

Classificazione di dati LiDAR da aeromobile utilizzando algoritmi di machine learning: un confronto tra Random Forest e TensorFlow

Francesco Pirotti ^(a), Filippo Tonion ^(b)

^(a) CIRGEO – Università degli Studi di Padova, Viale dell'Università Legnaro, +39 0498272710, francesco.pirotti@unipd.it,

^(b) CIRGEO – Università degli Studi di Padova, T.E.R.R.A. S.r.l., Via Galleria Progresso 5, +390421332784, f.tonion@terrasrl.com

1. Introduzione

Il Machine Learning (ML) è la branca dell'Intelligenza Artificiale (IA) che riguarda l'individuazione automatica di modelli significativi nei dati (Shalev-Shwartz and Ben-David, 2014). L'approccio del ML all'analisi dei dati è nato nella seconda metà del 20° secolo, quando le teorie matematiche (come i minimi quadrati, il teorema di Bayes e le catene di Markov) hanno affiancato lo sviluppo dell'informatica moderna. A partire dall'ultima decade le tecniche di ML sono state adottate in ambiti di applicazione caratterizzati da un notevole ammontare di dati, come ad esempio l'astronomia, la biologia, la climatologia, la finanza e l'economia (Is et al., 2015). Grazie al gran numero di algoritmi, oggi il ML trova applicazione in numerosi ambiti connessi alla vita di tutti i giorni (internet, produzioni industriali, medicina.. ecc).

In generale i moderni algoritmi di ML consentono di usare differenti metodi di apprendimento dai dati. In particolare è possibile distinguere le seguenti tipologie di approcci:

- ∞ Non supervisionato. Quando schemi e strutture sconosciute sono individuate all'interno del dataset senza una conoscenza a priori.
- ∞ Supervisionato. Quando viene definito un sottoinsieme del dataset avente funzione di "training" e la restante parte con funzione di validazione.
- ∞ Semi supervisionato. Quando non è necessario definire il "training" e il modello viene perfezionato sulla base dell'esperienza accumulata.

L'utilizzo di un approccio supervisionato è comune per problemi di classificazione. Due algoritmi di classificazione affermati e di comune utilizzo sono Random Forest (RF) e Tensor Flow (TF); questi algoritmi eseguono analisi statistiche non parametriche e ottengono risultati eccellenti con distribuzioni multimodali.

Il modello RF è basato su un insieme di alberi decisionali (la foresta) che crescono attraverso il training verso la miglior combinazione. Infatti l'insieme consiste in un numero definito di alberi, allenati individualmente come classificatori, che sono combinati per classificare nuovi dati (Kulkarni, 2013).

Il modello RF richiede la definizione di due parametri che sono il numero di alberi da generare (Ntree) e il numero di variabili (Mtry) da selezionare e testare come miglior combinazione in fase di addestramento degli alberi (Belgiu and Drăgu, 2016).

Il numero di alberi può essere diverso e dipende dall'efficienza computazionale e dal rischio di overfitting; per esempio il numero di alberi standard nel

pacchetto R "Random Forest" è 500. Una volta che tutti i parametri sono definiti il modello costruisce tutti e 500 i classificatori. Relativamente alla selezione delle variabili sono di solito utilizzati i seguenti approcci:

- Random Forest utilizzando Selezione di input casuale (Random Input). Questo tipo di approccio è il più comune e prevede che m variabili siano selezionate a caso tra quelle disponibili.
- Random Forest utilizzando una combinazione lineare degli input (RC). Prima della selezione delle migliori variabili da selezionare, è creata una combinazione lineare delle variabili.

Comunque in fase di costruzione della foresta il numero di variabili da selezionare è mantenuto costante. Prima dell'utilizzo del modello il dataset è diviso in due parti: la parte di addestramento (training) e quella di validazione. Durante la fase di addestramento il dataset di addestramento è processato in ciascuno degli alberi decisionali della foresta.

In fase di classificazione il dataset è processato da ogni albero. La classificazione finale è assunta secondo un criterio di voto maggioritario.

L'algoritmo RF fornisce inoltre l'importanza delle variabili (VI) e la vicinanza, che sono importanti per capire il comportamento delle variabili nel modello (Breiman, 2001; Cutler et al., 2007).

Il TF è una libreria creata dalla sezione intelligenza artificiale di Google e rilasciata nel 2015 con una licenza open source. TF usa grafici per rappresentare sia il calcolo dell'algoritmo che lo stato in cui l'algoritmo opera (Abadi et al., 2016). TF è basato sulle Reti Neurali Convolutionali (CNN); questo tipo di approccio è simile a quello delle Reti Neurali (NN), ma usa funzioni di convoluzione e riduce il numero di parametri necessari al modello.

La riduzione del numero di parametri riduce i costi computazionali e migliora la capacità di generalizzare (Hemmes, 2018).

Una comune applicazione del TF è divisa in due fasi, ovvero:

- Definizione del programma. In questa fase una CNN da addestrare è definita e rappresentata graficamente.
- Ottimizzazione. Nella seconda fase la rete neurale definita è eseguita e ottimizzata sulla base dei dati disponibili.

In generale queste due tecniche di classificazione possono trovare interessanti applicazioni nella classificazione delle nuvole di punti. Infatti le tecniche di Laser Scanning consentono di acquisire informazioni spaziali sotto forma di nuvole di punti 3D. Prima di eseguire la classificazione è possibile fare una segmentazione dei punti 3D. Questo processo raggruppa i punti in diversi cluster, usando diversi approcci, come ad esempio segmentazione margini, segmentazione per regioni, segmentazione per fitting del modello, tecniche ibride di segmentazione e segmentazione con ML (Grilli et al., 2017).

Il risultato della fase di segmentazione (cluster) può essere usato come variabile per la fase di classificazione della nuvola di punti. Il processo di classificazione riguarda inizialmente la fase di estrazione delle variabili. Inizialmente questo processo riguarda la determinazione della prossimità per ciascuno dei punti, quindi l'estrazione delle geometrie tridimensionali è basata su ciascuno dei punti più prossimi, secondo la distanza stabilita. L'area di analisi della prossimità può essere di forme diverse (sferica, cilindrica...ecc) e per dimensione (fissa o variabile) (Weinmann et al., 2017). Sulla base di

queste analisi sono estratte le geometrie, che vengono alla fine classificate utilizzando un algoritmo di classificazione (RF, TF, ANN..). Una moltitudine di applicazioni per approcci ML per dati spaziali è mostrata in numerose pubblicazioni recenti che si focalizzano sull'analisi delle immagini (Pirotti et al., 2016) e anche sulle predizioni spaziali basate su immagini raster (Piragnolo et al., 2019).

In questo lavoro viene comparata l'accuratezza del RF e del TF nella classificazione di nuvole di punti 3D.

2. MATERIALI E METODI

2.1. DATI

I dati utilizzati per l'investigazione dei modelli sono stati forniti dal ISPRS e riguardano il benchmark sull'individuazione di oggetti e sulla ricostruzione 3D (Lukac et al., 2014). In particolare è stata utilizzato il dataset della scansione laser della città di Vaihingen (Germania). I punti nel dataset erano tutti classificati in 8 class differenti (Alberi, tetti, vegetazione bassa, linee elettriche..).

2.2. METODI

L'obiettivo dello studio è testare gli algoritmi RF e TF in un set di osservazioni (punti) usando un vettore con le variabili come predittore. I predittori sono derivati non solo dai dati presenti come attributi dei punti (ovvero l'intensità della riflessione), ma anche dalle caratteristiche geometriche estratte analizzando i punti nelle vicinanze (analisi della prossimità). Il vettore con le caratteristiche è successivamente utilizzato come input per RF e TF.

2.2.1. Estrazione delle Variabili

Il vettore delle variabili con i predittori sono calcolati considerando il contesto spaziale. Il contesto deve essere definito da un numero di punti più vicini (*nearest neighbours nn*), che sia largo abbastanza da rappresentare una classe, ma anche sufficientemente piccolo da evitare di includere punti che appartengono ad altre classi.

Quindi la cardinalità di *nn* non è fissata ma è determinata con un metodo che massimizza le consistenze geometriche. Ciò è fatto selezionando il numero *nn* che risulta avere il valore dell'indice di entropia di Shannon più basso, calcolato utilizzando l'autovettore normalizzato della matrice di covarianza in 3D, rappresentato dalla seguente espressione:

$$\min\{\sum_i^3 \lambda_i(nn) \cdot \ln[\lambda_i(nn)]\} \quad [1]$$

dove $nn = 15$ e $nn \max = 100$, λ_i sono gli autovettori normalizzati della matrice tensoriale 3D e $i = \{1, 2, 3\}$.

Complessivamente le variabili di predizione estratte sono state 20 (linearità, planarità, verticalità, anisotropia.. ecc).

2.2.2. Definizione dei parametri

Per la definizione dei parametri Numero di Alberi (NT) e Numero di Variabili (Nf) dell'algoritmo RF è stato utilizzato un approccio iterativo mediante "k-fold validation" utilizzando a ciascuna iterazione il 20% dei dati.

Al termine del processo sono stati utilizzati i seguenti parametri: $N_f = 8$ e $N_t = 200$.

2.2.3. Matrice dell'accuratezza

La valutazione dell'accuratezza è stata condotta fornendo le metriche sull'accuratezza per ciascuna classe, ovvero: Precisione, Recall, Indice di Jackard. La matrice di accuratezza utilizzata è:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \dots & \dots & a_{2n} \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad TP_i = a_{ii}$$

$$FP_i = \sum_j a_{ji} - TP_i \quad FN_i = \sum_j a_{ij} - TP_i$$

$$Pr = \frac{TP}{TP + FP} \quad Re = \frac{TP}{TP + FN}$$

$$Ji = \frac{TP}{FP + FN + TP} \quad F1 = 2 \cdot \frac{Pr \cdot Re}{Pr + Re}$$

[2]

dove A è la matrice degli errori, TP sono i veri positivi, FP sono i falsi positivi, FN sono i falsi negativi, Pr è la precisione, Re è la Recall, Ji è l'indice di Jackard ed $F1$ è il punteggio della classe F1.

I due algoritmi di Machine Learning sono stati addestrati con un campionamento del dataset stratificato per classe. La dimensione del campionamento è variata dal 5% al 50%. Quindi è stata analizzata la modificazione dell'accuratezza alla modificazione della dimensione del campione di training.

3. RISULTATI E DISCUSSIONE

Nel presente capitolo viene fornita una descrizione dei risultati ottenuti. In particolare vengono di seguito illustrati i risultati della classificazione dei due algoritmi utilizzati. Infine viene fornita una comparazione tra i due metodi utilizzati.

3.1. Risultati Random Forest

La tabella successiva mostra gli errori di omissione e di commissione della classificazione fatta con RF, con le metriche di accuratezza.

	Powerline	Low vegetation	Impervious surfaces	Car	Fence/Hedge	Roof	Facade	Shrub	Tree
0	419	7	1	1	2	23	9	11	127
1	686728	4720	25	77	2542	84	711	3797	
2	0	3544	97229	45	13	780	11	53	311
3	0	580	123	2316	14	195	4	93	383
4	0	1140	64	9	4588	205	34	243	1139
5	3	4849	2216	10	20	98685	95	272	2898
6	7	608	82	6	25	267	8383	115	1731
7	2	2929	256	8	43	731	99	15302	5448
8	12	2179	139	9	53	1255	288	856	49435
F1	0.799	0.862	0.940	0.755	0.749	0.924	0.829	0.721	0.827
Re	0.698	0.879	0.953	0.625	0.618	0.905	0.747	0.617	0.912
Pr	0.933	0.846	0.927	0.953	0.949	0.943	0.931	0.867	0.757
Ji	0.285	0.301	0.320	0.274	0.272	0.316	0.293	0.265	0.293

Tabella 1 – Risultati classificazione RF con metriche accuratezza

Nella figura successiva viene riportata la mappa dei punti classificati erroneamente con RF.

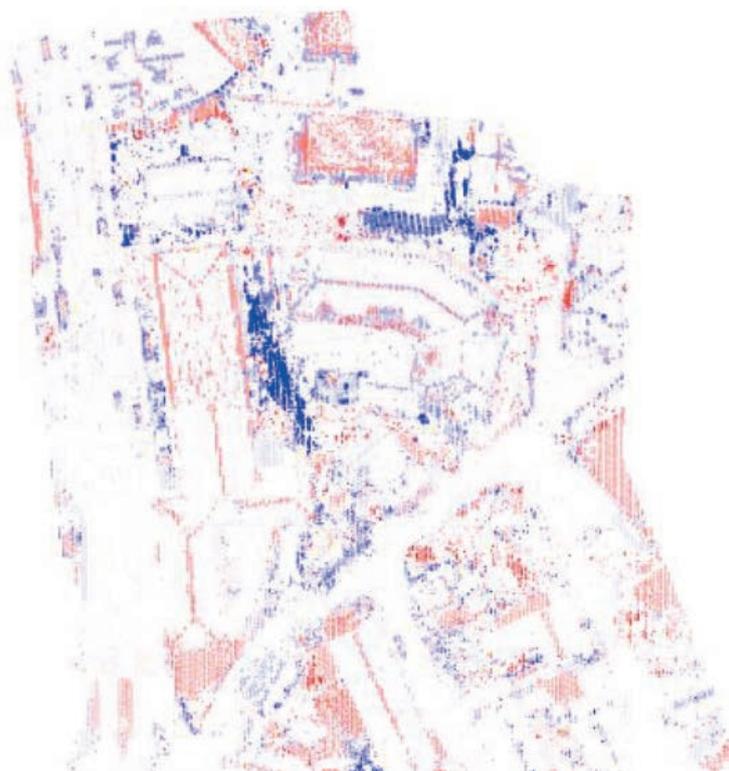


Figura 1 – Punti classificati erroneamente RF

3.2. Risultati Tensorflow

La tabella successiva mostra gli errori di omissione e di commissione della classificazione fatta con TF, con le metriche di accuratezza.

	Powerline	Low vegetation	Impervious surfaces	Car	Fence/Hedge	Roof	Facade	Shrub	Tree
0	265	12	1	1	0	29	42	0	250
1	2	63634	13124	26	74	8806	247	342	12435
2	1	7294	87276	22	11	5889	18	9	1466
3	3	1345	421	219	7	742	33	56	882
4	3	2784	232	5	119	519	246	115	3399
5	19	14350	6927	19	11	77313	238	122	10049
6	18	1051	230	9	28	812	4612	60	4404
7	15	5866	602	20	43	2174	319	391	15388
8	21	4065	457	30	30	3631	1033	258	44701
F1	0.560	0.639	0.827	0.108	0.031	0.740	0.512	0.030	0.607
Re	0.442	0.645	0.856	0.059	0.016	0.709	0.411	0.016	0.824
Pr	0.764	0.634	0.799	0.624	0.368	0.774	0.679	0.289	0.481
Ji	0.219	0.242	0.292	0.051	0.015	0.270	0.204	0.015	0.233

Tabella 2 – Risultati classificazione TF con metriche accuratezza

Nella figura successiva viene riportata la mappa dei punti classificati erroneamente con TF.

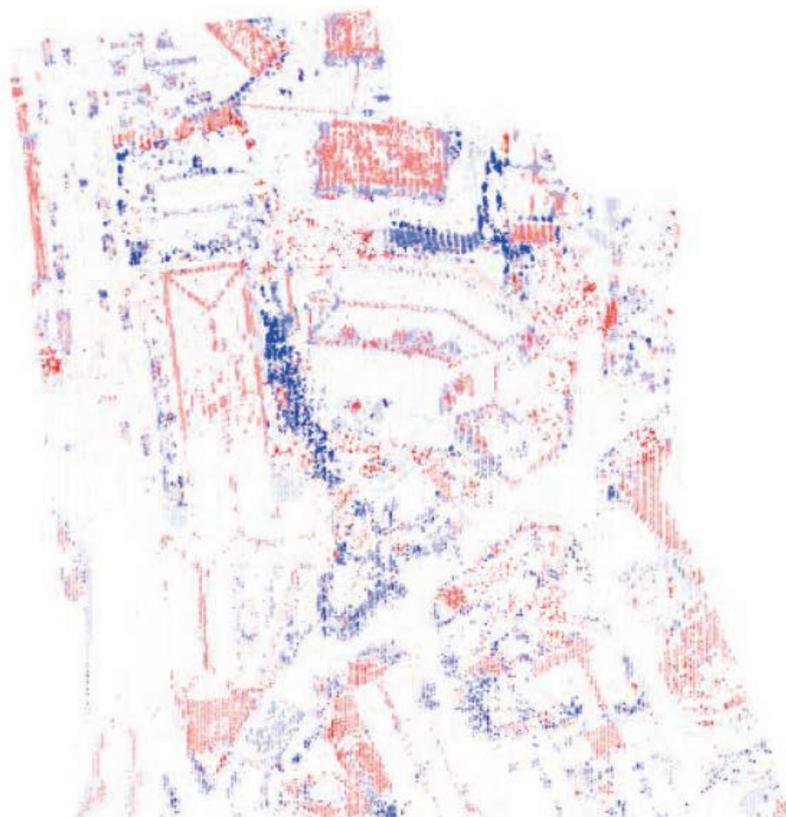
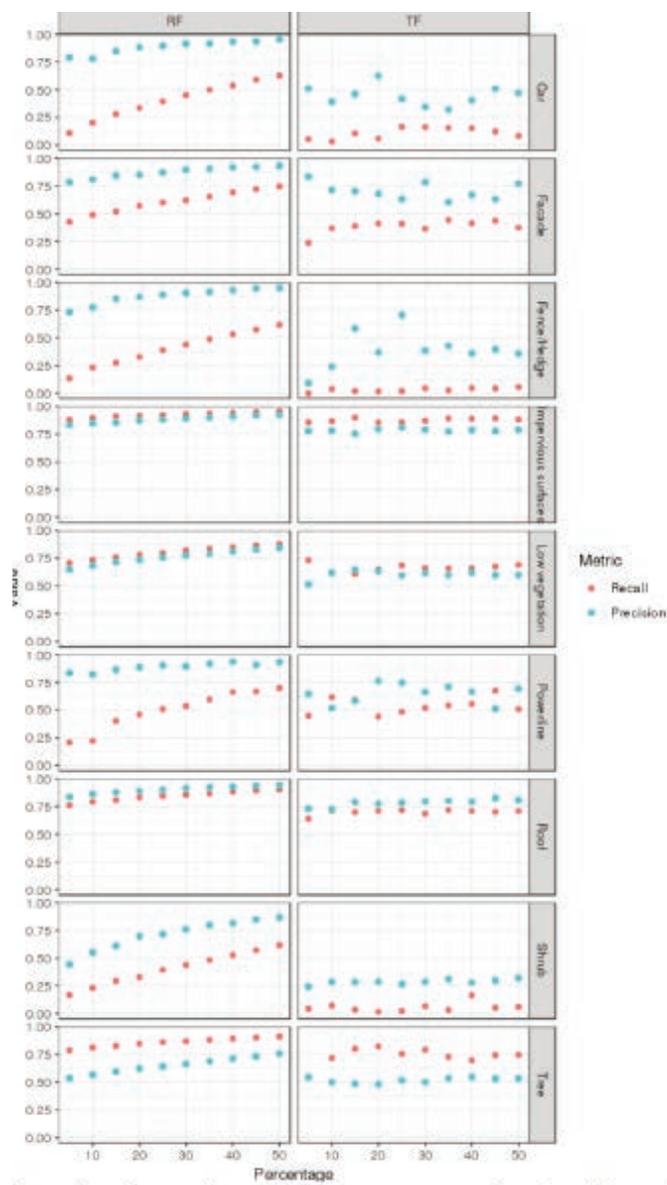


Figura 2 – Punti classificati erroneamente TF

3.3. Comparazione dei risultati di RF e TF

Nella figura successiva viene riportato l'andamento delle metriche di accuratezza (recall e precisione) per ogni classe e metodo in funzione della percentuale di dati usati per l'addestramento.



L'esame della figura precedente permette di riscontrare che il RF consente di incrementare rapidamente l'accuratezza per alcune classi (per esempio cespugli). Punti appartenenti alla classe cespugli vengono frequentemente classificati erroneamente come classe alberi. In merito a tale tipologia di distinzione il modello RF si è dimostrato più accurato.

4. Conclusioni

In questo studio è stata presentata la comparazione tra due modelli di Machine Learning (ML) per la classificazione semantica di nuvole di punti 3D. I modelli oggetto del confronto sono Random Forest (RF) e Tensoflow (TF). La comparazione delle accuratezze ottenute permette di riscontrare che RF permette di ottenere accuratezze maggiore. In particolare punteggi di F1 maggiori con RF sono dovuti a maggiore flessibilità di RF; TF necessita invece

di tarare e definire i parametri degli strati nascosti (operazione non sempre immediata).

La classificazione delle nuvole di punti con metodi avanzati sta fornendo risultati promettenti, che in futuro permetterà di utilizzare le nuvole di punti per scopi applicati.

5. Bibliografia

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., Brain, G., Osd, I., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow : A System for Large-Scale Machine Learning This paper is included in the Proceedings of the TensorFlow : A system for large-scale machine learning.
- Belgiu, M., Drăgu, L., 2016. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114, 24–31. doi:10.1016/j.isprsjprs.2016.01.011
- Breiman, L.E.O., 2001. Random Forests. *Machine Learning*, 5–32.
- Cutler, D.R., Edwards Jr., T.C., Beard, K.H., Cutler, A., Hess, K.T., Gibson, J., Lawler, J.J., 2007. RANDOM FORESTS FOR CLASSIFICATION IN ECOLOGY. *Ecology*, 88, 2783–2792. doi:10.1890/07-0539.1
- Grilli, E., Menna, F., Remondino, F., Scanning, L., Scanner, L., 2017. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII, 1–3. doi:10.5194/isprs-archives-XLII-2-W3-339-2017
- Hemmes, T., 2018. Classification of large scale outdoors point clouds using Covolutional Neural Networks. *Master in Science in Geomatics for Built Environment*.
- Is, G.K., Taha, KamalAl-Jarrah, O.Y., Yoo, P.D., Muhaidat, S., Karagiannid, 2015. Efficient Machine Learning for Big Data: A Review. *Big Data Research*, 2, 87–93. doi:10.1016/j.bdr.2015.04.001
- Kulkarni, V.Y., 2013. Random Forest Classifiers : A Survey and Future Research Directions 36, 1144–1153.
- Lukac, N., Meidow, J., Pohl, M., Solbrig, P., Wernerus, P., Habib, A., Lari, Z., Xiong, B., Elberink, S.O., 2014. Urban object detection and 3D building reconstruction '. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 143–144. doi:10.1016/j.isprsjprs.2014.04.009
- Piragnolo, M., Grigolato, S., Pirotti, F., 2019. Planning harvesting operations in forest environment: Remote sensing for decision support IV, 12–14.
- Pirotti, F., Sunar, F., Piragnolo, M., 2016. Benchmark of machine learning methods for classification of a sentinel 2 image XLI, 335–340. doi:10.5194/isprsarchives-XLI-B7-335-2016
- Shalev-Shwartz, S., Ben-David, S., 2014. Understanding machine learning: From theory to algorithms, *Understanding Machine Learning: From Theory to Algorithms*. doi:10.1017/CBO9781107298019
- Weinmann, M., Weinmann, M., Mallet, C., Brédif, M., 2017. A classification-segmentation framework for the detection of individual trees in dense MMS point cloud data acquired in urban areas. *Remote Sensing*,. doi:10.3390/rs903277