Report Challenge 2

Elisa Coceani

16 aprile 2024

1 Introduzione

Lo scopo di questa challenge è applicare i metodi di kernel alla regressione Ridge e all'algoritmo PCA, analizzare i risultati e trovare il metodo migliore per il caso specifico.

2 Ridge Regression vs Kernel Ridge

Un modello di regressione Ridge, applicato al primo dataset, non forsnisce un risultato soddisfiacente: con un valore dell'ipeparametro alpha pari a 1 (ma gli stesssi risultati valgono per valori inferiori a 1) l'RMSE resta intorno a 26.81.
Utilizziamo due kernel diversi per migliorare le performance delle regressione Ridge.

- Kernel Gaussiano: Il modello migliore ha parametri **gamma = 1** e **alpha = 0.01**. Con questi parametri il modello ha un errore RMSE di 0.84, nettamente migliore del modello di regressione ridge senza il kernel gaussiano.
- Kernel Polinomiale: Il modello migliore trovato ha parametri **gamma = 0.1** e **15 degrees**. Con questi parametri il kernel polinomiale performa meglio del kernel gaussiano arrivando ad un rmse di 0.69.

Un'osservazione importante riguardo al parametro alhpa, associato alla regolarizzaizone della ridge regression, è che diminuendo il valore di questo parametro non si corre il rischio di overfitting, questo perché i dati sono stati generati con un noise relativamente piccolo distribuito secondo una normale standard.

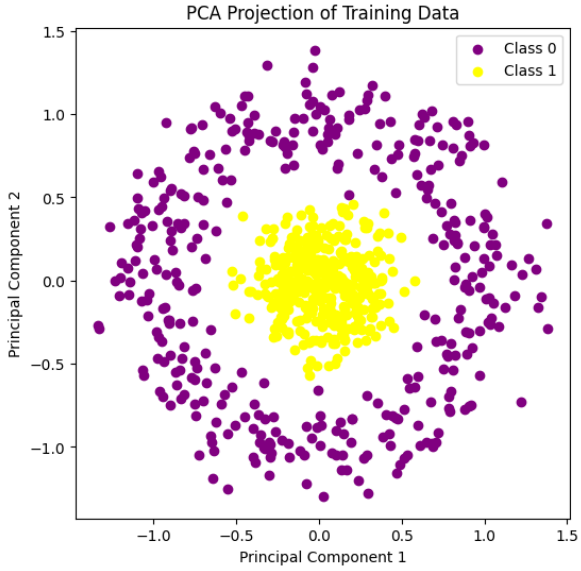
2.1 Numero di training points

Nel dataset iniziale sono stati forniti 20 punti di training, un numero così ridotto può influire sui training dei modelli. Aumentando il numero di punti i risultati non cambiano per quanto riguarda la Ridge Regression, mentre per i metodi di kernel vediamo che con un numero di punti pari a 1000 L'RMSE diminuisce, migliorando i modelli. Aumentando il numero di punti il kernel gaussiano diventa il modello miglio con un RMSE di 0.12, a confronto del kernel gaussiano con RMSE di 0.35.

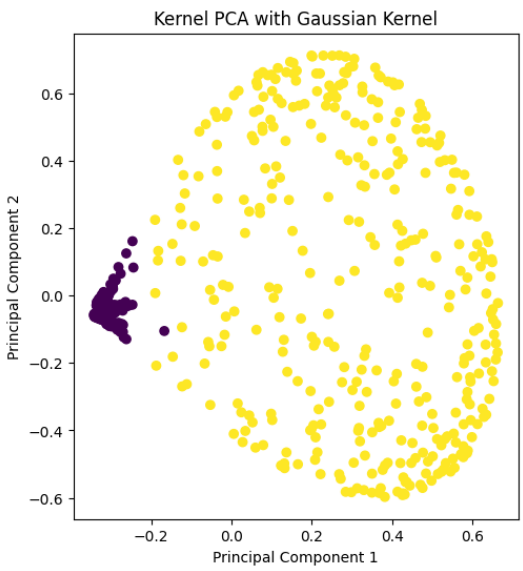
3 PCA vs Kernel PCA

Mettiamo a confronto PCA con Kernel PCA utilizzando un dataset creato con sklearn.datasets.make_circles(). Il dataset è formato da due classi che si dispongono su due cerchi concentrici, questo ci suggerisce che non sono linearmente separabili. Infatti PCA non riesce a trovare due componenti principali per ridurre la dimensionalità.

Vista la natura dei dati un kernel gaussiano potrebbe essere la scelta migliore: applicando PCA con un kernel gaussiano (gamma = 10) troviamo il seguente risultato.



(a) PCA



(b) Kernel PCA

Verifichiamo l'accuracy di entrambi i metodi con SVM:

- PCA: 0.47
- Gaussian Kernel PCA: 0.99

In conclusione kernel PCA con kernel gaussiano è la scelta più appropriata per il dataset in analisi.

3.1 Kernel PCA con sklearn.datasets.make_classification()

La funzione make_classification() genera dati per problemi di classificazione dove è possibile specificare diversi parametri: il dataset generato ha 1000 punti suddivisi in due classi.

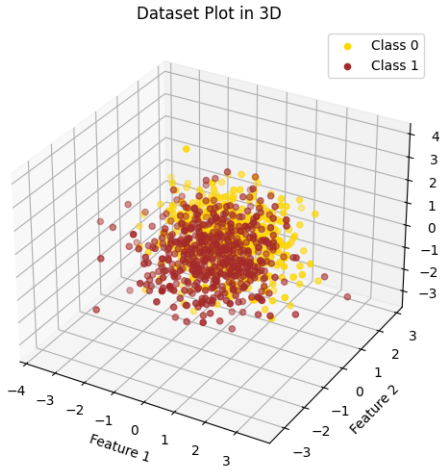
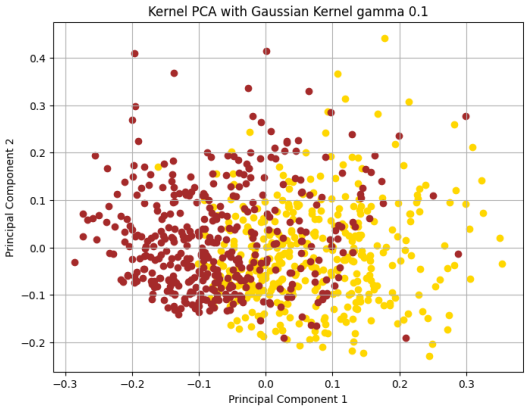
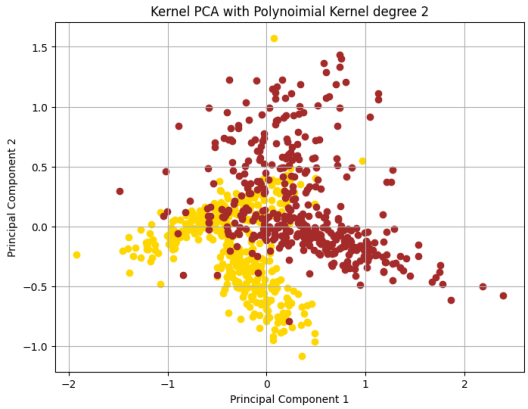


Figura 2: Dataset

Di seguito vengono riportati due grafici del kernel PCA con kernel gaussiano e polinomiale:



(a) Kernel Gaussiano



(b) Kernel Polinomiale

Il kernel polinomiale con 3 gradi ha un accuracy del 88%, leggermente migliore del kernel gaussiano, quindi si adatta meglio a questo specifico dataset.

Anche se non riusciamo a trovare un kernel adatto si potrebbe provare a modificare i parametri del dataset, ad esempio aumentando il parametro class_sep che regola la separazione delle due classi del dataset.