Università degli Studi di Pisa

Dipartimento di Matematica
Corso di Laurea Magistrale in Matematica

Tesi di Laurea

MODELLI REGRESSIVI IN PRESENZA DI MOLTI FATTORI E APPLICAZIONI

Relatore:
Prof. *Franco Flandoli*

Candidato:
Alessia Angelini

Controrelatore:
Prof. *Marco Romito*

Anno Accademico 2014-2015
16 Ottobre 2015

A mia sorella Francesca, agli inizi della sua carriera universitaria, con l'augurio che sia un'esperienza formativa e coinvolgente quanto la mia.

Indice

Capitolo 0

Introduzione

Lo scopo principale di questa tesi è riportare i principali metodi regressivi statistici, e una volta descritti questi, mostrare i loro risultati su dati reali.
In particolare la prima parte della tesi è dedicata al richiamo dei metodi statistici più noti, come la regressione lineare, la regressione lineare multipla e il metodo delle componenti principali (PCA). Questi metodi sono analizzati sia dal punto di vista matematico, che dal punto di vista pratico.
Il secondo capitolo introduce un metodo statistico meno noto dei precedenti: il metodo dei minimi quadrati parziali, in inglese partial least square (PLS). Dimostreremo che questo metodo è utile nel caso in cui si vogliano relazionare n variabili di input con m variabili di output. La principale innovazione di questo metodo consiste nell'idea sulla quale si basa la ricerca di nuove variabili nei due spazi generati dalle variabili di input e di output: mentre PCA cerca le componenti di massima variabilità, PLS cerca le coppie di componenti più correlate tra di loro. Questa strategia risulta vincente dal momento che vogliamo usare il metodo per scopi di predizione.
Il terzo capitolo raccoglie l'esperienza di stage che ho svolto nei mesi di Giugno e Luglio nell'azienda Geal SpA: l'azienda si occupa della gestione dell'acquedotto del comune di Lucca, che conta 40000 utenti. Durante lo stage ho cercato di risolvere il seguente problema: la creazione di un modello per la previsione del totale dei consumi mensili a partire dalla conoscenza di un gruppo ristetto di utenze (circa 700). L'azienda si è posta questo problema in quanto su circa 700 utenze ha installato un sistema di lettura del contatore molto veloce (sistema di telelettura), dunque non sarebbe un problema andare a leggere anche tutti questi contatori e, se si disponesse di un metodo per prevedere il consumo totale a partire da queste misurazioni, avrebbe rilevazioni mensili più precise. Infatti attualmente l'azienda effettua la lettura di tutti i contatori due volte all'anno, ogni sei mesi, e per ricavare il consumo mensile di un utente agisce nel seguente modo: al momento della lettura del contatore calcola la differenza di metri cubi dalla lettura effettuata sei mesi prima, a questo punto divide questa differenza per i giorni trascorsi tra le due letture ottenendo così un consumo giornaliero costante. Un volta ottenuto il consumo giornaliero della specifica utenza, per ottenere il consumo mensile, basta moltiplicare tale consumo per il numero dei giorni del mese interessato. Per ottenere

poi il consumo totale del comune di Lucca di un mese specifico, l'azienda non ha altro metodo a disposizione se non la ovvia somma del consumo di tutte le utenze di quel mese. Ovviamente questo dato non è un dato realistico, in quanto c'è una forzatura alla base del metodo che consiste nel considerare il consumo giornaliero della singola utenza costante.
Dopo la dettagliata illustrazione dei dati che ho avuto a disposizione seguono le prove con vari modelli per capire quale sia il migliore per il nostro scopo.
Infine, in appendice, sono riportati i codici usati per l'implementazione dei vari metodi con il software R.

Capitolo 1

Regressione lineare e PCA

In questo primo capitolo ricorderemo velocemente i concetti della regressione lineare e del metodo delle componenti principali, in preparazione al metodo del partial least square introdotto nel capitolo successivo.

1.1 Regressione lineare

Date due variabili aleatorie X e Y, la regressione lineare si pone come scopo quello di trovare una relazione lineare tra le due variabili, della forma

$$Y = aX + b + \epsilon.$$

La variabile aleatoria X verrà detta input, mentre Y variabile di output. La variabile aleatoria ϵ rappresenta una perturbazione della relazione tra le due variabili.
In pratica possiamo cercare una relazione di questo tipo ogni qual volta si abbiano due vettori $X \in \mathbb{R}^k$ e $Y \in \mathbb{R}^k$, dove ogni elemento dei vettori rappresenta un'osservazione, e si cerchi una relazione tra le due variabili.
Quello che vogliamo calcolare, in modo approssimativo, sono i coefficienti a e b e $Var(\epsilon)$ quando si hanno due vettori X e Y di dati sperimentali nel caso in cui si ipotizzi una relazione lineare. Per fare ciò partiremo da un teorema teorico per poi tradurlo in termini pratici.

Teorema 1.1. *Supponiamo che le variabili aleatorie X, Y ed ϵ siano legate dalla relazione lineare $Y = aX + b + \epsilon$. Supponiamo inoltre che $Cov(X, \epsilon) = 0$, $E[\epsilon] = 0$ e che $Var(X) \neq 0$ allora i coefficienti a e b sono univocamente determinati*

$$a = \frac{Cov(Y, X)}{Var(X)}$$

$$b = E[Y] - aE[X].$$

Inoltre vale che

$$Var(\epsilon) = Var(Y) - a^2 Var(X).$$

Dimostrazione. Facendo la speranza della relazione lineare otteniamo, per linearità

$$E[Y] = aE[X] + b.$$

Vale inoltre per le proprietà soddisfatte dalla varianza

$$Var(Y) = a^2 Var(X) + Var(\epsilon).$$

Inoltre calcolando la covarianza tra Y ed X otteniamo

$$Cov(Y, X) = Cov(aX + b + \epsilon, Y) =$$

$$= aCov(X, X) + Cov(\epsilon, X)$$

da cui

$$Cov(Y, X) = aVar(X).$$

Da queste relazioni si ottiene la tesi. □

Vediamo adesso come tradurre questo teorema in casi pratici. Supponiamo di avere k osservazioni della variabile X, che indicheremo con $x_1, \ldots, x_k$ e k osservazioni della variabile Y, che indichiamo con $y_1, \ldots, y_k$. Questi dati possono essere interpretati come k coppie (x_i, y_i), per $i = 1, \ldots, k$. A partire da questi dati possiamo andare a calcolarci delle stime dei valori che compaiono nella tesi del teorema. Più precisamente vediamo come approssimare ciascun valore.

1. La speranza di X, $E[X]$, può essere approssimata come $\bar{x} = \frac{1}{k}\sum_{i=1}^{k} x_i$, e analogamente la speranza di Y;
2. la varianza di X, può essere stimata da $\frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})^2$, e analogamente $Var(Y)$;
3. la covarianza tra X e Y può essere approssimata come $\frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})(y_i - \bar{y})$.

Andando a sostituire nelle formule dimostrate nel teorema, le varie stime possiamo ottenere delle stime empiriche dei coefficienti a e b che indicheremo con $\hat{a}$ e $\hat{b}$.
In seguito chiameremo retta di regressione, associata a dei dati sperimentali, la retta $y = \hat{a}x + \hat{b}$.
Vediamo adesso che i parametri $\hat{a}$ e $\hat{b}$ sono i coefficienti che minimizzano la norma di ϵ, con la condizione che la speranza di ϵ sia nulla.

Teorema 1.2. *Date k coppie di dati sperimentali, e supponendo che $\bar{\epsilon}$=0, i coefficienti $\hat{a}$ e $\hat{b}$, definiti come sopra, sono quelli che minimizzano $||\epsilon||$.*

Dimostrazione. Supponiamo di aver trovato una relazione della forma

$$y = ax + b + \epsilon,$$

e che $\bar{\epsilon} = 0$. Da questo segue che

$$\bar{\epsilon} = \bar{y} - a\bar{x} - b.$$

Dunque necessariamente

$$\hat{b} = \bar{y} - a\bar{x}.$$

Adesso che $\hat{b}$ è fissato, andiamo a determinare a in modo che la norma dell'errore sia minimizzato. Andiamo a riscrivere la norma dell'errore osservando che

$$\epsilon_i = y_i - (ax_i + \hat{b}) = y_i - ax_i - \bar{y} + a\bar{x} = (y_i - \bar{y}) - a(x_i - \bar{x})$$

e otteniamo

$$||\epsilon||^2 = \sum_{i=1}^{k} \epsilon_i^2 = \sum_{i=1}^{k} [(y_i - \bar{y}) - a(x_i - \bar{x})]^2 =$$

$$= \sum_{i=1}^{k} (y_i - \bar{y})^2 - 2a \sum_{i=1}^{k} (x_i - \bar{x})(y_i - \bar{y}) + a^2 \sum_{i=1}^{k} (x_i - \bar{x})^2.$$

Andando a derivare questa espressione rispetto ad a e ponendo uguale a 0 otteniamo che il minimo si ottiene per $\hat{a}$ così definito

$$\hat{a} = \sum_{i=1}^{k} \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{k} (x_i - \bar{x})^2},$$

ossia proprio come lo abbiamo ricavato nel teorema precedente. □

Osservazione 1.3. Avendo supposto che la media di ϵ sia nulla, andare a minimizzare la norma di tale vettore corrisponde a minimizzare la covarianza empirica di esso.

1.2 Regressione lineare multipla

Introduciamo adesso il concetto di regressione lineare multipla, strettamente collegato al modello precedente, ma nel caso in cui ci siano più di un fattore in input. Partiamo a descrivere il modello sempre partendo da variabili aleatorie, per poi tradurlo nella pratica. Supponiamo di avere $X_1, \ldots, X_n$ e Y variabili aleatorie e ci chiediamo se le variabili siano legate da una relazione funzionale, a meno di un errore

$$Y = f(X_1, \ldots, X_n, \epsilon).$$

Come fatto precedentemente, le variabili X_i sono dette variabili di input, mentre la variabile Y è detta di output.
Più precisamente ci poniamo lo scopo di trovare una relazione lineare del tipo

$$Y = a_1 X_1 + \cdots + a_n X_n + b + \epsilon.$$

Vediamo subito come tradurre la seguente scrittura nel caso si abbiano dei dati sperimentali. Supponiamo inanzitutto di avere k osservazioni raccolte nella seguente tabella:

	X_1	X_2	...	X_n	Y
1	x_{11}	x_{12}	...	x_{1n}	y_1
2	x_{21}	x_{22}	...	x_{2n}	y_2
...	...	...	...	...	...
k	x_{k1}	x_{k2}	...	x_{kn}	y_k

Ogni riga della tabella rappresenta un'osservazione. A partire da questi dati specifici cerchiamo una relazione del tipo

$$y_i = a_1 x_{i1} + \cdots + a_n x_{in} + b + \epsilon_i \text{ per } \ i = 1, \ldots, k$$

dove i numeri ϵ_i sono definiti dalla relazione come

$$\epsilon_i = y_i - (a_1 x_{i1} + \cdots + a_n x_{in} + b).$$

Il fatto principale da capire è quanto sono grandi questi errori. Se riuscissimo a trovare dei coefficienti $a_1, \ldots, a_n$ in modo che gli errori fossero piccoli, potremmo ritenere valida l'ipotesi che tra le variabili sussista una relazione lineare.
Per misurare la bontà del modello possiamo introdurre una grandezza chiamata scarto quadratico medio, che adesso andiamo a definire.

Definizione 1.4. Lo scarto quadratico medio associato a una tabella di dati è una funzione dei parametri $a_1, \ldots, a_n, b$ così definita:

$$SQM : \mathbb{R}^{n+1} \to \mathbb{R}$$

tale che

$$SQM(a_1, \ldots, a_n, b) = \frac{1}{k} \sum_{i=1}^{k} (y_i - (a_1 x_{i1} + \cdots + a_n x_{in} + b))^2 = \frac{1}{k} \sum_{i=1}^{k} \epsilon_i^2.$$

Tale grandezza, come già sottolineato, misura la bontà del modello lineare. La strada che seguiremo sarà quella di trovare i parametri che minimizzano questa quantità: chiameremo tali coefficienti $\hat{a}_1, \ldots, \hat{a}_n, \hat{b}$. Chiamiamo allora il modello

$$Y = \hat{a}_1 X_1 + \cdots + \hat{a}^n X_n + \hat{b} + \epsilon$$

modello di regressione lineare multipla associato alla tabella di dati precedenti. Nel prossimo paragrafo calcoleremo i coefficienti in vari modi.

1.2.1 Calcolo dei coefficienti da un modello teorico

Come prima cosa, vediamo il calcolo dei coefficienti a partire da un modello teorico, per analogia a quanto fatto nel caso della regressione lineare.

Teorema 1.5. *Supponiamo che le variabili aleatorie $X_1, \dots, X_n$, Y ed ϵ siano legate dalla relazione*

$$Y = a_1 X_1 + \cdots + a_n X_n + b + \epsilon.$$

Supponiamo inoltre che $Cov(X_i, \epsilon) = 0$ per ogni i, $E[\epsilon] = 0$, e la matrice di covarianza Q del vettore $(X_1, \dots, X_n)$ sia invertibile. Indichiamo con $c \in \mathbb{R}^n$ il vettore di coordinate $c_j = Cov(Y, X_j)$ e con $a_0 \in \mathbb{R}^n$ il vettore

$$a_0^T = (a_1, \dots, a_n).$$

Allora i parametri $a_1, \dots, a_n$, b sono univocamente determinati dalla seguente relazione

$$a_0 = Q^{-1} c,$$

$$b = E[Y] - (a_1 E[X_1 + \cdots + a_n E[X_n]]).$$

Dimostrazione. Iniziamo a calcolare il generico elemento del vettore c_j.

$$c_j = Cov(X_j, Y) = Cov(X_j, a_1 X_1 + \cdots + a_n X_n + b + \epsilon) =$$

$$= a_1 Cov(X_j, X_1) + \cdots + a_n Cov(X_j, X_n) + Cov(X_j, \epsilon) =$$

$$= a_1 Cov(X_j, X_1) + \cdots + a_n Cov(X_j, X_n).$$

Usando la matrice di covarianza del vettore $(X_1, \dots, X_n)$, possiamo riscrivere la relazione precedente come

$$c_j = \sum_{i=1}^{n} Q_{ij} a_i \overset{simm.}{=} \sum_{j=1}^{n} Q_{ji} a_i.$$

Quindi $Q a_0 = c$, e usando l'invertibilità della matrice Q abbiamo che $a_0 = Q^{-1} c$.
Per calcolare b basta calcolare il valor medio dell'equazione che definisce il modello, e usare la linearità della speranza, ottenendo

$$E[Y] = a_1 E[X_1] + \cdots + a_n E[X_n] + b,$$

e dunque vale

$$b = E[Y] - (a_1 E[X_1] + \cdots + a_n E[X_n]).$$

□

Analogamente a quanto fatto nel caso della regressione lineare semplice, traduciamo i risultati ottenuti in pratica: se abbiamo una matrice di dati per calcolare i coefficienti $\hat{a_0}$ e $\hat{b}$ è necessario calcolare la matrice di covarianza empirica $\hat{Q}$ nel seguente modo

$$\hat{Q}_{ij} = Cov(X_i, X_j) = \frac{1}{k} \sum_{h=1}^{k} (x_{hi} - \bar{x_i})(x_{hj} - \bar{x}_j),$$

dove

$$\bar{x}_i = \frac{1}{k}\sum_{h=1}^{k} x_{hi}.$$

Una volta calcolata $\hat{Q}$, si calcola il vettore $\hat{c}$ delle covarianze empiriche dei vettori di dati X_j e Y, e si calcolano i valori

$$\hat{a}_0 = \hat{Q}^{-1}\hat{c}.$$

In modo analogo si calcola il vettore $\hat{b}$.

1.2.2 Calcolo dei coefficienti mediante il metodo dei minimi quadrati

Illustriamo adesso un metodo alternativo per il calcolo dei coefficienti. Questo metodo ha il pregio di risultare più chiaro perchè parte direttamente dai dati empirici, piuttosto che dalle variabili aleatorie, ma risulta un poco più complicato nei calcoli.
Per semplicità riportiamo in forma vettoriale il modello. Introduciamo la matrice degli input, introducendo una colonna di 1 finale per manipolare l'intercetta. La matrice X è la seguente

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} & 1 \\ \dots & \dots & \dots & \dots \\ x_{k1} & \dots & x_{kn} & 1 \end{pmatrix}.$$

In modo analogo raccogliamo nei vettori a, y ed ϵ le seguenti componenti

$$a = \begin{pmatrix} a_1 \\ \dots \\ a_n \\ b \end{pmatrix}, \quad Y = \begin{pmatrix} y_1 \\ \dots \\ y_k \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \dots \\ \epsilon_k \end{pmatrix}.$$

Mediante le seguenti notazioni otteniamo la riscrittura vettoriale del modello come

$$Y = Xa + \epsilon.$$

Possiamo allora riscrivere l'errore quadratico medio come

$$SQM(a) = \frac{1}{k}\sum_{i=1}^{k} \epsilon_i^2 = \frac{1}{k}||\epsilon||^2 = \frac{1}{k}||Y - Xa||^2.$$

Lo scopo della regressione lineare multipla è la minimizzazione dell'errore quadratico medio.

Interpretazione geometrica

Grazie alla riscrittura del problema in forma vettoriale possiamo dare un'interpretazione geometrica interessante del problema.

Definizione 1.6. Data una matrice $X \in \mathbb{R}^{k\times n}$ definiamo il sottospazio vettoriale $\Im(X)$ come

$$\Im(X) := \{Z \in \mathbb{R}^k |\ \exists a \in \mathbb{R}^n \text{ t.c. } \ Z = Xa\}.$$

Osserviamo che $\Im(X) \subset \mathbb{R}^k$.

Nell'ambito del nostro problema di minimizzazione, abbiamo il vettore $Y \in \mathbb{R}^k$ e il sottospazio $\Im(X) = \{Z \in \mathbb{R}^k |\ Z = Xa, a \in \mathbb{R}^{n+1}\} \subset \mathbb{R}^k$.
Il nostro problema può essere riscritto come

$$\min_{\{a\in\mathbb{R}^{n+1}\}} SQM(a) = \min_{\{a\in\mathbb{R}^{n+1}\}} \frac{1}{k}||Y - Xa||^2 = \min_{\{Z\in\Im(X)\}} \frac{1}{k}||Y - Z||^2$$

e dunque, una volta trovato il vettore $\hat{Z}$ che soddisfa $\min_{\{Z\in\Im(X)\}} ||Y - Z||^2$, si tratta di trovare il vettore $\hat{a}$ tale che

$$\hat{Z} = X\hat{a}.$$

Per la ricerca dei punti $\hat{Z}$ ed $\hat{a}$ abbiamo bisogno di alcuni lemmi.

Lemma 1.7. *Dato un sottospazio $H \subset \mathbb{R}^k$ e un vettore $Y \in \mathbb{R}^k$ esiste un unico $Z \in H$, che denotiamo con $Z := P_H(Y)$, tale che*

$$||Z - Y|| \leq ||T - Y|| \text{ per ogni } \ T \in H.$$

Inoltre il vettore $P_H(Y) - Y$ è ortogonale al sottospazio H.

Omettiamo la dimostrazione, che risulta essere un banale conto.

Lemma 1.8. *Se* $\ker(X) = 0$ *allora* $\det(X^TX) \neq 0$.

Dimostrazione. Se $\ker(X) = 0$, l'applicazione lineare associata alla matrice X risulta iniettiva. Dunque si ha che se $Xv = 0$, necessariamente il vettore v risulta essere il vettore nullo. Dunque se $v \neq= 0$, si ha che $Xv \neq 0$ e dunque

$$0 \neq ||Xv||^2 =< Xv, Xv >=< X^TXv, v > .$$

Da questo segue che $X^TXv \neq 0$. Riassumendo abbiamo visto che per ogni $v \neq 0$si ha $X^TXv \neq 0$. Dunque abbiamo che la matrice X^TX ha nucleo nullo. Da questo segue anche che $\det(X^TX) \neq 0$, come volevasi dimostrare. □

Lemma 1.9. *Supponiamo che valga* $\ker(X) = 0$, *allora, esiste uno ed un solo vettore $\hat{a}$ che minimizza la funzione $SQM(a)$ ed è dato da*

$$\hat{a} = (X^TX)^{-1}X^TY.$$

Se inoltre $n + 1 = k$ allora $\hat{a} = X^{-1}Y$, ed in tal caso il minimo vale 0.

Dimostrazione. Sia $\hat{Z} \in \Im(X)$, $\hat{Z} = P_{\Im(X)}(Y)$. Allora per quanto dimostrato precedentemente abbiamo che $\hat{Z}$ minimizza $||Y - Z||$ al variare di Z. Sia inoltre $\hat{a}$ tale che $X\hat{a} = \hat{Z}$. Osserviamo che un tale $\hat{a}$, esiste ed è unico, per l'ipotesi che $\ker(X) = 0$. Vogliamo adesso dimostrare che $\hat{a}$ minimizza $SQM(a)$. Sia infatti $a \neq \hat{a}$, e $Z = Xa \in \Im(X)$. Si ha allora che $Z \neq \hat{Z}$ e dunque

$$||Y - \hat{Z}|| < ||Y - Z||,$$

ossia

$$||Y - X\hat{a}|| < ||Y - Xa||,$$

e dunque $SQM(\hat{a}) < SQM(a)$ per ogni $a \neq \hat{a}$. Con queste osservazioni abbiamo dimostrato che $\hat{a}$ è l'unico punto di minimo della funzione $SQM(a)$.
Poichè $\hat{a}$ soddisfa $X\hat{a} = \hat{Z}$, allora si ha

$$X^T X\hat{a} = X^T\hat{Z}.$$

Per il lemma precedente abbiamo che la matrice $X^T X$ è invertibile, e dunque vale che

$$\hat{a} = (X^T X)^{-1} X^T\hat{Z}.$$

Vogliamo adesso vedere che $\hat{a} = (X^T X)^{-1} X^T Y$. Per vedere ciò è sufficiente mostrare che $X^T\hat{Z} = X^T Y$. Questo segue dal fatto che $\hat{Z} - Y$ è perpendicolare a $\Im(X)$ e dunque, per ogni a vale

$$0 =< \hat{Z} - Y, Xa >=< X^T(\hat{Z} - Y), a > .$$

Da questo segue che $X^T(\hat{Z} - Y) = 0$ e dunque $X^T\hat{Z} = X^T Y$. Quindi segue che $\hat{a} = (X^T X)^{-1} X^T Y$.
Ci rimane da dimsotrare il caso $n + 1 = k$. In tal caso abbiamo che le matrici X ed X^T sono invertibili e quindi

$$\hat{a} = X^{-1}(X^T)^{-1} X^T Y = X^{-1} Y.$$

In tal caso inoltre

$$SQM(\hat{a}) = \frac{1}{k}||Y - X\hat{a}||^2 = \frac{1}{k}||Y - Y||^2 = 0,$$

come volevasi dimostrare. □

Osservazione 1.10. In generale, l'errore relativo ai parametri $\hat{a}$, $\hat{\epsilon} = Y - X\hat{a}$ è ortogonale a $\Im(X)$. Quindi abbiamo che

$$< \hat{\epsilon}, Xa >= 0 \text{ per ogni } \ a \in \mathbb{R}^{n+1},$$

ma da questo segue che $< X^T\hat{\epsilon}, a >= 0$ per ogni a, e dunque $X^T\hat{\epsilon} = 0$. Esplicitando l'ultima componente del vettore $X^T\hat{\epsilon}$ otteniamo che

$$\epsilon_1 + \cdots + \epsilon_k = 0,$$

ossia la speranza empirica del vettore ϵ risulta nulla, $\bar{\epsilon} = 0$. Da ciò segue che $SQM(\hat{a})$ risulta essere la varianza empirica del vettore ϵ. In conclusione abbiamo trovato i coefficienti che minimizzano la varianza empirica dell'errore.

Il caso appena illustrato funziona nel caso in cui $Ker(X) = 0$, ossia la matrice X risulti iniettiva. Ho riportato la dimostrazione perchè risulta molto chiara dal punto di vista geometrica. Vediamo adesso come si agisce nel caso in cui non richiediamo l'ipotesi di iniettività della matrice X.

1.2.3 Calcolo dei coefficienti nel caso generale

In tal caso abbiamo bisogno di alcuni richiami sulla decomposizione a valori singolari (SVD).

SVD

Richiamiamo il teorema fondamentale della decomposizione a valori singolari e alcuni semplici lemmi.

Teorema 1.11. *Sia $X \in \mathbb{R}^{k\times(n+1)}$. Esistono $U \in \mathbb{R}^{k\times k}$, $V \in \mathbb{R}^{(n+1)\times(n+1)}$ ortogonali e $S \in \mathbb{R}^{n\times m}$ tali che*

$$X = USV^T,$$

$$S = diag(\sigma_1, \dots, \sigma_m) \in \mathbb{R}^{k\times(n+1)}, \text{ dove } m = \min\{k, p+1\} \text{ e}$$

$$\sigma_i \geq \sigma_j \text{ se } i < j.$$

Lemma 1.12. *Sia $U \in \mathbb{R}^{n\times n}$ ortogonale e $x \in \mathbb{R}^n$. Allora vale che*

$$||x|| = ||U^T x|| = ||Ux||.$$

Dimostrazione. Per ipotesi la matrice U soddisfa $U^TU = Id$ e $UU^T = Id$. Abbiamo quindi che

$$||Ux||^2 = < Ux, Ux > = < U^TUx, x > = ||x||^2,$$

e da questo segue la nostra tesi.

□

Lemma 1.13. *Supponiamo che la matrice X ammetta una decomposizione SVD della seguente forma*

$$X = USV^T.$$

Allora si ha che

$$||Y - Xa|| = ||U^TY - SV^Ta||.$$

Dimostrazione. Si ha

$$||Y - Xa|| = ||Y - USV^Ta|| \overset{lemma 1.12}{=} ||U^T(Y - USV^Ta)|| = ||U^TY - SV^Ta||,$$

e questo conclude la nostra dimostrazione. □

Lemma 1.14. *Sia r il rango della matrice X ($r \leq m$). Definiamo $\alpha = V^T a \in \mathbb{R}^{n+1}$. Allora vale che*

$$||Y - Xa||^2 = \sum_{i=1}^{r}(\sigma_i\alpha_i - u_i^T Y)^2 + \sum_{i=r+1}^{k}(u_i^T Y)^2.$$

Dimostrazione. Sia $\bar{r}$ il più grande intero tale che $\sigma_{\bar{r}}$ sia positivo. Allora poichè in tal caso il rango della matrice S risulta essere $\bar{r}$, segue che $\bar{r} = r$, poichè S ed X hanno lo stesso rango. Usando il lemma precedente siamo riusciti a scrivere:

$$||Y - Xa|| = ||U^T Y - SV^T a||.$$

Andiamo a esplicitare la i-esima componente del vettore riportato a destra della precedente uguaglianza.

$$(U^T Y - SV^T a)_i = -\sigma_i\alpha_i + (u_i)^T Y.$$

Dunque

$$||Y - Xa||^2 = ||U^T Y - SV^T a||^2 = \sum_{i=1}^{r}(\sigma_i\alpha_i - u_i^T Y)^2 + \sum_{i=r+1}^{k}(u_i^T Y)^2,$$

come volevasi dimostrare.

□

Dopo questi fatti preliminari arriviamo finalmente al teorema fondamentale.

Teorema 1.15. *Una soluzione ottima del problema è data da*

$$\hat{a} = \sum_{i=1}^{r}\frac{u_i^T Y}{\sigma_i}v_i.$$

Dimostrazione. Il problema si è ridotto a minimizzare $\sum_{i=1}^{r}(\sigma_i\alpha_i - u_i^T Y)^2$ al variare di α. Per tale scopo basta prendere

$$\hat{\alpha}_i = \frac{u_i^T Y}{\sigma_i} \text{ per } i = 1, \ldots, r \text{ e}$$

$$\hat{\alpha}_i = 0 \text{ per } \ i = r+1, \ldots, n+1.$$

Ricordando che $\alpha = V^T a$, otteniamo

$$\hat{a} = V\hat{\alpha} = \sum_{i=1}^{n+1}\hat{\alpha}_i v_i = \sum_{i=1}^{r}\frac{u_i^T Y}{\sigma_i}v_i,$$

come volevasi dimostrare. □

Nel caso in cui il rango della matrice X sia r, definiamo la matrice

$$\Sigma^{-1} = \left(\begin{array}{c|c} \begin{matrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_r} \end{matrix} & 0 \\ \hline 0 & 0 \end{array}\right)$$

e la pseudo inversa della matrice X come $X^+ := V\Sigma^{-1}U^T$. Il teorema precedente ci dice quindi che

$$\hat{a} = X^+ Y.$$

Osserviamo adesso che la matrice pseudo inversa soddisfa la seguente proprietà

Lemma 1.16. *Se X ha rango massimo la pseudo inversa soddisfa*

$$X^+ = (X^T X)^{-1} X^T.$$

Quindi il caso precedente non è nient'altro che un caso particolare di questo teorema.

1.2.4 Osservazioni sulla regressione lineare multipla

Overfitting

Supponiamo che $\det X \neq 0$ e che $n + 1 = k$. In tal caso otteniamo dei parametri $\hat{a}$ che soddifsano

$$SQM(\hat{a}) = 0.$$

Da ciò si deduce che il modello costruito si adatta perfettamente ai dati usati per costruire il modello stesso, a tal punto da rendere gli errori tutti uguali a 0. A prima vista, sembrerebbe che un tale modello risulti il migliore possibile. In realtà, come vedremo nel terzo capitolo su dati reali, il suo potere predittivo può essere scarsissimo. Infatti un tale modello si adatta perfettamente ai dati sperimentali, seguendo i suoi accidenti casuali, senza riconoscere che alcuni variazioni sono rumore, non struttura da catturare. Per capire meglio quello che stiamo cercando di spiegare a parole, vediamo un esempio con due variabili aleatorie X e Y. Supponiamo di conoscere di queste due variabili solo due dati sperimentali: $(1, y_1)$ e $(2, y_2)$, dove y_1 e y_2 rappresentano dei numeri casuali $N(0, 1)$. Chiaramente il modello corretto sarebbe dato da $\hat{a} = \hat{b} = 0$, ossia

$$Y = \epsilon.$$

Se invece cerchiamo un modello di regressione lineare tra queste due variabili, il software ci fornirà come risposta la retta che passa per i punti $(1, y_1)$ e $(2, y_2)$. Da questo si capisce che otterremo dei residui nulli, ma il modello dato dalla retta è completamente senza senso, ed oltrettutto fortemente dipendente dai dati usati per costruire il modello. Nel caso in cui n sia troppo grande al numero di dati a disposizione si può incorrere in overfitting.

Variabilità statistica dei parametri ottimali

Supponiamo di fissare i valori x_{ij} mediante esperimenti. I valori della variabile Y corrispondenti non possono essere fissati a priori, ma li possiamo osservare e registrare. Supponiamo che le nostre grandezze siano legate dalla relazione

$$Y = \tilde{a}_1 X_1 + \cdots + \tilde{a}_n X_n + \tilde{b} + \epsilon$$

secondo dei precisi coefficienti $\tilde{a}_1,\ldots,\tilde{a}_n$, $\tilde{b}$, che però non conosciamo, dove ϵ rappresenta un disturbo casuale di media nulla e variazione strandard σ_ϵ. Effetuiamo k esperimenti, e otteniamo dei valori $y_1, \ldots, y_k$, che registriamo. Questi valori sono dati dalla formula

$$y_i = \tilde{a}_1 x_{1i} + \cdots + \tilde{a}_n x_{ni} + \tilde{b} + \epsilon_i$$

dove gli ϵ_i sono le realizzazioni casuali dell'errore accaduto in tali esperimenti. Come già sottolineato, non conosciamo i parametri $\tilde{a}_1,\ldots,\tilde{a}_n$, $\tilde{b}$, ma tramite il metodo dei minimi quadrati riusciamo ad arrivare ai coefficienti $\hat{a}_1,\ldots,\hat{a}_n$, $\hat{b}$. Come fatto precedentemente indichiamo con $\hat{a} = (\hat{a}_1, \ldots, \hat{a}_n, \hat{b})$ e analogamente per $\tilde{a}$. Vogliamo illustrare adesso le proprietà di $\hat{a}$.

Teorema 1.17. *Lo stimatore $\hat{a}$ è non distorto, ossia*

$$E[\hat{a}] = \tilde{a}$$

e la matrice di covarianza di $\hat{a}$ è

$$Q_{\hat{a}} = \Big[(X^T X)^{-1} X^T\Big] Q_\epsilon \Big[X (X^T X)^{-1}\Big].$$

In particolare, se $Q_\epsilon = \sigma_\epsilon^2 Id$, cioè se gli errori sono indipendenti tra loro, allora

$$Q_{\hat{a}} = \sigma_\epsilon^2 (X^T X)^{-1}.$$

Omettiamo la dimostrazione del precedente teorema, in quanto risulta molto semplice.

1.3 PCA

L'analisi delle componenti principali o PCA, dall'inglese principal component analysis, è una tecnica per la semplificazione dei dati utilizzata nell'ambito della statistica multivariata. Fu proposta nel 1901 da Pearson, e sviluppata successivamente da Hotelling (1933). Lo scopo primario di questa tecnica è la riduzione di un numero elevato di variabili in un numero minore o uguale di variabili scorrelate. In seguito dimostreremo che le nuove variabili ottenute sono combinazioni lineari delle precedenti e tramite esse le caratterizzazioni, le classificazioni e la struttura degli individui sono molto più chiare. Un grande vantaggio del metodo PCA, rispetto ad altri metodi di riduzione, è la perdita minima di informazione. Nel tentativo di ridurre le variabili PCA si propone altri obiettivi fondamentali per il raggiungimento dello scopo: ad esempio cerca di individuare la

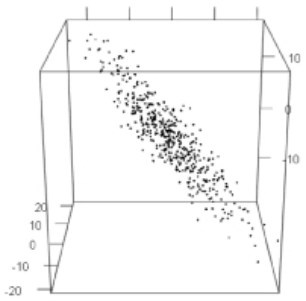

Figura 1.1: Nuvola di dati

rappresentazione dei dati più visibile possibile, in modo da riconoscere eventuali cluster e relazioni tra gli individui. Vediamo subito con un esempio cosa si intende con maggiore visibilità. Supponiamo di avere dei dati rappresentabili tramite la nuvola di punti di figura 1.1.

Ovviamente il cubo può essere ruotato in varie posizioni, alcune che permettono una migliore visibilità dei dati, altre che peggiorano la situazione. Ad esempio una buona soluzione del problema è la terza rotazione rappresentata in figura 2.7. In questa

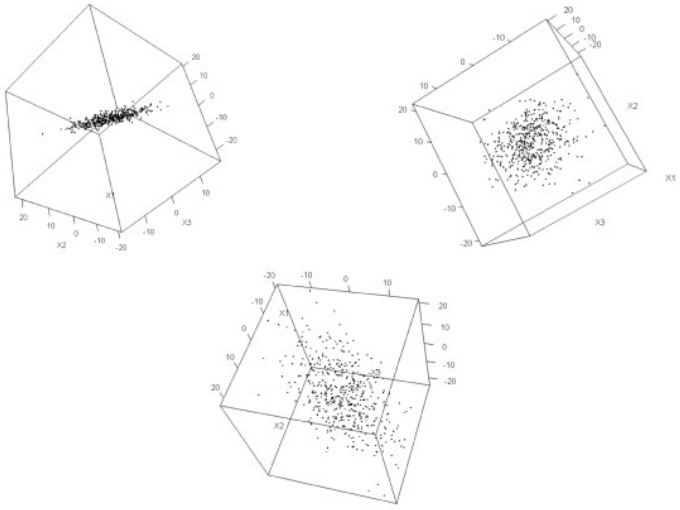

Figura 1.2: Rotazioni

prospettiva i punti appaiono più *sparpagliati* possibile. PCA ottiene facilmente la migliore rappresentazione mediante le prime componenti principali.
In questa trattazione esploreremo il metodo da due punti di vista, come fatto per i metodi precedenti: dal punto di vista matematico e dal punto di vista statistico.

1.3.1 Descrizione teorica

Supponiamo di avere $X = (X_1, \dots, X_n)$ un vettore aleatorio, che rappresenta l'informazione del nostro problema. Chiamiamo, come nei paragrafi precedenti Q la matrice di covarianza del vettore X, ossia la matrice con elementi

$$Q_{ij} = Cov(X_i, X_j) = E[(X_i - E[X_i])(X_j - E[X_j])].$$

Riportiamo nel teorema successivo tutte le proprietà di tale matrice, che ci saranno utili nel seguito.

Teorema 1.18. *La matrice di covarianza Q soddisfa le seguenti proprietà:*

1. *$Q^T = Q$, ossia è una matrice simmetrica;*
2. *$y^T Q y \geq 0$ per ogni $y \neq 0$, ossia è semidefinita positiva;*
3. *soddisfa la seguente proprietà: $< Qy, z >= Cov(< X, y >, < X, z >)$.*

Dimostrazione. 1. Segue dalla definizione della matrice Q.

2.
$$y^T Q y = \sum_{i,j} Q_{ij} y_i y_j = \sum_{i,j} Cov(X_i, X_j) y_i y_j = \sum_{i,j} Cov(y_i X_i, y_j X_j) =$$
$$= Cov\left(\sum_i y_i X_i, \sum_j y_j X_j\right) = Var[\sum_i y_i X_i] \geq 0$$

.

3.
$$< Qy, z >= \sum_{i,j} Q_{ij} y_i z_j = \sum_{i,j} Cov(X_i, X_j) y_i z_j = \sum_{i,j} Cov(y_i X_i, z_j X_j) =$$
$$= Cov(\sum_i y_i X_i, \sum_j z_j X_j) = Cov(< X, y >, < X, z >).$$

□

Dalle prime due proprietà segue che per la matrice Q vale il teorema spettrale. In particolare esiste una base ortonormale di $\mathbb{R}^n$ fatta di autovettori di Q che chiamiamo $z_1, \dots, z_n$ corrispondenti ad autovalori $\lambda_1, \dots, \lambda_n$ ordinati in modo decrescente. I suddetti vettori soddisfano una proprietà che fanno giocare loro un ruolo centrale nell'analisi delle componenti principali. Vale infatti il seguente teorema.

Teorema 1.19. *Con le notazioni precedenti, vale che*

$$z_1 = \underset{\{v \in \mathbb{R}^n |\ ||v||=1\}}{\operatorname{argmax}} Var(< X, v >)$$

e per ogni $j > 1$ vale

$$z_{j+1} = \underset{\{v \in span(z_1,\dots,z_j)^\perp |\ ||v||=1\}}{\operatorname{argmax}} Var(< X, v >).$$

Dimostrazione. Supponiamo, senza perdere di generalità, che $E[X] = 0$. Abbiamo allora che

$$Var(< X, v >) = Cov(< X, v >, < X, v >) \overset{teo\ 1.18}{=} < Qv, v > .$$

Poichè $z_1, \dots, z_n$ formano una base di $\mathbb{R}^n$, ogni $v \in \mathbb{R}^n$ può essere scritto come

$$v = \alpha_1 z_1 + \cdots + \alpha_n z_n$$

e se $||v|| = 1$ abbiamo che

$$||v||^2 = \sum_{i=1}^{n} \alpha_i^2 = 1.$$

Sia v un vettore di norma 1, allora

$$< Qv, v >=< Q(\alpha_1 z_1 + \cdots + \alpha_n z_n), \alpha_1 z_1 + \cdots + \alpha_n z_n >= \sum_{i,j} < Q\alpha_i z_i, \alpha_j z_j >=$$

$$\sum_{i=1}^{n} < \lambda_i \alpha_i z_i, \alpha_i z_i >= \sum_{i=1}^{n} \lambda_i < \alpha_i z_i, \alpha_i z_i >\leq \lambda_1 \sum_{i=1}^{n} < \alpha_i z_i, \alpha_i z_i >= \lambda_1 \sum_{i=1}^{n} \alpha_i^2 = \lambda_1.$$

Dunque per ogni $v \in \mathbb{R}^n$, con $||v|| = 1$, si ha che $< Qv, v >\leq 1$ e dunque

$$\sup_{\{v \in \mathbb{R}^n |\ ||v||=1\}} < Qv, v >\leq \lambda_1$$

ma, prendendo $v = z_1$ otteniamo

$$< Qz_1, z_1 >= \lambda_1$$

il che ci porta a concludere che

$$\operatorname*{argmax}_{\{v \in \mathbb{R}^n |\ ||v||=1\}} < Qv, v >= z_1,$$

ossia

$$z_1 = \operatorname*{argmax}_{\{v \in \mathbb{R}^n |\ ||v||=1\}} Var(< X, v >).$$

La dimostrazione del caso generale si conclude in modo analogo. □

Vogliamo adesso dimostrare un teorema più generale.

Teorema 1.20. *Sia X un vettore aleatorio in $\mathbb{R}^n$ con media μ e matrice di covarianza Q; sia $z_1, \dots, z_n$ una base ortonormale come nel teorema precedente, allora per ogni $j = 1, \dots, n$ vale che*

$$z_j = \operatorname*{argmax}_{\{v_i |\ <v_i, v_j>=\delta_{ij}\}} \sum_{i=1}^{j} Var(< X, v_i >).$$

Inoltre la proprietà enunciata sopra è equivalente alla seguente, che garantisce la perdita minima di informazione tramite gli z_j

$$z_j = \operatorname*{argmin}_{\{v_i |\ <v_i, v_j>=\delta_{ij}\}} E\left[\left|\left|X - \sum_{i=1}^{n} < X, v_i > v_i\right|\right|^2\right]$$

Dimostrazione. Per dimostrare la prima parte del teorema, procediamo per casi e supponiamo che $\mu = 0$.

1. j=1 La dimostrazione coincide con quella del teorema precedente.
2. j=2 Dato che nel caso precedente abbiamo dimostrato che z_1 massimizza la sommatoria delle varianze quando $j = 1$, ci rimane da mostrare che

$$z_2 = \underset{\{v|\ v \in span(z_1)^\perp\}}{\operatorname{argmax}} \Big(Var(< X, z_1 >) + Var(< X, v >) \Big)$$

   Dai risultati precedenti sappiamo che $Var(< X, z_1 >) = \lambda_1$ e questo non dipende da v, dunque possiamo riscrivere l'espressione precedente come

$$z_2 = \underset{\{v|\ v \in span(z_1)^\perp\}}{\operatorname{argmax}} \Big(Var(< X, z_1 >) + Var(< X, v >) \Big) =$$
$$= \lambda_1 + \underset{\{v|\ v \in span(z_1)^\perp\}}{\operatorname{argmax}} Var(< X, v >).$$

   Dunque per dimostrare la tesi è sufficiente dimostrare che

$$z_2 = \underset{\{v|\ v \in span(z_1)^\perp\}}{\operatorname{argmax}} Var(< X, v >),$$

   ma questo segue direttamente dal teorema precedente.
3. j qualsiasi Tutti i casi si dimostrano come il punto $j = 2$.

Vediamo adesso la seconda parte del teorema. Dobbiamo dimostrare che

$$\underset{\{v_i|\ <v_i,v_j>=\delta_{ij}\}}{\operatorname{argmax}} \sum_{i=1}^{j} Var(< X, v_i >) = \underset{\{v_i|\ <v_i,v_j>=\delta_{ij}\}}{\operatorname{argmin}} E\left[\left|\left| X - \sum_{i=1}^{n} < X, v_i > v_i \right|\right|^2 \right].$$

Iniziamo a sviluppare il secondo membro dell'uguaglianza.

$$E\left[\left|\left| X - \sum_{i=1}^{n} < X, v_i > v_i \right|\right|^2 \right] = E\left[\left|\left| X \right|\right|^2 - \sum_{i=1}^{n} < X, v_i >^2 \right] =$$

$$= E\big[\left|\left| X \right|\right|^2 \big] - \sum_{i=1}^{n} E[< X, v_i >^2] = E\big[\left|\left| X \right|\right|^2 \big] - \sum_{i=1}^{n} Var(< X, v_i >).$$

Quindi

$$\arg \min_{\{v_i|\ <v_i,v_j>=\delta_{ij}\}} E\left[\left|\left| X - \sum_{i=1}^{n} < X, v_i > v_i \right|\right|^2 \right] =$$

$$= \underset{\{v_i|\ <v_i,v_j>=\delta_{ij}\}}{\operatorname{argmin}} E\big[\left|\left| X \right|\right|^2 \big] - \sum_{i=1}^{n} Var(< X, v_i >) =$$

$$= E[||X||^2] - \underset{\{v_i|\ <v_i,v_j>=\delta_{ij}\}}{\operatorname{argmax}} \sum_{i=1}^{n} Var(< X, v_i >).$$

Questo dimostra la nostra tesi. □

1.3.2 Componenti principali

Adesso che abbiamo chiarito le proprietà di base, possiamo dare la definizione di componenti principali.

Definizione 1.21. Dato un vettore aleatorio X con media μ e matrice di covarianza Q, chiamiamo componenti principali gli autovettori $z_1, \ldots, z_n$ di Q, ordinati secondo la grandezza dei relativi autovalori. In particolare z_1, l'autovettore relativo al più grande autovalore, si chiama prima componente principale, z_2 seconda componente principale, e così via.
Lo spazio generato dalle prime due componenti principali si chiama piano principale.

Adesso che abbiamo trovato le componenti principali, possiamo proiettare il vettore aleatorio X su di esse, ottenendo le proiezioni

$$V_i =< X, z_i > .$$

Vogliamo adesso dimostrare che la varianza lungo queste componenti è massima, e che le variabili risultano scorrelate.

Teorema 1.22. *Con le notazioni precedenti abbiamo che*

$$Var(V_i) = \lambda_i \text{ per ogni } i = 1, \ldots, n,$$

e inoltre $Cov(V_i, V_j) = 0$, *se* $i \neq j$.

Dimostrazione. Abbiamo che

$$Cov(V_i, V_j) = Cov(< X, z_i >, < X, z_j >) =< Qz_i, z_j >= \lambda_i < z_i, z_j >= 0 \text{ se } i \neq j.$$

Ponendo $i = j$ nella uguaglianza precedente otteniamo che

$$Var(V_i) = Cov(V_i, V_i) = \lambda_i.$$

□

Dunque abbiamo trovato le variabili scorrelate e di varianza massima, che sono proprio le proiezioni di X sugli assi principali.

Osservazione 1.23. Osserviamo che le proiezioni V_i sono le coordinate di X rispetto alla base $z_1, \ldots, z_n$:

$$X = V_1 z_1 + \cdots + V_n z_n.$$

Definizione 1.24. Definiamo la varianza totale del vettore X come

$$\lambda_1 + \cdots + \lambda_n.$$

Inoltre definiamo il numero

$$\frac{\lambda_1}{\lambda_1 + \cdots + \lambda_n}$$

come proporzione di varianza spiegata dalla prima componente principale. Definiamo inoltre il numero

$$\frac{\lambda_1 + \lambda_2}{\lambda_1 + \cdots + \lambda_n}$$

come proporzione di varianza spiegata dal piano principale e così via.

Capitolo 2

Partial Least Square (PLS)

La regressione dei minimi quadrati parziali, in inglese partial least square (PLS), è usata in molte scienze applicate. Essa è utile nel caso in cui ci sia bisogno di modellizzare situazioni nelle quali abbiamo molte variabili ma non necessariamente molte osservazioni. Questa è una situazione comune in molti laboratori: tipicamente ci vuole molto tempo per ottenere una nuova osservazione, ma da ciascuna osservazione si ricavano molti parametri.

2.1 Introduzione

Il metodo dei minimi quadrati parziali viene usato per scopi di predizione. Vediamo in dettaglio in cosa consiste: dati due insiemi di variabili, $\{X_1, \ldots, X_n\}$, $X_i \in \mathbb{R}^k \ \forall i \in \{1, \ldots, n\}$ e $\{Y_1, \ldots, Y_m\}$, $Y_i \in \mathbb{R}^k \ \forall i \in \{1, \ldots, m\}$, dove il primo insieme rappresenta le variabili di input e il secondo le variabili di output, si cerca la migliore combinazione lineare delle variabili X per predire le variabili Y. Per praticità chiamiamo X la matrice le cui colonne sono i vettori $\{X_1, \ldots, X_n\}$ e Y la matrice le cui colonne sono i vettori $\{Y_1, \ldots, Y_m\}$. Dunque $X \in \mathbb{R}^{k \times n}$ e $Y \in \mathbb{R}^{k \times m}$. In questo contesto, k rappresenta il numero di osservazioni: ogni riga della matrice X e della matrice Y rappresentano un'osservazione.
Lo scopo principale della regressione dei minimi quadrati parziali è predire Y da X e descrivere la loro struttura comune. Quando $m = 1$, ossia lo spazio delle variabili di output è generato da un solo vettore, e X è di rango massimo, il nostro scopo può essere raggiunto mediante l'uso della regressione lineare multipla. Quando il numero dei predittori è molto grande rispetto al numero di osservazioni ($n >> k$), X potrebbe essere quasi singolare e la regressione lineare multipla potrebbe non esserci più di aiuto (ad esempio a causa di problemi di collinearità). In tal caso si possono seguire varie strade: la prima è l'eliminazione successiva di fattori, che viene eseguita a seconda di alcuni valori ottenuti nella regressione lineare multipla. Un'altra strada può essere quella di eseguire l'analisi delle componenti principali (PCA) nello spazio delle X e usare le componenti principali di X come predittori per Y, per eliminare i problemi di collinearità. Rimane il problema di quale sottoinsieme di componenti principali scegliere: ad esempio potremmo prendere solo le prime componenti principali. Tale metodo non è il migliore per il nostro sco-

po: infatti le componenti principali non sono quelle che catturano al meglio la variabilità nello spazio delle Y, ma, come abbiamo visto, solo quelle che meglio spiegano lo spazio X.

2.2 Introduzione al metodo PLS

Iniziamo esplicitando le matrici X e Y introdotte sopra per chiarezza. Abbiamo:

$$X = \begin{pmatrix} x_{11} & \dots & x_{1n} \\ x_{21} & \dots & x_{2n} \\ \dots & \dots & \dots \\ x_{k1} & \dots & x_{kn} \end{pmatrix} \quad Y = \begin{pmatrix} y_{11} & \dots & y_{1m} \\ y_{21} & \dots & y_{2m} \\ \dots & \dots & \dots \\ y_{k1} & \dots & y_{km} \end{pmatrix}$$

Le matrici X ed Y possono essere interpretate geometricamente in due modi diversi. Concentriamoci sulla matrice X, possiamo vedere questi dati in due modi distinti: si possono interpretare come k punti nello spazio delle variabili, ed in questo caso ogni punto rappresenta una osservazione, come in figura 2.1. In particolare il j-esimo individuo è rappresentato dalle coordinate $(x_{j1}, \dots, x_{jn})$. Un'altra interpretazione dei dati è di vederli come n punti nello spazio delle osservazioni, e in questo caso ogni punto rappresenta una variabile, come in figura 2.2. In tal caso la variabile i-esima è rappresentata dalla colonna i-esima della matrice, e più in particolare dalle coordinate $(x_{1i}, \dots, x_{ki})$.

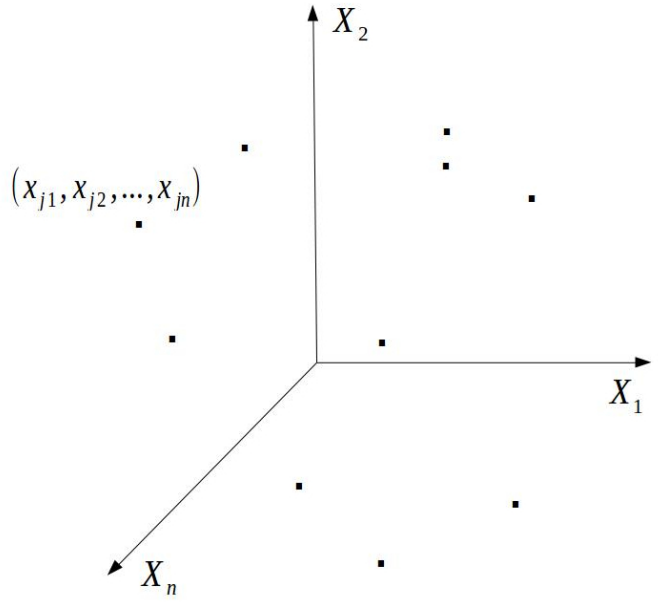

Figura 2.1: Spazio delle variabili

Analogamente possiamo fare gli stessi ragionamenti con la matrice Y. Vediamo adesso varie interpretazioni intuitive che ci guideranno verso la costruzione del metodo PLS.

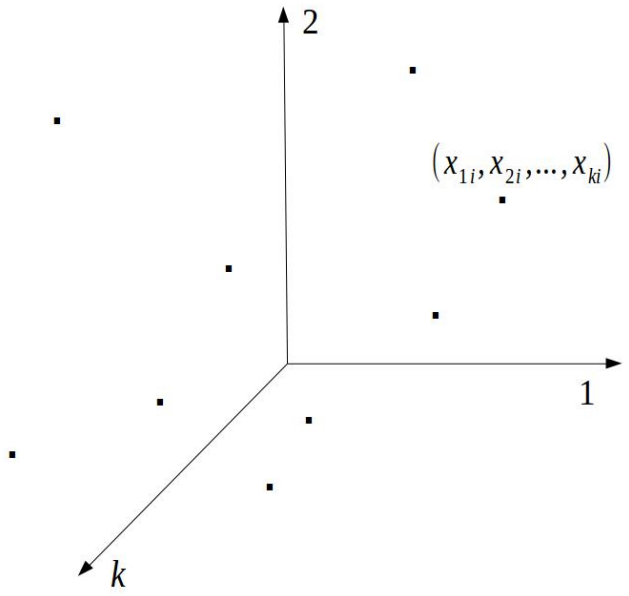

Figura 2.2: Spazio degli individui

2.2.1 Primo problema: massimizzazione della covarianza

Per enunciare questo problema risulta più naturale pensare al caso $m = 1$, ossia al caso in cui Y sia unidimensionale, per fare analogie con il caso della regressione lineare multipla. Come visto nel capitolo precedente, il metodo della regressione lineare multipla, dati la matrice X ed il vettore Y, trova dei coefficienti $a_1, \ldots, a_n$ che soddisfino:

$$Y = a_1 X_1 + \cdots + a_n X_n + \epsilon$$

dove ϵ è un vettore di $\mathbb{R}^k$ e rappresenta gli errori della relazione che siamo andati a creare. In particolare i coefficienti $a_1, \ldots, a_n$ vengono scelti per minimizzare la norma del vettore ϵ. Geometricamente quindi stiamo cercando una combinazione lineare delle colonne della matrice X per approssimare in modo **migliore** possibile il vettore Y, nel senso che vogliamo rendere questi due vettori più vicini possibile. Infatti

$$a = (a_1, \ldots, a_n)^T = \underset{b \in \mathbb{R}^n}{\operatorname{argmin}} ||\epsilon||^2 = \underset{b \in \mathbb{R}^n}{\operatorname{argmin}} ||Y - Xb||^2.$$

Ma c'è un altro modo possibile di interpretare il concetto di **migliore**: invece di cercare di rendere quanto più possibile vicini questi due vettori, si potrebbe cercare di massimizzare il legame tra loro, massimizzando la covarianza. Il problema può essere scritto come

$$\max_{\{d \in \mathbb{R}^n |\ ||d||=1\}} |Cov(Xd, Y)|.$$

Osserviamo che chiamando $\hat{d}$ il vettore che soddisfa il massimo, e indicando con $\hat{d}_i$ la componente i-esima del vettore abbiamo semplicemente trovato una combinazione lineare

delle colonne della matrice X, data da $\hat{d}_1 X_1 + \cdots + \hat{d}_n X_n$, in modo da rendere i due vettori più legati possibile. Vediamo adesso come tradurre questo teorema nel caso in cui $m > 1$. In tal caso potremmo voler cercare una combinazione delle colonne della matrice X, ma anche una combinazione lineare delle colonne della matrice Y in modo da massimizzare la covarianza. Quindi il problema enunciato sopra, nel caso di $m > 1$ può essere riscritto come

$$\max_{\{d\in\mathbb{R}^n,\ e\in\mathbb{R}^m|\ ||d||=1,||e||=1\}} |Cov(Xd, Ye)|.$$

Osserviamo che il caso $m = 1$ non è nient'altro che un caso particolare del problema appena citato. Vediamo quindi come risolvere il problema nel caso di m qualsiasi. Come prima cosa dobbiamo riscrivere in modo equivalente l'enunciato. Ricordiamo che dati due vettori f, $g \in \mathbb{R}^k$ centrati, la covarianza tra di essi può essere scritta come

$$Cov(f, g) = \frac{f^T g}{k}.$$

Da qui in avanti supporremo che le matrici X e Y siano centrate. Dunque il nostro problema può esssere scritto come

$$\max_{\{d\in\mathbb{R}^n,\ e\in\mathbb{R}^m|\ ||d||=1,||e||=1\}} |Cov(Xd, Ye)| = \max_{\{d\in\mathbb{R}^n,\ e\in\mathbb{R}^m|\ ||d||=1,||e||=1\}} \frac{|d^T X^T Y e|}{k}.$$

Chiamiamo per semplicità A la matrice $X^T Y$. Dunque

$$A = X^T Y\ ,\ A \in \mathbb{R}^{n\times m}.$$

Possiamo quindi riscrivere il nostro problema come

$$\max_{\{d\in\mathbb{R}^n,\ e\in\mathbb{R}^m|\ ||d||=1,||e||=1\}} |d^T A e|.$$

Per trovare una soluzione di questa massimizzazione abbiamo bisogno di alcuni richiami sulla decomposizione a valori singolari.

Decomposizione a valori singolari (SVD)

Ricordiamo il teorema di decomposizione a valori singolari.

Teorema 2.1. *Sia $A \in \mathbb{R}^{n\times m}$ con $n \geq m$. Esistono $U \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{m\times m}$ ortogonali e $S \in \mathbb{R}^{n\times m}$ tali che*

$$A = USV^T,$$

$$S = \begin{pmatrix} \Sigma \\ 0 \end{pmatrix},$$

dove $\Sigma \in \mathbb{R}^{m\times m}$ è una matrice diagonale con elementi σ_i sulla diagonale tali che

$$\sigma_i \geq \sigma_j \ \text{se}\ \ i < j.$$

Osservazione 2.2. Osservando che la matrice S è quasi diagonale si capisce che questa decomposozione è una sorta di decomposizione spettrale. In particolare possiamo scrivere A^TA come

$$A^TA = VS^TU^TUSV^T = V \begin{pmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & ... & \\ & & & \sigma_m^2 \end{pmatrix} V^T.$$

quindi se prendiamo come λ_i gli autovalori di A^TA, che sono reali perchè la matrice risulta essere simmetrica, ordinati in modo decrescente, abbiamo che $\sigma_i = \sqrt{\lambda_i}$ per ogni $i = 1, \ldots, m$.

Osservazione 2.3. Chiamiamo le colonne della matrice U u_i, e analogamente per le colonne della matrice V. Supponiamo che il rango di A sia k. Allora possiamo riscrivere la decomposizione come

$$A = \sum_{i=1}^{k} \sigma_i u_i v_i^T.$$

Enunciamo adesso un lemma che sarà fondamentale per alcune dimostrazioni del paragrafo successivo.

Lemma 2.4. *Una soluzione del problema*

$$\max_{\{u\in\mathbb{R}^n, v\in\mathbb{R}^m |\ ||u||=||v||=1\}} |u^TAv|$$

è data da $u = u_1$ e $v = v_1$, ossia la prima colonna della matrice U della SVD e la prima colonna della matrice V.

Dimostrazione. Siano $u \in \mathbb{R}^n$ e $v \in \mathbb{R}^m$ con $||u|| = ||v|| = 1$. Allora esistono $\eta_1, \ldots, \eta_n$ e $\xi_1, \ldots, \xi_m$ tali che

$$u = \sum_{i=1}^{n} \eta_i u_i, \quad v = \sum_{i=1}^{m} \eta_i v_i \text{ e } \quad \sum_{i=1}^{n} \eta_i^2 = \sum_{i=1}^{m} \xi_i^2 = 1.$$

Ora, poichè

$$u_i^TAv_j = 0 \text{ se } \ i \neq j,$$

si ha

$$|u^TAv| = |\sum_{i=1}^{m} \eta_i \xi_i \sigma_i| \leq \sigma_1 \sum_{i=1}^{m} |\eta_i \xi_i \leq \sigma_1.$$

Il massimo si raggiunge prendendo $u = u_1$ e $v = v_1$.

□

Osserviamo che dal teorema precedente segue un'ovvia interpretazione del maggiore valore singolare σ_1. Si ha infatti che

$$\sigma_1 = \max_{\{u\in\mathbb{R}^n, v\in\mathbb{R}^m |\ ||u||=||v||=1\}} |u^TAv|.$$

Dunque la teoria della decomposizione a valori singolari ci fornisce il seguente risultato

Teorema 2.5. *Una soluzione del problema*

$$\max_{\{d\in\mathbb{R}^n,\ e\in\mathbb{R}^m|\ ||d||=1,||e||=1\}} |Cov(Xd, Ye)|$$

si ottiene prendendo come vettore d l'autovettore relativo al più grande autovalore della matrice X^TYY^TX e come vettore e l'autovettore relativo al più grande autovalore della matrice Y^TXX^TY. In particolare il massimo vale σ_1.

La dimostrazione di questo teorema segue ovviamente dai teoremi enunciati precedentemente.

2.2.2 Secondo problema: allineamento delle colonne

In questo paragrafo vogliamo sottolineare un'altra idea che potremmo sfruttare per massimizzare la relazione tra i dati contenuti nella matrice X e i dati contenuti nella matrice Y.
Supponiamo di considerare una rotazione della matrice X. Questo corrisponde a moltiplicare la matrice X a destra per una matrice ortogonale, che chiamiamo $O_X \in \mathbb{R}^{n\times n}$. Sia $S \in \mathbb{R}^{k\times n}$ la rotazione di X:

$$S = XO_X.$$

Indichiamo con s_{ij} gli elementi della matrice S e, analogamente, con x_{ij} gli elementi della matrice X. Allora abbiamo che:

$$\sum_{i=1}^{k}\sum_{j=1}^{n} s_{ij}^2 = tr(S^TS) = tr(O_X^TX^TXO_X) = tr(X^TXO_XO_X^T) = tr(X^TX) = \sum_{i=1}^{k}\sum_{j=1}^{n} x_{ij}^2.$$

Questo mostra che la variazione totale rimane invariata per trasformazioni ortogonali. Similmente possiamo ruotare la matrice Y, con una matrice $O_Y \in \mathbb{R}^{m\times m}$:

$$Z = YO_Y.$$

Denotiamo con $\{S_1, \ldots, S_n\}$ le colonne della matrice S e, analogamente, $\{Z_1, \ldots, Z_m\}$ le colonne della matrice Z. Supponiamo che $n > m$.

Una questione di fondamentale importanza è quanto possiamo scegliere i vettori S_i vicini ai vettori Z_i. Formalmente, ci stiamo chiedendo in che modo possiamo risolvere la seguente minimizzazione:

$$\min_{\{O_X,\ O_Y|\ O_XO_X^T=Id,\ O_YO_Y^T=Id\}} \sum_{i=1}^{m} ||(XO_X)_i - (YO_Y)_i||^2 + \sum_{i=m+1}^{n} ||(XO_X)_i||^2 =$$

$$= \min_{\{O_X,\ O_Y|\ O_X O_X^T = Id,\ O_Y O_Y^T = Id\}} \sum_{i=1}^{m} ||S_i - Z_i||^2 + \sum_{i=m+1}^{n} ||S_i||^2,$$

dove, se $x^T = (x_1, \ldots, x_n) \in \mathbb{R}^n$, indichiamo con

$$||x|| = \sqrt{\sum_{i=1}^{n} x_i^2}$$

la norma ecluidea.
Per semplicità chiamiamo

$$I := \sum_{i=1}^{m} ||S_i - Z_i||^2 + \sum_{i=m+1}^{n} ||S_i||^2.$$

L'ultima sommatoria riflette il fatto che ci sono più componenti nello spazio X rispetto allo spazio Y. Supponiamo per semplicità che $n = m$. Questo non lede la generalità, in quanto, se così non fosse, basta estendere la matrice Y aggiungendo $m - n$ colonne di 0 e analogamente la matrice O_Y rendendola una matrice ortogonale di $\mathbb{R}^{n \times n}$. Dunque possiamo riscrivere il nostro problema come

$$\min_{\{O_X,\ O_Y|\ O_X O_X^T = Id,\ O_Y O_Y^T = Id\}} \sum_{i=1}^{n} ||(XO_X)_i - (YO_Y)_i||^2.$$

Vogliamo adesso trovare un'altra scrittura equivalente, ma abbiamo bisogno di alcuni richiami.

Definizione 2.6. Sia $A \in \mathbb{R}^{n \times m}$. Definiamo la norma di Frobenius:

$$||A||_F = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} |a_{ij}|^2} = \sqrt{tr(A^T A)} = \sqrt{\sum_{i=1}^{min\{n,m\}} \sigma_i^2}$$

dove σ_i sono i valori singolari di A.

Osservazione 2.7. Indichiamo con A_i per $i = 1, \ldots, n$ le colonne della matrice A. Vale la seguente proprietà:

$$||A||_F^2 = \sum_{i=1}^{n} ||A_i||^2.$$

Lemma 2.8. *Sia $A \in \mathbb{R}^{n \times m}$ e $B \in \mathbb{R}^{n \times m}$ allora*

$$||A - B||_F^2 = tr(AA^T) + tr(BB^T) - 2tr(AB^T)$$

Dimostrazione. Abbiamo:

$$||A-B||_F^2 = tr((A-B)(A^T-B^T)) = tr(AA^T+BB^T-AB^T-BA^T) =$$
$$= tr(AA^T)+tr(BB^T)-tr(AB^T+BA^T) = tr(AA^T)+tr(BB^T)-2tr(AB^T)$$

□

Ricordiamo inoltre una nota proprietà della traccia:

Osservazione 2.9. Siano A, B e C matrici di opportune dimensioni. Vale che

$$tr(ABC) = tr(CAB) = tr(BCA),$$

ossia la traccia è invariante per permutazioni cicliche.

Grazie a questi richiami possiamo allora scrivere

$$\min_{\{O_X,\ O_Y|\ O_XO_X^T=Id,\ O_YO_Y^T=Id\}} \sum_{i=1}^{n} ||(XO_X)_i-(YO_Y)_i||^2 \overset{oss.2.7}{=}$$

$$\min_{\{O_X,\ O_Y|\ O_XO_X^T=Id,\ O_YO_Y^T=Id\}} ||XO_X-YO_Y||_F^2 \overset{lemma2.8}{=}$$

$$\min_{\{O_X,\ O_Y|\ O_XO_X^T=Id,\ O_YO_Y^T=Id\}} tr(XO_XO_X^TX^T)+tr(YO_YO_Y^TY^T)-2tr(YO_YO_X^TX^T) \overset{oss.2.9}{=}$$

$$\min_{\{O_X,\ O_Y|\ O_XO_X^T=Id,\ O_YO_Y^T=Id\}} tr(XX^T)+tr(YY^T)-2tr(O_X^TX^TYO_Y)$$

Dunque possiamo riscrivere in modo equivalente il problema come

$$\max_{\{O_X,\ O_Y|\ O_XO_X^T=Id,\ O_YO_Y^T=Id\}} tr(O_X^TX^TYO_Y).$$

Vediamo allora come si possono scegliere le matrici di rotazioni per ottenere questo massimo. Abbiamo bisogno di alcuni richiami.

Definizione 2.10. Sia $A \in \mathbb{R}^{n\times m}$, con $n \geq m$. Definiamo la traccia di A come

$$tr(A) := \sum_{i=1}^{m} a_{ii}.$$

Teorema 2.11. *Sia $A \in \mathbb{R}^{n\times m}$, con $n \geq m$. Allora, abbiamo che*

$$\max_{\{X\in\mathbb{R}^{n\times n},\ Y\in\mathbb{R}^{m\times m}|\ XX^T=Id,\ YY^T=Id\}} tr(XAY^T) = \sum_{i=1}^{m} \sigma_i,$$

dove σ_i sono i valori singolari della matrice A. Inoltre se indichiamo come al solito con $A = USV^T$ la decomposizione SVD della matrice A, tale massimo si ottiene prendendo $X = U$ e $Y = V$.

Omettiamo la dimostrazione del teorema. Osserviamo quindi che abbiamo appena dimostrato il seguente risultato.

Teorema 2.12. *Una soluzione del problema*

$$\min_{\{O_X,\ O_Y|\ O_X O_X^T=Id,\ O_Y O_Y^T=Id\}} \sum_{i=1}^{m} ||(XO_X)_i - (YO_Y)_i||^2 + \sum_{i=m+1}^{n} ||(XO_X)_i||^2$$

si ottiene prendendo $O_X = U$ e $O_Y = V$ dove U e V sono le matrici che compaiono nella SVD della matrice X^TY:

$$X^TY = USV^T.$$

Inoltre tale minimo vale

$$\min_{\{O_X,\ O_Y|\ O_X O_X^T=Id,\ O_Y O_Y^T=Id\}} \sum_{i=1}^{m} ||(XO_X)_i - (YO_Y)_i||^2 + \sum_{i=m+1}^{n} ||(XO_X)_i||^2 =$$

$$= tr(X^TX) + tr(Y^TY) - 2\sum_{i=1}^{m} \sigma_i,$$

dove come al solito σ_i indicano i valori singolari della matrice X^TY.

2.2.3 Algoritmo teorico

Riassumiamo i due risultati riportati nei paragrafi precedenti, ricordando che indichiamo la SVD della matrice X^TY come

$$X^TY = USV^T,$$

e con σ_i i relativi valori singolari.

1. Una soluzione del problema

$$\max_{\{d\in\mathbb{R}^n,\ e\in\mathbb{R}^m|\ ||d||=1,||e||=1\}} |Cov(Xd, Ye)|$$

   si ottiene prendendo come vettore d l'autovettore relativo al più grande autovalore della matrice X^TYY^TX e come vettore e l'autovettore relativo al più grande autovalore della matrice Y^TXX^TY. In particolare il massimo vale σ_1.

2. Una soluzione del problema

$$\min_{\{O_X,\ O_Y|\ O_X O_X^T=Id,\ O_Y O_Y^T=Id\}} \sum_{i=1}^{m} ||(XO_X)_i - (YO_Y)_i||^2 + \sum_{i=m+1}^{n} ||(XO_X)_i||^2$$

   si ottiene prendendo $O_X = U$ e $O_Y = V$ dove U e V sono le matrici che compaiono nella SVD della matrice X^TY:

$$X^TY = USV^T.$$

Inoltre tale minimo vale

$$\min_{\{O_X,\ O_Y|\ O_X O_X^T=Id,\ O_Y O_Y^T=Id\}} \sum_{i=1}^{m} ||(XO_X)_i - (YO_Y)_i||^2 + \sum_{i=m+1}^{n} ||(XO_X)_i||^2 =$$

$$= tr(X^TX) + tr(Y^TY) - 2\sum_{i=1}^{m} \sigma_i.$$

Proviamo adesso a formulare un algoritmo basato su queste osservazioni. Per analogia di notazione con l'algoritmo di PLS che andremo a introdurre nel paragrafo successivo, denominiamo $w_1, \ldots, w_A$ i primi A autovettori della matrice X^TYY^TX, e con

$$t_i = Xw_i \text{ per } \ i = 1, \ldots, A.$$

Inoltre denoriamo con $c_1, \ldots, c_A$ i primi A autovettori della matrice Y^TXX^TY, e con

$$u_i = Yc_i \text{ per } \ i = 1, \ldots, A.$$

Con queste notazioni abbiamo che

$$|Cov(t_1, u_1)| = |Cov(Xw_1, Yc_1)| = \max_{\{||w||=||c||=1\}} |Cov(Xw, Yc)|,$$

come già dimostrato.
Definiamo inoltre le matrici C, W, T ed U che hanno come colonne rispettivamente per colonne i vettori c_i, w_i, t_i ed u_i. Allora, supponendo con $n = m$, il secondo risultato dimostrato può essere riscritto come

$$||T - U||_F^2 = ||XW - YC||_F^2 = \min_{\{O_X,O_Y|\ O_X O_X^T=Id,\ O_Y O_Y^T=Id\}} ||XO_X - YO_Y||_F^2.$$

Dunque se noi cercassimo A vettori che massimizzano la relazione tra le componenti dello spazio Y e dello spazio X in base alle precedenti osservazioni potremmo formulare un algoritmo come segue.

for($i = 1, ..., A$){w_i autovettore relativo all'i-esimo autovalore della matrice X^TYY^TX

$t_i = Xw_i$

c_i autovettore relativo all'i-esimo autovalore della matrice Y^TXX^TY

$u_i = Yc_i$

$b_i = \frac{u_i^T t_i}{||t_i||^2}$}

In seguito per riferirci al precedente pseudo codice parleremo di Algoritmo PLS teorico. In particolare il coefficiente b_i rappresenta il coefficiente di regressione del vettore t_i sul vettore u_i. In particolare dunque da questa relazione possiamo supporre che a meno di

errori valga $u_i = b_i t_i$.
Vediamo come agisce questo algoritmo in fase di predizione: supponiamo che $m = 1$ per semplicità. Abbiamo che dato un nuovo vettore $(\tilde{X}_1, \ldots, \tilde{X}_n)$, possiamo ricavare $\tilde{t}_1 = \tilde{X} w_1$, e da qui possiamo ricavare $\tilde{u}_1 = \tilde{Y} = b_1 \tilde{t}_1$. Se vogliamo aggiungere più componenti si procede allo stesso modo.
In realtà l'algoritmo PLS viene sviluppato in maniera un po' diversa da quanto proposto da noi. Introduciamo nel paragrafo successivo l'algoritmo e successivamente cerchiamo di spiegare intuitivamente il perchè delle modifiche introdotte.

2.3 L'algoritmo

Dopo questa prima costruzione dell'algortimo, vediamo l'algoritmo del metodo PLS sviluppato da Wold, e a partire da esso analizzeremo le proprietà geometriche dei vettori che esso genera. Il punto di partenza dell'algortimo, come già detto, sono due matrici $X \in \mathbb{R}^{k \times n}$ e $Y \in \mathbb{R}^{k \times m}$. Prima di iniziare l'algoritmo le matrici possono essere scalate o centrate. Scalarle corrisponde a lavorare con la matrice di correlazione, e centrarle corrisponde a sottrarre il valore medio da ogni colonna. Vediamo lo pseudo codice dell'algoritmo:

$X^{(1)} = X,\ Y^{(1)} = Y,\ u_1^0 = Y^{(1)} e_1$

for $(i = 1, \ldots, A)\{$ $\quad j = 0$, **if**$(i > 1)$ $u_i^0 = u_{i-1}$

while $(||t_i^{j-1} - t_i^{j-2}|| > \epsilon \cup j \leq 1)$**do**$\{$

$$w_i^{j+1} = \frac{(X^{(i)})^T u_i^j}{||(X^{(i)})^T u_i^j||}$$
$$t_i^{j+1} = X^{(i)} w_i^{j+1}$$
$$c_i^{j+1} = \frac{(Y^{(i)})^T t_i^{j+1}}{||(Y^{(i)})^T t_i^{j+1}||}$$
$$u_i^{j+1} = Y^{(i)} c_i^{j+1}$$
$$j = j + 1\}$$

$t_i = t_i^j,\ c_i = c_i^j,\ u_i = u_i^j,\ w_i = w_i^j$

$$p_i = \frac{(X^{(i)})^T t_i}{||t_i||^2}$$
$$q_i = \frac{(Y^{(i)})^T u_i}{||u_i||^2}$$
$$b_i = \frac{u_i^T t_i}{||t_i||^2}$$
$X^{(i+1)} = X^{(i)} - t_i p_i^T,\ Y^{(i+1)} = Y^{(i)} - b_i t_i c_i^T\}$

Le dimensioni dei vettori nell'algoritmo sono riassunte dall'immagine 2.3.

Osserviamo che nell'algoritmo selezioniamo A componenti: il numero di componenti da selezionare può essere scelto in base a vari criteri che vedremo nel corso del capitolo.

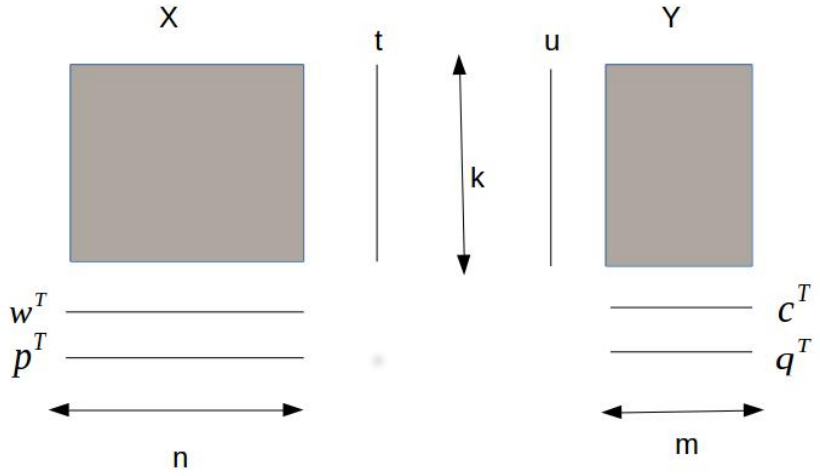

Figura 2.3: Dimensioni dei vettori

L'algoritmo è composto da un ciclo while e un ciclo for. Il primo ciclo che incontriamo è il ciclo for. Il ciclo for compie A passi, ed all'interno del passo i esimo lavora con le matrici $X^{(i)}$ e $Y^{(i)}$. Vediamo l'idea che sta dietro la riduzione delle matrici: lo scopo del ciclo for è di arrivare a scrivere le matrici X e Y come somma di matrici di rango uno. Per fare ciò una volta trovati i vettori t_i e p_i va a sottrarre il loro prodotto dalla matrice $X^{(i)}$ nel segente modo

$$X^{(i+1)} = X^{(i)} - t_i p_i^T.$$

In particolare dimostreremo più avanti nel capitolo che i vettori t_i trovati sono ortogonali. Analogamente ci si aspetterebbe che la matrice Y venisse modificata nel seguente modo

$$Y^{(i+1)} = Y^{(i)} - u_i c_i^T,$$

ma ciò non accade! Infatti la matrice Y subisce la seguente modifica

$$Y^{(i+1)} = Y^{(i)} - b_i t_i c_i^T.$$

Come mai si modifica in tal modo? Le due relazioni sono in qualche modo collegate? Le risposte a queste domande sono ovvie se andiamo a considerare il seguente passaggio dell'algoritmo:

$$b_i = \frac{u_i^T t_i}{||t_i||^2},$$

che mostra che lo scalare b_i è l'inclinazione della retta di regressione tra i vettori u_i e t_i. In particolare il coefficiente b_i è il coefficiente che rende minimo l'errore ϵ nella relazione

$$u_i = b_i t_i + \epsilon,$$

come mostrato nel capitolo precedente. Quindi a meno di approssimazioni possiamo scrivere che

$$u_i = b_i t_i,$$

il che ci porta a capire che le due relazioni enunciate non sono che la stessa relazione a meno di errori. Ma anche se le due relazioni sono simili, perchè non scegliere la prima? Questa scelta è dettata dal fatto che i vettori t_i sono ortogonali, mentre i vettori u_i non godono della stessa proprietà. In sostanza alla fine dell'algoritmo troviamo una decomposizione delle matrici, a meno di errori, nel seguente modo:

$$X = TP^T \qquad Y = TBC^T$$

dove la matrice T contiene come colonne i vettori t_i generati dall'algoritmo, la matrice P contiene come colonne i vettori p_i, la matrice C contiene come colonne i vettori c_i e la matrice B è una matrice diagonale con gli elementi b_i sulla diagonale.

Osservazione 2.13. Un'altra osservazione riguardante il ciclo for è l'idea con la quale viene modificata la matrice X ad ogni passo. Osserviamo infatti come cambiano la j-esima colonna della matrice $X^{(i+1)}$. Usiamo le notazioni già usate precedentemente: con $X_j^{(i)}$ indichiamo la colonna j-esima della matrice $X^{(i)}$ e con p_{ij} indichiamo l'elemento j-esimo del vettore p_i, e analogamente per gli altri vettori. Si ha che:

$$X_j^{(i+1)} = X_j^{(i)} - p_{ij} t_i = X_j^{(i)} - \frac{< X_j^{(i)}, t_i >}{||t_i||^2} t_i$$

e da questo segue che tutte le colonne della matrice $X^{(i+1)}$ sono ortogonali al vettore t_i. Infatti

$$< X_j^{(i+1)}, t_i > = < X_j^{(i)} - \frac{< X_j^{(i)}, t_i >}{||t_i||^2} t_i, t_i > = 0$$

per linearità. Da questo segue anche che $< t_j, t_i > = 0$ se $i \neq j$.

Vediamo adesso in dettaglio il ciclo while. Il ciclo while si stoppa nel momento in cui il vettore t_i^j non cambia più di una soglia fissata al variare di j. Possiamo quindi scrivere che

$$t_i = \lim_{j \to \infty} t_i^j$$

e la stessa cosa peri vettori c_i, w_i e u_i.
Vogliamo adesso mostrare che i vettori appena citati sono vettori definiti per ricorrenza. Vediamo che valgono infatti le seguenti relazioni:

Teorema 2.14. *I vettori creati dal ciclo while dell'algoritmo soddisfano le seguenti proprietà:*

1. $u_i^{j+1} = \frac{Y^{(i)}(Y^{(i)})^T X^{(i)}(X^{(i)})^T u_i^j}{||(Y^{(i)})^T t_i^{j+1}||||X^{(i)} u_i^j||}$
2. $c_i^{j+1} = \frac{(Y^{(i)})^T X^{(i)}(X^{(i)})^T Y^{(i)} c_i^j}{||(Y^{(i)})^T t_i^{j+1}||||X^{(i)} u_i^j||}$
3. $t_i^{j+1} = \frac{X^{(i)}(X^{(i)})^T Y^{(i)}(Y^{(i)})^T t_i^j}{||(Y_{(i)})^T t_i^{j+1}||||X^{(i)} u_i^j||}$
4. $w_i^{j+1} = \frac{(X^{(i)})^T Y^{(i)}(Y^{(i)})^T X^{(i)} w_i^j}{||(Y^{(i)})^T t_i^{j+1}||||X^{(i)} u_i^j||}$

Dimostrazione. Iniziamo a dimostrare la prima relazione:

$$u_i^{j+1} = Y^{(i)} c_i^{j+1} = \frac{Y^{(i)}(Y^{(i)})^T t_i^{j+1}}{||(Y^{(i)})^T t_i^{j+1}||} = \frac{Y^{(i)}(Y^{(i)})^T X^{(i)} w_i^{j+1}}{||(Y^{(i)})^T t_i^{j+1}||} = \frac{Y^{(i)}(Y^{(i)})^T X^{(i)}(X^{(i)})^T u_i^j}{||(Y^{(i)})^T t_i^{j+1}||||X^{(i)} u_i^j||}$$

Con analoghi calcoli si dimostrano le altre relazioni. □

Queste equazioni mostrano che l'algoritmo si comporta in maniera simile al metodo delle potenze, che riportiamo per completezza.

Lemma 2.15. *Metodo delle potenze Sia $A \in \mathbb{C}^{n\times n}$ una matrice diagonalizzabile con autovalori $\lambda_1, \ldots, \lambda_n$ tali che $|\lambda_1| > |\lambda_2| \geq \ldots |\lambda_n|$. Sia $x_0 \in \mathbb{C}^n$ e $x_n = Ax_{n-1}$. Allora $\lim_{n\to\infty} x_n = x$, dove indichiamo con x l'autovettore relativo a λ_1.*

Alla convergenza possiamo scrivere, per ogni $i = 1, \ldots, A$:

$$Y^{(i)}(Y^{(i)})^T X^{(i)}(X^{(i)})^T u_i = \lambda_i u_i$$
$$(Y^{(i)})^T X^{(i)}(X^{(i)})^T Y^{(i)} c_i = \lambda_i c_i$$
$$X^{(i)}(X^{(i)})^T Y^{(i)}(Y^{(i)})^T t_i = \lambda_i t_i$$
$$(X^{(i)})^T Y^{(i)}(Y^{(i)})^T X^{(i)} w_i = \lambda_i w_i$$

Il metodo delle potenze ci dice che λ_i è l'autovalore di modulo massimo e i vettori u, c, t e w sono gli autovettori delle rispettive matrici corrispondenti al massimo autovalore. Osserviamo che le matrici scritte sopra hanno tutte lo stesso spettro, vediamo la dimostrazione di questo lemma.

Lemma 2.16. *Lo spettro del prodotto di matrici è invariante per permutazioni cicliche.*

Dimostrazione. Dimostriamo il lemma per le permutazioni delle nostre matrici, togliendo l'indice i per semplicità.
Supponiamo che v sia autovettore della matrice YY^TXX^T, quindi

$$YY^TXX^Tv = \lambda v.$$

Vediamo allora che

$$Y^T X X^T Y (Y^T X X^T v) = \lambda Y^T X X^T v,$$

e dunque λ è anche un autovalore di questa matrice. Per la matrice $XX^T YY^T$ abbiamo che

$$XX^T YY^T (XX^T v) = \lambda XX^T v$$

e dunque λ appartiene anche al suo spettro. Infine

$$X^T YY^T X (X^T v) = \lambda X^T v.$$

Abbiamo dunque dimostrato che lo spettro di $YY^T XX^T$ è contenuto nello spettro delle altre tre matrici, ma similmente si possono dimostrare le altre inclusioni. Dunque lo spettro delle quattro matrici coincide, e quindi, in particolare, anche il massimo autovalore sarà il medesimo.

□

In particolare w_i è un autovettore della matrice di covarianza di $(Y^{(i)})^T X^{(i)}$ e analogamente gli altri vettori.

Fatte queste considerazioni, si capisce chiaramente che l'algoritmo PLS può anche essere visto in questo modo alternativo, che forse a prima vista risulta più chiaro. Vediamo lo pseudo codice:

$X^{(1)} = X$, $Y^{(1)} = Y$

for $(i = 1, \ldots, A)\{w_i =$ autovettore relativo al massimo autovalore della matrice

$(X^{(i)})^T Y^{(i)} (Y^{(i)})^T X^{(i)}$

$t_i = X^{(i)} w_i$

$c_i =$ autovettore relativo al massimo autovalore della matrice

$(Y^{(i)})^T X^{(i)} (X^{(i)})^T Y^{(i)}$

$u_i = Y^{(i)} c_i$

$p_i = \frac{(X^{(i)})^T t_i}{||t_i||^2}$

$q_i = \frac{(Y^{(i)})^T u_i}{||u_i||^2}$

$b_i = \frac{u_i^T t_i}{||t_i||^2}$

$X^{(i+1)} = X^{(i)} - t_i p_i^T$, $Y^{(i+1)} = Y^{(i)} - b_i t_i c_i^T\}$

Ovviamente la prima versione dell'algoritmo non fornisce esattamente gli autovettori delle rispettive matrici, ma una loro approssimazione, tanto buona quanto più piccola è la

soglia ϵ. Nella pratica si tende a usare la prima versione dell'algoritmo, in quanto spesso si hanno problemi computazionali nel calcolo degli autovettori poichè le matrici hanno dimensioni molto grandi.

Andiamo ad osservare una prima differenza con l'algoritmo teorico da noi introdotto precedentemente. Mentre l'algoritmo teorico prende come vettori w_i gli autovettori della matrice X^TYY^TX e come c_i gli autovettori della matrice Y^TXX^TY, l'algoritmo di PLS prende come w_i l'autovettore relativo al massimo autovalore della matrice $(X^{(i)})^TY^{(i)}(Y^{(i)})^TX^{(i)}$ e come c_i l'autovettore relativo al massimo autovalore della matrice $(Y^{(i)})^TX^{(i)}(X^{(i)})^TY^{(i)}$. Osserviamo che i due algoritmi risutano identici se ci fermiamo al primo passo, ossia se $A = 1$. Cerchiamo di capire intuitivamente perchè Wold abbia preferito questa formulazione dell'algoritmo. Non ci sono vere e proprie dimostrazioni del fatto che la versione di Wold sia migliore della nostra formulazione, ma possiamo vedere dei conti approssimati.

Osservazione 2.17. Nel seguito, per non avere ambiguità di notazioni, indichiamo con $\sigma_i(A)$ l'i-esimo valore singolare della matrice A.
Abbiamo visto precedentemente che il primo valore singolare di una matrice soddisfa

$$\sigma_1(A) = \max_{\{u\in\mathbb{R}^n, v\in\mathbb{R}^m |\ ||u||=||v||=1\}} |u^TAv|.$$

Dunque, per quanto sottolineato più volte,

$$\sigma_1((X^{(i)})^TY^{(i)}) = \max_{\{u\in\mathbb{R}^n, v\in\mathbb{R}^m |\ ||u||=||v||=1\}} |u^T(X^{(i)})^TY^{(i)}v|.$$

Vogliamo adesso capire come mai invece di prendere A autovettori della matrice X^TYY^TX e A autovettori della matrice Y^TXX^TY, come abbiamo proposto nell'algoritmo teorico, vengono selezionati gli autovettori relativi al massimo autovalore delle matrici $(X^{(i)})^TY^{(i)}(Y^{(i)})^TX^{(i)}$. Prima di enunciare una proprietà che fa capire il perchè della scelta di Wold, abbiamo bisogno di un lemma.

Lemma 2.18. *Sia $A \in \mathbb{R}^{n\times m}$. Vale*

$$\sigma_i(A - B) \geq \sigma_{i+k}(A)\ \forall\ B \in \mathbb{R}^{n\times m},\ rk(B) = k.$$

Teorema 2.19. *Vale che $\sigma_1((X^{i+1})^TY^{(i+1)}) \geq \sigma_2((X^i)^TY^{(i)})$.*

Dimostrazione. Ricordiamo le espressioni delle matrici $X^{(i+1)}$ ed $Y^{(i+1)}$ rispetto alle matrici relative al passo precedente:

$$X^{(i+1)} = X^{(i)} - t_ip_i^T,$$

$$Y^{(i+1)} = Y^{(i)} - t_ib_ic_i^T.$$

Dunque

$$\sigma_1((X^{i+1})^TY^{(i+1)}) = \sigma_1((X^{(i)} - t_ip_i^T)^T(Y^{(i)} - t_ib_ic_i^T)) =$$

$$= \sigma_1((X^{(i)})^T Y^{(i)} - (p_i t_i^T Y^{(i)} - p_i t_i^T t_i b_i c_i^T + (X^i)^T t_i b_i c_i^T)) =$$
$$= \sigma_1((X^{(i)})^T Y^{(i)} - (p_i t_i^T [Y^{(i)} - t_i b_i c_i^T + u_i c_i^T])),$$

dove, nell'ultimo passaggio abbiamo sostituito le espressioni di p_i ed b_i, nell'ultima matrice.
Possiamo applicare adesso il lemma precedente, e visto che la matrice che sottraiamo è una matrice di rango 1, otteniamo

$$\sigma_1((X^{i+1})^T Y^{(i+1)}) \geq \sigma_2((X^i)^T Y^{(i)}),$$

come volevasi dimostrare.

□

Dunque questa proprietà ci fa capire intuitivamente come mai sia meglio andare a selezionare l'autovettore relativo al più grande autovalore della matrice del passo $i+1$ rispetto all'autovettore relativo al secondo autovalore più grande della matrice del passo i.

Osservazione 2.20. Un'altra osservazione basata sulle proprietà degli autovalori è la seguente. Supponiamo sempre che la matrice Y non venga modificata durante le iterazioni. Il più grande valore singolare pùo anche essere interpretato come:

$$\sigma_1^2((X^{(i)})^T Y) = \max_{||f||=1} f^T (X^{(i)})^T Y Y^T X^{(i)} f$$

Dunque segue la seguente proprietà:

$$(\sigma_1((X^{(i+1)})^T Y))^2 \leq (\sigma_i((X^{(i)})^T Y))^2.$$

Infatti abbiamo che:

$$(\sigma_1((X^{(i+1)})^T Y))^2 = \max_{||f||=1} f^T (X^{(i+1)})^T Y Y^T X^{(i+1)} f =$$

$$= \max_{||f||=1} f^T (Id - p_i w_i^T)(X^{(i)})^T Y Y^T X^{(i)} (Id - w_i p_i^T) f$$

$$\leq \max_{||f||=1} f^T (X^{(i)})^T Y Y^T X^{(i)} f = (\sigma_1((X^{(i)})^T Y))^2.$$

Dalla proprietà appena enunciata segue che ciò che selezioniamo da $(X^{(i)})^T Y$ decresce sempre. Un buon criterio sul quale basarci per capire quante componenti selezionare potrebbe dunque essere di fermarsi quando $\sigma_1(X^{(i)} Y)$ diventa sufficientemente piccolo.

Prima di passare alle dimostrazioni delle proprietà geometriche che soddisfano i vettori creati dall'algoritmo vediamo un parallelismo con PCA che ci può aiutare nella comprensione dell'algortimo.

2.3.1 Legame con PCA

Vediamo adesso un legame con il metodo delle componenti principali, che ci permetterà di capire meglio il metodo PLS.
Nel capitolo precedente abbiamo visto come, data una matrice X, individuare le componenti principali $p_1, \ldots, p_A$. Riportiamo adesso un metodo alternativo fondamentale per la comprensione di PLS.
Abbiamo visto che i vettori $p_1, \ldots, p_A$ sono gli autovettori della matrice X^TX. Quindi in particolare

$$X^TXp_i = \lambda_i p_i.$$

Vogliamo adesso creare un algoritmo simile al ciclo while dell'algortimo di PLS che ci permetta di trovare gli autovettori della matrice X^TX in modo alternativo. Vediamo lo pseudo codice:

$X^{(1)} = X$, $Y^{(1)} = Y$, $t_1^0 = Y_0 e_1$

for $(i = 1, \ldots, A)\{j = 0$, **if**$(i > 1)$ $t_i^0 = t_{i-1}$

while $(||t_i^{j-1} - t_i^{j-2}|| > \epsilon \cup j \leq 1)$**do**$\{$

$$p_i^{j+1} = \frac{(X^{(i)})^T t_i^j}{||(X^{(i)})^T t_i^j||}$$
$$t_i^{j+1} = X^{(i)} p_i^{j+1}$$
$$j = j + 1\}$$

$t_i = t_i^j$, $p_i = p_i^j$, $X^{(i+1)} = X^{(i)} - t_i p_i^T\}$

Vediamo adesso che relazione soddisfano i vettori.

Teorema 2.21. *I vettori creati dall'algoritmo precedente soddisfano le seguenti relazioni per ricorrenza:*

1. $p_i^{j+1} = \frac{(X^{(i)})^T X_i p_i^j}{||(X^{(i)})^T t_i^j||}$
2. $t_i^{j+1} = \frac{X^{(i)}(X^{(i)})^T t_i^j}{||(X^{(i)})^T t_i^j||}$

Dimostrazione. Dimostriamo solo la prima relazione:

$$p_i^{j+1} = \frac{(X^{(i)})^T t_i^j}{||(X^{(i)})^T t_i^j||} = \frac{(X^{(i)})^T X_i p_i^j}{||(X^{(i)})^T t_i^j||}$$

Analogamente si dimostra l'altra identità.

□

Dunque, come visto precedemente per l'algoritmo di PLS abbiamo che alla convergenza vale, per ogni $i = 1, \ldots, A$:

$$(X^{(i)})^T X^{(i)} p_i = \lambda_i p_i$$

$$X^{(i)}(X^{(i)})^T t_i = \lambda_i t_i$$

dove λ_i è l'autovettore di massimo modulo delle relative matrici.
Vogliamo adesso mostrare di più, ossia che i vettori p_i per $i = 1, \ldots, A$, sono tutti autovettori della matrice X. Per dimostrare ciò abbiamo bisogno di alcuni lemmi.

Lemma 2.22. *I vettori p_i sono ortogonali:*

$$< p_i, p_j >= p_i^T p_j = 0 \ \textit{se} \ \ i \neq j.$$

Dimostrazione. Sia $i < j$. Si ha

$$< p_i, p_j >= p_i^T p_j = p_i^T (X^{(j)})^T p_i,$$

dunque per dimostrare la tesi basta vedere che $X^{(j)} p_i = 0$ se $i < j$.
Vediamo che relazione soddisfa la matrice X_j rispetto alle matrici precedenti. Si ha che:

$$X^{(j)} = X^{(j-1)} - t_{j-1} p_{j-1}^T = X^{(j-1)} - \frac{t_{j-1} t_{j-1}^T X^{(j-1)}}{||(X^{(j-1)})^T t_{j-1}||} =$$

$$= \left[I - \frac{t_{j-1} t_{j-1}^T}{||(X^{(j-1)})^T t_{j-1}||} \right] X^{(j-1)} = Z \left[X^{(i)} - \frac{t_i t_i^T X^{(i)}}{||(X^{(i)})^T t_i||} \right],$$

dove Z è un'opportuna matrice. Dunque per dimostrare la tesi basta vedere che

$$\left[X^{(i)} - \frac{t_i t_i^T X^{(i)}}{||(X^{(i)})^T t_i||} \right] p_i = 0,$$

ma questo è ovvio perchè:

$$\left[X^{(i)} - \frac{t_i t_i^T X^{(i)}}{||(X^{(i)})^T t_i||} \right] p_i = X^{(i)} p_i - \frac{t_i t_i^T X^{(i)}}{||(X^{(i)})^T t_i||} p_i = t_i - t_i = 0.$$

Questo conclude la dimostrazione. □

Lemma 2.23. *I vettori t_i sono ortogonali:*

$$< t_i, t_j >= t_i^T t_j = 0 \ \textit{se} \ \ i \neq j.$$

Dimostrazione. Sia $i < j$. Si ha che

$$t_i^T t_j = t_i^T X^{(j)} p_j$$

dunque per dimostrare la tesi basta vedere che $(X^{(j)})^T t_i = 0$ se $j > i$. Usando la relazione precedente della matrice $X^{(j)}$ otteniamo che

$$X^{(j)} t_i = Z \left[X^{(i)} - \frac{t_i t_i^T X^{(i)}}{||(X^{(i)})^T t_i||} \right] t_i = X^{(i)} t_i - \frac{t_i t_i^T X^{(i)} t_i}{||(X^{(i)})^T t_i||} = 0$$

Questo dimostra la mutua ortogonalità dei vettori t_i. □

Adesso che abbiamo dimostrato questi lemma, possiamo andare a dimostrare ciò che abbiamo annunciato prima, ossia:

Teorema 2.24. *I vettori p_i sono tutti autovettori della matrice X^TX:*

$$X^TXp_i = \lambda_i p_i \text{ per } i = 1, \dots, A.$$

Dimostrazione. Per $i = 1$ la tesi è vera per costruzione. Sia $i > 1$. Abbiamo che

$$X = \sum_{k=1}^{i-1} t_k p_k^T + X^{(i)},$$

e dunque ne segue che

$$X^TXp_i = \left[\sum_{k=1}^{i-1} t_k p_k^T + X^{(i)}\right]^T \left[\sum_{k=1}^{i-1} t_k p_k^T + X^{(i)}\right] p_i = (X^i)^T X^{(i)} p_i = \lambda_i p_i$$

per l'ortogonalità mostrata nei due lemmi precedenti. □

Dunque abbiamo dimostrato una proprietà fondamentale dell'algoritmo precedente, ossia che genera autovettori della matrice X^TX. Dunque questo algoritmo può essere usato alternativamente per la ricerca degli autovettori per il metodo delle componenti principali.
Vediamo adesso i collegamenti con il metodo PLS. Una prima idea che si potrebbe avere per la ricerca delle coppie dei vettori t_i e w_i e u_i e c_i può essere quella di applicare l'algoritmo precedente di PCA separatamente alle matrici X e Y. Come già sottolineato questo non porta alla soluzione del nostro scopo, in quanto non vogliamo spiegare separatamente gli spazi X e Y, ma anche catturare la loro struttura comune. Dunque l'idea sfruttata dall'algoritmo PLS è di mescolare i due algoritmi. Vediamo in dettaglio, omettendo gli indici per non appesantire la trattazione.
Introduciamo i due algoritmi che useremmo se volessimo applicare PCA alle matrici X e Y separatamente.

$t = Xe_1$
ciclo $\{w = \frac{X^Tt}{||X^Tt||}$
$t = Xw$
if(t converge) **end ciclo** $\}$

$u = Ye_1$
ciclo$\{c = \frac{Y^Tu}{||Y^Tu||}$
$u = Yc$
if(u converge) **end ciclo** $\}$

Come già sottolineato più volte vogliamo decomporre le matrici X e Y cercando di catturare la struttura comune. Un modo intuitivo di fare ciò è di scambiare t ed u prima di aggiornare w e c, e dopo continuare il ciclo. Il risultato è l'algoritmo seguente, che corrisponde al ciclo while nel primo algoritmo citato per PLS. Omettiamo anche qui, per maggiore leggibilità, gli indici.

$u = Ye_1$

ciclo$\{w = \frac{X^T u}{||X^T u||}$

$t = Xw$

$c = \frac{Y^T t}{||Y^T t||}$

$u = Yc$

if(t converge) **end ciclo** $\}$

Il ciclo precedente ci fornisce i primi vettori t_1, p_1, q_1 e u_1. Per trovare gli altri vettori, come già visto nel primo algoritmo, dobbiamo andare a modificare le matrici X e Y e ripetere nuovamente il ciclo appena enunciato.
Questo mostra che PLS non è nient'altro che una modifica di PCA.

Osserviamo adesso che la proprietà di massimizzazione della covarianza non sarebbe soddisfatta se come componenti avessimo scelto la prima componente principale nello spazio X e la prima componente principale nello spazio Y. In seguito abbiamo chiamato z_1 la prima componente principale nello spazio X, e k_1 la prima componente principale nello spazio Y. Osserviamo che usando questi due vettori la varianza non è massimizzata come nel caso di p_1 e q_1.

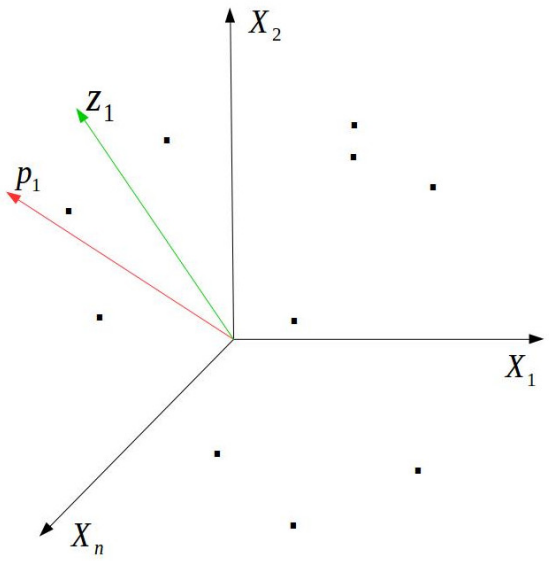

Figura 2.4: Spazio X

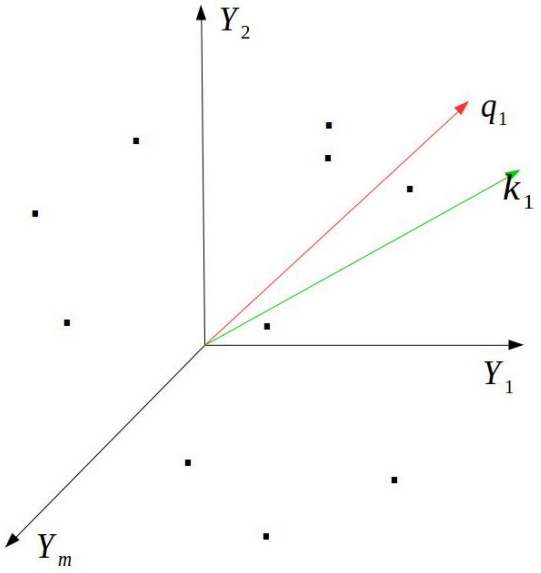

Figura 2.5: Spazio Y

Nell'immagine seguente 2.6 si può vedere come ci sia maggiore allineamento quando passiamo dai vettori z_1 e k_1 ai vettori p_1 e q_1. Notiamo che per costruzione dell'algoritmo il coefficiente b_1 è l'inclinazione della retta di regressione tra u_1 e t_1, ossia a meno di errori si ha che

$$u_1 = b_1 t_1 + \epsilon.$$

I ragionamenti fatti fino ad adesso, per i vettori riguardanti la prima iterazione, si possono ripetere anche per le altre iterazioni.

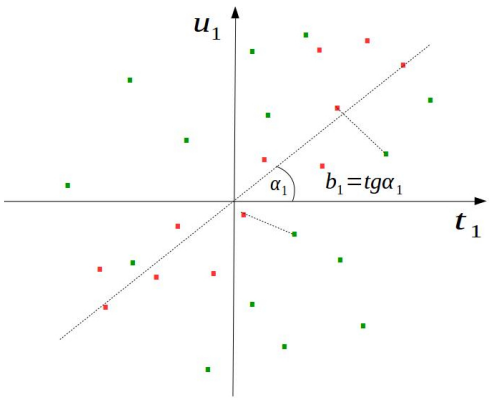

Figura 2.6: t_1 vs u_1

2.3.2 L'algoritmo nel caso di m=1

E' molto istruttivo guardare come diventa l'algoritmo nel caso in cui Y sia unidimensionale. In tal caso, il codice dell'algoritmo si semplifica molto, in quanto il ciclo while viene eliminato. Vediamo lo pseudo codice.

$X^{(1)} = X$, $Y^{(1)} = Y$

for$(i = 1, \dots, A)\{w_i = \frac{(X^{(i)})^T Y^{(i)}}{||(X^{(i)})^T Y^{(i)}||}$

$t_i = X^{(i)} w_i$

$p_i = \frac{(X^{(i)})^T t_i}{||t_i||^2}$

$b_i = \frac{(Y^{(i)})^T t_i}{||(Y^{(i)})^T t_i||}$

$X_{i+1} = X_i - t_i p_i^T$, $Y_{i+1} = Y_i - b_i t_i\}$

Andiamo ad analizzare i passi di questo algortimo. Il vettore t_i è una combinazione lineare delle colonne della matrice $X^{(i)}$, $X_1^{(i)}$, $\dots$, $X_n^{(i)}$, per $i = 1, \dots, A$. In particolare, se indichiamo con w_{ij}, $j = 1, \dots, n$ gli elementi del vettore w_i abbiamo che

$$t_i = w_{i1} X_1^{(i)} + \cdots + w_{in} X_n^{(i)}.$$

I pesi w_{ij} al variare di j sono semplicemente il prodotto scalare tra la j-esima colonna della matrice $X^{(i)}$ e il vettore $Y^{(i)}$, normalizzati. Infatti

$$w_{ij} = \frac{< X_j^{(i)}, Y^{(i)} >}{|| < X_j^{(i)}, Y^{(i)} > ||} \text{ per } \ j = 1, \dots, n.$$

Analogamente per gli elementi del vettore c_i vale che

$$c_{ij} = \frac{< Y_j^{(i)}, t_i >}{|| < Y_j^{(i)}, t_i > ||} \text{ per } \ j = 1, \dots, m.$$

Osserviamo inoltre una proprietà dei vettori w_i.

Osservazione 2.25. Vediamo che nel caso in cui $m = 1$ i vettori coinvolti nell'algoritmo possono essere direttamente calcolati dai dati.
Il vettore w_i è l'autovettore della matrice $(X^{(i)})^T Y^{(i)} (Y^{(i)})^T X^{(i)}$ relativo all'autovalore $(Y^{(i)})^T X^{(i)} (X^{(i)})^T Y^{(i)}$. Infatti

$$(X^{(i)})^T Y^{(i)} (Y^{(i)})^T X^{(i)} \left(\frac{(X^{(i)})^T Y^{(i)}}{||(X^{(i)})^T Y_i||} \right) = \big((Y^{(i)})^T X^{(i)} (X^{(i)})^T Y^{(i)} \big) \frac{(X^{(i)})^T Y^{(i)}}{||(X^{(i)})^T Y_i||}.$$

Dunque

$$w_i = \frac{(X^{(i)})^T Y^{(i)}}{||(X^{(i)})^T Y^{(i)}||}.$$

I vettori t_i e c_i sono rispettivamente

$$t_i = \frac{X^{(i)}(X^{(i)})^T Y^{(i)}}{||(X^{(i)})^T Y_i||},$$

$$c_i = \frac{(Y^{(i)})^T X^{(i)}(X^{(i)})^T Y^{(i)}}{||(Y^{(i)})^T X^{(i)}(X^{(i)})^T Y^{(i)}||}.$$

2.4 La geometria dell'algoritmo PLS

Vogliamo adesso studiare le proprietà geometriche che stanno dietro all'algoritmo riportato sopra. Useremo la notazione:

$$W = (w_1, w_2, \ldots, w_A).$$

Dunque $W \in \mathbb{R}^{k \times A}$ che ha per colonne i vettori w_i, e similmente faremo per gli altri insiemi di vettori. Per capire le proprietà di base andiamo a analizzare due passi successivi dell'algoritmo, ossia come viene calcolata la matrice residua $X^{(i)}$ a partire dalle matrici precedenti.

$$X^{(i)} = X^{(i-1)} - t_{i-1}p_{i-1}^T = X^{(i-1)} - \frac{t_{i-1}t_{i-1}^T X^{(i-1)}}{||t_{i-1}||^2} =$$

$$= \left[I - \frac{t_{i-1}t_{i-1}^T}{||t_{i-1}||^2}\right] X^{(i-1)} = \left[I - \frac{t_{i-1}t_{i-1}^T}{||t_{i-1}||^2}\right] \left[X^{(i-2)} - \frac{t_{i-2}t_{i-2}^T X^{(i-2)}}{||t_{i-2}||^2}\right]$$

2.4.1 Ortogonalità dei vettori w

Teorema 2.26. *I vettori w sono ortogonali:*

$$< w_i, w_j >= 0 \ se \ \ i \neq j.$$

Dimostrazione. Osserviamo inanzitutto che, essendo w_j un autovettore, abbiamo: $(X^{(j)})^T Y^{(j)}(Y^{(j)})^T X^{(}$ $a_j w_j$ e dunque $w_j^T = \frac{w_j^T (X^{(j)})^T Y^{(j)}(Y^{(j)})^T X_j}{a_j}$.
Arriviamo quindi a scrivere $< w_j, w_i >= w_j^T w_i = \frac{w_j^T (X^{(j)})^T Y^{(j)}(Y^{(j)})^T X^{(j)}}{a_j} w_i$.
Per dimostrare la tesi è dunque sufficiente vedere che $X^{(j)} w_i = 0$. Andiamo a vedere che relazione soddisfa la matrice $X^{(i)}$. Supponiamo che $i < j$. Vale allora la seguente relazione:

$$X^{(i)} = X^{(i-1)} - t_{i-1}p_{i-1}^T = X^{(i-1)} - \frac{t_{i-1}t_{i-1}^T X^{(i-1)}}{||t_{i-1}||^2} =$$

$$= \left[I - \frac{t_{i-1}t_{i-1}^T}{||t_{i-1}||^2}\right] X^{(i-1)} = \left[I - \frac{t_{i-1}t_{i-1}^T}{||t_{i-1}||^2}\right] \left[X^{(i-2)} - \frac{t_{i-2}t_{i-2}^T X^{(i-2)}}{||t_{i-2}||^2}\right]$$

Dunque, procedendo in tal modo, otteniamo:

$$X^{(j)} = Z\left[X^{(i)} - \frac{t_i t_i^T}{||t_i||^2} X^{(i)}\right]$$

dove $Z := \prod_{k=i+1}^{j-1}\left[I - \frac{t_k t_k^T}{||t_k||^2}\right]$.

Allora abbiamo che $X^{(j)}w_i = Z\left[X^{(i)} - \frac{t_i t_i^T}{||t_i||^2} X^{(i)}\right] w_i = Z\big[t_i - t_i\big] = 0$, e questo conclude la dimostrazione. □

2.4.2 Ortogonalità dei vettori t

Teorema 2.27. *I vettori t sono ortogonali:*

$$< t_i, t_j >= 0 \ se \ \ i \neq j.$$

Dimostrazione. Supponiamo che $i < j$. Abbiamo che $< t_i, t_j >= t_i^T t_j = t_i^T X^{(j)} w_j$. Quindi per dimostrare la tesi è sufficiente mostrare che $t_i^T X^{(j)} = 0$. Andiamo adesso a cercare una relazione soddisfatta dalla matrice $X^{(j)}$. Similmente a quanto fatto nella dimostrazione precedente abbiamo:

$$X^{(j)} = X^{(j-1)} - \frac{X^{(j-1)} w_{j-1} t_{j-1}^T X^{(j-1)}}{||t_{j-1}||^2} = X^{(j-1)}\left[I - \frac{w_{j-1} t_{j-1}^T X^{(j-1)}}{||t_{j-1}||^2}\right] = X^{(i+1)} Z,$$

dove $Z := \prod_{k=i+2}^{j-1}\left[I - \frac{w_k t_k^T X^{(k)}}{||t_k||^2}\right]$.
Usando la relazione dimostrata nel teorema precedente possiamo anche scrivere:

$$X^{(i+1)} Z = \left[X^{(i)} - \frac{t_i t_i^T X^{(i)}}{||t_i||^2}\right] Z$$

Abbiamo allora che $t_i^T X^{(j)} = t_i^T\left[X^{(i)} - \frac{t_i t_i^T X^{(i)}}{||t_i||^2}\right] = 0$, e dunque per quanto osservato prima abbiamo la tesi. □

Osservazione 2.28. I vettori t_i sono dunque una base ortogonale nello spazio generato dalle colonne di X. Possiamo scrivere la matrice X come:

$$X = \sum_{i=1}^{A} t_i p_i^T + X^{(A+1)}.$$

2.4.3 Ortogonalità dei vettori w e p

Andiamo a vedere adesso un'altra proprietà dei vettori generati dall'algoritmo.

Teorema 2.29. *I vettori w_i sono ortogonali ai vettori p_j se $i < j$:*

$$< w_i, p_j >= 0 \text{ se } \ i < j.$$

Dimostrazione. Abbiamo che $< w_i, p_j >= w_i^T p_j = w_i^T \frac{(X^{(j)})^T t_j}{||t_j||^2}$. Abbiamo già osservato nella dimostrazione del teorema precedente che $t_i^T X^{(j)} = 0$, se $i < j$ e dunque anche $\left(t_i^T X^{(j)}\right)^T = (X^{(j)})^T t_i = 0$ e da questo segue la tesi.

□

Questo è tutto quello che possiamo dire riguardo all'ortogonalità dei vettori. In generale abbiamo che

$$< p_i, w_j >\neq 0 \text{ se } \ i < j.$$

Queste proprietà seguono da come le matrici residue sono costruite a partire dalle precedenti, e non dipendono da come un nuovo vettore t è costruito. Dopo aver determinato un vettore t, un nuovo sottospazio ortogonale a tale vettore viene costruito.
Non ci sono speciali proprietà di ortogonalità negli insiemi dei vettori $\{u_i\}$,$\{c_i\}$ e $\{q_i\}$.
Vediamo adesso di chiarire quanto appena dimostrato, attraverso delle rappresentazioni grafiche.

2.4.4 Rappresentazioni

In questa prima figura abbiamo rappresentato lo spazio delle n variabili, spazio nel quale vivono le k osservazioni. I punti rappresentano infatti le k osservazioni, che, una volta individuato il nuovo asse p_1, vengono proiettati su di esso, ottendendo cosi k valori. In particolare, la proiezione della j-esima osservazione sull'asse p_1 fornisce l'elemento j-esimo del vettore t_1. Infatti t_1 è un vettore di $\mathbb{R}^k$, perchè contiene le proiezioni dei k elementi sul nuovo asse p_1.

Riportiamo, per completezza, anche un altro modo di vedere le cose. Infatti potremmo vedere anche lo spazio generato dalle osservazioni, ossia quello nel quale ogni asse è un'osservazione. In tale spazio vivono dunque le n variabili. Il nuovo asse trovato dalla prima iterazione dall'algoritmo in questo spazio è il vettore t_1. Se proiettiamo sul vettore t_1 la i-esima variabile otteniamo l'i-esimo elemento del vettore p_1. Infatto p_1 è un vettore di $\mathbb{R}^n$.

Analogamente, possiamo fare altrettanti ragionamenti per lo spazio delle Y. Iniziamo a vedere lo spazio generato dalle variabili. Anche qui, i punti rappresentano i vari individui, che, una volta individuato l'asse q_1 vengono proiettati su di esso.
In particolare proiettando il j-esimo individuo su questo nuovo vettore q_1 otteniamo la j-esima coordinata del vettore u_1.

Il modo alternativo di vedere le cose è il seguente: mettiamoci nello spazio generato dalle osservazioni, ossia quello nel quale ogni asse è un'osservazione. In tale spazio vivono dunque le m variabili Y. Il nuovo asse trovato dalla prima iterazione dall'algoritmo in questo spazio è il vettore u_1. Se proiettiamo sul vettore u_1 la i-esima variabile otteniamo l'i-esimo elemento del vettore q_1. Infatto q_1 è un vettore di $\mathbb{R}^m$.

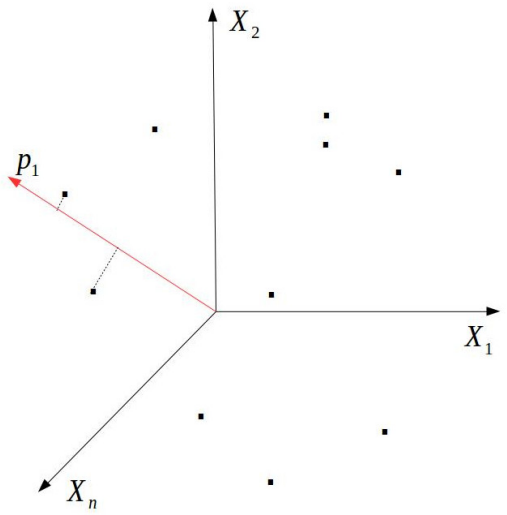

Figura 2.7: Vettore p_1 nello spazio delle variabili

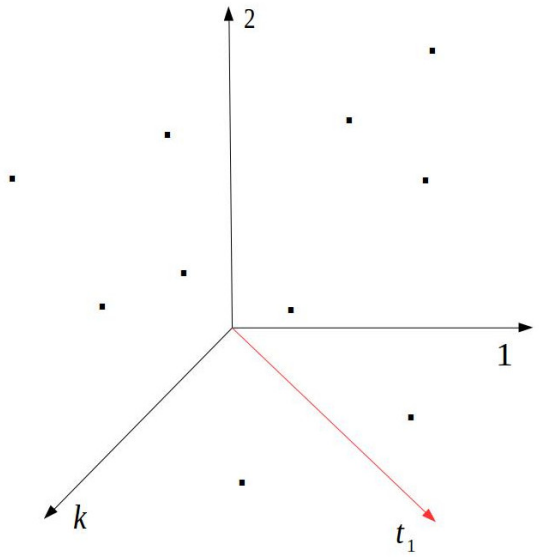

Figura 2.8: Vettore t_1 nello spazio degli individui

Per semplicità abbiamo spiegato in dettaglio solo la prima iterazione, ma quanto

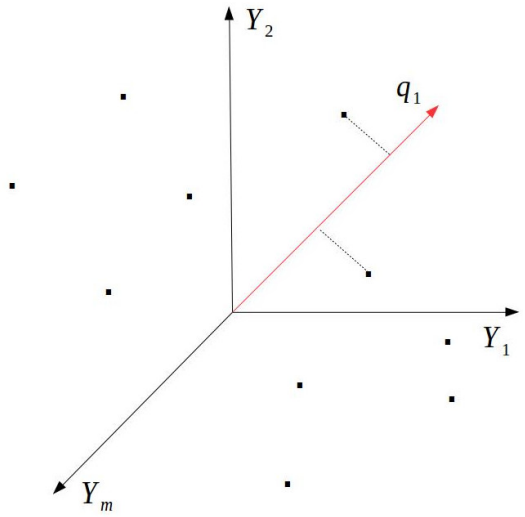

Figura 2.9: Vettore q_1 nello spazio delle variabili

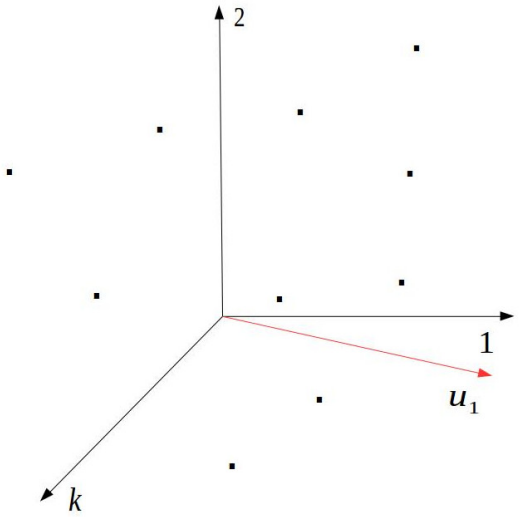

Figura 2.10: Vettore u_1 nello spazio degli individui

detto vale anche per le iterazioni successive.

2.5 PLS come regressione su componenti ortogonali

Nella regressione lineare le variabili sono selezionate sulla base della matrice X^TX. Generalmente prima si determina la matrice di trasformazione O e si calcolano le componenti:

$$F = XO$$

e dopo si calcola la regressione di Y su F. Nella regressione sulle componenti principali O è la matrice che ha come colonne gli autovettori di X^TX.
In molte situazioni è utile pesare la matrice di covarianza, ad esempio sostituendo X^TX con X^TVX dove V è una matrice definita positiva.
Un modo per introdurre questi pesi è guardare la dimesionalità della matrice Y. La matrice $V = YY^T$ può essere vista come la dimensione dei dati nella matrice Y. E' ragionevole dunque usare come $V = YY^T$ e considerare:

$$X^TVX = X^TYY^TX$$

Con questo cambiamento pesiamo secondo la taglia dei dati nella matrice Y: i valori in Y vicini a 0 corrispondono a piccoli pesi, mentre grandi valori corrispondono a grandi pesi. Nel caso estremo, se alcune righe della matrice Y sono 0, le corrispondenti righe di X non vengono usate nell'analisi. Questo metodo di pesare è interessante. E' ragionevole, per esempio, dare un piccolo peso a dati che sono a livello di rumore: infatti non vogliamo predire il rumore. Invece, vogliamo dare grande peso ai dati dove si concentra il segnale. Facendo la regressione, si può seguire la strategia del metodo della regressione sulle componenti principali: calcoliamo dapprima gli autovettori e autovalori della matrice di covarianza pesata:

$$X^TYY^TX = ODO^T$$

e dopo si calcola la matrice delle componenti nel seguente modo:

$$F = XO$$

e si applica la regressione di Y su F. Questo è in sostanza quello che fa l'algoritmo PLS, apparte il fatto che sceglie solo una componente alla volta. Andiamo a vedere più nel dettaglio.
Consideriamo l'algoritmo PLS al passo i-esimo. L'i-esimo passo consiste nel selezionare w_i come autovettore della matrice $(X^{(i)})^TY^{(i)}(Y^{(i)})^TX^{(i)}$ associato al più grande autovalore, e nel calcolare t_i come $t_i = X^{(i)}w_i$, e nel fare la regressione di $Y^{(i)}$ su t_i.

Teorema 2.30. *La proiezione di $Y^{(i)}$ su t_i è:*

$$t_ic_i^T,$$

la matrice residua è

$$Y^{(i+1)} = Y^{(i)} - t_ic_i^T$$

e la matrice di covarianza residua è

$$(Y^{(i+1)})^TY^{(i+1)} = (Y^{(i)})^TY^{(i)} - c_ic_i^T(t_i^Tt_i).$$

Dimostrazione. Sia w_i l'autovettore associato al più grande autovalore:

$$(X^{(i)})^T Y^{(i)} (Y^{(i)})^T X^{(i)} w_i = \lambda_i w_i \text{ e}$$

$$t_i = X^{(i)} w_i$$

La proiezione della matrice $Y^{(i)}$ su t_i è

$$t_i \left(\frac{t_i^T Y^{(i)}}{t_i^T t_i} \right)$$

che possiamo riscrivere come:

$$t_i \left(\frac{t_i^T Y^{(i)}}{t_i^T t_i} \right) = t_i c_i^T.$$

Questo dimostra le prima parte del teorema. Andiamo adesso a vedere la matrice di covarianza residua.

$$\begin{aligned}
(Y^{(i+1)})^T Y^{(i+1)} &= (Y^{(i)} - t_i c_i^T)^T (Y^{(i)} - t_i c_i^T) = \\
&= (Y^{(i)})^T Y_i - c_i t_i^T Y^{(i)} - (Y^{(i)})^T t_i c_i^T + c_i c_i^T (t_i^T t_i) = \\
&= (Y^{(i)})^T Y^{(i)} - c_i \frac{t_i^T Y^{(i)}}{t_i^T t_i} t_i^T t_i - \frac{(Y^{(i)})^T t_i}{t_i^T t_i} (t_i^T t_i) c_i^T + c_i c_i^T (t_i^T t_i) = \\
&= (Y^{(i)})^T Y^{(i)} - c_i c_i^T (t_i^T t_i) - c_i c_i^T (t_i^T t_i) + c_i^T c_i (t_i^T t_i) = \\
&= (Y^{(i)})^T Y^{(i)} - c_i c_i^T (t_i^T t_i).
\end{aligned}$$

□

La regressione della matrice Y su i primi $i-1$ vettori può essere rappresentata come

$$Y = \sum_{k=1}^{i-1} t_k c_k^T + Y^{(i)}$$

e la matrice di covarianza residua può essere scritta come:

$$(Y^{(i+1)})^T Y^{(i+1)} = Y^T Y - \sum_{k=1}^{i-1} c_k c_k^T (t_k^T t_k).$$

Osservazione 2.31. Osserviamo inoltre che vale

$$(Y^{(i)})^T t_i = Y^T t_i.$$

Infatti, usando la seguente relazione

$$Y = \sum_{k=1}^{i-1} t_k c_k^T + Y^{(i)}$$

otteniamo

$$Y^T t_i = \left[\sum_{k=1}^{i-1} t_k c_k^T + Y^{(i)}\right]^T t_i = (Y^{(i)})^T t_i,$$

per ortogonalità dei vettori t_j. Dunque la proiezione della matrice Y su t_i coincide con la proiezione della matrice $Y^{(i)}$ sullo stesso t_i.

2.6 PLS come modello di previsione

Uno degli scopi più importanti nell'analisi di regressione è quello di ridurre il numero delle variabili indipendenti. Il metodo PLS può essere visto come un buon metodo per fare la regressione, perchè le componenti vengono scelte in modo da descrivere in un qualche senso le variabili dipendenti. Il fatto principale è che il metodo PLS riesce a tenere il numero di variabili basso.
Un modello lineare di regressione può essere scritto come

$$Y = XA + \epsilon,$$

dove A è una matrice che contiene i coefficienti di regressione ed ϵ è la matrice dei residui. Nel seguito useremo la notazione

$$P = (p_1, \ldots, p_k)$$

ossia P è la matrice che ha per colonne i vettori p_i ed analogamente per la matrice T. Ricordiamo che $X \in \mathbb{R}^{k\times n}$ e supponiamo che X abbia rango k. Allora possiamo scrivere:

$$X = \sum_{i=1}^{k} t_i p_i^T = TP^T.$$

Osserviamo che da questa uguaglianza abbiamo necessariamente che la matrice P ha rango k. Consideriamo adesso la matrice $P^T \in \mathbb{R}^{k\times n}$, che ha rango k. Allora essa ammette un'inversa destra: $R := (P^T)^{-1}$. Possiamo allora riscrivere l'uguaglianza precedente come

$$T = XR.$$

Abbiamo visto che $Y = TBC^T$, dove B è la matrice diagonale che contiene gli elementi b_i, allora:

$$Y = TBC^T = XRBC^T$$

Dunque ci siamo ricondotti ad un modello di regressione tra X e Y, e dato un valore di X possiamo facilmente calcolare il rispettivo valore di Y.
Le colonne della matrice R possono essere facilmente calcolate durante le iterazioni dell'algoritmo. Infatti abbiamo:

$$t_1 = Xw_1 \quad \text{ma anche} \quad t_1 = Xr_1$$

e dunque

$$r_1 = w_1.$$

Proseguendo abbiamo

$$t_2 = X(I - w_1 p_1^T)w_2 \quad \text{ma anche} \quad r_2 = w_2 - w_1(p_1^T w_2) = w_2 - r_1(p_1^T w_2)$$

In generale abbiamo dunque

$$r_i = w_i - r_{i-1}(p_{i-1}^T w_i).$$

2.7 La matrice di proiezione

La regressione cerca di proiettare la matrice Y in un sottospazio di X generato dai vettori t. La proiezione sulle prime A componenti può essere scritta come

$$Y^{(A+1)} = P_T Y = \sum_{i=1}^{A} t_i \frac{t_i^T Y}{t_i^T t_i}$$

e dunque possiamo riscrivere la matrice di proiezione come

$$P_T = \sum_{i=1}^{A} t_i \frac{t_i^T}{t_i^T t_i} = \sum_{i=1}^{A} X_i w_i \frac{t_i^T}{t_i^T t_i}$$

Vediamo adesso le proprietà della matrice P_T.

Teorema 2.32. *Siano p_{ij} gli elementi della matrice P_T. La matrice P_T soddisfa le seguenti proprietà:*

1. *$P_T P_T = P_T$, ossia è idempotente;*
2. *$P_T^T = P_T$, ossia è simmetrica;*
3. *$\sum_{i=1}^{k} p_{ij}^2 = p_{jj}$;*
4. *$0 \leq pjj \leq 1$, per $i = 1, \ldots, k$;*

Dimostrazione. Inanzitutto osserviamo che $P_T \in \mathbb{R}^{k \times k}$ poichè t_i sono vettori di $\mathbb{R}^k$. Iniziamo adesso la dimostrazione:

1. Scriviamo per esteso la matrice P_T:

$$P_T = \frac{t_1 t_1^T}{t_1^T t_1} + \frac{t_2 t_2^T}{t_2^T t_2} + \cdots + \frac{t_A t_A^T}{t_A^T t_A},$$

dunque si osserva facilmente che andando a fare il prodotto tra P_T e la matrice stessa i termini

$$\left(\frac{t_i t_i^T}{t_i^T t_i}\right)\left(\frac{t_j t_j^T}{t_j^T t_j}\right) = 0 \text{ se } \; i \neq j.$$

I termini che invece non si annullano sono

$$\left(\frac{t_i t_i^T}{t_i^T t_i}\right)\left(\frac{t_i t_i^T}{t_i^T t_i}\right) = \frac{t_i t_i^T t_i t_i^T}{t_i^T t_i t_i^T t_i} = \frac{t_i t_i^T}{t_i^T t_i} \text{ per } \; i = 1, \ldots, A.$$

Da queste semplici considerazioni segue che

$$P_T P_T = \left(\frac{t_1 t_1^T}{t_1^T t_1} + \frac{t_2 t_2^T}{t_2^T t_2} + \cdots + \frac{t_A t_A^T}{t_A^T t_A}\right)\left(\frac{t_1 t_1^T}{t_1^T t_1} + \frac{t_2 t_2^T}{t_2^T t_2} + \cdots + \frac{t_A t_A^T}{t_A^T t_A}\right) =$$

$$= \frac{t_1 t_1^T}{t_1^T t_1} + \frac{t_2 t_2^T}{t_2^T t_2} + \cdots + \frac{t_A t_A^T}{t_A^T t_A} = P_T,$$

e questo conclude la dimostrazione della prima proprietà.

2. Dobbiamo adesso dimostrare la simmetria della matrice P_T. Scriviamo, come nel punto precedente, la matrice P_T per esteso:

$$P_T = \frac{t_1 t_1^T}{t_1^T t_1} + \frac{t_2 t_2^T}{t_2^T t_2} + \cdots + \frac{t_A t_A^T}{t_A^T t_A},$$

dunque

$$P_T^T = \left(\frac{t_1 t_1^T}{t_1^T t_1} + \frac{t_2 t_2^T}{t_2^T t_2} + \cdots + \frac{t_A t_A^T}{t_A^T t_A}\right)^T =$$

$$= \frac{t_1 t_1^T}{t_1^T t_1} + \frac{t_2 t_2^T}{t_2^T t_2} + \cdots + \frac{t_A t_A^T}{t_A^T t_A} = P_T.$$

3. Andiamo a scrivere la matrice P_T elemento per elemento, sfruttando il fatto che abbiamo appena dimostrato, ossia la simmetria.

$$P_T = \begin{pmatrix} p_{11} & \cdots & p_{1k} \\ \cdots & \cdots & \cdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix} = \begin{pmatrix} p_{11} & \cdots & p_{k1} \\ \cdots & \cdots & \cdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix}$$

Sfruttiamo adesso il fatto che la matrice P_T risulta essere anche idempotente:

$$P_T P_T = \begin{pmatrix} p_{11} & \cdots & p_{k1} \\ \cdots & \cdots & \cdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix}\begin{pmatrix} p_{11} & \cdots & p_{k1} \\ \cdots & \cdots & \cdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix} = \begin{pmatrix} p_{11} & \cdots & p_{k1} \\ \cdots & \cdots & \cdots \\ p_{k1} & \cdots & p_{kk} \end{pmatrix}$$

e dunque facendo il prodotto della riga j-esima della prima matrice, per la colonna j-esima della seconda matrice, per $j = 1, \ldots, k$, otteniamo gli elementi p_{jj} che risultano proprio essere

$$p_{jj} = \sum_{i=1}^{k} p_{ij}^2.$$

4. Vediamo adesso la quarta proprietà. Questa dimostrazione si basa sul risultato appena dimostrato. Infatti abbiamo appena visto che

$$p_{jj} = p_{1j}^2 + \cdots + p_{jj}^2 + \cdots + p_{kj}^2 \geq p_{jj}^2,$$

e dunque $p_{jj} \geq p_{jj}^2$ che vale, se e solo se, $0 \leq p_{jj} \leq 1$, come volevasi dimostrare.

□

Capitolo 3

Applicazioni

In questo capitolo ci occuperemo di un problema reale, che ho analizzato sotto diversi punti di vista, usando gli strumenti introdotti nei capitoli precedenti.

3.1 Il problema

Durante i mesi di Giugno e Luglio 2015 ho svolto uno stage presso l'azienda Geal SpA, situata a Lucca.
L'azienda si occupa della gestione dell'acquedotto del comune di Lucca, ed ha un totale di circa 40000 utenze. L'azienda per fare la bollettazione usa la seguente strategia: emette bollette quattro volte l'anno, con una frequenza trimestale; due di queste bollette vengono calcolate andando a leggere il contatore di ogni utenza, mentre le altre due vengono stimate (bollette d'acconto). Le bollette si alternano quindi, una volta vengono calcolate precisamente, e un'altra stimate. La stima viene effettuata in base al consumo medio giornaliero degli ultimi tre anni di quell'utenza. Chiaramente la bolletta stimata può avere un valore molto superiore al consumo effettivo di quei tre mesi, o anche molto inferiore. Dunque nella bolletta effettivamente calcolata viene sanata questa differenza, restituendo completamente il valore della bolletta d'acconto e addebitando il valore del consumo effettivo di sei mesi trascorsi dall'ultima bolletta effettivamente calcolata.

L'azienda andando a leggere il contatore ogni sei mesi, ovviamente non avrà mai a disposizione il consumo effettivo mensile di ogni utente, ma avrà dei consumi stimati, come andremo a spiegare più dettagliatamente nel paragrafo successivo. Un altro aspetto è che, con i sistemi attuali, l'azienda per conoscere il consumo totale del comune di Lucca non ha a disposizione nessun mezzo se non quello di sommare il consumo di tutti gli utenti del comune. Per l'azienda sarebbe dunque importante avere delle previsioni dei consumi del comune di Lucca mensilmente, ma ciò non può ovviamente essere realizzato andando a leggere 40000 contatori ogni mese, per motivi di tempo e di spese. La migliore soluzione sarebbe senz'altro l'autolettura che ogni cittadino può fare mensilmente leggendo il proprio contatore, e che può poi comunicare in vari modi a Geal. Attualmente però gli utenti che forniscono la propria telelettura sono troppi pochi. Dunque l'azienda ha pensato di

sfruttare la seguente idea: nel 2011 sono stati installati 961 contatori in telelettura, (nelle zone di San Concordio, San Donato e Sant'Anna) un sistema che permette agli addetti di leggere questi contatori con un apparecchio direttamente dal furgone. In realtà in questo gruppo di utenze, gli utenti attivi fino all'anno 2013 sono solamente 701, e quindi noi ci concentriamo su di essi. Per l'azienda si tratta quindi di una spesa minima andare a rilevare anche tutti i mesi i contatori di questo gruppo ristretto di persone. L'idea che mi è stata proposta è stata dunque quella di cercare un modello per prevedere il consumo mensile del comune di Lucca, a partire dalle misurazioni di questo gruppo ristretto di individui, o di una parte di essi.

3.2 I dati

I dati che ho avuto a disposizione sono stati le misurazioni mensili degli anni dal 2011 al 2013 di tutti le 40000 utenze del comune di Lucca. A partire da questi dati ho facilmente ricavato anche il consumo totale del comune di Lucca, per ogni mese, semplicemente sommando il consumo di tutti gli utenti. Poichè le misurazioni vengono effettuate ogni sei mesi i dati non sono misurazioni effettive ma dati stimati: vediamo in che modo vengono stimati.
Durante l'anno vengono effettuate due letture del contatore: non tutti i 40000 utenti vengono letti nello stesso giorno, per motivi di tempo. In particolare le due letture vengono effettuate nei seguenti periodi: la prima lettura dell'anno viene fatta nei mesi da Febbraio a Maggio, e la seconda lettura da fine Agosto a metà Dicembre. Per semplicità supponiamo che la lettura venga fatta il primo Aprile e il primo Ottobre. Nel momento della lettura di Ottobre, si guarda la differenza di metri cubi dall'ultima lettura, in questo caso dall'Aprile precedente. Calcolata la differenza di metri cubi, si va a dividere essa per il numero di giorni intercorsi dall'ultima lettura, ottenendo così un **consumo giornaliero costante** per tutti i giorni tra il primo Aprile e primo Ottobre. Ottenuto questo consumo giornliero, per ottenere il consumo mensile, basta moltiplicarlo per il numero di giorni del mese. Ad esempio per calcolare il consumo di Giugno andrà moltiplicato per 30, per il consumo di Luglio per 31. In questo modo si ottengono i consumi mensili della maggior parte delle utenze.
Fanno eccezione le utenze top, che vengono misurate più frequentemente, e gli utenti che mandano l'autolettura. Per questi ultimi il consumo mensile è calcolato nello stesso modo, ma considerando che le loro letture sono più frequenti.

Anche per le utenze che hanno il sistema di telelettura vengono fatte nello stesso modo, in quando questo sistema è stato installato ma, per ora, mai effettivamente utilizzato. Tra gli utenti in telelettura ci sono anche alcuni utenti top, come vedremo meglio in seguito. Sottolineiamo il fatto che anche **il consumo totale può considerarsi un dato stimato** e non effettivamente calcolato, essendo calcolato come somma dei consumi di tutti gli utenti. Il mio lavoro, dopo le dovute analisi preliminari, è stato dunque cercare di capire quanto buona fosse la previsione basata su questo gruppo ristretto di utenti con il sistema di telelettura, e anche cercare di restringere ulteriormente il numero

di utenti, fino a capire quale fosse il numero minimo di utenti da considerare per una buona previsione.

3.3 Notazioni

Abbiamo raccolto nella matrice X i dati dei 701 utenti in telelettura, in questo modo: ogni colonna rappresenta un'utenza, ed ogni riga un mese da Gennaio 2011 a Dicembre 2013. Dunque $X \in \mathbb{R}^{36\times 701}$. Nel vettore y abbiamo invece messo il totale mensile dei consumi del comune di Lucca, per i mesi da Gennaio 2011 a Dicembre 2013. Dunque y è un vettore di $\mathbb{R}^{36}$. Riportiamo le dimensioni dei dati nella figura 3.1. Come già fatto

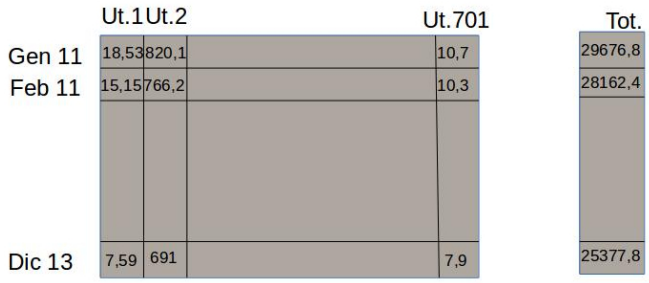

Figura 3.1: Dimesioni dei dati

precentemente, indicheremo con X_i le colonne della matrice X: quindi, in particolare, il vettore X_i sarà un vettore con 36 elementi, e ogni elemento rappresenterà il consumo relativo a quel mese dell'utente i-esimo.
Il modello che stiamo cercando è un modello del tipo

$$y = a_{k_1}X_{k_1} + \cdots + a_{k_p}X_{k_p},$$

dove $1 \leq k_1 < \cdots < k_p \leq 701$, e $a_{k_1}, \ldots, a_{k_p}$ sono scalari scelti in maniera oppurtuna.
Sottolineiamo il fatto che, per quanto controintuitivo, consideriamo gli utenti come variabili e come unità sperimentali i mesi. Ciò è forzato dal fatto che vogliamo predire y che ha il significato di consumo mensile totale.

3.4 Analisi preliminari

Tutte le analisi svolte sono state effettuate con il software R.
Come prima cosa ho riportato in un grafico il consumo totale mensile del comune di Lucca per gli anni dal 2009 al 2013, come si può vedere nel grafico 3.2. Da questa rappresentazione si evince un notevole calo nei mesi invernali, rispetto ai mesi estivi, cosa ragionevole se si pensa al consumo di una normale abitazione. Inoltre si nota anche un notevole calo nell'anno 2013.
Un'ulteriore analisi che ho svolto è stata quella di dividere gli utenti per il loro tipo di utilizzo. Le utenze sono infatti divise tra tipo di utilizzo domestico, tipo di utilizzo

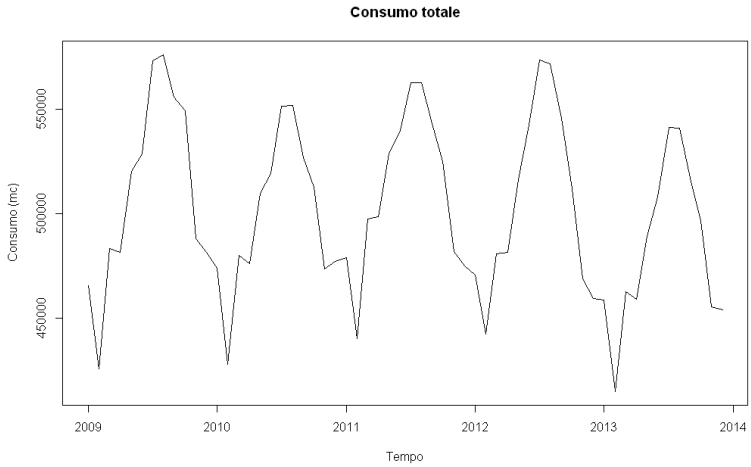

Figura 3.2: Consumo mensile del comune di Lucca

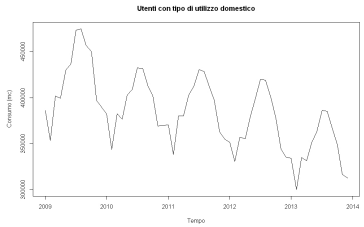

Figura 3.3: Consumo mensile degli utenti con utilizzo domestico

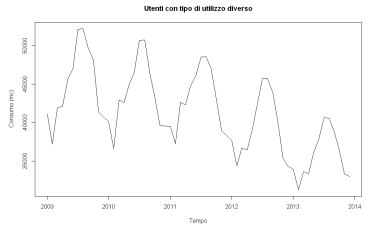

Figura 3.4: Consumo mensile degli utenti con utilizzo diverso

diverso (come ad esempio l'ospedale, la prigione, gli studi, le fabbriche,..) e altri tipi di utilizzi. Ho deciso di rappresentare il consumo mensile degli utenti con tipo di utilizzo domestico e di quelli con tipo di utilizzo diverso nelle figure 3.3 e 3.4.

Gli utenti con tipo di utilizzo domestico sono circa 32000 e quelli con tipo di utilizzo diverso sono circa 5000. Il rimanente sono utenti con altri tipi di utilizzo. Come possiamo vedere dal grafico di figura 3.6 in media il consumo risulta maggiore per gli utenti di tipo di utilizzo domestico: ciò può stupire, ma in realtà risulta chiaro se si pensa che tra gli utenti con tipo di utilizzo diverso ci sono anche piccoli studi che consumano molto poco.Nel grafico 3.6 si nota in entrambi gli storici un trend negativo, che invece risulta meno accentuato nel grafico del consumo totale.

Analizziamo adesso il profilo medio degli utenti con la telelettura e di tutti gli altri, come mostrato in figura 3.6.

Come si può notare il consumo medio degli utenti con il contatore in telelettura risulta notevolmente superiore rispetto al consumo medio degli altri utenti. Indagando ho scoperto che questo fenomeno è dovuto al fatto che tra gli utenti in telelettura sono presenti degli utenti con altissimo consumo, come l'ospedale di Lucca. Essendo così pochi gli utenti in telelettura anche solo pochi individui con un consumo veramente alto

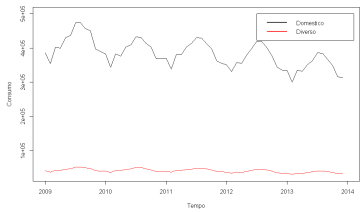

Figura 3.5: Consumo mensile degli utenti con utilizzo domestico e diverso

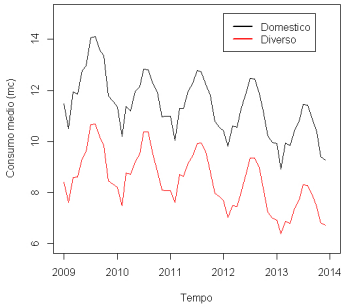

Figura 3.6: Consumo mensile medio degli utenti con utilizzo domestico e diverso

influenzano la media. Nel grafico 3.8 si può notare quanto sia alto il consumo di alcuni utenti con sistema di telelettura: in questo grafico troviamo sull'asse x il numero di utente e sull'asse y il consumo mensile medio; inoltre gli utenti disegnati in nero sono gli utenti con il sistema di telelettura, mentre quelli in rosso sono gli altri utenti. L'asse y ha scala logaritmica. L'utente con consumo maggiore è l'ospedale di Lucca.

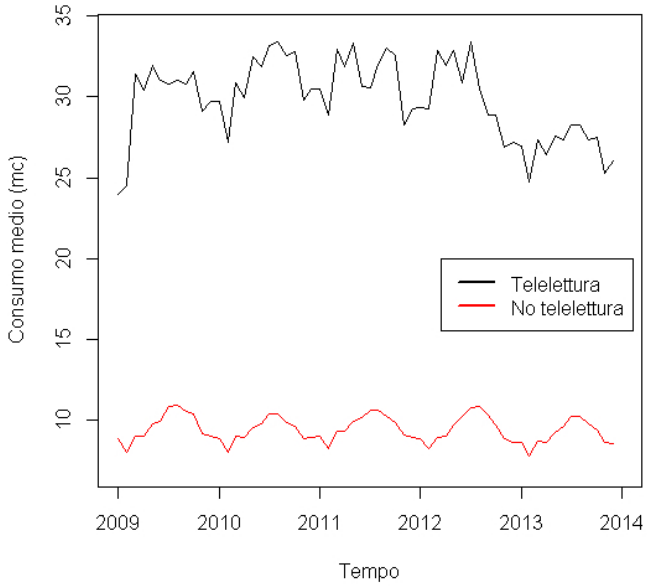

Figura 3.7: Consumo mensile medio degli utenti con telelettura e non

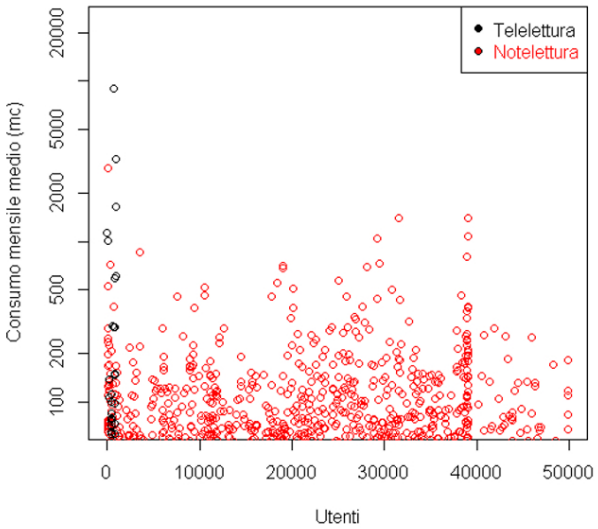

Figura 3.8: Consumo mensile medio degli utenti con telelettura e non

3.5 PCA

Dopo queste prime analisi preliminari, proviamo ad effettuare l'analisi delle componenti principali.

3.5.1 PCA nello spazio dei mesi

Lavoriamo inizialmente con una matrice diversa da quella analizzata fino ad adesso: prendiamo la matrice che ha come righe tutti gli utenti (non solo quelli con il sistema di telelettura), e come colonne i mesi da Gennaio 2009 a Dicembre 2013. Chiamiamo questa matrice A. La matrice A avrà dunque circa 50000 righe e 60 colonne. Chiariamo quindi che stiamo lavorando in maniera diversa da quanto fatto fino ad adesso: in questo momento stiamo lavorando nello spazio generato dai mesi, nel quale dunque vivono le 50000 utenze. Eseguendo il comando PCA sulla matrice A otteniamo la figura 3.9. Da una

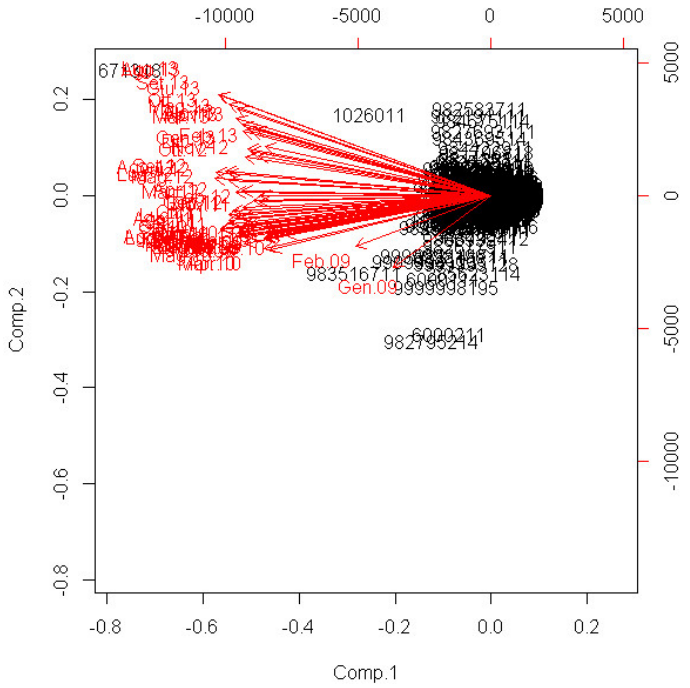

Figura 3.9: PCA

prima analisi le due componenti principali sembrano avere una chiara interpretazione: la componente 1 sembra avere il significato di quantità del consumo, mentre la componente 2 sembra avere una interpretazione temporale. Sottoliniamo che ogni numero riportato rappresenta univocamente un utente, e gli assi rossi rappresentano i mesi. Per capire se l'interpretazione data è giusta, andiamo a togliere dalla matrice A alcuni utenti che risultano isolati rispetto al gruppo, per vedere se riusciamo a far sparpagliare maggior-

mente la nuvola di utenti. Notiamo che l'utente situato in alto a sinistra, il cui numero è 981671318 (in figura si vede solo la parte finale del numero) è l'ospedale: questo avvalora l'ipotesi che la componente orizzontale rappresenti l'intensità dei consumi; infatti, come già sottolineato, l'utenza dell'ospedale è quella con il consumo maggiore (non solo tra gli utenti con sistema di telelettura ma anche tra tutti gli utenti). Anche l'interpretazione temporale della seconda componente è ragionevole: infatti l'utente 982795214, che si trova molto in basso nella figura, cessa di esistere nell'Aprile 2013, mentre, l'utente 982583711 che si trova molto in alto rispetto alla seconda componente inizia a esistere in Ottobre 2012. Cerchiamo adesso di ottenre una rappresentazione più chiara della precedente: come primo step andiamo a eliminare la riga della matrice A corrispondente all'utenza con consumo maggiore, e rieseguiamo nuovamente PCA su questa nuova matrice. Quello che otteniamo è mostrato dalla figura 3.10. Questa rappresentazione risulta

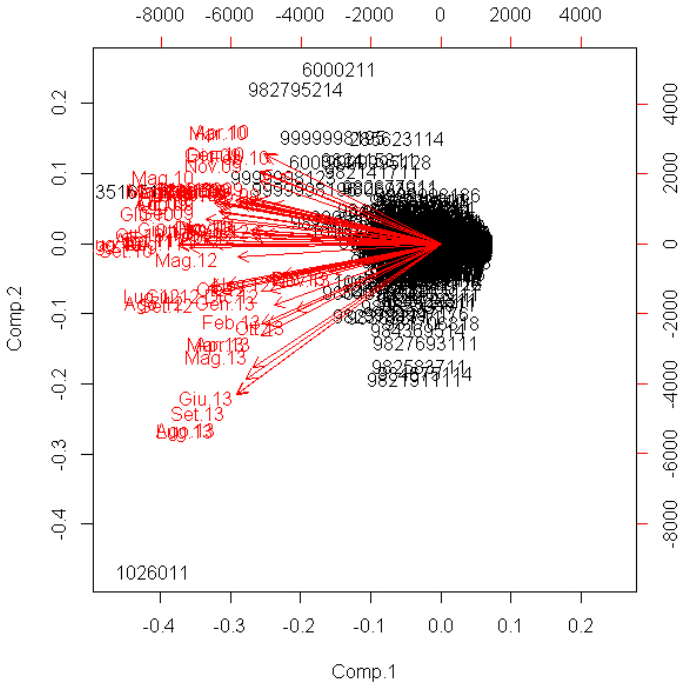

Figura 3.10: PCA senza ospedale

già molto più sparpagliata della precedente, e si riescono a individuare molti più utenti rispetto alla precedente. Osserviamo inoltre che la componente 2 risulta essere l'opposta della componente 2 precedente: infatti gli utenti che prima si trovavano in basso adesso si trovano in alto, e viceversa. Questo fatto non contraddice assolutamente quanto detto precedentemente, in quanto è solo un cambio di direzione e niente di più. Adesso che si leggono più chiaramente altri utenti andiamo a indagare anche su di esse. Inanzittutto osserviamo che l'utente 1026011, che si trova in basso a sinistra, è il carcere, che, dopo l'ospedale, è l'utenza che consuma maggiormente. L'utenza 6000211, che si trova ades-

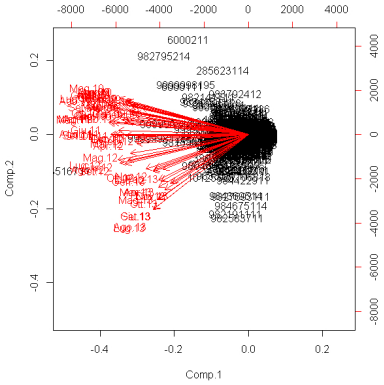

Figura 3.11: PCA

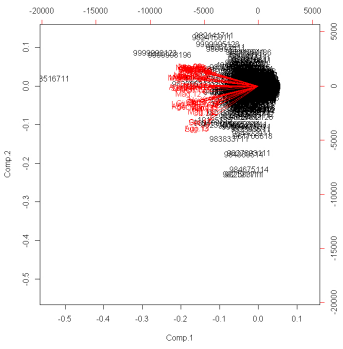

Figura 3.12: PCA

so molto in alto rispetto all'asse verticale, cessa di esistere nell'Ottobre 2012; l'utenza 982795214, come già detto, cessa di esistere nell'Aprile 2012, e questo torna, poichè si trova sotto rispetto all'utenza precedente. Riporto per completezza altri grafici che ho ottenuto andando a eliminare progressivamente alcuni utenti nelle figure 3.11 e 3.12.

3.5.2 PCA nello spazio generato dai mesi, con solo le utenze in telelettura

Mettiamoci adesso nello spazio generato dagli individui con sistema di telelettura. La matrice con la quale andiamo a lavorare è dunque la matrice che si ottiene semplicemente prendendo soltanto le 974 colonne, della matrice A, corrispondenti agli individui con la telelettura. Chiameremo in seguito $\tilde{A}$ la matrice così ottenuta. Osserviamo che $\tilde{A} = X^T$. Ripetendo PCA con la matrice $\tilde{A}$ otteniamo i risultati riportati in figura 3.13. L'ospedale è l'utenza in basso a destra, mentre notiamo che il carcere non compare, infatti non ha il contatore in telelettura. Questo grafico non dice nient'altro in più rispetto a quanto osservato precedentemente. Risulta solo un po' più chiaro in quanto compaiono molti meno individui.

3.5.3 PCA nello spazio degli individui

Dopo questa prima analisi, proviamo a lavorare nello spazio generato dagli individui. Consideriamo la matrice A^T, quindi una matrice con 60 righe e circa 50000 colonne. Sottoliniamo il fatto che stiamo lavorando con tutti gli individui, non solo quelli con sistema di telelettura. Ripetendo PCA quello che otteniamo è mostrato nella figura 3.14. Notiamo che gli utenti che si distinguono dalla nuvola centrale sono l'utente 1026011, ossia il carcere, l'utente 981671318, ossia l'ospedale e l'utente 982795214, che sono proprio gli stessi utenti che si distinguevano nella PCA fatta precedentemente nell'altro spazio. In questo caso il significato delle due componenti principali non rislta molto comprensibile: anche qui la componente orizzontale sembra avere un'interpretazione temporale. La cosa

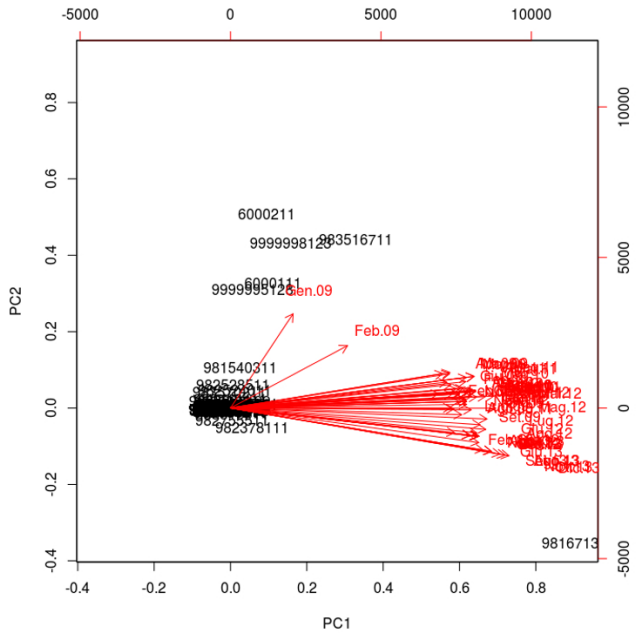

Figura 3.13: PCA nello spazio dei mesi con solo utenti in telelettura

che possiamo sicuramente affermare è che tranne poche utenze abbiamo dei fortissimi allineamenti.

3.5.4 PCA nello spazio generato dai soli utenti in telelettura

Mettiamoci adesso nello spazio generato dai soli utenti in telelettura. Questo equivale a lavorare con la matrice X introdotta all'inizio di questo capitolo. Questa matrice si ottiene facilmente trasponendo la matrice A, ed predendo solo le 974 colonne corrispondenti agli utenti con contatore in telelettura. La ricerca delle componenti principali produce il risultato mostrato in figura 3.16. Anche in questo caso la prima componente sembra avere un'interpretazione temporale. La componente 2 può essere interpretata come intensità, infatti i mesi invernali si trovano più in basso rispetto ai mesi estivi dei rispettivi anni. L'unico individuo che risulta emergere rispetto agli altri è l'ospedale. Nei disegni che seguono (figure 3.16 e 3.50) ho rimosso progressivamente alcuni mesi per rendere più leggibile e sparpagliato il disegno.

Concludendo, le varie analisi effettutate tramite PCA ci portano a dire che sia nello spazio degli utenti, sia nello spazio dei mesi, ci sono forti allineamenti.
Dopo queste prime analisi, iniziamo a formulare dei modelli che ci permettano di prevedere il mese di Dicembre 2013.

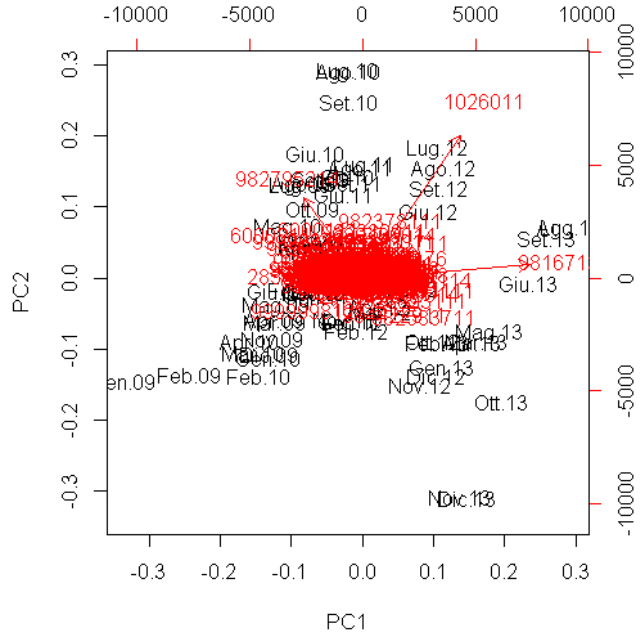

Figura 3.14: PCA nello spazio degli individui

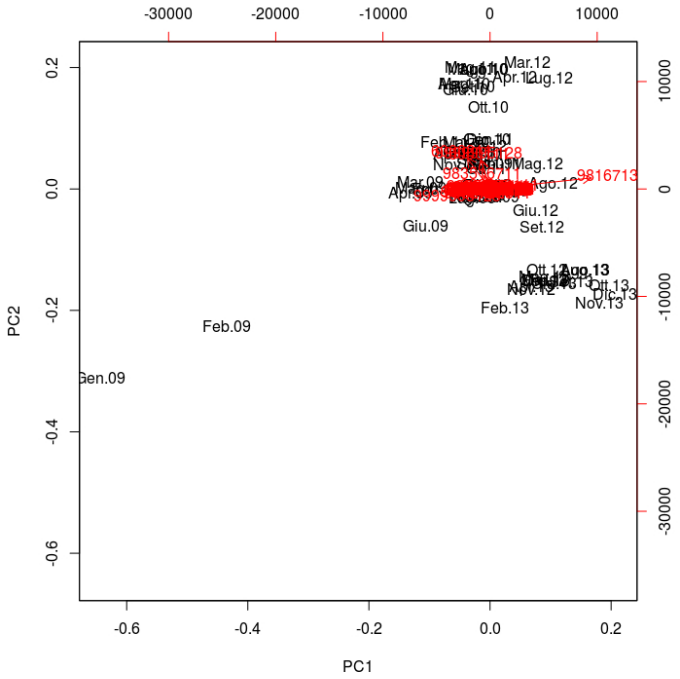

Figura 3.15: PCA nello spazio degli individui con telelettura

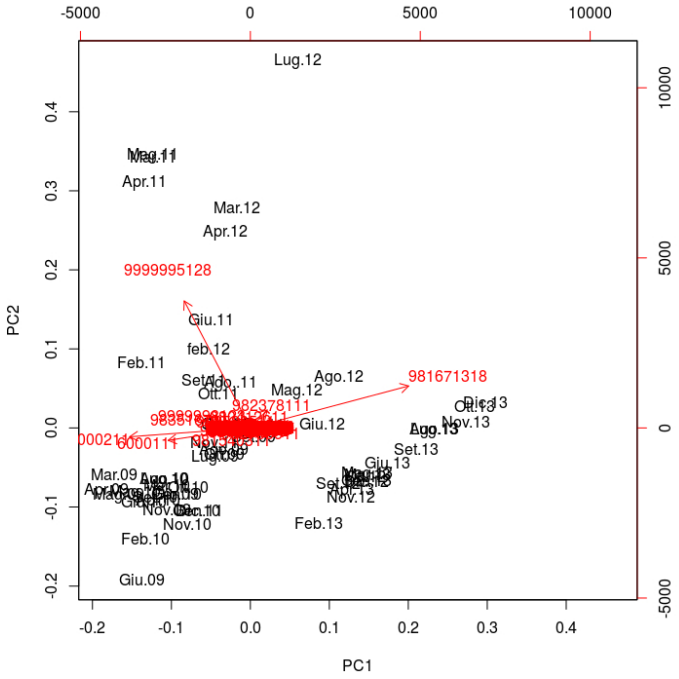

Figura 3.16: PCA nello spazio degli individui con telelettura

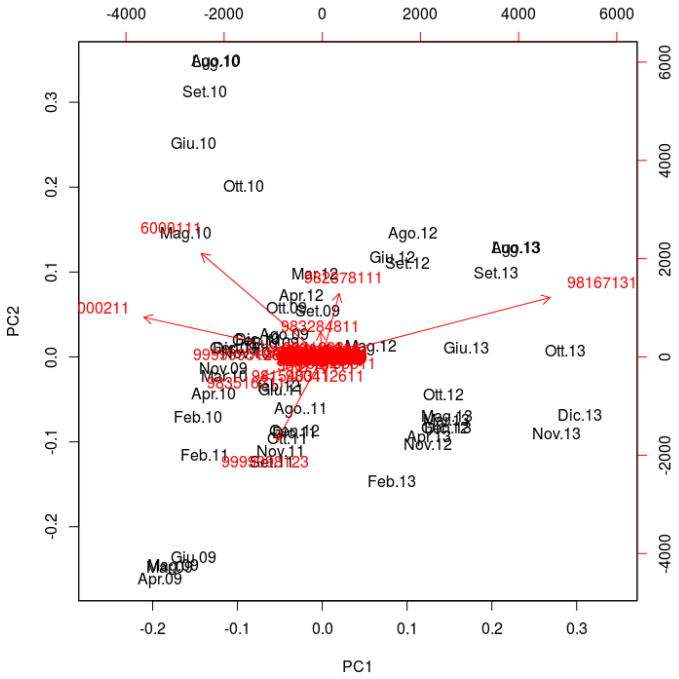

Figura 3.17: PCA nello spazio degli individui con telelettura

3.6 Regressione lineare multipla

Dalla descrizione dei dati, è chiaro che si tratta di un problema regressivo: quindi come primo tentativo cerchiamo di applicare la classica regressione lineare multipla al nostro modello. Lavoriamo con la matrice X introdotta inizialmente, ma di essa selezioniamo solo le righe corrispondenti ai mesi da Gennaio 2011 a Ottobre 2013. Chiamiamo questa matrice $\tilde{X}$. Analogamente consideriamo il vettore y introdotto all'inizio del capitolo, e di esso consideriamo solo le righe corrispondenti ai mesi da Gennaio 2011 a Dicembre 2013: chiamiamo questo vettore $\tilde{y}$. Il nostro scopo è creare un modello sulla base dei dati contenuti in $\tilde{X}$ e $\tilde{y}$, per poi andare a prevedere il consumo totale dei mesi di Novembre e Dicembre 2013. Il modello che stiamo cercando è dunque della forma

$$\tilde{y} = a_{k_1}\tilde{X}_{k_1} + \cdots + a_{k_p}\tilde{X}_{k_p}, \text{con } 1 \leq k_1 < \cdots < k_p \leq 701$$

dove i coefficienti $a_{k_1}, \ldots, a_{k_p}$ sono i coefficienti dati dalla regressione lineare multipla. Proviamo adesso, una volta implementato questo modello, a prevedere il totale dei mesi di Novembre e Dicembre prendendo $p = 701$, ossia con tutti gli utenti. I risultati ottenuti sono i seguenti:

$$Nov_{2013} = 455526.2 \qquad Dic_{2013} = 453831.2$$
$$Nov_{2013}^{stimato} = -532369297 \qquad Dic_{2013}^{stimato} = -624268567$$

Da questi risultati possiamo capire che il modello non funziona. Infatti andando a osservare meglio il modello otteniamo che i coefficienti sono calcolati in modo da non ottenere nessun errore di previsione sui dati da Gennaio 2011 a Ottobre 2013, ma commettono errori grandissimi sui mesi successivi. Un'altra cosa che ci spaventa è che tra i 701 coefficienti fissati, la maggior parte risultano NA. Questo risultato è dovuto al fatto che il nostro modello non include tutti gli utenti a causa di allineamenti: in particolare, se ci sono due o più utenti particolarmente allineati, sceglie il coefficiente per solo uno di loro, e decide di non includere nel modello gli altri utenti allineati, in quanto non servirebbero per migliorare il modello. Inoltre alcuni dei coefficienti non NA, risultano negativi, e questa cosa non ha molto senso. Addirittura si arriva a risultati per i mesi di Novembre e Dicembre 2013 negativi, e ciò non ha senso. Questo è un chiaro esempio di modello che si adatta perfettamente ai dati, ma ha uno scarsissimo potere predittivo. Questo difetto è dovuto dal fatto che le variabili, ossia gli utenti, sono linearmente dipendenti e dunque i coefficienti risultano estremamente instabili. La regressione ha quindi due difetti irrisolvibili: la grande instabilità dei coefficienti e l'allineamento dei fattori.
Proviamo adesso a riformulare la regressione lineare multipla con meno utenti.

3.6.1 Previsione con 600 utenti

Selezioniamo adesso $p = 600$ e cerchiamo di prevedere il totale dei mesi di Novembre e Dicembre 2013 40 volte, al variare dei 600 utenti selezionati. Ogni volta calcoliamo l'errore relativo commesso. Riportiamo i valori ottenuti nel grafico di figura 3.18.

Come si può osservare, gli errori relativi vengono molto grandi, e questo è un effetto dell'overfitting. Proviamo a vedere se i risultati migliorano abbassando il numero d variabili, ossia il numero di utenti.

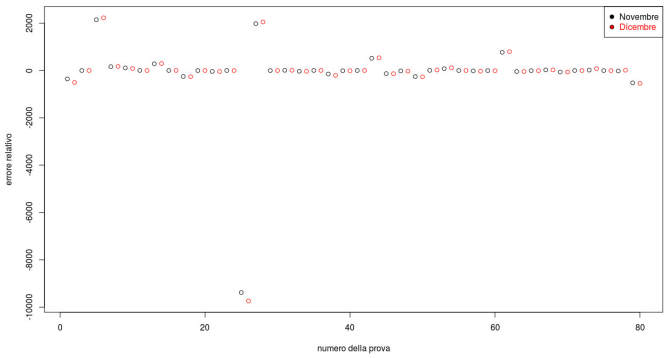

Figura 3.18: Errori per le previsioni con 600 utenti

3.6.2 Previsioni con 200 utenti

Riportiamo nella figura 3.19 i risultati ottenuti selezionando $p = 200$. Anche in questo

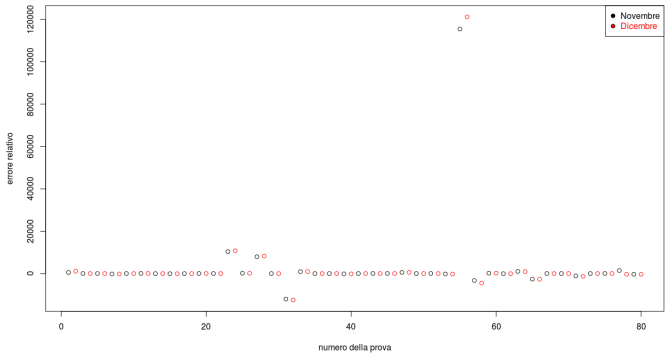

Figura 3.19: Errori per le previsioni con 200 utenti

caso, gli errori vengono enormi, sempre a causa di un overfitting. Infine proviamo a selezionare $p = 20$.

3.6.3 Previsione con 20 utenti

Riportiamo nella figura 3.20 i risultati ottenuti selezionando $p = 20$. In questo caso gli errori risultano più moderati, ma sempre troppo grandi considerando che si trattano di errori relativi.
Possiamo concludere che la regressione lineare multipla effettuata in questo modo risulta inadeguata al nostro scopo per problemi di overfitting. Passiamo quindi all'applicazione di un altro metodo, il metodo PLS.

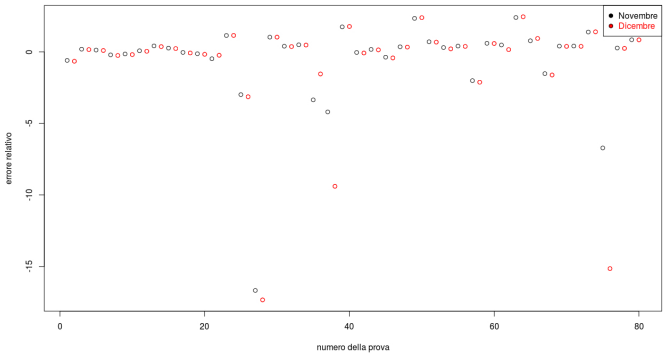

Figura 3.20: Errori per le previsioni con 20 utenti

3.7 PLS

Come prima variante alla regressione lineare multipla, testiamo il metodo PLS.
Gli errori ottenuti per i mesi di Novembre e Dicembre, nel modello creato selezionando tutti gli utenti, risultano i seguenti:

$$Nov_{2013} = 455526.2 \qquad Dic_{2013} = 453831.2$$

$$Nov_{2013}^{stimato} = 464316.3 \qquad Dic_{2013}^{stimato} = 478932.4$$

$$Nov_{2013} - Nov_{2013}^{stimato} = -8790.061 \qquad Dic_{2013} - Dic_{2013}^{stimato} = -25101.18.$$

L'errore è minore per il mese di Dicembre, ed ammonta a solo 1500 mc. Inoltre osserviamo che il modello commette, per entrambi i mesi, un errore in eccesso.
Riportiamo anche gli errori relativi, per maggiore chiarezza.

$$Errore_{Nov_{2013}} = -0.019 \qquad Errore_{Dic_{2013}} = -0.055$$

Osserviamo che il modello commette un errore maggiore per il mese di Dicembre, cosa molto ragionevole per quanto già osservato. Osserviamo inoltre che il modello commette errori per eccesso in entrambe le previsioni. Come fatto precedentemente andiamo a fare varie prove, selezionando un numero fissato di utenti.

3.7.1 Previsioni con 600 utenti

La prima prova che ho effettuato è stata con 600 utenti, sempre facendo 40 prove. I risultati ottenuti sono illustrati nella seguente figura 3.21. I risultati ottenuti sono molto stabili, e variano intorno a un errore medio di -0.02 per il mese di Novembre, mentre variano intorno a -0.06.

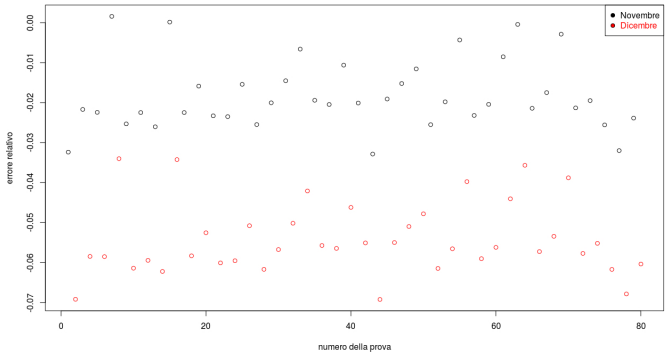

Figura 3.21: Errori per le previsioni con 600 utenti

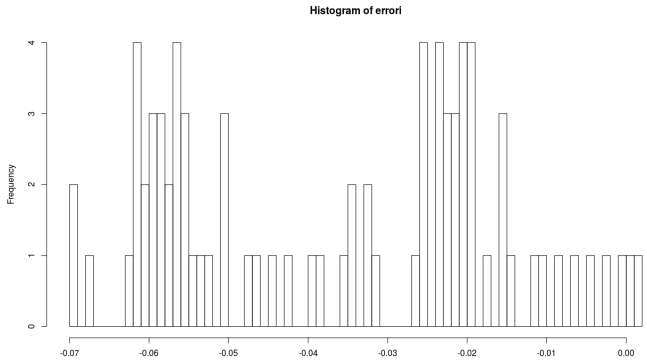

Figura 3.22: Istogramma degli errori con 600 utenti

3.7.2 Previsione con 200 utenti

I risultati ottenuti selezionando 200 utenti sono riportati in figura 3.23. In questo caso

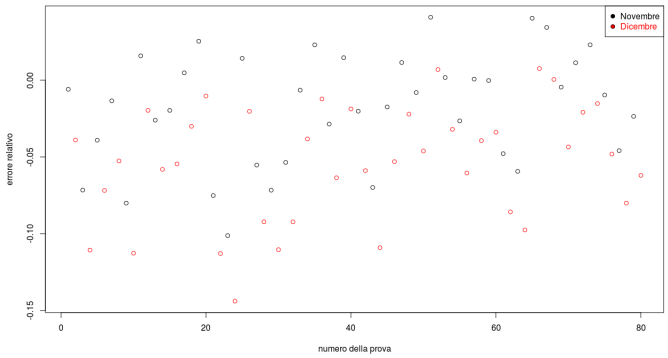

Figura 3.23: Errori per le previsioni con 200 utenti

gli errori risultano in prevalenza per eccesso, anche se alcuni errori commessi per il mese

di Novembre risultano per difetto. Gli ordini di grandezza risultano i medesimi del caso con 600 utenti.

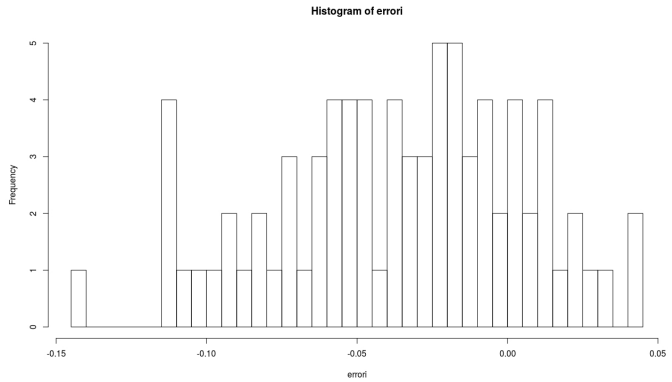

Figura 3.24: Istogramma degli errori con 200 utenti

3.7.3 Previsioni con 20 utenti

Gli errori ottenuti con 20 utenti sono riportati in figura 3.28. Osserviamo che in tal

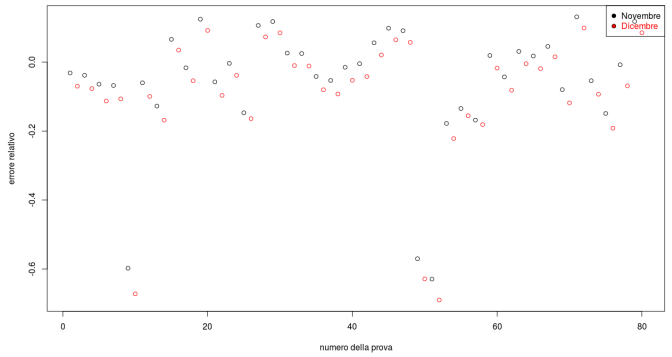

Figura 3.25: Errori per le previsioni con 20 utenti

caso gli errori risultano quasi sempre dello stesso ordine di grandezza, anche se talvolta risultano enormemente maggiori.

3.7.4 Conclusioni

Analogamente a quanto fatto per il modello precedente, riportiamo in figura 3.27 le curve dell'errore medio, del quantile di livello 0.95 e della deviazione standard, al variare del numero di utenti. Come si può notare, questi risultati presentano il classico andamento a gomito. L'errore medio si stabilizza intorno a 0.019. Andando a osservare i grafici pare opportuno scegliere un numero di utenti pari a 400, poichè la varie curve risultano abbastanza stabili dopo tale soglia.

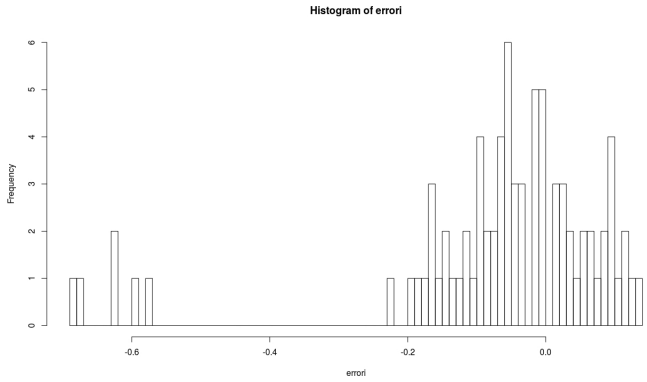

Figura 3.26: Istogramma degli errori con 20 utenti

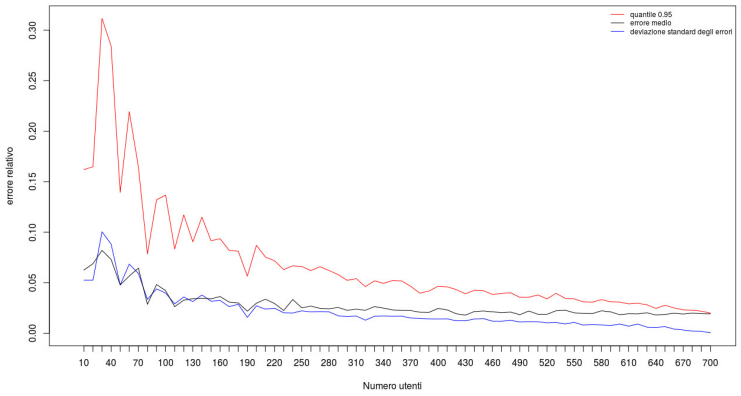

Figura 3.27: Risultati PLS Faraway

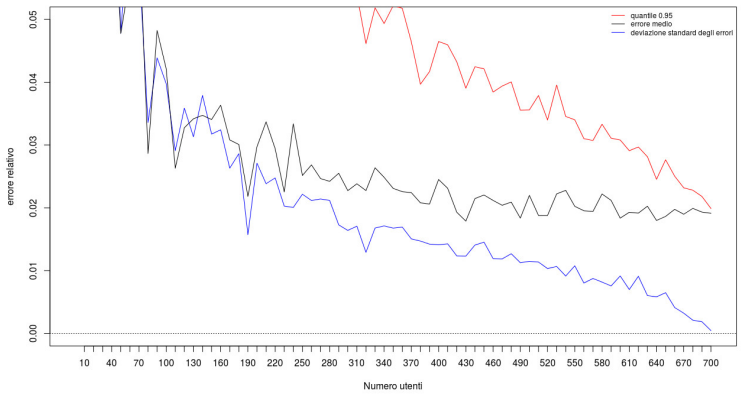

Figura 3.28: Risultati PLS Faraway

Prima di andare a fare ulteriori analisi per il questo modello con 400 utenti andiamo a introdurre un altro metodo.

3.8 PLS approssimato

Analizziamo adesso un'approssimazione del metodo PLS, ossia il modello come abbiamo visto nel metodo iterativo, fermando però le iterazioni dopo solo pochi passi.
Cerchiamo solo due componenti nello spazio delle X. Applicando il modello con tutti gli utenti con sistema di telelettura, senza contare gli utenti che hanno anche solo un consumo mensile uguale a 0 otteniamo i seguenti risultati:

$$Nov_{2013} = 455526.2 \qquad Dic_{2013} = 453831.2$$

$$Nov_{2013}^{stimato} = 463347.1 \quad Dic_{2013}^{stimato} = 468722.8$$

$$Nov_{2013} - Nov_{2013}^{stimato} = -7820.909 \qquad Dic_{2013} - Dic_{2013}^{stimato} = -14891.598,$$

che tradotti in errori relativi significano

$$Errore_{Nov_{2013}} = -0.017 \qquad Errore_{Dic_{2013}} = -0.032.$$

Osserviamo che il modello commette un errore maggiore per il mese di Dicembre, cosa molto ragionevole per quanto già osservato. Osserviamo inoltre che il modello commette errori per eccesso in entrambe le previsioni. Come fatto precedentemente andiamo a fare varie prove, selezionando un numero fissato di utenti.

3.8.1 Previsioni con 600 utenti

La prima prova che ho effettuato è stata con 600 utenti, sempre facendo 40 prove. I risultati ottenuti sono illustrati nella seguente figura 3.29. I risultati sono molto stabili,

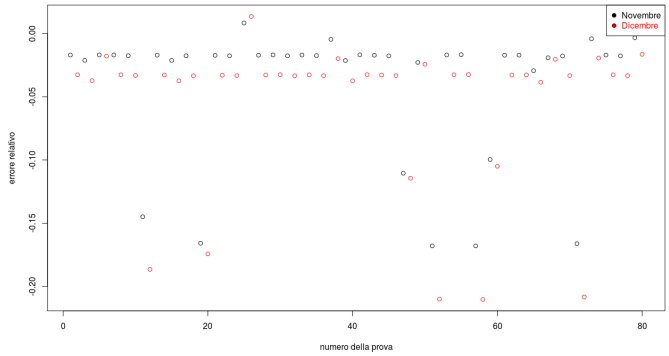

Figura 3.29: Errori per le previsioni con 600 utenti

al variare degli utenti, tranne per 3 casi in cui gli errori sono molto più grandi. Inoltre quello che si può osservare è che l'errore per Novembre risulta sempre minore in valore assoluto rispetto al valore di Dicembre.

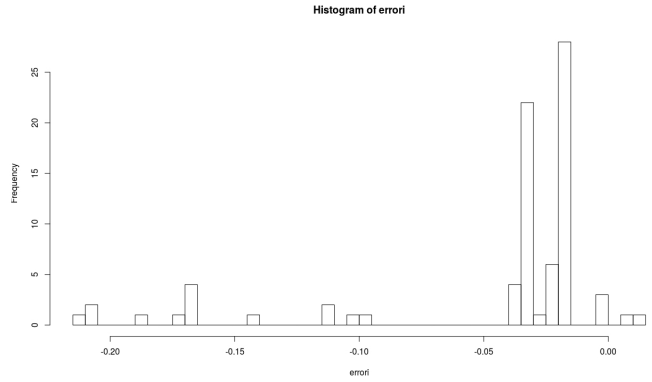

Figura 3.30: Istogramma degli errori con 600 utenti

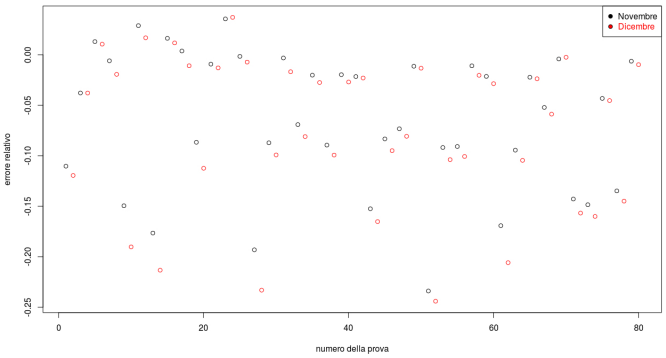

Figura 3.31: Errori per le previsioni con 200 utenti

3.8.2 Previsione con 200 utenti

Proviamo adesso selezionando 200 utenti. I risultati sono riportati in figura 3.31. La prima cosa che notiamo è che i risultati diventano molto meno stabili rispetto al caso precedente, ma l'errore in valore assoluto per Novembre risulta anche in questo caso minore del relativo errore per Dicembre.

3.8.3 Previsione con 20 utenti

Proviamo adesso a prevedere con soli 20 utenti. I risultati sono rappresentati in figura 3.33. Andiamo adesso a rappresentare l'andamento di questi errori in un unico grafico, per poter trarre delle conclusioni.

3.8.4 Conclusioni

Riportiamo in figura 3.35 un grafico che riporta l'errore medio, il quantile di livello 0.95 e la deviazione standard del **valore assoluto degli errori** commessi per il mese di Novembre al variare del numero di utenti da 10 a 700. Consideriamo il valore assoluto in

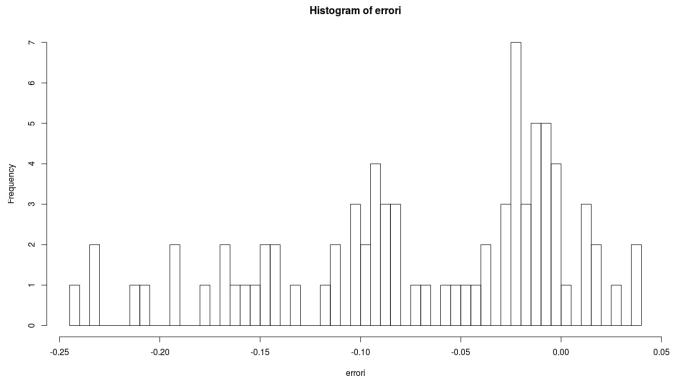

Figura 3.32: Istogramma degli errori con 200 utenti

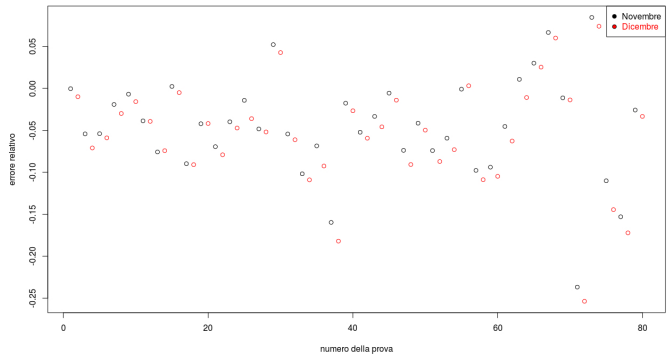

Figura 3.33: Errori per le previsioni con 20 utenti

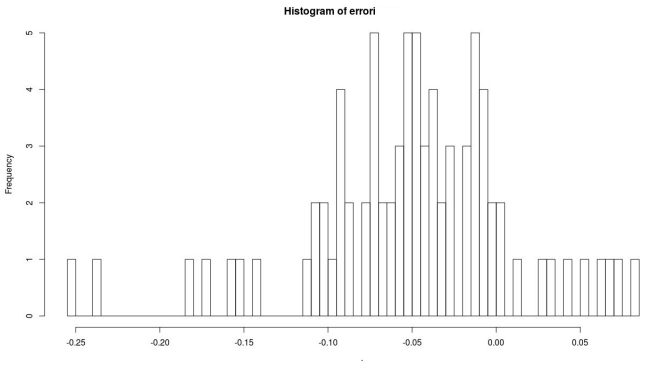

Figura 3.34: Istogramma degli errori con 20 utenti

quanto, se considerassimo gli errori con segno, l'errore medio verrebbe falsato. Andando a ingrandire il grafico otteniamo i risultati mostrati in figura 3.36.

La cosa che si può notare è la differenza di andamento con il grafico precedente: solitamente i grafici di questo tipo presentano un andamento a gomito, in questo grafico

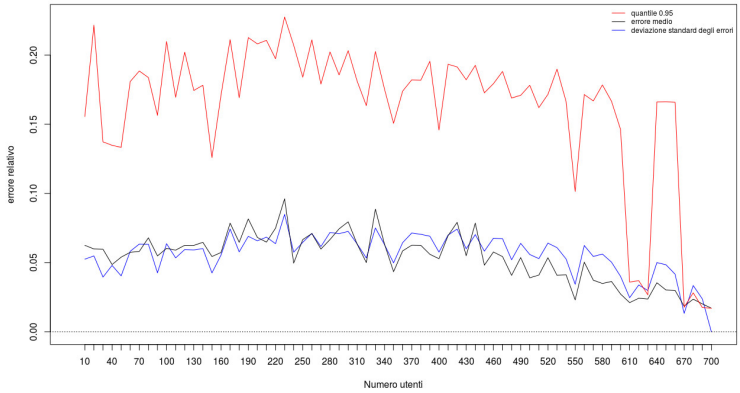

Figura 3.35: Risultati PLS

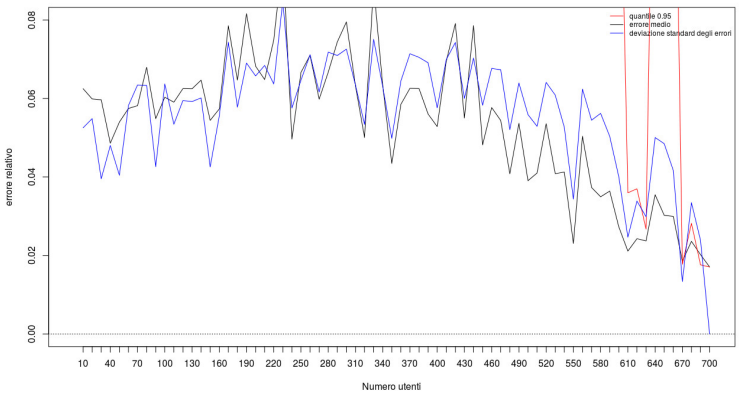

Figura 3.36: Risultati PLS

l'errore medio sembra essere piccolo per pochi utenti, aumentare in seguito per poi tornare a diminuire da 600 utenti in poi. Una cosa interessante è che l'errore medio più vicino a 0 si ottiene proprio per $p = 40$, ossia con soli 40 utenti. In tal caso si ottiene un errore medio di 0.048. Ovviamente avendo fatto solo 40 prove con $p = 40$ il risultato può essere influenzato dagli utenti selezionati, quindi, prima di procedere facciamo 400 prove con $p = 40$ con il metodo PLS. I risultati ottenuti in questo caso sono i seguenti:

$$Errore_{medio} = 0.052$$

$$Quantile_{0.95} = 0.15$$

$$Sd = 0.047$$

Proviamo anche ad effettuare un maggior numero di prove al variare di p da 10 a 50. Otteniamo i risultati riportati nella tabella che segue:

	$p = 10$	$p = 20$	$p = 30$	$p = 40$	$p = 50$
Errore medio	0.063	0.051	0.052	0.052	0.054
Quantile 0.095	0.17	0.14	0.15	0.15	0.17
Deviazione standard	0.055	0.044	0.047	0.047	0.051

Dai precedenti risultati, si nota che le differenze sono veramente minime, ma i risultati ottenuti suggeriscono la scelta di $p = 20$.
Concludendo possiamo ritenere che un buon metodo per risolvere il nostro problema, sia l'applicazione del metodo PLS approssimato, che fornisce risultati accettabili già selezionando pochi utenti. Se inoltre l'azienda non ritiene troppo dispendioso andare a leggere 700 contatori, il metodo PLS approssimato applicato con $p = 700$ fornisce risultati ancora migliori, con un errore medio di circa 0.19.

3.9 Terzo modello

Dopo aver applicato modelli di tipo regressivi, introduciamo un modello più banale, basato su semplici intuizioni.
Il modello che andiamo a creare è il seguente:

$$\tilde{y} = \alpha_{k_1}\tilde{X}_{k_1} + \cdots + \alpha_{k_p}\tilde{X}_{k_p}, \text{ con } \ 1 \leq k_1 < \cdots < k_p \leq 701$$

dove

$$\alpha_{k_1} = \frac{1}{p}\frac{||\tilde{y}||_1}{||\tilde{X}_{k_1}||_1}, \ldots, \ \alpha_{k_p} = \frac{1}{p}\frac{||\tilde{y}||_1}{||\tilde{X}_{k_p}||_1}.$$

In particolare il modello seleziona p utenti tra i 701 con la telelettura, e il coefficiente che assegna ad ogni utenza è proporzionale al rapporto tra il consumo totale di quell'utente dal mese di Gennaio 2011 al mese di Ottobre 2013, e il consumo totale del comune di Lucca rispettivo agli stessi mesi. Inoltre questo coefficiente viene diviso per il numero di utenti selezionati per una sorta di normalizzazione.
Una volta stimati questi coefficienti, possiamo andare a prevedere i mesi di Novembre e Dicembre 2013.
La prima prova che ho effettuato, dopo aver implementato il modello, è stata fatta prendendo tutti gli utenti con telelettura. Applicando il modello sopra discritto, con $p = 701$, i risultati per i mesi predetti sono i seguenti:

$$Nov_{2013} = 455526.2 \qquad Dic_{2013} = 453831.2$$

$$Nov_{2013}^{stimato} = 461645.5 \qquad Dic_{2013}^{stimato} = 476143.5$$

$$Nov_{2013} - Nov_{2013}^{stimato} = -6119.28 \qquad Dic_{2013} - Dic_{2013}^{stimato} = -22312.33.$$

L'errore è minore per il mese di Dicembre, ed ammonta a solo 1500 mc. Inoltre osserviamo che il modello commette, per entrambi i mesi, un errore in eccesso.
Riportiamo anche gli errori relativi, per maggiore chiarezza.

$$Errore_{Nov_{2013}} = -0.013 \qquad Errore_{Dic_{2013}} = -0.049$$

Osserviamo che andando a guardare il consumo per i mesi di Novembre e Dicembre dei 701 individui in telelettura per 701 di essi il consumo risulta maggiore nel mese di Dicembre (questo per la maggior parte degli utenti deriva dal fatto che Dicembre ha un giorno in più di Novembre).

3.9.1 Previsione con 600 utenti

Continuiamo a lavorare con lo stesso modello: quello che ho fatto è stato di selezionare 600 utenti casuali tra i 701, ed ho fatto questa selezione 40 volte. Quindi il modello che ho creato ha $p = 600$, ed ho creato 40 modelli, facendo variare i 600 utenti selezionati. Per ogni modello creato ho salvato gli errori ottenuti rispettivamente per il mese di Novembre 2013 e Dicembre 2013, ottenendo dunque un vettore contenente 80 errori. Plottando il vettore otteniamo i risultati riportati in figura 3.37. Anche in questo caso il modello

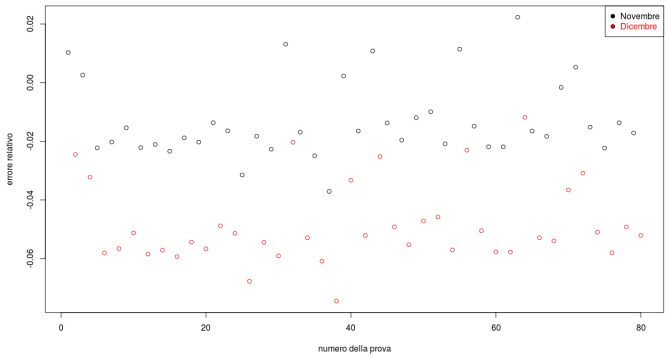

Figura 3.37: Errori per le previsioni con 600 utenti

commette quasi sempre errori per eccesso. Riportiamo anche l'istogramma degli errori, in quanto da esso si può notare la chiara divisione tra gli errori di Novembre e quelli di Dicembre, già visibile nel grafico precedente. Come ci si poteva aspettare, abbiamo due gruppi separati: il gruppo più a sinistra rappresenta gli errori del mese di Dicembre, mentre il gruppo più a destra rappresenta gli errori del mese di Novembre.

3.9.2 Previsione con 200 utenti

Continuiamo a lavorare con lo stesso modello: quello che ho fatto è stato di selezionare 200 utenti casuali tra i 701, ed ho fatto questa selezione 40 volte. Quindi il modello che ho creato ha $p = 200$, ma ho creato 40 modelli, facendo variare i 200 utenti selezionati. Per ogni modello creato ho salvato gli errori ottenuti rispettivamente per il mese di Novembre

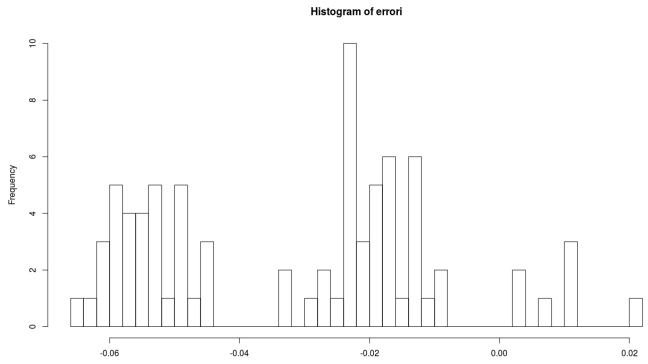

Figura 3.38: Istogramma degli errori con 600 utenti

2013 e Dicembre 2013, ottenendo dunque un vettore contenente 80 errori. Plottando il vettore otteniamo i risultati riportati in figura 3.39. La cosa che si nota è che gli errori

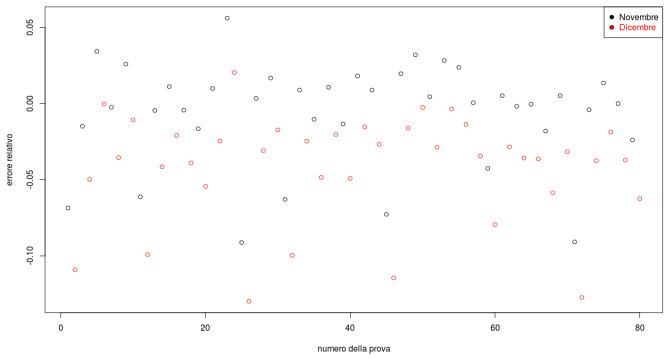

Figura 3.39: Errori per le previsioni con 200 utenti

sono tutti positivi, quindi il nostro modello sbaglia sempre in eccesso: l'errore è calcolato come valore stimato meno valore effettivo.
In questo caso, da questo grafico non sembra ovvia la divisione dei due mesi come nel caso precedente. Riportiamo l'istogramma, in figura 3.51, per vedere se sussiste una tale divisione oppure no. In tal caso la divisione tra i due mesi non sembra apparire. Osserviamo inoltre che il picco più alto dell'istogramma è raggiunto in corrispondeza del valore 0, mentre nel caso precedente non c'era nessuna barra in corrispondenza dello 0.

3.9.3 Previsione con 20 utenti

Continuiamo a lavorare sulla stessa idea precedente, ma con $p = 20$. Plottando il vettore degli errori otteniamo i risultati riportati in figura 3.50. Anche in questo caso riporiamo l'istogramma degli errori.

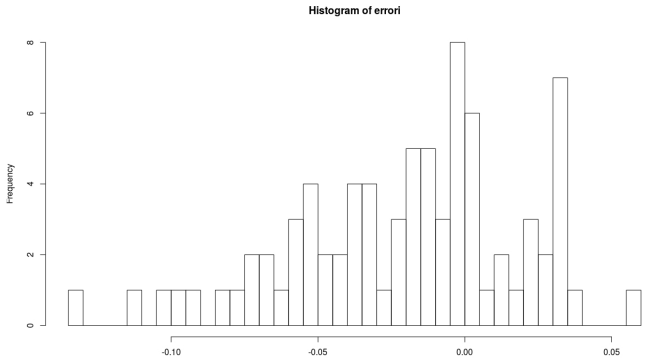

Figura 3.40: Istogramma degli errori con 200 utenti

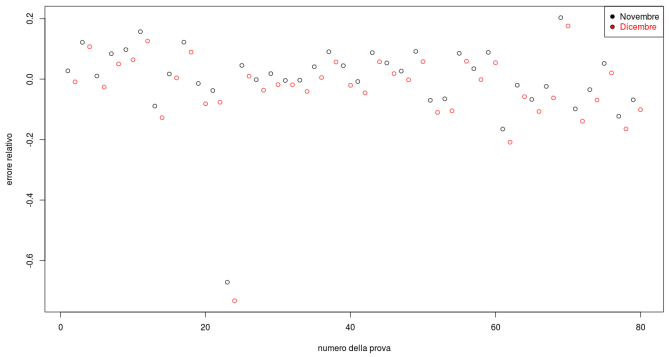

Figura 3.41: Errori per le previsioni con 20 utenti

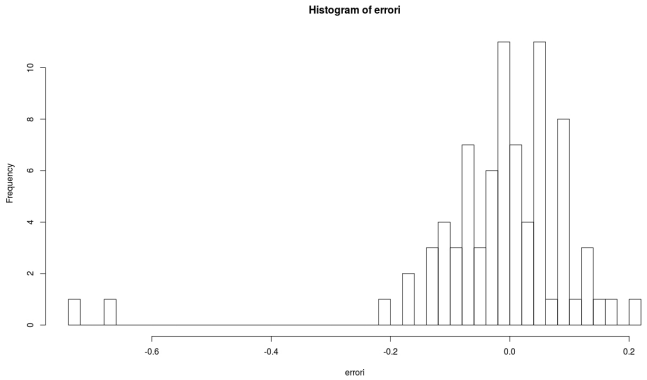

Figura 3.42: Istogramma degli errori con 20 utenti

3.9.4 Conclusioni

Riportiamo in figura 3.43 un grafico che riporta l'errore medio, il quantile di livello 0.0.5 e la deviazione standard del valore assoluto degli errori commessi per il mese di Novembre al variare del numero di utenti da 10 a 700. Andando a ingrandire il grafico nelle immagini

successive possiamo osservare che l'errore medio all'aumentare del numero degli utenti si stabilizza vicino al valore 0.013.

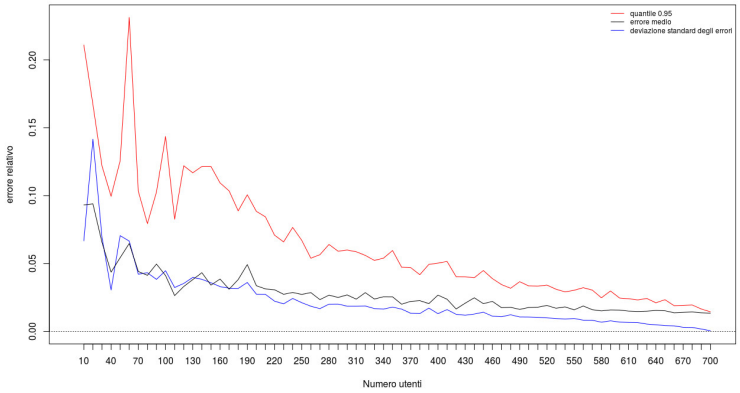

Figura 3.43: Risultati terzo modello

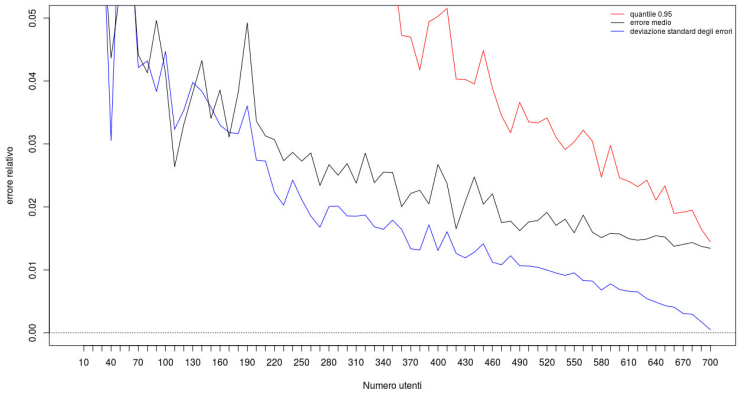

Figura 3.44: Risultati terzo modello

Dall'osservazione di questi grafici possiamo notare l'andamento a gomito abbastanza simmetrico delle varie curve. Questo ci fa osservare che le prestazioni di questo modello sono governate dalla legge dei grandi numeri. Un riscontro di ciò possiamo ottenerlo dal grafico riportato in figura 3.45: in questo grafico troviamo sull'asse delle x il logaritmo dei numeri da 10 a 700 con frequenza 10, mentre sull'asse delle y troviamo il logaritmo degli errori al variare del numero degli utenti. Si capisce che l'andamento di questo grafico è quello di una retta con inclinazione $-\frac{1}{2}$, e questo ci conferma che c'è dietro la legge dei grandi numeri.
Osserviamo inoltre la somiglianza di questi grafici di riassunto e di quelli del modello PLS Faraway.
In base a questi grafici potremmo ipotizzare che questo modello si può ritenere accettabile con un numero di utenti pari a 400: infatti da 400 utenti in poi le varie curve risultano abbastanza stabili, per cui non c'è ragione per selezionare più utenti. Prima

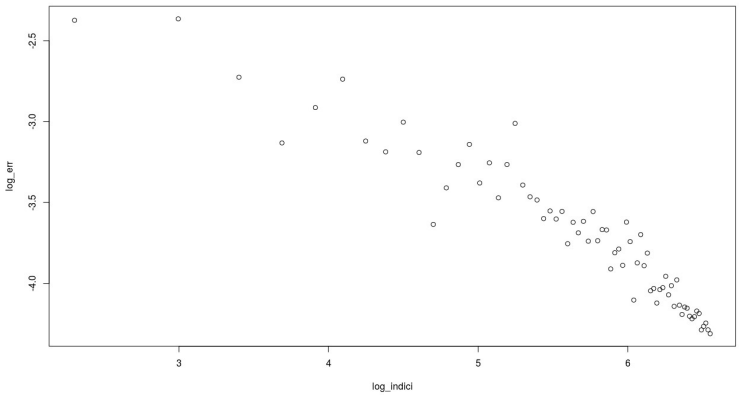

Figura 3.45: Grafico log-log

di andare a fare ulteriori prove con questo modello con 400 utenti introduciamo altri modelli plausibili.

3.10 Quarto modello

Andiamo a introdurre un modello un pochino più avanzato del precedente. Il modello è simile al precedente, ma cambiano coefficienti in questo modo

$$\tilde{y} = \beta_{k_1}\tilde{X}_{k_1} + \cdots + \beta_{k_p}\tilde{X}_{k_p}, \text{ con } \ 1 \leq k_1 < \cdots < k_p \leq 701$$

dove, i coefficienti β_{k_j} sono costruiti nel seguente modo

$$\beta_{k_j} = \frac{\rho_{k_j}}{\rho}\frac{||\tilde{y}||_1}{||\tilde{X}_{k_j}||_1} \quad \text{per} \quad j = 1, \ldots, p$$

dove

$$\rho_{k_j} = Cor(\tilde{X}_{k_j}, \tilde{y}) \text{ per } \ j = 1, \ldots, p \text{ e}$$

$$\rho = \sum_{j=1}^{p} \rho_{k_j}.$$

Vediamo allora in base a questo modello come vengono i risultati.

$$Nov_{2013} = 455526.2 \qquad Dic_{2013} = 453831.2$$

$$Nov_{2013}^{stimato} = 330820.9 \qquad Dic_{2013}^{stimato} = 339398.1$$

$$Nov_{2013} - Nov_{2013}^{stimato} = 124705.3 \qquad Dic_{2013} - Dic_{2013}^{stimato} = 114433.1$$

$$Errore_{Nov_{2013}} = 0.27 \qquad Errore_{Dic_{2013}} = 0.25$$

Notiamo che gli errori commessi da questo modello sono incredibilmente maggiori rispetto agli errori commessi dal primo modello. Inoltre questo modello compie per entrambi i mesi errori in difetto. Come fatto per il modello 1 riportiamo i risultati ottenuti facendo 40 prove con un numero di utenti pari a 600, 200 e 20.

3.10.1 Previsione con 600 utenti

Dai risultati riportati in figura 3.46 notiamo che anche selezionando 600 utenti, il secondo modello commette sempre errori in eccesso. Un'altra cosa che può essere osservata nella

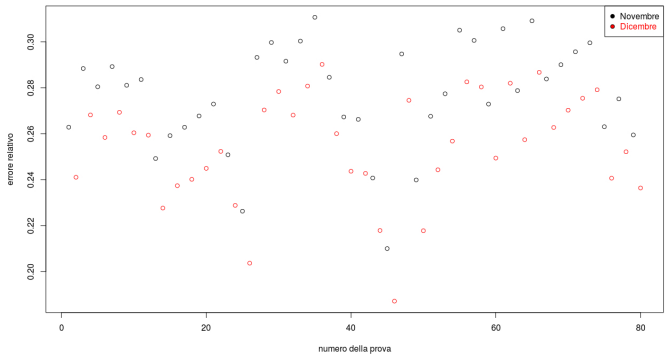

Figura 3.46: Errori per le previsioni con 600 utenti

figura è che, per ogni selezione di utenti, il modello commette un errore maggiore per il mese di Novembre, rispetto al mese di Dicembre. Ciò può risultare strano in quanto basandoci sui mesi precedenti a Novembre, l'errore dovrebbe risultare minore per il mese di Novembre, rispetto al mese di Dicembre. Anche in questo caso riporiamo l'istogramma degli errori.

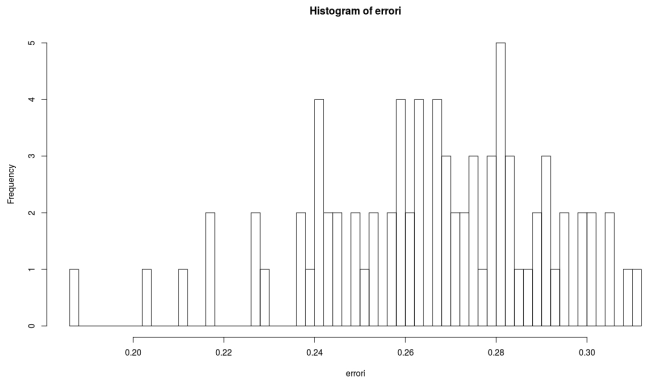

Figura 3.47: Istogramma degli errori con 600 utenti

3.10.2 Previsione con 200 utenti

Selezionando solo 200 utenti, possiamo notare che l'errore in media aumenta rispetto al modello basato con 600 utenti. Anche in questo caso, in ogni simulazione, l'errore del mese di Novembre risulta maggiore rispetto all'errore relativo al mese di Dicembre. Anche in questo caso riporiamo l'istogramma degli errori.

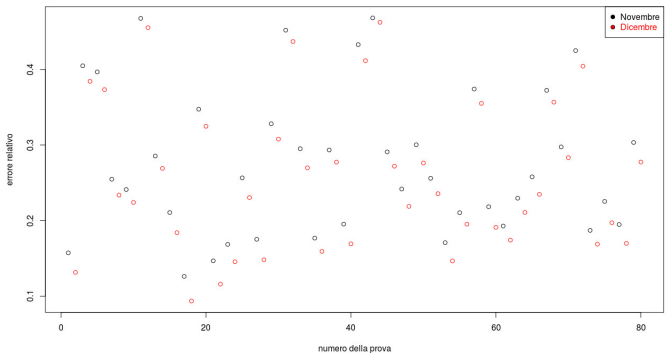

Figura 3.48: Errori per le previsioni con 200 utenti

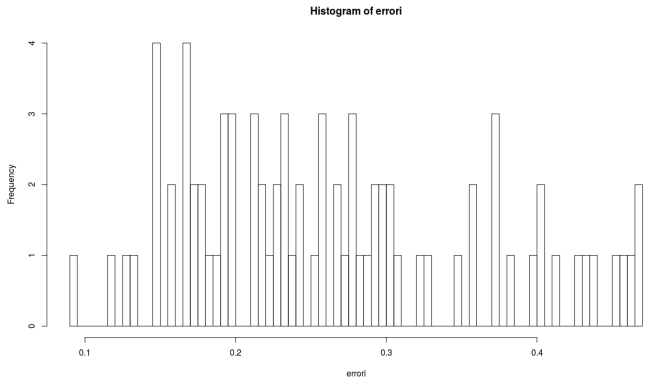

Figura 3.49: Istogramma degli errori con 200 utenti

3.10.3 Previsione con 20 utenti

Selezionando solo 20 utenti, notiamo che l'errore medio cresce notevolmente. Possiamo quindi a priori scartare una selezione di un numero così basso di utenti.

3.10.4 Conclusioni

Riportiamo in figura 3.52 un grafico che riporta l'errore medio, il quantile di livello 0.95 e la deviazione standard degli errori fatti al variare del numero di utenti da 10 a 700, con frequenza 10, per la previsione del mese di Novembre 2013. Abbiamo deciso di

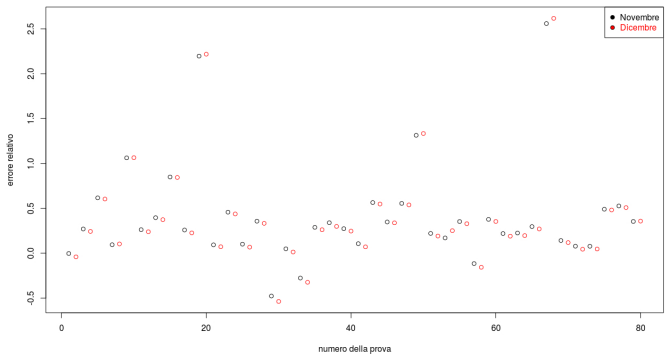

Figura 3.50: Errori per le previsioni con 20 utenti

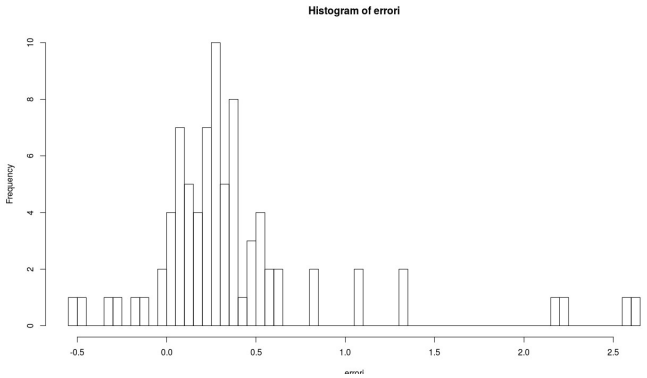

Figura 3.51: Istogramma degli errori con 20 utenti

concentrarci sulla previsione del solo mese di Novembre, in quanto dalle prove precedenti risulta più precisa rispetto alla previsione del mese di Dicembre.
Con pochi utenti otteniamo un quantile troppo grande, e per questo non riportiamo l'immagine. Andando a ingrandire il grafico si osserva che all'aumentare del numero di utenti l'errore medio si stabilizza intorno a un errore di 0.27, un errore incredibilmente più alto rispetto all'errore medio del terzo modello.

In base a queste semplici osservazioni possiamo preferire il terzo modello al quarto, e quindi scartare ques'ultimo modello.

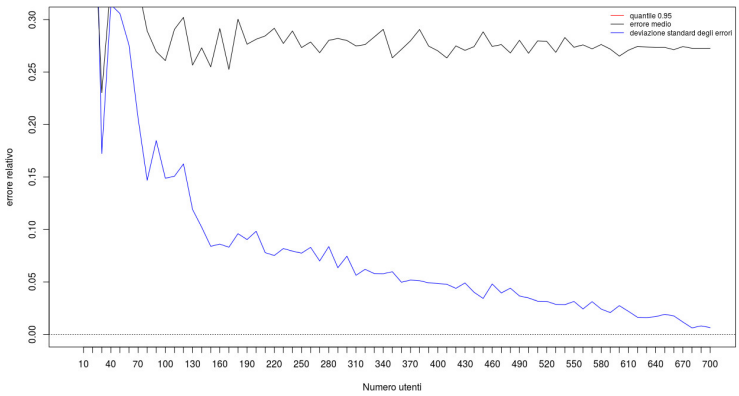

Figura 3.52: Risultati quarto modello

Capitolo 4

Conclusioni

In seguito alle analisi svolte nel capitolo precedente, cerchiamo di trarre delle conclusioni su quale sia il metodo migliore da applicare al nostro problema.

4.1 Scelta del metodo

Riassumiamo i risultati ottenuti in figura 4.1 e in figura 4.2 scaliamo tutti e quattro i metodi sulla stessa scala per fare un paragone migiore. Non riportiamo il metodo della regressione lineare che abbiamo escluso a priori. Dalle figure scalate si percepisce

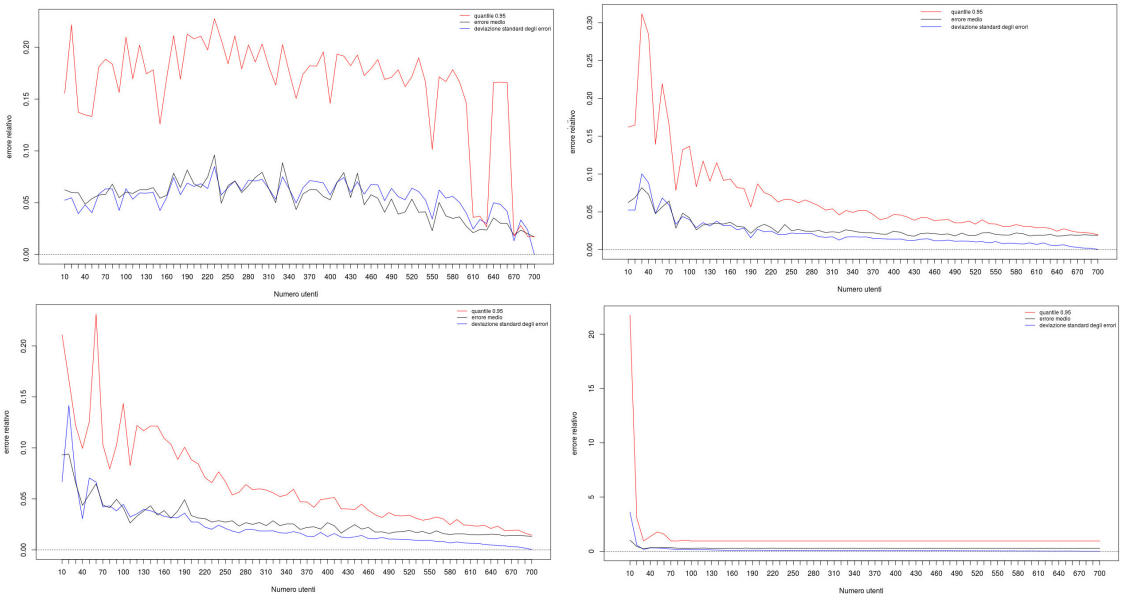

Figura 4.1: Risultati dei quattro modelli

immediatamente che il quarto modello risulta chiaramente il peggiore dei quattro modelli,

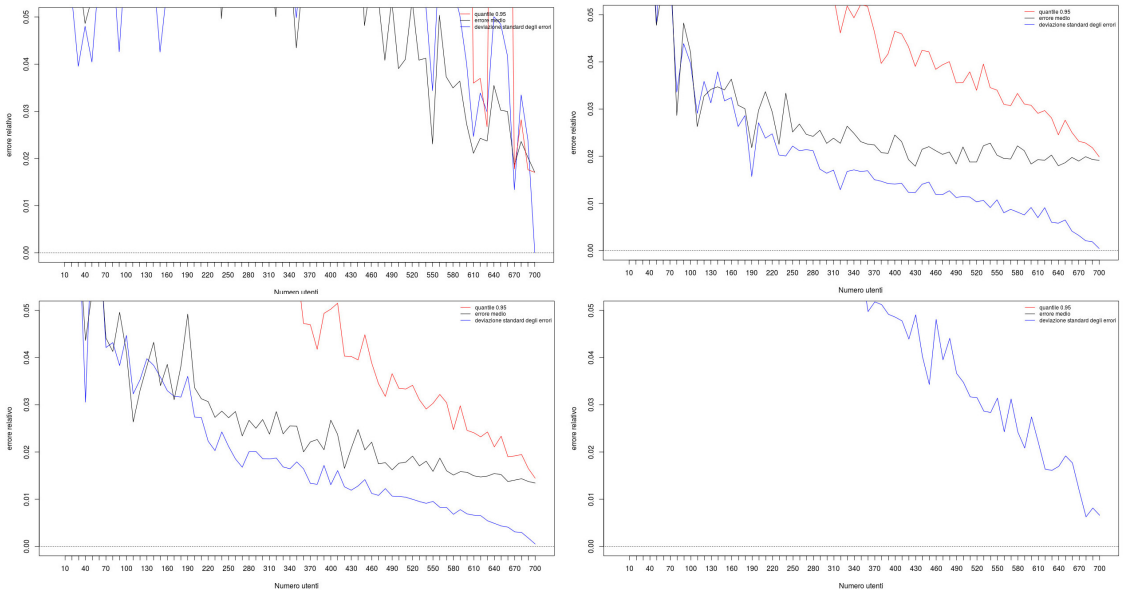

Figura 4.2: Risultati dei quattro modelli

e perciò è il primo che escludiamo. Il metodo PLS e il terzo modello hanno un andamento molto simile, e perciò giudichiamo migliore tra i due quello che commette un errore medio minore nella maggior parte dei casi. Come si può osservare dalla figura 4.3, il modello migliore in questo senso è il terzo modello, anche se le differenze risultano davvero minime. L'unica scelta rimasta da fare è tra il metodo PLS approssimato e il

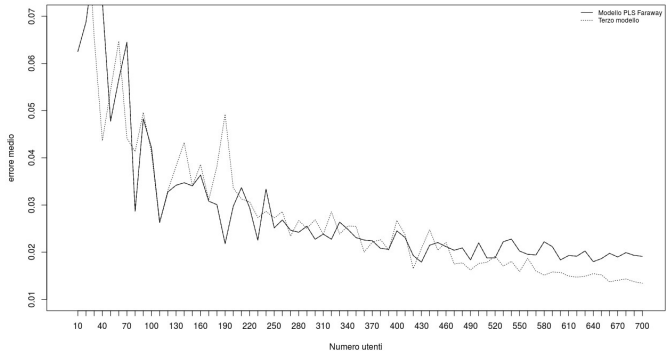

Figura 4.3: Errore medio primo e terzo modello

terzo modello. Riportiamo anche in questo caso gli errori medi dei due modelli al variare degli utenti. Riportiamo inoltre i grafici a confronto del quantile di livello 0.95 e della deviazione standard. Se andiamo a confrontare i vari grafici al variare del numero di utenti chiaramente il terzo modello risulta il migliore. La cosa che stupisce dal grafico

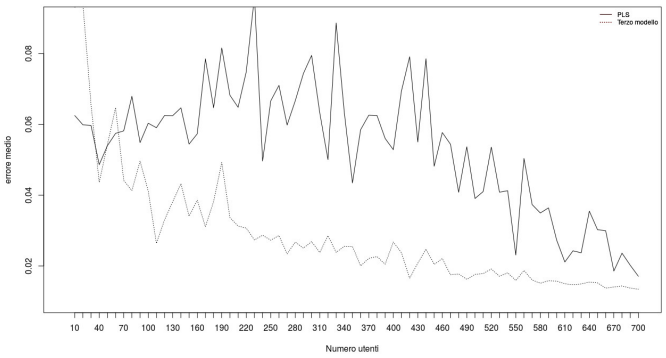

Figura 4.4: Errore medio PLS e terzo modello

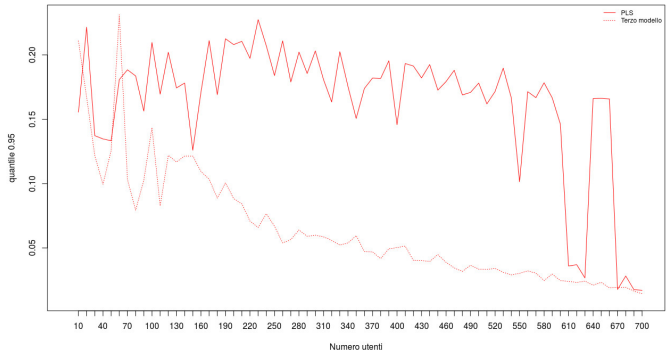

Figura 4.5: Quantile 0.95 PLS e terzo modello

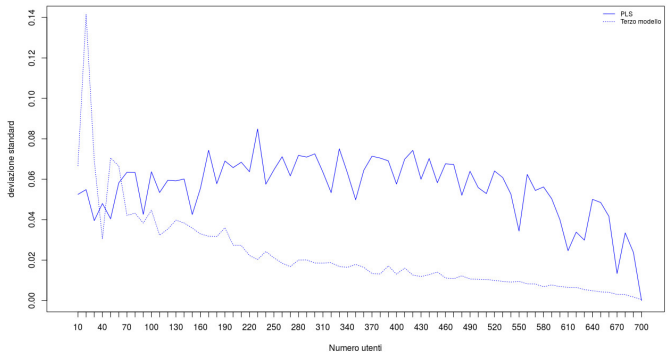

Figura 4.6: Deviazione standard PLS e terzo modello

dell'errore medio è che il metodo PLS approssimato con pochi utenti (ossia 10 o 20 utenti) risulta abbastanza buono. Scegliamo allora di fare ulteriori prove con il metodo PLS con 20 utenti e con il terzo modello con 400 utenti. Per il terzo modello scegliamo 400 utenti poichè il metodo dopo risulta abbastanza stabile al crescere degli utenti, come già sottolineato. Dunque i metodi che andremo a confrontare sono $PLSapp^{(20)}$

e $Terzomodello^{(400)}$, dove gli apici indicano il numero di utenti coinvolti. Facciamo di entrambi i metodi 10000 prove e concentriamoci sulla previsione del solo mese di Novembre 2013. Riportiamo i risultati ottenuti nella seguente tabella.

	$Errore_{medio}$	$Quantile_{0.95}$	Sd
$Primomodello^{(400)}$	0.021	0.046	0.013
$PLSapp^{(20)}$	0.052	0.145	0.046

Inoltre riportiamo anche gli istogrammi in figura 4.7 degli errori relativi al primo modello ed al modello PLS. Da queste osservazioni si vede che il terzo modello risulta chiaramente

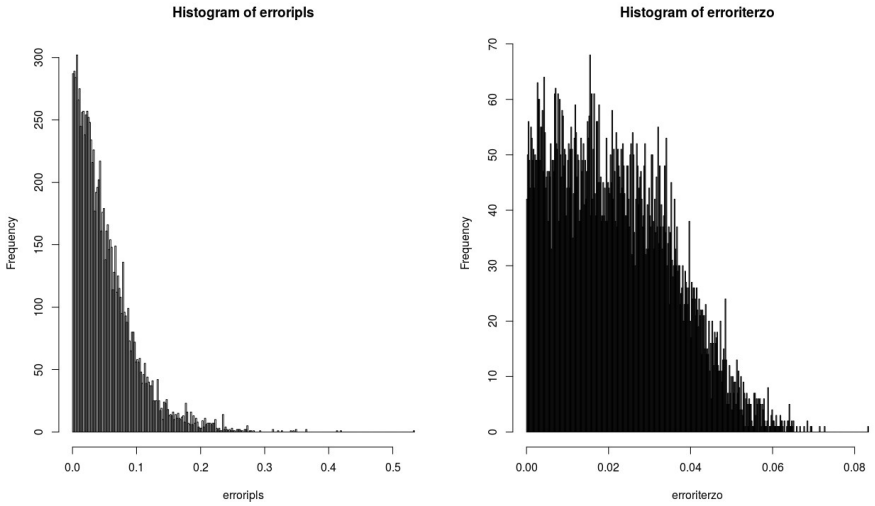

Figura 4.7: Istogrammi degli errori

migliore. A questo punto non ci sono altri termini di paragone tre i due modelli e la conclusione naturale è la seguente: se per l'azienda non rappresenta una spesa enorme andare a leggere 400 contatori, conviene applicare il terzo modello o alternativamente il metodo PLS. Se invece il costo per leggere 600 contatori risulta troppo alto, possiamo applicare il metodo PLS approssimato andando a leggere solamente 20 contatori.
Più precisamente, si potrebbe applicare la seguente strategia: andare a leggere una volta a settimana il contatore di 20 utenti scelti a caso tra i 701 con la telelettura e applicare il metodo PLS approssimato; in questo modo si avrebbe un'approssimazione abbastanza buona del risultato. Poi, per avere una precisione maggiore, si potrebbe andare a leggere il contatore di 400 utenti, sempre scelti in maniera casuale, ed affinare il risultato precedente andando ad applicare il terzo modello o analogamente il PLS.
Si potrebbero pensare altre correzioni banali del modello, dettate dal buon senso, come ad esempio escludere dai 700 utenti usati per costruire i vari modelli le utenze top.

Capitolo 5

Appendice

Riporto in questa sezione i vari codici usati in R, in riferimento al capitolo precedente.

5.1 Regressione lineare multipla

Riporto la funzione usata per l'implementazione del modello di regressione lineare multipla. In questo modello X rappresenta la matrice con i mesi da Gennaio 2011 a Dicembre 2013, contenente solo gli utenti che non hanno nessun valore mensile pari a 0. Tali utenti possono essere selezionati mediante questo algoritmo:

```
cont1 = 0
for(i in 1 : ncol(X)){if(0%in%X[, i])
                      {cont1 = cont1 + 1}
                      }
indici = numeric(cont1)
cont = 0
for(i in ncol(X)){
                      if(0%in%X[, i])
                      {cont = cont + 1
                      indici[cont] = i}}
X = X[, -indici]
```

Il vettore chiamato indici contiene gli utenti che hanno dei consumi mensili nulli. L'ultima riga del codice modifica la matrice X, selezionando solo gli utenti con tutti i consumi mensili strettamente positivi.

Dopo la modifica della matrice X riportiamo la funzione utilizzata per l'implementazione del primo modello. Il vettore *tot* contiene i consumi mensili totali del comune di Lucca, da Gennaio 2011 a Dicembre 2013. Questo vettore viene ulteriormente diviso nel vettore y e nel vettore *ytest*. Il vettore *ytest* contiene i dati del vettore *tot* dei mesi Novembre e Dicembre 2013, il vettore y i mesi rimanenti. La matrice $Xtest$ contiene i dati delle righe della matrice X corrispondenti alle righe di Novembre 2013 e Dicembre 2013. Tali dati saranno usati per testare il modello creato con i dati relativi ai mesi precedenti. La matrice $Xmod$ contiene i dati relativi ai mesi da Gennaio 2011 a Ottobre 2013.

$$
\begin{aligned}
&modellorlm <- function(nprove, nut)\\
&\{\\
&ystimato = matrix(data = 0, nrow = 2, ncol = nprove)\\
&for(j\ in\ 1 : nprove)\{\\
&\qquad utenti = sample(1 : ncol(Xmod), nut, replace = FALSE)\\
&\qquad model = lm(y \sim Xmod[, utenti] - 1)\\
&\qquad Xtestsel = Xtest[, utenti]\\
&\qquad coef = model\$coeff\\
&\qquad coef[is.na(coef)] = 0\\
&\qquad ystimato[, j] = Xtestsel\% * \%coef\\
&\qquad \}\\
&return(ystimato)\\
&\}
\end{aligned}
$$

Questa funzione restituisce il vettore *ystimato*, ossia il vettore che contiene i dati di Novembre e Dicembre 2013 stimati dal nostro modello. A questo punto avendo i dati effettivi dei mesi basterà fare la differenza per ottenere gli errori.
Riportiamo per completezza anche i codici relativi ai cicli fatti facendo variare gli utenti. Il parametro $freq$ indica con che frequenza andrà fatto variare il numero di utenti, ad esempio se $freq = 10$, verranno scelti prima 10 utenti, poi 20 e così via.

$$
\begin{aligned}
&ciclomodellorlm <- function(nprove, freq)\\
&K = seq(from = 0, to = ncol(Xmod), by = freq)\\
&Y = matrix(2, data = ytest, nrow = floor(ncol(Xmod)/freq), ncol = nprove * 2)\\
&errori = matrix(data = 0, nrow = floor(ncol(Xmod)/freq), ncol = nprove * 2)
\end{aligned}
$$

```
h = 1
for(k ∈ 1 : K){
        ystim = modellorlm(nprove, k)
        ystimvett = matrix(2, data = ystim, nrow = 1, ncol = nprove∗2)
        errori[h, ] = Y[h, ] − ystimatovett
        h = h + 1
        }
return(errori)
}
```

Il ciclo restituisce la matrice *ystimato* che in ogni riga contiene gli errori ottenuti con un numero di utenti fissati, e al variare delle righe cambia proprio il numero di utenti fissati. Una volta ottenuta questa matrice è facile calcolarsi l'errore medio, i quantili e la deviazione standard.

5.2 PLS

Usiamo le notazioni introdotte nella sezione precedente. Riportiamo la funzione per il metodo PLS.

```
modelloplsf <−function(nprove, nut)
{
ystimato = matrix(data = 0, nrow = 2, ncol = nprove)
for(j in 1 : nprove){
        utenti = sample(1 : ncol(Xmod), nut, replace = FALSE)
        Xsel = Xmod[, utenti]
        Xtestsel = Xtest[, utenti]
        a1 = numeric(nut)
        for(i in 1 : nut){
            a1[i] = crossprod(Xsel[, i], y)/crossprod(Xsel[, i], Xsel[, i])
                    }
```

$$nPXsel = sweep(Xsel, 2, a1, "*")$$

$$t1 = apply(nPXsel, 1, mean)$$

$$Xsel2 = Xsel$$

$$for(i\ in\ 1 : nut)\{$$

$$Xsel2[, i] = lm(Xsel[, i] \sim t1 - 1)\$res$$

$$\}$$

$$y2 = lm(y \sim t1 - 1)\$res$$

$$a2 = numeric(nut)$$

$$for(i\ in\ 1 : nut)\{$$

$$a2[i] = crossprod(Xsel2[, i], y)/crossprod(Xsel2[, i], Xsel2[, i])$$

$$\}$$

$$nPXsel2 = sweep(X2sel, 2, a2, "*")$$

$$t2 = apply(nPXsel2, 1, mean)$$

$$fpls = lm(y \sim t1 + t2 - 1)$$

$$t1pre = matrix(data = 0, nrow = 2, ncol = 1)$$

$$Xtestsel = Xtest[, utenti]$$

$$for(i\ in\ 1 : nut)\{$$

$$t1pre = t1pre + (a1[i] * Xtestsel[, i])/nut$$

$$\}$$

$$Xtest2 = Xtestsel$$

$$for(i\ in\ 1 : nut)\{$$

$$Xtest2[, i] = lm(Xtestsel[, i] \sim t1pre - 1)\$res$$

$$\}$$

$$t2pre = matrix(data = 0, nrow = 2, ncol = 1)$$

$$for(i\ in\ 1 : nut)\{$$

$$t2pre = t2pre + (b2[i] * X2test2[, i])/nut$$

```
                                    }
        ystimato[1,(2*j-1):(2*j)] = coef(fpls)[1]*t1pre+coef(fpls)[2]*t2pre
                                }
return(ystimato)
}
```

Anche in questo caso, il ciclo al variare degli utenti è una semplice modifica del codice precedente.

5.3 PLS approssimato

Riportiamo in questo paragrafo il codice usato per l'algoritmo PLS approssimato. Per usare i comandi descritti in seguito è necessario aver installato il pacchetto *pls* e richiamarlo con il comando *library(pls)*.

```
modellopls = function(nut, nprove){
                        ystimato = matrix(nrow = 1, ncol = 2 * nprove)
                        for(i in 1 : nprove){
                        utenti = sample(1 : ncol(Xmod), nut, replace = FALSE)
                         Xsel = Xmod[, utenti]
                        Xtestsel = Xtest[, utenti]
                         model = plsr(y ~ Xsel, ncomp = 2)
                ystimato[1,(2*i-1):(2*i)] = predict(model, Xtestsel, ncomp = 2)
                                        }
                        return(ystimato)
                                }
```

5.4 Terzo modello

Usando le notazioni del paragrafo precedente, riporto il codice usato per l'implementazione del primo modello.

```
terzomodello <- function(nprove, nut)
{
ystimato = matrix(data = 0, nrow = 2, ncol = nprove)
for(j in 1 : nprove){
                    utenti = sample(1 : ncol(Xmod), nut, replace = FALSE)
                    Xsel = Xmod[, utenti]
                    Xtestsel = Xtest[, utenti]
                    a = numeric(nut)
                    for(i in 1 : nut){
                                a[i] = sum(y)/sum(Xsel[, i])
                                ystimato[, j] = ystimato[, j]+a[i]*Xtestsel[, i]/nut
                                    }
                     }
return(ystimato)
}
```

Riportiamo per completezza anche i codici relativi ai cicli fatti facendo variare gli utenti. Il parametro $freq$ indica con che frequenza andrà fatto variare il numero di utenti, ad esempio se $freq = 10$, verranno scelti prima 10 utenti, poi 20 e così via.

```
cicloterzomodello <- function(nprove, freq)
K = seq(from = 0, to = ncol(Xmod), by = freq)
Y = matrix(2, data = ytest, nrow = floor(ncol(Xmod)/freq), ncol = nprove * 2)
errori = matrix(data = 0, nrow = floor(ncol(Xmod)/freq), ncol = nprove * 2)
for(k  in  1 : K){
                ystim = terzomodello(nprove, k)
                ystimvett = matrix(2, data = ystim, nrow = 1, ncol = nprove*2)
                errori[k, ] = Y[k, ] - ystimatovett
```

```
                    }
return(errori)
}
```

La matrice errori ha dunque tante colonne quante il numero di variazioni fatte al numero di utenti, e tante righe quante il doppio del numero di prove fatte. Nella riga k-esima contiene alternativamente gli errori commessi sulle previsioni di Novembre e Dicembre, al variare dei k utenti selezionati.

5.5 Quarto modello

Continuiamo a usare le notazioni dei precedenti modelli.

```
quartomodello <- function(nprove, nut)
{
ystimato = matrix(data = 0, nrow = 2, ncol = nprove)
for(j in 1 : nprove){
                        utenti = sample(1 : ncol(Xmod), nut, replace = FALSE)
                        Xsel = Xmod[, utenti]
                        Xtestsel = Xtest[, utenti]
                        a = numeric(nut)
                        for(i in 1 : nut){
                                a[i] = sum(y)/sum(Xsel[, i])
                                rho[i] = cor(Xsel[, i], y)
                                    }
                        r = sum(rho)
                        w = rho/r
                        for(i in 1 : nut){
                            ystimato[, j] = ystimato[, j]+(a[i]*w[i]*Xtestsel[, i])
                                    }
```

```
return(ystimato)
}
```

Non riporto il ciclo al variare di utenti, essendo una semplice modifica del ciclo per il primo modello.

Bibliografia

[1] A. Höskuldsson, *PLS Regression Methods*. Journal of Chemiometrics, 1998, 2, 211-228;

[2] P. Geladi e B. R. Kowalski, *Partial Least Squares Regression: a Tutorial*, Analytica Chimica Acta, 1986, 185, 1-17;

[3] Achiya Dax, *From Eigenvalues to Singular Values: A Review*, Advances in Pure Mathematics, 2013, 3, 8-24;

[4] Julian J.Faraway, *Practical Regression and Anova using R*, www.stat.lsa.umich.edu/~faraway/book, 2002;

[5] F. Flandoli, *Materiale didattico*, Università di Pisa, A.A.2014/2015, http://users.dma.unipi.it/ flandoli/dispense_StatII_2013_14.pdf;

[6] Bjørn-Helge Mevik e Ron Wehrens, *The pls Package: Principal Component and Partial Least Squares Regression in R*, Journal of Statistic Software, 2007, Vol.18, Issue 2;

[7] H. Abdì, *PLS-Regression*, Encyclopedia of Measurement and Statistic, 2007.

Ringraziamenti

Desidero ringraziare tutti coloro che mi hanno fornito il loro supporto nella stesura di questa tesi, e nel corso di questi cinque anni.
I ringraziamenti più speciali vanno al mio relatore, il professor Franco Flandoli, la persona più disponibile e geniale che io abbia mai incontrato. Inutile dire che senza il suo aiuto questo lavoro non avrebbe mai preso forma. Porterò sempre nel cuore i suoi insegnamenti ed i suoi consigli: essere professore vuol dire anche essere un esempio, ed io ho incontrato il miglior esempio che potessi desiderare.
Un ringraziamento di cuore anche al professor Maurizio Pratelli, che mi ha sempre saputo consigliare al meglio nel mio percorso.
Ringrazio inoltre la ditta Geal SpA, per avermi fornito il materiale sul quale lavorare. L'ambiente nel quale sono stata accolta mi ha fatto conoscere ed apprezzare il mondo del lavoro. In particolare ringrazio il mio tutor, Enrico Giorgi, che ha contribuito alla realizzazione del progetto.
Ringrazio la mia famiglia, che ha sempre creduto in me e non ha mai ostacolato nessun mio desiderio.
Ringrazio il mio Babi che mi ha accompagnato durante la mia crescita personale, con il più grande amore che potesse dimostrarmi.
Ringrazio la nonna Chica, che ha vissuto ogni mio esame con più ansia di me.
Ringrazio la mia zia Laura, che per cinque anni mi ha accolto nella sua casa facendomi sentire come a casa mia.
Ringrazio gli stupendi amici che ho conosciuto durante questo percorso, con i quali ho condiviso ogni momento di studio, di disperazione e di gioia.
Infine ringrazio le mie amiche, compagne di una vita.