**UNIVERSITÁ DEGLI STUDI DI MILANO-BICOCCA**

Scuola di Economia e Statistica

Corso di laurea in

SCIENZE STATISTICHE ED ECONOMICHE

****

**SHRINKAGE METHODS:**

**UN’APPLICAZIONE IN AMBITO MEDICO**

Relatore: Prof. Migliorati Sonia

Tesi di Laurea di:

Nardi Giorgio

Matr. N. 819961

Anno Accademico 2018/2019

Introduzione

Alla base di questo studio vi è l’analisi dei metodi moderni di selezione delle variabili e di regolarizzazione dei modelli statistici. In particolare, si pone l’attenzione sui metodi *Lasso* e *Ridge regression* andando a mettere in evidenza le differenze tra essi. Il metodo *Ridge regression*, anche noto come metodo di Tichonov-Miller, prevede una *residual sum of squares* caratterizzata da un termine di penalità che, conseguentemente alla soluzione del problema di ottimizzazione, genera una contrazione delle stime dei coefficienti verso lo zero rispetto al caso dei minimi quadrati, senza effettuare *feature selection*. Al contrario, la penalità che si considera utilizzando il metodo *Lasso*, introdotto da Robert Tibshirani nel 1996, spesso porta all’annullamento di alcuni dei coefficienti, permettendo di escludere delle variabili. Le motivazioni che mi hanno spinto ad approfondire questo argomento sono legate all’interesse per i modelli statistici che si è originato durante gli studi triennali. L’obiettivo di questa tesi di laurea è quello di fornire un’analisi accurata al fine di valorizzare i vantaggi che scaturiscono dall’utilizzo dei metodi di regolarizzazione rispetto al metodo dei minimi quadrati. Il capitolo 1 è dedicato alla descrizione del processo di stima dei parametri negli *Shrinkage models*. Nello specifico, nella prima parte del capitolo viene esaminato il metodo *Ridge regression*, mentre, successivamente, l’attenzione viene focalizzata sul *Lasso*, con approfondimento della rispettiva proprietà di *variable selection*. Infine, la parte conclusiva si sofferma interamente sulle procedure dedicate alla scelta ottimale del parametro di *tuning*. Il capitolo 2 riporta i risultati e le relative interpretazioni di un’applicazione svolta su un *set* di dati proveniente dall’ambito medico. I dati sono stati analizzati attraverso il *software* R: dopo una breve fase iniziale di pre-processing e analisi descrittiva si è proceduto con la stima dei diversi modelli con conseguenti confronti riguardo ai coefficienti stimati e alle performance predittive. Grazie a questo lavoro è stato possibile dimostrare che in determinati contesti vi è la possibilità di superare gli *OLS* in termini di efficienza e interpretabilità, e questo avviene con l’utilizzo degli *Shrinkage methods*. I risultati saranno esposti in maniera dettagliata nelle conclusioni finali dell’elaborato.

CAPITOLO 1

I metodi di selezione delle variabili possono anche non coinvolgere l’utilizzo dei minimi quadrati per stimare un modello lineare che contenga un sottoinsieme di predittori. In questo senso, si può stimare un modello usando una tecnica che regolarizza le stime dei coefficienti, o equivalentemente, che contrae le stime dei coefficienti verso lo zero. Generalmente questa tecnica può ridurre significativamente la varianza dei coefficienti stessi. Le 2 più note tecniche per la regolarizzazione delle stime sono *Ridge regression* e *Lasso.*

Ridge regression

La *Ridge regression* è molto simile alla regressione dei minimi quadrati eccetto che il processo di stima dei coefficienti avviene attraverso la minimizzazione di 2 quantità leggermente diverse mostrate nelle formulazioni 1.1 e 1.2.

|  |  |
| --- | --- |
| Minimi quadrati | Ridge regression |
| * 1. RSS=∑ni=1 (yi  - β0 - ∑pj=1βjxij)2 | * 1. PRSS=∑ni=1 (yi  - β0 - ∑pj=1βjxij)2 + λ∑pj=1 β2j |

Dove λ≥0 è un parametro di tuning che deve essere determinato separatamente e si può definire come parametro di complessità: maggiore è λ, maggiore è l’impatto del restringimento. Il termine della *penalized residual sum of squares (PRSS)* che si aggiunge alla *residual sum of squares* è chiamato penalità di *shrinkage* ed è tanto più piccolo quanto β1,β2,…,βp sono vicini allo zero, quindi ha l’effetto di contrarre le stime verso 0.

Risulta evidente che avere un **λ = 0** significa non avere una penalità nel modello, ovvero produrre le stesse stime che si otterrebbero con i minimi quadrati. In altro modo avere un **λ → ∞** (molto grande) significa avere un effetto di penalità elevato, che porta molti coefficienti ad essere prossimi a 0. Si può inoltre notare che aumentando il parametro di tuning (λ) si ha una minore flessibilità del modello, che comporta una varianza minore, ma una bias maggiore. Dunque, una scelta ottimale di λ permette di trovare il giusto compromesso tra bias e varianza. Se il numero di predittori è elevato ma minore della numerosità delle osservazioni (p < n) l’utilizzo degli OLS potrebbe riscontrare alcune difficoltà in seguito all’elevata variabilità. Se il numero di predittori è superiore alla numerosità (p > n) gli OLS non possono essere usati perché non c’è un’unica soluzione. In entrambi i casi, la *Ridge regression* può svolgere il suo compito. Inoltre, questo metodo non permette mai l’esclusione dei parametri stimati simili a 0 dal modello. Di conseguenza, un problema connesso a tale limite riguarda l’interpretabilità dei modelli, dato il possibile elevato numero di predittori inclusi.

Si può notare che la penalità è applicata a tutti i parametri, ma non all’intercetta β0. L’obiettivo è quello di regolarizzare l’associazione stimata di ogni variabile con la risposta; quindi l’intercetta non va considerata, poiché è semplicemente una misura della media della variabile risposta quando le altre variabili sono uguali a zero.

In generale la *PRSS* (*penalized residual sum of squares*) si può scrivere come:

(*Y*-*X*β)T (*Y*-*X*β) + λ||β||2 , ||β||2 =∑pj=1 βj 2

Poiché *PRSS* è una funzione convessa di β, essa ha un’unica soluzione. Per derivare lo stimatore *Ridge regression* bisogna risolvere il seguente problema di ottimizzazione convessa:

*min*β {(*Y-X* β)T (*Y-X* β) + λ|| β ||2 }= *min*β {PS(β)}

Dove *YTY* *– 2* β*TXTY* + β T*XTX* β + λ β T β è la *PRSS*. Risolvendo *=-2XTY +2XTX* β *+2λ*β*=0* rispetto a β si ottiene lo stimatore *Ridge regression*:

β ^RR *=*(*XTX +λIp*)*-1XTY*

In aggiunta, si può dimostrare che il termine di penalità della *Ridge regression* contrae gli autovalori della matrice dei dati *X*:

Se si considera la decomposizione in valori singolari della matrice di input *X* si ottiene *X=UΑVT*, dove *U(****nxp)***e *V****(pxp)*** sono matrici ortogonali e *Α*=(a1,a2,…,ap) con aj>0 j=1,…,p è la matrice degli autovalori corrispondente a *XTX*.

β ^LS*=(XTX)-1XTY*

*=(VAUTUAVT)-1VAUTY*

*=(VA2VT)-1VAUTY*

*=VA-2VTVAUTY*

*=V{(A2)-1A}UTY*

Si può ottenere anche:

β *^RR=(XTX +λIp)-1XTY*

*=(VAUTUAVT +λIp)-1VAUTY*

*=(VA2VT +λVVT)-1VAUTY*

*= V{A2 +λIp}-1 VTVAUTY*

*= V{(A2 +λIp)-1A}UTY*

È dunque evidente la differenza tra i due stimatori:

1. *A-2A* =*A-1*= *Diag*()
2. *(A2 +λIp)-1A* =*Diag*()

Siccome≤ il termine di penalità restringe gli autovalori.

Perché ridge regression migliora *OLS*?

Il vantaggio del metodo *Ridge regression* su *OLS* è racchiuso nel *trade off* *bias*-varianza. Nel momento in cui λ cresce, la flessibilità della stima *Ridge regression* diminuisce, portando a una diminuzione della varianza a spese di un incremento in *bias*. Ciò è mostrato nella Figura 1, che si basa su un *dataset* contenente 45 predittori e 50 osservazioni. La curva verde nel grafico mostra che la varianza dei valori fittati di *Ridge regression* è una funzione di λ. Con il metodo *OLS*, che corrisponde a una *Ridge regression* con λ=0, la varianza è alta ma non è presente *bias*, che è rappresentata in nero. Si può notare che all’aumentare di λ, la regolarizzazione delle stime dei coefficienti porta a una evidente riduzione nella varianza, a spesa di un leggero aumento di *bias*. Per valori di λ fino a circa 10, la varianza decresce rapidamente, con un incremento molto basso di *bias*. Di conseguenza, l’*MSE* si abbassa considerevolmente quando λ cresce da 0 a 10.

Per λ >10, la diminuzione della varianza rallenta, e la regolarizzazione sui coefficienti li rende significativamente sottostimati con un alto incremento di *bias*. L’*MSE* minimo è raggiunto approssimativamente con λ =30.

Nelle situazioni dove le relazioni tra la variabile risposta e i predittori è vicina alla linearità, le stime dei minimi quadrati avranno bassa *bias* ma potrebbero avere alta varianza. Questo vuol dire che un piccolo cambiamento nel *training dataset* può causare un grande cambiamento nelle stime dei coefficienti dei minimi quadrati. In particolare, quando il numero di variabili p è grande quasi quanto il numero di osservazioni n, le stime dei minimi quadrati avranno una varianza molto alta. Nel caso in cui p>n, le stime dei minimi quadrati non hanno un'unica soluzione, mentre la *Ridge regression* può ancora performare bene scambiando un piccolo aumento di bias per una alta diminuzione della varianza. La *Ridge regression* inoltre ha anche vantaggi computazionali sulla selezione del miglior sottoinsieme di variabili, che richiede la ricerca attraverso 2p modelli. Per ogni valore fissato di λ, con il metodo *Ridge regression* si può stimare un singolo modello, e la procedura di stima può avvenire abbastanza rapidamente. Infatti, i calcoli per stimare un modello con il metodo *Ridge regression*, usando simultaneamente tutti i valori di λ, sono quasi quantitativamente identici a quelli effettuati per stimare un modello usando *OLS*.

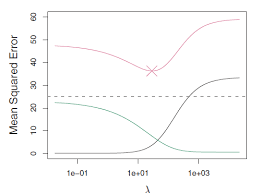


Figura 1(G.James, D. Witten, T. Hastie, R. Tibshirani. An introduction to statistical learning,Springer)

Si può inoltre dimostrare che lo stimatore *Ridge regression* è proporzionale allo stimatore OLS. In caso di ortonormalità si ha *XTX=Ip*e lo stimatore *Ridge regression* diventa:

β^RR *=(XTX +λIp)-1XTY*

*=(1+λ)-1Ip XTY*

*=(1+λ)-1(XTX)XTY*

= β ^LS

I primi due momenti dello stimatore Ridge regression

Lo stimatore *Ridge regression* è distorto:

E(β ^RR)= *E*[*(XTX +λIp)-1XTY*]

= *(XTX +λIp)-1XTE[Y]*

*=* *(XTX +λIp)-1 (XTX)* β

È evidente che E(β ^RR) ≠ β per ogni λ>0. In aggiunta il valore atteso dello stimatore *Ridge regression* è uguale a 0 nel momento in cui il parametro λ tende a infinito:

=*(XTX +λIp)-1 (XTX)* β]=0p

Dunque, tutti i coefficienti vengono ristretti verso zero al crescere del parametro di complessità. Questa proprietà si mantiene anche quando nella matrice *X* il numero dei predittori è maggiore del numero delle osservazioni. Tuttavia, tale comportamento non è strettamente monotono infatti λa >λbnon implica necessariamente |βj^RR(λa)| <|βj^RR(λb)|.

Per focalizzarsi sulla varianza di β ***^RR*** bisogna introdurre:

*Wλ=(XTX +λIp)-1XTX*

È possibile quindi esprimere lo stimatore *Ridge regression* in questo modo:

*Wλ* β *^LS = Wλ(XTX )-1XT**Y*

*=[(XTX )-1(XTX +λIp)]-1(XTX )-1XTY*

*=(XTX +λIp)-1 XTX(XTX )-1 XTY*

*=(XTX +λIp)-1 XTY*

*=* β *^RR*

Quindi *Wλ*trasforma lo stimatore dei coefficienti della regressione, ottenuto con la massima verosimiglianza, nello stimatore *Ridge regression*. Ora si può mostrare facilmente:

*Var*[β ^RR] *=Var*[*Wλ* β*^LS*]

*=Wλ Var*[β*^LS*] *WλT*

*=σ2 Wλ(XTX )-1 WλT*

*=σ2(XTX +λIp)-1 XTX[(XTX +λIp)-1]T*

In cui è stata utilizzata la proprietà delle matrici *Var(AY)=AVar(Y)AT* e la formula della matrice varianze e covarianze *Var*[β *^LS*]*= σ2(XTX )-1*.

Come avviene per il valore atteso anche la varianza tende a zero per **λ → ∞:**

*Var*[β *^RR*] *= σ2 Wλ(XTX )-1 WλT= 0p*

Questo vuol dire che all’aumentare del parametro di *shrinkage* la varianza delle stime dei coefficienti di *Ridge regression* diminuisce. A questo punto è possibile confrontare la varianza dello stimatore *Ridge regression* con la varianza dello stimatore dei minimi quadrati:

*Var*[β *^LS*] *- Var*[β *^RR*] *= σ2 [(XTX )-1- Wλ(XTX )-1 WλT]*

*= σ2 Wλ{[I+λ(XTX)-1] (XTX)-1[I+λ(XTX)-1]T -(XTX)-1} WλT*

*= σ2 Wλ[2λ(XTX)-2+λ2(XTX)-3] WλT*

*= σ2(XTX +λIp)-1[2λIp+λ2(XTX)-1]{[ XTX +λIp]-1}T*

La differenza è non negativa, quindi la varianza dello stimatore di massima verosimiglianza supera quella dello stimatore *Ridge regression*:

*Var*[β^RR] ≥ *Var*[β^LS]

Dove se λ >0 si ha *Var*[β^RR] > *Var*[β^LS]

**Gradi di libertà con Ridge regression**

**Conoscere i gradi di libertà è utile perché può essere vantaggioso nel processo di decisione del parametro di complessità. Dalla regressione lineare si ha:**

***Y^= X(XTX)-1XTY= HY***

**Dove *H* è la *hat matrix.* Il numero di gradi di libertà usati nella regressione dei minimi quadrati è uguale a *tr(H)*. In particolare, se *X* è di rango pieno, *rango(X)=p* e *tr(H)=p*.**

**Segue che la versione della *hat matrix* nella *Ridge regression* è:**

***H(λ) =X(XTX+λIp)-1XT***

**I gradi di libertà della *Ridge regression* sono dati dalla traccia della *Ridge hat matrix* *H(λ)*:**

***tr[H(λ)]=tr[X(XTX+λIp)-1XT]=*****

**Inoltre il numero di gradi di libertà diminuisce al crescere di λ. In particolare:**

******

Lasso

|  |  |
| --- | --- |
| Lasso | Ridge regression |
| * 1. Minβ ∑ni=1 (yi  - β0 - ∑pj=1Bjxij)2 + λ∑pj=1|βj | | * 1. Minβ ∑ni=1 (yi  - β0 - ∑pj=1Bjxij)2 + λ∑pj=1 β2j |

Il metodo *Ridge regression* ha un limite evidente: a differenza di selezione *forward* e *backward* che selezionano i modelli individuando un sottoinsieme di variabili da includere, esso comprende tutti i p predittori nel modello finale. La penalità di *shrinkage* forza tutti i coefficienti verso 0, ma nessuno di essi è esattamente uguale a 0. Questo può non essere un problema per l’accuratezza della previsione, ma può creare difficoltà nell'interpretazione del modello in situazioni in cui il numero di variabili p è abbastanza grande. Incrementare il valore di λ tenderà a ridurre l’impatto dei coefficienti, ma non vi sarà l’esclusione di alcuna variabile. *Lasso* (*least absolute shrinkage and selector operator*) è una alternativa relativamente recente al metodo *Ridge regression*. *Lasso* e *Ridge regression* si approcciano alla procedura di stima cercando di minimizzare quantità simili, mostrate nelle formulazioni 1.3 e 1.4, ciò che varia è il termine di penalità. Nel caso di *Lasso,* la penalità ha l’effetto di forzare alcuni coefficienti a essere esattamente uguali a 0 quando il prametro di *tuning* λ è sufficientemente elevato. Quindi *Lasso* effettua *variable selection*. Per questo, si può constatare come sia più facile interpretare i modelli generati con metodo *Lasso* piuttosto che con *Ridge regression*. Come con la *Ridge regression* scegliere un buon valore di λ per *Lasso* è un passaggio fondamentale. Quando λ=0 *Lasso* restituisce le stime dei minimi quadrati, e quando λ diventa sufficientemente grande *Lasso* restituisce il modello nullo nel quale le stime dei coefficienti sono tutte uguali a 0. Quindi, in conclusione, in relazione al valore di λ *Lasso* può produrre un modello includendo un numero qualsiasi di variabili. Per quanto riguarda invece *Ridge regression*, essa includerà sempre tutti predittori nel modello, sebbene l’impatto dei coefficienti dipenderà da λ. Il metodo *Lasso* fu proposto da Tibshirani nel 1996, egli suggerì di utilizzare un vincolo che imponesse il valore assoluto a ciascun parametro e definì la *PRSS* come:

(*Y*-*X* β)T(*Y*-X β) +λ|| β||T1P, || β||T1P=∑pj=1 | βj |

Dove || β||=(| β1|,…,| βp|)T e *1p* è un vettore di 1.

Lo stimatore *Lasso* si ottiene dunque risolvendo il seguente problema di ottimizzazione convessa:

*minB(Y-X* β*)T(Y-X* β*) +λ*|| β||*T1P*

Un’altra formulazione per Lasso e Ridge regression

È utile introdurre una forma alternativa dei problemi di minimizzazione rispettivamente di *Lasso* e *Ridge regression*.

|  |  |
| --- | --- |
| Lasso | Ridge regression |
| * 1. Minβ {∑ni=1 (yi  - β0 - ∑pj=1βjxij)2 } sotto il vincolo:   ∑pj=1 |βj |≤ s | * 1. MinB {∑ni=1 (yi  - β0 - ∑pj=1βjxij)2 } sotto il vincolo:   ∑pj=1 βj 2≤ s |

Per qualsiasi valore di λ c’è qualche valore s tale che le formule 1.3 e 1.5 restituiscono le stesse stime dei coefficienti, usando *Lasso*. Allo stesso modo, per qualsiasi valore di λ c’è un valore corrispondente di s tale che 1.4 e 1.6 portano alle stesse stime dei coefficienti di *Ridge regression*. Nel semplice caso in cui p=2, la 1.5 indica che le stime dei coefficienti *Lasso* hanno la più piccola *RSS* al di fuori del rombo definito dalla disequazione |β1 |+ |β2 |≤ s. Allo stesso modo, le stime *Ridge regression* hanno la più piccola *RSS* al di fuori del cerchio definito da β12 + β22 ≤ s. Quando si effettua *Lasso* l’obiettivo è trovare l’insieme di stime dei coefficienti che portano alla *RSS* più piccola, considerando però un vincolo definito da un *budget* s che indica quanto può essere grande ∑pj=1 | βj |. Quando s è estremamente grande, questo *budget* non è così restrittivo, quindi le stime dei coefficienti possono essere più elevate. Infatti, se s è grande abbastanza tale che la soluzione dei minimi quadrati rientra nel *budget*, 1.5 porterà semplicemente alle stime *OLS*. D’altra parte, se s è piccolo, allora ∑pj=1 | βj | dovrà essere piccola per evitare di superare tale *budget*.

La 1.6 indica che quando si utilizza il metodo *Ridge regression*, l’obiettivo è quello di trovare un insieme di stime dei coefficienti tale che la *RSS* sia la più piccola possibile, considerando il vincolo che prevede che ∑pj=1 βj 2non debba superare il *budget* s.

Le formulazioni 1.5 e 1.6 rivelano una stretta connessione tra *Lasso*, *Ridge regression* e la miglior selezione del sottoinsieme di predittori. Si consideri il problema:

* 1. *minB {∑ni=1 (yi  -* β0 *- ∑pj=1* βj*xij)2 }*sotto il vincolo: *∑pj=1 I(*βj ≠0*)≤ s*

Dove I è la funzione indicatrice: assume valore 1 se βj≠0 e 0 altrimenti. Quindi risolvere 1.7 equivale a trovare un insieme di stime dei coefficienti tali che *RSS* sia la più piccola possibile, soggetta al vincolo che non più di s coefficienti possono essere diversi da 0. La soluzione del problema 1.7 permette di giungere alla migliore selezione del sottoinsieme di predittori. Sfortunatamente, risolvere 1.7 è computazionalmente infattibile quando p è grande, dal momento che richiede tutti imodelli contenenti s predittori. Perciò, *Ridge regression* e *Lasso*, computazionalmente meno complessi, sono delle valide alternative alla selezione del miglior sottoinsieme di predittori e sostituiscono l’intrattabile forma del budget in 1.7 con vincoli che sono più facili da utilizzare. Ovviamente, *Lasso* è molto più connessa con la selezione del miglior sottoinsieme di variabili, dal momento che con tale metodo si effettua *feature selection* per s sufficientemente piccolo.

Gradi di libertà con Lasso

Tibshirani(1996) propose di riscrivere il termine di penalità presente in *Lasso* come:

∑pj=1 | β j | *=*

Questo permette di giungere ad una approssimazione tramite *Ridge regression* in modo da riuscire ad approssimare i gradi di libertà di *Lasso*. La stima dei gradi di libertà coincide con il numero di coefficienti diversi da zero:

*df(λ)=|C(λ)|*

# dove *C(λ)={j:* βj^*(λ)≠0*} è l’insieme dei coefficienti diversi da zero e *|C(λ)|* è la sua cardinalità. Inoltre, se y si distribuisce come una normale con media μ e varianza σ2I

e il rango(*X*)=p tale stima è consistente e non distorta. Nel 2012 fu provato da Tibshirani, Taylor e Dossal che l’ipotesi di rango pieno della matrice *X* non era strettamente necessaria, infatti tale stima può essere usata anche con p>n.

La proprietà di *variable selection* di Lasso

Perché usando il metodo *Lasso* alcune stime dei coefficienti risultano essere esattamente uguali a zero? La figura 2 illustra la situazione. La soluzione dei minimi quadrati è denotata con β^, mentre il rombo e il cerchio azzurri rappresentano i vincoli di *Lasso* e *Ridge regression* rispettivamente in 1.5 e 1.6. Se s è sufficientemente grande, le regioni di vincolo conterranno β^, e quindi le stime di *Lasso* e *Ridge regression* saranno uguali alle stime dei minimi quadrati. Ad ogni modo, nella Figura 2, dove si considera un numero p di predittori uguale a 2, le stime dei minimi quadrati si trovano al di fuori del rombo e del cerchio, quindi i coefficienti ottenuti con i metodi di regolarizzazione sono diversi da quelli dei minimi quadrati. Le ellissi centrate intorno a β^ rappresentano le regioni di *RSS* costante. In altre parole, tutti i punti su una data ellisse condividono un valore comune di *RSS*. Man mano che le ellissi si espandono lontano dalle stime *OLS* la *RSS* aumenta. Le equazioni 1.5 e 1.6 indicano che le stime ottenute con *Lasso* e *Ridge regression* sono date dal primo punto nel quale un’ellisse interseca la regione di vincolo. Dal momento che la *Ridge regression* ha una regione di vincolo circolare, senza vertici , questa intersezione generalmente non si troverà su un asse, e quindi le stime dei coefficienti di *Ridge regression* saranno diverse da zero. D’altra parte, la regione di vincolo di *Lasso* ha i vertici sugli assi, quindi si potrà notare di frequente l’intersezione di tale regione con l’ellisse su un asse. Quando questo accade, uno dei coefficienti è uguale a 0. In più dimensioni più stime dei coefficienti possono essere uguali a zero contemporaneamente.

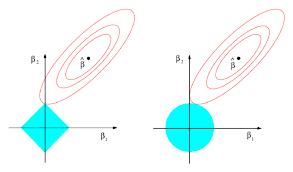


Figura 2(G.James, D. Witten, T. Hastie, R. Tibshirani. An introduction to statistical learning,Springer

Lasso standard errors

Essendo la stima *Lasso* una funzione non lineare e non differenziabile dei valori di risposta, anche per un valore di s fissato è difficile ottenere una stima accurata del suo *standard error*. Un possibile approccio è attraverso il *bootstrap*.

Si può derivare una stima approssimativa considerando il termine ∑pj=1 |β j | come ∑pj=1  . Quindi se si considera una *Ridge regression* con stimatore in forma

β*\*=(XTX +λW-)-1XTY*

dove *W*è la matrice diagonale con gli elementi | β j| sulla diagonale, *W-*indica la matrice inversa di *W* e λ è scelto in modo tale che ∑pj=1 | β j |\*=s. la matrice di varianza covarianza delle stime può essere approssimata con:

*(XTX +λW-)-1XTX(XTX +λX-)-1σ^2*

Dove *σ^2*è una stima della varianza dell’errore. Un problema evidente nell’utilizzo di questa formula risiede nel fatto che attribuisce una varianza stimata uguale a 0 per i predittori con βj^=0. Ad ogni modo, essa si rivela utile per la selezione del parametro s.

Lasso e Ridge regression a confronto

È evidente che *Lasso* presenta un notevole vantaggio rispetto alla *Ridge regression*, infatti produce modelli più semplici e interpretabili che coinvolgono solo un sottoinsieme di predittori. La Figura 3 mostra la varianza, *bias* al quadrato e il *test* *MSE* di *Lasso* applicato agli stessi dati simulati della Figura 1. Chiaramente *Lasso* mostra un comportamento qualitativamente simile alla *Ridge regression*, nel momento in cui λ aumenta, la varianza decresce e *bias* aumenta. Nel grafico di destra della Figura 3 le linee tratteggiate rappresentano i *fit* della *Ridge regression*. Qui vengono rappresentati con R2 sui dati di *training*. In questo esempio *Lasso* e *Ridge regression* producono distorsioni quasi identiche. In ogni caso, la varianza di *Ridge regression* è leggermente più bassa rispetto alla varianza di *Lasso*. Di conseguenza, l’*MSE* minimo di *Ridge regression* è leggermente più piccolo di quello di *Lasso*. I dati utilizzati nella Figura 3 sono stati generati in modo tale che tutti i 45 predittori fossero relazionati con la variabile risposta ma *Lasso* implicitamente assume che alcuni coefficienti siano esattamente uguali a 0. Di conseguenza, non è una sorpresa che *Ridge regression* performi meglio di *Lasso* in termini di errore di previsione in questa situazione. Nel caso più semplice in cui la variabile risposta è funzione di solo 2 predittori *Lasso* tende a performare meglio di *Ridge regression* in termini di bias, varianza, e *MSE*.

Questi 2 esempi illustrano che né *Ridge regression* né *Lasso* prevale universalmente sull’altro metodo. In generale, ci si può aspettare che *Lasso* performi meglio in una situazione dove c’è un numero relativamente piccolo di predittori, che hanno coefficienti significativi, e i predittori rimanenti hanno coefficienti che sono molto piccoli o che sono uguali a zero. *Ridge regression* performerà meglio quando la risposta è una funzione di tanti predittori, tutti con coefficienti di dimensione approssimativamente uguale. In ogni caso, il numero di predittori che ha relazioni con la variabile risposta non è mai conosciuto a priori nei *dataset*. Una tecnica come la *cross-validation* può essere usata per determinare quale approccio è meglio su un particolare *dataset*. Come con il metodo *Ridge regression*, quando le stime dei minimi quadrati hanno varianza eccessivamente alta, la soluzione con *Lasso* può portare a una riduzione nella varianza a spesa di un piccolo incremento di bias, e conseguentemente può generare predizioni più accurate.

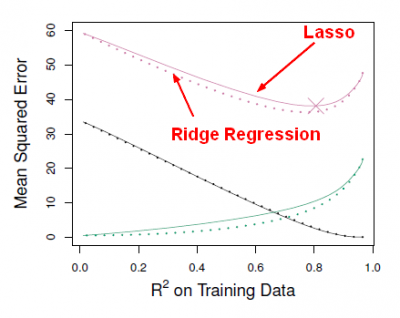
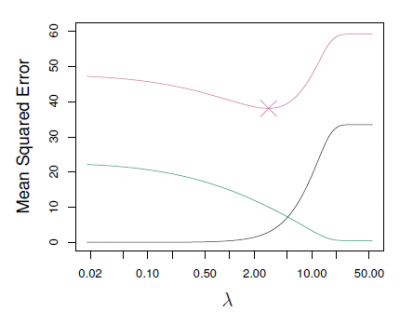


Figura 3(G.James, D. Witten, T. Hastie, R. Tibshirani. An introduction to statistical learning,Springer)

Ridge regression e Lasso da un punto di vista bayesiano

È possibile fornire una interpretazione bayesiana dei due metodi. Si supponga che il vettore dei coefficienti β abbia una certa distribuzione a priori, p(β), dove β =( β0, β1,…, βp)T. È utile inoltre indicare la funzione di verosimiglianza dei dati con f(Y|X,β), dove X=(X1,…,Xp). Procedendo con la moltiplicazione della distribuzione a priori con la funzione di verosimiglianza si ottiene la distribuzione a posteriori:

*p(*β*|X,Y)αf(Y|X,*β*)p(*β*|X)=f(Y|X,*β*)p(*β*)*

Si prenda in considerazione il modello lineare:

Y=β0+X1 β1+…+Xp βp+ ε

Si assuma che gli errori siano indipendenti e provenienti da una distribuzione normale. In aggiunta, si consideri p(β)=∏j=1p g(βj), dove g è una funzione di densità. Si può dunque constatare che *Ridge regression* e *Lasso* derivano da due particolari casi della funzione g:

* Se g è una normale avente media nulla e come varianza una funzione del parametro di complessità λ, segue che la moda a posteriori, che coincide con il più probabile valore di β, è data dalla soluzione del metodo *Ridge regression*.
* Se g è una doppia esponenziale caratterizzata da media nulla e parametro di scala corrispondente a una funzione di λ, segue che la moda a posteriori per β è la soluzione del metodo *Lasso*.

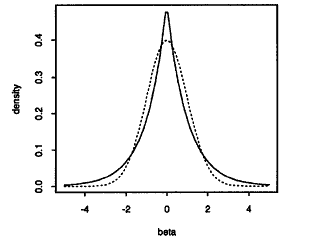


Figura 4(Article: Regression Shrinkage and Selection via the Lasso, by Robert Tibshirani, Journal of the Royal Statistical Society)

Scelta del parametro di complessità

Utilizzare i metodi descritti richiede una selezione appropriata del parametro di *tuning* λ, o equivalentemente del vincolo nelle formulazioni 1.5 e 1.6. Il valore di λ infatti deve essere scelto dal ricercatore e vi sono delle procedure o criteri che facilitano la decisione.

Information criterion

Una strategia molto conosciuta è quella di scegliere un parametro di complessità che porti a un buon modello ma con l’inclusione di poche variabili esplicative. I criteri d’informazione mirano a un equilibrio tra l’adeguatezza del modello e la sua complessità. A tal proposito, un criterio molto utilizzato è quello dell’*AIC* ( Aikaike’s information criterion ): esso fornisce una misura della qualità della stima di un modello statistico tenendo conto sia della bontà di adattamento che della complessità del modello. Ad esempio, con *Ridge regression* si ha:

*AIC(λ) = 2 p − 2 log(L)*

*= 2 tr[H(λ)] − 2 log{L[*β*ˆ(λ), σˆ 2 (λ)]}*

*= 2+ 2nlog[σˆ(λ)] +*

Dove *L* è il valore massimizzato della funzione di verosimiglianza del modello stimato. Il valore di λ che minimizza l’*AIC* corrisponde al perfetto bilanciamento tra complessità del modello e *overfitting*. Con il metodo *Lasso* la procedura di scelta del parametro λ avviene nello stesso modo, ma l’AIC assume una formulazione differente:

*AIC(λ) =*

Sono numerosi gli *information criteria* che possono essere utilizzati, il più comune dopo l’*AIC* è il *ΒIC* (*Βayesian information criterion*) o il *modified ΒIC*. In generale, questi criteri guidano a una decisione nel caso in cui si considerano differenti modelli, i quali usano diversi insiemi di variabili esplicative per spiegare la risposta. In questo senso l’uso di *information criteria* per la scelta del parametro di *tuning* con *Ridge regression* può essere considerato inappropriato, infatti, tale regressione utilizza lo stesso insieme di variabili esplicative indipendentemente dal valore di λ perché non effettua *variable selection*. Si può dunque concludere che l’uso di tali procedure è sicuramente più efficiente con il metodo *Lasso*.

Cross-validation

La tecninca *cross-validation* punta a produrre un modello con buone prestazioni di previsione. Comunemente la performance viene valutata su dei nuovi dati che vengono prodotti direttamente con i dati a portata di mano. A tal proposito, il *dataset* viene diviso in due: il *training set*, che gioca il ruolo di *dataset* originario dove il modello viene costruito, e il *test set*, che contiene i nuovi dati su cui valutare la performance di previsione del modello. Questa procedura viene svolta per una serie di possibili valori del parametro di complessità e quello che porta al modello con la miglior qualità in termini di *performance* predittiva è da preferire agli altri. Bisogna però tenere in considerazione che effettuando una sola volta la suddivisione del dataset la valutazione delle prestazioni dipende dai set di dati ottenuti. Per annullare questo effetto la suddivisione dei dati originali in *training set* e *test set* viene esercitata molte volte in modo da creare ogni volta insiemi di osservazioni differenti. Successivamente, per ogni suddivisione vengono stimati i parametri del modello per tutti i possibili valori di λ per poi passare alla fase di valutazione sul *test set* corrispondente. Il parametro di *tuning* che, in media, presenta le prestazioni migliori viene selezionato.

Quando il frazionamento ripetitivo dei dati viene fatto casualmente, dei campioni possono nella maggior parte dei *training set* o equivalentemente dei *test set*. Tali campioni quindi potrebbero avere un effetto di sbilanciamento sulla costruzione del modello o sulla valutazione della performance. Per evitare ciò *k-fold cross validation* struttura il frazionamento dei dati e i campioni vengono divisi in k sottoinsiemi esaustivi e mutualmente esclusivi di pari dimensioni. A turno, ognuno di questi sottoinsiemi svolge il ruolo del *test set* mentre il *training set* è costituito dall’unione dei sottoinsiemi rimanenti.

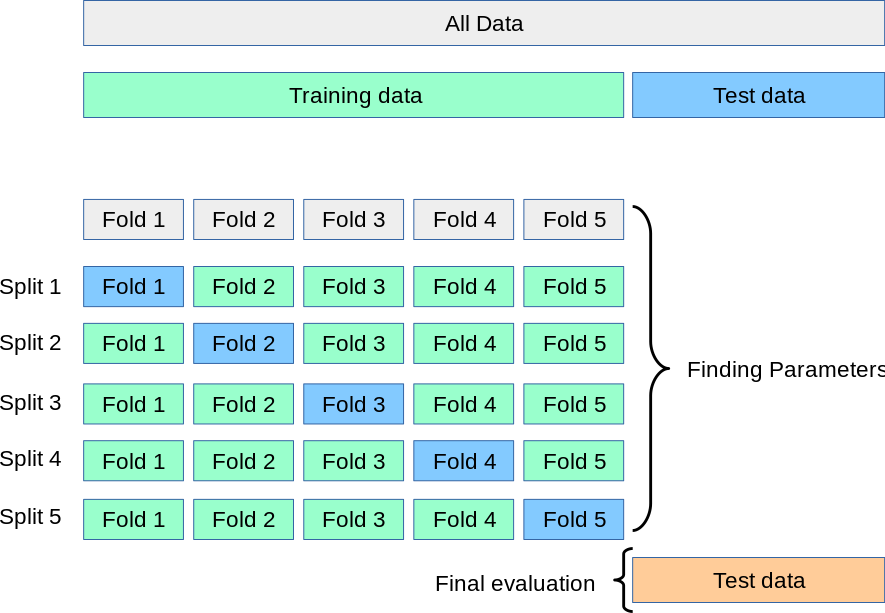


Figura 5(scikit-learn.org)

Quest’ultima procedura garantisce una rappresentazione bilanciata di ciascun campione in *training set* e *test set*, tuttavia la divisione in k sottoinsiemi mantiene un certo grado di casualità che può essere del tutto esclusa scegliendo k=n. Questo caso particolare è denominato *leave-one-out cross-validation(LOOCV)*. Considerando il metodo *Ridge regression* si può illustrare secondo i seguenti passi:

1. Definire un intervallo d’interesse per il parametro di complessità
2. Dividere il *dataset* in *training set e test set* costituiti rispettivamente da {1, ..., n} \i e {i}
3. Stimare il modello di *Ridge regression* per ciascun λ presente nella griglia di valori d’interesse:

******

1. Valutare la performance di previsione dei modelli con *log{L[Yi, Xi,\*;* βˆ−i(λ), *σˆ2−i(λ)]}* oppure con l’errore di previsione elevato al quadrato [*Yi - Xi*,\*; βˆ−i(λ)]2
2. Ripetere i passi 1 e 3 fino al punto in cui ogni campione svolge il ruolo del *test set* una volta
3. La media delle *performance* predittive per ciascun valore di λ presente nella griglia è:



La quantità sopra espressa è chiamata *cross-validated* log-verosimiglianza.

1. Il valore del parametro di complessità che massimizza la *cross-validated* log-verosimiglianza è il valore de selezionare.

CAPITOLO 2

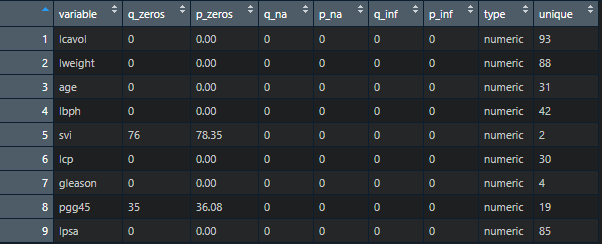
Applicazione

Il *dataset* è costituito da osservazioni provenienti da uno studio che ha esaminato la correlazione tra il livello di antigene prostatico specifico e una serie di misure cliniche negli uomini che stavano per ricevere una prostatectomia radicale. Il *dataset* è stato ricavato dalla libreria lasso2 di R ed è composto da 97 osservazioni e 9 variabili:

* *lcavol*: logaritmo in base 10 del volume del cancro
* *lweight*: logaritmo in base 10 del peso della prostata
* *age*: età dell’individuo
* *lbph*: logaritmo in base 10 della quantità di iperplasia prostatica benigna, riferita all’ingrossamento benigno della ghiandola prostatica
* *svi*: variabile che identifica se il cancro ha invaso le vescicole seminali dell’individuo
* *lcp*: logaritmo in base 10 della quantità di invasione della capsula. Si riferisce a quanto il tumore si è approfondito nella capsula che lo delimita dal tessuto sano.
* *gleason*: punteggio Gleason. Si basa su una scala che va da 2 a 10 e determina il grado di aggressività del tumore.
* *pgg45*: percentuale punteggi di Gleason 4 o 5
* *lpsa*: logaritmo in base 10 dell’antigene prostatico specifico

La variabile *target* è *lcavol*. E’ utile osservare la tipologia delle variabili nella Tabella 1.

Tabella 1



Non sono presenti valori mancanti. Le tipologie di alcune variabili non sono adatte per l’analisi, occorre quindi ricodificarle. In particolare:

* *age* in una variabile *factor* che identifica se il paziente è vecchio o giovane prendendo come valore di soglia i 62 anni
* *svi* in *factor*

Analisi descrittive e pre-processing

Prima di cominciare con l’elaborazione dei modelli è bene riassumere attraverso grafici e tabelle le caratteristiche delle variabili presenti. Nel Grafico 1 sono rappresentati i *boxplot* riferiti a tutte le variabili quantitative, essi danno un’idea generale della dispersione e della asimmetria delle distribuzioni. In particolare, la variabile *target* *lcavol* sembra essere caratterizzata da distribuzione simmetrica.

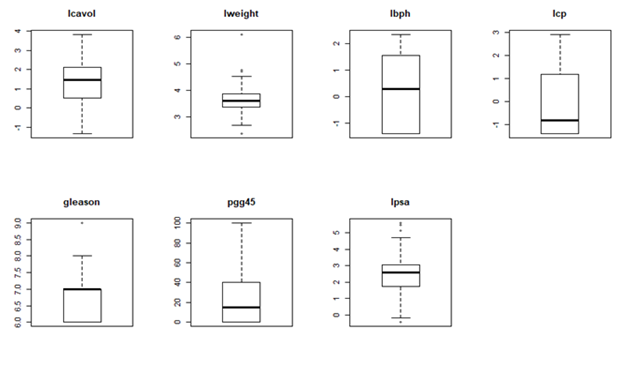


Grafico 1

Per la variabile *target* si ha:

Tabella 2



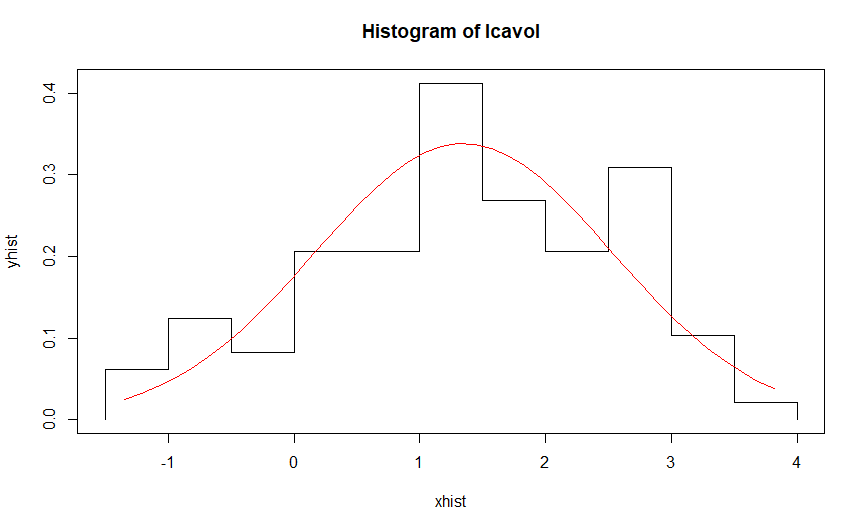


Grafico 2

La variabile *lcavol* sembra presentare una distribuzione normale (Grafico 2). Ciò viene confermato dal *test* di Anderson-Darling che con un *p-value* pari a 0.3375 suggerisce di accettare l’ipotesi nulla di normalità.

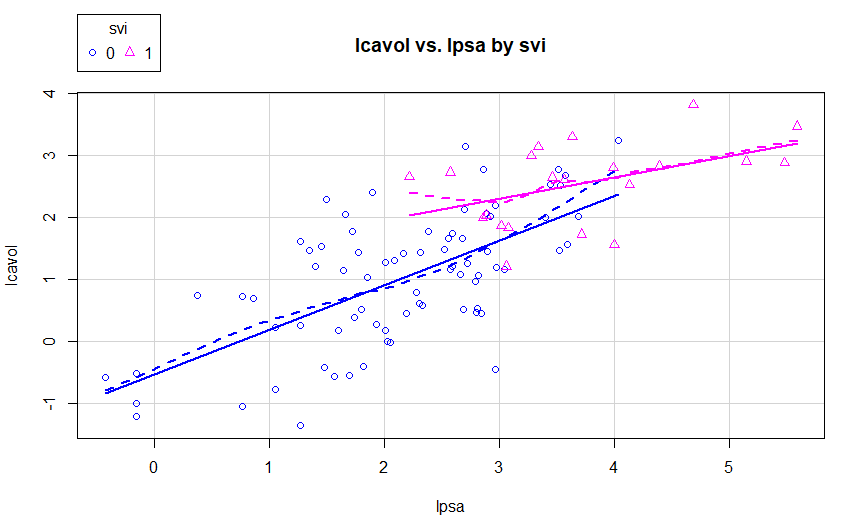


Grafico 3

Grazie allo *scatterplot* (Grafico 3) si può notare come al crescere del valore dell’antigene prostatico specifico cresca anche il volume del tumore, ma soprattutto, i casi che sono caratterizzati dall’invasione delle vescicole seminali sembrano evidenziare in media tumori con volumi maggiori.

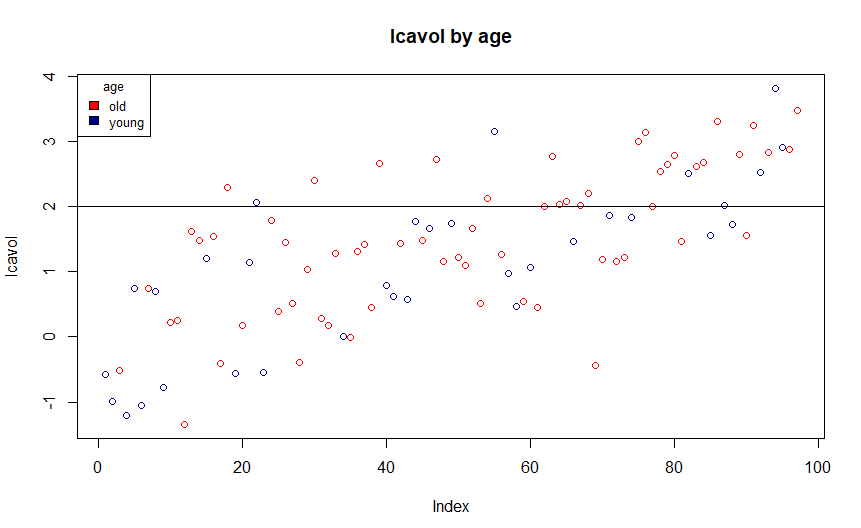


Grafico 4

Gli uomini dopo i 62 anni sembrano essere caratterizzati da tumori con volumi più elevati in media, anche se, il volume maggiore viene identificato in un individuo al di sotto della soglia di età (Grafico 4).

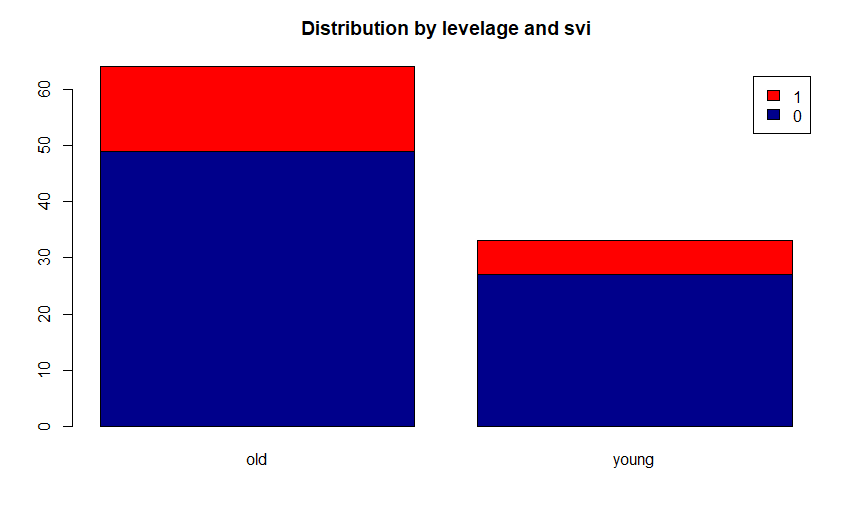


Grafico 5

Il grafico a barre (Grafico 5) mostra come si distribuiscono i pazienti in base a età e invasione delle vescicole seminali da parte del tumore: si può notare come più di 60 su 97 individui hanno superato i 62 anni. Inoltre, il fattore di invasione delle vescicole si distribuisce in proporzione molto simile tra le 2 classi di età, di conseguenza si può concludere che la dipendenza tra le due variabili categoriali è di carattere lieve.

È utile inoltre sottoporre a un’attenta valutazione la matrice di correlazione nel Grafico 6 per determinare se ci si trova in presenza di multicollinearità. La conferma verrà data dal calcolo dei *variance inflation factors* (vif) dopo la stima del modello lineare.

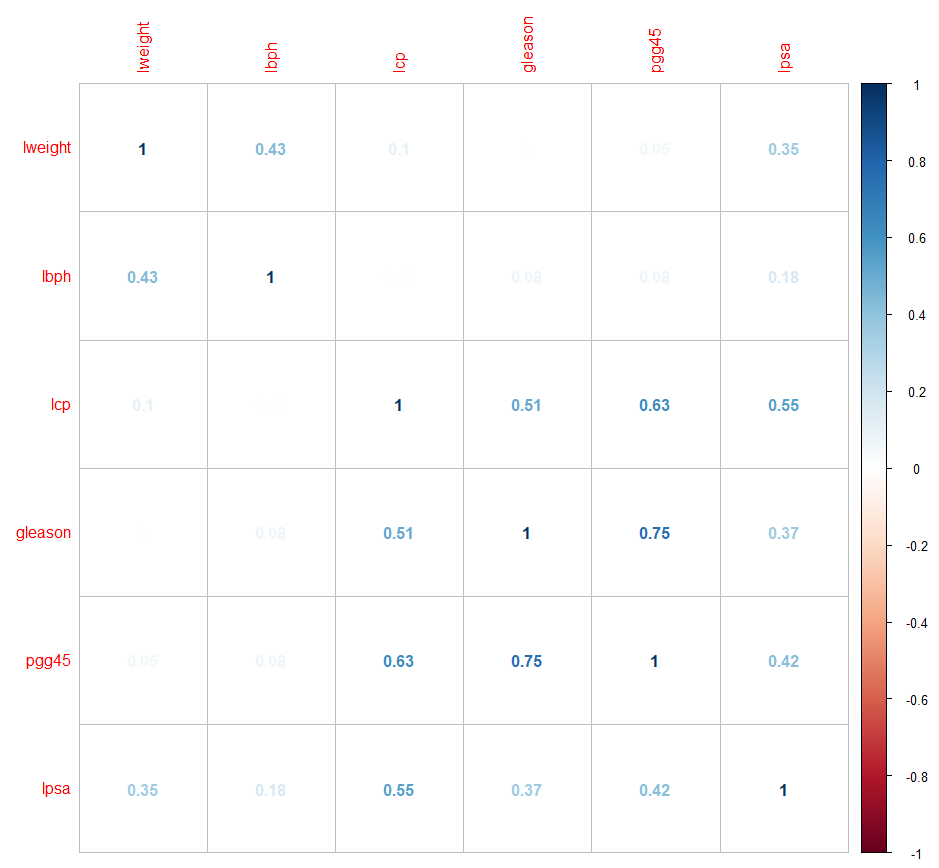


Grafico 6

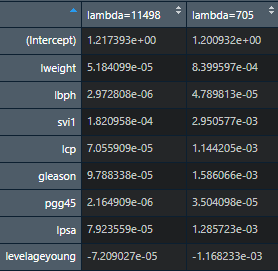
Le variabili *gleason* e *pgg45* hanno un’alta correlazione, 0.75, ma non tale da generare multicollinearità.

Infine, il *set* di dati viene suddiviso in *training set* e *test set*, i quali rispettivamente contengono il 75% delle osservazioni e il 25% delle osservazioni, in modo da poter elaborare i modelli sul *training set* e poi testare la performance predittiva di questi sul *test set*.

Elaborazione dei modelli

Per stimare modelli con i metodi *Ridge regression* e *Lasso* occorre utilizzare su R il pacchetto glmnet (). Per prima cosa si crea la matrice delle covariate togliendo dal *training set* la variabile *target* *lcavol*, isolandola in un vettore. Dopo aver creato una griglia di possibili valori di λ si stima il modello sul *training set* per ogni valore della griglia. È importante osservare che in corrispondenza di un parametro di *tuning* maggiore si osservano dei coefficienti stimati più vicini a 0. Nella Tabella 3 sono confrontati i coefficienti stimati con *Ridge Regression* in corrispondenza di λ=11498 e λ=705. Si nota che le stime associate a un parametro di complessità maggiore sono più contratte verso lo 0, fatta eccezione per l’intercetta che non viene influenzata dalla regolarizzazione.

Tabella 3



Per selezionare il miglior valore di λ da utilizzare si usa la procedura di *ten-fold cross-validation*, con la funzione cv.glmnet (). Nel Grafico 6 viene rappresentato il valore del *mean squared error* in funzione di log(λ) durante il processo di *cross-validation*. I valori che si trovano al di sopra del grafico indicano il numero di predittori che il modello sta usando. Siccome *Ridge regression* non effettua *variable selection* il numero di covariate incluse nel modello rimane costante per ogni valore di λ ed è 8.

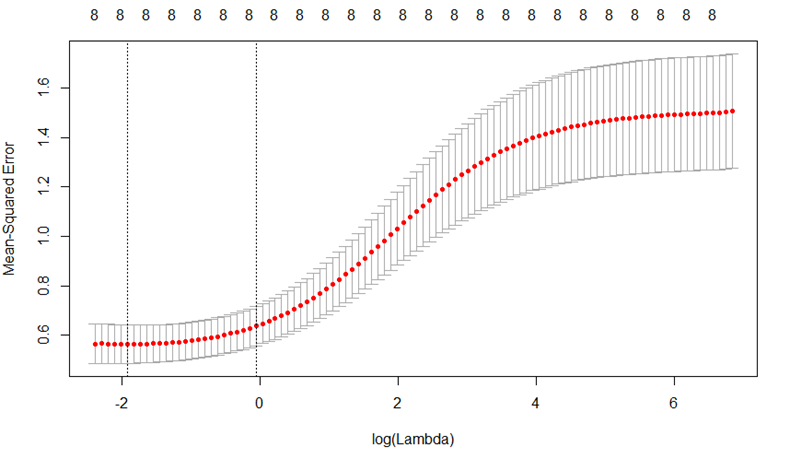
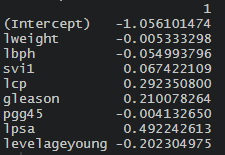


Grafico 6

Il valore di λ selezionato è pari a 0.11112. In conclusione, il modello *Ridge regression* viene ristimato sul *dataset* iniziale utilizzando il parametro di complessità selezionato. I coefficienti ottenuti sono illustrati nella tabella 4.

Tabella 4



Per la stima dei coefficienti *Lasso* il metodo da seguire è identico. Il grafico dei coefficienti (Grafico 7) mostra che in base alla scelta del parametro di *tuning* alcune stime possono essere esattamente uguali a zero, a differenza di ciò che accade con il metodo *Ridge regression*.

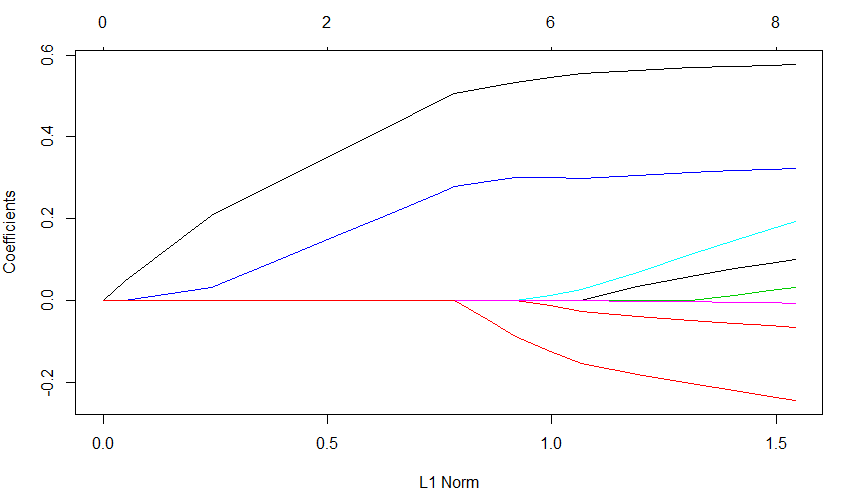


Grafico 7

La proprietà di *variable selection* di *Lasso* è confermata anche dal Grafico 8, dove si nota che al crescere di λ, e quindi anche di log(λ), il numero di predittori inclusi nel modello diminuisce in maniera progressiva.

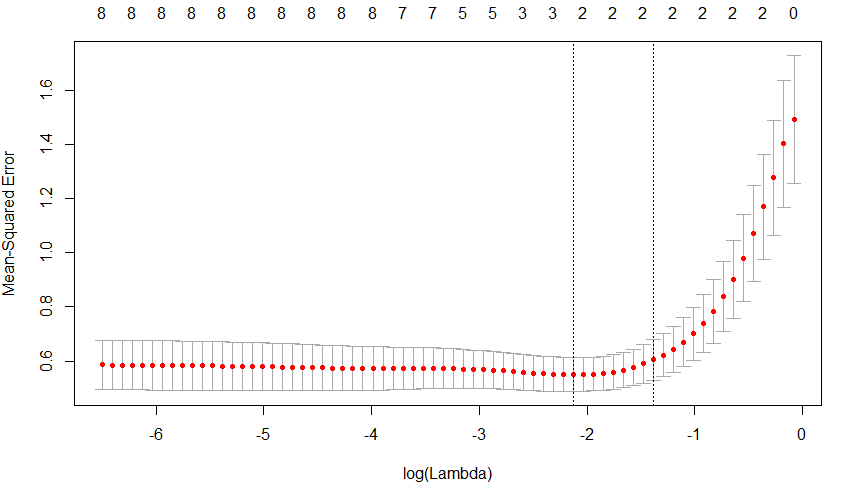
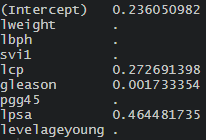


Grafico 8

Il valore di λ che viene selezionato è pari a 0.11915. Le stime ottenute sul *dataset* iniziale sono mostrate nella Tabella 5.

Tabella 5



Lasso ha dunque operato in maniera massiccia per quanto riguarda la selezione delle variabili, infatti sono stati annullati ben 5 predittori su 8 originari.

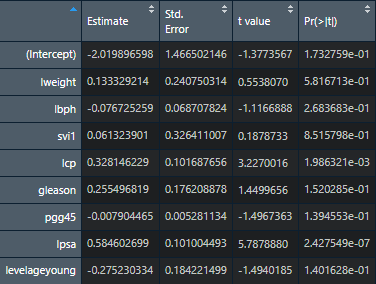
Si procede, infine, con la stima del modello di regressione lineare. Come primo passo si calcolano i vif (Tabella 6) per confermare l’ipotesi di assenza di multicollinearità formulata osservando la matrice di correlazione. Non vi è alcun valore al di sopra della soglia critica di 10, quindi si mantengono tutte le variabili all’interno del modello.

Tabella 6



I coefficienti stimati sono elencati nella Tabella 7.

Tabella 7

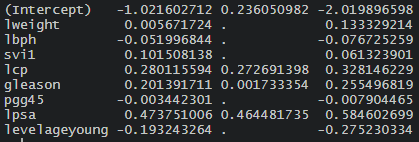


Vi sono solamente 2 coefficienti significativi, ovvero quelli riferiti alle variabili *lcp* e *lpsa*. Il coefficiente di *lpsa* indica che in seguito a un incremento unitario del logaritmo dell’antigene prostatico specifico e a parità di tutte le altre covariate, il logaritmo del volume del cancro aumenta di 0.584 in media. Il modello presenta un R2=0.7015 e R2adj=0.6636.

Confronto delle performance predittive

Per prima cosa è utile confrontare i coefficienti stimati dai vari modelli (tabella 8).

Tabella 8



Nella colonna più a sinistra sono evidenti gli effetti di contrazione dei coefficienti rispetto alla colonna più a destra (regressione dei minimi quadrati). Nella colonna centrale, dove si trovano i coefficienti *Lasso*, sono cristallini gli effetti della regolarizzazione, poiché 5 coefficienti sono stati annullati e gli altri sono ancora più contratti verso 0 rispetto ai coefficienti *Ridge regression*. Si possono inoltre confrontare le performance predittive dei modelli calcolando il *mean squared error* sul *test set* (tabella 9).

Tabella 9



Entrambi i metodi di regolarizzazione forniscono performance migliori rispetto ai minimi quadrati perché presentano *MSE* minori. La prestazione migliore in assoluto è fornita dal metodo *Lasso*. Quindi, il metodo *Lasso* non fornisce il modello migliore solo dal punto di vista della interpretabilità, infatti esso è anche il più preciso nelle previsioni. Come ultimo passo si confronta la performance ottenuta dal modello Lasso con quella ottenuta dal modello lineare dopo avere effettuato *stepwise selection*. La *stepwise selection* ha come obiettivo quello di selezionare il *set* di predittori che hanno la miglior relazione con la variabile risposta, basandosi sull’*AIC*.

Tabella 10

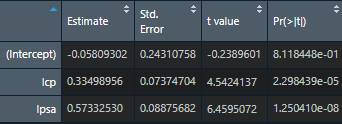


Tabella 11



La *stepwise selection* ha mantenuto solo 2 variabili all’interno del modello, ovvero *lcp* e *lpsa* (Tabella 10). Questa coppia di variabili infatti minimizza l’*AIC*. Tuttavia, la performance migliore resta quella del modello *Lasso* (Tabella11).

Conclusione

Questo studio ha cercato di rispondere alla domanda: “I metodi di regolarizzazione migliorano *OLS* in termini di efficienza nelle previsioni e interpretabilità dei modelli?” A tal fine, dopo l’esposizione dei concetti in forma teorica, è stata condotta un’analisi su un *dataset* contenente delle osservazioni riguardanti dei casi di cancro alla prostata recuperato direttamente all’interno del *software* R. I risultati finali ottenuti con l’applicazione dimostrano che nel contesto preso in esame sia il modello *Lasso* che il modello *Ridge* forniscono prestazioni predittive migliori rispetto a *OLS*. Inoltre, col metodo *Lasso* non solo si ottiene il minor *MSE* sul *test set*, che quindi identifica la miglior performance, ma si nota anche una pesante opera di *feature selection* poiché 5 degli 8 predittori iniziali vengono annullati dagli effetti della regolarizzazione. Per argomentare i risultati è fondamentale considerare il *set* di dati analizzato, composto da 97 osservazioni e 9 variabili. È un contesto in cui vi è un numero molto ristretto di osservazioni, considerando la quantità di variabili presenti. In queste situazioni le stime dei minimi quadrati hanno una varianza molto alta. Quindi, nei casi in cui il *training set* è composto da poche osservazioni, dei piccoli cambiamenti all’interno di esso possono causare delle grandi variazioni nelle stime. D’altra parte, con i modelli *Lasso* e *Ridge regression* si ottiene un’elevata diminuzione della varianza a fronte di un piccolo aumento di *bias*, quindi i coefficienti stimati sono meno sensibili alle variazioni nel *training set*. Inoltre, *Lasso* performa meglio di *Ridge regression* poichè in questa situazione solo alcuni predittori hanno relazioni forti con la variabile risposta, dunque, la proprietà di *feature selection* si rende molto utile, non solo dal punto di vista della intepretabilità, ma anche in merito alla performance del modello. In conclusione, è importante tenere presente che nella fase di applicazione si è deciso di utilizzare dei dati aventi delle caratteristiche confacenti al raggiungimento dell’obiettivo. Infatti, nel caso in cui dovessero cambiare la dimensionalità dei dati o le relazioni delle covariate con la variabile *target*, i risultati potrebbero variare.

Bibliografia

1. JAMES G., WITTEN D., HASTIE T., TIBSHIRANI R., *An Introduction to Statistical Learning, Springer, 2013*
2. SALEH A., ARASHI M., KIBRIA M., *Theory of Ridge Regression Estimation with Applications, Wiley, 2019*

Sitografia

1. Tibshirani R., 1996,

<https://www.jstor.org/stable/2346178?readnow=1&seq=1#page_scan_tab_contents>

1. Van Wieringen W.,2019,

<https://arxiv.org/pdf/1509.09169;Lecture>

1. Kirkland L., Kanfer F., Millard S., 2015, <https://www.researchgate.net/publication/287727878_LASSO_Tuning_Parameter_Selection>
2. <https://rpubs.com/haj3/ridgelasso>

Appendice

1. Script R utilizzato per l’applicazione

library(lasso2)

data(Prostate)

dati=Prostate

attach(dati)

library(funModeling)

tesi1=data.frame(df\_status(dati))

#pre-processing, ricodifico le varibile age e svi

dati$svi=as.factor(svi)

dati$levelage[dati$age<=62]="young"

dati$levelage[dati$age>62]="old"

dati$levelage=as.factor(dati$levelage)

df\_status(dati)

#elimino la variabile age

dati=dati[,-c(3)]

#analisi descrittive

attach(dati)

summary(lcavol)

h<-hist(lcavol,breaks=15)

xhist<-c(min(h$breaks),h$breaks)

yhist<-c(0,h$density,0)

xfit<-seq(min(lcavol),max(lcavol),length=40)

yfit<-dnorm(xfit,mean=mean(lcavol),sd=sd(lcavol))

plot(xhist,yhist,type="s",ylim=c(0,max(yhist,yfit)),

main="Histogram of lcavol")

lines(xfit,yfit,col="red")

ad.test(lcavol)

par(mfrow=c(1,1))

counts=table(dati$svi,dati$levelage)

barplot(counts,legend=rownames(counts),col=c("darkblue","red"),main="Distribution by levelage and svi")

datanum=dati[,-c(1,4,9)]

df\_status(datanum)

par(mfrow=c(2,4))

v=colnames(datanum)

for (i in 1:7){

boxplot(datanum[,i],main=v[i])

}

library(car)

scatterplot(lcavol~lpsa|svi,main="lcavol vs. lpsa by svi")

col1<-c("red", "darkblue")

par(mfrow=c(1,1))

plot(dati$lcavol,

main="lcavol by age", ylab="lcavol",

col = col1[dati$levelage])

legend("topleft", title="age", cex = 0.8,

legend=levels(dati$levelage), fill = col1)

abline(h=2)

#analisi della correlazione

C=cor(datanum)

library(corrplot)

corrplot(C,method="number")

#.....................

#ridge regression

x=model.matrix(lcavol~.,dati)[,-1]

y=dati$lcavol

library(glmnet)

grid=10^seq(10,-2,length=100)

ridge.mod=glmnet(x,y,alpha=0,lambda=grid)

set.seed(123)

train=sample(1:nrow(dati),nrow(dati)\*0.75)

test=(-train)

y.test=y[test]

ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid,thresh=1e-12)

#visualizzazione dei coefficienti per diversi valori di lambda

ridge.mod$lambda[50]

coef(ridge.mod)[,50]

ridge.mod$lambda[60]

coef(ridge.mod)[,60]

tesi2=as.data.frame(cbind(coef(ridge.mod)[,50],coef(ridge.mod)[,60]))

colnames(tesi2)=c("lambda=11498","lambda=705")

set.seed(123)

cv.out=cv.glmnet(x[train,],y[train],alpha=0)

plot(cv.out)

bestlam=cv.out$lambda.min

bestlam

#MSE modello Ridge

ridge.pred=predict(ridge.mod,s=bestlam,newx=x[test,])

ridge=mean((ridge.pred-y.test)^2)

out=glmnet(x,y,alpha=0)

ridge.coef=predict(out,type="coefficients",s=bestlam,dati)

ridge.coef

plot(ridge.mod)

#lasso

lasso.mod=glmnet(x[train,],y[train],alpha=1,lambda=grid)

plot(lasso.mod)

set.seed(123)

cv.out=cv.glmnet(x[train,],y[train],alpha=1,nfolds=10)

plot(cv.out)

bestlam=cv.out$lambda.min

bestlam

#Mse modello Lasso

lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])

lasso=mean((lasso.pred-y.test)^2)

out=glmnet(x,y,alpha=1,lambda=grid)

lasso.coef=predict(out,type="coefficients",s=bestlam)

lasso.coef

cbind((ridge.coef),(lasso.coef),summary(mod)$coefficients)

#modello lineare

training=dati[-c(1,7,16,18,22,29,32,33,34,40,54,55,59,61,62,63,64,67,68,71,72,73,92,93,97),]

testing=dati[c(1,7,16,18,22,29,32,33,34,40,54,55,59,61,62,63,64,67,68,71,72,73,92,93,97),]

mod=lm(lcavol~lweight+lbph+svi+lcp+gleason+pgg45+lpsa+levelage,data=training)

tesi4=as.data.frame(summary(mod)$coefficients)

summary(mod)

library(car)

tesi3=as.data.frame(vif(mod))

#MSE modello lineare

mod.pred=predict(mod,testing)

modlin=mean((mod.pred-y.test)^2)

#stepwise selection del modello lineare

library(MASS)

mod2=stepAIC(mod)

summary(mod2)

tesi6=as.data.frame(summary(mod2)$coefficients)

#MSE del modello dopo stepAIC

mod2.pred=predict(mod2,testing)

modstep=mean((mod2.pred-y.test)^2)

vett=c(lasso,modstep)

tesi6=round(data.frame(t(vett)),4)

colnames(tesi6)=c("mod.lasso","mod.stepAIC")

rownames(tesi6)=c("MSE")