



Assignment V

1920-2-F1801Q151, Advance Machine Learning, 2019-2020

Università degli studi di Milano Bicocca

Dipartimento di Informatica, Sistemistica e Comunicazione

Villa Giacomo 807462

9 Dicembre 2019

Indice

1	Dataset	3
2	Punto uno: Ottimizzazione due iper-parametri	4
2.1	Introduzione	4
2.2	Implementazione e Confronto	5
3	Punto due: Ottimizzazione quattro iper-parametri	9
3.1	Introduzione	9
3.2	Implementazione e Confronto	9
4	Conclusioni	10

1 Dataset

Per questo assignment è stato richiesto di **ottimizzare, mediante SMBO**, degli **iperparametri** di una **rete neurale** al fine di **massimizzare l'accuracy su 10 cross-validation** di un task di classificazione su un dataset fornito; veniva richiesto anche un **confronto** con i metodi di **Grid Search** e **Random Search**.

In particolare il **dataset** conteneva **dati riguardanti donne tra i 18 e 36 anni al fine di valutarne la fertilità**; gli attributi presi in considerazione riguardano diverse informazioni pertinenti alla donna in questione, come ad esempio: malattie infantili, incidenti, interventi chirurgici, febbre alta nell'ultimo anno, frequenza di consumo dall'alcool, ecc.

È stata **effettuata un'operazione di normalizzazione** dei dati mediante la funzione *StandardScaler()* della libreria *Sklearn*. L'assignment è stato svolto utilizzando *python 3.7* con la libreria *pyGPGO* per implementare SMBO e Sklearn per la rete neurale. È stato effettuato uno **studio riguardante il dataset** e sono state registrate **due grandi criticità**; la **prima** è sicuramente la **distribuzione dei record**:

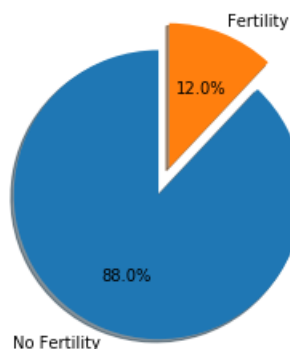


Figura 1: Distribuzione record

Possiamo osservare come i record associati alla classe Fertility (classe 1) risulti essere pari al 12% mentre quelli appartenenti alla classe No Fertility (classe 0) risulti essere pari al 88%; tale **sbilanciamento** è **ancora più accentuato dalla seconda criticità** incontrata durante la fase di analisi.

La **seconda** criticità riguarda la **numerosità del dataset**, questo è dotato di **solo 100 istanze**; osservando la riga della **matrice di correlazione**, riguardante

l'attributo di classe:



Figura 2: Correlazione degli attributi in funzione dell'attributo di classe

Possiamo affermare che **non sembrano esserci attributi altamente discriminativi**. Risulta possibile affermare che il **task di classificazione** proposto risulterà essere **molto complicato**.

2 Punto uno: Ottimizzazione due iper-parametri

2.1 Introduzione

Il **primo punto dell'assignment** richiedeva un **approccio SMBO** al fine di effettuare la **migliore scelta per gli iper-parametri di learning rate e momentum**, per poi passare a un **confronto con Grid Search e Random Search**; il **budget** di cui si è dotati è pari a **25 iterazioni** (di cui **5 punti iniziali comuni** e **20 iterazioni** per quanto riguarda SMBO). In questa fase, come nella successiva, è stata utilizzata la libreria **Sklearn per implementare la rete neurale MLP**, **non sono stati effettuati attribuzioni di pesi alle classi al fine di avvantaggiare la classe di minoranza**.

La **struttura della rete** vede **9 neuroni di input** (numero degli attributi della rete), **4 neuroni nascosti nel primo layer**, **2 nel secondo** e un **neurone di output**. Viene utilizzato **SGD come ottimizzatore**, il quale permette di fissare i parametri di learning rate e momentum.

Per quanto riguarda la **funzione di attivazione per i layers nascosti** è stata scelta la **ReLU**; scelta giustificata dallo stato dell'arte e dal fatto che questa permette di ottenere un apprendimento relativamente più rapido, grazie alla sua natura, rispetto ad altre funzione di attivazione.

Per la **funzione di output** si è deciso di utilizzare la **sigmoide** al fine di decretare l'output binario della rete, infine per la **funzione di loss** si è scelto la **binary cross entropy**: sostanzialmente una cross entropy (quindi una quantificazione della differenza tra due distribuzioni di probabilità) applicata a una realtà binaria dato il problema.

Il confronto tra i modelli SMBO ha visto contrapporsi la funzione di acquisizione Expected Improvement e Probability Improvement, in seguito si è passati a un confronto con i metodi di Grid Search e Random Search.

Il modello SMBO utilizzato per questo punto è il **Gaussian Process**, tale scelta è stata guidata dal fatto che gli **iper-parametri da ottimizzare risultano essere solo continui**.

2.2 Implementazione e Confronto

Fissati, come da richiesta, **i 5 punti comuni** ai due processi si è **passato alla fase di scelta di nuovi punti** tante volte quanto richiesto (20 iterazioni).

I risultati ottenuti, confrontando le due funzione di acquisizione, sono i seguenti:

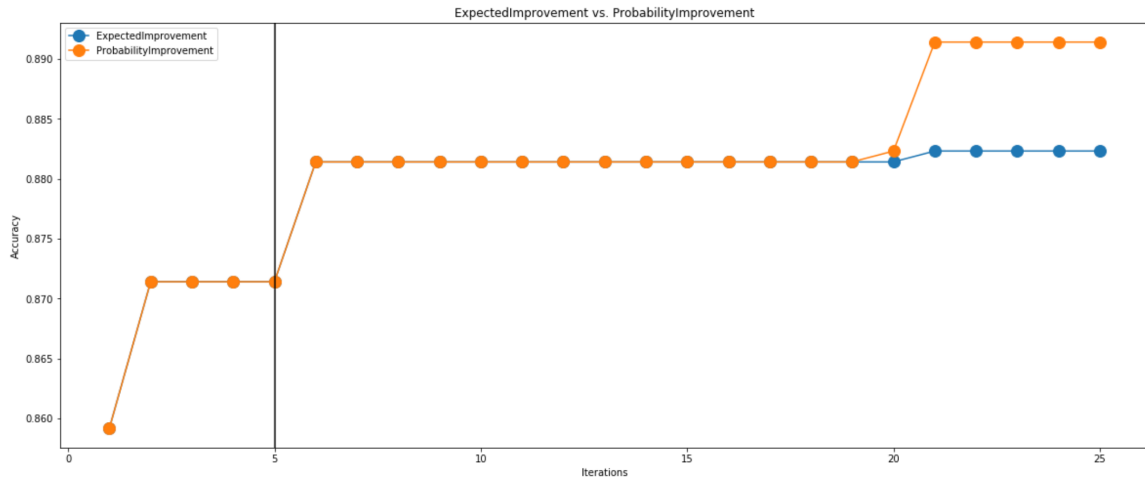


Figura 3: Best Seen EI vs PI

La **linea nera verticale** in prossimità della 5 iterazione denota la **divisione tra le performance della configurazione iniziale** comune ai due modelli e **le performance dei punti selezionati**, date le funzioni di acquisizione definite, **dai due modelli**.

Possiamo osservare come i **risultati siano abbastanza deludenti**: l'**accuracy pari all'89% non deve trarre in inganno**; questo valore è dovuto al fatto che l'**88% degli attributi appartiene alla classe 0**, utilizzando una **10 cross-validation stratificata** questo valore è mantenuto nei vari fold. Quello che succede è che **si tende ad attribuire a tutto l'etichetta 0**, dunque il risultato ottenuto si discosta

di poco dall'approccio: "etichetta tutto con 0". In ogni caso **le configurazioni iniziali portano a un valore inferiore** rispetto alla soglia dell'88%, dunque **vi sono punti della funzione nascosta che rappresentano di fatto un minimo**; dunque la funzione obiettivo (accuracy) non è del tutto piatta.

In questo caso possiamo osservare come un **approccio basato sulla Probability Improvement permetta di selezionare il punto ottimo**, dunque la valutazione dei punti in funzione della probabilità che questi siano migliori dell'ottimo corrente, porta a risultati migliori rispetto a valutare i punti in funzione del miglioramento atteso.

La **configurazione che permette**, per prima, l'ottenimento delle **performance migliori** è **Learning Rate: 0.01031** e **Momentum: 0.67986** per una **accuracy pari a 0.89141**. La **Precision della classe di minoranza** va a un **minimo di 0** a un **massimo di 0.2**, per quanto riguarda la **Recall** si parte da un **minimo di 0** a un **massimo di 0.2**, nella maggior parte dei casi i valori sono pari a 0; risultati alquanto scadenti.

Sono state **eseguite** anche le **ricerche mediante Grid Search e Random Search**, i risultati di confronto finale sono i seguenti:

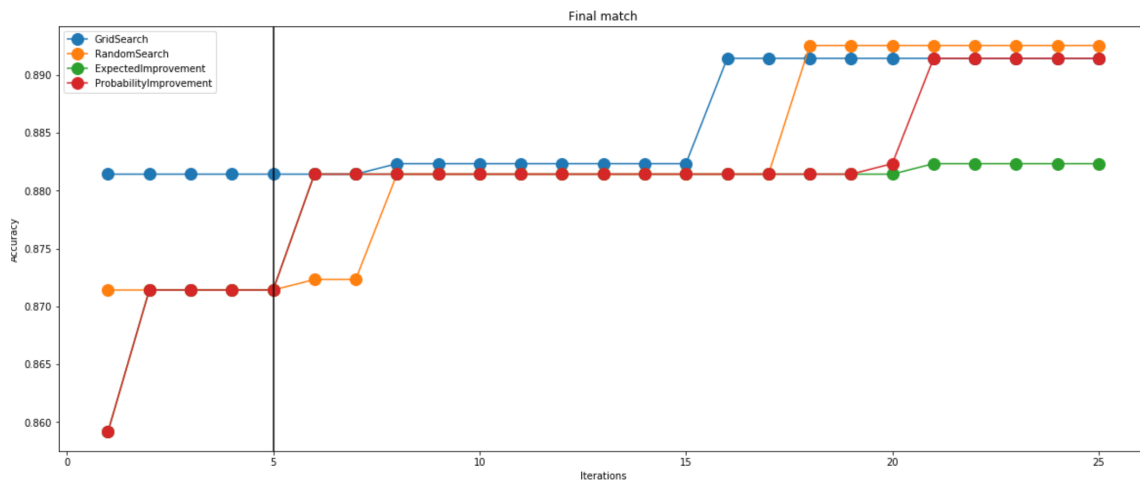


Figura 4: Final Match

In questo caso la **linea nera verticale** indica ancora la **differenza tra punti iniziali comuni** ai modelli SMBO e **punti selezionati successivamente**; tuttavia i punti iniziali sono comuni solo ai modelli SMBO, **Grid Search e Random Search**

non li condividono.

La **configurazione ottima** è ottenuta dall'approccio basato sul **Random Search**, anche l'approccio **Grid Search** ottiene risultati migliori o uguali (dipende da quale approccio SMBO si guarda) rispetto agli altri metodi. L'**ottimo** è ottenuto mediante la **configurazione con Learning Rate pari a 0.02769 e Momentum pari a 0.60945**; l'accuracy ottenuta è pari a **0.89253**.

Ho provveduto anche a **recuperare le 25 configurazioni** provate da tutti e **4 le metodologie** (due modelli SMBO, Grid e Random Search) al fine di **visualizzare lo spazio di ricerca**.

I risultati da me ottenuti sono i seguenti:

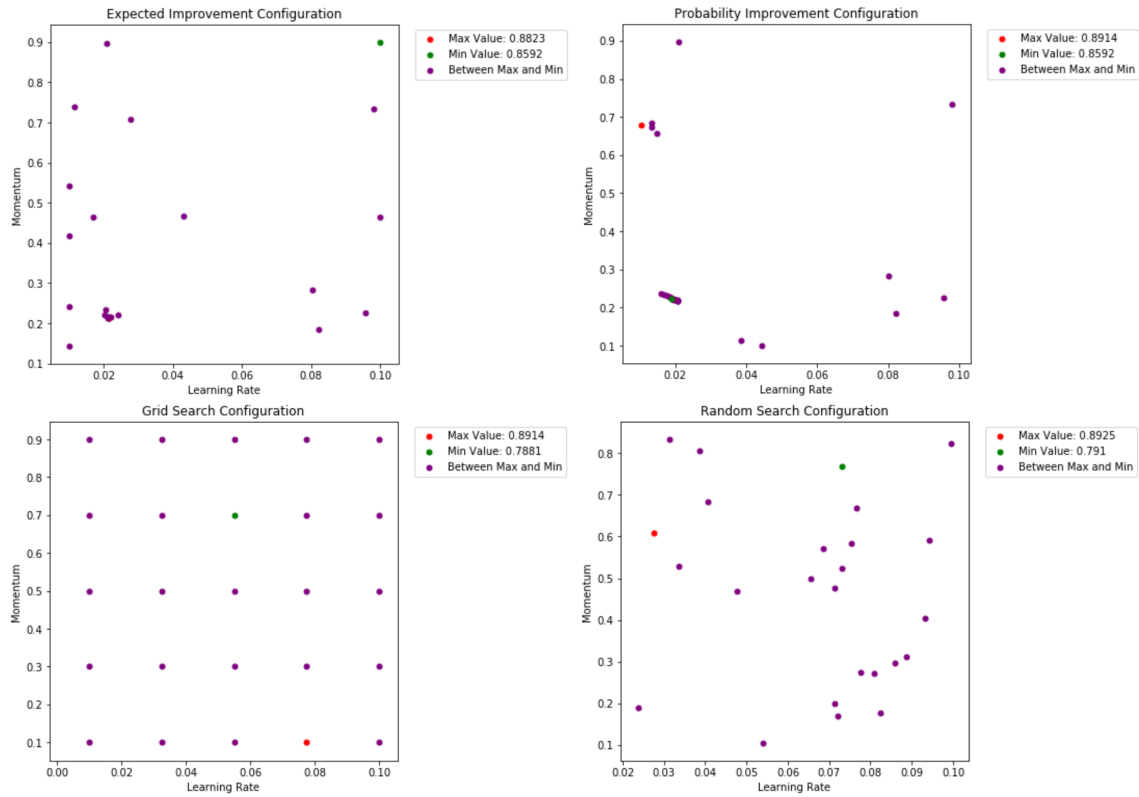


Figura 5: Configurazioni provate

Per quanto riguarda il **grafico Expected Improvement**, il **valore massimo ottenuto si discosta di pochissimo dal baseline (88%)**, osservando con attenzione

l'area densa intorno al punto $(0.02, 0.2)$ è possibile osservare il colore rosso del massimo.

Possiamo osservare come l'**andamento Expected Improvement risulti essere più sparso nell'area di ricerca a differenza di Probability Improvement** che tende a concentrarsi di più in certi punti; il minimo trovato da questo approccio è stato identificato dopo una serie di iterazioni nell'area dove è evidenziato, in seguito all'identificazione dello stesso ci si è spostati verso l'area dell'ottimo, identificandolo.

L'approccio Grid e Random Search non necessitano di particolari commenti.

3 Punto due: Ottimizzazione quattro iper-parametri

3.1 Introduzione

Nel **secondo punto dell'assignment** veniva richiesto di **ottimizzare**, oltre agli iper-parametri precedenti, **anche il numero di neuroni** al primo e al secondo layer; in particolare **questi valori sarebbero dovuti oscillare tra 1 e 4 per il primo e tra 1 e 2 per il secondo**.

Come **modello SMBO** è stata utilizzata una **Random Forest**, scelta giustificata dal fatto che **siamo in presenza anche di attributi discreti** (neuroni per i layers nascosti); la scelta è dunque obbligata.

3.2 Implementazione e Confronto

La **struttura della rete** (fatta eccezione per il numero di neuroni) è **la stessa già discussa** nella sezione precedente. In questo caso veniva richiesto **solo un approccio di confronto tra due modelli SMBO** dotati di **due funzioni di acquisizione diverse**; ancora una volta sono state **utilizzate Expected Improvement e Probability Improvement**. Il **numero di iterazioni** richieste è pari a **110** (10 punti iniziali comuni e 100 nuovi punti).

Il grafico Best Seen sulle iterazioni totali è il seguente:

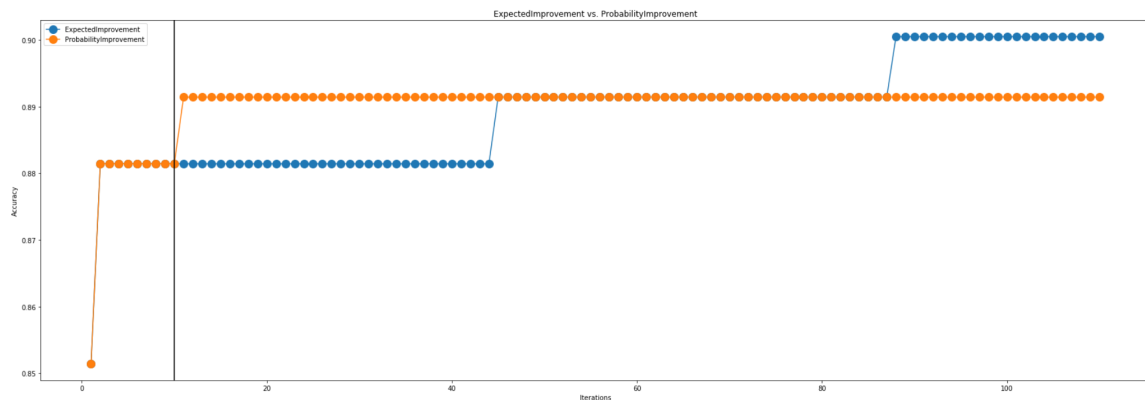


Figura 6: Final Match

Anche in questo caso la linea nera verticale divide le 10 configurazioni

iniziali comuni dalle **configurazioni scelte dai modelli** data la funzione di acquisizione impostata.

I **risultati ottenuti mostrano** come, in questo caso, un **approccio di selezione** dei nuovi punti **basato su un'aspettativa di miglioramento** permetta di ottenere la **configurazione migliore** degli iper-parametri al fine di massimizzare l'accuracy. Inizialmente la funzione di acquisizione Probability Improvement permette di ottenere un ottimo, tuttavia in seguito non riesce trovare un punto migliore. La **configurazione migliore** risulta essere la seguente:

- 3 neuroni primo layer.
- 1 neurone secondo layer.
- 0.09647 di learning rate.
- 0.41509 di momentum.

L'accuracy ottenuta con la configurazione sopra è pari a **0.90**; i valori di precision e recall per la classe di minoranza risultano essere sempre non soddisfacenti.

4 Conclusioni

I **risultati ottenuti non sono molto soddisfacenti**, l'ottimizzazione in questo contesto è molto difficoltosa in quanto il dataset fornito è dotato di diverse problematiche. La scelta di SMBO, Grid Search o Random Search non vede differenza alcuna nel caso in cui si decidesse di non pesare le classi. Al contrario sembra che approcci basati su Random Search possano fornire **risultati migliori con il budget e il dataset attuali**; possiamo quindi immaginare la **funzione obiettivo** (accuracy), composta da un'area **piatta** che porta una accuracy a 0.88% con delle **aree di minimo** e **piccole aree di massimo** che portano a valori, di poco, superiori a 0.88%.

Questa **situazione è sicuramente causata dal dataset**, di fatto questo è profondamente sbilanciato e di dimensione molto ridotte; **le reti neurali indotte non sono in grado di apprendere e dunque inferire correttamente**. Sarebbe **necessario un dataset di dimensioni più importanti e meno sbilanciato** rispetto a quello in nostro possesso.

Sarebbe **interessante provare**, per quanto concerne il punto uno dell'assignment, a

“sparare” in maniera random un grande numero di configurazioni; in seguito procedere a **colorare in funzione dell’accuracy** in modo da ottenere, tenendo in considerazione come soglie il minimo dei massimi e il massimo dei minimi ottenuti con gli esperimenti condotti, le aree di minimo e massimo della funzione nascosta.

Similmente sarebbe motivante provare ad utilizzare una nuova funzione al posto dell’accuracy (ad esempio: *balanced accuracy* da Sklearn) e osservare le differenze tra i punti ottenuti rispetto agli esperimenti condotti.