UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Matematica
Corso di Laurea Magistrale in Matematica

Tesi di Laurea Magistrale

TEOREMI DI NO FREE LUNCH

Laureando

Alessandro Berti

Matricola 1035267

Relatore

Prof. Marco Ferrante

Anno Accademico 2012/13

A Sampei, folle cane.

It was the institution of the 'free lunch' I had struck.
You paid for a drink and got as much as you wanted to eat.
For something less than a rupee a day a man
can feed himself sumptuously in San Francisco,
even though he be a bankrupt.

Remember this if ever you are stranded in these part.

(Rudyard Kipling, 1891)

Indice

Introduzione

There ain't such a thing as a free lunch: nulla di ciò che possiamo avere si ottiene senza pagare un qualche tipo di dazio.

Erano numerosi i locali in America, nel diciannovesimo secolo, che dicevano di offrire un pasto gratis, *free lunch!*, a chiunque si sedeva a un tavolo per bere qualcosa. La natura dei piatti offerti, piccanti, salati o caldi, era però tale da favorire numerose ordinazioni di bevande. Niente era dato per niente, dunque.

Le scienze ci dicono che nulla nasce dal nulla. Non esiste per esempio una sorgente in grado di creare energia o materia, ma tutto si deve a trasformazioni di ciò che c'è già. There ain't such a thing as a free lunch.

Saremmo però portati a pensare che tale principio non valga nel caso degli algoritmi di risoluzione per problemi di ottimizzazione, ossia che esistano tecniche migliori rispetto, per esempio, alla ricerca casuale dell'ottimo.

La mia tesi è dedicata proprio allo studio di questi problemi, noti come teoremi di No Free Lunch (NFL).

Nel primo capitolo ho mostrato che, nel caso finito, vale il *no free lunch* (NFL) e quindi tutti gli algoritmi di ottimizzazione in media si equivalgono, quando analizzati sull'intera classe di funzioni (oppure su sottoinsiemi C.U.P.). Ho proseguito l'esposizione spiegando un criterio "debole" di scelta tra algoritmi (il criterio Mini-Max) e alcune strategie di ottimizzazione che sono valide quando consideriamo sottoinsiemi di funzioni su cui NFL non vale.

Nel secondo capitolo ho trattato il *no free lunch* nel caso numerabile. Non ho potuto trasportare il teorema del caso finito: tuttavia, ho mostrato che vale una proprietà leggermente più debole, la *generalized no free lunch*, $GNFL$; e quindi gli algoritmi in media si equivalgono anche nel caso numerabile.

Nel terzo capitolo ho mostrato che, se considero insiemi che siano infiniti non numerabili, non vale nè la NFL nè la $GNFL$.

Questo fatto è stato presentato nell'articolo [3] di Anne Auger e Olivier Teytaud. Tuttavia, la dimostrazione proposta contiene alcuni passaggi poco chiari, i quali possono autorizzare a pensare che la proprietà $GNFL$ sia in realtà soddisfatta. Ho quindi ritenuto opportuno lavorare a una nuova dimostrazione che fosse consistente, e che mostrasse che la proprietà $GNFL$ non è soddisfatta nel caso di insiemi che siano infiniti non numerabili. Quindi il *no free lunch* non vale ed è, almeno teoricamente parlando, possibile trovare algoritmi migliori di altri.

Se le funzioni in questione sono continue, posso sfruttare proprio la continuità per ricavare algoritmi ben funzionanti rispetto al problema di ottimizzazione posto. Per un problema di massimo o di minimo di una funzione su un determinato dominio, posso ad esempio usare il *simulated annealing* o l'*algoritmo delle api*, come riportato nell'appendice B.

Capitolo 1

Teoremi per il caso finito

1.1 Algoritmi e funzioni di costo

Gli oggetti fondamentali di questa tesi sono gli *algoritmi* per i problemi di ottimizzazione. Arriveremo a mostrare che, almeno nel caso finito, sono tutti equivalenti (in un senso che sarà chiarito nel teorema NFL).

Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi *finiti*, e $\mathcal{F}$ l'insieme di tutte le applicazioni da $\mathcal{X}$ in $\mathcal{Y}$, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Consideriamo un problema di ottimizzazione per $f \in \mathcal{F}$, per esempio

$$\min_{x \in \mathcal{X}} \ f(x) \quad \text{oppure} \quad \max_{x \in \mathcal{X}} \ f(x)$$

la cui risoluzione può essere effettuata applicando un opportuno algoritmo a, cioè una procedura che ci consente di trovare l'ottimo richiesto.

Questa funziona in maniera iterativa, visitando nuovi punti e calcolandovi i valori della funzione.

Più precisamente, ci permette di costruire due vettori, $X(f, m, a)$ e $Y(f, m, a)$, di dimensione m ($m \in \{1, \ldots, |\mathcal{X}|\}$) che rappresentano rispettivamente il vettore delle ascisse (ciascuna delle quali $\in \mathcal{X}$) e i valori che la funzione assume in quei punti (ciascuno dei quali $\in \mathcal{Y}$), in maniera iterativa:

- si sceglie l'ascissa iniziale $x_0 \in \mathcal{X}$, e si pone $X(f, 1, a) = x_0$, $Y(f, 1, a) = f(x_0)$.

- per $2 \leq i \leq m$ l'algoritmo a determina $X(f, i, a)$ e $Y(f, i, a)$, conoscendo $X(f, i-1, a)$ e $Y(f, i-1, a)$, scegliendo la nuova ascissa $X(f, i, a)(i)$, diversa da tutte le precedenti, e calcolandovi il valore della funzione trovando $Y(f, i, a)(i)$.

Possiamo anche considerare algoritmi *casuali*, dove a è una variabile aleatoria, definita su uno spazio di probabilità $(\tilde{\Omega}, \tilde{\mathcal{A}}, \tilde{P})$, a valori nello spazio degli algoritmi (deterministici).

Siamo interessati ora a "misurare" la bontà di un algoritmo rispetto al nostro problema di ottimizzazione.

Questo può essere fatto tramite le *funzioni di costo*, che mandano $Y(f, m, a)$ in un valore $k \in \mathbb{R}$. Per esempio, se $\mathcal{Y}$ è un sottoinsieme finito di $\mathbb{R}$ e si sta affrontando un problema di minimo, allora una buona scelta di funzione è quella di mandare un vettore $Y(f, m, a)$ nel valore di $\mathbb{R}$ che è il minimo delle sue componenti.

1.2 Proprietà NFL finite

Nella sezione successiva, arriveremo a provare un teorema di *No Free Lunch* per insiemi finiti, ed è necessario definire cosa intendiamo per NFL.

Definizione 1.1:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria, c funzione di costo. Si dice che la proprietà $NFL(\mathcal{X}, f, c)$ è soddisfatta se le due variabili aleatorie $c(Y(f, m, a_1))$ e $c(Y(f, m, a_2))$ hanno la stessa distribuzione, per ogni $m \in \{1, \ldots, \mathcal{X}\}$ e ogni coppia di algoritmi a_1 e a_2.

Definizione 1.2:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria. Si dice che la proprietà $NFL(\mathcal{X}, f)$ è soddisfatta se, per ogni possibile funzione di costo c, abbiamo che $NFL(\mathcal{X}, f, c)$ è soddisfatta.

Il seguente lemma ci permetterà di ridefinire la proprietà $NFL(\mathcal{X}, f)$ in una maniera più elegante.

Lemma 1.3:

Sia $\mathcal{M}$ l'insieme delle funzioni misurabili da $\mathbb{R}^n$ in $\mathbb{R}$. Sia $\mathcal{M}'$ l'insieme delle funzioni indicatrici degli intervalli della forma $[-\infty, r_1] \times \ldots \times [-\infty, r_n] \subset \mathbb{R}^n$. Sia $\mathcal{A}$ l'insieme delle variabili aleatorie su $\mathbb{R}^n$. Allora sono equivalenti

a) $\forall (a_1, \ldots, a_n) \in \mathcal{A}$, $\forall c \in \mathcal{M}$ $\quad c(a_1, \ldots, a_n)$ ha la stessa distribuzione

b) $\forall (a_1, \ldots, a_n) \in \mathcal{A}$, $\forall c \in \mathcal{M}'$ $\quad c(a_1, \ldots, a_n)$ ha la stessa distribuzione

c) $\forall (a_1, \ldots, a_n) \in \mathcal{A}$, $\forall c \in \mathcal{M}'$ $\quad$ i valori attesi $E[c(a_1, \ldots, a_n)]$ sono tutti uguali

d) le $(a_1, \ldots, a_n) \in \mathcal{A}$ hanno tutte la stessa distribuzione

Dimostrazione. $d) \to a)$, $a) \to b)$ e $b) \to c)$ sono ovvie.

$c) \to d)$: siano $r_1, \ldots, r_n \in \mathbb{R}$, $\varepsilon \in \mathbb{R}^+$. Sia $c \in \mathcal{M}'$ funzione indicatrice dell'intervallo $[-\infty, r_1] \times \ldots \times [-\infty, r_n]$. Sia $c_\varepsilon \in \mathcal{M}'$ funzione indicatrice dell'intervallo $[-\infty, r_1 + \varepsilon] \times \ldots \times [-\infty, r_n + \varepsilon]$. Se $(a_1, \ldots, a_n)$ e $(b_1, \ldots, b_n) \in A$, con rispettive funzioni densità di probabilità f e g, abbiamo che

$$\int_{\mathbb{R}^n} c(x) f(x) dx = \int_{\mathbb{R}^n} c(x) g(x) dx$$

$$\int_{\mathbb{R}^n} c_\varepsilon(x) f(x) dx = \int_{\mathbb{R}^n} c_\varepsilon(x) g(x) dx$$

Quindi

$$\int_{\mathbb{R}^n} c(x) \left(f(x) - g(x)\right) dx = 0 = \int_{\mathbb{R}^n} c_\varepsilon(x) \left(f(x) - g(x)\right) dx$$

e

$$\int_{\mathbb{R}^n} \left(c(x) - c_\varepsilon(x)\right) \left(f(x) - g(x)\right) dx = 0$$

dato che questo deve valere per ogni scelta di $r_1, \ldots, r_n \in \mathbb{R}$ e $\varepsilon \in \mathbb{R}^+$, allora $f(x)$ è uguale a $g(x)$ su ogni n-cubo di $\mathbb{R}^n$, quindi su ogni aperto di $\mathbb{R}^n$, di conseguenza $(a_1, \ldots, a_n)$ e $(b_1, \ldots, b_n)$ devono avere la stessa distribuzione. □

Possiamo quindi dire che

Proposizione 1.4:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria. Se le due variabili aleatorie $Y(f, m, a_1)$ e $Y(f, m, a_2)$ hanno la stessa distribuzione, per ogni $m \in \{1, \ldots, \mathcal{X}\}$ e ogni coppia di algoritmi a_1 e a_2, allora la proprietà $NFL(\mathcal{X}, f)$ è soddisfatta.

1.3 Teorema NFL finito

Enunciamo ora il teorema fondamentale di questo capitolo, che è il teorema di No Free Lunch finito. Esso ci mostra che, quando due algoritmi a_1 e a_2 sono considerati sull'intera classe dei problemi, è perfettamente equivalente usare uno o l'altro.

Teorema 1.5 (Teorema NFL finito)**:**
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria uniformemente distribuita su $\mathcal{F}$. Allora la proprietà $NFL(\mathcal{X}, f)$ è soddisfatta.

Per vedere la dimostrazione, ci portiamo in una forma equivalente:

Proposizione 1.6:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria uniformemente distribuita su $\mathcal{F}$. $NFL(\mathcal{X}, f)$ è soddisfatta se e solo se abbiamo che

$$\sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a_1) = \sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a_2)$$

per ogni sequenza d_m^y in $\mathcal{Y}^m$, per ogni $m \in \{1, \ldots, \mathcal{X}\}$ e ogni coppia di algoritmi a_1 e a_2, dove $P(d_m^y | f_0, m, a_1) = 1_{\{Y(f_0, m, a_1) = d_m^y\}}$.

Dimostrazione. $Y(f, m, a_1)$ ha la stessa distribuzione di $Y(f, m, a_2)$ se e solo se

$$P(Y(f, m, a_1) = d_m^y) = P(Y(f, m, a_2) = d_m^y)$$

per ogni $d_m^y \in \mathcal{Y}^{\mathcal{X}}$. Ricordando che f è uniformemente distribuita su $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$, questo è equivalente a scrivere

$$\sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a_1) = \sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a_2)$$

dove $P(d_m^y | f_0, m, a_1)$ può essere considerata $P(Y(f, m, a_1) = d_m^y | f = f_0)$. □

Teorema 1.7:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Allora

$$\sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a_1) = \sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a_2)$$

per ogni sequenza d_m^y in $\mathcal{Y}^m$, per ogni $m \in \{1, \ldots, \mathcal{X}\}$ e ogni coppia di algoritmi a_1 e a_2.

Dimostrazione. Dobbiamo mostrare che $\sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a) = \sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m)$ per ogni $f_0 \in \mathcal{F}$ e $m \in \{1, \ldots, |\mathcal{X}|\}$, cioè non dipende da a.

Osserviamo che se d_m^y non è realizzabile, ossia non esistono f_0 e a tali che $P(d_m^y | f_0, m, a) \neq 0$, allora banalmente non c'è dipendenza da a e $P(d_m^y | f_0, m) = 0$.

Supponiamo quindi d'ora in avanti d_m^y realizzabile, ossia che esista una funzione $f_0 \in \mathcal{F}$ e un vettore d_m^x tali che $f_0(d_m^x) = d_m^y$.

La dimostrazione si svolge per induzione su m.
Se $m = 1$, supponiamo di avere $d_1^y \in \mathcal{Y}$ e $d_1^x \in \mathcal{X}$. Allora

$$\sum_{f_0 \in \mathcal{F}} P(d_1^y | f_0, m = 1, a) = \sum_{f_0 \in \mathcal{F}} \delta(d_1^y, f_0(d_1^x)) = |\mathcal{Y}|^{|\mathcal{X}|-1}$$

che è la cardinalità dell'insieme delle funzioni di $\mathcal{F}$ che in d_1^x assumono il valore d_1^y.

Supponiamo vera la tesi per m. Proviamola per $m+1$.

$$\sum_{f_0 \in \mathcal{F}} P(d^y_{m+1}|f_0, m+1, a) = \sum_{f_0 \in \mathcal{F}} P(d^y_{m+1}(m+1)|d_m, f_0, m+1, a) \cdot P(d^y_m|f_0, m+1, a)$$

Il nuovo valore delle y dipenderà solo dal valore delle x, da f_0 e da nient'altro. Quindi

$$\sum_{f_0 \in \mathcal{F}} P(d^y_{m+1}|f_0, m+1, a) = \sum_{f_0, x} P(d^y_{m+1}(m+1)|f_0, x) \cdot$$

$$\cdot P(x|d_m, f_0, m+1, a) \cdot P(d^y_m|f_0, m+1, a) =$$

$$= \sum_{f_0, x} \delta(d^y_{m+1}(m+1), f_0(x)) \cdot P(x|d_m, f_0, m+1, a) \cdot P(d^y_m|f_0, m+1, a)$$

Ora, $x = a(d^x_m, d^y_m)$ (ossia si trova applicando l'algoritmo) e quindi possiamo scrivere

$$\sum_{f_0 \in \mathcal{F}} P(d^y_{m+1}|f_0, m+1, a) = \sum_{f_0, d^x_m} \delta(d^y_{m+1}(m+1), f_0(a(d_m)) \cdot$$

$$\cdot P(d_m|f_0, m, a)$$

Il termine $P(d_m|f_0, m, a)$ dipende solo dai valori di x compresi in d^x_m, mentre $\delta(d^y_{m+1}(m+1), f_0(a(d_m)))$ dipende solo dai valori di x non compresi in d^x_m ($a(d_m)$ non appartiene a d^x_m). Possiamo quindi scrivere

$$\sum_{f_0 \in \mathcal{F}} P(d^y_{m+1}|f_0, m+1, a) = \sum_{d^x_m} \sum_{f_0(x \in d^x_m)} P(d_m|f_0, m, a) \cdot$$

$$\cdot \sum_{f_0(x \notin d^x_m)} \delta(d^y_{m+1}(m+1), f_0(a(d_m)))$$

L'ultimo termine è una costante, $|\mathcal{Y}|^{|\mathcal{X}|-m-1}$ (la cardinalità dell'insieme delle funzioni che in d^x_{m+1} assumono d^y_{m+1}), e può essere tirato fuori dalla sommatoria

$$\sum_{f_0 \in \mathcal{F}} P(d^y_{m+1}|f_0, m+1, a) = |\mathcal{Y}|^{|\mathcal{X}|-m-1} \sum_{d^x_m, f_0(x \in d^x_m)} P(d_m|f_0, m, a) =$$

$$= \frac{1}{|\mathcal{Y}|} \sum_{d_m^x, f_0} P(d_m | f_0, m, a) =$$

$$= \frac{1}{|\mathcal{Y}|} \sum_{f_0 \in \mathcal{F}} P(d_m^y | f_0, m, a)$$

e il termine a sinistra non dipende da a dato che il termine a destra non vi dipende. □

Vediamo anche la seguente (semplice!) riformulazione del teorema NFL

Proposizione 1.8:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria uniformemente distribuita su $\mathcal{F}$. $NFL(\mathcal{X}, f)$ è soddisfatta se e solo se abbiamo che

$$\sum_{f_0 \in \mathcal{F}} \delta(k, c(Y(f_0, m, a_1))) = \sum_{f_0 \in \mathcal{F}} \delta(k, c(Y(f_0, m, a_2)))$$

per ogni $m \in \{1, \ldots, \mathcal{X}\}$, per ogni coppia di algoritmi a_1 e a_2, per ogni funzione di costo c, per ogni $k \in \mathbb{R}$, dove δ indica la funzione delta di Kronecker.

Dimostrazione. Abbiamo che $NFL(\mathcal{X}, f)$ è soddisfatta se e solo se $c(Y(f, m, a_1))$ ha la stessa distribuzione di $c(Y(f, m, a_2))$ per ogni $m \in \{1, \ldots, \mathcal{X}\}$, per ogni coppia di algoritmi a_1 e a_2, per ogni funzione di costo c, per ogni $k \in \mathbb{R}$. Questo è equivalente a dire che

$$\sum_{f_0 \in \mathcal{F}} \delta(k, c(Y(f_0, m, a_1))) = \sum_{f_0 \in \mathcal{F}} \delta(k, c(Y(f_0, m, a_2)))$$

□

1.4 Sottoinsiemi C.U.P.

Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti. Siamo interessati a vedere se il teorema NFL vale anche su determinati sottoinsiemi di $\mathcal{Y}^{\mathcal{X}}$. Questi sottoinsiemi sono quelli chiusi per permutazioni.

Definizione 1.9:
$\mathcal{F}$, sottoinsieme di $\mathcal{Y}^{\mathcal{X}}$, si dice chiuso per permutazioni (C.U.P.) se, $\forall \pi$ permutazione di $\mathcal{X}$, e $\forall f \in \mathcal{F}$, abbiamo $f \circ \pi \in \mathcal{F}$.

Enunciamo ora il teorema di NFL per sottoinsiemi C.U.P.

Teorema 1.10:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti, $\mathcal{F}$ sottoinsieme C.U.P. di $\mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria uniformemente distribuita su $\mathcal{F}$. Allora la proprietà $NFL(\mathcal{X}, f)$ è soddisfatta.

In questo caso, notiamo come le funzioni che possono essere assunte da f sono solo quelle contenute in $\mathcal{F}$ sottoinsieme C.U.P. di $\mathcal{Y}^{\mathcal{X}}$, ciascuna delle quali può essere scelta con la stessa probabilità.

Tuttavia, il numero di sottoinsiemi C.U.P. è esiguo rispetto al totale. Abbiamo infatti che

Teorema 1.11:
Il numero di sottoinsiemi di $\mathcal{Y}^{\mathcal{X}}$ è $2^{|\mathcal{Y}|^{|\mathcal{X}|}}$ Il numero di sottoinsiemi C.U.P. è

$$2^{\binom{|\mathcal{X}|+|\mathcal{Y}|-1}{|\mathcal{X}|}}$$

Quindi la frazione di sottoinsiemi C.U.P. è

$$\frac{2^{\binom{|\mathcal{X}|+|\mathcal{Y}|-1}{|\mathcal{X}|}}}{2^{|\mathcal{Y}|^{|\mathcal{X}|}}}$$

Dimostrazione. Definiamo l'*istogramma* di una funzione $f : \mathcal{X} \to \mathcal{Y}$ come un vettore di dimensione $|\mathcal{Y}|$ che contiene le dimensioni delle preimmagini di ogni $y \in \mathcal{Y}$

$$h(y) = \#\{x : f(x) = y\}$$

Due funzioni possono avere lo stesso istogramma. Sia $\sim$ la relazione di equivalenza "avere lo stesso istogramma". Possiamo chiederci quante classi di equivalenza di $\sim$ ci sono in $\mathcal{Y}^{\mathcal{X}}$.

Queste sono in numero uguale a $\binom{|\mathcal{X}|+|\mathcal{Y}|-1}{|\mathcal{X}|}$. Infatti, sono le soluzioni intere non negative dell'equazione $x_1 + \ldots + x_{|\mathcal{Y}|} = |\mathcal{X}|$ (vedere il libro [8] di Sheldon M. Ross, pag. 12)

I sottoinsiemi C.U.P., data una funzione, devono contenere tutte le funzioni con lo stesso istogramma. Quindi il loro numero è uguale al numero di sottoinsiemi di un insieme $\mathcal{A}$ che contiene un rappresentante per ogni classe di equivalenza, e la tesi segue. □

Già per $\mathcal{X}$ e $\mathcal{Y}$ piccoli la frazione (di sottoinsiemi C.U.P.) è molto piccola, per esempio se consideriamo le funzioni Booleane $\{0,1\}^3 \to \{0,1\}$ la frazione è $\approx 10^{-74}$.

Usando le disuguaglianze $\binom{n}{m} \leq n^m/(m!)$ e

$$\sqrt{2\pi} m^{m+1/2} e^{-m} e^{(12m+1)^{-1}} < m! < \sqrt{2\pi} m^{m+1/2} e^{-m} e^{(12m)^{-1}}$$

per $n, m \in \mathbb{N}$ abbiamo

$$\left(\frac{2^{\binom{|\mathcal{X}|+|\mathcal{Y}|-1}{|\mathcal{X}|}}}{2^{|\mathcal{Y}|^{|\mathcal{X}|}}} \right) < 2^{(e+e|\mathcal{Y}|/|\mathcal{X}|-e/|\mathcal{X}|)^{|\mathcal{X}|}-|\mathcal{Y}|^{|\mathcal{X}|}}$$

Per $|\mathcal{Y}| > e\frac{|\mathcal{X}|}{|\mathcal{X}|-e}$ e $|\mathcal{X}| > 2$ quest'espressione non è più grande di

$$2^{(e+e|\mathcal{Y}|/|\mathcal{X}|-e/|\mathcal{X}|-|\mathcal{Y}|)^{|\mathcal{X}|}}$$

e converge a zero esponenzialmente.

1.5 NFL non uniforme

La dimostrazione del teorema 1.11 ci ha mostrato che si può costruire un sottoinsieme C.U.P. scegliendo una funzione (deterministica) $f_0 : \mathcal{X} \to \mathcal{Y}$ e considerando

$$\Pi_{f_0}^{\mathcal{X}} = \{f_0 \circ \pi_0, \pi_0 \text{ permutazione di } \mathcal{X}\}$$

Questi sono sottoinsiemi C.U.P. generati dalle permutazioni di una singola funzione f_0.

Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti. Ci chiediamo cosa succede se prendiamo f variabile aleatoria che non sia uniformemente distribuita nè su $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$ nè su $\mathcal{F}$ sottoinsieme C.U.P. di $\mathcal{Y}^{\mathcal{X}}$. In generale, $NFL(\mathcal{X}, f)$ non vale; possiamo però dimostrare che se f è variabile aleatoria che, su ognuno dei sottoinsiemi $\Pi_{f_0}^{\mathcal{X}}$ di $\mathcal{Y}^{\mathcal{X}}$, ha distribuzione uniforme, allora $NFL(\mathcal{X}, f)$ vale.

Teorema 1.12:
Siano $\mathcal{X}, \mathcal{Y}$ insiemi finiti, $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $f : \Omega \to \mathcal{Y}^{\mathcal{X}}$ variabile aleatoria che, su ognuno dei sottoinsiemi $\Pi_{f_0}^{\mathcal{X}}$ di $\mathcal{Y}^{\mathcal{X}}$, ha distribuzione uniforme. Allora

$$\sum_{f_0 \in \mathcal{F}} P(f_0)\delta(k, c(Y(f_0, m, a_1))) = \sum_{f_0 \in \mathcal{F}} P(f_0)\delta(k, c(Y(f_0, m, a_2)))$$

per ogni $m \in \{1, \ldots, \mathcal{X}\}$, per ogni coppia di algoritmi a_1 e a_2, per ogni funzione di costo c, per ogni $k \in \mathbb{R}$, dove δ indica la funzione delta di Kronecker e $P(f_0)$ la probabilità che f assuma f_0.

Dimostrazione. Sia $\mathcal{H}$ l'insieme di tutti gli istogrammi delle funzioni di $\mathcal{Y}^{\mathcal{X}}$. Allora

$$\sum_{f_0 \in \mathcal{F}} P(f_0)\delta(k, c(Y(f_0, m, a_1))) = \sum_{h \in \mathcal{H}} \sum_{f \in B_h} P(f_0)\delta(k, c(Y(f_0, m, a_1)))$$

Abbiamo supposto che f abbia distribuzione uniforme su ognuno dei sottoinsiemi

C.U.P. di $\mathcal{Y}^{\mathcal{X}}$, quindi per opportuni p_h abbiamo

$$\sum_{h\in\mathcal{H}}\sum_{f\in B_h} P(f_0)\delta(k,c(Y(f_0,m,a_1))) = \sum_{h\in\mathcal{H}} p_h \sum_{f\in B_h} \delta(k,c(Y(f_0,m,a_1)))$$

Valendo il teorema NFL per sottoinsiemi C.U.P., abbiamo

$$\sum_{h\in\mathcal{H}} p_h \sum_{f\in B_h} \delta(k,c(Y(f_0,m,a_1))) = \sum_{h\in\mathcal{H}} p_h \sum_{f\in B_h} \delta(k,c(Y(f_0,m,a_2))) =$$

$$= \sum_{f_0\in\mathcal{F}} P(f_0)\delta(k,c(Y(f_0,m,a_2)))$$

come voluto. □

Esempio 1.13:

Consideriamo $\mathcal{X} = \{0,1\}$, $\mathcal{Y} = \{0,1\}$. Allora $\mathcal{Y}^{\mathcal{X}} = \{(0,0),(0,1),(1,0),(1,1)\}$ (sono funzioni: per ognuna, si è inteso che il valore di $\mathcal{Y}$ che la funzione assume nell'i-esimo valore di $\mathcal{X}$ sia l'i-esima componente del vettore. Per esempio, $(1,0) \to f(0) = 1,\ f(1) = 0$). Una variabile aleatoria f che si distribuisce su $\mathcal{Y}^{\mathcal{X}}$ nel modo seguente:

$$P(f=(0,0)) = \frac{1}{8} \qquad P(f=(1,0)) = \frac{2}{8}$$

$$P(f=(0,1)) = \frac{2}{8} \qquad P(f=(1,1)) = \frac{3}{8}$$

soddisfa le condizioni del teorema di NFL non uniforme.

1.6 Proprietà NFL "deterministiche"

La dimostrazione del teorema 1.11 ci ha mostrato che si può costruire un sottoinsieme C.U.P. scegliendo una funzione (deterministica) $f_0 : \mathcal{X} \to \mathcal{Y}$ e considerando

$$\Pi_{f_0}^{\mathcal{X}} = \{f_0 \circ \pi_0, \pi_0 \text{ permutazione di } \mathcal{X}\}$$

Ciò ci permette di definire una proprietà NFL per funzioni[1] (deterministiche) $f_0 : \mathcal{X} \to \mathcal{Y}$. Per arrivarci, definiamo il concetto di *permutazione casuale*

Definizione 1.14:
Sia

$$\Pi^{\mathcal{X}} = \{\pi_0, \pi_0 \text{ permutazione di } \mathcal{X}\}$$

l'insieme di tutte le permutazioni di $\mathcal{X}$. Una variabile aleatoria π uniformemente distribuita su $\Pi^{\mathcal{X}}$ è detta permutazione casuale su $\mathcal{X}$.

Si osservi che, se π è una permutazione casuale su $\mathcal{X}$, $f = f_0 \circ \pi$ è una variabile aleatoria uniformemente distribuita su $\Pi^{\mathcal{X}}_{f_0}$ che soddisfa la proprietà $NFL(\mathcal{X}, f)$. Le seguenti proprietà ridefiniscono NFL prendendo una funzione (deterministica) f_0 e componendola con la permutazione casuale π.

Definizione 1.15:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti. Sia $f_0 : \mathcal{X} \to \mathcal{Y}$. Sia π permutazione casuale su $\mathcal{X}$. Diciamo che la proprietà $NFL(\mathcal{X}, \pi, f_0)$ è soddisfatta se, per ogni coppia di algoritmi a_1 e a_2 e ogni $m \in \{1, \ldots, \mathcal{X}\}$ abbiamo che $Y(f_0 \circ \pi, m, a_1)$ e $Y(f_0 \circ \pi, m, a_2)$ hanno la stessa distribuzione.

Definizione 1.16:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti. Sia $f_0 : \mathcal{X} \to \mathcal{Y}$. Diciamo che la proprietà $NFL(\mathcal{X}, f_0)$ è soddisfatta se esiste permutazione casuale π su $\mathcal{X}$ tale che $NFL(\mathcal{X}, \pi, f_0)$ sia soddisfatta

Definizione 1.17:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti. Diciamo che la proprietà $NFL(\mathcal{X})$ è soddisfatta se, per ogni $f \in \mathcal{Y}^{\mathcal{X}}$, abbiamo che $NFL(\mathcal{X}, f)$ è soddisfatta.

[1] Sinora si era presa f variabile aleatoria uniformemente distribuita su $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$ o su $\mathcal{F}$ C.U.P.

è importante osservare che le definizioni appena viste non dicono nulla di diverso rispetto alle definizioni date a inizio capitolo, ma nascono semplicemente da un punto di vista differente e complementare. Possiamo usare il teorema di NFL per i sottoinsiemi C.U.P. per dire che

Teorema 1.18:
Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti. Allora $NFL(\mathcal{X})$ è soddisfatta.

1.7 Distinzioni minimax

Il teorema di No Free Lunch finito ci dice che le prestazioni di due algoritmi a_1 e a_2 sono equivalenti se si considerano sottoinsiemi C.U.P. di $\mathcal{F} = \mathcal{Y}^{\mathcal{X}}$. Questo, da un punto di vista computazionale, ci lascia indecisi su quale algoritmo adottare per risolvere un problema di ottimizzazione.

Ci chiediamo se esistano criteri di scelta per far *vincere* un algoritmo.

Un criterio utile di scelta si ha quando esiste una classe di problemi dove a_1 funziona "molto meglio" di a_2, mentre non esiste una classe di problemi dove a_2 funziona "molto meglio" di a_1. Diamo una definizione precisa di questo.

Sia c funzione di costo: questa assume un valore minimo c_{min} e un valore massimo c_{max}. Fissiamo $m \in \{1, \dots, |\mathcal{X}|\}$. Se riusciamo a trovare una funzione $f_1 \in \mathcal{F}$ tale che $c(Y(f_1, m, a_1)) = c_{max}$ e $c(Y(f_1, m, a_2)) = c_{min}$, ma non riusciamo a trovare una funzione $f_2 \in \mathcal{F}$ tale che $c(Y(f_2, m, a_1)) = c_{min}$ e $c(Y(f_2, m, a_2)) = c_{max}$, allora diciamo che a_1 *vince secondo il criterio minimax* rispetto ad a_2.

Esempio 1.19:
Prendiamo $\mathcal{X}, \mathcal{Y}$ tali che $|\mathcal{X}| = |\mathcal{Y}| = 3$; possiamo scrivere
$\mathcal{X} = \{x_1, x_2, x_3\} \qquad \mathcal{Y} = \{y_1, y_2, y_3\}$

Fissiamo $m = 2$. Consideriamo una funzione di costo c tale che

$c((y_2, y_3)) = c((y_3, y_2)) = 2 \qquad c((y_1, y_2)) = c((y_2, y_1)) = 0 \qquad c = 1$ altrimenti

Siano a_1 e a_2 due algoritmi.
Facciamo partire l'algoritmo a_1 dal punto x_2: questo, se vi trova il valore y_1, sceglie come nuova ascissa x_1, altrimenti sceglie x_3.
Facciamo partire l'algoritmo a_2 dal punto x_1: questo, se vi trova i valori y_1 o y_2 sceglie come nuova ascissa x_2, altrimenti sceglie x_3.

Se prendiamo f_1 tale che $f_1(x_1) = y_1$, $f_1(x_2) = y_2$ e $f_1(x_3) = y_3$, allora $c(Y(f_1, 2, a_1)) = 2 = c_{max}$ e $c(Y(f_1, 2, a_2)) = 0 = c_{min}$.

Ci chiediamo se esiste f_2 tale che $c(Y(f_2, 2, a_1)) = 0$ e $c(Y(f_2, 2, a_2)) = 2$. Esaminiamo le possibili combinazioni di esiti di a_1 e a_2 che rispettano tali condizioni.

1) $\{(y_1, y_2), (y_2, y_3)\}$ 2) $\{(y_1, y_2), (y_3, y_2)\}$
3) $\{(y_2, y_1), (y_2, y_3)\}$ 4) $\{(y_2, y_1), (y_3, y_2)\}$

Se 1) fosse possibile, allora f dovrebbe essere contemporaneamente della forma (y_2, y_1, s) e (y_2, y_3, z), ciò è impossibile.

Se 2) fosse possibile, allora f dovrebbe essere contemporaneamente della forma (y_2, y_1, s) e (y_3, z, y_2), ciò è impossibile.

Se 3) fosse possibile, allora f dovrebbe essere contemporaneamente della forma (s, y_2, y_1) e (y_2, y_3, z), ciò è impossibile.

Se 4) fosse possibile, allora f dovrebbe essere contemporaneamente della forma (s, y_2, y_1) e (y_3, z, y_2), ciò è impossibile.

Quindi possiamo applicare il criterio minimax per dire che a_1 vince su a_2.

1.8 Generazione di un algoritmo di ottimizzazione

In questa sezione introdurremo un metodo per generare un algoritmo di ottimizzazione laddove il teorema di *No Free Lunch* non vale.

Proposizione 1.20:

Siano $\mathcal{X}$ e $\mathcal{Y}$ insiemi finiti. Sia $\mathcal{F}$ un sottoinsieme di $\mathcal{Y}^{\mathcal{X}}$ che non è C.U.P. Allora, in generale,

$$\sum_{f_0 \in \mathcal{F}} \delta(k, c(Y(f_0, m, a_1))) \neq \sum_{f_0 \in \mathcal{F}} \delta(k, c(Y(f_0, m, a_2)))$$

per c, m, a_1 e a_2 scelti.

Supponiamo di avere un gioco con due giocatori (A e B): il giocatore A sceglie una funzione di costo f_0 in $\mathcal{F}$ (senza rivelare la sua scelta a B) e il giocatore B deve scoprire f_0. Per ogni turno del gioco, il giocatore B puo' scegliere un valore $x \in \mathcal{X}$ e il giocatore A è obbligato a rivelare $f_0(x)$. L'obiettivo del giocatore B è di trovare la soluzione f_0 nel più piccolo numero possibile di passi. Assumiamo che B conosca esattamente ciascuna delle funzioni $f \in \mathcal{F}$.

In ogni turno la strategia migliore è chiedere il valore di f_0 in quei valori $x \in \mathcal{X}$ che "escludono" il maggior numero di funzioni, minimizzando il numero atteso di funzioni che rimangono. Questo si traduce, per ogni punto $x \in \mathcal{X}$, nel:
1- trovare una lista di valori $(s_1, \ldots, s_{|\mathcal{Y}|})$ dove s_i è il numero di funzioni che in x assumono l'i-esimo valore di $\mathcal{Y}$.
2- calcolare, per ogni $x \in \mathcal{X}$, il valore $\sum s_i^2$ (che chiameremo costo), trovando l'x^* migliore (che è quella col costo più basso).
3- chiedere ad A il valore $f_0(x^*)$.

Tale procedura viene iterata fino a rimanere con una sola possibile candidata: e questa funzione è l'f_0 cercata.

Esempio 1.21:

Sia $\mathcal{X} = \mathcal{Y} = \{0, 1, 2, 3, 4\}$ e consideriamo l'insieme di funzioni $\mathcal{F} = \{f_1, \ldots, f_8\}$ con $f_1 = (1, 2, 3, 4, 0)$, $f_2 = (3, 1, 2, 0, 0)$, $f_3 = (4, 1, 1, 2, 3)$, $f_4 = (0, 0, 0, 1, 2)$, $f_5 = (0, 0, 0, 1, 1)$, $f_6 = (0, 0, 0, 0, 1)$, $f_7 = (4, 1, 0, 2, 3)$, $f_8 = (0, 1, 0, 0, 0)$.

Supponiamo che la funzione obiettivo (quella scelta dal giocatore A) sia f_6. Costruiamo

le liste di valori per la prima iterazione

x	lista	valore
0	4|1|1|2	22
1	3|4|1|0	26
2	5|1|1|1	28
3	3|2|2|1	18
4	3|2|1|2	18

Potremmo chiedere ad A il valore della funzione in 3 o in 4: glielo chiediamo in 3. La risposta del giocatore A è $f_0(3) = 0$. Ora, il giocatore B può modificare l'insieme di soluzioni candidate eliminando quelle funzioni per cui $f(3) \neq 0$. Rimaniamo con $\mathcal{F} = \{f_2, f_6, f_8\}$. Possiamo iterare il processo, per altre due iterazioni, e alla fine ci resta solo una funzione possibile, cioè f_6.

Capitolo 2

Teoremi per il caso numerabile

2.1 Proprietà NFL

Siamo interessati a vedere se il teorema NFL continua a valere quando $\mathcal{X}$ è un insieme numerabile e $\mathcal{Y}$ è un sottoinsieme di $\mathbb{R}$.

Per insiemi $\mathcal{X}$ e $\mathcal{Y}$ finiti abbiamo definito una variabile aleatoria f uniformemente distribuita su $\mathcal{Y}^{\mathcal{X}}$ e abbiamo detto che la proprietà NFL è verificata se per ogni $m \in \{1, \ldots, |\mathcal{X}|\}$ e ogni coppia di algoritmi a_1, a_2, $Y(f, m, a_1)$ e $Y(f, m, a_2)$ hanno la stessa distribuzione.

Nel caso in cui $\mathcal{X}$ è infinito risulta impossibile definire una variabile aleatoria uniformemente distribuita su $\mathcal{Y}^{\mathcal{X}}$. La definizione stessa di variabile aleatoria implica la misurabilità. Affinchè ci sia misurabilità deve essere possibile definire una σ-algebra su $\mathcal{Y}^{\mathcal{X}}$. Definire tale σ-algebra è un'operazione leggermente più delicata.

Dobbiamo quindi introdurre uno strumento che ci permetta di ovviare a tali difficoltà, e rienunciare il NFL. Il concetto è quello di *processo stocastico*.

Definizione 2.1:
Siano $\mathcal{X}$ insieme numerabile e $\mathcal{Y} \subset \mathbb{R}$. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità. Un processo stocastico è una funzione misurabile

$$f : \Omega \times \mathcal{X} \to \mathcal{Y}$$

Un processo stocastico su $\mathcal{X}^{\mathcal{X}}$ che ci è utile per definire la proprietà NFL è quello

di *permutazione casuale*. Iniziamo definendo il concetto di permutazione per insiemi infiniti.

Definizione 2.2:
Sia $\mathcal{X}$ insieme numerabile. Una *permutazione* di $\mathcal{X}$ è una biezione di $\mathcal{X}$ in $\mathcal{X}$.

Definizione 2.3:
Sia $\mathcal{X}$ insieme numerabile. Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità. Una *permutazione casuale* è un processo stocastico

$$\pi : \Omega \times \mathcal{X} \to \mathcal{X}$$

tale che $\pi(\omega, \cdot)$ sia permutazione di $\mathcal{X}$, $\forall\ \omega \in \Omega$.

Vediamo ora come è conveniente riformulare le proprietà di NFL.

Definizione 2.4:
Sia $\mathcal{X}$ insieme numerabile e $\mathcal{Y} \subset \mathbb{R}$. Sia π permutazione casuale su $\mathcal{X}$. Sia f_0 funzione da $\mathcal{X}$ in $\mathcal{Y}$. Si dice che la proprietà $NFL(\mathcal{X}, \pi, f_0)$ è soddisfatta se, per ogni m e ogni coppia di algoritmi a_1 e a_2, $Y(f_0 \circ \pi, m, a_1)$ e $Y(f_0 \circ \pi, m, a_2)$ hanno la stessa distribuzione.

Definizione 2.5:
Sia $\mathcal{X}$ insieme numerabile e $\mathcal{Y} \subset \mathbb{R}$. Sia f_0 funzione da $\mathcal{X}$ in $\mathcal{Y}$. Si dice che la proprietà $NFL(\mathcal{X}, f_0)$ è soddisfatta se esiste su $\mathcal{X}$ una permutazione casuale π tale che $NFL(\mathcal{X}, \pi, f_0)$ sia soddisfatta.

Definizione 2.6:
Sia $\mathcal{X}$ insieme numerabile e $\mathcal{Y} \subset \mathbb{R}$. Si dice che la proprietà $NFL(\mathcal{X})$ è soddisfatta se per ogni f_0 funzione da $\mathcal{X}$ in $\mathcal{Y}$ abbiamo che $NFL(\mathcal{X}, f_0)$ è soddisfatta.

Queste proprietà sono perfettamente analoghe a quella che abbiamo visto in 1.15.

Tuttavia, visto che stiamo considerando processi stocastici invece che variabili aleatorie, non siamo sicuri che la permutazione casuale π che fa valere NFL esista.

Quindi, a differenza del caso finito, non siamo sicuri che $NFL(\mathcal{X})$ valga, ed infatti possiamo provare che essa non è verificata per esempio su $\mathbb{N}$.

Teorema 2.7:
Se $\mathcal{X} = \mathbb{N}$ *allora* $NFL(\mathcal{X})$ *non è soddisfatta.*

Dimostrazione. Mostriamo che $NFL(\mathcal{X}, f_0)$ non è soddisfatta per $f_0(i) = i$.

Sia π una qualsiasi permutazione casuale su $\mathcal{X}$. Scegliamo un qualsiasi algoritmo a. Applichiamo lo stesso con diversi punti iniziali (al primo passo sceglie $x_1 = i$ per $i \in \mathbb{N}$) e osserviamo che se la proprietà $NFL(\mathcal{X}, \pi, f_0)$ fosse soddisfatta, allora
$P(f_0(\pi(i)) = 1) =$ costante, che implica dato che $f_0(i) = i$
$P(\pi(i) = 1) =$ costante.

Ma sappiamo che $1 = \sum_{i=0}^{+\infty} P(\pi(i) = 1)$.
La possibilità $P(\pi(i) = 1) = 0$ è così esclusa.
Anche l'altra possibilità $P(\pi(i) = 1) > 0$ è da escludere, perchè porterebbe a una somma infinita.

Quindi non esiste nessuna permutazione casuale π tale che $NFL(\mathcal{X}, \pi, f_0)$ sia soddisfatta. □

è lecito a questo punto chiedersi se esistono funzioni che soddisfano $NFL(\mathbb{N}, f_0)$. Possiamo subito osservare che la funzione identicamente nulla è tale da soddisfare $NFL(\mathbb{N}, f_0)$.

Esempio 2.8:
Se f_0 è la funzione nulla, allora $NFL(\mathbb{N}, f_0)$ è soddisfatta.

Dimostrazione. Ovvia: qualunque algoritmo decidiamo di usare, quest'ultimo non trova altro che 0 □

Tuttavia, esistono anche esempi meno banali.

Esempio 2.9:
Se f_0 è uguale a 1 sui numeri pari e 0 sui numeri dispari, allora $NFL(\mathbb{N}, f_0)$ è soddisfatta.

Dimostrazione. Il teorema di estensione di Kolmogorov ci permette di creare un processo stocastico f in cui le $f(i)$ (si sottointende la dipendenza da ω) siano indipendenti e uniformemente distribuite su $\{0, 1\}$. Allora per ogni m, e per ogni scelta di algoritmo a, $Y(f, m, a)$ è uniformemente distribuito su $\{0, 1\}^{\{1,\dots,m\}}$ (le funzioni che mandano 1, 2, ..., m nei valori 0 o 1).

Mostriamo che riusciamo a trovare una funzione deterministica $f_0 : \mathcal{X} \to \mathbb{R}$ tale che $NFL(\mathcal{X}, f_0)$ sia soddisfatta.
Scegliamo $f_0(i) = 1$ se i è pari, e $f_0(i) = 0$ se i è dispari. Consideriamo una permutazione casuale π su $\mathcal{X}$ scelta in questo modo (f è il processo appena costruito)

se $f(i) = 1$ allora $\pi(i) = 2 \times k(m)$ dove $k(m)$ è minimale tale che $2 \times k(m) \neq \pi(i)$ per $i < m$.

se $f(i) = 0$ allora $\pi(i) = 2 \times k(m) + 1$ dove $k(m)$ è minimale tale che $2 \times k(m) + 1 \neq \pi(i)$ per $i < m$.

Allora $f = f_0 \circ \pi$ e f_0 soddisfa la proprietà $NFL(\mathcal{X}, \pi, f_0)$. □

2.2 Proprietà GNFL

Nella sezione precedente abbiamo visto un esempio non banale tale che $NFL(\mathbb{N}, f_0)$ è soddisfatta (f_0 uguale a 1 sui numeri pari e 0 sui numeri dispari). Il fatto che *esistano* tali funzioni, che sono tali per cui $Y(f_0 \circ \pi, m, a_1)$ e $Y(f_0 \circ \pi, m, a_2)$ hanno la stessa

distribuzione, ci pone la questione se sia possibile *indebolire* la proprietà NFL. Vogliamo enunciare una proprietà più generale, che chiameremo *Generalized No Free Lunch* su $\mathcal{X}$ ($GNFL(\mathcal{X})$ nel seguito), dove $\mathcal{X}$ è un insieme numerabile, che sia valida non appena riusciamo a trovare una f_0 non banale tale che $NFL(\mathcal{X}, f_0)$ sia soddisfatta.

La definizione di GNFL non considera più le permutazioni casuali usate per definire NFL, ma lavora nell'ambiente più generale dei processi stocastici a mediana costante.

Definizione 2.10:
Sia $\mathcal{X}$ insieme numerabile e $\mathcal{Y} \subset \mathbb{R}$, $f : \mathcal{X} \to \mathcal{Y}$ funzione. Si dice che M_f è *mediana propria* di f se i due insiemi

$$\{x \in \mathcal{X}, f(x) < M_f\} \qquad \{x \in \mathcal{X}, f(x) > M_f\}$$

hanno la stessa misura e $\{x \in \mathcal{X}, f(x) = M_f\}$ è insieme finito.

Questo implica, nel nostro caso, che $\{x \in \mathcal{X}, f(x) < M_f\}$ e $\{x \in \mathcal{X}, f(x) > M_f\}$ debbano entrambi essere infiniti e quindi la proprietà appena enunciata è equivalente a chiedere che f non sia costante fuori da un insieme finito di $\mathcal{X}$.

Definizione 2.11:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità e siano $\mathcal{X}$ insieme numerabile, $\mathcal{Y} \subset \mathbb{R}$. Consideriamo un processo stocastico $f : \Omega \times \mathcal{X} \to \mathcal{Y}$. Si dice che f è a *mediana costante* se esiste $M_f \in \mathbb{R}$ tale che, per ogni $f(\omega, \cdot)$ con $\omega \in \Omega$, M_f sia mediana propria.

Si noti che se f_0 è una funzione con mediana propria e π una permutazione casuale su $\mathcal{X}$, allora $f = f_0 \circ \pi$ è un processo a mediana costante.

La condizione sulla mediana ci serve a impedire che processi 'banali' (per esempio un processo identicamente nullo, o un processo costante da un certo punto in poi) siano considerati per affermare che GNFL vale. Possiamo ora enunciare le proprietà GNFL.

Definizione 2.12:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità e siano $\mathcal{X}$ insieme numerabile, $\mathcal{Y} \subset \mathbb{R}$. Sia $f : \Omega \times \mathcal{X} \to \mathcal{Y}$ processo stocastico con mediana costante. Si dice che la proprietà $GNFL(\mathcal{X}, f)$ è soddisfatta se per ogni m e ogni coppia di algoritmi a_1 e a_2, $Y(f, m, a_1)$ e $Y(f, m, a_2)$ hanno la stessa distribuzione.

Definizione 2.13:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità e siano $\mathcal{X}$ insieme numerabile, $\mathcal{Y} \subset \mathbb{R}$. Si dice che la proprietà $GNFL(\mathcal{X})$ è soddisfatta se *esiste* processo stocastico $f : \Omega \times \mathcal{X} \to \mathcal{Y}$ a mediana costante, tale che $GNFL(\mathcal{X}, f)$ sia soddisfatta.

Rimarchiamo la differenza tra le due definizioni di $NFL(\mathcal{X})$ e $GNFL(\mathcal{X})$: nel primo caso, si richiede che $NFL(\mathcal{X}, f)$ sia soddisfatta da *tutte* le funzioni $f : \mathcal{X} \to \mathcal{Y}$ mentre nel secondo si richiede di trovare un solo processo f, con le ipotesi specificate, tale che $GNFL(\mathcal{X}, f)$ venga soddisfatta.

Mostriamo che $GNFL(\mathcal{X})$ è soddisfatta quando $NFL(\mathcal{X})$ lo è. Sia infatti f_0 una qualunque funzione che soddisfi $NFL(\mathcal{X}, f_0)$, e si consideri π permutazione casuale di $\mathcal{X}$: allora $f = f_0 \circ \pi$ è un processo stocastico a mediana costante che soddisfa $GNFL(\mathcal{X}, f)$.

Teorema 2.14:
Se $\mathcal{X} = \mathbb{N}$ allora $GNFL(\mathcal{X})$ è soddisfatta.

Dimostrazione. Basta considerare il processo stocastico f definito in 2.9. Tale processo è a mediana costante ed è tale per cui, per ogni m e per ogni scelta di algoritmo a, $Y(f, m, a)$ è uniformemente distribuito su $\{0, 1\}^{\{1,\dots,m\}}$. Di conseguenza $GNFL(\mathcal{X}, f)$ è soddisfatta, e quindi lo è anche $GNFL(\mathcal{X})$. □

Raccogliamo i risultati visti sinora nel seguente teorema, a cui ci riferiremo come *teorema NFL discreto*

Teorema 2.15:

- Se $\mathcal{X}$ è finito, allora $NFL(\mathcal{X})$ è valida.

- Se $\mathcal{X} = \mathbb{N}$, allora $NFL(\mathcal{X})$ NON è valida.

- Se $\mathcal{X} = \mathbb{N}$, esiste f_0 tale che $NFL(\mathcal{X}, f_0)$ è valida. Questa f_0 è uguale a 1 sui numeri pari e uguale a 0 sui numeri dispari. Quindi, se $\mathcal{X} = \mathbb{N}$, allora $GNFL(\mathcal{X})$ è valida.

2.3 Ottimizzazione su insiemi numerabili

Supponiamo di avere $\mathcal{X} = \mathbb{N}^d$ e $\mathcal{Y} \subset \mathbb{R}^+$. Se $f : \mathcal{X} \to \mathcal{Y}$, non siamo sicuri che $NFL(\mathcal{X}, f)$ valga. Quindi in generale non possiamo dire che due algoritmi di ottimizzazione forniscono prestazioni equivalenti.

Se limitiamo l'attenzione a problemi di massimo, ovvero trovare $x \in \mathcal{X}$ punto di massimo, allora l'equivalenza prestazionale è sicuramente valida sull'insieme delle funzioni positive che tendono a zero all'infinito come o più velocemente di $\frac{1}{||x||}$.

Teorema 2.16:

Siano $\mathcal{X} = \mathbb{N}^d$, $\mathcal{Y} \subset \mathbb{R}^+$. Supponiamo che, scelti a e $b \in \mathbb{R}^+$, $\mathcal{Y}$ contenga solo un numero finito di valori compresi tra a e b. Siano $c, C, \delta \in \mathbb{R}^+$, $\delta < min\{1, C\}$ e

$$F_{c,\delta,C} = \{f : \mathcal{X} \to \mathcal{Y} \quad \delta < f(x) \leq C \quad ||x|| \leq \frac{c}{\delta}, \qquad f(x) \leq \frac{c}{||x||} \quad ||x|| > \frac{c}{\delta}\}$$

$$\bar{\mathcal{X}}_{c,\delta} = \{x \in \mathcal{X}, ||x|| \leq \frac{c}{\delta}\}$$

da cui possiamo introdurre

$$\bar{F}_{c,\delta,C} = \{f_{|\bar{\mathcal{X}}_{c,\delta}}, f \in F_{c,\delta,C}\}$$

Allora:

1) i punti di massimo delle funzioni in $F_{c,\delta,C}$ stanno in $\bar{\mathcal{X}}_{c,\delta}$.

2) $\bar{F}_{c,\delta,C}$ è insieme finito C.U.P.

Quindi, se il problema è cercare punti di massimo di funzioni in $F_{c,\delta,C}$, ogni coppia di algoritmi a_1 e a_2 fornisce prestazioni equivalenti.

Dimostrazione. 1) Se $f \in F_{c,\delta,C}$ e $||x|| > \frac{c}{\delta}$ allora

$$f(x) < \frac{C}{\frac{c}{\delta}} = \delta < f(0,\ldots,0)$$

e $f(0,\ldots,0) > f(x)$; dunque è sufficiente cercare il massimo in $\bar{\mathcal{X}}_{c,\delta}$

2) $\bar{F}_{c,\delta,C}$ è l'insieme di tutte le funzioni da $\bar{\mathcal{X}}_{c,\delta}$ (insieme finito) in $\{y \in \mathcal{Y}, \delta < y \leq C\}$ ed è quindi finito e C.U.P. □

La definizione di $F_{c,\delta,C}$ può sembrare troppo restrittiva; tuttavia, con piccoli stratagemmi, è possibile mostrare l'equivalenza prestazionale per problemi di massimo sull'insieme di tutte le funzioni positive e limitate. Chiamiamo

$$F_{\delta,C} = \{f : \mathcal{X} \to \mathcal{Y} \quad \delta < f(x) \leq C\}$$

l'insieme di tutte le funzioni positive e limitate tali che per ogni $x \in \mathcal{X}$ abbiamo $\delta < f(x) \leq C$. Vale il seguente teorema.

Teorema 2.17:
Siano $\mathcal{X} = \mathbb{N}^d$, $\mathcal{Y} \subset \mathbb{R}^+$. Supponiamo che, scelti a e $b \in \mathbb{R}^+$, $\mathcal{Y}$ contenga solo un numero finito di valori compresi tra a e b. Siano $C, \delta \in \mathbb{R}^+$, $\delta < min\{1, C\}$ e $f \in F_{\delta,C}$. Allora esiste $\varepsilon_0 > 0$ tale che $x \in \mathcal{X}$ è punto di massimo di norma minima di $f(\cdot)$ se e solo se x è punto di massimo di norma minima di

$$\frac{f(\cdot)}{1 + \varepsilon||\cdot||}$$

per ogni $\varepsilon \in [0, \varepsilon_0]$.
è possibile scegliere $c > 0$ tale che $\frac{f(\cdot)}{1+\varepsilon||\cdot||} \in F_{c,\delta,C}$.

Quindi è possibile ricondurre la ricerca di un massimo di norma minima di una funzione $f \in F_{\delta,C}$ *alla ricerca di un massimo di norma minima di una funzione* $g \in F_{c,\delta,C}$.

Dimostrazione. Una volta trovato $\varepsilon_0 > 0$ che soddisfi la tesi, fissato $\varepsilon \in [0, \varepsilon_0]$, è sufficiente scegliere

$$c = \delta \left(\frac{C}{\delta\varepsilon} - \frac{1}{\varepsilon} \right)$$

Allora se $||x|| > \frac{c}{\delta}$ abbiamo

$$\frac{f(x)}{1+\varepsilon||x||} \leq \frac{C}{1+\varepsilon||x||} < \frac{C}{1+\frac{C}{\delta}-1} = \delta < \frac{f(0,\ldots,0)}{1+\varepsilon 0}$$

□

Il precedente teorema ci fornisce una strategia per la risoluzione di problemi di ottimizzazione riguardanti funzioni positive e limitate su domini numerabili.
1) Data $f(\cdot)$, individuiamo $\delta > 0$ e $C > 0$, $\delta < min\{1, C\}$, tali che sicuramente abbiamo

$$\delta < f(x) \leq C \qquad \forall x \in \mathcal{X}$$

2) Data $f(\cdot)$ si sceglie un parametro "di penalità" $\varepsilon > 0$ sufficientemente piccolo tale che, almeno in linea teorica, il massimo (di norma minima) di $\frac{f(\cdot)}{1+\varepsilon||\cdot||}$ è lo stesso di $f(\cdot)$.
3) Il punto di massimo di $\frac{f(\cdot)}{1+\varepsilon||\cdot||}$ è da cercarsi in una palla di centro l'origine e raggio

$$r = \frac{C}{\delta\varepsilon} - \frac{1}{\varepsilon}$$

questa contiene solo un numero finito di punti e, nel teorema 2.16 abbiamo mostrato l'equivalenza prestazionale di tutti gli algoritmi di ottimizzazione su $F_{c,\delta,C}$; quindi per cercare il punto di massimo possiamo procedere, per esempio, tramite *hill climbing* o anche tramite ricerca casuale.

Esempio 2.18:

Sia $\mathcal{X} = \mathbb{N}^2$. Consideriamo la funzione[1]

$$f(x_1, x_2) = \frac{3}{5} + max\left(sin\left(\frac{\pi}{2}x_1x_2\right), 0\right)$$

Vogliamo trovare il punto di massimo di norma minima su $\mathcal{X}$. Possiamo ottenere i seguenti limiti inferiore e superiore per f

$$\frac{1}{2} < f(x_1, x_2) \leq \frac{8}{5} \qquad \forall\ x \in \mathcal{X}$$

Quindi $f \in F_{1/2,8/5}$. Scegliamo il parametro di penalità $\varepsilon = \frac{1}{2}$. Questo è sufficientemente piccolo da evitare di "modificare il comportamento" di f.

Allora sappiamo che dobbiamo cercare il massimo nella palla di centro l'origine e raggio

$$r = \frac{C}{\delta\varepsilon} - \frac{1}{\varepsilon} = \frac{\frac{8}{5}}{\frac{1}{2}\cdot\frac{1}{2}} - \frac{1}{\frac{1}{2}} = \frac{24}{5} - 2 = \frac{14}{5}$$

E quindi dobbiamo controllare i punti: $(0,0), (0,1), (1,0), (1,1), (2,0), (0,2), (2,1), (1,2), (2,2)$.
Abbiamo:

$f((0,0))/(1+\frac{1}{2}\cdot 0) = \frac{3}{5}$
$f((0,1))/(1+\frac{1}{2}\cdot 1) = \frac{3}{5}\cdot\frac{2}{3} = \frac{2}{5}$
$f((1,0))/(1+\frac{1}{2}\cdot 1) = \frac{3}{5}\cdot\frac{2}{3} = \frac{2}{5}$
$f((1,1))/(1+\frac{1}{2}\cdot\sqrt{2}) \approx 0.9373$
$f((2,0))/(1+\frac{1}{2}\cdot 2) = \frac{3}{5}\cdot\frac{1}{2} = \frac{3}{10}$
$f((0,2))/(1+\frac{1}{2}\cdot 2) = \frac{3}{5}\cdot\frac{1}{2} = \frac{3}{10}$
$f((2,1))/(1+\frac{1}{2}\cdot\sqrt{5}) \approx 0.2833$
$f((1,2))/(1+\frac{1}{2}\cdot\sqrt{5}) \approx 0.2833$
$f((2,2))/(1+\frac{1}{2}\cdot\sqrt{8}) \approx 0.2485$

Dunque $(1,1)$ è il punto di massimo di f di norma minima.

[1] di cui sappiamo perfettamente quali sono i punti di massimo su $\mathbb{N}^2$

Capitolo 3

Teoremi per il caso continuo

L'unico esempio che ci resta da analizzare è quello nel quale $\mathcal{X}$ sia un insieme continuo. Vedremo che in questo caso non vale nè la $NFL(\mathcal{X})$ nè la $GNFL(\mathcal{X})$.

3.1 Proprietà NFL

La definizione della proprietà NFL si trasporta immediatamente dal caso numerabile, con la differenza che in questo caso $\mathcal{X} \subset \mathbb{R}^d$.

L'unica attenzione da porre riguarda le permutazioni casuali: non possiamo considerare tutte le permutazioni possibili su $\mathcal{X}$, ma solo quelle che preservano le misure con probabilità 1, cioè quelle per cui $\mu(A) = \mu(\pi^{-1}(\omega, A))$ per quasi ogni ω. Questo per evitare di trovare permutazioni analoghe a quelle mostrate nel seguente esempio e che violerebbero le proprietà che definiremo in seguito.

Esempio 3.1:
$\pi : [0,1] \to [0,1]$ definita da
$\pi(x) = \frac{4}{3}x \quad x \leq \frac{1}{2}$
$\pi(x) = \frac{2}{3} + \frac{1}{3}(x - \frac{1}{2}) \quad x > \frac{1}{2}$
è permutazione casuale che NON conserva le misure. Manda $[0, \frac{1}{2}]$ in un insieme di misura $\frac{2}{3}$ e $[\frac{1}{2}, 1]$ in un insieme di misura $\frac{1}{3}$.

Definizione 3.2:
Siano $\mathcal{X} \subset \mathbb{R}^d$ e $\mathcal{Y} \subset \mathbb{R}$ insiemi misurabili. Sia π permutazione casuale su $\mathcal{X}$ che preserva le misure con probabilità 1. Sia f_0 funzione misurabile da $\mathcal{X}$ in $\mathcal{Y}$. Si dice che

la proprietà $NFL(\mathcal{X}, \pi, f_0)$ è soddisfatta se, per ogni m e ogni coppia di algoritmi a_1 e a_2, $Y(f_0 \circ \pi, m, a_1)$ e $Y(f_0 \circ \pi, m, a_2)$ hanno la stessa distribuzione.

Definizione 3.3:
Siano $\mathcal{X} \subset \mathbb{R}^d$ e $\mathcal{Y} \subset \mathbb{R}$ insiemi misurabili. Sia f_0 funzione misurabile da $\mathcal{X}$ in $\mathcal{Y}$. Si dice che la proprietà $NFL(\mathcal{X}, f_0)$ è soddisfatta se esiste su $\mathcal{X}$ una permutazione casuale π che preserva le misure con probabilità 1 tale che $NFL(\mathcal{X}, \pi, f_0)$ sia soddisfatta.

Definizione 3.4:
Siano $\mathcal{X} \subset \mathbb{R}^d$ e $\mathcal{Y} \subset \mathbb{R}$ insiemi misurabili. Si dice che la proprietà $NFL(\mathcal{X})$ è soddisfatta se per ogni f_0 funzione misurabile da $\mathcal{X}$ in $\mathcal{Y}$ abbiamo che $NFL(\mathcal{X}, f_0)$ è soddisfatta.

Teorema 3.5:
La proprietà $NFL(\mathcal{X})$ non è soddisfatta con $\mathcal{X} = \mathbb{R}$.

Dimostrazione. Basta considerare una coppia di algoritmi a_1 e a_2 tali da visitare soltanto punti di $\mathbb{N}$, e poi sfruttare il fatto che NFL non vale su $\mathbb{N}$. □

3.2 Proprietà GNFL

Anche qui, le definizioni si trasportano facilmente dal caso numerabile. Richiamiamo la definizione di mediana propria.

Definizione 3.6:
Siano $\mathcal{X} \subset \mathbb{R}^d$ e $\mathcal{Y} \subset \mathbb{R}$ insiemi misurabili, $f : \mathcal{X} \to \mathcal{Y}$ funzione misurabile. Si dice che M_f è *mediana propria* di f se i due insiemi

$$\{x \in \mathcal{X}, f(x) < M_f\} \qquad \{x \in \mathcal{X}, f(x) > M_f\}$$

hanno misura uguale e $\{x \in \mathcal{X}, f(x) = M_f\}$ ha misura nulla.

Definizione 3.7:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità e siano $\mathcal{X} \subset \mathbb{R}^d$ e $\mathcal{Y} \subset \mathbb{R}$ insiemi misurabili. Consideriamo un processo stocastico misurabile $f : \Omega \times \mathcal{X} \to \mathcal{Y}$. Si dice che f è a *mediana costante* se esiste $M_f \in \mathbb{R}$ tale che, per ogni $f(\omega, \cdot)$ con $\omega \in \Omega$, M_f sia mediana propria.

Osservazione: Notiamo che la condizione che π preservi le misure gioca un ruolo essenziale. Infatti tale assunzione ci assicura che, presa una funzione f_0 che ammette mediana propria M_f, il processo $f_0 \circ \pi$ è a mediana costante (M_f è mediana propria per q.o. ω). Questo fa in modo che la $GNFL$ sia effettivamente una generalizzazione della NFL anche nel caso continuo.

Definizione 3.8:
Siano $(\Omega, \mathcal{A}, P)$ spazio di probabilità e siano $\mathcal{X}$ insieme numerabile, $\mathcal{Y} \subset \mathbb{R}$. Sia $f : \Omega \times \mathcal{X} \to \mathcal{Y}$ processo stocastico misurabile con mediana costante. Si dice che la proprietà $GNFL(\mathcal{X}, f)$ è soddisfatta se per ogni m e ogni coppia di algoritmi a_1 e a_2, $Y(f, m, a_1)$ e $Y(f, m, a_2)$ hanno la stessa distribuzione.

Definizione 3.9:
Siano $(\Omega, \mathcal{A}, P)$ spazio di probabilità e siano $\mathcal{X}$ insieme numerabile, $\mathcal{Y} \subset \mathbb{R}$. Si dice che la proprietà $GNFL(\mathcal{X})$ è soddisfatta se *esiste* processo stocastico misurabile $f : \Omega \times \mathcal{X} \to \mathcal{Y}$ a mediana costante, tale che $GNFL(\mathcal{X}, f)$ sia soddisfatta.

3.3 Free lunch!

Nel caso non numerabile, perdiamo ogni proprietà di *no free lunch*. Faremo due presentazioni di questo fatto. La prima è tratta dall'articolo [3] di Anne Auger e Olivier Teytaud, pubblicato su Algorithmica nel 2010; tuttavia, alcuni passaggi di quest'ultima sono discutibili e ha fatto nutrire dubbi ad alcuni matematici sulla validità della tesi proposta (si veda [4]). Abbiamo quindi dovuto elaborare una seconda presentazione,

che è più semplice, per confermare il fatto che, se $\mathcal{X}$ è insieme continuo, $GNFL(\mathcal{X})$ non è valida.

3.3.1 Prima presentazione

Lemma 3.10:
Supponiamo che g sia un processo stocastico misurabile, con valori in $\{0,1\}^{[0,1]}$, tale che le $g(\cdot,x)$ per $x \in [0,1]$ siano indipendenti e identicamente distribuite. Sia $J(\omega) = g(\omega,\cdot)^{-1}(1)$ ($\omega \in \Omega$) l'insieme dei punti $x \in [0,1]$ tali che $g(\omega,x) = 1$. Sia $\mu(J(\omega))$ la sua misura di Lebesgue. Allora esiste $p \in [0,1]$ tale che la variabile $\mu(J(\cdot))$ è q.c. uguale a p. Inoltre, $p = P(g(\cdot,x) = 1)$ per $x \in [0,1]$.

Dimostrazione. $\mu(J(\cdot))$ è una variabile aleatoria a valori in $[0,1]$. Indichiamo con $p = E[\mu(J(\cdot))]$ la sua media e $v = Var[\mu(J(\cdot))]$ la varianza. Mostriamo che $v = 0$.

Fissato x, la variabile $g(\cdot,x)$ è distribuita come (ad esempio) $g(\cdot,x/2)$ e $g(\cdot,(1+x)/2)$ e, quindi, ha la stessa distribuzione di $\frac{g(\cdot,x/2)+g(\cdot,(1+x)/2)}{2}$. Sfruttando l'indipendenza e le proprietà della varianza, questo ci permette di dire che

$$v = Var[\mu(J(\cdot))] = \frac{v+v}{4} = \frac{v}{2}$$

e concludiamo che $v = Var[\mu(J(\cdot))] = 0$. Questo ci permette di dire che $\mu(J(\cdot))$ è q.c. uguale alla sua media che è p.

Prendendo una variabile uniforme x su $[0,1]$, indipendente da g, e usando il teorema di Fubini, otteniamo che tale p è uguale alla probabilità che $g(\cdot,x)$ sia uguale a 1, $p = \mu(J(\cdot)) = P(g(\cdot,x) = 1)$. □

D'ora in avanti sottointenderemo la dipendenza da ω: $g(x) = g(\cdot,x)$.

Lemma 3.11:
Supponiamo che g sia un processo stocastico misurabile, con valori in $\{0,1\}^{[0,1]}$, tale che

le $g(x)$ per $x \in [0,1]$ siano indipendenti e identicamente distribuite. Sia $(a,b) \subset [0,1]$ intervallo aperto. Allora $(a,b) \cap g^{-1}(1)$ ha q.c. misura di Lebesgue $p * (b-a)$.

Dimostrazione. Basta applicare il lemma 3.10 a $\tilde{g}(x) = g(a + x * (b-a))$. Quasi certamente

$$\tilde{g}^{-1}(1) = \{x \in [0,1], a + x * (b-a) \in E\}$$

ha misura di Lebesgue p e dunque $g^{-1}(1) \cap (a,b)$ ha q.c. misura di Lebesgue $p * (b-a)$. □

Richiamiamo alcune nozioni di misurabilità.

Definizione 3.12:
La densità di un insieme $A \subset \mathbb{R}^d$ in un ε-intorno di un punto $x \in \mathbb{R}^d$, con $\varepsilon > 0$, è

$$d_\varepsilon(x, A) = \frac{\mu(A \cap B(x,\varepsilon))}{\mu(B(x,\varepsilon))}$$

la densità di un insieme A in un punto x è

$$d(x, A) = \lim_{\varepsilon' \to 0} d_{\varepsilon'}(x, A)$$

Teorema 3.13:
(teorema di densità di Lebesgue) Sia A insieme misurabile di $\mathbb{R}^d$. Allora, q.o. $x \in A$ soddisfa $d(x, A) = 1$.

Lemma 3.14:
Supponiamo che g sia un processo stocastico misurabile, con valori in $\{0,1\}^{[0,1]}$, tale che le $g(x)$ per $x \in [0,1]$ siano indipendenti e identicamente distribuite. Allora esiste un intervallo aperto $(a,b) \subset [0,1]$ di misura non nulla, e $p' > p$ tale che $(a,b) \cap g^{-1}(1)$ ha misura di Lebesgue $p' * (b-a)$ con probabilità > 0.

Dimostrazione. Possiamo sfruttare il teorema di densità di Lebesgue per dire che esiste un intervallo $(a, b) \subset [0, 1]$ di misura non nulla che ha probabilità > 0 di soddisfare

$$\mu((a, b) \cap g^{-1}(1)) \geq \left(\frac{1+p}{2}\right) \mu((a, b))$$

visto che $\mu((a, b)) = (b - a)$ allora, individuato $p' > 0$ tale che

$$\mu((a, b) \cap g^{-1}(1)) = p' * (b - a)$$

abbiamo che

$$p' \geq \frac{1+p}{2} > p$$

e questo conclude la dimostrazione. □

Osservazione: La dimostrazione precedente, riportata dall'articolo [4] di Anne Auger e Olivier Teytaud, non è completa: sicuramente andava spiegato in modo più dettagliato l'uso del teorema di densità di Lebesgue, nato per insiemi misurabili (deterministici) e riportato in precedenza, per dire che esiste l'intervallo $(a, b) \subset [0, 1]$ in questione.

Teorema 3.15:
Supponiamo che g sia un processo stocastico misurabile, con valori in $\{0, 1\}^{[0,1]}$, tale che le $g(x)$ per $x \in [0, 1]$ siano identicamente distribuite. Allora le $g(x)$ non sono indipendenti.

Dimostrazione. Supponiamo, per assurdo, che le $g(x)$ siano indipendenti. Se $p = P(g(x) = 1)$, allora per il lemma 3.10 $g^{-1}(1)$ ($\subset [0, 1]$) ha q.c. misura di Lebesgue p, e per 3.14 esiste q.c. un intervallo aperto $(a, b) \subset [0, 1]$ tale che $g^{-1}(1)$ abbia densità $p' > p$ in (a, b), cioè $\mu((a, b) \cap g^{-1}(1)) = p' * (b - a)$, con probabilità non nulla.

Ma per 3.11 deve essere $\mu((a, b) \cap g^{-1}(1)) = p * (b - a)$ q.c., assurdo! □

Teorema 3.16:
Se $\mathcal{X} = [0,1]$*, allora* $GNFL(\mathcal{X})$ *non è soddisfatta.*

Dimostrazione. Consideriamo infatti un processo stocastico, a mediana costante, f su $\mathbb{R}^{[0,1]}$, supponendo per assurdo che soddisfi $GNFL(\mathcal{X}, f)$.
Allora il processo stocastico g su $\{0,1\}^{[0,1]}$ che è tale per cui

$$g(x) = 0 \quad se \quad f(x) < M_f$$

$$g(x) = 1 \quad se \quad f(x) > M_f$$

(dove M_f è mediana propria di f) è uniformemente distribuito (vale 0 con probabilità $\frac{1}{2}$ e 1 con la stessa probabilità).

Inoltre, se valesse $GNFL(\mathcal{X}, f)$, il vettore $(g(x_1), \ldots, g(x_n))$ avrebbe la stessa distribuzione di una ricerca casuale e, in quest'ultima, sappiamo che $(g(x_1), \ldots, g(x_n))$ è uniformemente distribuito su $\{0,1\}^n$, quindi le $g(x)$ dovrebbero essere anche indipendenti.

Tuttavia, non possiamo avere processi stocastici che siano contemporaneamente uniformemente distribuiti e indipendenti su $\mathcal{X} = [0,1]$, come dimostrato dal precedente teorema. □

Osservazione: Ci pone dubbi l'uso dell'algoritmo di ricerca casuale. Quest'ultimo aggiunge una fonte di aleatorietà, che non dipende dallo spazio di probabilità di f. Possiamo infatti dire che f dipende da uno spazio di probabilità $(\Omega, \mathcal{A}, P)$ e l'algoritmo di ricerca casuale dipende da un altro spazio $(\Omega_1, \mathcal{B}, Q)$.

L'insieme dei punti visitati, $(x_1, \ldots, x_n)$, è quindi aleatorio, e dipende da $\omega_1 \in \Omega_1$, dunque dovremmo in realtà scrivere $(x_1(\omega_1), \ldots, x_n(\omega_1))$. Allora dire che il vettore $(g(x_1(\omega_1)), \ldots, g(x_n(\omega_1)))$, ottenuto applicando l'algoritmo di ricerca casuale $\tilde{a}$, è uniformemente distribuito su $\{0,1\}^n$ non implica necessariamente che le $g(x)$ siano indipendenti su Ω, perchè tale vettore dipende dallo spazio $\Omega \times \Omega_1$.

3.3.2 Seconda presentazione

Vogliamo analizzare alcune caratteristiche dei processi continui che possono soddisfare GNFL, arrivando ad alcune contraddizioni che ci dicono che un tale processo non esiste.

Teorema 3.17:
Sia $(\Omega, \mathcal{A}, P)$ *spazio di probabilità,* $\mathcal{X} = [0,1]$, $f : \Omega \times \mathcal{X} \to \mathbb{R}$ *processo stocastico a mediana costante. Allora* $GNFL(\mathcal{X}, f)$ *può essere soddisfatta solo se le* $f(x)$ *sono identicamente distribuite e* $Cov(f(x), f(y))$ *è costante per ogni* x, $y \in \mathcal{X}$, $x \neq y$.

Dimostrazione. Se $GNFL(\mathcal{X}, f)$ è soddisfatta, allora i vettori $Y(f, m, a_1)$ e $Y(f, m, a_2)$ devono avere la stessa distribuzione per ogni m e per ogni coppia di algoritmi a_1 e a_2.

Scegliamo $m = 1$. Dati due punti x e y, sia a_x un algoritmo (deterministico) che come punto iniziale sceglie x e a_y un algoritmo che come punto iniziale sceglie y. La precedente ci dice allora che $f(x)$ ha la stessa distribuzione di $f(y)$ per ogni x, $y \in \mathcal{X}$.

Ora fissiamo $m = 2$. Date due coppie di punti (x_{r_1}, x_{r_2}) e (x_{s_1}, x_{s_2}), sia a_1 un algoritmo (deterministico) che come primi due punti visita la prima coppia e a_2 un algoritmo che visita come primi due punti la seconda coppia. Allora i vettori $(f(x_{r_1}), f(x_{r_2}))$ e $(f(x_{s_1}), f(x_{s_2}))$ devono avere la stessa distribuzione per ogni x_{r_1}, x_{r_2}, x_{s_1}, $x_{s_2} \in \mathcal{X}$, $x_{r_1} \neq x_{r_2}$, $x_{s_1} \neq x_{s_2}$. Questo ci dice che, dato che x_{r_1} ha la stessa distribuzione di x_{s_1}, e x_{r_2} ha la stessa distribuzione di x_{s_2}, debba essere anche

$$Cov(f(x_{r_1}), f(x_{r_2})) = Cov(f(x_{s_1}), f(x_{s_2}))$$

ossia che le covarianze debbano essere tutte uguali. □

Proposizione 3.18:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $\mathcal{X} = [0,1]$, $f : \Omega \times \mathcal{X} \to \mathbb{R}$ processo stocastico a mediana costante in cui le $f(x)$ sono identicamente distribuite. Sia μ la (comune)

media delle $f(x)$ e σ^2 la varianza. Allora $GNFL(\mathcal{X}, f)$ è soddisfatta se e solo se $GNFL(\mathcal{X}, (f-\mu)/\sigma)$ è soddisfatta.

Le due precedenti asserzioni ci permettono di restringere la ricerca di un processo che soddisfa $GNFL(\mathcal{X}, f)$ a processi in cui le $f(x)$ siano identicamente distribuite, di media nulla e varianza uguale a 1, in cui $Cov(f(x), f(y)) = \rho$ per ogni $x \neq y$.

Indicheremo con ρ la "comune" covarianza del processo f. Abbiamo tre possibilità:
- $\rho < 0$
- $\rho > 0$
- $\rho = 0$

Mostriamo che non può essere $\rho < 0$.

Proposizione 3.19:
Siano $X_1, \ldots, X_n$ $(n \geq 2)$ variabili aleatorie identicamente distribuite, di media nulla e varianza uguale a 1. Sia $\rho = Cov(X_i, X_j)$ $(1 \leq i \neq j \leq n)$ la comune covarianza delle variabili. Allora

$$\rho \geq -\frac{1}{n-1}$$

Dimostrazione. Se facciamo la somma delle variabili, abbiamo

$$Var(\sum_{i=1}^{n} X_i) = \sum_{i=1}^{n} Var(X_i) + \sum_{i=1}^{n-1} \sum_{j>=(i+1)}^{n} Cov(X_i, X_j)$$

Visto che $Var(\sum_{i=1}^{n} X_i) \geq 0$ abbiamo

$$-\sum_{i=1}^{n-1} \sum_{j>=(i+1)}^{n} Cov(X_i, X_j) \leq \sum_{i=1}^{n} Var(X_i) = n$$

ma $\sum_{i=1}^{n-1} \sum_{j>=(i+1)}^{n} Cov(X_i, X_j) = n(n-1)\rho$, quindi

$$\rho \geq -\frac{1}{n-1}$$

□

Da cui segue

Proposizione 3.20:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità. $\mathcal{X} = [0,1]$, $f : \Omega \times \mathcal{X} \to \mathbb{R}$ processo stocastico a mediana costante che soddisfa $GNFL(\mathcal{X}, f)$. Se indichiamo con ρ la "comune" covarianza del processo, allora non può essere $\rho < 0$.

Non può essere neanche $\rho = 0$. Infatti non possiamo avere processi che siano identicamente distribuiti e non correlati, come mostrato dal seguente teorema

Teorema 3.21:
Supponiamo che f sia un processo stocastico misurabile, con valori in $\mathbb{R}^{[0,1]}$, tale che le $f(x)$ per $x \in [0,1]$ siano identicamente distribuite. Se le $f(x)$ sono non correlate allora f è costante.

Dimostrazione. Trattiamo prima il caso in cui il processo è limitato e centrato. Allora il risultato può essere facilmente esteso al caso generale.

Primo caso: assumiamo esista una costante $m \in \mathbb{N}$ tale che per ogni x, $|f(x)| < m$ e $E[f(x)] = 0$.

Dato che $f(x)$ è limitato $\forall\ x \in [0,1]$, ha senso considerare l'integrale di Lebesgue

$$F(x) = \int_0^x f(t)dt \qquad x \in [0,1]$$

Per le proprietà dell'integrale, le variabili $F(x)$ sono q.c. limitate (per esempio, da m) e quindi $E[F(x)^2]$ è finito. Possiamo vedere che è uguale a 0:

$$E[F(x)^2] = E[\int_0^x f(s)ds \cdot \int_0^x f(r)dr] = E[\int_0^x \int_0^x f(s)f(r)dsdr] =$$

$$= \int_0^x \int_0^x E[f(s)f(r)]dsdr =$$

$$= \int_0^x \int_0^x E[f(s)^2] \cdot 1_{\{s=r\}} dsdr + \int_0^x \int_0^x E[f(s)f(r)] \cdot 1_{\{s \neq r\}} dsdr$$

dove lo scambio tra integrale e valore atteso è giustificato dalla limitatezza di tutte le funzioni coinvolte, che ci permette di applicare il teorema di Fubini. Ora, il primo integrale è uguale a zero perchè stiamo integrando sopra la linea $\{s = r\}$, che ha misura di Lebesgue nulla. Il secondo, grazie all'ipotesi che $f(s)$ e $f(r)$, se $s \neq r$, non sono correlate, è uguale a

$$\int_0^x \int_0^x E[f(s)]E[f(r)] \cdot 1_{\{s \neq r\}} dsdr$$

ed essendo $f(r)$ e $f(s)$ v.a. centrate, vale 0. Otteniamo allora che $E[F(x)^2] = 0$ per tutti gli $x \in [0, 1]$, il che implica che q.c. $F(x) = 0$. Questo ci dice che l'integrando, $f(x)$, è anch'esso q.c. nullo. Quindi $f(x) = 0$ q.c.

Caso generale:

Sia $m \in \mathbb{N}$ e definiamo:

$$\tilde{f}_m(x) = f(x) \cdot 1_{|f(x)|<m}$$

e

$$\bar{f}_m(x) = \tilde{f}_m(x) - E[\tilde{f}_m(x)]$$

applicando il primo caso abbiamo che per ogni m, $\bar{f}_m(x) = 0$ q.c., di conseguenza $\tilde{f}_m(x) = a(t, m)$ q.c., dove $a(t, m)$ è il valore atteso di $\tilde{f}_m(x)$. Ora, per $m \to \infty$, $\tilde{f}_m(x)$ è q.c. uguale a $f(x)$, quindi $f(x)$ è q.c. uguale ad $a(t, m)$ per $m \to \infty$, quindi è q.c. uguale alla sua media, e la tesi è soddisfatta. □

Non può essere neanche $\rho > 0$. La ragione principale è racchiusa in questa proposizione

Proposizione 3.22:

Siano $X_1, \ldots, X_n$ $(n > 2)$ variabili aleatorie identicamente distribuite, di media nulla

e varianza uguale a 1. Sia $\rho = Cov(X_i, X_j)$ $(1 \leq i \neq j \leq n)$ la comune covarianza delle variabili.

Se $n \to \infty$, la covarianza condizionale tra X_n e X_{n-1}, dati $X_1, \ldots, X_{n-2}$, tende a 0; ossia

$$\lim_{n\to\infty} Cov(X_n, X_{n-1}|X_1, \ldots, X_{n-2}) = 0$$

Limitiamo la nostra trattazione ai processi gaussiani, in cui è semplice effettuare il condizionamento.

Supponiamo di avere una distribuzione normale multivariata con media 0 e matrice di covarianza Σ. La densità congiunta è data da

$$f(x) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$$

Scrivendo x in due componenti $\begin{bmatrix} a \\ b \end{bmatrix}$, siamo interessati nello trovare la distribuzione condizionale $p(a|b)$.

Separiamo le componenti della matrice di covarianza Σ in una matrice a blocchi $\begin{bmatrix} A & C^T \\ C & B \end{bmatrix}$, tale che A corrisponda alla covarianza per a, e B corrisponda alla covarianza per b, e C contiene i termini incrociati.

Ora notiamo che la densità congiunta può essere scritta come

$$p(a, b) = c_0 exp\left(-\frac{1}{2}\begin{bmatrix} a \\ b \end{bmatrix}^T \begin{bmatrix} A & C^T \\ C & B \end{bmatrix}^{-1} \begin{bmatrix} a \\ b \end{bmatrix}\right)$$

(dove $c_0 = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}}$) Usando il complemento di Schur possiamo scrivere

$$\begin{bmatrix} A & C^T \\ C & B \end{bmatrix} = \begin{bmatrix} I & 0 \\ -B^{-1}C & I \end{bmatrix} \begin{bmatrix} (A - C^T B^{-1} C)^{-1} & 0 \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} I & -C^T B^{-1} \\ 0 & I \end{bmatrix}$$

Inserendo quest'ultima nella densità congiunta (e semplificando, moltiplicando i vettori per le due matrici "esterne") otteniamo

$$p(a,b) = c_0 exp\left(-\frac{1}{2} \begin{bmatrix} a - C^T B^{-1} b \\ b \end{bmatrix}^T \begin{bmatrix} (A - C^T B^{-1} C)^{-1} & 0 \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} a - C^T B^{-1} b \\ b \end{bmatrix} \right)$$

Usando il fatto che la matrice centrale è diagonale a blocchi, abbiamo

$$p(a,b) = c_0 exp\left(-\frac{1}{2} (a - C^T B^{-1} b)^T (A - C^T B^{-1} C)^{-1} (a - C^T B^{-1} b) \right) exp\left(-\frac{1}{2} b^T B^{-1} b \right)$$

Condizionando su b, il secondo termine esponenziale è una costante e otteniamo

$$p(a|b) \sim N(C^T B^{-1} b, (A - C^T B^{-1} C)).$$

Dall'algebra lineare, sappiamo che la matrice $A - C^T B^{-1} C$ può anche essere ottenuta invertendo Σ, ed eliminando quindi le righe e le colonne relative a b e invertendo poi ciò che si è ottenuto.

Supponiamo ora di avere un processo gaussiano di n variabili e di conoscere la matrice di covarianza Σ del processo gaussiano, e di voler effettuare il condizionamento relativamente alle prime $n-2$ variabili. Supponiamo cioè di voler trovare i valori di $Var(X_{n-1}|X_1, \ldots, X_{n-2})$, $Var(X_n|X_1, \ldots, X_{n-2})$ e $Cov(X_{n-1}, X_n|X_1, \ldots, X_{n-2})$.

Per trovare i valori richiesti, si può seguire il seguente procedimento:
1) calcolare l'inversa di Σ, $B = \Sigma^{-1}$.
2) eliminare le prime $n-2$ righe e colonne di B, ottenendo una matrice 2×2 che

chiamiamo C.

3) invertire la matrice C, trovando la matrice $D = C^{-1}$.

Osservazione: Le matrici B e C in questo algoritmo sono diverse da quelle definite in precedenza.

La matrice D ha nella posizione $(1,1)$ la $Var(X_{n-1}|X_1,\ldots,X_{n-2})$, nella posizione $(2,2)$ la $Var(X_n|X_1,\ldots,X_{n-2})$, nelle posizioni $(1,2)$ e $(2,1)$ la $Cov(X_{n-1},X_n|X_1,\ldots,X_{n-2})$.

Applichiamo il precedente procedimento per mostrare che se $n \to \infty$ la matrice D è diagonale, ossia le covarianze condizionali tendono a 0.

La matrice Σ del processo è nella forma

$$\Sigma_{i,i} = 1 \qquad \Sigma_{i,j} = \rho \quad (i \neq j)$$

Per esempio, se $n = 5$:

$$\Sigma = \begin{bmatrix} 1 & \rho & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho & \rho \\ \rho & \rho & 1 & \rho & \rho \\ \rho & \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & \rho & 1 \end{bmatrix}$$

La matrice inversa di Σ, che abbiamo chiamato B, è nella forma

$$B_{i,i} = a \qquad B_{i,j} = b \quad (i \neq j)$$

dove a e b sono tali da soddisfare le due equazioni

$$a + (n-1)\rho b = 1$$

$$b + \rho a + (n-2)\rho b = 0$$

Risolvendo le due precedenti equazioni, si trova che

$$a = \frac{1 + (n-2)\rho}{1 - \rho^2(n-1) + (n-2)\rho^2}$$

$$b = \frac{-\rho}{1 - \rho^2(n-1) + (n-2)\rho}$$

La matrice C, che è ottenuta da B togliendo le prime $n-2$ righe e colonne, è dunque una matrice 2×2:

$$C = \frac{1}{1 - \rho^2(n-1) + (n-2)\rho} \begin{bmatrix} 1 + (n-2)\rho & -\rho \\ -\rho & 1 + (n-2)\rho \end{bmatrix}$$

e allora D, inversa di C, è nella forma

$$D = \frac{1 - \rho^2(n-1) + (n-2)\rho}{(1 + (n-2)\rho)^2 - \rho^2} \begin{bmatrix} 1 + (n-2)\rho & \rho \\ \rho & 1 + (n-2)\rho \end{bmatrix}$$

è evidente come, per $n \to \infty$, gli elementi fuori dalla diagonale tendano a 0. Quindi per i processi gaussiani che abbiamo considerato vale che

$$\lim_{n \to \infty} Cov(X_n, X_{n-1} | X_1, \ldots, X_{n-2}) = 0$$

Abbiamo quindi dimostrato la seguente proposizione nel caso in cui f sia gaussiano:

Proposizione 3.23:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità. $\mathcal{X} = [0,1]$, $f : \Omega \times \mathcal{X} \to \mathbb{R}$ processo stocastico a mediana costante che soddisfa $GNFL(\mathcal{X}, f)$. Sia $\rho > 0$ la "comune" covarianza del processo. Allora, dati i valori del processo su $\mathbb{Q} \cap [0,1]$, le $f(i)$ negli altri punti risultano essere tra loro (condizionatamente) non correlate.

Da questa segue immediatamente che

Teorema 3.24:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità. $\mathcal{X} = [0,1]$. Sia $f : \Omega \times \mathcal{X} \to \mathbb{R}$ processo stocastico misurabile in cui le $f(x)$ sono identicamente distribuite. Sia $\rho > 0$ la "comune" covarianza del processo. Allora $GNFL(\mathcal{X}, f)$ non è soddisfatta.

Dimostrazione. Consideriamo il processo $\tilde{f} : \Omega \times \mathcal{X} \to \mathbb{R}$ definito da:

$$\tilde{f}(\cdot, x) = 0 \qquad x \in [0,1] \cap \mathbb{Q}$$

$$\tilde{f}(\cdot, x) = f(\cdot, x) | \{f(\cdot, y)\}_{y \in \mathbb{Q} \cap [0,1]} \qquad x \in [0,1] \backslash \mathbb{Q}$$

dato che f è processo in cui le $f(\cdot, x)$ sono identicamente distribuite, allora $\tilde{f}$, escludendo un'insieme trascurabile, è un processo in cui le $\tilde{f}(\cdot, x)$ sono identicamente distribuite, e la covarianza comune delle variabili di $\tilde{f}$ è 0. Per il teorema 3.21, sappiamo che $\tilde{f}$ è quasi certamente costante. Ma allora anche f è quasi certamente costante, dato che $\tilde{f}$ è stato ottenuto condizionando f sui valori assunti (da f) su $\mathbb{Q} \cap [0,1]$; quindi viola la condizione sulla mediana e non può soddisfare $GNFL$. □

La dimostrazione che abbiamo fatto riguarda solo processi gaussiani. Il caso gaussiano è piu' semplice perchè le $Var(X_n | X_1, \ldots, X_{n-2})$ e $Cov(X_n, X_{n-1} | X_1, \ldots, X_{n-2})$ (che sono variabili aleatorie) hanno varianza nulla:

$$Var(Var(X_n | X_1, \ldots, X_{n-2})) = 0$$

$$Var(Cov(X_n, X_{n-1} | X_1, \ldots, X_{n-2})) = 0$$

Questo grazie al fatto che la legge condizionale rimane comunque congiuntamente gaussiana.

In generale queste varianze non sono nulle. Dunque, cosa possiamo dire nel caso generale? Possiamo trovare una stima approssimante che ci faccia comunque capire che le covarianze condizionali tendono a 0.

Siano $X_1, \ldots, X_n$ $(n \geq 2)$ variabili aleatorie identicamente distribuite, di media nulla e varianza uguale a 1. Sia $\rho = Cov(X_i, X_j)$ $(1 \leq i \neq j \leq n)$ la comune covarianza delle variabili. Indichiamo con:

$$s_n = E[Var(X_n|X_1, \ldots, X_{n-1})] \qquad \rho_n = E[Cov(X_{n+1}, X_n|X_1, \ldots, X_{n-1})]$$

dove $s_1 = 1$ e $\rho_1 = \rho$.

Possiamo approssimare computazionalmente le s_n e le ρ_n nel modo seguente:

$$s_{n+1} = s_n - \rho_n^2 s_n$$

$$\rho_{n+1} = \rho_n - \rho_n^2 s_n$$

Per dare un'idea del perchè la precedente è valida, si può pensare a tre variabili aleatorie X_1, X_2, X_3 identicamente distribuite, di media nulla e varianza uguale a 1. Sia $\rho = Cov(X_i, X_j)$ $(1 \leq i \neq j \leq 3)$ la comune covarianza delle variabili. Allora per i lemmi della varianza e della covarianza totale abbiamo

$$Var(X_2) = E[Var(X_2|X_1)] + Var(E[X_2|X_1])$$

$$Cov(X_2, X_3) = E[Cov(X_2, X_3|X_1)] + Cov(E[X_2|X_1], E[X_3|X_1])$$

quindi:

$$s_2 = E[Var(X_2|X_1)] = Var(X_2) - Var(E[X_2|X_1]) = 1 - Var(E[X_2|X_1]) = s_1 - Var(E[X_2|X_1])$$

$$\rho_2 = E[Cov(X_2, X_3|X_1)] = \rho - Cov(E[X_2|X_1], E[X_3|X_1]) = \rho_1 - Cov(E[X_2|X_1], E[X_3|X_1])$$

Il problema è approssimare i termini $E[X_2|X_1]$ e $E[X_3|X_1]$. Sappiamo che il miglior

stimatore lineare di $E[X_2|X_1]$ è ρX_1; quindi sarebbe lecito stimare

$$Var(E[X_2|X_1]) \approx \rho^2 = \rho_1^2 s_1$$

$$Cov(E[X_2|X_1], E[X_3|X_1]) \approx \rho^2 = \rho_1^2 s_1$$

Quindi

$$s_2 = s_1 - \rho_1^2 s_1$$

$$\rho_2 = \rho_1 - \rho_1^2 s_1$$

La stima degli s_n e dei ρ_n risulta in generale essere sufficientemente buona per delineare il comportamento del condizionamento.

Se prendiamo un processo gaussiano e fissiamo ad esempio $\rho = 0.5$, abbiamo il seguente confronto tra la stima e il condizionamento reale:

N	s_N	$E[Var(X_N\|X_1,\ldots,X_{N-1})]$	c_N	$E[Cov(X_N, X_{N+1}\|X_1,\ldots,X_{N-1})]$
10	0.59463	0.55000	0.094635	0.055556
50	0.53008	0.51000	0.030082	0.010204
100	0.51676	0.50500	0.016761	0.005050
200	0.50899	0.50250	0.008993	0.002512
500	0.50379	0.50100	0.003797	0.001002

Se consideriamo la ricorsione:

$$s_{n+1} = s_n - \rho_n^2 s_n$$

$$\rho_{n+1} = \rho_n - \rho_n^2 s_n$$

allora abbiamo $\lim_{n\to\infty} \rho_n = 0$.

Infatti, è immediato notare che $s_n \leq 1 \ \forall \ n$ e le successioni s_n e ρ_n sono $\geq 0 \ \forall n$. In particolare, la successione s_n è > 0 per ogni n. Infatti, $s_1 > \rho_1$ e ognuno dei termini successivi si trova togliendo la stessa quantità per entrambe le successioni.

Riusciamo inoltre a trovare $0 < C < 1$ tale che $s_n \geq C$ per ogni n (basta prendere $C = 1 - \rho$).

Allora la successione

$$c_1 = 1 \qquad c_{n+1} = c_n - Cc_n^2$$

è tale che $\rho_n \leq c_n$

La successione c_n è tale che $\lim_{n\to\infty} c_n = 0$, quindi $\lim_{n\to\infty} \rho_n = 0$ e le covarianze condizionali tendono quindi a zero.

Per concludere e riassumere: se $\mathcal{X} = [0, 1]$ e se f è processo stocastico a mediana costante, allora può soddisfare $GNFL(\mathcal{X}, f)$ solo se le sue variabili sono identicamente distribuite e hanno covarianze tutte uguali. Se chiamiamo ρ tale covarianza, non può essere $\rho < 0$ (vedere la proposizione 3.20) nè $\rho = 0$ (vedere 3.21), nè $\rho > 0$ (vedere la proposizione 3.24); quindi semplicemente non esiste.

Conclusioni

Nel primo capitolo abbiamo visto e dimostrato il teorema NFL finito, che ci dice che la proprietà di *no free lunch* vale per tutti gli algoritmi di risoluzione di problemi di ottimizzazione su insiemi C.U.P. finiti, cioè per ogni coppia di algoritmi a_1 e a_2, per ogni m, abbiamo che $Y(f, m, a_1)$ e $Y(f, m, a_2)$ hanno la stessa distribuzione, per f variabile aleatoria uniformemente distribuita (sull'insieme C.U.P.)

Nel secondo capitolo abbiamo visto come questa proprietà si generalizza per insiemi numerabili. Per una funzione f_0 e una permutazione casuale π, definiamo $NFL(\mathcal{X}, \pi, f_0)$ come il fatto che per ogni intero m e ogni coppia di algoritmi a_1 e a_2, $Y(f_0 \circ \pi, m, a_1)$ e $Y(f_0 \circ \pi, m, a_2)$ seguono la stessa distribuzione. Per $\mathcal{X} = \mathbb{N}$ siamo in grado di trovare una funzione non banale f_0 tale che esiste una permutazione casuale π che fa valere $NFL(\mathbb{N}, \pi, f_0)$.

Abbiamo definito una forma più debole di NFL, GNFL. Per un processo stocastico a mediana costante f definiamo $GNFL(\mathcal{X}, f)$ come il fatto che per ogni intero m e ogni coppia di algoritmi di ottimizzazione a_1 e a_2, $Y(f, m, a_1)$ e $Y(f, m, a_2)$ seguono la stessa distribuzione.

I teoremi di *no free lunch* sono veri per tutti i casi finiti, e abbiamo mostrato come siano "genericamente" veri quando il dominio è numerabile, mentre sono del tutto falsi quando andiamo in insiemi non numerabili (terzo capitolo).

Le nostre conclusioni possono essere riassunte da:

Dominio $\mathcal{X}$	Finito	Numerabile	Non numerabile
$\exists f_0$, $NFL(\mathcal{X}, f_0)$ è soddisfatta	S	S	N
$\forall f_0$, $NFL(\mathcal{X}, f_0)$ è soddisfatta	S	N	N
$\exists f$, $GNFL(\mathcal{X}, f)$ é soddisfatta	S	S	N

Appendice A

Teorema di estensione di Kolmogorov

Un teorema che è risultato essenziale per le nostre conclusioni è il teorema di estensione di Kolmogorov, che qui brevemente richiamiamo.

Consideriamo $\mathcal{T}$ insieme di tempi, $\{X_t\}_{t\in\mathcal{T}}$ processo stocastico definito su $(\Omega, \mathcal{A}, P)$ spazio di probabilità, a valori in $\mathbb{R}$. Per ogni $k \in \mathbb{N}$ e ogni successione finita di tempi $t_1, \ldots, t_k \in \mathcal{T}$, possiamo associare una misura di probabilità $\mu_{t_1,\ldots,t_k}$ su $\mathbb{R}^k$ definita come

$$\mu_{t_1,\ldots,t_k}(F_1, \ldots, F_k) = P(X_{t_1} \in F_1, \ldots, X_{t_k} \in F_k)$$

Tali misure prendono il nome di *distribuzioni finito-dimensionali* e sono tali da rispettare le seguenti due condizioni di coerenza:
1) $\mu_{t_1,\ldots,t_k}(F_1 \times \ldots \times F_k) = \mu_{t_{\pi(1)}\ldots t_{\pi(k)}}(F_{\pi(1)} \times \ldots \times F_{\pi(k)})$
per ogni permutazione π degli indici $1, \ldots, k$.
2) $\mu_{t_1,\ldots,t_k,t}(F_1 \times \ldots \times F_k \times \mathbb{R}) = \mu_{t_1,\ldots,t_k}(F_1 \times \ldots \times F_k)$
per $t \in \mathcal{T}$.

Vogliamo vedere se possiamo fare il cammino inverso, cioè se date delle distribuzioni finito dimensionali è possibile trovare un processo stocastico che le soddisfi. A tal scopo, enunciamo

Teorema A.1:
Sia $\mathcal{T}$ insieme dei tempi, $D = \{\mu_{t_1,\ldots,t_k},\ k \in \mathbb{N},\ t_1, \ldots, t_k \in \mathcal{T}\}$, dove $\mu_{t_1,\ldots,t_k}$ è misura di probabilità su $\mathbb{R}^k$. Supponiamo che D soddisfi le suddette condizioni di coerenza. Allora esiste uno spazio di probabilità $(\Omega, \mathcal{A}, P)$ e un processo stocastico $\{X_t\}_{t\in T}$ tale che gli elementi di D siano distribuzioni finito-dimensionali di X.

Esempio A.2:

Sia $\mathcal{T} = \mathbb{N}$. Vogliamo costruire un processo stocastico $\{X_n\}_{n\in\mathbb{N}}$ tale che le variabili X_n siano indipendenti e identicamente distribuite.

Sia $E_k = \{v \in \mathbb{R}^k, v(i) \in \{0,1\}\ \forall i\}$ (ad esempio, $E_2 = \{(0,0),(0,1),(1,0),(1,1)\}$). Allora definiamo come insieme D l'insieme di tutte le misure per cui $\mu_{1,\dots,k}(e_k) = \frac{1}{2^k}$, $\forall e_k \in E_k$, $\forall k \in \mathbb{N}$.

Il teorema di estensione di Kolmogorov ci permette di costruire un processo stocastico $\{X_n\}_{n\in\mathbb{N}}$ che abbia quelle determinate distribuzioni finito dimensionali. Proprio per la scelta sulle distribuzioni abbiamo che le X_n sono indipendenti e identicamente distribuite.

Appendice B

Alcuni algoritmi black-box

Abbiamo potuto vedere che non vale il *no free lunch* per i processi in $\mathbb{R}^{[0,1]}$. Tuttavia, rimane complicato trovare esplicitamente un algoritmo *migliore* rispetto agli altri, almeno in una classe così ampia di funzioni.

Se però ci limitiamo alla classe delle funzioni continue, di cui conosciamo bene le caratteristiche, possiamo sfruttare proprio la continuità per realizzare buoni metodi di ottimizzazione.

Considereremo di qui in avanti dei problemi di massimo su funzioni di classe $C^0(\Omega)$ dove $\Omega \subset \mathbb{R}^n$. Tutte le conclusioni a cui arriveremo si possono facilmente adattare a problemi di minimo.

Se facciamo l'ulteriore ipotesi che f sia di classe $C^1(\Omega)$, possiamo dire dove cade il massimo: o su $\partial\Omega$ o in un punto stazionario di f in $\mathrm{Int}(\Omega)$.

I punti stazionari si possono determinare come gli zeri del gradiente di f. Se supponiamo $f \in C^2(\Omega)$, questi si possono trovare usando per esempio il metodo di Newton: si sceglie $v \in \Omega$ "sufficientemente vicino" a uno zero di ∇f e, per ogni iterazione del metodo, si pone $v = v - (\Delta(f)(v))^{-1}\nabla f(v)$. Per una buona scelta di v, e un numero sufficiente di iterazioni, v è un'ottima approssimazione di un punto stazionario di f.

Esempio B.1:

Sia $\Omega = [0,1]^2$ e sia $f : \Omega \to \mathbb{R}$ definita come

$$f(x) = \frac{3}{4}e^{-\left((9x-2)^2+(9y-2)^2\right)/4} + \frac{3}{4}e^{-\left((9x+1)^2/49-(9y+1)/10\right)}$$

$$+\frac{1}{2}e^{-\left((9x-7)^2+(9y-3)^2\right)/4} - \frac{1}{5}e^{-\left((9x-4)^2+(9y-7)^2\right)}$$

f viene detta *funzione bidimensionale di Franke* ed è spesso usata come funzione di test per metodi numerici. Noi la useremo per valutare l'efficienza dei vari algoritmi per problemi di massimo.

Siamo interessati a un'approssimazione quanto più possibile precisa dei massimi relativi di f. Applicando il metodo di Newton ne possiamo trovare due:

- il primo, partendo da $v = (0.2, 0.2)$, è approssimabile (in *double precision*) come

$$(0.205991570380508, 0.208050138342942) \tag{B.0.1}$$

e la funzione di Franke vale 1.220032571234521 (si rivela essere il massimo assoluto)

- il secondo, partendo da $v = (0.7, 0.3)$, si approssima come

$$(0.754741554582439, 0.326338194709410) \tag{B.0.2}$$

e la funzione di Franke vale 0.642603191078121.

Abbiamo visto nell'esempio precedente come sia facile, quando siamo bene informati sulla funzione (sappiamo ad esempio quanto valgono le derivate fino al secondo ordine), affrontare e risolvere il problema di massimo. Tuttavia, questo non sempre è il caso.

B.1 Hill climbing e Simulated annealing

Cominciamo adesso ad analizzare metodi *black box*, che non richiedono la differenziabilità delle funzioni in questione.

Le tecniche di *hill climbing*, che in italiano possiamo tradurre come "scalata alla collina", per i problemi di massimo consistono nello scegliere in modo casuale un punto x_0 all'interno del nostro dominio Ω e considerare i valori della funzione in un intorno di x_0, trovando il punto x che mostra un incremento maggiore del valore rispetto a x_0. Dopodichè, si ripete lo stesso ragionamento con $x_0 = x$, un numero sufficiente di volte. Quando non esisterà, intorno a x_0, nessun punto migliorativo, possiamo dire che x_0 è un punto di massimo.

Tuttavia, visto che $\Omega \subset \mathbb{R}^n$, ci risulta computazionalmente impossibile valutare tutti i punti in un intorno di x_0. Come facciamo ad applicare l'hill climbing? Possiamo applicarlo tramite:

- *salita del gradiente*: dato un punto x_0 e una funzione $f \in C^1(\Omega)$, si può calcolare il gradiente di f nel punto x_0, $\nabla f(x_0)$, e cercare di muoversi in $x = x_0 + \varepsilon \nabla f(x_0)$ dove $\varepsilon > 0$ è "sufficientemente piccolo". Tuttavia, dato che f deve essere in $C^1(\Omega)$, questa tecnica non è *black box*.

- *hill climbing stocastico*: dato un punto x_0, si sceglie un punto casuale x in un intorno di x_0 e, se è migliorativo ($f(x) > f(x_0)$) e il miglioramento supera una certa soglia (solitamente scelta a priori), ci si muove in x. Altrimenti, si scelgono altri punti casuali (in un intorno di x_0) finchè non se ne trova uno che soddisfi tali condizioni. Se dopo un certo numero di tentativi non riusciamo a trovare nessun punto migliorativo, allora ci arrestiamo e x_0 è punto di massimo. Questa tecnica è *black box*.

Il grande problema delle tecniche di *hill climbing* è che spesso rimangono intrappolate in punti di massimo locale, non raggiungendo il vero massimo di f su Ω. La tecnica di *simulated annealing* permette di limitare tale eventualità.

simulated annealing: dato un punto x_0, si sceglie un punto casuale x in un intorno di x_0 e, se è migliorativo ($f(x) > f(x_0)$), ci si muove in x. Se invece è peggiorativo ($f(x) < f(x_0)$), possiamo muoverci in x con una probabilità (piccola) scelta a priori. Se nessuna delle due precedenti condizioni è soddisfatta, si scelgono altri punti casuali (in un intorno di x_0).

Questa tecnica, se eseguita per un numero sufficiente di iterazioni, ha meno probabilità di rimanere bloccata in un massimo locale (questo perchè c'è anche la possibilità di muoversi in un punto peggiorativo).

Esempio B.2:

Proviamo ad applicare le tecniche di *hill climbing* e *simulated annealing* per trovare il massimo della funzione di Franke su $\Omega = [0, 1]^2$. Entrambi i metodi sono stati applicati a 10 punti iniziali diversi e le iterazioni sono state 10000. Per il metodo di *hill climbing* otteniamo il seguente output (una linea per ogni esecuzione)

Val. migliore	Punto max
0.64229210	(0.75999604, 0.32897426)
1.21577565	(0.18919926, 0.20728876)
1.20700589	(0.20375100, 0.17790949)
1.21797944	(0.20272126, 0.19657281)
1.21783355	(0.21485455, 0.21644198)
1.21845441	(0.19657144, 0.21174277)
0.64170518	(0.76430115, 0.32919103)
1.21988801	(0.20306729, 0.20894952)
1.21999137	(0.20739798, 0.20722497)
0.64078351	(0.74740300, 0.31439540)

Vediamo che per 3 volte su 10 ci siamo bloccati nel massimo locale $B.0.2$, mentre abbiamo raggiunto il massimo assoluto $B.0.1$ per 7 volte su 10.

Applichiamo invece *simulated annealing*, ottenendo il seguente output

Val. migliore	Punto max
1.22000655	$(0.20718721, 0.20755054)$
1.21971071	$(0.20175198, 0.20612446)$
1.21985173	$(0.20582335, 0.20454232)$
1.21915987	$(0.21318758, 0.21053322)$
1.22000733	$(0.20725500, 0.20784509)$
1.21878212	$(0.21495263, 0.20715589)$
1.21920985	$(0.20692857, 0.21546982)$
1.21950916	$(0.20627132, 0.20209311)$
1.21879867	$(0.21297479, 0.20248011)$
1.21794244	$(0.19482336, 0.20402174)$

Vediamo che abbiamo sempre raggiunto il massimo assoluto $B.0.1$.

B.2 L'algoritmo delle api

Un altro algoritmo usato in ottimizzazione è l'algoritmo delle api. Esso imita il comportamento di una colonia di api alla ricerca di cibo in un territorio molto ampio.

Le api operaie di un alveare esplorano il territorio in questione e, quando trovano una fonte abbondante di cibo, tornano all'alveare e iniziano la "danza dell'ape", una danza con cui comunicano alle altre api la direzione e la distanza della fonte di cibo.

L'algoritmo che presentiamo (sempre per problemi di massimo) riprende tale idea: esploriamo il nostro dominio Ω in un certo numero di punti e "segnaliamo ad altre api" la localizzazione dei punti migliori.

Algoritmo:

1- Scegliamo casualmente n punti $x_1, \ldots, x_n$ di Ω

Ripetiamo per N_{it} volte:

2- Individuiamo gli m punti migliori (la funzione assume valori più grandi)

3- Per ognuno di questi, facciamo altre ne esplorazioni in un (piccolo) intorno di x_m e sostituiamo x_m con la migliore di queste esplorazioni.

4- Gli altri $n - m$ punti vengono sostituiti da punti scelti casualmente.

Se N_{it} è sufficientemente grande, otteniamo una buona approssimazione del massimo, senza possibilità di rimanere bloccati in un massimo locale.

Esempio B.3:

Applichiamo l'algoritmo delle api per trovare il massimo su $\Omega = [0,1]^2$ della funzione di Franke. I parametri usati sono $n = 30, m = 10, ne = 10$. L'intorno di x_m scelto e' una palla di raggio 0.1 e l'algoritmo è stato eseguito 10 volte. Otteniamo il seguente output

Val. migliore	Punto max
1.22000173	(0.20569742, 0.20945738)
1.21995004	(0.20707017, 0.20598188)
1.21972889	(0.20766730, 0.21229357)
1.21978568	(0.20901972, 0.20543981)
1.21989663	(0.20578359, 0.21107236)
1.21963583	(0.21107928, 0.20829554)
1.21969468	(0.20142427, 0.20675402)
1.21999611	(0.20682078, 0.20674479)
1.21973714	(0.20305117, 0.20463015)
1.21994888	(0.20742786, 0.20620662)

Vediamo che abbiamo sempre raggiunto il massimo assoluto $B.0.1$.

Esempio B.4:

Consideriamo la seguente EDS parametrica

$$dY(t) = \cos(2\pi t)dt + \varepsilon Y(t)dW(t) \qquad Y(0) = 1 \tag{B.2.1}$$

dove $W(t)$ è un moto browniano e $\varepsilon \geq 0$ è il parametro.

Sia $X(t) = X(t, \varepsilon)$ soluzione (dipende dal parametro!) di $B.2.1$. Sia $\Omega = [0, 1]$. Sia $E[X(t)]$ il valore atteso di $X(t)$. Vogliamo trovare il massimo di $E[X(t)]$ per $t \in \Omega$.

Se $\varepsilon = 0$, $E[X(t)]$ risulta essere differenziabile, mentre non lo è per ogni scelta di $\varepsilon > 0$. Inoltre, non è così immediato risolvere esplicitamente tale equazione. In questi casi, per affrontare il problema di massimo conviene agire numericamente tramite i metodi *black box*.

Una buona approssimazione, per t fissato, di $E[X(t)]$ è trovata mediante il Metodo Monte Carlo, cioè simulando il processo più volte e prendendo la media. La simulazione numerica può essere condotta attraverso i metodi di Runge Kutta stocastici. Abbiamo applicato il più semplice, cioè il metodo di ordine 0.5, che si rivela comunque efficace per il tipo di simulazione condotta.

Quindi adesso, fissato $\varepsilon > 0$, sappiamo come associare a ogni $t \in \Omega$ il corrispondente $E[X(t)]$

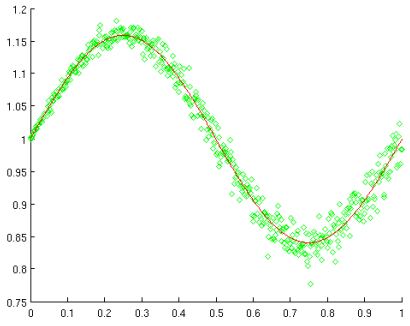

Grafico di $E[X(t)]$ per $\varepsilon = 0$ (linea continua) e $\varepsilon = 0.2$ (punti staccati)

Possiamo quindi pensare di usare l'algoritmo delle api per trovare il massimo. Lo abbiamo applicato, per vari valori di ε, con parametri $N_{it} = 50$, $n = 100$, $m = 20$, $ne = 10$, e abbiamo preso, come intorni degli m punti migliori di ogni iterazione, palle di raggio 0.02.

Nella seguente tabella abbiamo rappresentato l'output del nostro algoritmo. Ci siamo limitati a due cifre dopo la virgola dato che nel nostro algoritmo di Runge-Kutta stocastico abbiamo usato come passo di integrazione $h = 0.01$.

ε	$E[t_{max}]$	t_{max}
0.00	1.15	0.27
0.20	1.17	0.27
0.40	1.18	0.26
0.60	1.19	0.25
0.80	1.21	0.28
1.00	1.23	0.29
1.20	1.25	0.27
1.30	1.27	0.23
1.40	1.34	0.96
1.50	1.41	0.98
2.00	9.37	0.99

Notiamo che il punto di massimo rimane vicino a quello del caso $\varepsilon = 0.00$ fino a $\varepsilon = 1.30$ mentre per valori di ε più grandi la perturbazione del termine stocastico diventa significativa e sposta il punto di massimo verso l'estremo superiore del nostro insieme Ω.

Appendice C

Ottimizzazione di processi

Siamo interessati all'ottimizzazione di processi stocastici, cioè nello stabilire un algoritmo di ricerca che funzioni meglio della ricerca casuale.

La strategia generale è quella di sfruttare le correlazioni tra le variabili per avere un'indicazione di come muoversi. Per mostrare l'efficacia di questo approccio, è necessario provvedere alla simulazione di un processo stocastico di dimensioni opportune.

Una classe di processi interessanti sotto questo punto di vista è quella dei processi gaussiani; infatti, è semplice simulare variabili normali di media nulla e varianza uguale a 1; da queste, attraverso la moltiplicazione per un'opportuna matrice, siamo in grado di passare a un processo con una data matrice di covarianza.

Esempio C.1:
Consideriamo un processo gaussiano con matrice di covarianza Σ. Questo tipo di processi potrebbe soddisfare $GNFL$ solo se la loro matrice è nella forma

$$\Sigma_{i,i} = 1 \qquad \Sigma_{i,j} = \rho \quad (i \neq j)$$

Per esempio, se $n = 5$ le matrici sono tutte nella forma:

$$\Sigma = \begin{bmatrix} 1 & \rho & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho & \rho \\ \rho & \rho & 1 & \rho & \rho \\ \rho & \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & \rho & 1 \end{bmatrix}$$

affinchè questa sia possa essere matrice di covarianza di un processo gaussiano, occorre che sia definita positiva.

Notiamo immediatamente che, se $\rho > 1$, non è definita positiva: infatti, la seconda sottomatrice principale

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

avrebbe determinante negativo.
Per $\rho = 1$ invece la matrice è definita positiva: tutti i determinanti delle sottomatrici principali sono nulli.

Se $\rho < -\frac{1}{n-1}$ allora la matrice non può essere definita positiva: se prendiamo $d = (1, 1, \ldots, 1)^T$ allora

$$d^T \Sigma d < 0$$

Se $\rho = -\frac{1}{n-1}$ allora la matrice ha determinante uguale a 0: infatti, $d = (1, 1, \ldots, 1)^T$ è un vettore di $Ker(\Sigma)$; e i determinanti delle sottomatrici sono tutti positivi.

In conclusione, deve essere $\rho \in [-\frac{1}{n-1}, 1]$.

Esempio C.2:
Come si può generare un processo gaussiano con n variabili con media μ e una data matrice di covarianza Σ?

La matrice Σ di un processo gaussiano è definita positiva ed ammette una sola radice quadrata S definita positiva. Tale matrice S è tale che, se generiamo un vettore v di valori con distribuzione gaussiana di media nulla e varianza uguale a 1 (ognuno dei quali può essere generato p.e. considerando due valori $U1$ e $U2$ scelti con distribuzione uniforme nell'intervallo $[0, 1]$ e ponendo $v(i) = \sqrt{-2ln(U1)}\sin(2\pi U2)$), allora il vettore w ottenuto ponendo

$$w = S * v + \mu$$

è un vettore di valori che seguono una distribuzione gaussiana di media μ e matrice di covarianza uguale a Σ.

Se poniamo $\sigma^2 = 1$, le matrici dei processi gaussiani che potrebbero soddisfare $GNFL$ sono tutte nella forma

$$\Sigma_{i,i} = 1 \qquad \Sigma_{i,j} = \rho \quad (i \neq j)$$

La radice quadrata (definita positiva) S risulta essere nella forma

$$S_{i,i} = a \qquad S_{i,j} = b \quad (i \neq j)$$

dove a e b sono tali da soddisfare le equazioni

$$a^2 + (n-1)b^2 = 1 \qquad nab = \rho$$

dalla seconda ci si ricava $a = \frac{\rho}{nb}$ e b risulta essere

$$b = \sqrt{\frac{n^2 - \sqrt{n^4 - 4n^2(n-1)\rho^2}}{2n^2(n-1)}}$$

il precedente calcolo ci permette di trovare la matrice S senza effettuare la decomposizione di Cholesky di Σ, risparmiando notevole tempo computazionale.

Esempio C.3:
Sia $(\Omega, \mathcal{A}, P)$ spazio di probabilità, $\mathcal{X}$ insieme discreto. Consideriamo un processo stocastico f definito su $\Omega \times \mathcal{X}$: richiediamo che le variabili siano di media nulla e varianza uguale a 1. Fissato $\omega \in \Omega$, vogliamo trovare $x \in \mathcal{X}$ che massimizzi $f(\omega, \cdot)$. Possiamo procedere per ricerca casuale: questo algoritmo, per trovare x, impiegherà un numero di passi che è distribuito in modo uniforme su $\{1, \ldots, |\mathcal{X}|\}$.

Un altro algoritmo potrebbe considerare le covarianze tra le variabili del processo f: possiamo eseguire un'opportuna fase di training in cui si osserva il processo per un fissato numero di elementi di Ω; nel seguito, si usano le informazioni acquisite per

stimare le covarianze tra le variabili.

Avere le covarianze tra le variabili ci aiuta a sapere quali punti $x \in \mathcal{X}$ visitare. Per esempio, supponiamo di avere un processo tale che $f(x_1, \omega) > 0$ e di avere $Cov(f(x_1, \cdot), f(x_2, \cdot)) = -0.9$ e $Cov(f(x_1, \cdot), f(x_3, \cdot)) = 0.9$; per un problema di massimo, risulta più plausibile visitare il punto x_3 piuttosto che il punto x_2.

L'algoritmo che proponiamo esegue un certo numero di osservazioni, che chiamiamo H, del processo in questione, e assegna, dopo la fase di training, una certa significanza a ogni punto x (sommando i valori assoluti delle covarianze $Cov(f(x, \cdot), f(y, \cdot))$, $y \neq x$). Per esempio, se $|\mathcal{X}| = 3$ e $Cov(f(x_1, \cdot), f(x_2, \cdot)) = -0.9$, $Cov(f(x_1, \cdot), f(x_3, \cdot)) = 0.9$ e $Cov(f(x_2, \cdot), f(x_3, \cdot)) = 0.1$, allora assegniamo a x_1 significanza 1.8, x_2 e x_3 hanno invece significanza 1.0.

Nella fase seguente, fissato ω, si cominciano a visitare gli N punti più significativi (secondo le indicazioni precedenti), annotando i valori che il processo assume. Dopo questo, si assegna una nuova significanza a ogni punto $x \in \mathcal{X}$. Questa si calcola sommando, per ogni y visitato, il valore osservato in y moltiplicato per la covarianza tra $f(x, \cdot)$ e $f(y, \cdot)$.

Infine, è sufficiente cominciare a visitare i punti, secondo l'ordine dato da questa significanza, aggiornando via via la significanza (secondo le indicazioni precedenti).

Fissiamo $|\mathcal{X}| = 50$.

Per quanto riguarda la ricerca casuale, riusciamo a raggiungere l'ottimo in meno di 25 iterazioni con una probabilità uguale a 0.5, e a raggiungere l'ottimo in meno di 20 iterazioni con una probabilità uguale a 0.4.

Vedremo dalla tabella seguente che il nostro algoritmo funziona sempre meglio rispetto alla ricerca casuale, sia se lo paragoniamo al numero di volte in cui raggiunge l'ottimo in meno di 25 iterazioni, sia al numero di volte in cui raggiunge l'ottimo in meno di 20 iterazioni.

-	$H = 20,\ It \leq 25$	$H = 20,\ It \leq 20$
N = 10	+32.5 %	+37.5 %
N = 12	+20.0 %	+25.0 %
N = 14	+22.5 %	+18.8 %
N = 16	+20.0 %	+21.9 %
N = 18	+22.5 %	-
N = 20	+22.5 %	-

Proviamo il nostro algoritmo con $|\mathcal{X}| = 100$ e $|\mathcal{X}| = 150$ variabili.

Nel caso $|\mathcal{X}| = 100$:

-	$H = 80,\ It \leq 50$	$H = 80,\ It \leq 40$
N = 14	+1.7 %	+4.2 %
N = 16	+10.0 %	+10.4 %
N = 18	+1.67 %	+4.2 %
N = 20	+18.3 %	+6.3 %

Nel caso $|\mathcal{X}| = 150$:

-	$H = 140,\ It \leq 75$	$H = 140,\ It \leq 60$
N = 16	+1.3 %	+3.1 %
N = 18	+16.3 %	+15.6 %
N = 20	+2.5 %	-3.1 %
N = 22	+3.8 %	+7.8 %
N = 24	+1.3 %	+14.06 %

Ringraziamenti

Questa Tesi è stata la conclusione del mio percorso accademico, forse la fine del mio percorso di studi, e mi sento in dovere di ringraziare tutti coloro che mi hanno accompagnato dall'infanzia sino a diventare quello che sono.

Gli studi di matematica sono difficili ed è necessario molto impegno. Quindi è impossibile pensare di affrontarli senza avere dietro una base sempre pronta a darmi una mano.

Ringrazio infinitamente la mia famiglia, che mi ha insegnato a non farmi distrarre troppo dalle sirene del mondo ma a essere concentrato nei miei studi e nei miei obiettivi, a mettere sempre determinazione in quello che faccio; mi ha sostenuto economicamente e soprattutto moralmente durante gli Studi, incoraggiandomi nelle difficoltà e felicitandosi quando le cose andavano bene; è sempre stata attenta alle mie esigenze e propensioni, indirizzandomi verso gli studi scientifici. Quindi, grazie Gabriella, Giulietta e Silvano.

A scuola mi sono sempre trovato bene, tra insegnanti bravi e buoni compagni, e ho maturato un buonissimo ricordo di tutti voi.

Un ringraziamento particolare va ad Elia, ottimo amico che conosco dall'infanzia. Devo dire grazie a Silvia e Gioia per le belle serate passate assieme.

Non dimentico i miei compagni di università, sempre pronti a farmi sorridere nella quotidianità, per quanto monotona e meschina questa possa essere. Ringrazio in particolare i più vicini, Gianmarco, Andrea e Marta.

Grazie anche a tutti i professori che mi hanno “istruito” in questi 5 anni di Università, facendomi capire l’importanza del pensiero critico e libero, apprezzare il rigore e il linguaggio di questa disciplina e coniugare la matematica con l’informatica. Un ringraziamento particolare va al mio relatore, per avermi ricevuto, e corretto questa Tesi; senza di lui, e i suoi spunti, questo lavoro non sarebbe neppure nato.

Bibliografia

- [1] David H. Wolpert, William G. Macready, *No Free Lunch Theorems for Optimization*, IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, (Aprile 1997)

- [2] Christian Igel, Marc Toussaint, *On classes of functions for which No Free Lunch results hold*, Elsevier Information Processing Letters 86 (2003) 317-321

- [3] Anne Auger, Olivier Teytaud, *Continuos lunches Are Free Plus the Design of Optimal Optimization Algorithms*, Algorithmica (2010) 57, 121-1212

- [4] Aureli Alabert, Ricard Caballero, *Some remarks on No-Free-Lunch Theorems in the continuum*, Preprint (2012)

- [5] Christian Igel, Marc Toussaint, *A No-Free-Lunch Theorem for Non-Uniform Distributions of Target Functions*, Journal of Mathematical Modelling and Algorithms (2004) 3: 313-322

- [6] Andrea Valsecchi, *A Study of Some Implications of the No Free Lunch Theorem*, EvoWorkshops 2008, LNCS 4974 (2008), pp. 633-642

- [7] Kolmogorov A. N., *Concetti fondamentali di teoria della probabilità*, a cura di L. Accardi, Teknos (1995)

- [8] Sheldon M. Ross, *Calcolo delle Probabilità e Statistica*, Apogeo Editore (2004)

- [9] Paolo Baldi, *Equazioni Differenziali Stocastiche e Applicationi*, Pitagora (collana Quad. dell'Unione Matematica Italiana) (2000)

- [10] A. Quarteroni, *Matematica numerica*, Springer (2008)

- [11] V. Comincioli, *Analisi numerica, metodi modelli e applicazioni*, McGraw Hill-Italia (1995)

- [12] McCracken D.D., *Numerical methods and FORTRAN programming*, John Wiley, New York (1996)