Confronto algoritmi di apprendimento: Naive Bayes e Decision Tree

Ubaldo Puocci

September 7, 2017

1 Introduzione

In questa relazione si evidenziano i risultati ottenuti confrontando le implementazioni in linguaggio Python di due algoritmi di apprendimento: Naive Bayes e Decision Tree.

2 Raccolta dei dati

Il lavoro sperimentale è stato svolto utilizzando l'implementazione multinomiale di Naive Bayes e quella standard di Decision Tree incluse nella libreria di *scikit-learn*[1]. Sono stati usati cinque dataset per il confronto: 20 Newsgroups [2], Digits, MNIST [3], DMOZ web directory topics, e Yahoo! web directory topics.

I dataset 20Newsgroups e Digits sono inclusi nella libreria *scikit-learn* ed è stato quindi utilizzato il fetching standard per ottenerne i dati.

I restanti dataset sono stati scelti dal repository MLData. In particolare, i dataset DMOZ e Yahoo! web directory topics vengono caricati dai rispettivi file `libsvm`.

3 Elaborazione dei risultati

I risultati ottenuti da ciascun dataset sono graficati utilizzando la libreria Matplotlib [4]. I risultati sono riportati su dei grafici in cui le ascisse corrispondono alla percentuale di train utilizzata, e le ordinate corrispondono alla media dell'errore sul test set. Il train size varia in scala logaritmica fra il 10% ed il 50% dei dati, mentre il test set è fisso al 50% degli stessi. Il test viene ripetuto dieci volte per ogni percentuale di train per evitare valori anomali o "casi speciali". Nel grafico viene inoltre riportata la deviazione standard al variare della percentuale di train test.

20 Newsgroups

Il dataset 20 Newsgroups contiene circa 18000 post di un newsgroup divisi fra 20 topic diversi. I dati utilizzati sono stati elaborati rimuovendo *headers, quotes,* e *footers* poiché contengono parole chiave che rendono molto semplice la classificazione di un documento di questo dataset. La rimozione di queste informazioni peggiora notevolmente il punteggio ottenuto dai due classificatori, ma si avvicina di più a un valore reale. I dati di questo dataset sono testuali, viene quindi utilizzata una tecnica per trasformare questi dati in modo da renderli fruibili per l'analisi statistica. Quest'operazione è divisa in due fasi:

1. **Count vectorizer**: l'insieme dei documenti di testo è convertito in una matrice di token che rappresenta il numero di occorrenze per ogni parola.
2. **Tf-idf transformation**: questa trasformazione diminuisce l'impatto che hanno i token che occorrono molto frequentemente e che sono quindi meno informativi rispetto agli altri, attribuendogli un peso.

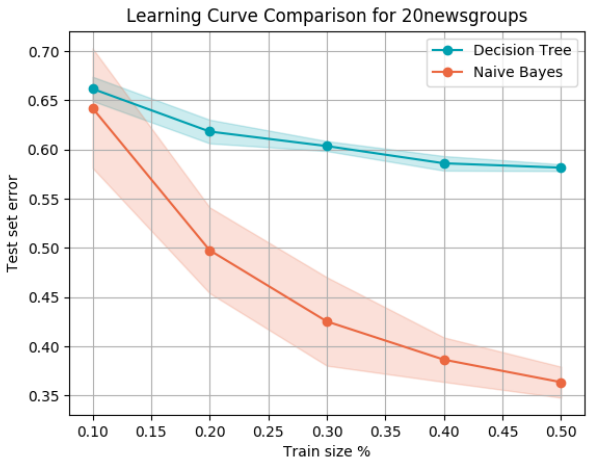


Figure 1

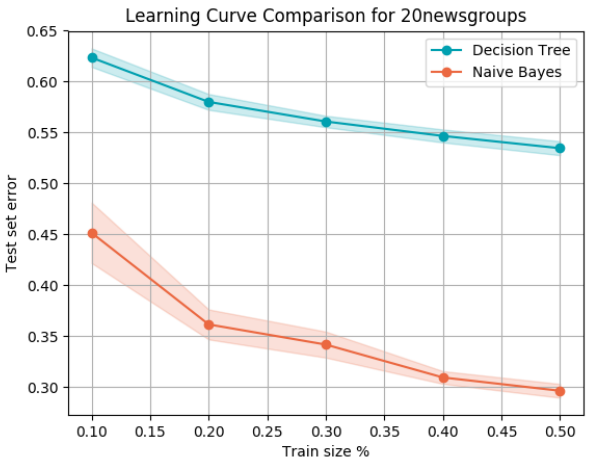


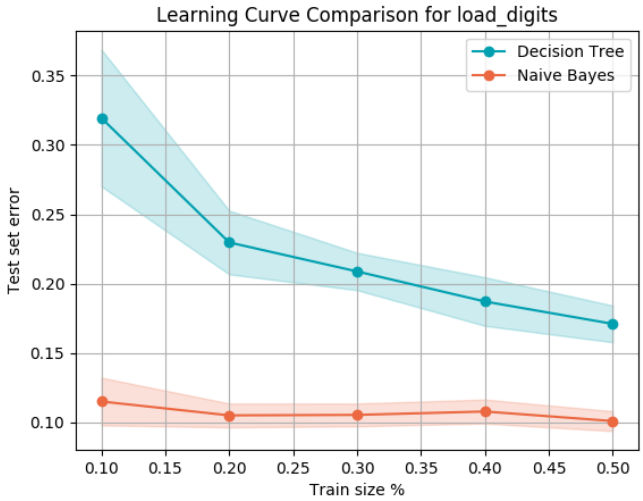
Figure 2

I due grafici sono ottenuti con due opzioni diverse per quanto riguarda il count vectorizing: in figura 1 non è stato usato nessun dizionario di stopwords, mentre in figura 2 è stato utilizzato un dizionario apposito per la lingua inglese. Dai grafici sopra riportati si nota l'elevata deviazione standard di Naive Bayes senza eliminazione di stopwords per la lingua Inglese: esso infatti si comporta sempre meglio di Decision Tree per quanto riguarda

l'accuratezza generale, ma a discapito di un elevata deviazione standard dei risultati del test in figura 1, che comunque tende a diminuire con l'aumentare del numero di esempi del train. Decision Tree, invece, migliora di poco il suo errore nel test set, ma mostra una bassa deviazione standard dei risultati sul test set. Con la rimozione delle stopwords, entrambi i classificatori ottengono un miglioramento, più apprezzabile per Naive Bayes, in cui anche la deviazione standard diminuisce sensibilmente inizialmente e migliora all'aumentare della grandezza del train set.

Digits

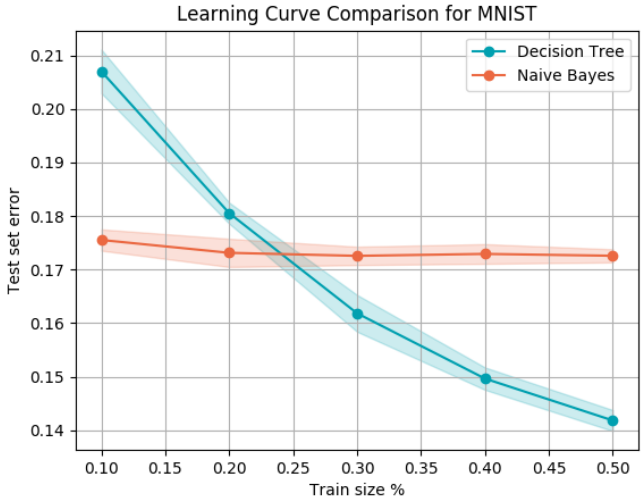
Il dataset Digits contiene 1797 sample di immagini 8×8 dei numeri fra 0 e 9. Ogni classe contiene $\sim$180 sample, ognuno rappresentato da set di interi compresi fra 0 e 16. Questo dataset, essendo numerico, non necessita di preprocessing come 20 Newsgroups poiché è già processato, quindi può essere usato direttamente con i suoi valori originali, senza alterarli.



Dal grafico si evince che entrambi i classificatori riescono a migliorare il proprio risultato di partenza. In particolare, Naive Bayes mostra una curva molto piatta che descrive un buon comportamento dell'algoritmo anche con una bassa quantità di dati. L'algoritmo presenta un lieve peggioramento quando si utilizza il 40% di train set. Per quanto riguarda Decision Tree, la deviazione standard dell'errore è molto grande con una piccola percentuale di test, e diminuisce con l'aumentare di quest'ultima.

MNIST

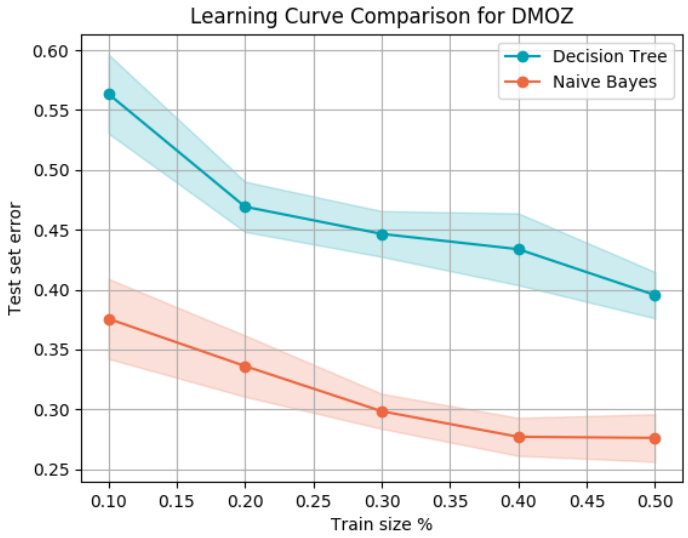
Il dataset MNIST contiene 70000 sample di immagini 28×28 dei numeri fra 0 e 9. Ogni sample è rappresentato da un set di interi compresi fra 0 e 255. Come per Digits, anche il dataset MNIST non necessita di preprocessing.



Il grafico mostra un andamento del classificatore Naive Bayes che è simile a quello mostrato per il dataset Digits. L'implementazione di Decision Tree, invece, riesce a diminuire il proprio errore sul test set ed anche a superare quello di Naive Bayes. In entrambi i casi la deviazione standard dei risultati sul test set è molto bassa, come mostrato dalla rappresentazione della deviazione standard sul grafico.

DMOZ web directory topics

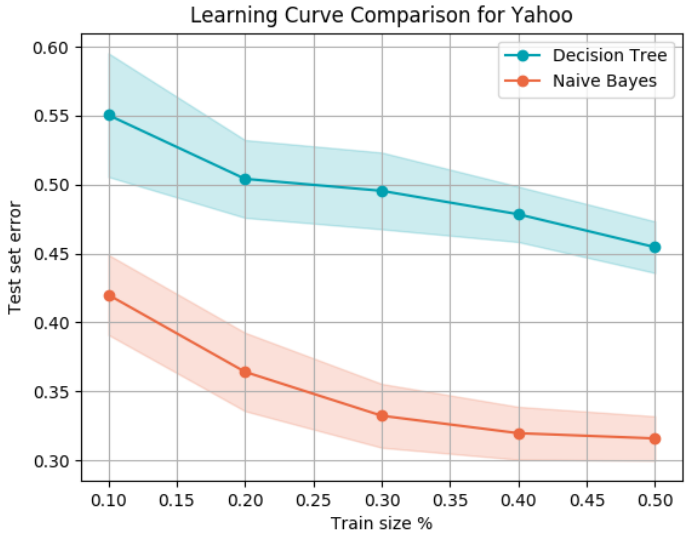
Il dataset DMOZ web direcotry topics contiente 1329 pagine web indicizzate sul sito web di DMOZ. Le pagine sono state riprese dai topic: Arts, Games, Kids and Teens, Shopping, e Society. I dati sono già stati processati e sono distribuiti come "bag of words".



Il grafico mostra che entrambi i classificatori soffrono di un'alta deviazione standard sui risultati e che, seppur lentamente, diminuiscono il proprio errore sul test set. L'implementazione di Naive Bayes presenta un appiattimento finale ed una crescita della deviazione standard. Decision Tree, invece, mantiene la sua deviazione standard quasi invariata, se non per il test ripetuto con il 40% di train size.

Yahoo! web directory topics

Il dataset Yahoo! web directory topics contiene ∼1100 pagine web riprese da diversi topic indicizzati dall'omonima azienda. I topic ripresi sono: Arts, Business and Economy, Education, e Entertainment. Di questi dati è già stato effettuato il parsing e sono rappresentati come "bag of words".



Il grafico mostra un andamento simile a quello ottenuto testando i classificatori in esame con il dataset DMOZ: entrambi gli algoritmi di apprendimento hanno una considerevole deviazione standard sui risultati ed hanno un lento miglioramento, sia per quanto riguarda la deviazione standard sia per quanto riguarda l'errore sul test set, fino ad un appiattimento finale per Naive Bayes. Decision Tree migliora gradualmente la sua deviazione standard, arrivando a dei valori simili a quelli di Naive Bayes.

4 Conclusioni

I dati sperimentali ci mostrano come, sul dataset 20 Newsgroups, l'algoritmo di apprendimento Decision Tree si comporti in modo molto peggiore rispetto a Naive Bayes: è possibile che Decision Tree non riesca a interpretare inizialmente i dati e quindi capire quali feature definiscono in modo concreto un documento. L'albero, crescendo, continua ad espandersi e quando l'algoritmo è riuscito a definire i dati che rappresentano in modo corretto un documento, ci sono pochi sample su cui può usare questa conoscenza acquisita. Questa differenza di prestazione può anche essere frutto del preprocessing effettuato sul dataset e, quindi, dovuto alla trasformazione dei dati. Il dataset in questione infatti, era inizialmente testuale. Con il preprocessing abbiamo alterato in modo concreto la rappresentazione dei dati nel dataset: essi non sono più un insieme discreto, ma solo una rappresentazione pesata di ogni parola. Il classificatore multinomiale di Naive Bayes, nonostante la sua implementazione, riesce comunque ad ottenere dei risultati soddisfacenti, in accordo con i risultati ottenuti in McCallum, A. Nigam, K., (1998) [5].

I dataset Digits e MNIST sono in qualche modo simili: entrambi rappresentano la stessa classe di dati, seppur con qualche piccola differenza. Nonostante ciò, i grafici ottenuti per i due classificatori sono molto diversi: per il dataset Digits, l'algoritmo più performante è quello Naive Bayes, mentre per MNIST è Decision Tree. Questa inversione di tendenza può essere dovuta al numero di dati contenuti in questi due dataset. Possiamo quindi concludere che, per valori puramente numerici, Naive Bayes non ha bisogno di un numero grande di esempi per essere performante, ma al crescere di questi ultimi, Decision Tree può migliorare molto.

Gli ultimi due dataset presi in considerazione sono anch'essi simili per struttura e tipo di dati rappresentati: sia DMOZ che Yahoo! web directory topics rappresentano pagine web che hanno subito il medesimo preprocessing. I dati sperimentali ottenuti sono infatti molto simili: entrambi gli algoritmi migliorano il proprio punteggio e presentano un'alta deviazione standard che tende a diminuire. La dimensione di entrambi i dataset è molto limitata e ciò potrebbe spiegare i risultati ottenuti.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[2] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–339, 1995.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[5] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *In AAAI-98 workshop on learning for text categorization*, vol. 752, no. 41–48, 1998.