funzione di costo cross-entropy, in questo caso essi sono calcolati sulla base dei parametri delle distribuzioni $p(\mathbf{x}|C_1)$ e $p(\mathbf{x}|C_2)$, oltre che delle probabilità a priori $p(C_1), p(C_2)$ nel modo seguente

$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$
$$b = \frac{1}{2}(\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1) + \log\frac{p(C_1)}{p(C_2)}$$

Si noti che

$$p(C_2|\mathbf{x}) = 1 - p(C_1|\mathbf{x}) = 1 - \sigma(\mathbf{w}^T\mathbf{x}+b) = 1 - \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}+b}} = \frac{e^{\mathbf{w}^T\mathbf{x}+b}}{1+e^{\mathbf{w}^T\mathbf{x}+b}} = \frac{1}{1+e^{-(\mathbf{w}^T\mathbf{x}+b)}} = \sigma(-\mathbf{w}^T\mathbf{x}-b)$$

Funzione discriminante

La funzione discriminante, che separa le due partizioni dello spazio delle feature associate alle due classi, può essere ottenuta imponendo la condizione $p(C_1|\mathbf{x}) = p(C_2|\mathbf{x})$, cioè, $\sigma(\mathbf{w}^T\mathbf{x}+b) = \sigma(-\mathbf{w}^T\mathbf{x}-b)$.

Questo è equivalente a $\mathbf{w}^T\mathbf{x}+b = -\mathbf{w}^T\mathbf{x}-b$ e quindi a $-\mathbf{w}^T\mathbf{x}-b = 0$. Di conseguenza, si ottiene che la separazione è lineare

$$\mathbf{w}^T\mathbf{x}+b = 0$$

Classi multiple

In questo caso, indicando con K il numero di classi, si può mostrare che risulta

$$p(C_k|\mathbf{x}) = s(\mathbf{w}_k^T\mathbf{x}+b_k) = \frac{e^{\mathbf{w}_k^T\mathbf{x}+b_k}}{\sum_{i=1}^K e^{\mathbf{w}_i^T\mathbf{x}+b_i}}$$

dove s è la funzione *softmax*.

Il classificatore risulta quindi avere la stessa struttura di un classificatore softmax, con la differenza (come nel caso binario) che i parametri non sono appresi mediante minimizzazione del rischio empirico ma calcolati dalle distribuzioni $p(\mathbf{x}|C_k)$ e $p(C_k)$ nel modo seguente

$$\mathbf{w}_k = \frac{1}{2}\Sigma^{-1}\boldsymbol{\mu}_k$$
$$b_k = \frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \log p(C_k) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma|$$

Ancora una volta, $p(C_k|\mathbf{x}) = s(\mathbf{w}_k^T\mathbf{x}+b_k)$ è calcolato applicando una funzione non lineare a una combinazione lineare delle caratteristiche (**modello lineare generalizzato**)

Massima verosimiglianza

Le distribuzioni condizionate alla classe $p(\mathbf{x}|C_k)$ possono essere derivate dal training set tramite stima di massima verosimiglianza.

È necessario stimare $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma$ e $\pi = p(C_1)$ (chiaramente, $p(C_2) = 1 - \pi$).

Il training set $\mathcal{T}$: contiene n elementi $(\mathbf{x}_i, t_i)$, con

$$t_i = \begin{cases} 0 & \text{se } \mathbf{x}_i \in C_2 \\ 1 & \text{se } \mathbf{x}_i \in C_1 \end{cases}$$

Se $\mathbf{x} \in C_1$, allora $p(\mathbf{x}, C_1) = p(\mathbf{x}|C_1)p(C_1) = \pi \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \Sigma)$
Se $\mathbf{x} \in C_2$, $p(\mathbf{x}, C_2) = p(\mathbf{x}|C_2)p(C_2) = (1 - \pi) \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \Sigma)$

La verosimiglianza del training set $\mathcal{T}$ è quindi

$$p(\mathcal{T}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = \prod_{i=1}^{n} (\pi \cdot \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \Sigma))^{t_i} ((1 - \pi) \cdot \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_2, \Sigma))^{1-t_i}$$

La log-verosimiglianza corrispondente è

$$\begin{aligned} \log p(\mathcal{T}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = &\sum_{i=1}^{n} (t_i \log \pi + t_i \log(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \Sigma))) + \\ &+ \sum_{i=1}^{n} ((1 - t_i) \log(1 - \pi) + (1 - t_i) \log(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_2, \Sigma))) \end{aligned}$$

La derivata rispetto a π è

$$\frac{\partial l}{\partial \pi} = \frac{\partial}{\partial \pi} \sum_{i=1}^{n} (t_i \log \pi + (1 - t_i) \log(1 - \pi)) = \sum_{i=1}^{n} \left(\frac{t_i}{\pi} - \frac{(1 - t_i)}{1 - \pi} \right) = \frac{n_1}{\pi} - \frac{n_2}{1 - \pi}$$

che è uguale a 0 per

$$\pi = \frac{n_1}{n}$$

Il massimo rispetto a $\boldsymbol{\mu}_1$ (e $\boldsymbol{\mu}_2$) si ottiene calcolando il gradiente

$$\nabla_{\boldsymbol{\mu}_1} l = \nabla_{\boldsymbol{\mu}_1} \sum_{i=1}^{n} t_i \log(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \Sigma)) = \Sigma^{-1} \sum_{i=1}^{n} t_i (\mathbf{x}_i - \boldsymbol{\mu}_1)$$

e ponendolo a 0, dal che risulta

$$\boldsymbol{\mu}_1 = \frac{1}{n_1} \sum_{\mathbf{x}_i \in C_1} \mathbf{x}_i$$

e

$$\boldsymbol{\mu}_2 = \frac{1}{n_2} \sum_{\mathbf{x}_i \in C_2} \mathbf{x}_i$$

Massimizzare la log-verosimiglianza rispetto a Σ fornisce infine

$$\Sigma = \frac{n_1}{n} \mathbf{S}_1 + \frac{n_2}{n} \mathbf{S}_2$$

dove

$$\mathbf{S}_1 = \frac{1}{n_1} \sum_{\mathbf{x}_i \in C_1} (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^T$$

$$\mathbf{S}_2 = \frac{1}{n_2} \sum_{\mathbf{x}_i \in C_2} (\mathbf{x}_i - \boldsymbol{\mu}_2)(\mathbf{x}_i - \boldsymbol{\mu}_2)^T$$