



Università degli Studi di Milano Bicocca

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea magistrale in Informatica

ASSIGNEMENT 5

Advance Machine Learning

Hyperparameter Optimization of a SVM Classifier

Andrea Guzzo, 761818

Descrizione dell'assegnement

L'assegnement consiste nel tentativo di fare Hyperparameter Optimization su un classificatore SVM implementato con un kernel sigmoid, in modo da massimizzare l'accuratezza in una 10 Fold Cross Validation considerata come funzione obiettivo.

L'analisi compiuta è reperibile anche al seguente indirizzo su GitHub:

<https://github.com/JeyDi/BayesianMLOptimization>

Il Dataset utilizzato per gli esperimenti è chiamato "wine" ed è stato ottenuto nella versione presente all'interno della libreria python: sklearn

Il codice è disponibile all'interno del pacchetto consegnato.

Soluzione e Risultati

La soluzione proposta prevede la seguente inizializzazione degli hyperparameters:

C: da 0,1 a 5

Gamma: da 0,1 a 10

Coef0: da 0,1 a 10

Per una descrizione degli iperparametri, si rimanda alle seguenti documentazioni:

<http://pyml.sourceforge.net/doc/howto.pdf>

<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

È stato impostato un seed a 42 in modo da ripetere gli esperimenti con lo stesso punto di partenza per la generazione dei numeri pseudocasuali.

L'initial point è stato impostato a 30, mentre il further evaluations (sequenzial optimization) a 120 come da consegna per avere un overall budget di 150 funzioni da valutare e considerare.

L'evoluzione del best seen durante la computazione per i vari metodi non ha subito un sostanziale incremento, rimane stabile nell'intorno di 0.4 come si può verificare dai risultati ottenuti nei vari esperimenti

```

(OrderedDict([('C', 3.188207174058357), ('gamma', 0.1), ('coef0', 2.773493726928153)]), 0.43390092879256964)
Gaussian Model 1 Time: -1551.4342771

(OrderedDict([('C', 3.188207174058357), ('gamma', 0.1), ('coef0', 2.773493726928153)]), 0.43390092879256964)
Gaussian Model 2 Time: -1396.0705223000002

(OrderedDict([('C', 3.188207174058357), ('gamma', 0.1), ('coef0', 2.773493726928153)]), 0.43390092879256964)
Gaussian Model 3 Time: 0.0

(OrderedDict([('C', 2.5436345570967553), ('gamma', 5.4614937894235736), ('coef0', 0.5554213352021627)]), 0.43390092879256964)
Random Forest 1 Time: -184.2613707999999

(OrderedDict([('C', 2.5436345570967553), ('gamma', 5.4614937894235736), ('coef0', 0.5554213352021627)]), 0.43390092879256964)
Random Forest 2 Time: -207.28545630000008

(OrderedDict([('C', 2.5436345570967553), ('gamma', 5.4614937894235736), ('coef0', 0.5554213352021627)]), 0.3992539559683522)
Random Forest 3 Time: -302.4150015

```

Non abbiamo quindi notato dei sostanziali miglioramenti tra i vari modelli Gaussiani, ma invece abbiamo notato una sostanziale differenza rispetto ai Random Forest per i relativi Hyperparameters.

Anche il Best Seen del Grid search si presenta simile ai risultati ottenuti con i metodi precedenti.

```
Grid Search, Best on Grid:0.43%%
```

Infine, il tempo di computazione impiegato nei vari esperimenti è molto variabile, dopo diversi esperimenti, il tempo migliore raggiunto è il seguente:

Esperimento	Modalità di acquisizione	Tempi (secondi)
Gaussian Model 1	Expected Improvement	1551,43
Gaussian Model 2	Probability Improvement	1396,07
Gaussian Model 3	UCB	0
Random Forest 1	Expected Improvement	184,26
Random Forest 2	Probability Improvement	207,28
Random Forest 3	UCB	302,41
Grid Search		17,93

L'esperimento ha avuto successo, ma i risultati ottenuti non sono significativi per evidenziare un miglioramento sostanziale nell'ottimizzazione del classificatore così come è stato configurato.