

Classificazione di immagini sulla base della rugosità del terreno

Gaia Penta, Gianluca Ivo Tori, Nicolò Zane

08-01-2023

1. Descrizione del problema e dei dati

Il dataset originale è composto da 12.982 fotogrammi, ognuno di dimensioni 3840 x 2160 pixels, raffiguranti varie sezioni di un terreno sterrato: questi sono estratti da video (effettuati in 5 giornate differenti) ripresi da una telecamera monoculare collocata sul manubrio di una mountain bike. Ad accompagnare i fotogrammi sono presenti altri dati relativi a giroscopio (*gyroscope_calibrated_split.csv*), accelerometro (*accelerometer_calibrated_split.csv*), magnetometro (*magnetometer_split.csv*), rilevati con cadenza di 10 ms, e gps (*gps.csv*) con cadenza di 100 ms. Inoltre, sono presenti anche dati relativi a longitudine, latitudine, altitudine e velocità, rilevati con cadenza di 1000 ms, presenti nel dataset *record.csv*. Tra queste, i ricercatori in questione hanno utilizzato come indicatore per la ruvidezza del terreno *z-axis acceleration* che rappresenta i movimenti verso l'alto e il basso del veicolo durante il percorso. Sono infatti disponibili 8 diverse serie di etichette, sei di queste create attraverso un k-means clustering (con $k = 2, 3$ e 4) basato su statistiche della variabile *z-axis acceleration* e due analizzando la distribuzione dei dati, suddividendo quindi le immagini in classi sulla base della ruvidezza del terreno. Infine, per rispettare alcuni criteri riguardanti i sensori e, affinché in ogni fotogramma fosse presente il percorso, l'insieme originale è stato filtrato fino a raggiungere una numerosità di 7.061 fotogrammi.



Figure 1: Esempio di immagine a dimensioni reali

2. Obiettivi

Nella creazione di questo insieme di dati, l'obiettivo dei ricercatori è stato quello di definire delle metriche di ruvidezza del terreno per poter allenare algoritmi in grado di riconoscere terreni più o meno sconnessi, in

modo da poter essere utilizzati in applicazioni per la guida autonoma sia su strada, sia per gli *Autonomous Ground Vehicles*, cioè dei veicoli creati per completare delle azioni specifiche senza la supervisione umana (come ad esempio l'esplorazione spaziale).

L'obiettivo del presente progetto è stato quello di proporre diverse metodologie per la definizione delle etichette (da confrontare con le etichette proposte originariamente dai ricercatori), utilizzando poi metodi adeguati per cercare degli spunti interpretativi, sfruttando la sparsità, che può essere ipotizzabile data la struttura delle immagini, a scapito della capacità di previsione.

Per quest'ultimo fine, infatti, i ricercatori hanno utilizzato un *Convolutional Neural Network* (nome), modello che permette di considerare la complessa struttura dei dati, in particolare l'utilizzo simultaneo dei 3 canali RGB di ciascuna immagine.

3. Dataset

Per prima cosa si è creato il dataset sul quale successivamente sono stati stimati i modelli.

Per quanto riguarda le immagini, al fine di poterle utilizzare successivamente, esse sono state caricate per mezzo della funzione *load.image* della libreria *imager*, ottenendo inizialmente un array di dimensioni 3840x2160x3 per ciascun fotogramma. Visto quindi l'onere computazionale che sarebbe stato richiesto utilizzando tutte e tre le matrici, si è deciso di utilizzare solo uno dei tre canali contenuti nei file PNG. Inoltre, si passa da una dimensione di 3840x2160 (7 516 800 covariate), come visibile in Figura 1, a 190x340 (64 600 covariate), visibile in Figura 2.

Quindi, una volta caricate le immagini precedentemente selezionate dai ricercatori, queste sono state vettorizzate: ogni colonna dell'immagine è stata concatenata alla successiva per formare la singola osservazione.



Figure 2: Esempio di immagine alla grandezza utilizzata nell'analisi

Sono stati processati anche i dati relativi all'accelerometro. Questi danno informazione sul movimento del veicolo verso l'alto e il basso e di conseguenza sulla ruvidezza del terreno. In particolare, si è tenuto conto della deviazione standard della variabile *accelerometer\$calibrated_accel_z...g.* nella finestra di tempo relativa alla foto.

Dei dati relativi al giroscopio invece si decide di tenere in considerazione la variabile relativa all'asse y (*gyroscope\$calibrated_gyro_y..deg.s.*), che tiene conto delle rotazioni del veicolo sul proprio asse. Anche qui, si considera la sua deviazione standard, che riflette, come prima, le oscillazioni del veicolo date dal suo andamento sul terreno.

Si è deciso poi di utilizzare anche il dataset *record.csv*, ed in particolare la variabile *enhanced_speed..m.s.*, in quanto, pur fornendo informazioni simili a quelle date dal sensore gps, esse sono state indicate dai ricercatori come più precise.

I dataset relativi al magnetometro e al gps quindi non sono stati utilizzati.

3.2 Definizione della variabile risposta e analisi esplorativa

A questo punto, si procede a creare le etichette per classificare le immagini.

Si può notare in Figura 3 come il numero di rilevazioni nei vari giorni sia molto variabile. Questo aspetto verrà tenuto in considerazione nella definizione dell'insieme di stima e di verifica nella fase di modellazione.

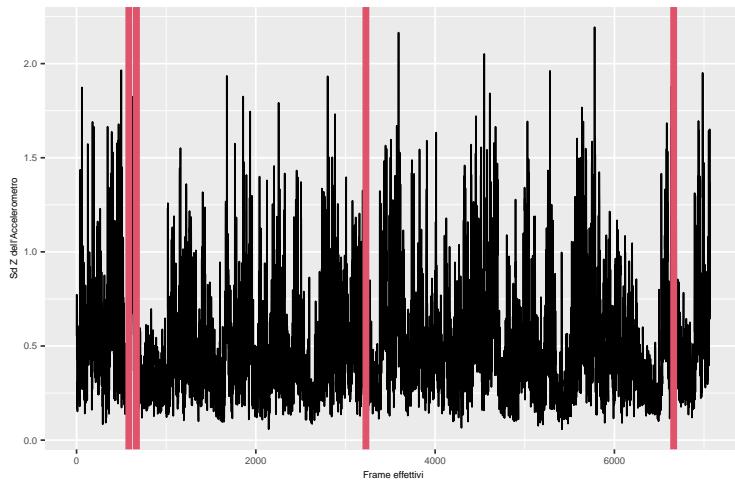


Figure 3: Andamento delle deviazioni standard delle rilevazioni dell'asse Z nei 5 giorni

3.2.1 Fused Lasso Signal Approximator

Il primo metodo utilizzato per definire le etichette è stato il *Fused Lasso Signal-Approximator*. Data la dipendenza temporale delle osservazioni si ritiene realistico ipotizzare una penalità riguardante rilevazioni contigue. In questo modo, approssimando l'andamento delle osservazioni, le immagini sono state suddivise in classi a seconda che siano uguali, maggiori o minori della media, definendo così i diversi livelli di ruvidezza.

Per applicare questo metodo i dati sono stati centrati.

In seguito vengono presentati soltanto i FLSA con i parametri scelti in seguito a varie prove, le quali non vengono riportati per sinteticità.

```
# Fused Lasso -----
Y <- as.numeric(scale(Sd_z_Acc$SD_z_Acc, scale = F))
# Definisco una y dalle deviazioni standard centrate per Fused Lasso

fit <- flsa(Y)
# Stimo Fused lasso
```

Per definire le etichette con due categorie si è impostata una penalità lasso prossima a zero, così da neutralizzarla e privilegiare quella di fusione.

Si prendono quindi le osservazioni minori di 0 come classe 1 e quelle maggiori di 0 come classe 2, indicando rispettivamente bassa e alta ruvidezza. Ciò è visibile in Figura 4. Questo metodo porta ad un insieme di dati bilanciato con rispettivamente il 56% delle osservazioni classificate come 1 e 44% come 2.

```
pred <- flsaGetSolution(fit, lambda1 = 0.000001, lambda2 = 0.5)
# Dopo diverse prove si scelgono come parametro lasso 0.000001 e parametro di
# fusione 0.5
```

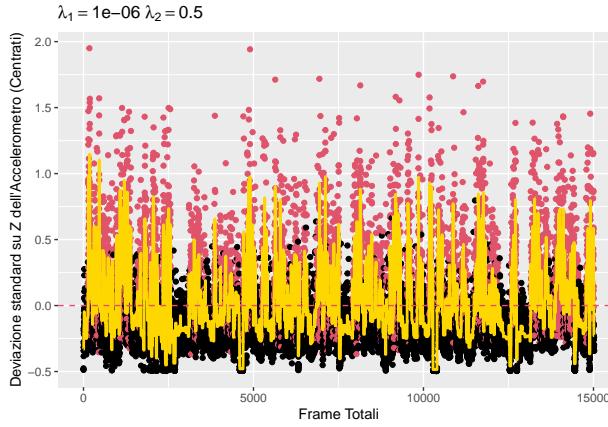


Figure 4: Suddivisione delle osservazioni in due classi sulla base di FLSA, con $\lambda_1 = 0.000001$, $\lambda_2 = 0.5$

Per definire tre categorie invece si è applicata una penalità lasso molto diversa da zero, così da favorire la selezione delle osservazioni. Di conseguenza si definiscono tre classi, aggiungendo, rispetto al caso precedente, le osservazioni uguali a zero, determinando così tre livelli di ruvidezza. Si veda Figura 5. Anche in questo caso l'insieme risultati è bilanciato con rispettivamente 35%, 33% e 32%.

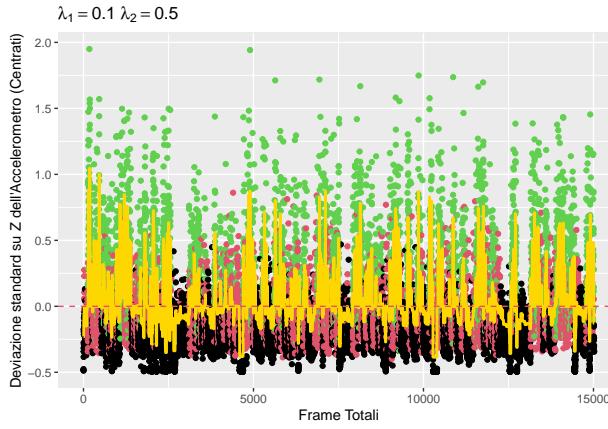


Figure 5: Suddivisione delle osservazioni in tre classi sulla base dell'FLSA, con $\lambda_1 = 0.1$, $\lambda_2 = 0.5$

3.2.2 K-means Clustering

Come secondo metodo per la definizione delle etichette, si è deciso di seguire quello usato dai ricercatori, ossia il *K-means Clustering*, aggiungendo però altre due variabili per la definizione dei gruppi. Nello specifico, oltre alla deviazione standard di *accelerometer\$calibrated_accel_z..g.*, si utilizza anche la deviazione standard di *gyroscope\$calibrated_gyro_y..deg.s.* e la velocità *record\$enhanced_speed..m.s..*

La scelta è data dal fatto che la presenza di buche o tratti di terreno impervio può causare, oltre che oscillazioni in alto e in basso, anche rotazioni del veicolo sul proprio asse. Inoltre, ci si aspetta che anche la velocità dia un contributo nei movimenti della bici. Ciò è confermato anche dalla matrice di correlazione in Figura 6, che mostra correlazione positiva tra tutte le variabili.

Dal momento che le variabili sono misurate su scale diverse, si è deciso di standardizzare i dati prima di effettuare il clustering.

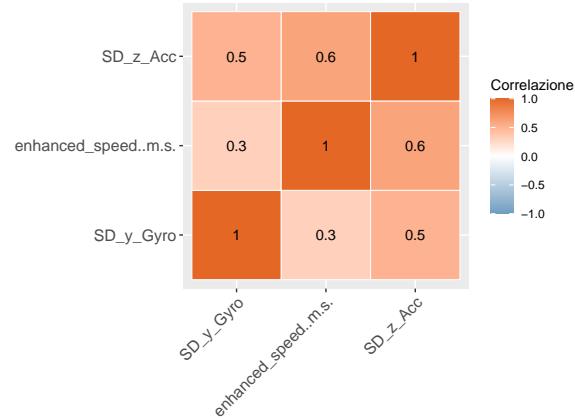


Figure 6: Correlazione tra le variabili utilizzate per il k-means clustering

In questo caso la definizione delle etichette attraverso *k-means* ha portato in entrambi i casi a insiemi di dati sbilanciati. Lo sbilanciamento può portare dei problemi durante la modellazione, quindi questo verrà tenuto in considerazione applicando opportuni accorgimenti successivamente.

3.3 Dataset alternativi

Coscienti del fatto che il ridimensionamento dell'immagine sia solo uno dei metodi per approcciare il problema della riduzione della complessità e della dimensionalità, decidiamo di provare alcune alternative.

Verifichiamo se effettivamente esiste un pattern comune nelle immagini, andando a creare un'immagine media, facendo la media per colonna in tutto il dataset. Com'è possibile notare dall'immagine in Figura 7.a, il sentiero si concentra nella parte centrale ed inferiore dell'immagine, mentre ai lati si trova presumibilmente la vegetazione; nella parte superiore spesso è invece presente il cielo o squarci di questo.

Un altro metodo possibile per capire come trattare le varie immagini è utilizzare il *Fused Lasso Signal Approximator* per eliminare del rumore presente nelle immagini andando ad appiattire la dinamica dei pixel. In questo caso la penalità, in riferimento all'immagine, agisce per colonna. Come si può vedere dall'immagine in Figura 7.b questo metodo porta ad identificare in particolare la parte inferiore e superiore dell'immagine, lasciando la parte centrale più scura.

Un'altra possibile alternativa sarebbe quella di utilizzare il *Fused Lasso 2D*, che penalizza i pixel vicini sia per colonna che per riga; si decide comunque di non intraprendere questa strada in primis per il carico computazionale ma anche perché riteniamo che non porti a risultati significativamente diversi.



Figure 7: Confronto tra immagine media (immagine a sinistra, (a.)), Fused Signal Approximator su pixel medi (immagine centrale (b.)) e Fused Signal Approximator sull'immagine mostrata in Figura 2 (immagine a destra (c.))

Dalle informazioni ricavate dall'immagine media e dal Fused Lasso Signal Approximator si è deciso di ritagliare poi le immagini conservando solo la parte riguardante il terreno. Si sono utilizzate le immagini a delle dimensioni iniziali pari a 340x200 pixels, tagliate in modo tale da ottenere una dimensione pari a 170x100. Un esempio è mostrato in Figura 8. L'obiettivo è quello di conservare più informazione riguardo la sezione d'interesse.

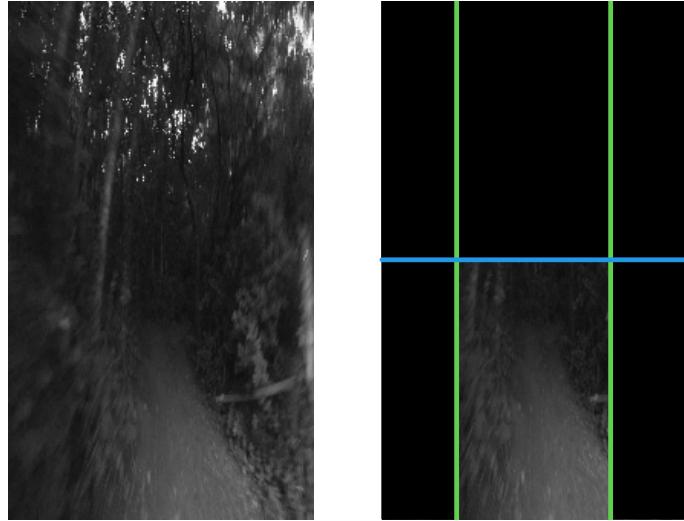


Figure 8: Confronto tra immagine originale e immagine tagliata

4. Modelli

Si decide di dividere il dataset in due parti: un insieme di train e uno di test. In particolare si è deciso di tenere come insieme di test i dati relativi al terzo giorno, in quanto dotato di una numerosità sufficiente, e i rimanenti giorni come training set. Un altro modo in cui si sarebbe potuto dividere il dataset sarebbe stato quello cronologico, utilizzando i primi 4 giorni come training e l'ultimo come test, oppure tenere nel train tutte le parti relative ai primi km di riprese e le ultime parti come test.

Inoltre, si fissano i 5 fold per la convalida incrociata utile per la regolarizzazione dei modelli, così da renderli comparabili.

Si mostreranno quindi le performance di ciascun modello su ogni etichetta, analizzandone eventuali fallimenti.

Come già detto in precedenza, i dataset con le etichette ottenute con k-means clustering e le etichette originali definite dai ricercatori risultano essere sbilanciate, con proporzioni visibili in Tabella 1.

Table 1: Bilanciamento delle etichette nei vari dataset

	1	2
Fused	0.568	0.432
K-Means	0.629	0.371
Kaggle	0.725	0.275

Al fine di tenere conto di tale sbilanciamento si è deciso di utilizzare come metrica l'indice Kappa di Cohen, definito come:

$$\kappa = \frac{p_0 - p_e}{1 - p_e}$$

dove p_e è la probabilità di classificare correttamente in modo casuale, mentre p_0 è l'accuracy totale. Esso, durante l'adattamento dei modelli, dovrà essere massimizzato.

Questo indice infatti consente di dare un'indicazione di quanto il modello stia classificando meglio rispetto ad un modello che classifica a caso, supponendo che quest'ultimo stia utilizzando semplicemente le frequenze di ciascuna classe.

Onde evitare di definire delle procedure ad hoc per la massimizzazione di tale indice, si è deciso di utilizzare comunque le funzioni di cross validation dei modelli utilizzati, in quanto la maggior parte di essi consentono comunque di ottenere in output la stima del predittore lineare per ogni parametro di penalizzazione testato per l'insieme *out of fold* utilizzando il tasso di errata classificazione. Si è quindi provveduto nei vari casi a calcolare il corrispondente valore di κ riproducendo quindi una convalida incrociata.

In alcuni modelli, dove il solo utilizzo di tale indice non sarà sufficiente, si utilizzerà anche una soglia specifica nella definizione delle previsioni.

4.1 Lasso Logistico

Il primo modello adattato è la regressione logistica con penalità lasso (norma L1). Per quanto riguarda le etichette *fused lasso*, si effettua una convalida incrociata su 5 fold su una griglia di valori di $\log(\lambda)$ che va da -5 a -2.5, di lunghezza 100. Utilizzando il $\log(\lambda)$ che minimizza l'errore di classificazione, pari a -3.889, vengono selezionati 60 coefficienti compresa l'intercetta.

```
Lasso_logistico_Fused <- cv.glmnet(X_train, factor(train$Etichette_Fused_K2),
                                         type.measure = "class", family = "binomial",
                                         alpha = 1, nfolds = 5, foldid = fold,
                                         lambda = lambda.grid, parallel = T,
                                         trace.it = 1, seed = 42)
```

Si vede poi in Figura 13.a quali sono i coefficienti selezionati dal modello, visualizzati nella posizione effettiva. Nonostante sia di difficile interpretazione, possiamo comunque notare una maggiore densità di pixel nella parte inferiore e centrale dell'immagine.

Per quanto riguarda le etichette k-means, si utilizza una griglia di $\log(\lambda)$ che va da -7 a -4, utilizzando come metrica di ottimizzazione il tasso di errata classificazione. Viene specificato in `cv.glmnet` l'opzione `keep = T`, che consente di tenere la stima del predittore lineare per ogni $\log(\lambda)$ testato per le osservazioni *out of fold*. A questo punto è possibile riprodurre una convalida incrociata utilizzando il Kappa di Cohen. Massimizzando quindi questo indice, si provvede a selezionare un λ pari a 0.0099911, che determina la selezione di 634 variabili, contro le 69 che sarebbero state scelte utilizzando l'errore di classificazione, il quale avrebbe portato tuttavia a predire soltanto la classe più numerosa. Si riporta in Figura 9 a fini esplicativi il grafico di Kappa di Cohen al variare di λ , confrontato con il relativo tasso di errata classificazione medio.

```
Lasso_logistico_Kmeans <- cv.glmnet(X_train, factor(train$Etichette_Kmeans_K2),
                                         type.measure = "class", family = "binomial",
                                         alpha = 1, nfolds = 5, foldid = fold,
                                         lambda = lambda.grid.lasso, parallel = T,
                                         trace.it = 1, seed = 42, keep = T)

eta.lasso <- Lasso_logistico_Kmeans$fit.preval
pi.train.lasso <- exp(eta.lasso)/(1+exp(eta.lasso))
pred.train.class.lasso <- ifelse(pi.train.lasso > 1/2, 2, 1)
# Mi riconduco dal predittore lineare alle classi stimate

kappa.vals.lasso <- matrix(NA, ncol(pred.train.class.lasso), 5)
# Una metrica ragionevole per il caso presente di sbilanciamento è il Kappa di
# cholen, per tanto inizializzo una matrice vuota con numero di righe pari al
# numero di lambda, e numero di colonne pari ai fold utilizzati

for( i in 1:ncol(pred.train.class.lasso)){
  for(j in 1:5){
    ind.fold.out <- which(Lasso_logistico_Kmeans$foldid == j)
    # Definisco il fold corrente

    if (length(unique(pred.train.class.lasso[ind.fold.out,i])) != 1){
      # Condizione per la quale si valuta la metrica relativa a quel fold, di
      # un lambda se e solo se le classi stimate non hanno solo un livello

      tab <- table(pred.train.class.lasso[ind.fold.out,i],
                    train$Etichette_Kmeans_K2[ind.fold.out])
      kappa.vals.lasso[i,j] <- Kappa(tab)$Unweighted[1]
      # Calcolo Kappa e salvo il risultato per il j-esimo fold
    }
  }
}
```

Anche per le etichette Kaggle si effettua la stessa procedura utilizzata per k-means. Si seleziona in questo caso un λ pari a 0.002130252, che determina la selezione di 2 476 variabili.

4.2 Elastic Net

Si propone poi un modello *Elastic Net* logistico, che può essere utile in questo caso per tenere conto della collinearità che ci si aspetta essere presente tra i pixel. Questo modello infatti ci consentirà di selezionare più coefficienti rispetto al modello precedente grazie all'utilizzo anche della penalità *ridge* (norma L2).

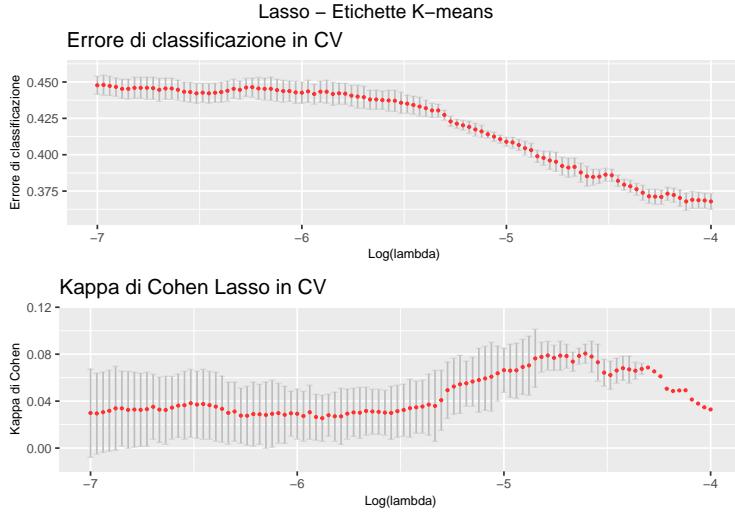


Figure 9: Andamento dell'errore di classificazione medio (in alto) e del Kappa di Cohen (in basso) al variare di log-lambda nel Lasso Logistico con etichette k-means

Quindi, oltre alla regolarizzazione di λ , verrà selezionato tramite convalida incrociata anche il parametro α , che pesa le due penalità.

Nel caso delle etichette *fused lasso*, la prima regolarizzazione porta ad un α pari a 0.3, dando quindi un peso alla penalità L1 pari a 0.3, contro un peso alla penalità ridge di 0.7. L'errore di classificazione medio al variare di α è visibile in Figura 10.

Il λ selezionato invece è pari a 0.06062607 con un numero di coefficienti selezionati che ammonta a 116 compresa l'intercetta, cosa che conferma quanto detto in precedenza.

```
# Elastic net Logistico Fused Lasso -----
lambda.grid <- exp(seq(-5,-2.5,1=100))
alpha.grid.enet <- c(0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1)
error.alpha.enet <- rep(NA, length(alpha.grid))
se.alpha.enet <- rep(NA, length(alpha.grid))

for(i in 1:length(alpha.grid.enet)){
  registerDoMC(cores = 5)
  # Al fine di parallelizzare la funzione cv.glmnet dichiaro il numero di core
  # che intendo utilizzare per parallelizzare il modello

  fit <- cv.glmnet(X_train, factor(train$Etichette_Fused_K2), alpha = alpha.grid.enet[i],
                    type.measure = "class", family = "binomial", nfolds = 5,
                    foldid = fold, lambda = lambda.grid, parallel = T,
                    trace.it = 1, seed = 42)
  # Viene stimato un modello logistico con penalità Elastic-Net, in CV a 5 fold con
  # metrica di ottimizzazione l'errore di errata classificazione, per ogni valore di alpha
  # contenuto in alpha.grid, viene oltretutto fissato un seed per la riproducibilità dei risultati
  error.alpha.enet[i] <- fit$cvm[fit$index["min",]]
  se.alpha.enet[i] <- fit$cvsd[fit$index["min","]]
  # La procedura valuta per ogni modello di parametro alpha_i il miglior
  # errore con standard deviation associata, per poi essere tra loro confrontati
}
```

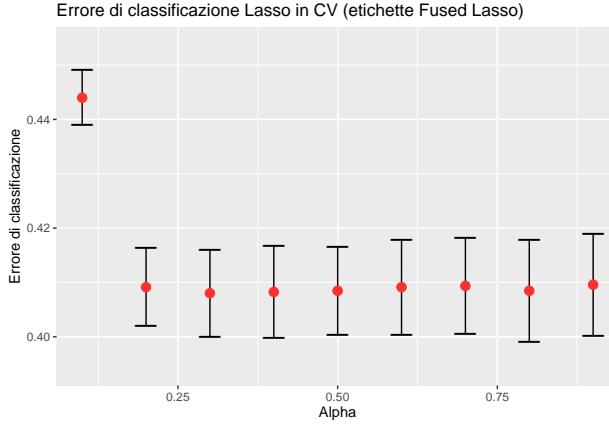


Figure 10: Andamento del tasso di errata classificazione medio al variare di alpha per le etichette fused lasso, dato il miglior lambda

Per quanto riguarda *k-means*, si adatta lo stesso procedimento precedente, valutando però l'indice Kappa di Cohen. Si adattano i modelli su una griglia di $\log(\lambda)$ che va da -7 a -4 e si valutano 9 valori di α equispaziati da 0.1 a 0.9, scegliendo un valore pari a 0.7. Si seleziona un valore di λ pari a 0.01273201 corrispondente a 873 coefficienti diversi da 0.

Stessa procedura viene adottata anche per *Kaggle*, sempre utilizzando come metrica per la cross validation il Kappa di Cohen con il procedimento già esplicitato in precedenza. Si sceglie un α pari a 0.9 e un λ pari a 0.009403563. I coefficienti selezionati sono 738.

4.3 SCAD E MCP

Si propongono poi due modelli con penalità non convessa, al fine di ridurre la distorsione che caratterizza il lasso. Si decide di non implementare una procedura di lasso adattivo, in quanto essa, svolgendosi in due passi e, volendo definire un peso per ogni coefficiente (ad esempio con una regressione ridge), sarebbe stato troppo oneroso per la nostra applicazione. In entrambi i casi ci si aspetta che vengano selezionati meno coefficienti rispetto ai modelli precedenti. Partendo sempre dalle etichette fused, sia per SCAD che per MCP, si utilizzano i valori di γ proposti di default dalla libreria *ncvreg*, pari rispettivamente a 3.7 e 3.

Per quanto riguarda SCAD, il γ scelto rende la regione non convessa molto ampia. Il λ selezionato dalla procedura di cross validation, pari a 0.01981706, determina un numero di variabili diverse da 0 pari a 31. Il valore di λ più piccolo per cui la regione è convessa risulta essere pari a 0.0525133; ne consegue che, come visibile in Figura 11.a, il modello stimato ricade nella regione di non convessità. Ne consegue che il risultato non è affidabile, in quanto si potrebbe essere incorsi in un minimo locale.

Per quanto riguarda invece MCP, il γ scelto fa sì che il λ ottimo ricada all'interno della regione localmente convessa, rendendo il risultato affidabile. Vengono selezionate infatti 27 variabili diverse da 0, con un limite che invece questa volta si trova in corrispondenza di un numero di variabili pari a 99. Ciò è visibile nella Figura 11.b.

Per quanto riguarda invece *k-means*, si utilizza un valore di γ elevato, pari a 20, per permettere di selezionare modelli con più variabili ed evitare di incorrere nella previsione di una sola classe. Per quanto riguarda MCP, si seleziona un λ pari a 0.007454109, corrispondente ad un numero di variabili pari a 83. Il λ più piccolo per cui la funzione risulta convessa è pari a 0.00922615, contro un λ minimo selezionato dalla cross validation pari a 0.007454109, che cade quindi nella regione non convessa e non rende i risultati affidabili. La stessa cosa viene effettuata anche per SCAD. In questo caso si seleziona un valore di λ pari a 0.005617783, che consente di selezionare 89 variabili; anche in questo caso, il modello corrispondente cade nella regione di non convessità.

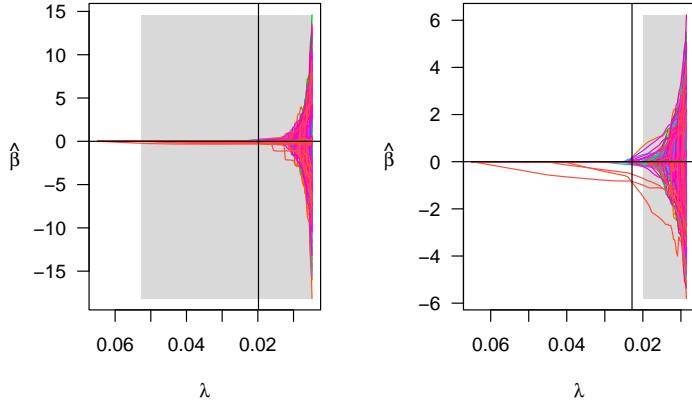


Figure 11: Andamento dei coefficienti di SCAD (grafico a sinistra (a.)) e MCP (grafico a destra (b.)) al variare di lambda per il fused lasso

Non si sono ottenuti risultati soddisfacenti con le etichette Kaggle. Come visibile in Figura 12, infatti, l'andamento del Kappa di Cohen ha un comportamento sostanzialmente asimmetrico tra test set e cross validation. Si può notare come, basandosi sul Kappa massimo ottenuto in cross validation, si tenda a scegliere un λ nel test set che rappresenta un minimo locale, quando in realtà il modello migliore nel test set è molto più complesso.

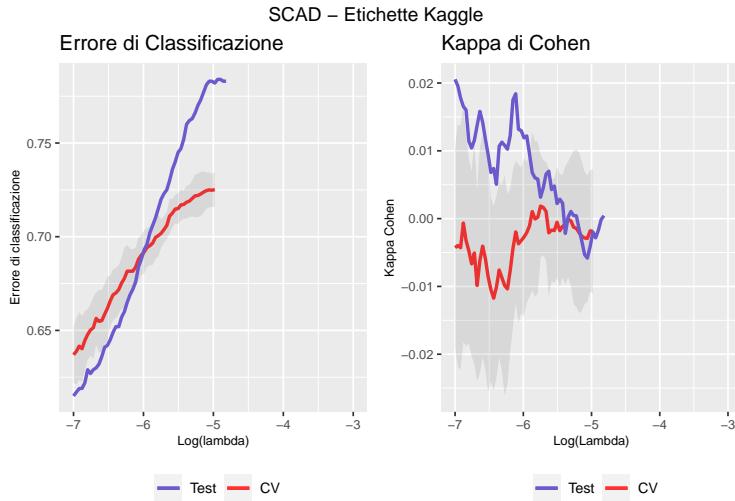


Figure 12: Confronto tra errore di classificazione e kappa di cohen in CV e sul test set su SCAD con le labels Kaggle

4.4 SVM

Si adatta poi un SVM sparso con penalità *elastic net*, selezionando prima α e poi λ . Si utilizza una griglia di λ più ristretta, aspettandosi un comportamento simile a quello del lasso.

Per quanto riguarda le etichette fused, viene selezionato in questo caso un α pari a 1, che corrisponde quindi

ad una penalità L1. Il corrispondente λ selezionato tramite successiva convalida incrociata risulta 0.0432. Inaspettatamente, questo modello seleziona ben 5.163 covariate, molte di più rispetto agli altri metodi; inoltre, l'errore di classificazione in cross validation risulta essere maggiore rispetto a quello riscontrabile nel test set.

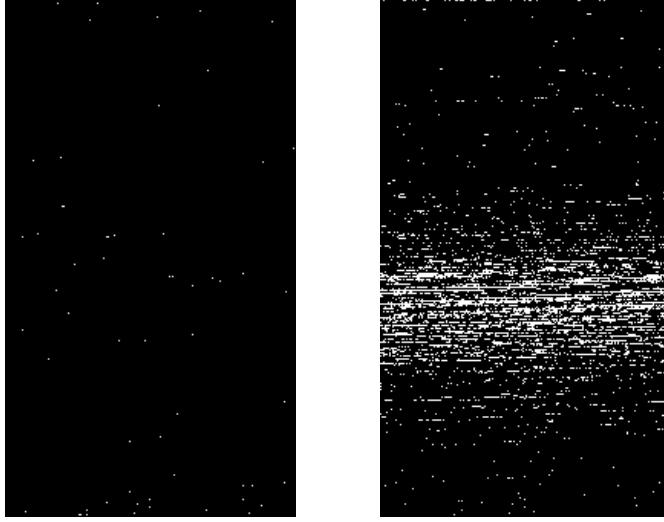


Figure 13: Rappresentazione dei pixel selezionati dal Lasso (a sinistra, (a.)) e dall'SVM (a destra (b.)) relative alle etichette fused

In Figura 13.b sono visibili i pixel selezionati dall'SVM sempre per quanto riguarda le etichette fused. Si nota una notevole concentrazione nella parte centrale dell'immagine. Interessante risulta anche il confronto con i pixel selezionati dal lasso.

Adattando il modello alle etichette Kaggle, si individua un valore di α ottimo pari a 0.7 e un λ pari a 0.1223328. Il numero di coefficienti diversi da 0 selezionati è 150.

Per quanto riguarda le etichette k-means, la scelta dei parametri viene effettuata su un insieme di validazione composto dai giorni 1, 2 e 5. Con tale procedura si ottiene un valore di α pari a 0.8. Si ristima pertanto un modello su una griglia di $\log(\lambda)$ che va da -4.5 a -2, scegliendo, tramite la massimizzazione del Kappa di Cohen come precedentemente descritto, un valore di $\log(\lambda)$ pari a -2.833333, portandoci a selezionare un modello con una complessità elevata, in particolare con 15 972 coefficienti. Come si può notare in Figura 14.b, in corrispondenza del massimo per il Kappa di Cohen nell'insieme di validazione (curva blu), si ottiene un massimo locale sull'insieme di verifica (curva rossa). Si nota infatti come le due curve sui due diversi insiemi seguano lo stesso andamento. Tale massimizzazione va però in conflitto con l'errore di classificazione; si nota infatti in Figura 14.a come in prossimità di valori di $\log(\lambda)$ pari a -2.833333, l'errore di classificazione nell'insieme di verifica sia intorno al suo massimo.

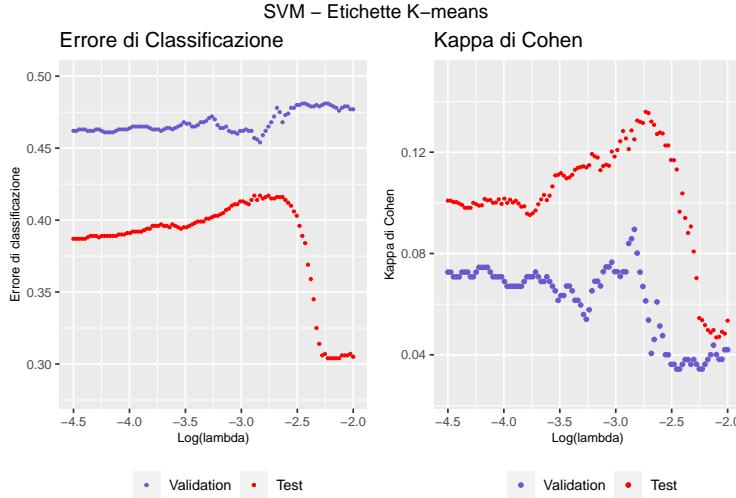


Figure 14: Andamento errore di classificazione e kappa al variare di lambda nell'SVM con etichette k-means

4.5 NSC

Infine, si adatta un modello *Nearest Shrunken Centroids*, in cui si vanno a stimare i centroidi, penalizzandoli, per poi utilizzarli con la regola del centroide più vicino. Questo in genere consente di selezionare solo un piccolo sottoinsieme di centroidi diversi da 0 (e conseguentemente di variabili).

Al contrario degli altri modelli, per i quali la selezione de parametri di regolarizzazione viene effettuata tramite cross validation, in questo caso si effettua tale selezione sull'insieme di stima, in quanto la cross validation porta a scegliere il modello più semplice che porterebbe ad una classificazione casuale e alla previsione di una sola classe.

Per le etichette fused, come è possibile vedere dai grafici, dopo aver centrato e scalato i dati, solo una piccola porzione di centroidi per ogni classe risulta essere diversa dai centroidi generali. Ciò è coerente con i risultati forniti dai modelli precedenti. Per entrambe le classi vengono selezionate le stesse variabili, che sono in totale 73.

```
# NSC -----
result_Fused <- list(x = t(scale(X_train)), y = factor(train$Etichette_Fused_K2))
# Definisco una lista contenente la matrice del disegno e la variabile risposta

nsc_Fused <- pamr.train(result_Fused)
# Stimo il modello

best.t_Fused <- nsc_Fused$threshold[which.min(nsc_Fused$errors)]
# Delimito la miglior soglia (parametro di regolarizzazione) in base all'errore
# ottenuto sul train

nsc.opt_Fused <- pamr.train(result_Fused, threshold = best.t_Fused)
# Ristimo il modello per la soglia ottimale
```

Sulle etichette k-means e kaggle, invece, non è stato possibile ricondursi a risultati soddisfacenti, conseguentemente non si riportano i relativi risultati.

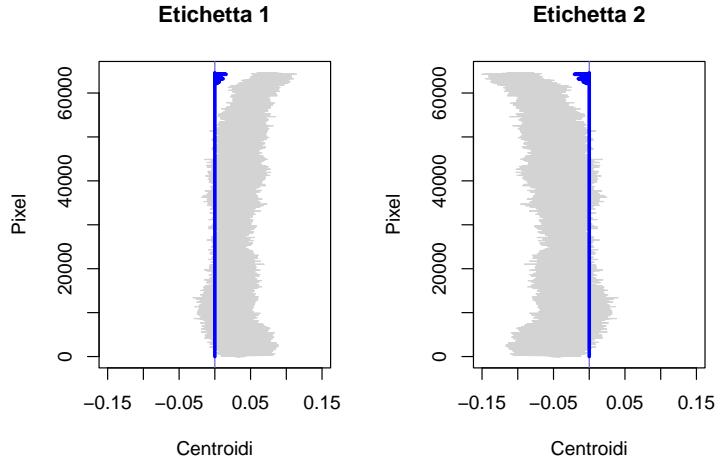


Figure 15: Grafico NSC per etichette fused: in blu i centroidi ristretti (shrunken), in grigio la media specifica della classe per ogni pixel

4.6 Risultati finali

Si analizzano per prima cosa le metriche derivanti dalle etichette fused lasso.

Table 2: Risultati Fused

	E.Net	E.Net SVM	Lasso	MCP	NSC	SCAD
Errata.Classificazione	0.4	0.428	0.391	0.42	0.368	0.392
Accuratezza	0.6	0.572	0.609	0.58	0.632	0.608
Sensibilità	0.208	0.338	0.159	0.197	0.023	0.188
Specificità	0.831	0.71	0.874	0.805	0.989	0.854
False.positive.rate	0.169	0.29	0.126	0.195	0.011	0.146
False.negative.rate	0.792	0.662	0.841	0.803	0.977	0.812
False.Disc.Rate	0.448	0.462	0.442	0.497	0.324	0.437
Precisione	0.552	0.538	0.558	0.503	0.676	0.563
F1.Score	0.278	0.369	0.232	0.258	0.045	0.262
Kappa di Cohen	0.043	0.05	0.038	0.003	0.016	0.048
Coefficienti diversi da 0	116	5163	60	27	73	31

Come visibile dalla Tabella 2, considerando il tasso di errata classificazione, il modello migliore risulta essere NSC, anche se più del 98% delle osservazioni vengono predette da esso come 1, risultato che non ci si augurerebbe da un buon modello. I restanti modelli producono approssimativamente gli stessi risultati. In particolare, SCAD è in grado di ottenere lo stesso errore di errata classificazione pur utilizzando 31 variabili, quando gli altri modelli ne tengono un numero superiore.

Infine, si può notare come SVM produca il Kappa di Cohen più elevato tra tutti i modelli, nonostante l'accuracy sia la più bassa. Questo indica che le previsioni sono più bilanciate e maggiormente in grado di discriminare tra le due classi, come visibile dai valori di sensibilità e specificità, diversi rispetto agli altri modelli.

Passando poi alle etichette k-means, come è possibile vedere dalla Tabella 3, l'errore di classificazione rimane in media simile per tutti i modelli stimati con le etichette fused, fatta eccezione per SCAD e MCP che peggiorano visibilmente. Tuttavia, si ottiene con l'Elastic Net SVM il Kappa di Cohen maggiore tra tutti

i modelli stimati, a scapito però di un elevato numero di coefficienti. Tenendo in considerazione l'errore di classificazione, il Lasso ottiene i risultati migliori, con un errore pari a 0.378. SCAD e MCP, pur ottenendo un errore di classificazione più alto, che si attesta circa a 0.48, consentono un miglior bilanciamento di sensibilità e specificità, assieme anche un numero di coefficienti basso, attorno agli 80.

Table 3: Risultati K-Means

	El. Net SVM	El. Net	Lasso	MCP	SCAD
Errata.Classificazione	0.417	0.381	0.378	0.487	0.485
Accuratezza	0.583	0.619	0.622	0.513	0.515
Sensibilità	0.767	0.753	0.783	0.491	0.494
Specificità	0.353	0.286	0.222	0.568	0.567
False.positive.rate	0.647	0.714	0.778	0.432	0.433
False.negative.rate	0.233	0.247	0.217	0.509	0.506
False.Disc.Rate	0.458	0.487	0.498	0.468	0.467
Precisione	0.542	0.513	0.502	0.532	0.533
F1.Score	0.671	0.738	0.747	0.59	0.593
Kappa di Cohen	0.125	0.04	0.006	0.047	0.049
Coefficienti diversi da 0	15972	832	634	83	89

Infine, per quanto riguarda le etichette Kaggle, come già detto in precedenza, MCP e SCAD non sono stati tenuti in considerazione poiché non fornivano risultati sufficienti. Si riportano quindi i risultati relativi a Elastic Net, con SVM e glmnet, e il Lasso.

Rispetto alle altre etichette, queste in oggetto, forniscono i risultati peggiori, con errore di classificazione bassi associati a specificità molto elevata. Infatti, anche il Kappa di Cohen risulta essere negativo o leggermente positivo in tutti i casi.

In generale le etichette Kaggle portano anche ad un numero di variabili selezionate più elevato rispetto alle altre etichette, a riprova della scarsa efficacia dei modelli.

Table 4: Risultati Kaggle

	El. Net SVM	El. Net	Lasso
Errata.Classificazione	0.226	0.232	0.36
Accuratezza	0.774	0.768	0.64
Sensibilità	0.013	0.253	0.235
Specificità	0.985	0.785	0.751
False.positive.rate	0.015	0.215	0.249
False.negative.rate	0.987	0.747	0.765
False.Disc.Rate	0.536	0.459	0.514
Precisione	0.464	0.541	0.486
F1.Score	0.024	0.063	0.22
Kappa di Cohen	-0.004	0.01	-0.013
Coefficienti diversi da 0	738	2476	150

4.3 Dataset Alternativi

Infine, si è deciso di applicare i modelli migliori per ciascuna etichetta ai relativi dataset modificati. Per ogni etichetta si adatta un modello Elastic Net, in quanto aveva portato a risultati più stabili; per le etichette Fused si adatta anche NSC, in quanto aveva mostrato una buona performance.

Per quanto riguarda le immagini su cui è stato applicato il *Fused Lasso Signal Approximator*, i risultati ottenuti sono riportati in Tabella 5. Si può notare come Elastic Net per le etichette fused peggiori leggermente l'errore di classificazione rispetto all'analogo sulle immagini originali; rimane invece pressoché identico per l'Elastic Net effettuato sulle etichette k-means e per l'NSC sulle etichette Fused. Non si riportano i risultati per l'Elastic Net effettuato sulle etichette Kaggle in quanto non hanno consentito una classificazione soddisfacente. In genere quindi non si può affermare che questa tecnica sia vantaggiosa in quanto porta anche a ridondanza dei dati e variabili vicine tra loro saranno simili.

Table 5: Risultati con dataset FLSA

	El. Net Fused	El. Net Kmeans	NSC Fused
Errata.Classificazione	0.423	0.378	0.37
Accuratezza	0.577	0.622	0.63
Sensibilità	0.398	0.772	0.5
Specificità	0.639	0.248	0.633
False.positive.rate	0.361	0.752	0.367
False.negative.rate	0.602	0.228	0.5
False.Disc.Rate	0.476	0.493	0.423
Precisione	0.524	0.507	0.577
F1.Score	0.327	0.745	0.065
Kappa di Cohen	0.033	0.021	0.018
Coefficienti diversi da 0	70	167	438

Table 6: Risultati con dataset con immagini ridotte

	El. Net Fused	El. Net Kaggle	El. Net Kmeans	NSC Fused
Errata.Classificazione	0.37	0.248	0.335	0.378
Accuratezza	0.63	0.752	0.665	0.622
Sensibilità	0.6	0.177	0.876	0.436
Specificità	0.63	0.782	0.139	0.634
False.positive.rate	0.37	0.218	0.861	0.366
False.negative.rate	0.4	0.823	0.124	0.564
False.Disc.Rate	0.381	0.552	0.496	0.456
Precisione	0.619	0.448	0.504	0.544
F1.Score	0.006	0.065	0.789	0.128
Kappa di Cohen	0.002	-0.015	0.018	0.022
Coefficienti diversi da 0	42	45	439	141

Per quanto riguarda invece le immagini ridotte, i risultati sono visibili nella Tabella 6. In termini di accuratezza, i risultati non variano particolarmente rispetto al dataset iniziale per l'Elastic Net adattato sulle etichette kaggle e per il NSC sulle etichette fused. Si nota invece un leggero miglioramento per l'Elastic Net sulle etichette fused, che da 0.6 va a 0.63, e sulle etichette k-means, in cui l'accuracy aumenta da 0.619 a 0.665.

5. Conclusione

Per concludere, si nota che, com'era da aspettarsi, nessuno dei modelli abbia una particolare capacità predittiva. I modelli adattati in questa sede, infatti, non tengono in considerazione le interazioni e la dipendenza sia tra osservazioni che l'ordinamento delle variabili.

Particolarmente efficace, rispetto alle altre etichette, sono quelle definite tramite *Fused Lasso Signal Approximator*, che, oltre a rendere l'insieme di dati bilanciato, consente di stimare modelli che pur non raggiungendo errori minimi, forniscono un Kappa di Cohen sempre positivo.

Pur essendo a conoscenza della scarsa capacità interpretativa dei modelli adattati a questo tipo di dati, è stato comunque di interesse verificare se fossero presenti dei pattern nella selezione delle variabili nei diversi modelli.

Per questioni di tempo, prolixità del progetto, scarse performance dei modelli o limiti computazionali, si è deciso di accantonare l'adattamento di modelli con etichette multiclass (FLSA e k-means, che sono comunque state definite), nonché ulteriori riduzioni della dimensionalità del dataset, come ad esempio attraverso PCA sparsa e denoising di immagini.

6. Bibliografia

- *Statistical Learning with Sparsity: The Lasso and Generalizations.* (7 maggio 2015) Libro di Martin J. Wainwright, Robert Tibshirani e Trevor Hastie.
- *Off-Road Terrain Dataset for Autonomous Vehicles - A dataset with images from a monocular camera and relevant sensor data.* <https://www.kaggle.com/datasets/magnumresearchgroup/offroad-terrain-dataset-for-autonomous-vehicles>
- Gresenz, G., White, J., & Schmidt, D.C. (2021). “An Off-Road Terrain Dataset Including Images Labeled With Measures of Terrain Roughness.” **Proceedings of the IEEE International Conference in Autonomous Systems**, 309-313.