Intelligenza Artificiale (2020/21)

Marco Guerra, Univeristà degli Studi di Firenze 01/12/2021

Riconoscimento di Caratteri

Si vuole utilizzare implementazioni disponibili di Random Forest (RF) e Decision Tree (DT) (scikit-learn in Python) per classificare immagini di caratteri manoscritti del dataset EMNIST Letters. In particolare, si vuole produrre una figura analoga alla Fig. 6 di (Cohen et al.2017) per RF ed una per DT. Si vuole confrontare infine le accuratezze dei due metodi.

Riprodurre i risultati

```
pip install sklearn matplotlib numpy
```

Per produrre le immagini in Figura 4:

```
python decisiontree.py features_reduced.npz labels_original.npz
python decisiontree.py features_reduced.npz labels_corrected.npz
```

Per produrre le immagini in Figura 3:

```
python decisiontree.py features_reduced.npz labels_original.npz
python randomforest.py features_reduced.npz labels_original.npz 150
```

Il Dataset

Il dataset EMNIST completo è reperibile su NIST[1]. In particolare utilizzeremo una parte di questo, costituita di caratteri scritti a mano, chiamata EMNIST Letters, che si compone di un Training Set di 124.800 features e di un Test Set di 20.800.

Ognuna di queste è un'immagine grayscale di 28x28 pixels, raffigurante una lettera dell'alfabeto a 26 lettere.

Le Librerie

Si riportano di seguito le librerie utilizzate:

- Classificatori: DecisionTreeClassifier da sklearn.tree e RandomForestClassifier da sklearn.ensemble.
- Dimensionality Reduction: Principal Component Analysis (PCA) da sklearn.decomposition.
- Elaborazione immagini: Histogram of Oriented Graphs (hog) da skimage.feature, Contour Finding Algorithm (find_contours) da skimage.measure. Fourier Elliptic Descriptor da pyefd.elliptic_fourier_transform.
- Grafici: pyplot da matplotlib, plot_confusion_matrix da sklearn.metrics.
- Utility: os, sys, gzip, time, numpy.

Euristiche

Sono state compiute operazioni di Features Extraction al fine di:

- Ridurre la dimensione dei file contenenti le features.
- Ridurre la dimensione delle features senza perdita significative di informazioni
- Ridurre il tempo impiegato dagli algoritmi di apprendimento

Le tecniche impiegate sono le seguenti, descritte in dettaglio in Optical Handwritten Character Recognition[3]:

- Elliptic Fourier Contour Analysis
- Histogram of Oriented Gradients (Hog) o 'Zoning'
- Letter Profiles
- Projection Histogram

Per ridurre ulteriormente la dimensione delle features, è stata applicata la Principal Component Analysis (PCA), con una dimensione e un numero di cifre significative opportuno per ogni euristica.

Le euristiche sono confrontate in Tabella 1. Ogni euristica è stata combinata con la Fourier Analysis, praticamente concatenando le features, poiché questa raggiunge ottimi risultati con un numero ridotto di features.

La scelta dell'euristica per confrontare in ultima analisi Random Forest e Decision Tree, è ricaduta su una combinazione di features Projection con PCA e Fourier, poiché realizza il miglior risultato con la minor dimensione.

Euristica	Originali	Acc.	PCA	Acc.	PCA&Fourier	Acc.
Nessuna	784	83%	7	45%	20	78%
Profiles	112	82%	12	58%	25	80%
Hog	128	80%	15	46%	28	70%
Projection	56	72%	7	60%	20	81%

Table 1: Euristiche - per ognuna si confrontano le dimensioni delle features e l'accuratezza dell'apprendimento, applicando la PCA e combinandole con le features della Fourier Analysis. L'apprendimento è tramite Random Forest con 10 alberi.

Figure 1: Un campione del dataset EMNIST Letters, si evidenziano alcune lettere relativamente ambigue. L'immagine originale è reperibile su ReseachGate[4]

Figure 2: Un campione di features di 152 i e 148 l, molte di queste ambigue. E' evidente come anche una persona reale difficilmente riuscirebbe a distinguerle accuratamente senza contesto.

Risolvere Errori Sistematici

Sono state prodotte delle labels alternative per evitare alcuni errori sistematici.
Per fare un esempio, si prenda in considerazione come 'i' maiuscola e 'L' minuscola vengano spesso rappresentate con lo stesso tratto grafico 'linea verticale'. Lo stesso vale per una i minuscola senza puntino.

Questa confusione viene riscontrata anche in fase di apprendimento, come evidenziato in Figura 4. Per ovviare a questo problema, si prevede una nuova label 'linea verticale': la condizione di appartenenza a questa classe prevede di essere classificata sia come l quando si tolgono tutte le i dal train, sia come i quando si tolgono tutte le l dal train. Il classificatore utilizzato nel nostro caso è Random Forest con 150 alberi. Nella Figura 2 si può notare come in effetti, le features con label 'linea verticale' siano identificate correttamente.

Decision Tree

E' stato applicato il Decision Tree Classifier senza alcuna riduzione della dimensione del dataset. Si confrontano in Tabella 2 i dati relativi all'applicazione di Decision Tree su diverse tipologie e combinazioni di features.

Random Forest

E' stato applicato il Random Forest Classifier senza alcuna riduzione della dimensione del dataset. Ci si pone il problema di individuare un valore ottimo del numero di alberi. Ipotizziamo che il valore ottimo si trovi tra 10 e 600. Consideriamo un valore ottimo se è il limite superiore oltre il quale l'accuratezza non migliora sensibilmente. Dati i risultati in Tabella 3, scegliamo come valore ottimo 150. Produciamo infine la Tabella 4 analoga a quella di Decision Tree per confrontarle.

Conclusione

Random Forest ottiene risultati di apprendimento migliori di Decision Tree, come mostrato in Figura 3, tuttavia i tempi di esecuzione dell'algoritmo di apprendimento sono più lunghi: questo è dovuto a un maggior numero di alberi da prendere in considerazione. Il problema del tempo di apprendimento viene in parte risolto tramite la Dimensionality Reduction.

Notiamo inoltre come gli errori siano concentrati su alcune lettere piuttosto che altre. Abbiamo visto inoltre come questo problema sia spesso legato alla forma stessa dei caratteri grafici, che necessita del contesto della parola per essere correttamente riconosciuta, proprio come farebbe una persona reale con delle lettere ambigue.

Features	Labels Originali	Labels Corrette	Tempo	Dimensione
Originali	70.60%	72.00%	75 s	784
ProjectionPCA & Fourier	70.89%	72.91%	3 s	20
Fourier	63.28%	65.02%	3 s	13
Projection	59.01%	60.23%	5 s	56
ProjectionPCA	46.71%	48.11%	0.9 s	7

Table 2: Decision Tree - per "ProjectionPCA" si intende l'applicazione dell'euristica Projection con PCA alle features originali, considerando soltanto la parte intera dei coefficienti. Per 'Corrette' si intende le labels con la classe 'linea verticale'. I tempi sono sostanzialmente identici per le due tipologie di labels.

Features	n=10	n=90	n=100	n=150	n=300	n=600
Originali	85.89%	90.16%	90.05%	90.48%	90.52%	90.64%
Fourier	75.64%	79.29%	79.13%	79.10%	79.46%	79.62%
ProjectionPCA	58.71%	63.56%	63.80%	64.00%	63.90%	64.06%
Hog	82.50%	88.45%	88.49%	88.77%	89.12%	89.16%

Table 3: Perfomance di Random Forest al variare delle features e del numero di alberi (n in tabella). Sono state utilizzate le labels con la correzione degli errori sistematici.

Features	Labels Originali	Labels Corrette	Tempo	Dimensione
Original	88.64%	90.48%	108 s	784
ProjectionPCA & Fuorier	85.80%	87.40%	25 s	20
Fourier	77.73%	79.29%	21 s	13
Projection	77.86%	79.36%	31 s	56
ProjectionPCA	61.81%	63.85%	16 s	7

Table 4: Random Forest - il tempo di esecuzione dell'algoritmo di apprendimento riportato è il peggiore tra quelli impiegati per le due tipologie di labels. La differenza è sempre comunque trascurabile.

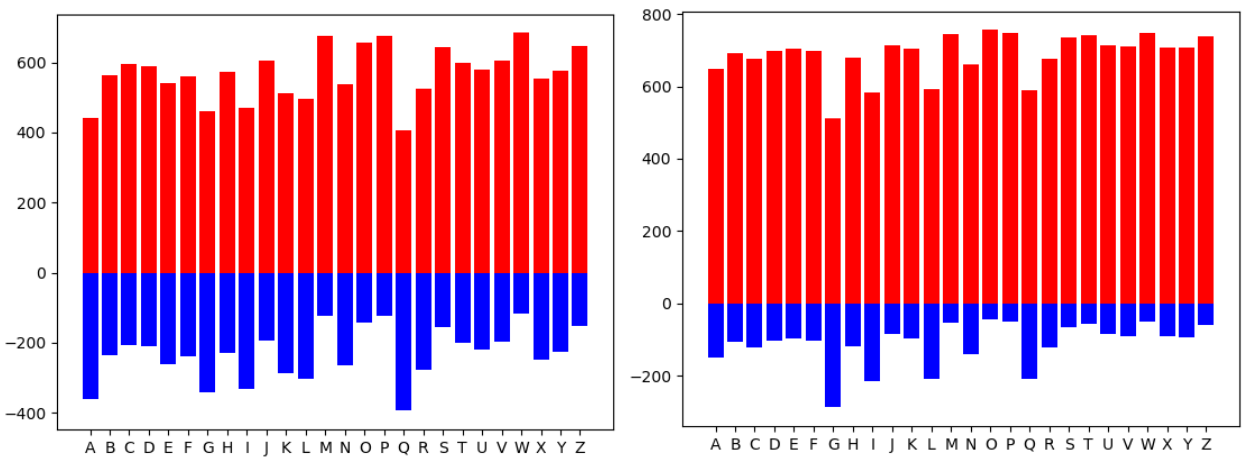

Figure 3: Confronto tra Decision Tree, a sinistra, e Random Forest, a destra. Le due figure evidenziano come il primo abbia una performance peggiore del secondo.

Bibliografia

[1] NIST: The EMNIST Dataset,
`https://www.nist.gov/itl/products-and-services/emnist-dataset`

[2] PyPI: PyEFD,
`https://pypi.org/project/pyefd/`

[3] Pulak Purkait. *Optical Handwritten Character/Numeral Recognition*,
`https://www.isical.ac.in/~lrg/sites/default/files/Pulak/Off-Line%20Handwritten%20OCR.pdf`

[4] Alejandro Baldominos. ResearchGate: Samples of all letters and digits in the EMNIST dataset,
`https://www.researchgate.net/figure/Samples-of-all-letters-and-digits-in-the-EMNIST-dataset_fig2_334957576`

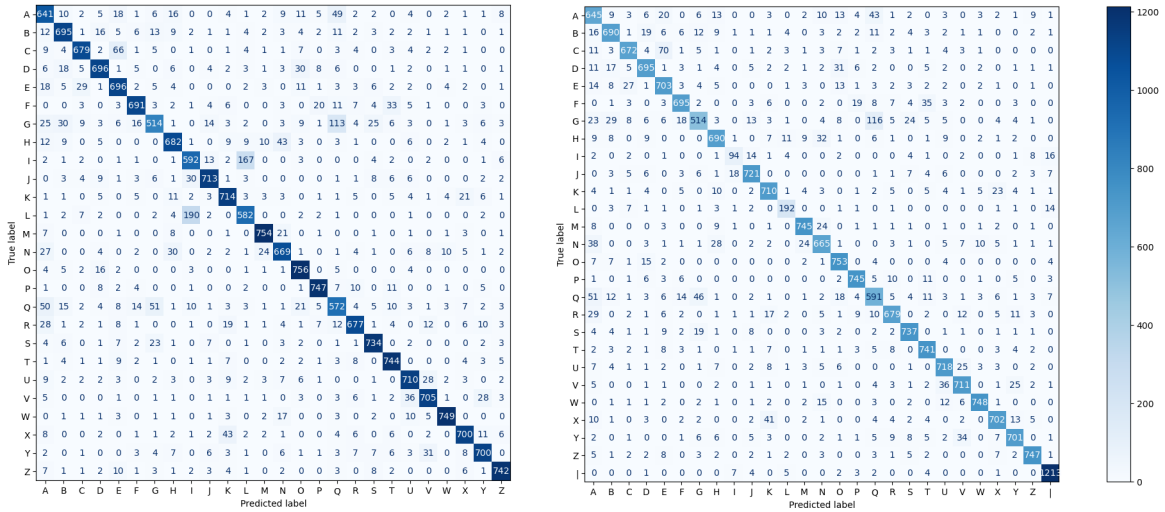

Figure 4: Confusion Matrix - confronto tra le due tipologie di labels. E' evidente come la tecnica impiegata per risolvere le confusioni tra i e l abbia avuto successo, sebbene si siano ridotti inevitabilmente i campioni di questi caratteri. Lo stesso processo, si potrebbe applicare per i caratteri 'g' e 'q', 'u' e 'v'.