ISSN/0393-3016

MARCO RAO

Dipartimento Fusione e Tecnologie per la Sicurezza Nucleare
Divisione Tecnologie Fisiche per la Sicurezza e la Salute
Laboratorio Sviluppo di Acceleratori di Particelle
e Applicazioni Medicali
Centro Ricerche Frascati, Roma

FRANCESCA CUBEDDU

Università di Roma Tre

MACHINE LEARNING E MODELLI SOCIALI PER L'EFFICIENZA ENERGETICA

Un esperimento qualitativo per la Regione Lazio

RT/2017/18/ENEA

AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

MARCO RAO

Dipartimento Fusione e Tecnologie per la Sicurezza Nucleare
Divisione Tecnologie Fisiche per la Sicurezza e la Salute
Laboratorio Sviluppo di Acceleratori di Particelle
e Applicazioni Medicali
Centro Ricerche Frascati, Roma

FRANCESCA CUBEDDU

Università di Roma Tre

MACHINE LEARNING E MODELLI SOCIALI PER L'EFFICIENZA ENERGETICA

Un esperimento qualitativo per la Regione Lazio

RT/2017/18/ENEA

AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE,
L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

I rapporti tecnici sono scaricabili in formato pdf dal sito web ENEA alla pagina
http://www.enea.it/it/produzione-scientifica/rapporti-tecnici

I contenuti tecnico-scientifici dei rapporti tecnici dell'ENEA rispecchiano
l'opinione degli autori e non necessariamente quella dell'Agenzia

The technical and scientific contents of these reports express the opinion
of the authors but not necessarily the opinion of ENEA.

MACHINE LEARNING E MODELLI SOCIALI PER L'EFFICIENZA ENERGETICA

Un esperimento qualitativo per la Regione Lazio

Marco Rao, Francesca Cubeddu

Riassunto

Questo lavoro presenta un esperimento condotto applicando delle tecniche di machine learning, in particolare una macchina a vettori di supporto (SVM) per l'analisi dei risultati contenuti in un dataset frutto di un'indagine qualitativa sul tema dell'efficienza energetica nella Regione Lazio. Lo scopo dell'esperimento è illustrare le potenzialità delle suddette tecniche nei problemi di classificazione e previsione di dati nel contesto della modellistica sociale. L'esperimento è condotto mediante l'uso di un codice sviluppato in R.

Parole chiave: Machine Learning, Modelli Sociali, Efficienza Energetica, R

Abstract

This paper presents an experiment conducted using machine learning techniques, in particular a support vector machine (SVM) machine in analyzing the results of a dataset from a qualitative survey on energy efficiency in the Lazio Region. The purpose of the experiment is to illustrate the potential of these techniques in the problems of classifying and predicting data in the context of social modeling. The experiment is conducted using a code developed in R.

Keywords: *Machine Learning, Social Models, Energy Efficiency, R*

INDICE

Introduzione - Cosa è il machine learning (ML) e perché è importante preoccuparsene

Machine learning, nell'attuale linguaggio comune[1], rappresenta uno slogan molto diffuso sia nell'ambito della ricerca che in quello industriale, in modo particolare nei settori a più elevato contenuto tecnologico. Tentando di evitare di dire il molto già detto, presente e ridondante in particolar modo sulla rete, limiteremo al minimo definizioni e inquadramento della materia segnalando alcuni riferimenti utili ad approfondire la materia, sia a livello divulgativo sia specialistico.

In lingua italiana, ML suonerebbe come apprendimento automatico, parole che mettono in evidenza di cosa si stia parlando: dell'area che rappresenta il cuore della ricerca sul tema dell'Intelligenza Artificiale (IA). In parole estremamente povere, abbiamo a che fare col ML ogni volta che intendiamo insegnare ad una macchina a "camminare con le sue gambe", vale a dire a sviluppare e consolidare autonomia nell'eseguire una serie di compiti.
Un esempio di quanto sopra detto potrebbe essere la capacità di riconoscere un volto umano in un immagine (magari distorta, sfocata, danneggiata o parziale), con la stessa abilità con cui potrebbe essere fatto ciò da una persona. Oppure la capacità di riconoscere nella foto di una lesione cutanea una patologia sospetta, con la sensibilità di un medico professionista.
Le applicazioni del ML, inteso nel senso molto generale sopra accennato, sono da considerarsi infinite e, come prevedibile, rappresentano ormai una consolidata realtà di ricerca teorica e applicata per tutte le scienze, da quelle naturali a quelle sociali.
Avere un'idea di cosa il ML sia e di come sia inteso dagli addetti ai lavori non è cosa semplice: un ottima introduzione, di taglio divulgativo e comprensibile anche a persone prive di competenze scientifiche e matematiche è l'ottimo libro di Pedro Domingos, pioniere del ML, scritto per i tipi della Basic Books nel 2015 (**Domingos, 2015**) e tradotto in italiano da Bollati Boringhieri nel 2016 (**Domingos, 2016**).

In concreto, di cosa parliamo? Se usiamo come guida introduttiva alla materia il libro di cui sopra, potremmo essere tentati di rispondere usando i diversi metodi e strumenti sviluppati dalle "tribù" di scienziati identificati da Domingos, raggruppate attorno ai loro strumenti cardine (Reti Bayesiane, Reti Neurali, Induzione, Evoluzione, Analogia). Di fatto è possibile identificare sia un excursus storico nella rassegna dei metodi (dalle prime reti neurali all'attuale deep learning (**Goodfellow, Bengio, & Courville, 2016**)) sia una mappa classificatoria che spesso rappresenta una varietà di

[1] Ma il termine non è attuale, come dimostra (**Samuel, 1959**)

espressioni che lascia piuttosto confusi. Dire che rientrano nel ML, ad esempio, il *riconoscimento di pattern* e le *reti neurali artificiali* è corretto ma mette insieme un compito specifico (trovare dentro una marea di dati grezzi una maniera per riconoscere e classificare un certo oggetto) con un sistema che può e deve svolgere anche quel compito (**Bishop, 1994**).

Non essendo possibile in breve spazio nemmeno presentare le questioni profonde legate a temi complessi come la IA, di cui il ML rappresenta un tema cruciale, possiamo tuttavia in forma molto sintetica e solo per fornire un'immagine visuale della materia, dire che per ML o apprendimento automatico dobbiamo intendere una collezione di metodi computazionali che mirano a fornire a una macchina la capacità di imparare dandogli un solo set di regole iniziali, lasciando poi che essa applichi le stesse allo scopo di agire autonomamente.

I diversi metodi considerati, di cui uno è presentato ed applicato qui, sebbene lontani per provenienza e ancora distanti nei loro percorsi evolutivi non vanno intesi come scollegati ma convergenti a un ideale finale, creare una macchina tanto capace quanto essenziale, dotata del minimo per raggiungere l'autonomia di pensare ed agire per suo conto. E' questo un obiettivo da tenere sempre presente ogni volta che si applica o si specula su questi metodi, sia per motivi professionali che per puro amore di conoscenza.[2]

ML e modelli sociali

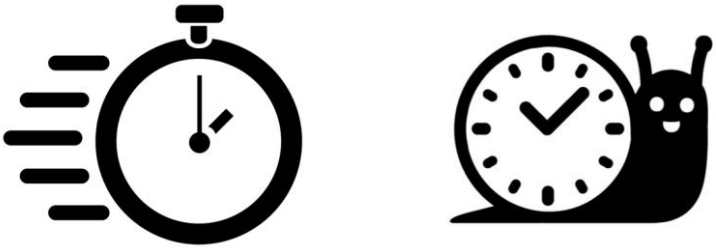
Le scienze sociali, ed in particolar modo la sociologia tra esse, hanno manifestato da tempo un forte interesse per i metodi computazionali, favorite in questo dalla semplicità e potenza degli strumenti operativi sotto forma di software ormai disponibili in forma gratuita e ricchi di supporti didattici ma, più ancora dalla forte attività e condivisione di idee e risultati di una comunità di studiosi e persone interessate in crescita.
Una delle aree di maggior interesse della ricerca sociale è quella basata sulla simulazione ad agenti[3]: in particolare, questa area di ricerca è stata considerata dagli autori nello sviluppo del modello sociale cui si fa cenno nel parlare dell'esperimento svolto in questa sede. E' opportuno fare cenno anche a due direzioni di ricerca prese in considerazione nel modello globale in costruzione: la fisica sociale e le modalità di interazione tra scelte individuali e influenza del contesto sociale, cui i due brevi paragrafi successivi accennano.

[2] "*The grand aim of science is to cover the greatest number of experimental facts by logical deduction from the smallest number of hypotheses or axioms.*" in (**Ratcliffe, 2011**) pag. 223, cit. 23 da (**Bartlett, 1949**) .
[3] Per una introduzione, si veda (**Squazzoni, 2008**).

La fisica sociale del MediaLab

Quello che oggi intendiamo per fisica sociale si pone l'ambizioso obiettivo di tracciare il flusso delle idee e l'interazione di tale flusso con la rete sociale, che ha il ruolo di tradurre le idee in comportamenti. Uno dei limiti della ricerca sociologica è, ad esempio, il suo fondarsi su interviste, essendo consapevoli che ciò che viene dichiarato, non necessariamente corrisponde ad intenti e comportamenti reali. Sappiamo però che gli esseri umani sono restii a farsi classificare da etichette e teorie economiche e storiche e che rispondono, fortemente, ad una serie di stimoli non solo relativi al proprio interesse personale, ma anche al rafforzare i legami con la propria rete sociale, a ricevere e dare gratificazione, anche immateriale, oltre ai consueti schemi di interazione razionale propri delle scienze sociali (**Pentland, 2015**).

Un cenno alla velocità dei pensieri.[4]



Il richiamo è naturalmente al lavoro di Kahneman (**Kahneman, 2011**) e Simon (**Simon, 1978**), ovvero al modello della mente umana da loro proposto in cui si distingue tra pensiero "veloce" (intuizioni e azioni automatiche, abitudini essenzialmente inconsce, prevalentemente basato sul principio di analogia tra esperienze personali ed osservate esternamente) e "lento" (pensiero consapevole e strutturato in regole, fondato sull'uso della ragione).
Il pensiero veloce è pratico, limitato da ciò che si osserva e immediato da attuare, duro a modificarsi senza una consistente mole di esempi contrari all'esperienza pratica in grado di modificare i nostri comportamenti consueti. Il pensiero lento (meno adatto, in prima battuta, alla sopravvivenza di un uomo primitivo nella savana), si è evoluto parallelamente al veloce prendendo tutto ciò che avrebbe potuto un giorno rivelarsi utile e sperimentando e giocando con esso, per aprire nuove strade alla nostra mente e all'azione. Più che parlare di una supremazia dell'uno sull'altro, è utile comprendere la loro interazione. Anche se non dobbiamo più fuggire da un predatore in agguato nella boscaglia,

[4] Le immagini sono tratte da http://niagarahomeheating.com/why-choose-us/fast-and-reliable-service/ e https://www.youtube.com/channel/UCoJarCLYeFzja6JOQ0tp07A

nondimeno le nostre esistenze continuano ad essere costellate di una miriade di situazioni nelle quali "siamo costretti a decidere velocemente", "senza pensare". O ancora, in molte occasioni il pensiero veloce, meglio ancora se lo chiamiamo automatico, gestisce il tran-tran quotidiano.
Come studiare il modo in cui pensiero lento e veloce si integrano e completano?
Ebbene, l'utilizzo dei cosiddetti "Big Data" e il ML, giungono a proposito. La possibilità di disporre di enormi quantità di dati e di macchine in grado di processarli, getta una luce del tutto nuova sulle potenzialità delle scienze sociali: un conto è tentare costose, limitate e relativamente affidabili indagini sulle preferenze e sui comportamenti degli individui, altra cosa è tenere traccia e manipolare ai limiti del possibile l'enorme mole di informazioni da essi stessi immessa nel sistema attraverso l'esercizio delle loro scelte sociali, economiche, di consumatori come di cittadini.
Sappiamo che istinto ed abitudini promanano dal pensiero veloce, mentre esplorazione e concentrazione mentali caratterizzano il pensiero lento, che si prende il lusso di giocare e sperimentare con quanto apprende per tracciare nuovi possibili percorsi al pensiero e all'azione che un giorno saranno incamerati come "veloci". Esaminare i comportamenti umani alla luce della distinzione tra due modalità di pensiero distinte, porta a considerare tali comportamenti come prodotto simultaneo sia del nostro libero arbitrio che del contesto sociale (**Pentland, 2015**)[5].

L'applicazione considerata

In questo lavoro si utilizza il risultato di un'attività di indagine condotta da uno degli autori nel contesto di un'attività di ricerca realizzata per l'Università di Roma Tre con la collaborazione scientifica dell'ENEA. Il tema è l'efficienza energetica, teatro nel quale è stato realizzato un lavoro di modellistica sociale ed economica congiunto. Risultato finale del lavoro, la creazione di un modello sociale teso a spiegare le azioni di alcuni attori sociali primari (Famiglie ed Imprese, nello schema adottato) in merito a decisioni di investimento in tecnologie mirate all'efficienza energetica (riqualificazione residenziale, utilizzo di impianti a minore consumo energetico e altro). Il modello sociale messo a punto interagirà con un modello macroeconomico standard, vale a dire la Matrice di Contabilità Sociale, nella versione realizzata per l'Italia nel 2010 dall'Università di Tor Vergata (**Scandizzo, 2009**), e messa a disposizione da ENEA. Lo strumento totale così assemblato, si pone l'obiettivo di poter sia modellare l'agire sociale sul tema, sia di valutare l'impatto macroeconomico (variazione del Prodotto Interno Lordo, del valore aggiunto settoriale e dell'occupazione) dell'agire medesimo.

[5] Pagine 221-225 dell'Appendice C.

L'esperimento di questo progetto di ricerca si pone dunque lo scopo di fornire un contributo alla ricerca sia in tema di modellistica sociale, sia riguardo all'interazione di tali modelli con altri di diversa natura, nella prospettiva del supporto scientifico al pubblico decisore.

Nel contesto del progetto sopra richiamato, sono stati effettuati diversi esperimenti, tra i quali uno caratteristico dell'indagine sociologica, ovvero l'effettuazione di una serie di interviste ad alcuni degli attori considerati come rilevanti nel modello: il lavoro qui proposto si pone l'obiettivo di valorizzare in particolare uno degli esperimenti condotti, relativo ai Tecnici[6] intervistati, utilizzando il dataset composto dalle loro risposte.
Si tratta di un dataset estremamente limitato, sono 60 le domande ottenute dalle interviste effettuate per le cinque province del Lazio nel quale è stata effettuata la rilevazione, nondimeno è stato possibile utilizzarlo per mostrare uno dei possibili modi di valorizzare tali dati. Più ancora, dato quanto finora detto, oltre al canale classico e certamente significativo dell'intervista diretta, è interessante riflettere su quanto tali sistemi di indagine possano essere integrati ed in parte sostituiti dall'utilizzo delle grandi moli di dati sempre più frequentemente e accuratamente raccolte da soggetti istituzionali ed imprenditoriali, in particolar modo su settori delle attività umane come la EE.
Il semplice esperimento condotto serve quindi solo a mostrare alcune delle applicazioni del machine learning utili ad estrarre valore dai dati. In particolare, in questa sede si mostrerà l'applicazione di una macchina a vettori di supporto, di seguito introdotta metodologicamente, alla classificazione dei soggetti intervistati sotto un determinato criterio, mostrando in particolare sia la significatività di alcuni dati (che da soli spiegano già efficacemente i fenomeni studiati) sia il contributo derivante dall'utilizzare tutti i dati disponibili.

Metodologia

Un consistente gruppo di studiosi fonda sul ragionamento analogico il proprio operato, sebbene con intenti e risultati spesso molto diversi tra loro. In queste note si introduce il nocciolo concettuale da cui muove la costruzione delle macchine prodotte con questo approccio, usando degli esempi dalla vita quotidiana per illustrare i concetti essenziali. Questo excursus ricalca in forma sintetica la trattazione del testo divulgativo di riferimento[7]. Il ragionamento analogico è un cardine del pensiero umano, come dimostra la lunga tradizione esistente nella storia dalla filosofia e come chiunque può

[6] Si considerano come Tecnici le figure professionali qualificate che tipicamente operano nel settore della EE (Architetti, Ingegneri, Geometri, Energy Manager e altri).
[7] Pagine 177-203 dell'edizione inglese.

facilmente constatare in base alla propria esperienza. Le nostre decisioni sono costantemente fondate *anche* su criteri di somiglianza della situazione in cui ci troviamo rispetto ad altre situazioni note sotto aspetti che riteniamo essenziali. Un illuminante esempio storico (riportato da Domingos) di ragionamento analogico lo troviamo in Kennedy nella crisi di Cuba: non avendo a disposizione una teoria integrale delle relazioni internazionali per decidere il da farsi statunitense riguardo alla minaccia dei missili sovietici, vide un'analogia tra la crisi cubana e lo scoppio della prima guerra mondiale e prese la decisione corretta.

L'algoritmo Nearest Neighbor

Iniziamo dall'algoritmo più semplice, il Nearest Neighbor (NN) (**Fix, 1951**), che Domingos etichetta come il più semplice (e "pigro") algoritmo di apprendimento mai inventato: "non fa nulla", dice lui. A parte le facezie, esiste ovviamente un ottimo motivo per cui Domingos scherza sul NN e attiene alla sua straordinaria semplicità, che tuttavia va di pari passo con la potenza che è in grado di dimostrare in una serie di applicazioni. Se ad esempio vogliamo imparare a riconoscere dei volti, tutto quello che dobbiamo fare è caricare immagini in un database e darle in pasto all'algoritmo (un gioco da ragazzi, per i social network). Il prezzo da pagare per questa semplicità, è la "pigrizia" dell'algoritmo, perché per riconoscere se la faccia di Monica è un volto, potremmo essere costretti a esaminare quella di Laura e di qualche altra miliardata di persone simultaneamente in una frazione di secondo. All'inizio il nearest neighbor farebbe la figura che fa il tipico studente impreparato: ma la procrastinazione inerente al suo modo di operare, alla lunga paga e in questo contesto la pigrizia non assume alcun connotato negativo.

Per capire con uno degli esempi più semplici possibili come funziona questo algoritmo, immaginiamo di dover tracciare il confine tra due nazioni senza sapere null'altro che la posizione delle sue capitali: dove molti dei learner esistenti non saprebbero che pesci pigliare, l'algoritmo NN agisce risolutamente nel modo più semplice.

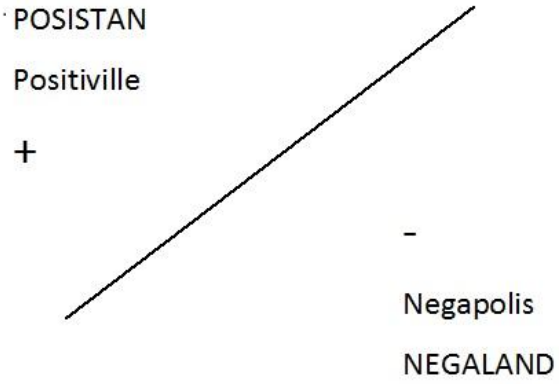


Figura 1 - Algoritmo Nearest Neighbor in azione nel confine tra due stati

Il NN traccia una bella linea retta equidistante tra le due città. Fine della storia.
Bella forza, chiunque poteva farlo, verrebbe da dire (e certo dovremmo avere una gran fortuna ad averci azzeccato). Non si riesce a comprendere bene la potenza di questo semplice approccio, fino a quando di città sul bordo del confine iniziano ad essercene molte, e ci si accorge che il NN ha la capacità di disegnare anche confini straordinariamente contorti pur non facendo altro che le due seguenti semplici cose: 1) ricordarsi della posizione delle città 2) assegnare i punti alle due nazioni di conseguenza. Punto.

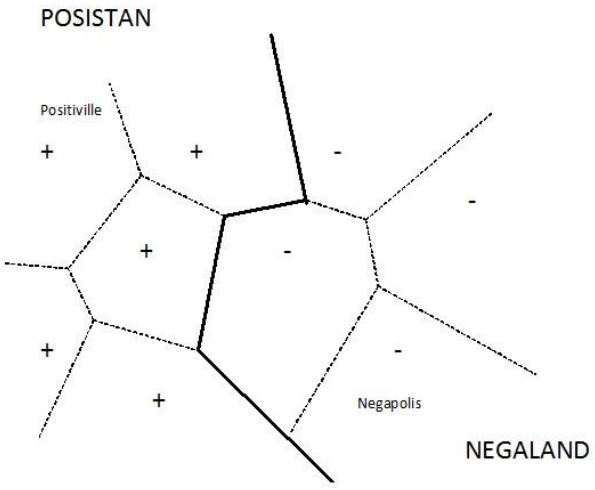


Figura 2 - Algoritmo Nearest Neighbor in azione nel confine tra due stati considerando i dati di più città

Se definiamo l'"area metropolitana di una città" come l'insieme dei punti più vicini a quel centro abitato che a tutti gli altri, riusciamo a tracciare le linee tratteggiate di figura e Posistan e Negaland emergono come la risultante dell'unione di tutte le loro aree metropolitane. In questo caso il ML "pigro" di un NN ha la meglio su un albero di decisione, in quanto è estremamente più complesso costruire un modello di quel tipo rispetto al limitarsi alla semplice stima della posizione dei punti cercati. Naturalmente per un problema elementare come quello delle figure bastano database di dimensioni ridotti, laddove per un compito come il riconoscimento facciale ne servono di enormi: oggi tali database esistono e l'unico costo per il NN in questo senso è il tempo di ricerca all'interno dei medesimi.

Nell'algoritmo NN, ogni dato è il classificatore di se stesso, in quanto prevede la classe di tutte le ricerche nelle quali compare tra i risultati. Domingos fa il paragone con un esercito di formiche: i singoli soldati fanno poco da soli ma sono capaci di tutto, quando fanno massa critica. Una variante del NN e il kNN (k-Nearest Neighbor) in cui ogni esempio di test è classificato facendo votare i suoi k vicini più prossimi. Se l'immagine che ho caricato nel database è quella di un volto ma le due più prossime ad essa non lo sono, il 3-Nearest Neighbor classificherà tale immagine come *non* rappresentante un volto.

In pratica il kNN è più robusto del NN in quanto si sbaglia solo quando i k vicini prossimi del test sono errati, laddove l'algoritmo NN è a rischio di overfitting[8] (se abbiamo classificato male un dato, ce lo portiamo dietro per tutta la sua "area metropolitana", rimanendo nell'esempio di figura 2).

Il rovescio della medaglia è ovviamente una classificazione meno precisa: il voto a maggioranza, nell'esempio di figura 2, produrrebbe dei confini meno netti. Inoltre se cresce k, cresce il bias. Una variante è l'algoritmo kNN pesato messo a punto dall'Università del Minnesota e dal MIT (1994). Il concetto di base è il cosiddetto filtraggio collaborativo. Immaginiamo di essere alla ricerca di un sistema per classificare gli utenti di un catalogo di film come Netflix: l'ipotesi che facciamo è che le persone che si trovano d'accordo su un certo argomento in passato hanno buone probabilità di trovarsi d'accordo anche in futuro (la pietra angolare su cui oggi si appoggiano tutti i siti di e-commerce). Se immaginiamo che ogni utente di Netflix possa dare un voto ai film in catalogo, allora se intendiamo indovinare i voti che darà Monica ad essi, potrebbe essere una buona idea osservare chi in passato a dato i voti più simili ai suoi: se in passato tutti gli utenti hanno dato una buona valutazione su *The Revenant*, potete raccomandare questo film anche a Monica. Se invece il loro parere sul film è controverso, allora potrete compiere un ulteriore passo e classificare gli utenti in base al loro grado di correlazione con Monica. Quindi, se la correlazione di Franco con Monica fosse più elevata di quella con Laura, i suoi voti dovrebbero contare di più. Alla fine la previsione

[8] Vedi appendice.

sul voto di Monica è la media pesata di quelli dei suoi vicini, in cui il peso degli stessi è espresso dal suo coefficiente di correlazione con Monica.
Una possibile complicazione in questo ottimistico quadro insorge quando modelliamo più finemente i protagonisti dell'analisi. Immaginiamo che Monica e Laura abbiano gusti simili ma che Monica abbia delle divergenze con Laura ed ogni volta che quest'ultima da cinque stelle ad un film Monica ne dia tre, e viceversa. Se usiamo i voti di Laura per prevedere quelli di Monica, ci troveremo sempre con uno scarto di due stelle. Per ovviare a questo problema, possiamo incorporare la differenza di previsione tra i voti di Monica e la sua media a partire dalla differenza di quelli di Laura: dato che Monica è sempre due stelle sopra la sua media, e via di seguito, le previsioni torneranno ad essere azzeccate. Nei sistemi basati sul filtraggio collaborativo, in ogni caso, non abbiamo bisogno di conoscere esplicitamente che voti danno gli utenti. Se Monica ha ordinato un film di Netflix, significa che si aspetta di apprezzarlo: possiamo quindi modellare il "voto" con "ordinato" o "non ordinato" e due utenti verranno classificati come simili se avranno ordinato un certo numero di volte lo stesso film. E' interessante sapere che i sistemi di raccomandazione sono responsabili di una bella fetta del fatturato di aziende come Netflix e Amazon[9].

Uno dei principali pregi del NN è che al tendere della quantità di dati all'infinito l'algoritmo non supera mai di due volte il **Bayes error rate** (il minimo errore dovuto alla distribuzione dei dati). Per alcuni valori di *k*, con *k* che cresce in funzione della mole di dati, l'algoritmo raggiunge il Bayes error rate.

La maledizione della dimensionalità di Bellman

Quando le dimensioni sono poche, nell'ordine delle unità, il NN funziona a dovere. Se ci spostiamo di ordini di grandezza, migliaia o milioni di attributi come capita di dover fare, le cose si complicano. Su Amazon, ogni volta che un cliente fa clic, crea un attributo, tanto per capire la situazione. Il primo punto è che occorre scremare gli attributi significativi da quelli che non lo sono: se intendiamo sapere dove andrà in vacanza Monica quest'estate, è abbastanza improbabile che ci possa aiutare sapere che ha comperato un set di brugole (a meno che non intenda passare l'estate a rifare casa). Il secondo problema, più sorprendente, è che anche avere un enorme numero di attributi pertinenti può generare problemi. E' chiaro che più informazioni si hanno più aumenta la conoscenza, ma gli esempi di addestramento necessari a individuare le frontiere di un concetto

[9] Secondo (**Domingos, 2016**) si parla di un terzo del fatturato per Amazon e di tre quarti per Netflix (pagina 218).

crescono in modo esponenziale: 20 attributi booleani danno oltre 1 milione di esempi possibili, 21, oltre 2 milioni e la frontiera disporrà in tal caso di altrettante modalità di snodamento. Inoltre, in un contesto a molte dimensioni, la stessa nozione di somiglianza è a rischio: se abbiamo un'arancia fatta per il 90% del suo raggio da polpa e per il restante 10% dalla buccia, allora il 73% del suo volume, ($0,9^3$) è fatto da polpa. Ma prendiamo un iperarancia a 100 dimensioni: con le stesse proporzioni di prima, la polpa diventa 30 milionesimi del volume ($0,9^{100}$). Tutta buccia! Non esistono learner immuni alla *maledizione della dimensionalità*, espressione coniata dall'esperto di teoria dei controlli Richard Bellman negli anni cinquanta (**Bellman, 1957**). Bellman constatò personalmente che algoritmi di controllo che funzionavano a meraviglia in 3 dimensioni arrancavano non appena si superava tale limite, cosa semplice a verificarsi se il compito da eseguire è, ad esempio, il controllo di tutte le articolazioni di un braccio robotico.

Come uscirne? La prima azione è sbarazzarsi delle dimensioni superflue: il NN può riuscire in questo compito scartando tutti gli attributi il cui guadagno di informazione è sotto a una certa soglia, indi misurando la somiglianza solo nel ridotto spazio risultante. Ovviamente ci sono costi da sopportare, ad esempio non si è più in grado di apprendere concetti come l'XOR (se un attributo da informazioni per una classe solo quando è usato in combinazione con altri, verrà scartato). Un opzione "costosa" ma più efficiente è avviluppare la selezione degli attributi sullo stesso learner, cancellando tutti quelli che non contribuiscono all'accuratezza del NN sui dati di validazione. *Last but not least*, spesso la non uniformità dei dati utili al problema si trova in una piccola regione dell'iperspazio e, per quanti attributi si possano avere, essi potrebbero "vivere" in uno spazio con un numero di dimensioni assai più piccolo. NN ha dato vita nel tempo a numerose varianti, per cui si rimanda ai riferimenti di letteratura[10].

Le macchine a vettori di supporto (SVM, Support Vector Machine)

Le SVM sono frutto della migrazione del sovietico Vladimir Vapnik dall'Istituto di Scienze dei Controlli di Mosca ai Bell Labs negli anni 90. In apparenza, una SVM assomiglia molto a un kNN pesato: le frontiere tra classi positive e negative sono definite da un insieme di esempi e dai loro pesi, più una misura di somiglianza. Un dato esempio viene classificato con una certa classe se, in media, assomiglia di più agli esempi di quella classe che agli altri. La media è pesata e la SVM

[10] Ricordiamo: il k-Most Similar Neighbour (k-MSN) **(Hernàndez-Rodrìguez, Martìnez-Trinidad, & Carrasco-Ochoa, 2010**), il Linear scan (**Cui, Huang, Wang, & Liu, 2013**), il Kd-trees (**Goswami, Erol, Mukhi, Pajarola, & Gobbetti, 2013**), il Balltrees (**Nielsen, Piro, & Barlaud, 2009**), il Metric trees (**Pestov, 2013**), il Locality-sensitive Hashing (LSH) (**Datar, Immorlica, Indyk, Mirrokni, & Vahab.S, 2004**), l'Agglomerative-Nearest-Neighbour **(Beaugranda, Ibañezb, & Lindleya, 2003**) e il Redundant Bit Vectors (RBV) (**Goldstein, Plat, & Burges, 2005**).

ricorda solo gli esempi chiave necessari a definire con esattezza la frontiera. Nell'esempio del Posistan e Negaland, questo sarebbe il risultato ottenuto eliminando tutte le città che non sono sulla frontiera:

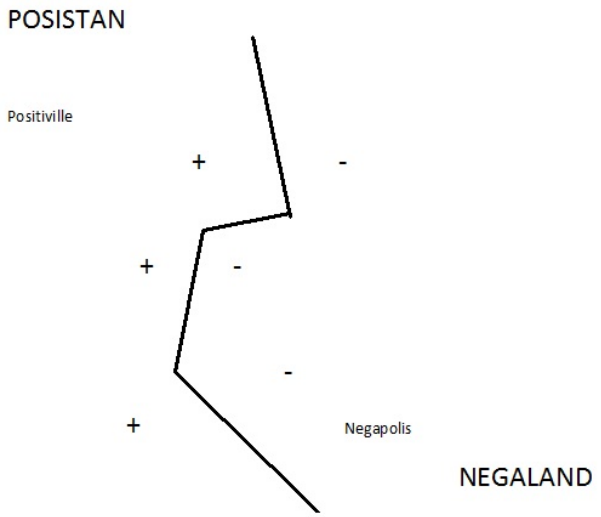


Figura 3 - Algoritmo SVM in azione nel confine tra due stati considerando i dati di più città

Gli esempi rimasti sono quello che chiamiamo "vettori di supporto", poiché sono essi a "sorreggere" la frontiera: toglietene anche solo uno e la spezzata del confine si sposterà altrove rispetto a dove è ora. I concetti reali non corrispondono molto ad una spezzata, tendono ad avere confini più regolari, il che indica che forse l'approssimazione del vicino più prossimo non è quella ideale. Con le SVM questo non rappresenta un problema e si possono ottenere risultati anche di questo tipo:

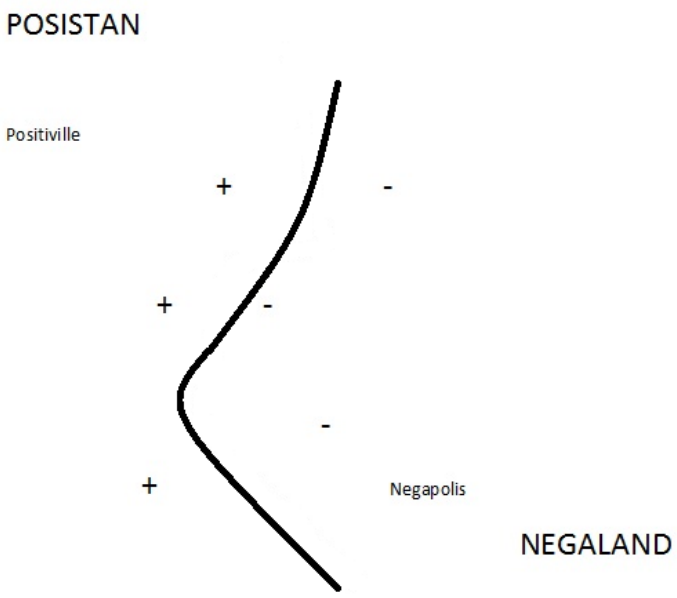


Figura 4 - Algoritmo SVM in azione nel confine tra due stati considerando i dati di più città con confini regolari

L'apprendimento per una SVM presuppone la scelta dei vettori di supporto e dei loro pesi. La misura di somiglianza è invece ciò è identificato come kernel (nucleo) ed è solitamente scelta a priori. Uno dei contributi fondamentali di Vapnik è stato comprendere che non tutte le frontiere sono uguali. Per capire bene questo punto possiamo ricorrere ad una metafora militare. Immaginiamo che Posistan e Negaland siano in guerra e che siano divisi da una zona neutra cosparsa di mine da un lato e dall'altro: la nostra SVM è come un soldato che deve muoversi in tale zona senza saltare per aria. Se sostituiamo al posto delle mine gli esempi utilizzati, il movimento del soldato allora corrisponde alla frontiera appresa e il margine di sicurezza è la distanza minima dagli esempi: compito di una SVM è quello di tracciare la frontiera migliore col minimo margine di sicurezza necessario.

A questo punto si tratta di trovare il margine più ampio con cui passare tra le "mine", tra gli esempi positivi e negativi, in modo da avere una buona capacità di classificazione e non cadere nel rischio dell'overfitting (un confine troppo contorto, che contiene troppa informazione e alla fine non dice niente di utile)[11]. Naturalmente occorre trovare i pesi che rendono il margine massimo e tutti gli esempi che dimostrano un peso nullo possono essere scartati. Così facendo però si potrebbero far aumentare i pesi in modo illimitato: quindi è necessario massimizzare il margine imponendo come

[11] Un modello assurdo e sbagliato può adattarsi perfettamente ai dati se è abbastanza complesso rispetto alla quantità di dati disponibili. Spesso si sostiene che l'*overfitting* sia una violazione del principio del Rasoio di Occam https://it.wikipedia.org/wiki/Overfitting.

vincolo che i pesi non superino un certo valore prefissato (oppure si può agire in modo equivalente imponendo che tutti gli esempi abbiano un certo margine arbitrario, ad esempio 1, di solito nella pratica le SVM lavorano così).

Il problema dell'ottimizzazione vincolata che ci troviamo a contemplare può essere reso, a livello metaforico immaginando di scalare una montagna dovendo rimanere su un certo sentiero tracciato, per cui il risultato finale sarà la massima altezza raggiungibile percorrendo la strada prescelta: se la strada arriva fino in cima, il problema vincolato e quello non vincolato hanno la stessa soluzione. I dettagli tecnici in merito sono questioni specialistiche, per chiudere qui il discorso si può ricordare che nella pratica alle SVM è consentita una certa tolleranza, gli viene consentito a volte di violare qualche vincolo, classificando erroneamente qualche esempio o rimanendo al di sotto del margine di sicurezza per evitare l'overfitting.

Il caso studio - Interviste a tecnici e formatori nella Regione Lazio sul tema dell'efficienza energetica e della formazione

Ora che abbiamo fatto conoscenza con la teoria, possiamo illustrare efficacemente come funziona una SVM, applicandola ad un problema molto semplice. Possiamo quindi fare ricorso al package **e1071** (**CRAN, 2017**) di R, contenente una implementazione della SVM e un semplice codice utilizzante il dataset delle nostre interviste per effettuare il nostro esperimento.

La sfida scelta per il nostro learner è classificare correttamente i tecnici del database in base ad un criterio, in questo caso individuare i tecnici favorevoli ed attivi all'implementazione delle policy di EE in base al valore assunto da diverse variabili, come ad esempio il livello di formazione ed informazione degli stessi.

Prima di iniziare, diamo prima un'occhiata al dataset **Tecnici**. Si tratta di 58 record, contenenti 44 attributi:

1. Provincia
2. Tipologia Professionale
3. Fiducia su previsioni ENEA 2015 sul settore
4. Fiducia nelle potenzialità reddituali dell'Efficienza Energetica
5. Partnership con imprese che fanno Efficienza Energetica
   Tecnologie utilizzate o utilizzabili:
6. Pompe di Calore
7. Caldaie a condensazione
8. Cogenerazione/trigenerazione
9. Aria Compressa
10. Cucine a induzione
11. Solar cooling
12. Fotovoltaico
13. Solare Termico
14. Controllo Solare
15. Building automation
16. Caldaie a biomasse

17. Tecnologie per illuminazione
18. Motori elettrici ad alta efficienza
19. Inverter
20. Isolamento copertura all'estradosso con isolante sottotegola
21. Sostituzione tetto con copertura isolata e ventilata
22. Isolamento all'estradosso a cappotto
23. Isolamento all'esterno con parete ventilata
24. Isolamento in cassa vuota
25. Isolamento all'interno con contro-parete isolata
26. Sostituzione del serramento
27. Sostituzione del vetro su telaio esistente
28. Installazione sistemi schermatura solare esterni
29. Serramenti con vetrocamera taglio termico
30. Serramenti con vetrocamera a giunto aperto
31. Serramenti a taglio termico
32. Serramenti nuova generazione
33. Opinione su livello sostenibilità economica delle precedenti tecnologie
34. Opinione su incentivi statali per E.E
35. Partecipazione a corsi di Formazione sulla E.E.
36. Partecipazione a Campagne di informazione sulla E.E.
37. Atteggiamento verso le Politiche di EE
38. Importanza Barriere Culturali
39. Importanza Barriere Economiche
40. Importanza Barriere Normative
41. Importanza Barriere Tecnologiche
42. Presenza di rapporti con la P.A.
43. Partecipazione a bandi pubblici
44. Presenza di Energy manager

La prima cosa che possiamo osservare è che si tratta di un numero notevole di variabili (ne sono inoltre state omesse diverse, per semplificare la descrizione seguente), che permette di effettuare una notevole serie di domande potenzialmente interessanti. Per semplificare le cose, naturalmente, abbiamo scelto di concentrarci soltanto su tre delle 44 variabili, ritenute di particolare interesse, ovvero la variabile 1, la 33 e la 34. In pratica vogliamo sapere se la partecipazione ad attività di

Formazione e l'esposizione a Campagne di Formazione abbiano effetto sul grado di fiducia che i tecnici ripongono nelle potenzialità reddituali insiste nelle politiche di Efficienza Energetica.
Detto in parole povere, potremo esprimere quanto sopra riformulando la domanda come: "La Formazione e l'Informazione sono efficaci nel convincere i Tecnici che fare Efficienza Energetica conviene?". Dal momento che coloro che abbiamo considerato Tecnici sono soggetti a medio - alta qualificazione incaricati di realizzare concretamente le decisioni di investimento dei soggetti sociali, il loro grado di convinzione è una variabile rilevante dello studio dell'intero modello sociale.

Analizzando le risposte ottenute in cerca di un primo risultato statistico che ci mostri la correlazione tra il grado di fiducia mostrato otteniamo quanto segue:

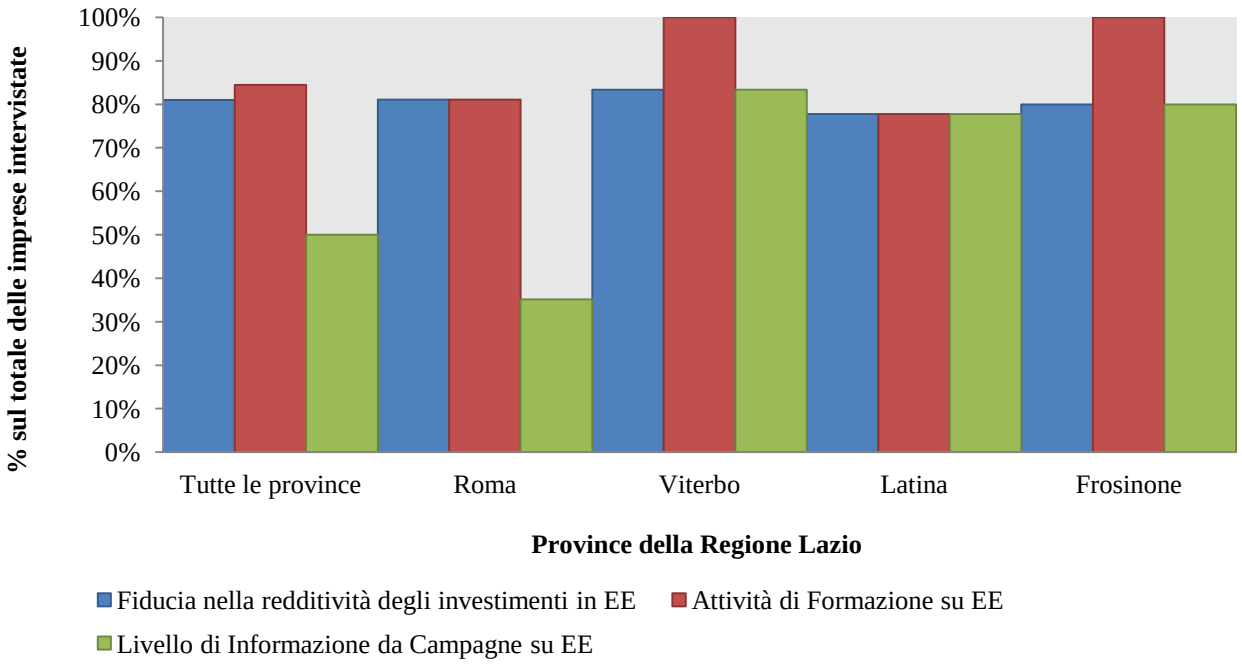


Figura 5 – Quota di imprese intervistate per provincia riguardo a tre parametri: fiducia nelle politiche di EE, attività di formazione e partecipazione a campagne di informazione

La figura 5 ci dice che, in modo piuttosto disomogeneo tra Roma e le altre province, la fiducia nella redditività degli investimenti segue molto di più la Formazione che l'Informazione (un processo attivo di acquisizione conoscenze contro uno passivo di assimilazione di dati e nozioni). Proviamo dunque a verificare che succede mettendo al lavoro una SVM e verificandone i risultati. Nel package **e1071** un apposito codice (**svm**), permette di costruire un modello appropriato e di calcolare e graficare i risultati di test dello stesso. Facciamo un esperimento duplice, modellando sia usando tutti i dati, che usando il solo dato della Formazione.

Costruito il modello, andiamo ora a vedere quanto il codice ha azzeccato le previsioni verificando in una tabella di corrispondenza per tipologie, quante occorrenze sono state classificate nei due tipi possibili:

	Fiducioso	Non Fiducioso
Fiducioso	47	11
Non fiducioso	0	0

Tabella 1 - Risultati classificazione della SVM sul dataset Tecnici utilizzando il solo dato della Formazione

	Fiducioso	Non Fiducioso
Fiducioso	47	0
Non fiducioso	0	11

Tabella 2 - Risultati classificazione della SVM sul dataset Tecnici utilizzando tutti i dati

Nel primo caso, l'algoritmo ha correttamente classificato i risultati nel 81% dei casi, una percentuale più che accettabile: nel secondo, con tutti i dati a disposizione, nel 100%.

Questo semplice esempio, è sufficiente a mostrare il principio per cui, una maggiore quantità di dati, in particolare se in qualche modo pertinente, aiuta a migliorare il livello di successo nel riconoscimento. E' importante sottolineare che spesso non è a noi affidato il compito di stabilire cosa sia pertinente (usiamo i learner anche e molto per vedere ciò che non ci balza facilmente sotto gli occhi, o ciò che sarebbe per un essere umano ostico o impossibile da trovare).

Possiamo visualizzare il lavoro fatto dalla SVM anche graficamente, con poche linee di codice R dedicate al classic multidimensional scaling[12] (la funzione di R usata è **cmdscale**). Rimandando ai riferimenti appropriati per approfondire il tema dello scaling, di fatto il grafico evidenzia le differenze che i dati esaminati presentano rispetto agli attributi per i quali sono stati classificati (i

[12] Lo **scaling multidimensionale** (MDS, dall'inglese *MultiDimensional Scaling*) è una tecnica di analisi statistica usata spesso per mostrare graficamente le differenze o somiglianze tra elementi di un insieme. È una generalizzazione del concetto di ordinamento: partendo da una matrice quadrata, contenente la "somiglianza" di ogni elemento di riga con ogni elemento di colonna, l'algoritmo di scaling multidimensionale assegna a ogni elemento una posizione in uno spazio N-dimensionale, con N stabilito a priori. Se N è sufficientemente piccolo, questo spazio può essere rappresentato con un grafico o una visualizzazione 3D. In pratica questa tecnica parte con un sistema con tante dimensioni quanti gli elementi del sistema, e riduce le dimensioni fino a un certo numero N. Nel fare questo quindi c'è un'inevitabile perdita di informazione (*loss*) ed esistono quindi diversi algoritmi per fare scaling multidimensionale, che si adattano meglio alle diverse situazioni di utilizzo: in particolare si distinguono algoritmi *metrici* e *non-metrici*.

diversi colori mostrano che le varie tipologie sono, naturalmente, abbastanza omogenee tra loro) e la posizione dei vettori di supporto.

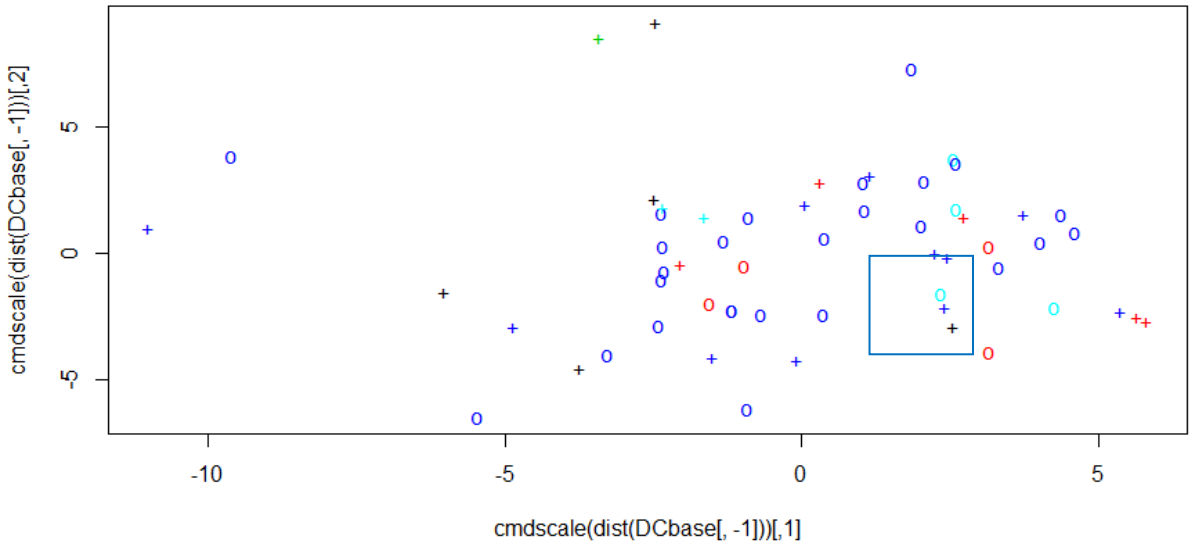


Figura 6 – Rappresentazione dei dati e dei vettori di supporto del dataset Tecnici per la variabile "Fiducia_EE"

Il grafico evidenzia le classi per colore ed i vettori di supporto usati per mezzo di croci. Le imprese fiduciose sono 53 (blu scuro), 5 le non fiduciose (azzurro). Si evidenzia la distribuzione dei dati a fronte di quella dei vettori di supporto che fungono da "cani pastore" per i dati "sparsi" delimitandone in modo efficace i confini. E' possibile notare che sia per il primo che per il secondo gruppo di imprese esistono vettori di supporto allocati in aree "confuse", nelle quali si è abbastanza equidistanti tra i due gruppi (ciò spiega gli errori di previsione riscontrati per i gruppi in questione (si noti il quadrato blu di figura con vettori di supporto delle due classi a stretto contatto).

Codice

Di seguito l'esempio didattico disponibile nelle librerie di R quando si installa il package e1071. Il codice è in blu, l'eseguito di R è in nero e le figure mostrano i grafici dei risultati.

```
library(rJava)
library(xlsx)
library(e1071)

DCbase <- read.xlsx("H:/Base/Francesca//Dati/Dati Tecnici 2.xlsx",1)
data(DCbase)
names(DCbase)
#DCbase=DCbase[ ,c()] #per eliminare eventuali colonne superflue
#DCbase<-na.omit(DCbase)

attach(DCbase)
names(DCbase)

#Fiducia EE

model <- svm(formula = factor(Fiducia_EE) ~ ., data = DCbase, cost = 1, gamma = 1, type = "C-classification", tolerance = 1)
#model <- svm(giorni_inizio ~ ., data = DCbase)

x <- subset(DCbase, select = -Fiducia_EE)
y <- Fiducia_EE

print(model)
summary(model)

#pred <- fitted(model)
pred<-predict(model)

table(pred, y)
```

```
# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(DCbase[,-1])),
     col = as.integer(DCbase[,1]),
     pch = c("o","+")[1:150 %in% model$index + 1])

#Formato

model <- svm(formula = factor(Formato) ~ ., data = DCbase, cost = 1, gamma = 1, type = "C-classification", tolerance = 1)
#model <- svm(giorni_inizio ~ ., data = DCbase)

x <- subset(DCbase, select = -Formato)
y <- Formato

print(model)
summary(model)

#pred <- fitted(model)
pred<-predict(model)

table(pred, y)

# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(DCbase[,-1])),
     col = as.integer(DCbase[,1]),
     pch = c("o","+")[1:150 %in% model$index + 1])

# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(DCbase[,-1])),
     col = as.integer(DCbase[,1]),
     pch = c("o","+")[1:150 %in% model$index + 1])

#Informato
```

```
model <- svm(formula = factor(Informato) ~ ., data = DCbase, cost = 1, gamma = 1, type = "C-classification", tolerance = 1)
#model <- svm(giorni_inizio ~ ., data = DCbase)

x <- subset(DCbase, select = -Informato)
y <- Informato

print(model)
summary(model)

#pred <- fitted(model)
pred<-predict(model)

table(pred, y)

# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(DCbase[,-1])),
     col = as.integer(DCbase[,1]),
     pch = c("o","+")[1:150 %in% model$index + 1])



#Opinione_incentivi

model <- svm(formula = factor(Opinione_incentivi) ~ ., data = DCbase, cost = 1, gamma = 1, type = "C-classification", tolerance = 1)
#model <- svm(giorni_inizio ~ ., data = DCbase)

x <- subset(DCbase, select = -Opinione_incentivi)
y <- Opinione_incentivi

print(model)
summary(model)
```

```
#pred <- fitted(model)
pred<-predict(model)

table(pred, y)

# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(DCbase[,-1])),
    col = as.integer(DCbase[,1]),
    pch = c("o","+")[1:150 %in% model$index + 1])

#Atteggiamento_verso_Pol_EE

model <- svm(formula = factor(Atteggiamento_verso_Pol_EE) ~ ., data = DCbase, cost = 1, gamma
= 1, type = "C-classification", tolerance = 1)
#model <- svm(giorni_inizio ~ ., data = DCbase)

x <- subset(DCbase, select = -Atteggiamento_verso_Pol_EE)
y <- Atteggiamento_verso_Pol_EE

print(model)
summary(model)

#pred <- fitted(model)
pred<-predict(model)

table(pred, y)

# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(DCbase[,-1])),
    col = as.integer(DCbase[,1]),
    pch = c("o","+")[1:150 %in% model$index + 1])

#Energy_manager
```

```
model <- svm(formula = factor(Energy_manager) ~ ., data = DCbase, cost = 1, gamma = 1, type = "C-classification", tolerance = 1)
#model <- svm(giorni_inizio ~ ., data = DCbase)

x <- subset(DCbase, select = -Energy_manager)
y <- Energy_manager

print(model)
summary(model)

#pred <- fitted(model)
pred<-predict(model)

table(pred, y)

# visualize (classes by color, SV by crosses):
plot(cmdscale(dist(DCbase[,-5])),
        col = as.integer(DCbase[,5]),
        pch = c("o","+")[1:150 %in% model$index + 1])
```

Bibliografia

1. Bartlett, L. (1949). *The Universe and Dr. Einstein.* Victor Gollancz.
2. Beaugranda, G., Ibañezb, B., & Lindleya, J. (2003). An overview of statistical methods applied to CPR data. *Progress in Oceanography* , 253-262.
3. Bellman, R. (1957). *Dynamic programming.* Princeton University Press.
4. Bishop, C. M. (1994). *Neural Networks for Pattern Recognition.* Oxford: Oxford University Press.
5. CRAN. (2017, febbraio 09). *e1071*. Tratto il giorno febbraio 09, 2017 da cran.r.project.org: https://cran.r-project.org/web/packages/e1071
6. Cui, J., Huang, Z., Wang, B., & Liu, Y. (2013). Near-Optimal Partial Linear Scan for Nearest Neighbor Search in High-Dimensional Space. In *Database Systems for Advanced Applications* (p. 101-115). Berlin: Springer.
7. Datar, M., Immorlica, N., Indyk, P., Mirrokni, & Vahab.S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. *Proceedings of the twentieth annual symposium on Computational geometry* (p. 253-262). New York: ACM.
8. Di Ghiles - Opera propria, C. B.-S. (s.d.).
9. Domingos, P. (2016). In P. Domingos. Torino: Bollati Boringhieri.
10. Domingos, P. (2015). *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World* . Basic Books.
11. Fix, E. H. (1951). *Discriminatory analysis, nonparametric discrimination: Consistency properties.* Randolph Field, Texas: USAF School of Aviation Medicine.
12. G Beaugranda, ,. ,. (s.d.). An overview of statistical methods applied to CPR data.
13. Ghiles. (s.d.). Tratto da https://commons.wikimedia.org/w/index.php?curid=47471056
14. Ghiles. (2017). Tratto da https://commons.wikimedia.org/w/index.php?curid=47471056
15. Goldstein, J., Plat, J. C., & Burges, C. J. (2005). Redundant Bit Vectors for Quickly Searching High-Dimensional Regions. In J. Winkler, M. Niranjan, & N. Lawrence, *Deterministic and Statistical Methods in Machine Learning* (p. 137-158). Berlin: Springer.
16. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning (Adaptive Computation and Machine Learning series).* Cambridge: MIT Press.
17. Goswami, P., Erol, F., Mukhi, R., Pajarola, R., & Gobbetti, E. (2013). An efficient multi-resolution framework for high quality interactive rendering of massive point clouds using multi-way kd-trees. *The Visual Computer* , 69-83.
18. Hernàndez-Rodrìguez, S., Martìnez-Trinidad, J., & Carrasco-Ochoa, J. (2010). Fast k most similar neighbor classifier for mixed data (tree k-MSN). *Pattern recognition* , 873-886.
19. Kahneman, D. (2011). *Thinking, Fast and Slow.* New York: Farrar, Straus and Giroux.
20. Nielsen, F., Piro, P., & Barlaud, M. (2009). Tailored Bregman Ball Trees for Effective Nearest Neighbors. *Proceedings of the 25th European Workshop on Computational Geometry (EuroCG)* (p. 29-32). Brussels: HAL.
21. Pentland, A. (2015). *Fisica sociale - come si propagano le buone idee.* Milano: Università Bocconi Editore.
22. Pestov, V. (2013). Lower Bounds on Performance of Metric Tree Indexing Schemes for Exact Similarity Search in High Dimensions. *Algorithmica* , 310-328.
23. Ratcliffe, S. (2011). *Oxford Treasury of Sayng and Quotations.* New York: Oxford University Press.
24. Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development* , 535-554.
25. Scandizzo, P. B. (2009). *La matrice di contabilità sociale (SAM): uno strumento per la valutazione, IPI 2009.* Roma: IPI.

26. Simon, H. (1978). Rational Decision Making in Business Organizations. In A. Lindbeck, *Nobel Lectures, Economics 1969-1980.* Singapore: Assar Lindbeck, World Scientific Publishing Co.
27. Squazzoni, F. (2008). *Simulazione sociale - Modelli ad agenti nell'analisi sociologica.* Pisa: Carocci.
28. Thiele, J. C. (2014). R Marries NetLogo: Introduction to the RNetLogo Package. *Journal of Statistical Software* .

Appendice

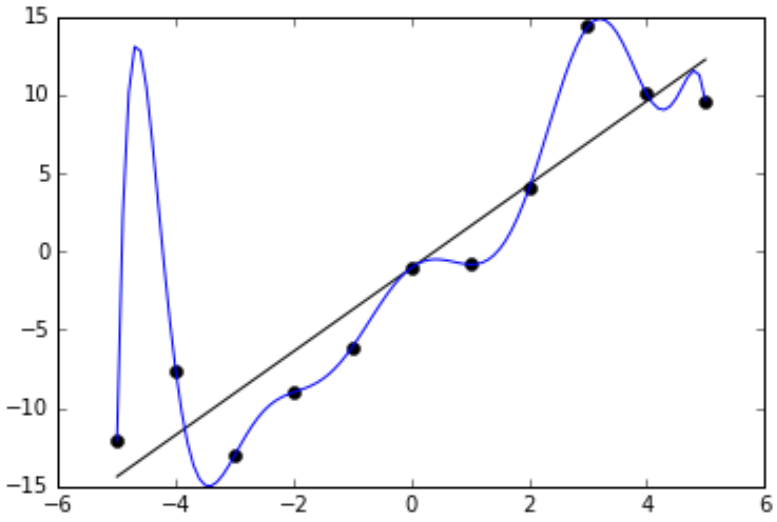
Overfitting

L'overfitting è l'eccessivo adattamento di un modello statistico, generalmente complesso, ai dati osservati. Il "troppo" adattamento è sospetto in quanto è il numero di parametri, in luogo dell'appropriatezza logica a livello concettuale, a dare risultati apparentemente entusiasmanti. Nel ML, normalmente a livello operativo si partizionano i dati disponibili in dati di train, di allenamento, e di test, di verifica. Quello che intendiamo ottenere come risultato è "infondere" intelligenza al nostro algoritmo di apprendimento, al learner: questo significa che ci attendiamo che, da un certo momento dell'allenamento in poi, esso sia in grado di astrarre, generalizzare e quindi decidere. In caso di overfitting, otterremo prestazioni eccellenti solo sui dati supervisionati e non su dati diversi e, per il learner, imprevedibili. Normalmente, l'overfitting si verifica quando ci si è allenati troppo a lungo sugli stessi dati, o quando tali dati sono troppo scarsi per rappresentare la realtà in modo significativo.

L'immagine sottostante, tratta da Wikipedia (**Ghiles, 2017**), illustra il problema sotto forma grafica: in pratica i dati da prevedere, i punti neri, possono essere efficacemente modellizzati sia con una retta sia con una interpolazione polinomiale. Ma, da quello che la figura lascia intuire, è facile prevedere che la semplice retta sia in grado di dare migliori risultati in futuro, in termini di previsione, se la serie presenterà la stessa dispersione di quella rappresentata, sebbene l'interpolazione polinomiale appaia, in quell'intervallo, perfetta.

In pratica l'overfitting ci ricorda che è bene non perdere di vista i fondamenti essenziali del fenomeno da studiare in nome della ricerca di accuratezza estrema nell'adattamento ai dati disponibili.

Figura 7 – Esempio grafico di Overfitting



ENEA
Servizio Promozione e Comunicazione
www.enea.it

Stampa: Laboratorio Tecnografico ENEA - C.R. Frascati
giugno 2017