Lez19_ProbabilisticModel

December 5, 2023

1 Modelli probabilistici

1.1 Esperimento di Bernoulli

Tiriamo una moneta, e sappiamo che il risultato può essere 0 0 1, con stessa probabilità. Se tiriamo nuovamente la moneta, può essere utile basarci sulla storia passata?

Abbiamo $P(x = 1, \theta) = \theta$ e $P(x = 0, \theta) = 1 - \theta$, che possiamo compattare con: $P(x, \theta) = \theta^x \cdot (1 - \theta)^{1-x}$

Ha senso assumere, in questo caso, che i vari risultati siano *indipendenti* ed *identicamente distribuiti*. Possiamo allora prendere la *probabilità congiunta delle uscite* come:

$$P(x_1, \ldots, x_N; \theta) = \prod_{i=1}^{N} \theta^{x_i}(1 - \theta)^{1-x_i}$$

Ciò che vorremmo capire è **quale sia il valore di** θ. Abbiamo l'esperimento, e vogliamo trovare questo parametro.

1.2 Likelihood Function (verosimiglianza)

Definisce la probabilità che, quello che abbiamo visto, avvenga. In formule:

$$L(\theta) = \prod_{i=1}^{N} \theta^{x_i}(1 - \theta)^{1-x_i}$$

che sarebbe uguale a $P(x_1, x_2, .., \theta)$. Ci sta dicendo quanto è probabile osservare ciò che abbiamo osservato, dato uno specifico θ.

Visto ciò che ho osservato, il valore θ più probabile è quello che **massimizza la funzione di verosimiglianza**.

Per semplicità, si utilizza la **log-verosimiglianza**, ovvero coi logaritmi. Il massimo coindice, e il vantaggio è che, lavorando coi prodotti, passando ai logaritmi, in realtà si lavora con somme, sicuramente più facile.

$$\ell(\theta) = \sum_{i=1}^{N} x_i \log \theta + (1 - x_i) \log(1 - \theta)$$

A livello di calcolo, ciò che facciamo è calcolare la derivata ed impostarla a 0

$$\begin{aligned} \frac{d\ell}{d\theta} &= \frac{d}{d\theta} \left(\sum_{i=1}^{N} x_i \log \theta + (1 - x_i) \log(1 - \theta) \right) \\ &= \frac{d}{d\theta} \left(N_H \log \theta + N_T \log(1 - \theta) \right) \\ &= \frac{N_H}{\theta} - \frac{N_T}{1 - \theta} \end{aligned}$$

dove $N_H = \sum_i x_i$ e $N_T = N - \sum_i x_i$

e quindi, impostandola a 0, esplicitiamo rispetto θ:

$$\hat{\theta}_{ML} = \frac{N_H}{N_H + N_T}$$

che possiamo vedere, nell'esempio di Bernoulli, come il numero di teste diviso il numero di lanci.

1.3 Generative vs Discriminative

Ricordiamo che, nelle regressioni logistiche, tiriamo una retta che separa le classi, sopra c'è il positivo, sotto il negativo. Questo è un approccio **discriminativo**.

Ora vediamo un approccio **generativo**, ovvero ci chiediamo da che distribuzione vengono i nostri dati. Non ci siamo mai chiesti da dove venissero!

Iniziamo quindi a chiederci *come sono fatte le classi?*

1.4 Bayes Classifier

E' un tipo di modello generativo. Usato per classificare le mail in *spam* o *non-spam*.

Supponiamo una mail avente tale oggetto:

"You are one of the very few who have been selected as a winners for the free $1000 Gift Car".

Possiamo usare un vettore "bag-of-words feature" $\bar{x}$ di cardinalità D, contenente parole associate allo spam, per indicare le parole che occorrono nella e-mail. Quindi vediamo, per ogni parola, se è presente in questo "dizionario" dello spam.

Il classificatore di Bayes si basa sul *teorema di Bayes*:

$$\underbrace{P(c|\mathbf{x})}_{\text{Pr. class given words}} = \frac{P(\mathbf{x}, c)}{P(\mathbf{x})} = \frac{\overbrace{P(\mathbf{x}|c)}^{\text{Pr words given class}} \quad P(c)}{P(\mathbf{x})}$$

dove abbiamo la probabilità congiunta $P(x, c)$.

Supponiamo di avere una *proprietà a priori*, ovvero nel 30% dei casi, le email sono spam.

$$\text{posterior} = \frac{\text{Class likelihood} \times \text{Prior}}{\text{Evidence}}$$

Quindi **prior** perchè la sappiamo prima, **posterior** perchè la troviamo in funzione di *prior*.

Possiamo trovare il *denominatore* $P(x)$ mediante il teorema della probabilità totale, ovvero:

$$P(\mathbf{x}) = P(\mathbf{x}|c = 0)P(c = 0) + P(\mathbf{x}|c = 1)P(c = 1)$$

Tipicamente, visto che vogliamo massimizzare, compareremo varie espressioni a parità di denominatore, e quindi in realtà non serve calcolarlo se è uguale per tutti!d

1.5 Naive Bayes

Risolto il problema del denominatore nella formula, c'è però un problema al numeratore:

Come trovo $P(c)$ *e di* $P(x, c)$*?*

Assumiamo le due classi *spam* e *non-spam* con un dizionario di D parole e binary feature $\bar{x} = [x_1, ..., x_D]$.

Specificare una distribuzione congiunta $P(c, c_1, ..., x_D)$ ci porterebbe abbastanza informazioni per determinare $P(c)$ e $P(\bar{x}, c)$.

Tuttavia, una distribuzione congiunta su $D+1$ variabili binarie richiede $2^{D+1}-1$ entries. Ci serve quindi un modo compatto e che ci permetta di eseguire *learning* ed *inferenza*.

Entra in gioco *Naive Bayes*, che semplifica il lavoro poichè introduce il concetto di *condizionalmente indipendenti* nella classe c.

Questo vuol dire che x_i ed x_j sono indipendenti sotto $P(\bar{x}|c)$, non indipendimenti in *generale*.

$$P(c, x_1, \ldots, x_D) = P(c)P(x_1, \ldots, x_D|c)$$
reduces to
$$P(c, x_1, \ldots, x_D) = P(c)P(x_1|c)\ldots, P(x_D|c)$$

Stiamo stimando meno parametri rispetto a $2^{D+1}-1$, ne abbiamo $2D+1$.

Chiamiamo:

- $P(c=1) = \pi$
- $P(x_j = 1|c) = \theta_{jc}$, ovvero la probabilità condizionata di una parola data una classe.

1.6 Learning

I parametri possono essere appresi in modo efficiente, in quanto possiamo scomporre in termini indipendenti.

$$\begin{aligned}
\ell &= \sum_{i=1}^{N} \log P(c^{(i)}, \mathbf{x}^{(i)}) \\
&= \sum_{i=1}^{N} \log \left(P(\mathbf{x}^{(i)}|c^{(i)}) P(c^{(i)}) \right) \\
&= \sum_{i=1}^{N} \log \left(P(c^{(i)}) \prod_{j=1}^{D} P(x_j^{(i)}|c^{(i)}) \right) \\
&= \sum_{i=1}^{N} \left(\log P(c^{(i)}) + \sum_{j=1}^{D} \log P(x_j^{(i)}|c^{(i)}) \right) \\
&= \sum_{i=1}^{N} \log P(c^{(i)}) + \sum_{j=1}^{D} \sum_{i=1}^{N} \log P(x_j^{(i)}|c^{(i)})
\end{aligned}$$

Ciò che facciamo è chiederci: Abbiamo dei *documenti*, da 1 ad N, a cui associamo delle mail, e di cui sappiamo quali sono o non sono spam. Questo è il nostro *dataset*.

Ora ci chiediamo, quale è la probabilità di avere *questi documenti* a disposizione?

Essa è pari a $P(\bar{x}^i, c^i)$, ma dobbiamo farlo per tutte!

Allora è $\prod_{i=1}^{N} P(\bar{x}^i, c^i)$

- A riga 2, scriviamo $P(x,c) = P(x|c)P(c)
- A riga 3, usiamo Bayes, cioè $P(x|c) = P(x_1, ..., x_D|c) = \prod_{i=1}^{N} P(x_i|c)$
- A riga 4, usiamo la proprietà del logaritmo, in quanto nell'argomento c'è un prodotto, e lo spezziamo in somma dei logaritmi.
- A riga 5 risolviamo le parentesi.

Questa sommatoria finale, ha una proprietà interessante. Abbiamo infatti riscritto la funzione come *somma di cose*.

Supponiamo di avere $f(\pi, \theta_0, ..., \theta_D 1)$ con π probabilità che la mail sia spam. Per trovare i punti di massimo e minimo, dobbiamo calcolarci le derivate. Ma con le somme è più facile, possiamo riscrivere la funzione come $f_1(x_1) + f_1(x_1) + ...$ cioè stiamo trovando massimi e minimi in modo semplice.

Inoltre, così facendo, π compare solo una volta.

Possiamo immaginarlo come la formula della moneta, ovvero: $P(c^{(i)} = 1) = \pi$

$$\sum_{i=1}^{N} \log P(c^{(i)}) = \sum_{i=1}^{N} c^{(i)} \log \pi + \sum_{i=1}^{N} (1 - c^{(i)}) \log(1 - \pi)$$

Obtain MLEs by setting derivatives to zero:

$$\hat{\pi} = \frac{\sum_i \mathbb{1}[c^{(i)} = 1]}{N} = \frac{\text{\# spam in dataset}}{\text{total \# samples}}$$

Come vediamo, alla fine ritorniamo sempre al rapporto tra spam nel dataset e i sample totali.

Possiamo riversare queste osservazioni anche sui restanti elementi!

Log-likelihood:

$$\sum_{i=1}^{N} \log P(x_j^{(i)}|c^{(i)}) = \sum_{i=1}^{N} c^{(i)} \left(x_j^{(i)} \log \theta_{j1} + (1 - x_j^{(i)}) \log(1 - \theta_{j1}) \right) + \sum_{i=1}^{N} (1 - c^{(i)}) \left(x_j^{(i)} \log \theta_{j0} + (1 - x_j^{(i)}) \log(1 - \theta_{j0}) \right)$$

Obtain MLEs by setting derivatives to zero:

$$\hat{\theta}_{jc} = \frac{\sum_i \mathbb{1}[x_j^{(i)} = 1 \;\&\; c^{(i)} = c]}{\sum_i \mathbb{1}[c^{(i)} = c]} \overset{\text{for } c=1}{=} \frac{\text{\# word } j \text{ appears in spam}}{\text{\# spams in dataset}}$$

Stiamo trovando cose ovvie, ma lo abbiamo fatto in modo rigoroso. Assurdo!

Abbiamo creato un modello matematico per classificare le parole nelle email in *c'è* o *non c'è* la parola. Non contiamo ancora quante volte ci sia.

Ma come le usiamo? Come classifichiamo?

1.7 Inference - Inferenza

Mi arriva una mail, come la classifico?

Applichiamo Bayes:

$$P(c|\mathbf{x}) = \frac{P(c)P(\mathbf{x}|c)}{\sum_{c'} P(c')P(\mathbf{x}|c')} = \frac{P(c)\prod_{j=1}^{D} P(x_j|c)}{\sum_{c'} P(c')\prod_{j=1}^{D} P(x_j|c')}$$

qualche calcolo dopo:

$$h(\mathbf{x}) = \arg\max_{c_k} \log \left(P(c = c_k) \prod_{j=1}^{D} P(x_j|c = c_k) \right)$$

$$= \arg\max_{c_k} \log P(c = c_k) + \sum_{j=1}^{D} \log P(x_j|c = c_k)$$

1.8 Di nuovo su Bayes

Abbiamo sempre **training time** (per stimare i parametri usando la massima verosimiglianza) e **test time** (usando Bayes). E' un approccio Naive, quindi non il più accurato.

1.9 MLE issue: Data sparsity

Con tanti documenti, può capitare che certe occorrenze siano sbilanciate.

Basta lanciare una moneta due volte, ed avere due volte lo stesso risultato:

$$\theta_{ML} = \frac{N_H}{N_H + N_T} = \frac{2}{2+0} = 1$$

Ovviamente, due volte è troppo poco. Però, con poche parole, qualcuna potrebbe non occorrere, e portare a **data sparsity**, quindi non mi appaiono nel dataset.

Un fix facile è aggiungere 1 a ciascuna occorrenza. Tale fix porta al **Laplace smoothing**:

$$\theta_{ML} = \frac{(N_H + 1)}{(N_H + 1) + (N_T + 1)} = \frac{N_H + 1}{N_H + N_T + 2} = \frac{3}{4} = 0.75$$

sicuramente meglio rispetto alla stima precedente!