Resoconto sulla “AMS Short Course on Artificial Intelligence” Atlanta 28–29 Gennaio 2006

Agostino Manzato
ARPA – Agenzia Regionale per la Protezione dell’Ambiente del FVG,
settore OSMER – Osservatorio Meteorologico Regionale, Visco (UD), Italy

1 Introduzione

Nel contesto del 86–esimo Annual Meeting della American Meteorological Society (AMS) si è svolto il quarto Short Course organizzato dal comitato “Artificial Intelligence Applications to Environmental Science” dell’AMS. Il sito del comitato organizzatore è

http://www.ametsoc.org/stacpges/ai/

Il comitato è presieduto dal 2004 ad ora dal prof. Caren Marzban delle Università di Washington e di Oklahoma (http://www.nhn.ou.edu/~marzban/), mentre per i prossimi anni la presidenza è passata alla professoressa Sue Ellen Haupt della Pennsylvania State University. Prima del prof. Marzban, il fondatore e primo presidente del comitato è stato il prof. Vladimir Krasnopolsky dell’NCEP/NWS/NOAA e Università di Maryland (http://polar.ncep.noaa.gov/mmab/people/kvladimir.html).

I partecipanti sono stati circa una ventina, in rappresentanza di diverse nazioni. Di seguito potete trovare un riassunto delle lezioni, mentre tutte le presentazioni in formato PDF sono disponibili in un CD in biblioteca.

2 Rich Caruana

INTRODUCTION TO ALGORITHMS FOR SUPERVISED LEARNING
caruana@cs.cornell.edu,
Cornell University, Ithaca (NY), USA
http://www.cs.cornell.edu/~caruana

Era l’unico insegnante che aveva un back-ground prettamente informatico (Computer Science) piuttosto che legato alle scienze ambientali. Probabilmente per questo

è stato quello che ha dimostrato maggior esperienza relativamente a tutte quelle tecniche che vanno sotto il nome generico di "machine learning" e in particolare al loro confronto. Le tecniche più utilizzate attualmente sembrano essere: alberi (*Decision Tree*, DT), reti neurali (ANN), Support Vector Machine (SVM, Vapnik, V. 1995: *The Nature of Statistical Learning Theory*, Springer). Altre tecniche, quali i *K–Nearest Neighbor* o altri metodi "memory based", sembrano non essere altrettanto performanti come le tre tecniche appena citate.

Ha fatto degli esempi di alberi (presi dal testo "Machine Learning" di Tom Mitchell, McGraw Hill 1997) che sono fondamentalmente una sequenza di condizioni su valori di soglia dei vari predittori per arrivare ad una probabilità che l'evento da prevedere si verifichi. Il tutto assume la forma di grafo, in cui le diverse condizioni si biforcano dal punto attuale in base alle condizioni su un predittore diverso da quelli già considerati. La costruzione dell'albero si basa su un dataset di dati (*training set*) che deve sempre essere tenuto separato dal dataset usato alla fine per verificare le performance (*validation set*). Per costruire l'albero si parte scegliendo il predittore che di più separa le classi di appartenenza (evento sì o evento no) in base ad una soglia, dopodiché, sui due sottoinsiemi creati, si cerca per ognuno quale sia il nuovo predittore che separa nel modo migliore i casi con accadimento dell'evento. Questo processo viene iterato finché si raggiunge il massimo grado di previsione possibile per quel dataset, ovvero i sottoinsiemi generati non sono diventati sufficientemente piccoli. L'uscita di ogni nodo finale è il valor medio degli elementi contenuti. Per misurare quanto bene separi i casi la soglia su un dato predittore si usa di solito *l'entropia* $= -\frac{1}{N}\sum_{i=1}^{N}[t_i \cdot \ln(y_i) + (1-t_i)\cdot\ln(1-y_i)]$, dove t_i è il valore osservato per l'evento i-esimo, mentre y_i è la previsione fatta dal predittore.

Il vantaggio degli alberi è che sono molto veloci computazionalmente e che sono "intellegibili", ovvero si capisce "a occhio" come l'output viene calcolato, mentre le reti neurali sono spesso considerate come "scatole nere". Il difetto maggiore degli alberi è che per molti predittori (o problemi complessi) diventano grandissimi e sono difficilmente gestibili. Una tecnica applicata in tali casi è quella del *pruning* che consiste nel "potare" via i rami che hanno poche biforcazioni sotto di loro o che si riferiscono a pochi casi (si rischia in tal caso di overfittare il training). Particolarmente interessante il fatto che esistano versioni di alberi fatte per risolvere problemi di regressione (predittando continuo) e non solo di classificazione (predittando categorico), vedi ad esempio i CART (Breiman et al. 1984: *Classification and Regression Trees*, Chapman & Hall, New York). In **R** gli alberi sono implementati dalla libreria `rpart`.

Negli ultimi anni gli alberi erano stati messi un po' in disparte rispetto alle reti neurali, finchè non è stata introdotta la tecnica del *bagging* (Breiman, L., 1996: *Bagging predictors*, Machine Learning, **24**, 123–140) che li ha riportati in auge. Il bagging consiste nell'estrarre dal dataset di training tante sue diverse realizzazioni (tramite il *bootstrap*, che permette ripetizioni anche degli stessi elementi). A questo punto si crea un albero diverso per ognuno di questi training dataset e alla fine si prende come output la *media* delle previsioni fatte da tutti questi alberi. In realtà

la tecnica del bagging si può applicare anche ad altri metodi (esempio alle reti neurali), ma si sposa particolarmente bene con gli alberi grazie alla loro velocità di esecuzione. In realtà la filosofia sottostante il bagging è quella di lasciare che ogni modello overfitti i dati di training, ma essendo questi diversi per ogni modello, la media dei diversi modelli mantiene solo le caratteristiche comuni su tutti i dataset. In **R** i Bagged Decision Tree sono disponibili all'interno della libreria `ipred`.

Nell'ultima parte della lezione ha mostrato i risultati degli esperimenti che ha fatto per confrontare una decina di tecniche diverse (Logistic/Ridge regression, K–Nearest Neighbor, Bagged DT, Boosted DT, SVM, ANN, Naïve Bayes, Random Forests, Boosted Stumps) tutte riscalate col metodo di Platt (1999). Alla fine trova che i risultati migliori li danno i Boosted[1] DT, le Random Forecast, le reti neurali e i Support Vector Machine. Ma la cosa più interessante è *l'ensemble selection method*, in cui i risultati delle previsioni fatte con diverse tecniche vengono a loro volta combinati, mettendo insieme i metodi che danno i risultati migliori quando combinati assieme (stepwise selection). Ovviamente il peso computazionale è molto alto, ma sembra che le performace siano le massime raggiungibili. Gli esperimenti più recenti li stanno facendo per far simulare ad una rete neurale (di tipo "vanilla") i risultati prodotti dal ensemble selection model, in modo da avere poi un'implementazione operativa molto più semplice e restringere la parte computazionalemente più onerosa solo nella fase di "studio e risoluzione" del problema.
Concludo riportando le tre regole d'oro dell'AI secondo Caruana:
1. Mai calcolare le performance su dati che sono stati utilizzati durante la scelta del modello.
2. Riportare sempre più misure di performance e le loro definizioni operative.
3. Nessun modello può essere migliore di tutti gli altri in tutte le misure, per questo gli ensemble model vanno molto di moda.

3 Valliappa Lakshmanan

MACHINE LEARNING TECHNIQUES ON SPATIAL GRIDS
Valliappa.Lakshmanan@noaa.gov,
National Severe Storm Laboratory e CIMMS Università dell'Oklahoma, Norman OK, USA
http://cimms.ou.edu/~lakshman

È un esperto di analisi di mappe radar e da satellite. La sua lezione riguardava

[1] Il *boosting* è una tecnica simile al bagging, ma invece di fare la media aritmetica dei diversi modelli fa una media pesata in base alle loro performance sui diversi bootstrap del training dataset. Per approfondimenti vedi Freund, Y. and Schapire, R. E. 1996: *Experiments with a new boosting algorithm*, in Machine Learning: Proc. Thirteenth International Conference, 148–156, Morgan Kauffman, San Francisco.

le tecniche basilari dell'*image processing* al fine di riconoscere le caratteristiche peculiari di un'immagine. L'idea di base è che è molto più semplice seguire l'evoluzione temporale di alcune caratteristiche estratte da un'immagine (esempio il nucleo di una cella temporalesca) piuttosto che studiare l'evoluzione temporale di tutta l'immagine.

Alla base dell'image processing ci sono i filtri di convoluzione, in cui ogni punto dell'immagine viene sostituito con un valore derivato da tutte le misure in un suo intorno. I diversi filtri nascono dalla dimensione di tale intorno e dalla funzione usata per pesare i valori dell'intorno (matrice dei pesi). Ad esempio si può fare la media (la cosa peggiore, meglio usare la mediana) o scegliere il massimo (fusione delle celle) o il minimo (erosione delle celle) o usare un filtro gaussiano (smoothing). Quello che viene fatto operativamente è di applicare tanti di questi filtri per ognuna delle caratteristiche (*features*) che si cercano (nuclei ad alta riflettività, strutture allungate come le squall lines, presenza di rotazioni, ecc.) e per le diverse scale che si vogliono indagare. Visto che il numero di filtri da applicare può diventare molto alto, si preferisce studiare solo le singole zone dell'intera immagine che contengono strutture interessanti: ovvero si applica *l'image segmentation*. Un'altro modo di velocizzare il processo è quello di usare le Fast Fourier Transforms.

Una volta estratte tutte le features ricercate nell'immagine le si usa come input di un classificatore (tipicamente una rete neurale) per fare l'identificazione dell'oggetto (ad esempio la presenza o meno di un mesociclone). Applicazioni interessanti di queste metodologie si possono trovare per esempio in Lakshmanan et al. 2006: *An Automated Technique to Quality Control Radar Reflectivity Data*, in pubblicazione su J. Applied Meteorology ma già scaricabile dal sito di Lakshmanan, o in Lakshmanan, V., R. Rabin, and V. DeBrunner, 2003: *Multiscale storm identification and forecast*, J. Atm. Res., 367–380. Infine colgo l'occasione per informare gli interessati dell'esistenza di un software gratuito per l'analisi delle immagini (in particolare mappe radar), chiamato "Warning Decision Support System", sviluppato da NSSL e CIMMS (Norman, OK) e disponibile gratuitamente per scopi di ricerca all'URL:

http://www.wdssii.org

4 Caren Marzban

ON THE CONNECTION BETWEEN AI AND STATISTICS e PERFORMANCE ASSESSMENT
marzban@stat.washington.edu,
Università di Washington (Seattle, WA) e Center for Analysis and Prediction of Storms (CAPS) dell'Università di Oklahoma (Norman OK), USA
http://www.nhn.ou.edu/~marzban
Caren Marzban è l'organizzatore del corso e ha fatto due presentazioni distinte.

Nella prima ha cercato di dimostrare come la "statistica classica" e "l'AI" non siano più in contrapposizione tra di loro, ma piuttosto siano modi diversi e *quasi* equivalenti per fare le stesse cose. Esempi di statistica classica sono la linear discriminant analysis, la clustering analysis, la linear o logistic regression, le Principal Components. Invece esempi di AI sono considerate le reti neurali, i Genetic Algorithm, le SVM, le Self Organized Map (Kohonen, T., 1984: *Self–Organization and Associative Memory*, Springer–Verlag: New York), i K-Nearest-Neighbor, la Fuzzy Logic.

Una tecnica che sta a metà strada tra le due filosofie può essere considerata la Naïve Bayes, che viene spiegata di seguito. Siano $x_1,...,x_n$ un insieme di predittori e sia y il predittanto discreto ($y =$ YES, o $y =$ NO). La probabilità *a priori* che si verifichi l'evento è data semplicemente da $p(y = \text{YES}) = N_{\text{YES}}/(N_{\text{YES}} + N_{\text{NO}})$, con N_{YES} numero di casi in cui si verifica l'evento e N_{NO} in cui non si verifica. $p(\text{YES})$ viene chiamata anche *base rate* dell'evento. La probabilità di verosimiglianza (*likelihood*) che l'evento si verifichi è la distribuzione dei valori dei predittori studiando solo il sottoinsieme dei casi YES e si indica con la notazione $p(x_1,...,x_n\,|\,\text{YES})$. Il teorema di *Bayes* ci dice come calcolare la probabilità *a posteriori* che un evento si verifichi per una data combinazione dei valori dei predittori, $p(YES\,|\,x_1,...,x_n)$:

$$p(YES\,|\,x_1,...,x_n) = \frac{p(x_1,...,x_n\,|\,YES)\cdot p(YES)}{p(x_1,...,x_n)}, \tag{1}$$

dove $p(x_1,...,x_n)$ è la probabilità (congiunta) a priori con cui si verifica tale combinazione di predittori. Ovviamente la stessa probabilità a posteriori si può calcolare nel caso in cui l'evento *non* si verifichi sostituendo i YES con i NO. Se un classificatore dà come output la classe (YES o NO) che ha la massima probabilità a posteriori allora *minimizza la possibilità di sbagliare*. Il problema è che la stima delle probabilità condizionate a tutte le possibili combinazioni dei predittori, $p(x_1,...,x_n\,|\,y)$, non è banale.

La tecnica *Naïve Bayes* si basa su un'approssimazione della formula precedente. Ovvero si fa l'ipotesi che la probabilità condizionata dei predittori per i soli casi attivi sia proporzionale al prodotto delle likelihood di ogni singolo predittore, ovvero che la likelihood di ogni predittore sia statisticamente indipendente dai valori degli altri predittori. In tal caso la probabilità a posteriori di avere o non avere un evento per una data combinazione di predittori è semplicemente proporzionale a:

$$p(y|x_1,...,x_n) \propto p(y)\cdot\prod_{i=1}^{n} p(x_i|y), \tag{2}$$

con $y =$ YES per avere l'evento e $y =$ NO per non averlo. In questo modo le due likelihood di ogni singolo predittore sono facilmente calcolabili, ad esempio facendo un istogramma delle distribuzioni del predittore nei casi con e senza evento rispettivamente. Il classificatore costruito con questa tecnica legge i valori dei predittori $x_1,...,x_n$, calcola la probabilità a posteriori per la classe $y =$YES e per la classe

y =NO con la formula precedente e dà come output la classe che ha la probabilità maggiore. In **R** c'è il pacchetto `klaR` che implementa la Naïve Bayes e la documentazione si trova all'URL: http://cran.r-project.org/doc/packages/klaR.pdf.

Nella seconda lezione il prof. Marzban ha fatto una review dei metodi di verifica delle previsioni sia per casi binari che continui. Indicando con o l'osservazione dell'evento e con f la sua previsione si possono definire le seguenti fondamentali misure: $p(o)$ (event base rate, che è una costante del database), $p(f)$ frequenza della previsione, $p(o|f)$ probabilità che si verifichi l'evento o quando si prevede f, $p(f|o)$ probabilità che si preveda f quando si verifica o. Le ultime tre probabilità riassumono completamente le qualità della previsione.

Per le variabili binarie $p(f)$ è un numero che dovrebbe essere simile a $p(o)$, invece per variabili continue di solito si guarda l'istogramma. $p(o|f)$ è legato alla fattorizzazione "calibration–refinement" e di solito si plotta come un grafico che ha il valore previsto in ascissa e il valor medio dell'osservazione per classi di valore previsto in ordinata. Una previsione è *reliable* se tale grafico tende alla diagonale. Infine $p(f|o)$ è legato alla fattorizzazione "likelihood–base rate" e di solito si plotta con il valore osservato in ascissa e quello previsto in ordinata. La conclusione della lezione è un incoraggiamento ad usare i grafici (inclusa la curva di ROC), possibilmente con le barre di errore, piuttosto che singole misure scalari perché "Performance is a *multifaceted* thing".

5 Sue Ellen Haupt

APPLICATIONS OF GENETIC ALGORITHMS IN THE ENVIRONMENTAL SCIENCES
seh19@psu.edu,
Pennsylvania State University, University Park (PA), USA

È una professoressa di meccanica dei fluidi, che ha scoperto gli algoritmi genetici come metodo per la soluzione di equazioni differenziali non risolvibili analiticamente. In particolare si occupa di dispersione degli inquinanti. Assieme al marito Randy Haupt ha scitto il libro *Practical Genetic Algorithms"* (John Wiley & Sons, second edition 2004). Storicamente gli algoritmi genetici sono nati come sequenze (dette *cromosomi*) di variabili binarie, ma oggigiorno vengono usati soprattutto nella loro estensione a variabili continue. Ognuno di questi cromosomi rappresenta una possibile soluzione del problema, ovvero la sequenza dei parametri che si stanno cercando. I genetic algorithm sono un metodo di *ottimizzazione* per trovare il cromosoma migliore, ovvero quello che minimizza la *cost function* prescelta. Ad

esempio nel caso della soluzione di un'equazione parametrica un cromosoma può rappresentare la sequenza dei parametri cercati.

Si comincia con una distribuzione casuale di cromosomi e li si sottopone a due diversi processi: il primo è il *mating*, dove due cromosomi diversi vengono combinati insieme (prendendo alcuni bit da uno e gli altri dell'altro) per creare un cromosoma nuovo, e l'altro è il *mutation* in cui alla popolazione di cromosomi attuale vengono aggiunti altri generati dal cambiamento casuale di alcuni bit in alcuni cromosomi. La vera *arte* dei genetic algorithm è quella di stabilire la miglior cost function e le regole di mating e muting, affinché vengano favoriti i cromosomi migliori e scartati quelli peggiori.

Computazionalmente non sono molto veloci ma ben si adattano a computer con più processori in parallelo. Un grosso vantaggio di tale tecnica è che la funzione da minimizzare non serve che sia "continua e differenziabile" come in molte altre tecniche di minimizzazione (gradient descent, metodo di Newton, ecc.), per cui, ad esempio, uno potrebbe usare il Kuipers Skill Score della classificazione corrispondente a quel cromosoma come cost function. Questa è la ragione per cui vanno di moda i genetic algorithm come metodo di ottimizzazione dei pesi delle reti neurali in problemi di classificazione. Una buona introduzione ai genetic algorithm, ma anche ad altre tecniche, può essere trovata in http://www.wikipedia.org, l'enciclopedia libera più grande di Internet.

6 John K. Williams

FUZZY LOGIC
jkwillia@ucar.edu,
RAL NCAR, Boulder (CO), USA
http://www.rap.ucar.edu/staff/williams-staff.html

Apparentemente la Fuzzy Logic è passata di moda negli ambienti AI. Nonostante questo ci sono gruppi che la usano ancora e tra questi, in ambito meteo, il Research Applications Laboratory dell'NCAR è uno dei più significativi. La Fuzzy Logic nasce nel 1965 con l'articolo di Lotfi A. Zadeh *Fuzzy sets* (Inf. Control **8**, 338–353). Le sue applicazioni si diffondono molto dalla fine degli anni '80 e in campo meteo possiamo notare tre applicazioni importanti nella seconda metà degli anni '90: NCAR–RAP Doppler radar Microburst Automatic Detection (MAD) algorithm, 1994; NCAR–RAP Improved Moments Algorithm (NIMA) per wind profilers, 1998, e Vivekanandan et al. 1999: *Cloud Microphysics Retrieval Using S–band Dual–Polarization Radar Measurements*, Bulletin of AMS: Vol. **80**, No. 3, 381–388.

La filosofia Fuzzy è quella di unire lo studio dei dati con le informazioni provenienti da "esperti", definendo delle regole (fuzzy sets) che "modellano" la classifi-

cazione. A differenza dell'insiemistica classica, in cui gli insiemi di appartenenza hanno dei confini netti, nel caso dei fuzzy sets questi confini possono essere "sfumati", per tener conto dell'incertezza che si ha nella transizione tra una classe e l'altra. In tal modo, in base al valore numerico di un predittore x (o di una combinazione di più predittori) si definisce la funzione di "membership" per ogni classe, disegnando una curva che varia tra 0 e 1 sul dominio di x. Sulle diverse funzioni di membership costruite si possono applicare operatori logici (unione, intersezione, ecc.) per combinarle insieme, realizzano la "fuzzy inference". In particolare esistono almeno due *stili* di fuzzy inference: quella di Mamdami (1975) e quella di Sugeno (1985). All'NCAR usano una variante dello stile Sugeno che si chiama *Consensus Fuzzy Reasoning*, in cui vengono inseriti anche dei valori di *confidenza* che tengono conto di quanto la misura di ogni predittore sia "affidabile" (è una specie di errore associato ad ogni x). Poi c'è la possibilità di avere le funzioni di membership con dei parametri, che vengono ottimizzati (*tuning*, effettuato spesso con reti neurali o algoritmi genetici) in base ai dati osservati.

Molte applicazioni sono state fatte per lo studio della qualità dei dati, come ad esempio l'Intelligent Outlier Detection Algorithm (IODA) e il Radar Echo Classifier (REC), oppure alla previsione, come ad esempio il Thunderstorm AutoNowcast System (ANC) e l'Integrated Turbulence Forecasting Algorithm (ITFA). Un elenco di applicazioni meteorologiche della Fuzzy Logic si può trovare al sito:

http://www.chebucto.ns.ca/Science/AIMET/applications/

7 Antonello Pasini

NEURAL NETWORK MODELING IN CLIMATE CHANGE STUDIES

pasini@iia.cnr.it,
CNR - Istituto sull'Inquinamento Atmosferico, Roma, ITALY
http://www.iia.cnr.it/iia/CurriculaRicFP/PasiniFP.htm

È un un fisico teorico che si è dedicato alla meteorologia lavorando prima nell'Aeronautica Militare e successivamente nel CNR. È un esperto di reti neurali, che ha applicato sia per prevedere variabili fisiche nel boundary layer, come la visibilità (Pasini, Pelino and Potestà, 2001: *A neural network model for visibility nowcasting from surface observation: Results and sensitivity to physical input variables*, J. Geophysical Res., **106**, n. D14) o la concentrazione di Radon al suolo (Pasini and Ameli, 2003: *Radon short range forecasting through time series preprocessing and neural network modeling*, Geophysical Res. Letters, **30**, n. 7), sia per studi di climatologia (Pasini, Lorè and Ameli, 2006: *Neural network modelling for the analysis of forcings/temperatures relationships at different scales in the climate system*, Ecolo-

gical Modelling, **191**, 58–67) o più teorici sulla predicibilità dell'atmosfera (Pasini and Pelino, 2005: *Can We Estimate Atmospheric Predictability by Performance of Neural Network Forecasting? The Toy Case Studies of Unforced and Forced Lorenz Models*, CIMSA 2005 IEEE, Int. Conf. on Computational Intelligence for Measurement Systems and Applications).

Nella sua lezione, piuttosto che spiegare come funzioni una rete neurale, ha parlato dei risultati più recenti che ha trovato applicando le reti neurali alla climatologia. La filosofia di base è che i modelli climatici che cercano di simulare esplicitamente l'atmosfera (incluse le sue interazioni con oceani e suolo) possono essere poco precisi, a causa dell'elevato numero di processi in atto. L'idea è quindi di provare a simulare l'andamento negli ultimi 150 anni della temperatura con una rete neurale, provando a usare come input combinazioni di tre parametri: il forcing naturale, quello antropogenico e il SOI (South Oscillation Index, legato all'ENSO). Il risultato mostra che il forcing antropogenico è fondamentale per ottenete buoni risultati, che diventano molto precisi aggiungendo anche il SOI. A scala regionale le cose possono essere diverse, ad esempio l'andamento della temperatura (ma anche i temporali) in "Central England" è molto legato al NAO (North Atlantic Oscillation), che quindi va tenuto in considerazione nei lavori di *downscaling* a livello europeo. Interessante il fatto che non abbiano utilizzato le reti neurali costruite per fare previsioni sugli scenari futuri, in quanto ogni modello statistico si basa sull'ipotesi di *stazionarietà* della serie, ovvero che la varianza del fenomeno non aumenti in futuro. Ciò non vale per le previsioni di temperatura dei prossimi 50 anni! Per questo motivo non possiamo estrapolare i risultati statistici trovati in passato su uno scenario futuro che può essere molto diverso.

La seconda parte della lezione era dedicata allo studio della predicibilità del sistema "Lorenz '63". Ha introdotto i *bred vectors* (Toth and Kalnay, 1997: *Ensemble forecasting at NCEP and the breeding method*, Mon. Wea. Rev., **125**, 3297–3319) per vedere quanto due punti inizialmente vicini si allontanano durante iterazioni successive del modello. Detto in parole povere, si definisce bred vector $\delta\mathbf{x}(n)$ la distanza raggiunta dopo n step tra due run dello stesso modello inizializzato con condizioni leggermente diverse, $\delta\mathbf{x_0}$. La predicibilità del sistema si può stimare con il *bred growth rate*, definito come:

$$g = \frac{1}{n} \cdot \ln \frac{|\delta\mathbf{x}(n)|}{|\delta\mathbf{x_0}|}. \tag{3}$$

Se $g < 0.4$ la predicibilità è ancora accettabile, mentre se $g > 0.6$ è molto bassa. Il loro studio consiste nel far simulare ad una rete neurale l'evoluzione del sistema "Lorenz '63" e di una sua variante "forzata" (analogo del forcing antropogenico) e nello studiare la predicibilità di ogni punto del dominio. La rete neurale sembra essere in grado di distinguere bene le zone con diversa predicibilità e questo dimostra che è uno strumento adatto a sistemi complessi come l'atmosfera.

Infine ha fatto un accenno all'uso, sempre più diffuso, che si fa delle reti neurali nei lavori di *downscaling*. Ad esempio: Yuval and Hsieh 2003: *An adaptive*

nonlinear MOS scheme for precipitation forecasts using neural networks, Wea. Forecasting, **18**, 303–310; Trigo and Palutikof 1999: *Simulation of daily temperatures for climate change scenarios over Portugal: a Neural Network Model approach*, Clim. Research, **13**, 45–59; Kuligowski and Barros 1998: *Experiments in short–term precipitation forecasting using artificial neural networks*, Mon. Wea. Rev., **126**, 470–482. Pasini è anche autore di un libro sui cambiamenti climatici, che è disponibile presso la biblioteca dell'OSMER.

8 Sašo Džeroski

ANALYSIS OF ENVIRONMENTAL DATA WITH MACHINE LEARNING METHODS
Saso.Dzeroski@ijs.si,
Jožef Štefan Institute, Lubiana, SLOVENIA
http://www-ai.ijs.si/SasoDzeroski/SasoDzeroski.html

Lavora presso il dipartimento di Tecnologie della Conoscenza dell'Istituto Jožef Štefan di Lubiana e inoltre insegna al Politecnico di Nova Gorica (http://www.p-ng.si/en). Si è occupato della soluzione di molti problemi di ecologia applicata, usando tecniche di AI per la loro soluzione.

In particolare si è specializzato nel cercare di unire i due approcci distinti che di solito si applicano: il primo è quello di riferirsi a degli *esperti* che insegnano "le regole del problema" (conoscenza *a priori*), mentre il secondo è quello di inferire tali regole dallo studio dei dati (*empirical modeling*). Una via semplice di implementare tale filosofia è quella di descrivere il modello con delle equazioni parametriche in cui la forma delle funzioni viene dedotta dalla conoscenza a priori, mentre i parametri vengono fittati dallo studio dei dati. Purtroppo molte volte la dipendenza funzionale dai singoli predittori non è nota a priori, e quindi bisogna trovarla in altri modi. La tecnica utilizzata in tal caso è quella chiamata *equation discovery* (Langley et al. 1987: *Scientific Discovery*, MIT Press, Cambridge, MA; o vedi http://www-ai.ijs.si/~ljupco/ed). Tale tecnica è basata su una specie di *grammatica matematica*, che parte dalle funzioni più semplici e via via indaga le forme più complicate, in una specie di ricerca ad albero. Il vantaggio principale è che la soluzione analitica di un modello è molto più facilmente accettata di un'altra tecnica AI più "imperscrutabile" come le ANN o i SVM.

Purtroppo non ha spiegato i dettagli di come applicare tale tecnica, ma ha fatto una carrellata di problemi di cui si è occupato. A titolo d'esempio cito lo studio della crescita delle alghe nella laguna di Venezia o nel lago di Glumsoe (Danimarca), lo studio della popolazione di orsi in Slovenia o le correlazioni tra radon al suolo e attività sismica nella zona di Krsko (dove c'è la centrale nucleare).

9 Conclusioni

La discussione finale ha cercato di riassumere i lavori presentati nei due giorni in poche linee guida, riportate di seguito. Primo, individuare se si ha a che fare con un problema di classificazione o di regressione. Secondo, chiedersi se si ha una conoscenza a priori sufficiente a descrivere il problema. In caso affermativo usare la Fuzzy Logic o alternativamente i Genetic Algorith per trovare i parametri di un'equazione analitica che descriva tale conoscenza. In caso negativo usare altre tecniche di AI, quali i Baddeg DT o i SVM o le ANN se si tratta di classificazione, oppure le ANN o DT per regressione in caso di variabili continue (se i risultati della multiregressione lineare non fossero abbastanza soddisfacenti).

Per me è stata l'occasione per conoscere personalmente esperti del settore e per farmi un'idea di quali, tra le molteplici tecniche applicabili, siano quelle che hanno maggior credenziali e quanto sia difficile implementarle. In particolare, mi pare importante sottolineare come anche in AI l'approccio *multimodel* (bagging, boosting, ensemble selection) sia sempre più diffuso, anche contro l'overfitting.

Per l'OSMER è stata un'occasione di vedere come tecniche più sofisticate della semplice regresssione lineare o dell'analisi del discriminante trovino sempre più diffusione nei campi del controllo della qualità dei dati (consistenza, outliers,...), nella classificazione di eventi (presenza di mesocicloni o altre "strutture"), nella previsione a breve termine partendo dai dati osservati (nowcasting) o infine nel migliorare la previsione fatta dai modelli (downscaling). Per i dettagli di molte delle applicazioni citate in queste note (soprattutto per Williams e Lakshmanan) invito ad andare a guardare le loro presentazioni, disponibili in un CD in biblioteca.

Visco, 20 febbraio 2006.

Stretto all'angolo tra Caruana e Marzban...