High Dimentional Data Analysis

Sommario

Il corso è erogato, parzialmente in via telematica, in italiano. L'esame è composto da un progetto (individuale o svolto in gruppi di massimo tre persone) accompagnato da una prova scritta comprendente domande aperte e a risposta multipla.

Indice

L'analisi di dati può avere obiettivi di predizione o di inferenza: per il primo è sufficiente utilizzare un modello di *machine learning*, mentre per il secondo si passa da un modello allenato su un campione all'intera popolazione. Per la predizione, non è necessario conoscere il nesso tra la variabile risposta alle variabili di input: il modello è usato come *black box*; nell'inferenza invece è necessario conoscere il nesso tra le variabili, utilizzando modelli statistici interpretabili. Lo *statistical learning* si pone a metà tra le due discipline: tenta di prevedere una variabile risposta basandosi su processi interpretabili di *machine learning*

Con il proliferare dei *big data*, è facile raccogliere grosse quantità di dati da analizzare: *i dati non sono conoscenza* (Albert Einstein), quindi è necessario processarli per analizzarli. Per il processo di analisi, è tipico lavorare su una matrice di dati X di dimensioni $n \times p$ che possono diventare problematiche: se n diventa troppo grande si hanno problemi computazionali (troppi dati per poter analizzarli), mentre se p diventa troppo grande si rischia anche di cadere nella *maledizione della dimensionalità*.

Parte I
Decomposizione Distorsione-Varianza

Un modello di *supervised learning* prevede la variabile Y in funzione di una matrice di dati X:

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

dove f rappresenta una funzione fissa e non nota e ε la componente stocastica, indipendente da X e con media nulla. In pratica l'equazione è semplificata con:

$$\hat{Y} = \hat{f}(X)$$

siccome non è possibile prevedere l'errore ε (per definizione). Per scopi predittivi, la funzione $\hat{f}$ è stimata e non necessariamente interpretabile; mentre per fare inferenza è necessario conoscerne almeno la forma funzionale. Per il processo di inferenza, si cerca di capire quali e quante sono le variabili più importanti nel processo di stima e il loro rapporto con la variabile risposta. Il processo di stima della funzione f si dice *parametrico* se si ipotizza una forma funzionale, altrimenti è detto *non parametrico*. Un metodo parametrico impone una forma funzionale, stimando i parametri in base ai dati a disposizione: si sceglie generalmente una funzione semplice e di facile interpretazione (anche a discapito della forza predittiva) per rendere più facile il processo di inferenza. Una funzione non parametrica invece predilige la bontà di adattamento a scapito dell'interpretabilità dei parametri: in questo caso è più importante il risultato del procedimento.

Il processo di stima si effettua con un processo di ottimizzazione di una generica funzione di perdita; una delle funzioni più utilizzate è l'errore quadratico medio:

$$E[(Y - f(X))^2]$$

che minimizza errori sia in positivo che in negativo (considerando i quadrati). Si dimostra che la funzione di regressione minimizza i quadrati dei residui. Non è possibile prevedere esattamente Y in funzione di X data la presenza di una componente stocastica ε: si ha in ogni caso un errore *irriducibile*. Si ha inoltre un errore dovuto alla *distorsione* nel caso in cui la stima della funzione f appartiene a una classe diversa da quella della funzione stimata $\hat{f}$. Può darsi anche che, avendo a disposizione pochi dati o di bassa qualità, si abbia una forte varianza nelle stime e i valori dei parametri siano particolarmente influenzati dal campione considerato (*varianza di stima*).

Si definiscono il vettore risposta y e la matrice del disegno X come:

$$\underset{n\times 1}{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{bmatrix} \qquad \underset{n\times p}{X} = \begin{bmatrix} x_{1.1} & \cdots & x_{1.j} & \cdots & x_{1.p} \\ \vdots & & \vdots & & \vdots \\ x_{i.1} & \cdots & x_{i.j} & \cdots & x_{i.p} \\ \vdots & & \vdots & & \vdots \\ x_{n.1} & \cdots & x_{n.j} & \cdots & x_{n.p} \end{bmatrix}$$

dove per convenzione l'indice i scorre sulle righe e j sulle colonne. Ogni valore y_i è realizzazione di una variabile casuale $Y_i = f(x_i) + \varepsilon_i$, dove la prima parte è la funzione (ignota) di regressione e la seconda l'errore. Si presuppone che ogni ε_i sia identicamente distribuito, con media nulla e varianza costante. Per il *test set* invece si contrassegna la matrice con un asterisco (star): $y^* = \hat{f}(x) + \varepsilon^*$.

Si calcola l'*errore quadratico medio* sul *test set*, che è generalmente l'indice da minimizzare:

$$MSE_{Te} = \frac{1}{n}\sum_{i=1}^{n}(y_i^* - \hat{f}(x_i))^2$$

Si minimizza il valore calcolato sul *test set* e non sul *train set* per garantire alte prestazioni su dati non osservati. Si definisce *errore di previsione* (PE) l'errore medio della stima sui nuovi dati:

$$\begin{aligned} PE &= E[MSE_{Te}] \\ &= E[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \hat{f}(x_i))^2] \\ &= \frac{1}{n}\sum_{i=1}^{n}E[(y_i^* - \hat{f}(x_i))^2] \end{aligned}$$

Questo indice è il valore da minimizzare: si cerca di ridurre l'errore sul *test set* indipendentemente dal campionamento effettuato. Essendo $\hat{y}^* = \hat{f}(x)$, si scompone la

varianza come:

$$
\begin{aligned}
\sigma_y^2 &= E[(y_i^* - \hat{y_i}^*)^2] \\
&= E[(y_i^* - f(x_i) + f(x_i) - \hat{y_i}^*)^2] \\
&= E[(y_i^* - f(x_i))^2] + E[(f(x_i) - \hat{y_i}^*)^2] + 2E[(y_i^* - f(x_i))(f(x_i) - \hat{y_i}^*)] \\
&= E[(y_i^* - f(x_i))^2] + E[(f(x_i) - \hat{y_i}^*)^2] + E[(y_i^* - f(x_i))]E[(f(x_i) - \hat{y_i}^*)] \\
&= E[(y_i^* - f(x_i))^2] + E[(f(x_i) - \hat{y_i}^*)^2] + E[f(x_i) + \varepsilon_i - f(x_i))]E[(f(x_i) - \hat{y_i}^*)] \\
&= E[(y_i^* - f(x_i))^2] + E[(f(x_i) - \hat{y_i}^*)^2] + E[\varepsilon_i]E[(f(x_i) - \hat{y_i}^*)] \\
&= E[(y_i^* - f(x_i))^2] + E[(f(x_i) - \hat{y_i}^*)^2] + 0 \\
&= \underbrace{E[(\varepsilon_i^*)^2]}_{\text{errore irriducibile}} + \underbrace{E[(f(x_i) - \hat{f}(x_i))^2]}_{\text{errore riducibile}} \\
&= \sigma_\varepsilon^2 + E[(f(x_i) - E[\hat{f}(x_i)] + E[\hat{f}(x_i)] - \hat{f}(x_i))^2] \\
&= \sigma_\varepsilon^2 + E[(\hat{f}(x_i) - f(x_i))^2] + E[(\hat{f}(x_i) - E[\hat{f}(x_i)])^2] + 2E[(\hat{f}(x_i) - E[\hat{f}(x_i)])(E[\hat{f}(x_i)] - f(x_i))] \\
&= \sigma_\varepsilon^2 + E[(\hat{f}(x_i) - f(x_i))^2] + E[(\hat{f}(x_i) - E[\hat{f}(x_i)])^2] + 0 \\
&= \sigma_\varepsilon^2 + (E[\hat{f}(x_i)] - f(x_i))^2 + Var(\hat{f}(x_i)) \\
&= \underbrace{\sigma_\varepsilon^2}_{\text{errore irriducibile}} + \frac{1}{n}\sum_{i=1}^{n}\underbrace{(E[(\hat{f}(x_i)] - f(x_i))^2}_{\text{distorsione}^2} + \frac{1}{n}\sum_{i=1}^{n}\underbrace{Var(\hat{f}(x_i))}_{\text{varianza}}
\end{aligned}
$$

Tutte e tre le quantità sono positive: è impossibile minimizzare contemporaneamente le tre quantità, bisogna trovare un compromesso che si adatti alle esigenze del singolo caso. Si ha un'alta distorsione con modelli non flessibili (lineare) e alta varianza con modelli particolarmente variabili e flessibili (funzioni complesse).

Nel caso specifico del modello lineare, utilizzando lo stimatore a minimi quadrati (non distorto, Teorema di Gauss-Markov[1]), la distorsione sarà nulla e l'unico errore (riducibile) è dovuto alla varianza del modello. Quindi qualsiasi altro metodo di stima

[1]Lo stimatore a minimi quadrati è BLUE: *Best Linear Unbiased Estimator*.

offre un MSE maggiore o uguale a quello ottenuto coi minimi quadrati.

$$
\begin{aligned}
Err &= \sigma^2 + 0 + \frac{1}{n}\sum_{i=1}^{n} Var(\hat{f}(x_i)) \\
&= \sigma^2 + \frac{1}{n}\sum_{i=1}^{n} Var(x_i'\hat{\beta}) \\
&= \sigma^2 + \frac{1}{n} \cdot trace(Var(X\hat{\beta})) \\
&= \sigma^2 + \frac{1}{n} \cdot trace(\underbrace{X(X'X)^{-1}X'}_{H} y) \\
&= \sigma^2 + \frac{1}{n} \cdot trace(H \underbrace{\sigma^2 I_p}_{Var(y)} H) \\
&= \sigma^2 + \frac{\sigma^2}{n} \cdot trace(H) \\
&= \sigma^2 + \frac{\sigma^2}{n} \cdot trace(X(X'X)^{-1}X') \\
&= \sigma^2 + \frac{\sigma^2}{n} \cdot trace(I_p) \\
&= \sigma^2 + \frac{\sigma^2 \cdot p}{n}
\end{aligned}
$$

La varianza riducibile del modello lineare dipende solamente dal numero di regressori p: per ridurre la varianza del modello sono preferibili matrici di bassa dimensionalità.

1 Applicazione in R

Si consideri la simulazione in R:

```
n <- 50
p <- 30
X <- matrix(rnorm(n * p), nrow = n)
# si estraggono 10 parametri con valori "bassi"
# e 20 con valori "alti": alcuni coefficienti sono
# utili e altri no
b.star <- c(runif(10, 0.5, 1), runif(20, 0, 0.3))
mu <- as.numeric(X %*% bstar)
```

Così facendo si genera una matrice X di dimensioni 50×30 (contenente valori casuali) e un vettore di parametri stimati $\hat{\beta}$ contenente alcuni coefficienti significativi e altri no. Si ottiene, col metodo Montecarlo la distribuzione dell'errore del modello:

```r
R <- 100
fit <- matrix(0, R, n)
err <- numeric(fit)
for (i in 1:R) {
  y <- mu + rnorm(n)
  y.hat <- mu + rnorm(n)
  mod <- lm(y ~ X + 0)  # modello di regressione senza intercetta
  bls <- coef(mod)
  fit[i, ] <- X %*% bls
  err[i] <- mean((y_hat - fit[i, ])^2)  # MSE
}
# si calcolano quindi le singole componenti della varianza:
prediction.error <- mean(err)
bias <- sum((colMeans(fit) - mu)^2) / n
var <- sum(apply(fit, 2, var)) / n
```

I valori dovrebbero tendere ai valori teorici: `bias` tende a 0, `var` tende a $\frac{p}{n}$ e la loro somma (più 1 per la varianza dell'errore casuale dato da `rnorm`) tende al valore di `prediction.error`.

In generale, aumentando il grado del polinomio interpolante, l'MSE diminuisce progressivamente sul *train set*: il modello diventa sempre più flessibile, riducendo la distorsione. Tuttavia, all'aumento del grado del polinomio, la variabilità del modello aumenta drasticamente rischiando di ridurre la capacità di generalizzazione: il modello si adatta perfettamente al *train set* ma perde completamente capacità predittiva sul *test set* e quindi significato (si verifica *overfitting*). Il modello quindi si adatta al rumore trascurando il segnale di fondo.

2 Ottimismo

Con *ottimismo* si intende la differenza tra le prestazioni di previsione del modello sul *train set* rispetto a quelle del *test set*:

$$Opt = E[MSE_{Te}] - E[MSE_{Tr}] \quad > 0$$

Nel caso del *fixed-x settings*:

$$\begin{aligned} Opt &= E[\frac{1}{n}\sum_{i=1}^{n}(y_i^* - \hat{f}(x_i))^2 - \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2] \\ &= \frac{2}{n}\sum_{i=1}^{n} Cov(y_i, \hat{f}(x_i)) \end{aligned}$$

più alta è la correlazione e i valori stimati dal modello, più l'ottimismo è elevato.

Nel caso del modello lineare, l'ottimismo si riscrive come:

$$Err_F = E[MSE_{Te}] = E[MSE_{Tr}] + Opt_F$$

L'ottimismo offre una stima dell'errore di previsione:

$$\hat{Err}_F = MSE_{Tr} + \hat{Opt}_F = MSE_{Tr} + \frac{2\sigma^2 p}{n}$$

Si dimostra che il valore atteso dell'MSE per il *train set* è minore del valore atteso per il *test set* (e dunque che l'*ottimismo* è positivo):

$$\begin{aligned} E[(y_i - \hat{y}_i)^2] &= Var(y_i - \hat{y}_i) + E[(y_i - \hat{y}_i)]^2 \\ &= Var(y_i) + Var(\hat{y}_i) - 2Cov(y_i - \hat{y}_i) + (E[y_i] + E[\hat{y}_i])^2 \\ &= Var(y_i) + Var(\hat{y}_i) - 2Cov(y_i - \hat{y}_i) \end{aligned}$$

$$\begin{aligned} E[(y_i^* - \hat{y}_i)^2] &= Var(y_i^* - \hat{y}_i) + E[(y_i^* - \hat{y}_i)]^2 \\ &= Var(y_i^*) + Var(\hat{y}_i) - 2Cov(y_i^* - \hat{y}_i) + (E[y_i^*] + E[\hat{y}_i])^2 \\ &= Var(y_i) + Var(\hat{y}_i) \end{aligned}$$

y_i^* e y_i hanno la stessa distribuzione, stessa varianza e stesso valore atteso, inoltre y_i^* e $\hat{y}_i$ sono indipendenti. Dunque:

$$\begin{aligned} Opt &= E[MSE_{Te}] - E[MSE_{Tr}] \\ &= [Var(y_i) + Var(\hat{y}_i) - \frac{2}{n}\sum_{i=1}^{n} Cov(y_i - \hat{y}_i)] - [Var(y_i) + Var(\hat{y}_i)] \\ &= -\frac{2}{n}\sum_{i=1}^{n} Cov(y_i - \hat{y}_i) = -\frac{2}{n}\sum_{i=1}^{n} \sigma^2 H_{i.i} = \frac{2\sigma^2 p}{n} \end{aligned}$$

Dunque l'ottimismo è direttamente proporzionale rispetto a σ^2 e p e inversamente proporzionale rispetto a n. Ottimizzando l'MSE del *training set* si ignora quindi la distorsione data dall'ottimismo. Il problema della formula è che σ^2 è incognita: la migliore stima è usare la somma dei quadrati dei residui RSS (*Residual Sum of Squares*).

$$\sigma^2 = \frac{RSS}{n-p}$$

$$\hat{Err} = MSE_{Tr} + 2\frac{p}{n} \cdot \frac{RSS}{n-p}$$

La stima dell'errore è chiamato *indice C_p di Mallow*: è una penalità, che tiene conto della complessità del modello. Questo indice è usato per quantificare la bontà del modello: più è basso, migliore è il modello.