

Segmentazione dell'ippocampo con metodo di classificazione *RANDOM FORESTS*

*Laura Monno**
Roberto Bellotti
Gianfranco Gargano
Sonia Tangaro

MAGIC-5

Workshop on Medical Imaging applications and tools
Genova 2-4 Dicembre 2009



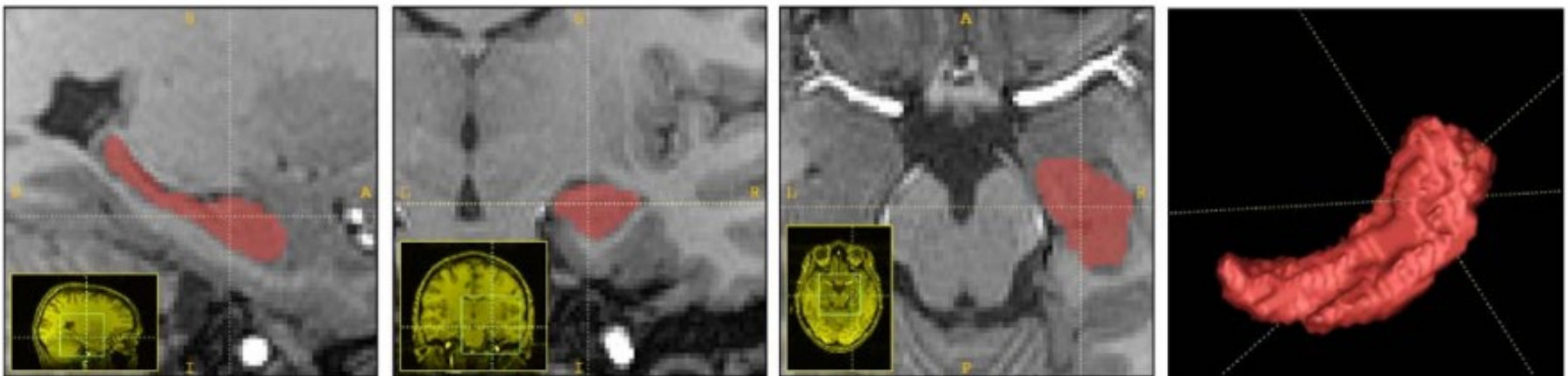
- La segmentazione dell'ippocampo
- Quali *features*?
- Analisi con Haar-like *features*
- Set di *features* da classificare
- Il *Meta-learning*
- Classificazione basata su **alberi decisionali**
- *RANDOM FORESTS*
- Primi risultati
- Conclusioni e prospettive future

La segmentazione dell'ippocampo

È il primo fondamentale passo:

A) per uno studio quantitativo dell'ippocampo stesso;

B) per lo sviluppo di una catena algoritmica finalizzata alla costruzione di indici quantitativi in grado di discriminare tra sani, malati e soggetti a rischio (*MCI*).

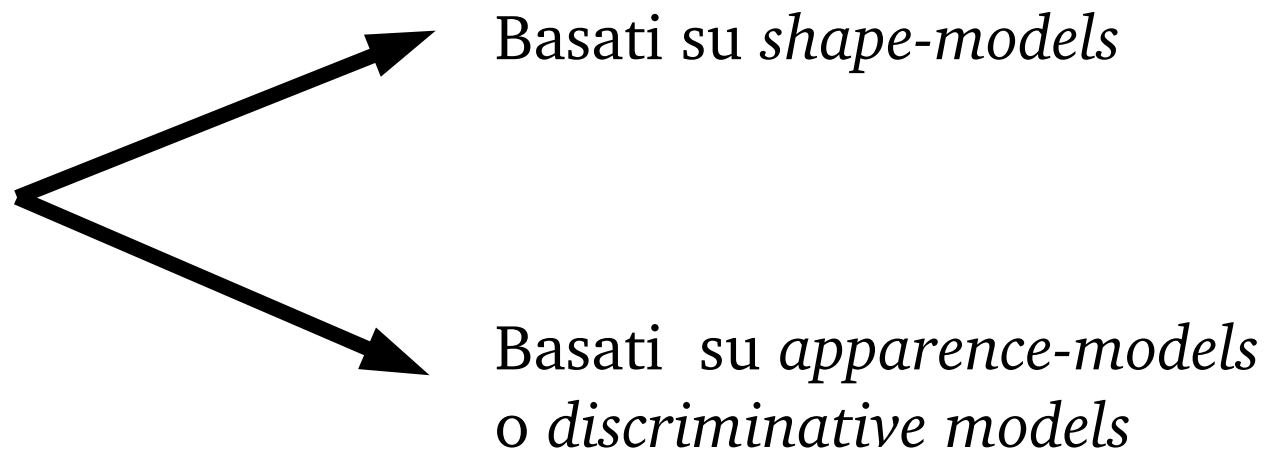


Hippocampus-specific fMRI group activation analysis using the continuous medial representation
Yushkevich et al. NeuroImage 35 (2007)

Cosa esiste già...?

Alcuni sistemi automatici per l'estrazione automatica dell'ippocampo esistono già [1]:

{ Barnes et al. 2004, Crum et al. 2001, Fischl et al. 2002, Hogan et al. 2000, Powell et al. 2008, Wang et al. 2007, Yunshkevich et al. 2006, ...}



Sono tuttavia non diffusi a causa della 'lentezza' computazionale e della scarsa capacità di generalizzazione.

Estrazione delle *features*

Un possibile approccio per la segmentazione dell'ippocampo consiste in una classificazione *voxel per voxel*, attraverso lo studio delle caratteristiche di ogni *voxel* ed in riferimento ai *voxel* vicini:



Contents lists available at [ScienceDirect](#)

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg



Validation of a fully automated 3D hippocampal segmentation method using subjects with Alzheimer's disease mild cognitive impairment, and elderly controls

Jonathan H. Morra^a, Zhuowen Tu^a, Liana G. Apostolova^{a,b}, Amity E. Green^{a,b}, Christina Avedissian^a, Sarah K. Madsen^a, Neelroop Parikshak^a, Xue Hua^a, Arthur W. Toga^a, Clifford R. Jack Jr.^c, Michael W. Weiner^{d,e}, Paul M. Thompson^{a,*}

The Alzheimer's Disease Neuroimaging Initiative¹

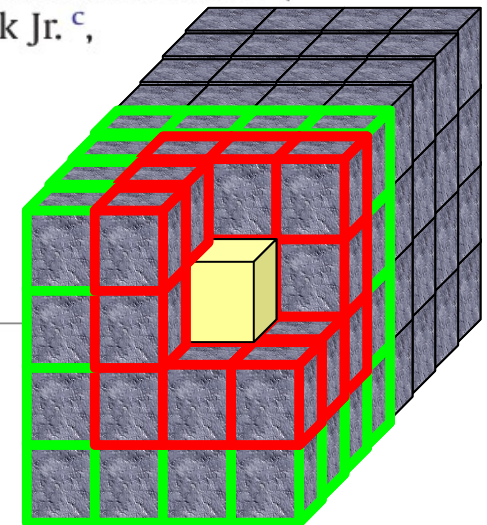
^a Laboratory of Neuro Imaging, Department of Neurology, UCLA School of Medicine, Los Angeles, CA, USA

^b Department of Neurology, UCLA School of Medicine, Los Angeles, CA, USA

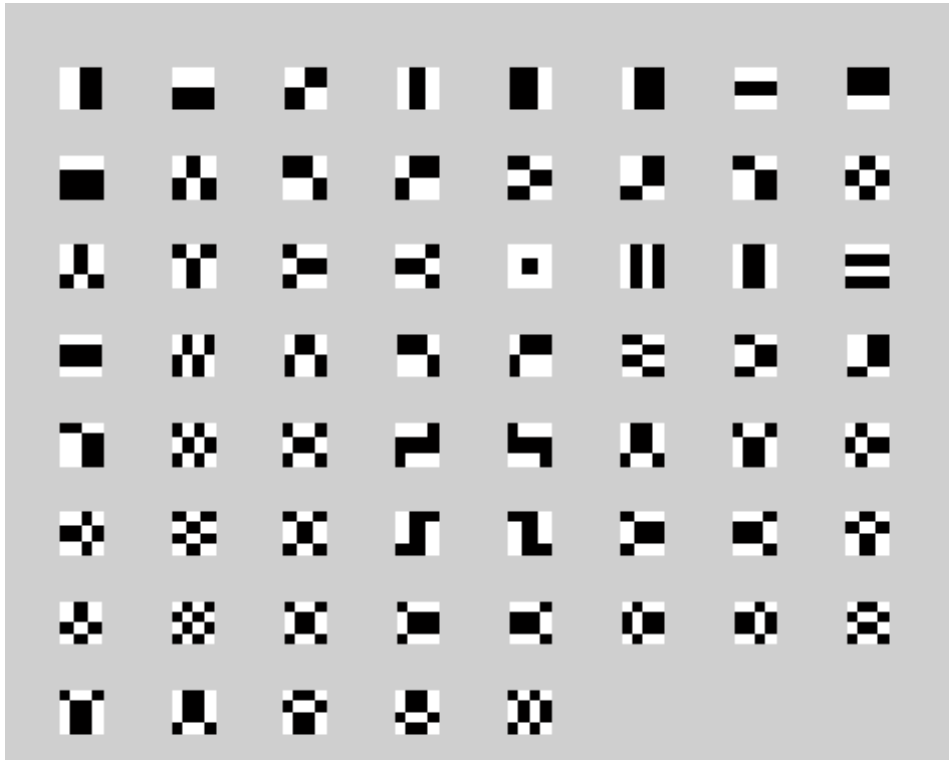
^c Mayo Clinic College of Medicine, Rochester, MN, USA

^d Department of Radiology, UC San Francisco, San Francisco, CA, USA

^e Department of Medicine and Psychiatry, UC San Francisco, San Francisco, CA, USA



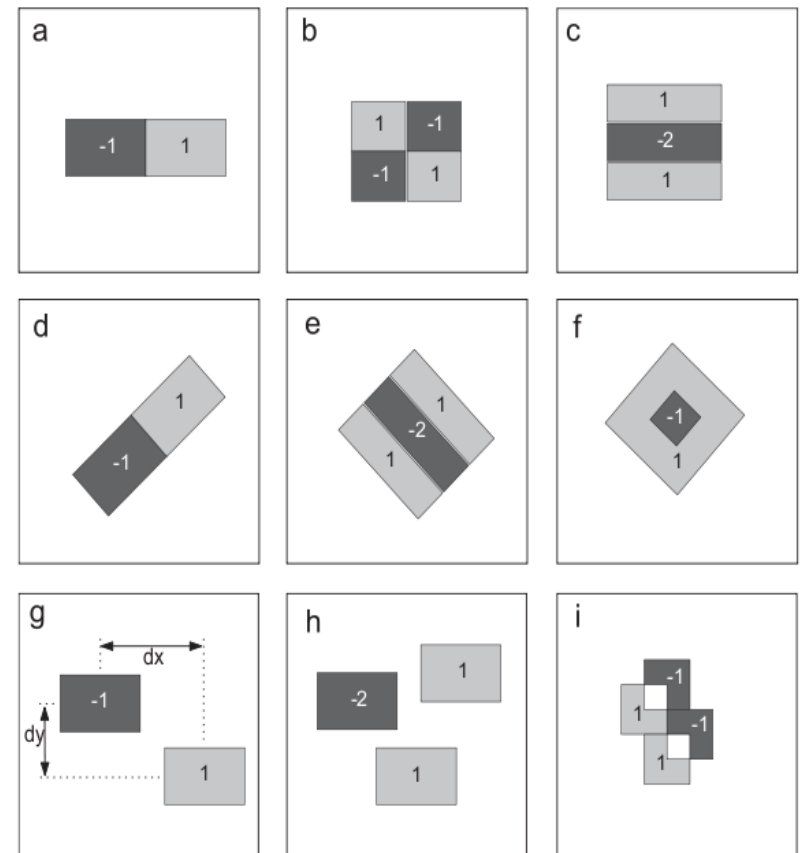
Analisi con Haar-like *features*



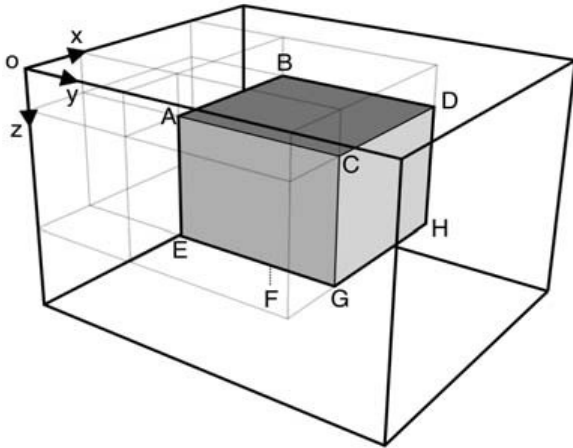
$$f = \sum_{i=1}^k w^{(i)} \cdot \mu^{(i)}$$

$$\sum_{i=1}^k w^{(i)} = 0$$

Haar-like features with optimally weighted rectangles for rapid object detection
Pavani et al. Pattern Recognition 43 (2010)

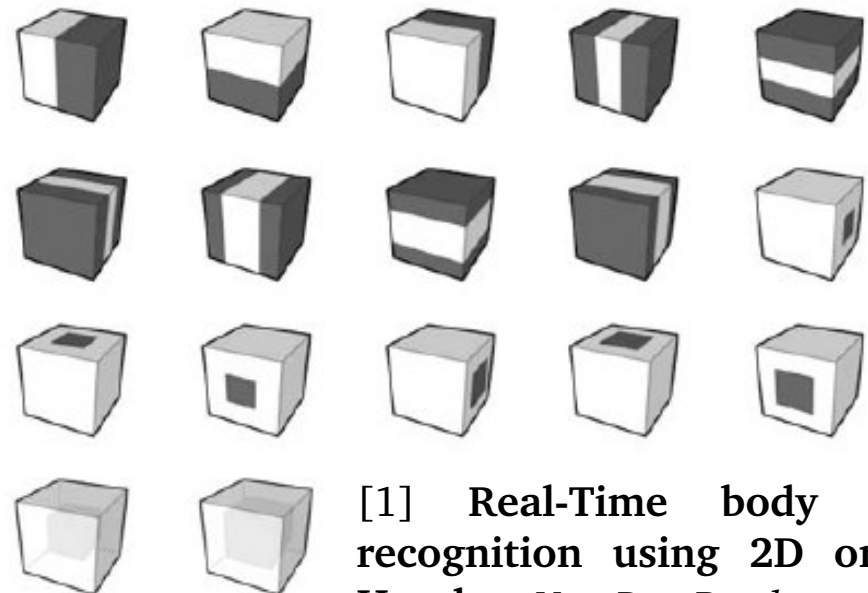


Haar-like *features* estensione al caso 3D



$$ii(x, y, z) = \sum_{x' \leq x, y' \leq y, z' \leq z} i(x', y', z')$$

La semplicità e la velocità del calcolo delle *features* è affidato all'**immagine integrale**



[1] Real-Time body pose recognition using 2D or 3D Haarlets Van Den Bergh et al. *Int. J. Computational Vision* 83 (2009)

Set di *features* da classificare

Per ciascun *voxel* possiamo ottenere un vettore di n *features* , $n \sim 10^4$

$$v_1 = \{f_1, f_2, \text{-----} f_n\}$$

Il *target* $\{0, 1\}$ che proviene dalla maschera 3D dell'ippocampo segmentato a mano

$$v_1 = \{f_1, f_2, \text{-----} f_n, t\}$$

...

$$v_N = \{f_1, f_2, \text{-----} f_n, t\}$$

Dato il numero elevato di *features* e considerato il grande quantitativo di *voxel* è necessario POTER classificare questi dati in maniera adeguata, al fine di ottenere un classificatore da impiegare per future predizioni su nuovi dati. È necessario un *classificatore ACCURATO*.

Valutazione

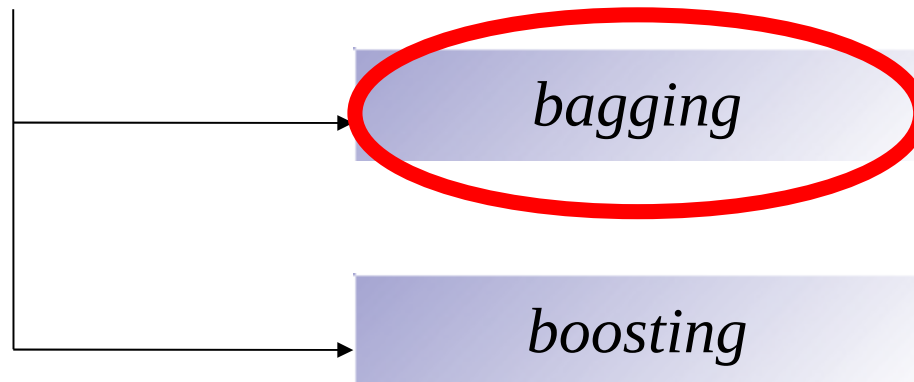
- ✓ **ERRORE:** rapporto tra il numero di oggetti classificati non correttamente ed il totale
 - **VARIANZA** errore in fase di *training*
 - **BIAS** errore in fase di classificazione
- ✓ **ACCURATEZZA: 1-ERRORE**
- ✓ **EFFICIENZA**
 - in fase di apprendimento
 - in fase di classificazione
- ✓ **SCALABILITA'**
 - rispetto al data set di training
 - rispetto al numero di variabili
- ✓ **ROBUSTEZZA** gestione del rumore e dei dati mancanti
- ✓ **INTERPRETABILITA'** comprensibilità e compattezza del modello

Il *Meta-learning*

Esistono vari modi per migliorare l'accuratezza di un sistema di classificazione:

- Cambiare schema/parametri;
- Modificare l'input:
{fare altre selezioni di attributi, sostituire valori mancanti, rimuovere *outliers*...};
- Modificare l'output:

Introdurre una **META-CLASSIFICAZIONE** che mira a creare un modello che sia combinazione di più modelli di base.

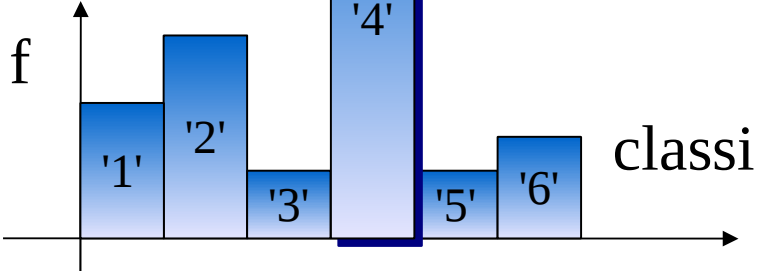
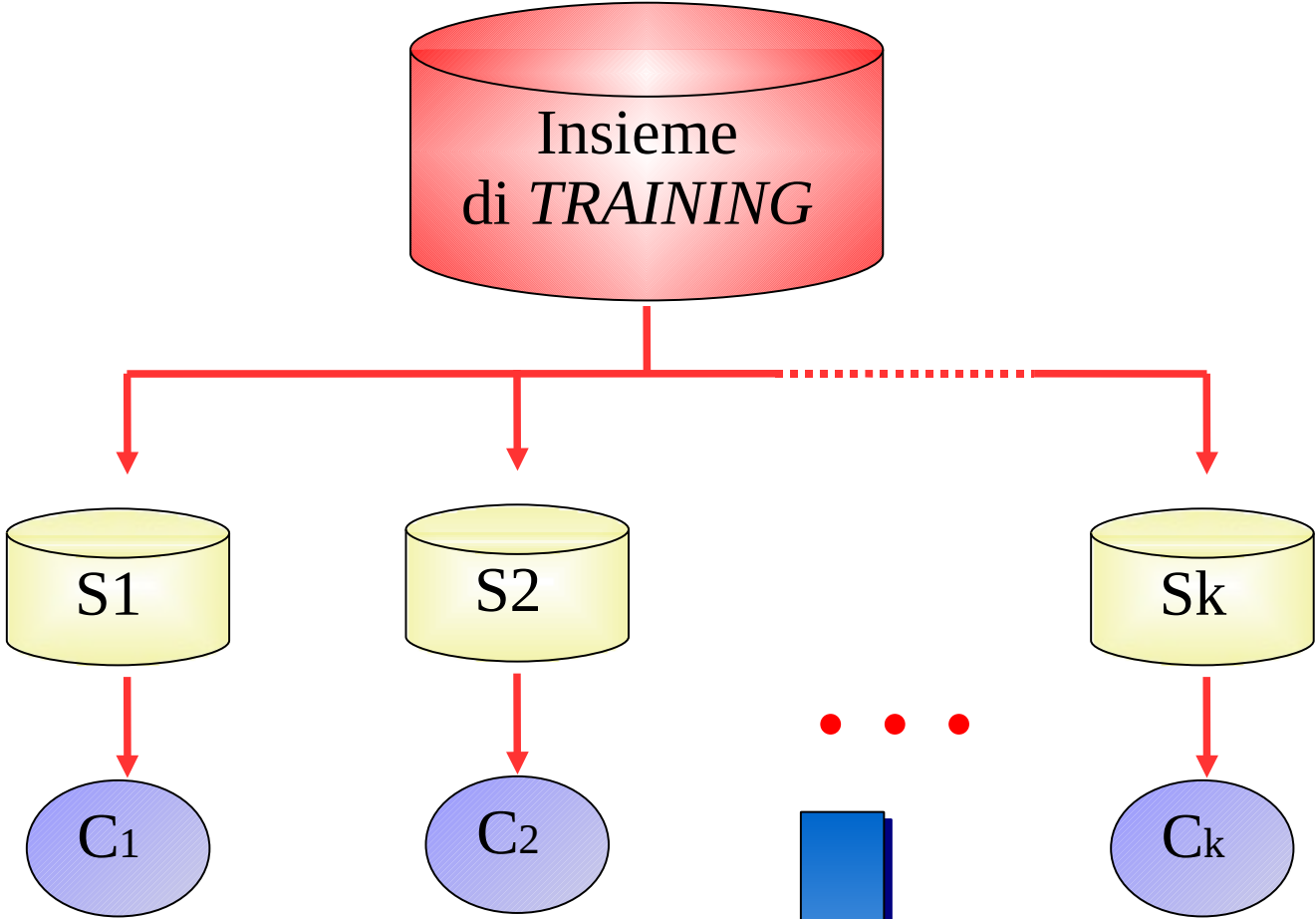


Combinazione di più classificatori

Campioni del dataset
ottenuti per selezione
random

Serie di Classificatori

Combinazione dei vari Classificatori
per maggiore VOTAZIONE tra le classi.



Combinazione di più classificatori: predizione errata

La probabilità che un insieme di n classificatori indipendenti, supposti simili con uno stesso rate di errore e fornisca una predizione errata è data da:

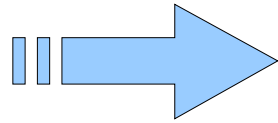
$$p = \sum_{i=n/2+1}^n \binom{n}{i} e^i (1-e)^{n-i}$$

Per $n=25$ ed $e = 0,35$ si ottiene una $p = 0,06$.

L'impiego di più classificatori dunque consente di ridurre la *varianza*.

Scelta del classificatore di base: un **albero decisionale**

DATI: insieme di casi caratterizzati da stesse *features* e con classe di appartenenza nota.



Si può creare un albero decisionale che dalla 'radice' sino alla 'foglia' selezioni ad ogni livello la variabile più discriminante.



Nodo t1



Nodo t2



Foglia f1



La foglia è generata quando i dati in arrivo appartengono tutti alla stessa classe

Caratteristiche di un **albero decisionale**

L'obiettivo principale di un albero decisionale è quello di realizzare un classificatore accurato e semplice. Nell'accrescimento di un albero decisionale si può valutare ad ogni livello l'*indice di impurità*

Impurità di **GINI**

$$i(t) = 1 - \sum_{j=1}^c P_j^2$$

- Per una regione pura $i(t) = 0$
- Per una regione altamente contaminata $i(t) = (c-1)/c$

L'accrescimento dell'albero sino al raggiungimento delle foglie (impurità zero) è possibile ma rischia di penalizzare il classificatore col rischio di *overtraining*.

È possibile scegliere tra possibili diversi criteri di 'stop' nella fase di accrescimento.



Random Forests

LEO BREIMAN

Statistics Department, University of California, Berkeley, CA 94720

Editor: Robert E. Schapire



Ciascun albero viene così creato:

Siano $Z = \{z_1, z_2, \dots, z_N\}$ i dati a disposizione:

- Vengono generati k nuovi insiemi di N elementi con la tecnica del *bootstrap*

BAGGING
Bootstrap AGGregatING



$$s_1 = \{z_1, z_3, z_2, z_1, \dots, z_7, \dots\}$$

$$s_k = \{z_2, z_3, z_6, z_2, \dots, z_8, \dots\}$$

- Con i k insiemi accresco k alberi;
- Selezioniamo casualmente un sottinsieme di $m \ll M$ *features* discriminanti;
- Ogni nodo esegue una classificazione (*split*) unidimensionale su la più discriminante delle m *features*;
- m rimane costante per tutto il processo di accrescimento della foresta;
- Ogni albero viene accresciuto senza '*pruning*' fino al raggiungimento di sole foglie;
- Esiste un range ottimale in cui scegliere l'opportuno valore di m e si ricorre all' errore OOB OUT-OF-BAG (metodo alternativo alla *validation*).

OUT-OF-BAG

Nel *random forests* non è necessaria la *cross-validation* per avere una stima UNBIAS dell'errore sul test set. Questo è già fatto durante il run:

Ogni albero viene accresciuto usando un diverso *bootstrap-sample* dai dati originali. Quindi si dimostra che il *k-esimo* albero sarà costruito con una determinata percentuale del *dataset*.

Tenendo da parte tutti i dati non utilizzati in ciascun albero, possiamo usarli in seguito per effettuare il test:

es. *Albero k* : $Z_{in} = \{z_1, z_3, z_3, z_5, z_7, \dots, z_{10}\}$ $Z_{out} = \{z_2, z_4, z_6\}$

Classifichiamo i Z_{out} sul *k-esimo* albero.

OOB si ottiene facendo la media su tutti i casi di Z_{out} dell'intera foresta.

An Empirical Evaluation of Supervised Learning in High Dimensions

Rich Caruana

Nikos Karampatziakis

Ainur Yessenalina

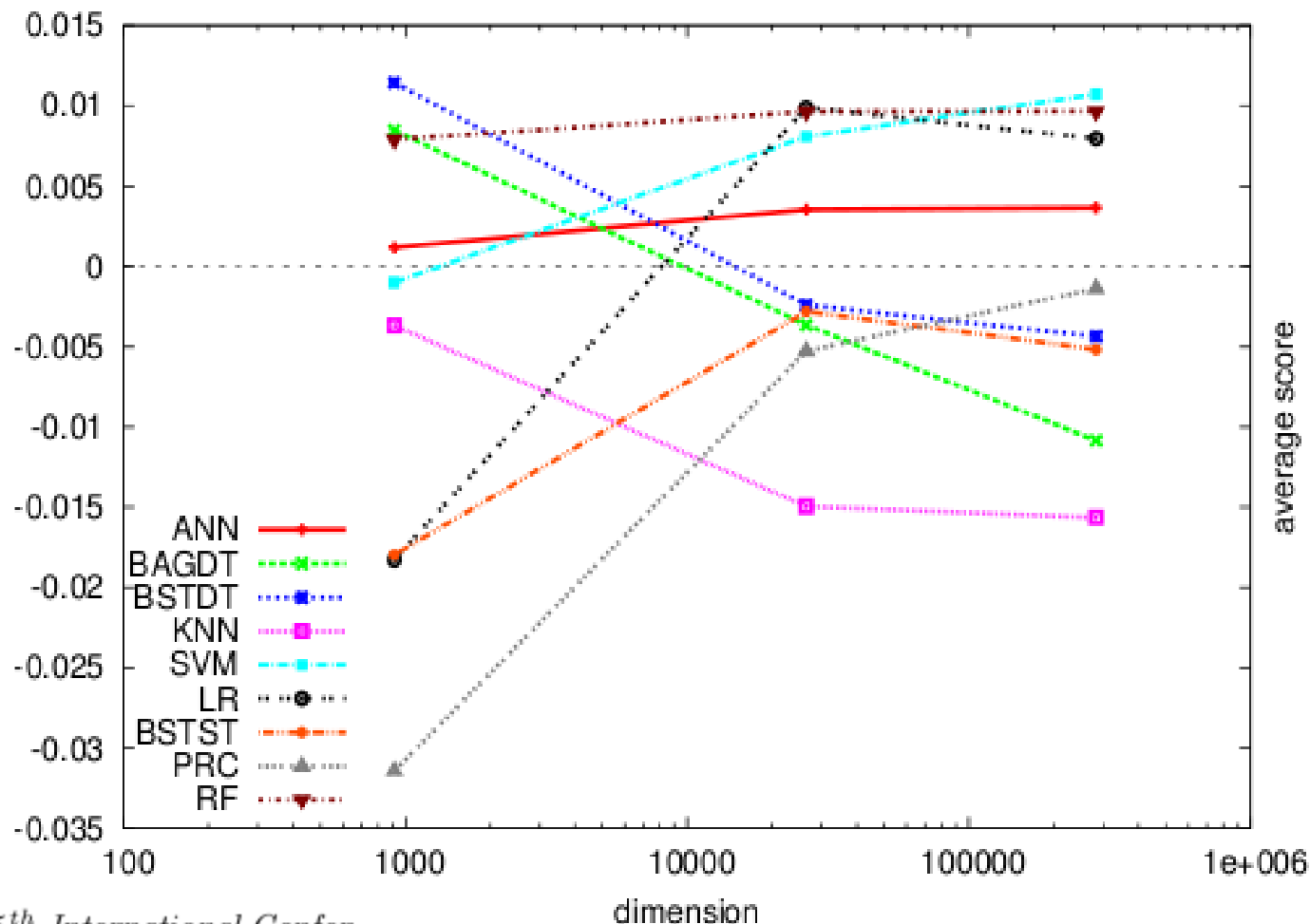
Department of Computer Science, Cornell University, Ithaca, NY 14853 USA

CARUANA@CS.CORNELL.EDU

NK@CS.CORNELL.EDU

AINUR@CS.CORNELL.EDU

- Rete neurale
- Bagged decision Trees
- Boosted trees
- K-Nearest Neighbor
- Support Vector Machine
- Logistic Regression
- Boosted Stumps
- Perceptrons
- Random Forests

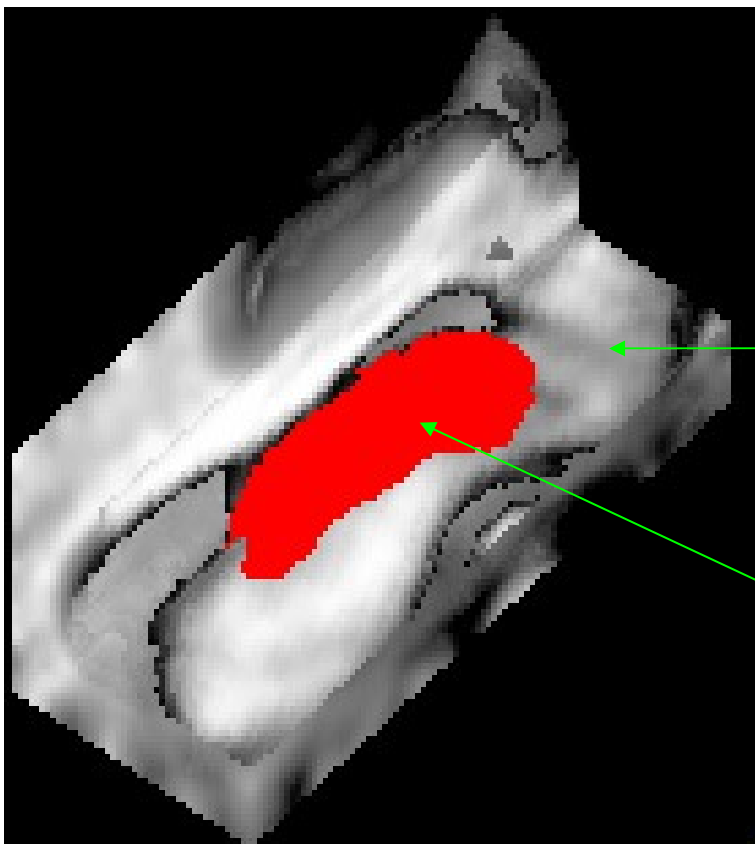


Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

CARATTERISTICHE del *RANDOM FORESTS*

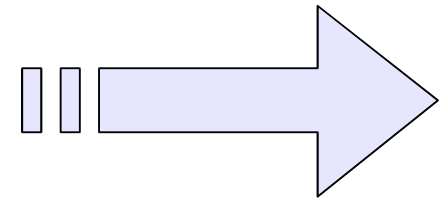
- ✓ Eccellente nell'accuratezza rispetto agli altri algoritmi;
- ✓ Efficiente su grandi *dataset*;
- ✓ Può trattare un numero molto elevato di *features* senza necessità di doverle ridurre;
- ✓ Al termine fornisce una stima di quali *features* sono importanti nella classificazione;
- ✓ Mantiene la sua efficienza anche in caso di dati mancanti;
- ✓ È molto insensibile agli *outliers* e ne facilita l'individuazione;
- ✓ Le foreste generate possono essere 'salvate' e riutilizzate per fare previsioni su dati nuovi.

Verso i primi risultati



Box contenente
l'ippocampo

Maschera 3D
dell'ippocampo
segmentato a mano



→ *Features* estratte per ciascun voxel

{x, y, z, Mean3, StandardDeviation3, Variance3, Kurtosis3,
Skewness3,... M5, SD5,.. M7, SD7 ...}

calcolo dei momenti su finestre di dimensioni 3x3x3, 5x5x5, 7x7x7 centrate nel *voxel*

→ **483859** *voxels* elaborati appartenenti ad una sola scatola ippocampale di misura 59x139x59 *voxel*.

Primi risultati

Per ragioni computazionali, sono stati selezionati due insiemi disgiunti tra tutti i *voxels*

‣ *TRAINING set 40,000 voxel {33387, 6614}*

‣ *TEST set 15,000 voxel {14765, 236}*

‣ *Foresta con 100 alberi*

training	No ippocampo	ippocampo
No ippocampo	33274	182
ippocampo	113	6432

test	No ippocampo	ippocampo
No ippocampo	14253	18
ippocampo	512	218

Training error 0.74 %

Test error 3.53 %

Conclusioni e sviluppi futuri

- ✓ L'approccio basato su classificatori semplici applicati a numeri molto elevati di *features* altrettanto semplici ha fornito, in molti contesti applicativi del *pattern recognition*, risultati significativi;
- ✓ Una declinazione di questo approccio è basata su *Random Forest* e *Features Haar-like*;
- ✓ Ispirati da un gran numero di ricerche presentate al MICCAI-09 abbiamo avviato lo sviluppo e la sperimentazione di questo approccio per la segmentazione dell'ippocampo nell'ambito delle attività *brain* di MAGIC-5.
- ✓ I primi risultati sono incoraggianti, sia per la relativa semplicità di sviluppo ed implementazione sia per i risultati ottenuti.

A photograph of a dense forest with sunlight filtering through the trees, creating a bright, dappled light effect. The text "Grazie per l'attenzione..." is overlaid in the center.

Grazie
per l'attenzione...

Bibliografia

- “Combining Pattern Classifiers, Methods and Algorithms” *L.I. Kuncheva 2004 Wiley&Sons*
- <http://dns2.icar.cnr.it/pontieri/didattica/LabSI/lezioni/>
- “Pattern Classification” *Duda, Hart, Stork 2001 Wiley&Sons*
- “Discriminative Learning for Medical Imaging Analysis” *Zhuowen Tu at MICCAI 2009*
- “Validation of a fully automated 3D hippocampal segmentation method using subjects with Alzheimer's disease mild cognitive impairment, and elderly controls” *Morra et al. NeuroImage 43 (2008)*
- http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm