Ringraziamenti

Ringrazio prima di tutto mia madre e la mia famiglia per tutti i sacrifici che sono stati necessari per permettermi di arrivare a questo punto. Ringrazio i miei amici di sempre per non avermi mai lasciato da solo anche se qualche volta me lo sarei meritato. E ringrazio i nuovi amici e compagni di vita di xtream, una realtà lavorativa che mi ha accolto a braccia aperte e grazie a cui ho imparato cose che non avrei mai pensato di dovere affrontare. Per ultimi ma non meno importanti, rigranzio molto il mio relatore Dott. Marco Fattore e il Dott. Alberto Arcagni che mi ha aiutato molto nella parte algoritmica e di programmazione in C. Non ho dimenticato niente di questi 3 anni, che sono stati probabilmente i più vivi della mia vita fino a questo punto. Dedico questa tesi a Papà Mauro a cui sarebbe sicuramente piaciuto molto poterla leggere, e a Celeste, grazie a cui ho scoperto nuovi lati di me e che mi piacerebbe ricordare col sorriso ogni volta che riaprirò questo volume.

Per fortuna, questo è solo l'inizio.

Sommario

La tesi prende spunto dall'immensa mole di lavoro originale del Dott. Marco Fattore nell'ambito delle applicazioni socio-economiche della teoria degli insiemi parzialmente ordinati, strutture proprie della matematica discreta che si prestano in maniera ottima a descrivere fenomeni sociali complessi quali la povertà e la disparità sociale esulando da un'ottica tradizionalmente unidimensionale.
In particolare, il lavoro svolto è teso ad implementare misure per il calcolo della polarizzazione sociale su dati parzialmente ordinati all'interno del pacchetto *PARSEC* disponibile per il software statistico *R*, esaminando anche le difficoltà computazionali incontrate e proponendovi soluzioni.
Il capitolo 1 fornisce una introduzione agli insiemi parzialmente ordinati e alla terminologia utilizzata nel resto dell'elaborato;
il capitolo 2 approfondisce il tema delle estensioni lineari di un insieme parzialmente ordinato distinguendone quattro diversi tipi;
il capitolo 3 introduce le misure di polarizzazione implementate e le fondazioni metodologiche e teoriche necessarie per la loro applicazione in ambito multidimensionale;
il capitolo 4 è una disamina delle difficoltà computazionali incontrate nel percorso e delle soluzioni proposte;
il capitolo 5 si concentra sull'aspetto pratico dell'implementazione in PARSEC e sulle tecniche di sviluppo utilizzate allo scopo;
il capitolo 6 contiene un'applicazione delle misure implementate su dati reali e fornisce indicazioni utili per sviluppi futuri del lavoro.

Indice

Capitolo 1

Introduzione

1.1 Ordini e posets

Gli insiemi parzialmente ordinati, in seguito *posets* dalla traduzione inglese *partially ordered sets* per brevità, sono una struttura propria della matametica discreta che utilizza metodi e risultati della *teoria degli ordini*, ossia quella branca della matematica che studia particolari tipi di relazioni binarie, *ordini* e *preordini*, che inducono sui loro insiemi supporto il concetto naturale di ordinamento.
Sono entrambi concetti formalizzati in maniera molto chiara e completa e dunque, nonostante una loro analisi a 360 gradi vada oltre gli scopi di questo lavoro, si ritiene necessario fornirne una breve panoramica vista l'ottica prettamente teorica e algoritmica dell'elaborato. [11]

Una relazione binaria su un insieme P è un sottoinsieme $R \subseteq P \times P$. Due elementi $x, y \in P$ sono messi in relazione da R se $(x, y) \in R$ e in tal caso si scrive xRy.
R è chiamato *ordinamento parziale* su P se soddisfa le seguenti proprietà:

1. *Riflessività*: $xRx \ \forall x \in P$

2. *Antisimmetria*: $(xRy \wedge yRx) \implies x = y \ \forall x, y \in P$

3. *Transitività*: $(xRy \wedge yRz) \implies xRz \ \forall x, y, z \in P$

ed in tal caso la relazione si può indicare con $\leq_P$ per richiamare il concetto di ordinamento e il pedice P relativo all'insieme supporto può essere omesso se il contesto non è ambiguo. Se $x \leq y$ e $x \neq y$ allora scriviamo $x < y$. Se $\neg(x \leq y \vee y \leq x)$ diciamo che x e y sono *incomparabili* fra loro e scriviamo $x \parallel y$. Un ordinamento in cui tutti gli elementi sono comparabili fra di loro è chiamato *ordinamento lineare* o *ordinamento completo*. La coppia $\Pi = (P, \leq_\Pi)$, dove P è un insieme di oggetti, identifica un poset.

Di seguito alcuni esempi notevoli di relazioni d'ordine:

- La relazione classica "minore o uguale", ossia xRy se $x \leq_\mathbb{R} y$ è un ordinamento sul campo dei numeri reali $\mathbb{R}$ e su ogni suo sottoinsieme. È un ordinamento completo perché è sempre possibile comparare due numeri reali.

- *Ordinamento lessicografico*: Dati $P = (X, \leq_P)$ e $Q = (Y, \leq_Q)$ un ordinamento lessicografico è una realzione $\leq_S$ su $X \times Y$ tale che dati $(a, b), (c, d) \in X \times Y$, $(a, b) \leq_S (c, d)$ se $a \neq c \wedge a \leq_P c$ oppure se $a = c \wedge b \leq_Q d$. Si può dimostrare in maniera triviale che $\leq_S$ è un ordinamento completo se $\leq_P$ e $\leq_Q$ sono ordinamenti completi. La definizione può essere estesa più generalmente a qualsiasi prodotto cartesiano di un numero finito di insiemi ordinati.

- *Ordinamento prodotto*: Dati P e Q come sopra un ordinamento prodotto è una relazione $\leq_T$ tale che, dati $(a, b), (c, d) \in X \times Y$, $(a, b) \leq_T (c, d)$ se $a \leq_P c \wedge b \leq_Q d$. A differenza dell'ordinamento lessicografico, $\leq_T$ può non essere un ordinamento completo anche se lo sono $\leq_P$ e $\leq_Q$.

All'interno di un poset $\Pi = (P, \leq_\Pi)$, si dice che x *copre* y, denotato come $x \prec y$ se $x \leq y$ e non esiste alcun altro elemento $z \in P$ tale che $x \leq z \leq y$.

Un elemento $x \in P$ tale che $y \leq x \ \forall y \in P$ si dice *massimo* di Π e viceversa un elemento x tale che $x \leq y \ \forall y \in P$ è detto *minimo* di Π.

La *dimensione* di un poset è definita come la cardinalità di P, $|P|$. Una *chain* di Π è un sottoinsieme di P in cui tutti gli elementi sono comparabili fra di loro. Una *antichain* di Π è un sottoinsieme di P in cui tutti gli elementi non sono comparabili con alcun altro elemento. L'*altezza* di un poset è la cardinalità della sua più grande chain e si denota

con $h(\Pi)$. La *larghezza* di un poset è la cardinalità della sua più grande antichain e si denota con $w(\Pi)$. Il senso di queste ultime due denominazioni risulterà più chiaro quando veranno esaminate le rappresentazioni grafiche dei poset.

Dati due poset $\Pi = (X, \leq_\Pi)$ e $\Theta = (X, \leq_\Theta)$ costruiti sullo stesso insieme X, si dice che Θ è una *estensione* di Π se $x \leq_\Pi y \implies x \leq_\Theta y \; \forall x, y \in X$, ossia se Θ è ottenibile da Π trasformando incomparabilità in comparabilità. Se oltre ad essere una estensione di Π, Θ possiede un ordinamento completo, allora Θ viene detto *estensione lineare* di Π. L'insieme delle estensioni lineari di Π viene denotato con $\Omega(\Pi)$. Un risultato fondamentale nella teoria dei poset ([30]) afferma che due poset di dimensione finita hanno diversi insiemi di estensioni lineari e che ogni poset finito può essere ottenuto come intersezione delle sue estensioni lineari, ossia considerando comparabili gli oggetti comparabili comuni a tutte le estensioni lineari. Le estensioni lineari saranno uno degli oggetti più importanti all'interno di questo lavoro e veranno approfondite in maniera esaustiva nel capitolo 2.

1.2 Rappresentazioni di un poset

Poiché per ogni poset finito è dimostrabile che $x \leq y$ se e solo se esiste una sequenza di elementi $z_0, z_1 \dots z_k$ tale che $x = z_0 \prec z_1 \prec \cdots \prec z_k = y$, l'ordinamento parziale è univocamente definito dalla relazione di *copertura*. Attraverso questa relazione è possibile costruire il cosiddetto *diagramma di Hasse*, un grafo aciclico direzionato il cui insieme dei nodi è l'insieme supporto P e in cui è presente uno *spigolo* che collega y a x se e solo se $x \prec y$.

La direzione degli spigoli è generalmente omessa in quanto per convenzione è sempre dall'alto verso il basso poiché un elemento x è posizionato sopra ad y se e solo se $x \leq y$. Il grafo presenta dunque il massimo del poset in alto e il minimo in basso.

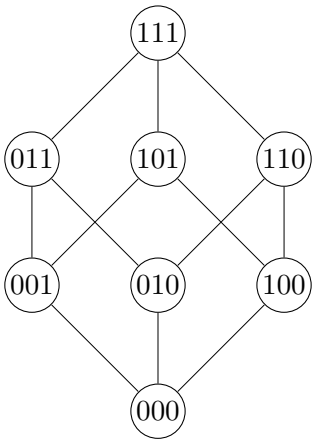

Figura 1.1 – Diagramma di Hasse di un poset composto dall'insieme $\{0,1\}^3$ e da ordinamento prodotto

Figura 1.2 – Diagramma di Hasse di una chain

Figura 1.3 – Diagramma di Hasse di una antichain

Dalle figure 1.2 e 1.3 è evidente il senso dei termini *altezza* e *larghezza* di un poset: il diagramma di Hasse di una chain si espande sempre verticalmente mentre quello di una antichain orizzontalmente.

La descrivibilità di ogni poset mediante una rappresentazione a grafo apre all'applicazione della *teoria dei grafi* allo studio dei poset. Un risultato tanto elementare quanto importante è, per esempio, che $x \leq y$ se e solo se esiste un percorso all'interno del diagramma di Hasse che colleghi x a y. Per altri e ben più complicati risultati che uniscono la teoria dei grafi agli insiemi parzialmente ordinati, riuscendo persino a creare nuovi campi di studio quali la cosiddetta *teoria dei reticoli*, si rimanda a [11].

In modo più adatto per le computazioni, un poset è definibile equivalentemente tramite matrici. Dato il poset standard $\Pi = (P, \leq_P)$ con $P = \{p_1, p_2, \ldots, p_n\}$ si definisce matrice di incidenza o *incidence* di Π una matrice Z tale che:

$$Z_{ij} = \begin{cases} 1, & \text{se } p_i \leq_P p_j \\ 0, & \text{altrimenti} \end{cases}$$

e si definisce *cover matrix* di Π una matrice C tale che:

$$C_{ij} = \begin{cases} 1, & \text{se } p_i \prec_P p_j \\ 0, & \text{altrimenti} \end{cases}$$

L'utilità della rappresentazione matriciale dei poset è smisurata. Per esempio, intersecare due poset significa moltiplicare elemento per elemento le loro matrici di incidenza (o le cover matrix). Più formalmente, siano $\Pi = (X, \leq_\Pi)$ e $\Theta = (X, \leq_\Theta)$ due poset sullo stesso insieme e sia $inc(\cdot)$ la funzione che mappa un poset all propria matrice di incidenza. Allora $inc(\Pi \cap \Theta) = inc(\Pi) \circ inc(\Theta)$[1].

1.3 Applicazioni dei poset

I posets sono una struttura molto versatile che trova applicazione in una infinità di campi di studio, dalla geometria computazionale ([32]), all'informatica pura ([6]), alle scienze sociali ed economiche ([16] e [13], fra gli altri). In questa sezione ci concentreremo sulle applicazioni sociali, strettamente collegate al lavoro svolto.

Tradizionalmente le analisi su povertà, progresso, benessere e disparità sociale si basano su indici costruiti a partire dal PIL, un indicatore molto limitante in quanto riduce fenomeni estremamente complessi a un solo numero. È indubbio inoltre che il reddito non catturi totalmente il concetto di benessere, come già suggerito dal premio Nobel Amartya Sen in [33] e [34] . Di questo parere è anche l'Unione Europea che ha istituito a partire dal 2007 l'iniziativa *Beyond GDP*, che si occupa di ricercare misure più adeguate e aderenti

[1] L'operatore binario $\circ$ è noto come prodotto di Hadamard, altro nome per il prodotto elemento per elemento fra matrici.

alla realtà dei fenomeni presi in esame. Si veda per esempio [17]. Tuttavia, la maggior parte della letteratura a tal proposito non esce da un'ottica unidimensionale, combinando vari indicatori oltre al PIL tramite opportuni pesi nella speranza di ottenere un indice che descriva accuratamente la realtà.

Le poche ricerche volte ad utilizzare una logica puramente multidimensionale, escludendo quelle del Dott. Fattore, hanno prodotto risultati soddisfacenti solamente nel caso di variabili quantitative (si veda [18]). Ridurre però fenomeni naturalmente qualitativi come la povertà o la disparità sociale ad una dimensione quantitativa significa comunque indurre a monte una distorsione sulla realtà (si veda sempre [33]). Delle variabili qualitative possono per esempio rappresentare possesso o non possesso di un determinato bene o accesso a una determinata risorsa e per quanto ciò possa sembrare semplicistico se utilizzate con le dovute accortezze hanno molto più valore informativo di variabili quantitative quali reddito o consumi in dollari, assunti come standard nell'analisi socio-economica.

Il lavoro originale del dott. Marco Fattore insieme al dott. Alberto Arcagni appare dunque come l'unico esempio di applicazione efficace di analisi multidimensionale applicata a dati qualitativi. Il loro lavoro si applica principalmente all'analisi della povertà e della deprivazione (fra gli altri [13], [14], [15]) ma introduce anche l'applicazione dei poset allo studio della polarizzazione sociale in [16].

All'interno dell'elaborato si andrà ad approfondire il tema della polarizzazione sociale a partire da [16] e [12], mostrando i vantaggi dell'approccio *posetico* rispetto alla logica unidimensionale.

Capitolo 2

Estensioni lineari in poset con ordinamento prodotto

Se l'incomparabilità appare *normale* e *naturale* agli occhi di un essere umano in determinati fenomeni e ciò giustifica l'applicazione dei poset, per un calcolatore è un concetto molto compllicato da gestire. *Linearizzare* un poset è pertanto un'operazione fondamentale per poter procedere a qualsiasi valutazione su di esso. Nessun calcolatore è infatti in grado di gestire un poset come oggetto a se stante e questa è una delle motivazioni per cui tutti le misurazioni socio-economiche in ottica posetica si basano sulla "semplificazione" dei poset negli insieme delle loro estensioni lineari.

L'ottica multidimensionale risiede dunque nel considerare la totalità delle estensioni lineari di Π [12], che identifica univocamente un poset come detto nel paragrafo 1.1, e nel non forzare l'intera struttura del poset in un unico ordinamento. Questo insieme può però essere molto ampio e di difficile trattazione.

Questo e altri problemi verranno affrontati nei prossimi capitoli. In questo, verrà affrontato più da vicino il tema delle estensioni lineari di un poset provvisto di ordinamento prodotto, dapprima definendone 4 categorie per poi concentrarsi su uno specifico tipo che risulterà di fondamentale importanza. La restrizione dela relazione d'ordine a un ordinamento prodotto non è limitante in ambito socio-economico, dato che in tale campo appare naturale ordinare i profili secondo una logica del tipo *"chi più ha, meglio sta"*.

2.1 Tassonomia delle estensioni lineari

Si prenda come esempio il poset della figura 1.1, $\Pi = (\{0,1\}^3, \leq_\Pi)$ con $\leq_\Pi$ ordinamento prodotto. Questo poset è anche chiamato, con leggero abuso di notazione, $\mathbf{2}^3$.

È verificabile che tale poset può essere scomposto nel seguente modo [14] come intersezione di 4 differenti poset:

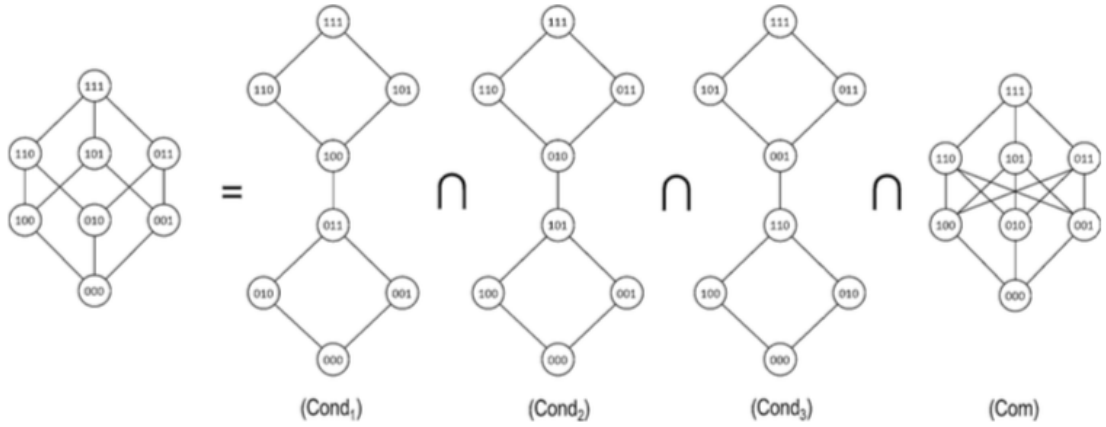

Figura 2.1 – Decomposione del poset $\mathbf{2}^3$

I poset $Cond_1, Cond_2$ e $Cond_3$ sono chiamati *condizionali* poiché ordinano gli elementi condizionatamente a uno degli attributi e variando gli altri. Ognuno di questi 3 poset ha 4 estensioni lineari, che a loro volta vengono chiamate *estensioni linearl condizionali*.

Il poset Com è chiamato *compensativo* ed è ottenuto aggiungendo degli spigoli al diagramma di Hasse originale in modo che ogni elemento su un *livello* copra tutti gli elementi al livello inferiore. Le estensioni lineari di Com sono facilmente calcolabili computando tutte le permutazioni degli elementi allo stesso livello: in questo caso le estensioni lineari ammissibili sono 36, chiamate *estensioni lineari compensative*.

È verificabile che questi 4 poset non hanno alcuna estensione lineare in comune e i loro insiemi di estensioni lineari $CON_1(\mathbf{2}^3) = \Omega(Cond_1), CON_2(\mathbf{2}^3) = \Omega(Cond_2), CON_3(\mathbf{2}^3) = \Omega(Cond_3)$ e $COM(\mathbf{2}^3) = \Omega(Com)$ forniscono dunque una partizione di $\Omega(\mathbf{2}^3)$:

$$\Omega(\mathbf{2}^3) = CON_1(\mathbf{2}^3) \cup CON_2(\mathbf{2}^3) \cup CON_3(\mathbf{2}^3) \cup COM(\mathbf{2}^3)$$

Più in generale è dimostrato [14] che un poset Π generato da k attributi ordinali può essere rappresentato dall'intersezione di k poset condizionali $Cond_1(\Pi), Cond_2(\Pi), \ldots, Cond_k(\Pi)$, un poset compensativo $Com(\Pi)$ e da un insieme residuo e denotato come $Mix(\Pi)$ che contiene *estensioni lineari mixing* che non rientrano in nessuna delle altre categorie. Nel caso di $\mathbf{2}^3$, $Mix(\mathbf{2}^3) = \emptyset$. Formalmente:

$$\Pi = \bigcap_{j=1}^{n} Con_j(\Pi) \cap Com(\Pi) \cap Mix(\Pi)$$

Poiché nessuno dei poset di cui sopra ha estensioni lineari in comune:

$$\Omega(\Pi) = CON(\Pi) \cup COM(\Pi) \cup MIX(\Pi)$$

È possibile affinare ulteriormente la tassonomia e suddividere le estensioni condizionali in due sottotipi.

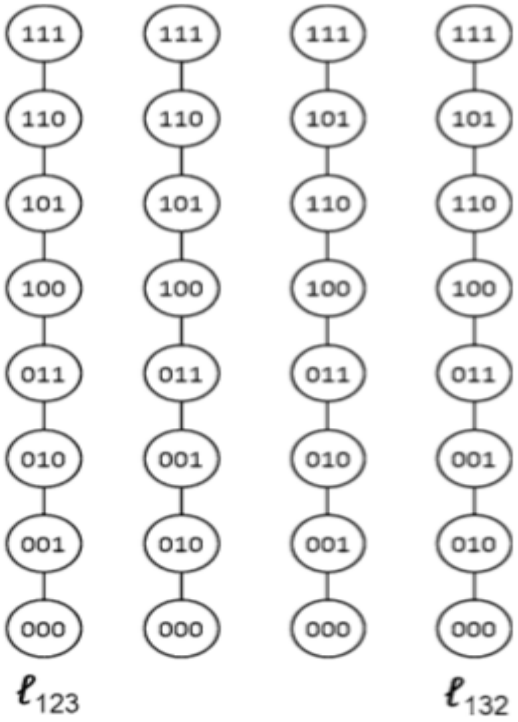

Figura 2.2 – Decomposizione di $Cond_1(\mathbf{2}^3)$

Fra le estensioni lineari di $Cond_1(\mathbf{2}^3)$ in figura 2.2, due ($\boldsymbol{\ell}_{123}$ e $\boldsymbol{\ell}_{132}$) meritano particolare attenzione poiché esse sono ottenute tramite un ordinamento lessicografico dei

profili secondo due specifici ordinamenti *sulle variabili* (v_1, v_2, v_3 e v_1, v_3, v_2). Il concetto di *ordinamento nello spazio delle variabili* verrà formalizzato nel prossimo paragrafo.

Questi tipi di estensioni lineari vengono dette *estensioni lineari lessicografiche*. Nel caso di $\mathbf{2}^3$ ogni poset condizionale ne possiede due, ottenute secondo diversi ordinamenti della seconda e della terza variabile, mantenendo fissa la prima. Per ogni poset condizionale, le estensioni lineari non lessicografiche vengono dette *estensioni lineari switching*. Dopo aver definito come $LEX(\mathbf{2}^3)$ l'insieme delle estensioni lineari lessicografiche di $\mathbf{2}^3$ e come $SWI(\mathbf{2}^3)$ l'insieme delle estensioni lineari switching di $\mathbf{2}^3$, possiamo dunque partizionare nuovamente $\Omega(\mathbf{2}^3)$:

$$\Omega(\mathbf{2}^3) = LEX(\mathbf{2}^3) \cup SWI(\mathbf{2}^3) \cup COM(\mathbf{2}^3)$$

.

Generalizzando le definizioni di estensione lineare lessicografica e switching al generico poset Π generato da k variabili ordinali, è triviale dimostrare che Π possiede esattamente $k!$ estensioni lineari lessicografiche. $\Omega(\Pi)$ può dunque essere decomposto in:

$$\boxed{\Omega(\Pi) = LEX(\Pi) \cup SWI(\Pi) \cup COM(\Pi) \cup MIX(\Pi)} \tag{2.1}$$

arrivando dunque alla decomposizione finale dell'insieme delle estensioni lineari di un poset in 4 sottoinsiemi.

2.2 Estensioni lessicografiche e ranking degli attributi

Poco sopra si è data la definizione di estensione lessicografica di un poset introducendo la nozione di *ranking degli attributi*. Ordinare le variabili per importanza è l'unico modo per integrare informazioni note a priori sulle variabili stesse nella struttura in esame poiché non è possibile assegnare alcun peso ad attributi ordinali, come invece si è soliti fare nel caso quantitativo.

Nel caso dei poset questo significa introdurre nuove informazioni e semplificare la struttura aggiungendo comparabilità o, equivalentemente, aggiungendo collegamenti nel loro diagramma di Hasse. Facciamo un esempio [13]:

Si supponga di essere nel caso della misura della polarizzazione sociale: si prendano due profili $\mathbf{p}$ e $\mathbf{q}$ all'interno del poset $\Pi = (P, \leq_\Pi)$ tali che $\mathbf{p} \parallel_\Pi \mathbf{q}$. Gli elementi sono incomparabili, ma il ricercatore riconosce nel profilo $\mathbf{p}$ una situazione di polarizzazione sociale meno grave rispetto al profilo $\mathbf{q}$ grazie alla sua esperienza, alla letteratura nell'ambito, ecc.. La relazione $\leq_\Pi$ non riflette dunque in maniera accurata la realtà e la comparabilità $\mathbf{p} \leq \mathbf{q}$ dovrebbe esservi integrata, ed oltre ad essa tutte le comparabilità ricavabili dalla proprietà di transitività della relazione.

Generalizzando, l'inserimento di nuove informazioni in un poset si svolge in due fasi distinte:

1. Si aggiunge un nuovo insieme di comparabilità Θ a Π, ottenendo la nuova relazione $\Pi \cup \Theta$ che però non soddisfa la proprietà di transitività;

2. Si calcola la chiusura transitiva $\Pi^* = \overline{\Pi \cup \Theta}$ inducendo in $\Pi \cup \Theta$ le comparabilità derivanti dalla transitività.

Ovviamente, inserire manualmente comparabilità in un poset tramite osservazione diretta diventa sempre più infattibile con l'aumentare del numero dei profili all'interno del poset. Urge dunque ricercare una procedura formale che permetta di risolvere una generica comparabilità a partire dal raking sugli attributi e dai punteggi dei profili sui diversi attributi.

È importante menzionare che il *ranking sugli attributi* non deve essere per forza un ordinamento completo, in quanto anche fra variabili può e deve essere ammessa incomparabilità. Questa considerazione induce naturalmente alla generazione di un poset sull'insieme degli attributi V, $\Lambda = (V, \leq_\Lambda)$.

Essendo un poset finito, Π è univocamente definito dall'insieme delle sue estensioni lineari $\Omega(\Pi)$. Un teorema (*Fattore* 2016 [13], a cui si rimanda per la dimostrazione) presenta però un risultato di fondamentale importanza, dimostrando che nel caso di poset con ordinamento prodotto, *l'insieme delle estensioni lessicografiche è sufficiente a caratterizzare univocamente un poset*. Formalmente:

$$\Pi = \bigcap LEX(\Pi)$$

Per definizione delle estensioni lessicografiche, indurre un ordinamento sugli attributi nel poset implica selezionare un sottoinsieme di estensioni lessicografiche di Π compatibili con l'ordinamento in Λ.

Nel caso di $\mathbf{2}^3$ e di $\Lambda = (v_1 > v_2 > v_3)$, con ovvia notazione, ciò signfica che all'interno di CON_1 si sarebbe considerata solo l'estensione $\boldsymbol{\ell}_{123}$ poiché l'unica compatibile con l'ordinamento descritto. Inserendo le informazioni contenute in Λ si ottiene il seguente diagramma di Hasse:

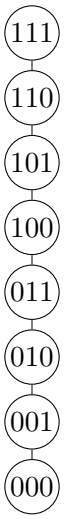

Figura 2.3 – Diagramma di Hasse di $\mathbf{2}^3$ con ordinamento degli attributi $\Lambda = (v_1 > v_2 > v_3)$

Come evidente, il poset risultante è una chain provvista di ordinamento completo. Generalizzando e formalizzando la procedura si associa ad ogni estensione lineare $\lambda \in \Omega(\Lambda)$ una estensione lineare $\boldsymbol{\ell} \in \Omega(\Pi)$ tramite la mappa $lex(\cdot)$:

$$\begin{aligned} lex : \Omega(\Lambda) &\mapsto \Omega(\Pi) \\ : \lambda &\to lex(\lambda) \end{aligned}$$

dove $lex(\lambda)$ è l'unica estensione lineare di Π ordinata in maniera lessicografica secondo l'ordinamento degli attributi in λ. Applicando $lex(\cdot)$ all'intero insieme $\Omega(\Lambda)$ si ottiene il sottoinsieme $Lex(\Pi, \Lambda) \subseteq LEX(\Pi) \subseteq \Omega(\Pi)$ e le estensioni di Π in $Lex(\Pi, \Lambda)$ vengono dette Λ-*ammissibili* e caratterizzano univocamente un poset che è una estensione di Π.

È importante menzionare che, mentre per $|\Omega(\mathbf{2}^{k+1})|$ è stato fornito un limite inferiore in [14] pari a:

$$(k+1)\cdot|\Omega(\mathbf{2}^{k})|^2+\prod_{s=1}^{k+1}\left[\binom{k+2}{s}!\right] \tag{2.2}$$

un numero che, per $\mathbf{2}^5$, un poset che contiene 32 profili, è pari a circa 1.8×10^{17}, invece $|Lex(\Lambda,\Pi)|\leq k!$, una quantità infinitamente più trattabile che tra l'altro dipende unicamente dal numero delle variabili. Quest'ultimo risultato è valido inoltre per qualsiasi poset ottenuto da k variabili ordinali, senza alcun vincolo sul numero di modalità di queste ultime.

Computazionalmente, per ottenere il poset finale $\Pi^*=(P,\leq_{\Pi^*|\Lambda})$ a partire da $Lex(\Pi,\Lambda)$ si opera come illustrato nel paragrafo 1.2. Si moltiplicano cioè elemento per elemento e in sequenza le matrici associate alle estensioni lineari, che per costruzione sono triangolari superiori, fino ad ottenerne una finale che è l'intersezione $\bigcap Lex(\Pi,\Lambda)$.

2.3 $\Omega(\Pi^*)$ o $LEX(\Pi^*)$?

Dopo aver formalizzato in maniera completa il concetto di ordinamento delle variabili indotto su un poset resta il problema di come trattare il poset risultante finale. Dopo avere introdotto nuove informazioni, la complessita dell'oggetto non sembra essere diminuita in maniera sensibile e ciò causa infatti numerose difficoltà computazionali se si vuole trattare l'intero insieme delle estensioni lineari di un poset. Nel prossimo capitolo, si esamineranno le fondazioni metodologiche attualmente presenti in letteratura che obbligano a considerare l'intero insieme $\Omega(\Pi^*)$ e nei successivi, alcune soluzioni ai problemi computazionali.

Capitolo 3

Indici e indicatori calcolati su poset

Questa sezione serve per fornire una breve introduzione alle fondazioni metodologiche che permettono di applicare indici statistici unidimensionali a oggetti multidimensionali. Dopo aver introdotto la teoria dei funzionali su poset enunciata in [12] verranno elencati gli indici statistici di polarizzazione sociale effettivamente implementati durante il lavoro.

3.1 Famiglie di aggregazione e famiglie generatrici di un poset

Fattore in [12] fornisce una formalizzazione completa della teoria dei funzionali su poset allo scopo di aggregare risultati calcolati su componenti del poset più semplici da trattare in un unico indice descrittivo. Se ne darà una breve introduzione, e si rimanda all'articolo per una trattazione completa.

Sia $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ un vettore di k numeri reali in [0,1], $\mathbf{w} = (w_1, w_2, \ldots, w_k)$ un vettore di k pesi in [0,1] che sommano a 1, e $g : [0,1] \to \mathbb{R}$ con g continua e strettamente monotona chiamata *funzione generatrice*.

Si possono dunque definire le *medie pesate quasi-aritmetiche* $M_{g,k}(\cdot)$ come:

$$M_{g,k}(\mathbf{x}, \mathbf{w}) = g^{-1}\left(\sum_{i_1}^{k} w_i g_i(x_i)\right)$$

Molte medie utilizzate comunemente in statistica descrittiva sono medie pesate quasi aritmetiche, come per esempio le *medie potenziate pesate* $M_{[r],k}$:

$$M_{[r],k}(\mathbf{x}, \mathbf{w}) = \left(\sum_{i=1}^{k} w_i x_i^r\right)^{1/r}$$

Con $w_i = \dfrac{1}{k}$ $\forall i, i = 1, \ldots, k$ si hanno invece le *classiche* medie, chiamate *medie quasi-aritmetiche*:

$$M_{g,k}(\mathbf{x}, \mathbf{w}) = g^{-1}\left(\frac{1}{k}\sum_{i_1}^{k} g_i(x_i)\right)$$

Si può dimostrare che le medie quasi-aritmetiche soddisfano alcune proprietà fondamentali dei cosiddetti *sistemi di aggregazione*, definiti come come una collezione

$$\mathbb{F} = \{F_1(\cdot), F_2(\cdot), F_3(\cdot), \ldots\}$$

di *funzionali*[2] rispettivamente con 1,2,3,... argomenti, con $F_1(x) = x$ per convenzione. Le medie quasi-aritmetiche sono gli unici funzionali tali che un sistema di aggregazione costruito con esse soddisfi le seguenti proprietà:

1. *Continuità*: Un sistema di aggregazione $\mathbb{F}$ è definito continuo se ognuno dei suoi elementi $F_k(\cdot)$ è continuo in ciascuno dei suoi k argomenti.

2. *Monotonicità stretta*: Un funzionale $F_k(\cdot)$ è strettamente monotono se, dati $\mathbf{x}, \mathbf{y} \in \mathbb{R}^k$ tali che $\mathbf{x} < \mathbf{y}$ con ordinamento prodotto, allora $F_k(\mathbf{x}) < F_k(\mathbf{y})$. Un sistema di aggregazione è strettamente monotono se ciascuno dei suoi elementi è strettamente monotono.

3. *Scomponibilità*: Un sistema di aggregazione $\mathbb{F}$ è detto scomponibile se per qualsiasi $m, n = 1, 2, \ldots$ e $\forall \mathbf{x} \in [0,1]^m$, $\forall \mathbf{y} \in [0,1]^n$:

$$F_{m+n}(\mathbf{x}, \mathbf{y}) = F_{m+n}\left(\underbrace{F_m(\mathbf{x}_m), \ldots, F_m(\mathbf{x}_m)}_{\text{m volte}}, \mathbf{y}\right)$$

[2]Si definisce funzionale una funzione mappante un vettore k-dimensionale nello spazio dei numeri reali

Questa proprietà si traduce in parole povere con la possibilità di sostituire ai primi m argomenti di $F_{m+n}(\cdot)$ il loro valore aggregato $F_m(\cdot)$ replicato m volte, il che risulterà sicuramente familiare a tutti se si è nel caso particolare delle medie aritmetiche.

Le medie potenziate sono inoltre l'unico tipo di media di quasi-aritmetica ad essere *omogeneo*. Un funzionale F si dice *omogeneo* se,

$$\forall c \in [0,1], F(c \cdot \mathbf{x}) = cF(\mathbf{x})$$

Poiché l'obiettivo è calcolare indicatori su oggetti più semplici del poset di partenza per poi aggregarli in un unico numero, appare necessario e naturale il concetto, peraltro già enunciato sotto altra forma, di *famiglia generatrice* di un poset. Una collezione $\{\pi_1, \pi_2, \ldots, \pi_k\}$ di estensioni di Π si dice *famiglia generatrice* di Π se $\Pi = \pi_1 \cap \pi_2 \cap \ldots \pi_k$. Si sono già viste famiglie generatrici di un poset come per esempio l'insieme delle sue estensioni lineari o delle sue estensioni lineari lessicografiche, ricordando che anche un'estensione lineare è un poset, provvisto in questo caso di ordinamento completo. Una famiglia generatrice si dice *non-overlapping* se gli insiemi delle estensioni lineari dei suoi elementi sono disgiunti. Ossia se:

$$\bigcap_{i=1}^{k} \Omega(\pi_i) = \emptyset$$

Sia l'insieme delle estensioni lineari e delle estensioni lineari lessicografiche risultano dunque famiglie generatrici *non-overlapping*.

Una famiglia generatrice è inoltre detta *completa* se l'unione degli insiemi delle estensioni lineari dei suoi elementi è uguale all'insieme delle estensioni lineari di Π. Ossia se:

$$\bigcup_{i=1}^{k} \Omega(\pi_i) = \Omega(\Pi)$$

Fra le due famiglie generatrici considerate sopra solo quella formata dall'insieme delle estensioni lineari di Π è, ovviamente, una famiglia completa.

Si può dire dunque che, secondo la formalizzazione presentata benché l'insieme delle estensioni lineari lessicografiche sia sufficiente a generare in maniera *"elegante"* un poset, esso non è un sostituto del poset stesso, non è *completo*.

Una famiglia generatrice che è sia *non-overlapping* che *completa* viene detta NOC. Ogni poset possiede almeno una famiglia generatrice NOC ossia l'insieme delle sue estensioni lineari, appunto, ma può possederne di più.

3.2 Funzionali su poset

Dopo aver introdotto il concetto di famiglia di aggregazione si definiranno ora le proprietà desiderabili ed appropriate per una famiglia di aggregazione applicata a un poset.

Innanzitutto si inizi associando al poset una distribuzione di frequenza $\mathbf{p}$, ossia collegando ad ogni profilo rappresentato nel poset la sua rappresentatività nel campione (o nella popolazione) in esame. Questo oggetto è fondamentale perché tutti gli indicatori statistici nel caso ordinale si basano sulla distribuzione di frequenza dei vari profili.

Sia $\gamma(\Pi, \mathbf{p})$ un funzionale calcolato su Π. Appare sensato per quanto enunciato nel paragrafo precedente che esso sia scomponibile applicando $\gamma(\cdot)$ a ciascun elemento di una sua famiglia generatrice. Poiché un poset può avere diverse famiglie generatrici si richiede inoltre che il valore di $\gamma(\Pi, \mathbf{p})$ non cambi variando la famiglia considerata.

Si ipotizzi però che $\{\pi_1, \pi_2\}$ e $\{\pi_3, \pi_4\}$ siano famiglie generatrici per Π. Allora anche $\{\pi_1, \pi_2, \pi_3, \pi_4\}$ è una famiglia generatrice di Π.

Devono dunque esistere due funzioni F_2 e F_4 tali che[3]:

$$\gamma(\Pi) = F_2\left(\gamma(\pi_1), \gamma(\pi_2)\right) = F_2\left(\gamma(\pi_3), \gamma(\pi_4)\right) = F_4\left(\gamma(\pi_1), \gamma(\pi_2), \gamma(\pi_3), \gamma(\pi_4)\right)$$

Ma poiché F_4 dovrebbe essere contemporaneamente indipendente da $\gamma(\pi_1), \gamma(\pi_2)$ e da $\gamma(\pi_3), \gamma(\pi_4)$, ciò porterebbe $F_4(\cdot) = F_2(\cdot) = k$ costante rendendo dunque i due funzionali inutili.

Per evitare questi problemi, ci si limita alle famiglie generatrici NOC. Formalmente si richiede che, date $\{\pi_1, \pi_2, \ldots, \pi_k\}$ e $\{\tau_1, \tau_2, \ldots, \tau_m\}$ due famiglie generatrici NOC per Π esista una famiglia di aggregazione $\mathbb{F}$ tale che:

$$\gamma(\Pi) = F_k\left(\gamma(\pi_1), \gamma(\pi_2), \ldots, \gamma(\pi_k)\right) = F_m\left(\gamma(\tau_1), \gamma(\tau_2), \ldots, \gamma(\tau_m)\right)$$

[3] Si omette $\mathbf{p}$ dandolo per fissato.

dove $F_k(\cdot), F_m(\cdot) \in \mathbb{F}$.

Questa formalizzazione permette la costruzione di indicatori compositi calcolando gli indici sulle singole componenti della più semplice famiglia generatrice NOC, l'insieme delle estensioni lineari di Π, $\Omega(\Pi)$ e poi aggregandoli secondo opportune $F_k(\cdot)$.

Poiché un indice statistico è tradizionalmente continuo, monotono ed omogeneo la scelta ricade su F *media potenziata*. Inoltre siccome non ha senso attribuire più o meno importanza ad una estensione lineare piuttosto che a un'altra il vettore dei pesi $\mathbf{w}$ sarà costante pari a $\frac{1}{|\Omega(\Pi)|}$. L'indice ottenuto ha dunque la forma:

$$\gamma(\Pi; r) = \left(\frac{1}{|\Omega(\Pi)|} \sum_{\lambda \in \Omega(\Pi)} \gamma(\lambda)^r \right)^{1/r} \tag{3.1}$$

dove λ è una generica estensione lineare di Π.

Le $\gamma(\lambda)$ possono dunque essere aggregate in indici descrittivi, ma è possibile anche considerarle come costituenti una distribuzione. Le funzioni implementate nel software presentano infatti la possibilità di calcolare i *quantili* di questa distribuzione dei valori di $\gamma(\cdot)$ sull'insieme $\Omega(\Pi)$

Una considerazione finale: poiché nella definizione delle proprietà richieste da un funzionale su un poset e nella conseguente scelta di utilizzare le famiglie generatrici NOC non è stato necessario fare uso della loro proprietà di *completezza* ci si potrebbe giustamente chiedere se non sia possibile applicare questa teoria assiomatica per calcolare indici su famiglie generatrici *non-overlapping* ma *non complete* come l'insieme delle estensioni lineari lessicografiche. Un lavoro in tal senso è svolto in [15], di prossima pubblicazione, nel dominio della misura della povertà. Sono indubbiamente necessarie ulteriore analisi e ricerche per ottenere una teoria assiomatica altrettanto completa e per verificare se e quanto la costruzione di un indice su un insieme così ristretto di estensioni lineari possa *distorcere* l'indicatore calcolato, anche in ambiti diversi.

3.3 Indicatori statistici implementati

Gli indici implementati nel pacchetto sono riportati nella tabella 3.1, indicando con M il numero di profili, con p_i la frequenza relativa del profilo i data la distribuzione di

frequenza **p** sul poset, con P_i la frequenza relativa cumulata nell'estensione lineare, con m la mediana all'interno dell'estensione lineare e con n il numero di unità statistiche:

Autore/i	*Indice normalizzato*
Abul Naga, Yalcin [1]	$I_{AY}(\alpha,\beta) = \frac{\sum_{i=1}^{m-1} P_i^{\alpha} - \sum_{i=m}^{M} P_i^{\beta} + (M+1-m)}{(m-1)\cdot 0.5^{\alpha} - (M-m)\cdot(0.5^{\beta}-1)}$ $(\alpha,\beta \leq 1)$
Apouey [2]	$I_{AP}(\alpha) = 1 - \frac{2^{\alpha}}{M-1}\sum_{i=1}^{M-1} \lvert P_i - 0.5\rvert^{\alpha}$ $(0 < \alpha < 1)$
Blair, Lacy (1) [5], Reardon (2) [29]	$I_{BL1} = I_{RE2} = \frac{4}{M-1}\sum_{i=1}^{M-1} P_i(1-P_i)$
Blair, Lacy (2) [5]	$I_{BL2} = 1 - \sqrt{\frac{4}{M-1}\sum_{i=1}^{M-1}(P_i - 0.5^2)}$
Kobus, Miłoś [22]	$I_{KM}(a,b) = 2\left(\frac{a\sum_{i=1}^{m-1} P_i - b\sum_{i=m}^{M} P_i + b(M+1-m)}{a(m-1)+b(M-m)}\right)$ $(a,b>0)$
Kvålseth [23]	$I_{KV} = 1 - \sqrt{1-\Delta}$ dove $\Delta = \frac{2}{M-1}\sum_{i=1}^{M}\sum_{j=1}^{M} p_i p_j \lvert i-j\rvert$
Leik [24]	$I_{LK} = \frac{2}{M-1}\left(\sum_{i=1}^{m-1} P_i + \sum_{m}^{M-1}(1-P_i)\right)$
Leti [25], Berry, Mielke [4]	$I_{BM} = \frac{4k}{M-1}\sum_{i=1}^{M-1} P_i(1-P_i)$ dove $k = \begin{cases} 1 & n \text{ pari} \\ \frac{n^2}{n^2-1} & n \text{ dispari} \end{cases}$
Reardon (1) [29]	$I_{RE1} = -\frac{1}{M-1}\sum_{i=1}^{M-1}\left[P_i \log_2 P_i + (1-P_i)\log_2(1-P_i)\right]$
Reardon (3) [29]	$I_{RE3} = \frac{2}{M-1}\sum_{i=1}^{M-1}\sqrt{P_i(1-P_i)}$
Reardon (4) [29]	$I_{RE4} = 1 - \frac{1}{M-1}\sum_{i=1}^{M-1} \lvert 2P_i - 1\rvert$

Tabella 3.1 – Indici statistici implementati

In particolare, nell'analisi empirica effettuata, si utilizzerà l'indice di Leti poiché possiede alcune proprietà che rendono possibile una sua scomposizione negli effetti delle singole variabili [19]. Il tema non verrà affrontato nel presente lavoro, ma lo si ritiene estremamente interessante in ottica di suoi sviluppi futuri.

Capitolo 4

Difficoltà computazionali e soluzioni algoritmiche

In questo capitolo vengono affrontate questioni di algoritmica e informatica pura, pertanto si ritiene necessario fornire una breve introduzione alla notazione che verrà utilizzata nel proseguio del lavoro.

Con *"tempo di esecuzione di un algoritmo"* si intende il numero di istruzioni elementari necessarie a un calcolatore per eseguire l'algoritmo dall'inizio alla fine. Sia $T(n)$ il tempo di esecuzione di un particolare algoritmo T su un input di dimensione $n > 0$. Si dirà che $T(n) = O(f(n))$ se $\exists\ n_0 \in \mathbb{N}, c > 0 : \forall n > n_0\ T(n) \leq c \cdot f(n)$. Ossia la funzione $f(n)$ è un limite asintotico superiore.

Si dirà che $T(n) = \Omega(g(n))$ se $\exists\ n_o \in \mathbb{N}, c > 0 : \forall n > n_0\ T(n) \geq c \cdot g(n)$. $g(n)$ è dunque un *limite inferiore* al tempo di esecuzione dell'algoritmo.

Si dirà infine che $T(n) = \Theta(h(n))$ se $\exists\ n_0 \in \mathbb{N}, c_1, c_2 > 0 : \forall n > n_0\ c_1 \cdot h(n) \leq T(n) \leq c_2 \cdot h(n)$. In questo caso $T(n)$ e $h(n)$ hanno lo stesso ordine di grandezza e $h(n)$ si dice *limite asintotico stretto.*

Si è soliti assumere che $T(n)$ indichi il tempo di esecuzione *nel caso peggiore*, ossia il massimo tempo impiegabile dall'algoritmo sul più *"scomodo"* degli input. Se si conoscono dettagli a priori sulla distribuzione degli input all'interno dello spazio dei casi possibili, si potrebbe essere più interessanti ad una funzione di n che descriva il tempo *medio*

di esecuzione. In questo lavoro tuttavia non si affronterà il problema della complessità asintotica *media* di un algoritmo.

Si può operare una suddivisione dei problemi, seppur decisamente poco formale, in base alla loro. Se un problema richiede un algoritmo il cui tempo di esecuzione è $O(f(n))$ esso è considerabile *semplice* o trattabile se $f(n)$ è una funzione polinomiale o lineare in n, ed è invece considerabile *difficile* o *intrattabile* se $f(n)$ è di un ordine di grandezza superiore (esponenziale, fattoriale, ...). Ovviamente esistono molti problemi difficili per cui sono sviluppati algoritmi che sono utilizzati nel quotidiano: la loro intrattabilità sta nel fatto che per particolari input *"scomodi"*, essi possono richiedere tempi di esecuzione astronomici, rendendoli all'atto pratico equivalenti a problemi effettivamente incomputabili.

Per una trattazione completa della teoria della complessità algoritmica si rimanda a [10].

4.1 Enumerazione delle sequenze lineari di un poset

Nel capitolo precedente è risultato più volte necessario utilizzare la cardinalità dell'insieme delle estensioni lineari di Π, $\Omega(\Pi)$, come nella formula 3.1. Per quanto possa sembrare banale in quanto puro esercizio di calcolo combinatorio, anche solo numerare e definire dunque questa cardinalità è un problema molto difficile.

Nella formula 2.2 è stato fonito un limite inferiore per calcolare questa cardinalità in poset del tipo $\mathbf{2}^k$. Per quanto elevato, quel numero è comunque calcolabile. Ma esistono poset ben più complicati di $\mathbf{2}^k$.

È stato dimostrato [6] che *enumerare* le estensioni lineari di un generico poset è un problema *#P-completo*, e ciò implica che non esiste un algoritmo eseguibile in tempo polinomiale per calcolare la cardinalità di $\Omega(\Pi)$.

La classe dei problemi #P-completi contiene problemi di conteggio *non polinomiali* e ciascuno è associato a un problema decisionale[4]. Particolare attenzione va posta nella

[4]Un problema decisionale è un qualsiasi problema il cui output richiesto sia di tipo booleano.

trivialità del corrispondente problema duale: *"Il poset* Π *possiede almeno una estensione lineare?"*, che è banalmente sempre vero perché ogni poset ammette una linearizzazione. Questo è dunque un caso tipico in cui il problema decisionale è banale mentre il corrispettivo problema di conteggio è *estremamente* complesso.

4.2 Estrazione uniforme di sequenze lineari

Considerare l'insieme $\Omega(\Pi)$ per intero è quindi impraticabile. È necessario perciò un metodo che permetta di *campionare* fra le estensioni lineari di un poset, ed è ovviamente *fondamentale* che tale campionamento sia uniforme. I metodi più utilizzati in letteratura ([11], [20]) fanno utilizzo di *catene di Markov* per campionare le estensioni. Una catena di Markov è un processo stocastico la cui transizione dal tempo t al tempo $t+1$ è determinata solo dallo stato al tempo t. Si darà ora una definizione di catena di Markov $\mathcal{M}_f$ su $\Omega(\Pi)$.

Sia $\sigma(i,j)$ un operatore che applicato a una sequenza ordinata scambia l'elemento di posto i con quello di posto j e sia f una distribuzione di probabilità discreta concava su $\{1,2,\ldots,n-1\}$. Se lo stato corrente è $X_t \in \Omega(\Pi)$ per un qualsiasi $t \geq 0$, lo stato X_{t+1} è determinato dal seguente algoritmo:

Algoritmo 1 Catena di Markov $\mathcal{M}_f$

1: si generi $p \in \{1,2,\ldots,n-1\}$ secondo la distribuzione f
2: si scelga $c \in \{0,1\}$ secondo una distribuzione uniforme
3: **if** $(c = 0$ **or** $\sigma(p,p+1)X_t \notin \Omega(\Pi))$ **then** $X_{t+1} \leftarrow X_t$
4: **else**$X_{t+1} \leftarrow \sigma(p,p+1)X_t$
5: **end if**

Una catena di Markov che ha valori su un insieme X si dice *ergodica* se

$$\exists k > 0 : P\left(X_{t+k} = x | X_t\right) \neq 0, \forall t > 0, \forall x \in X$$

La catena $\mathcal{M}_f$ è senza dubbio ergodica ed è anche *stazionaria* poiché *simmetrica*: la probabilità di passare da X_t a X_{t+1} è uguale alla probabilità di passare da X_{t+1} a X_t. Queste condizioni sono sufficienti per far sì che la distribuzione di probabilità sull'insieme supporto derivata dalle estrazioni di $\mathcal{M}_f$ converga alla distribuzione uniforme per $t \to \infty$.

La *distanza totale di variazione* di due distribuzioni di probabilità definite su uno spazio degli eventi X è data da $d_{tv} = \frac{1}{2}\sum_{x\in X}|P(x) - Q(x)|$. La *precisione* ϵ di una catena di Markov $\mathcal{M}$ è definita come limite superiore alla distanza totale di variazione fra la distribuzione di probabilità osservata e la distribuzione uniforme. Il *tempo di mescolmento* $\tau(\epsilon)$ di una catena di Markov è il numero di transizioni necessarie per raggiungere una precisione di ϵ.

La particolare catena di Markov $\mathcal{M}_f$ definita su $\Omega(\Pi)$ con f distribuzione uniforme è chiamata catena di Karzanov-Khachiyan e possiede tempo di mescolamento pari a $O\left(n^4 \log^2 n + n^3 \log n \log \epsilon^{-1}\right)$ [7], con ϵ precisione desiderata dalla catena, ed in questo caso il tempo di mescolamento è equivalente alla complessità computazionale dell'algoritmo. Tale catena genera un campionamento casuale uniforme sullo spazio delle estensioni lineari $\Omega(\Pi)$.

Bubley e Dier (1999) [7] [8], utilizzando metodi più raffinati, hanno dimostrato che il numero di iterazioni necessarie è pero diminubile definendo nell'algoritmo 1, $f : i \to i(n-i)/K$, dove $K = (n^3 - n)/6$ è la costante di normalizzazione, ed esso scende a $O(n^3 \log n\epsilon^{-1})$.

Il numero $n^3 \log n\epsilon^{-1}$ può però superare il numero intero più grande gestibile da un processore a 32 bit, ossia 2^{32}. Per garantire la compatibilità del software utilizzato si divide dunque il procedimento in blocchi ciascuno con numero di iterazioni minore di 2^{32} aggregando i risultati di ciascun blocco solamente alla fine del processo. Questo limite tecnico renderà necessari alcuni accorgimenti in fasi successive.

4.3 Estrazione di numeri casuali in $\mathbb{N}$ data una distribuzione di probabilità discreta

Avendo definito una distribuzione *target* specifica per generare numeri casuali ed ottimizzare dunque il tempo di mescolamento della catena di Markov in questione, è necessario trovare un metodo computazionalmente efficace per estrarli da una data la distibuzione di probabilità discreta f. L'algoritmo *naïve* è il cosiddetto metodo dell'inversione che consiste nel generare un numero casuale $Y \in [0,1]$, un'operazione che richiede un tempo

irrisorio per qualsiasi calcolatore, e poi nel calcolare $f^{-1}(Y)$ che, per una distribuzione discreta consiste nel calcolare le frequenze relative cumulate $\{F_1, F_2, \ldots, 1\}$ e nel trovare il primo indice $i : F_i \geq Y$. Il numero estratto sarà dunque i. L'inizializzazione della lista di frequenze relative cumulate ha complessità di $O(n)$ mentre l'operazione di ricerca dell'indice i dato Y è operabile tramite una ricerca binaria ed ha complessita pari a $O(\log n)$. Si considera dunque questa complessità come baseline e si trascura qualsiasi algoritmo con complessità uguale o maggiore.

Algoritmo 2 *Alias method* per l'estrazione di numeri casuali da una distribuzione di probabilità discreta p

function INITIALIZATION($p[n]$)
    si creino due array, $Alias[\,]$ e $Prob[\,]$, di dimensione n
    si creino altri due array, $Small[\,]$ e $Large[\,]$ di dimensione dinamica
    si moltiplichi ciascuna probabilità in p per n
    for ogni probabilità riscalata in p **do**
        if $p[i] < 1$ **then** si aggunga i a a $Small$
        else si aggiunga i a $Large$
        end if
    end for
    while (Small $\neq \emptyset \wedge$ Large $\neq \emptyset$) **do**
        si rimuova il primo elemento di Small e lo si denoti con l
        si rimuova il primo elemento di Large e lo si denoti con g
        $Prob[l] \leftarrow p[l]$
        $Alias[l] \leftarrow g$
        $p[g] \leftarrow p[g] + p[l] - 1$
        if $p[g] < 1$ **then** si aggiunga g a $Small$
        else si aggiunga g a $Large$
        end if
    end while
    while $Large \neq \emptyset$ **do**
        si rimuova il primo elemento di $Large$ e lo si denoti con g
        $Prob[g] \leftarrow 1$
    end while
    while $Small \neq \emptyset$ **do** ▷ Passaggio necessario per motivi di stabilità numerica
        si rimuova il primo elemento da $Small$ e lo si denoti con l
        $Prob[l] \leftarrow 1$
    end while
    return Alias, Prob
end function

function GENERATION($Alias[n], Prob[n]$)
    si generi un numero casuale in $\{1, \ldots, n\}$ con probabilità uniforme e lo si denoti con i
    si generi un numero in $\{0, 1\}$ con probabilità associate $\{1 - Prob[i], Prob[i]\}$ e lo si denoti con X
    if $X = 1$ **then return** i ▷ Questo è il numero casuale desiderato
    else return $Alias[i]$
    end if
end function

Il metodo più efficiente trovato in letteratura è quello di *Vose* [35], chiamato *alias*

method, il cui funzionamento è riportato nell'algoritmo 2. Intuitivamente, il procedimento è il seguente [31]:

1. Si costruiscono dei rettangoli $(n \cdot p_i) \times 1$ per ogni probabilità p_i.
2. Si tagliano orizzontalmente i rettangoli in blocchi più piccoli.
3. Si ridistribuiscono i pezzi nelle n colonne in modo che :
   - Ogni colonna abbia altezza unitaria.
   - In ogni colonna non siano presenti più di due blocchi appartenenti a rettangoli diversi.
   - Almeno un blocco corrispondente al rettangolo i si trova nella colonna i.
4. Si tiene traccia degli spostamenti attraverso due tabelle: la *Prob Table*, che indica per la colonna i la percentuale di area occupata dai blocchi del rettangolo i nella colonna, e la *Alias Table*, tiene traccia per la colonna i del rettangolo di appartenenza del blocco diverso da i presente nella colonna.
5. Si lancia una freccetta immaginaria casualmento all'interno del piano ottenuto e il numero estratto corrisponde al rettangolo di appartenenza del blocco su cui è finita la freccetta.

L'iniziazlizzazione delle due tabelle ha complessità di $O(n \log n)$, mentre la generazione ha complessità di $\Theta(1)$. La generazione è dunque praticamente istantanea mentre il passaggio preliminare, che però viene eseguito solo una volta, richiede più tempo rispetto all'algoritmo naïve.

Poiché la quantità di numeri casuali da generare è pari al limite asintotico del tempo di mescolamento della catena di Markov descritta nel capitolo precedente, ossia $n^3 \log n\epsilon^{-1}$ e questo numero può essere molto elevato, il maggior tempo dedicato alla inizializzazione degli array nell'algoritmo di Vose viene recuperato con un tempo di generazione *sensibilmente* più basso rispetto a qualsiasi altro algoritmo esaminato che moltiplicato per la quantità di numeri richiesti offre un notevole vantaggio computazionale.

4.4 Stima dinamica della varianza e dei quantili

Volendo descrivere una *distribuzione continua* a partire dai valori degli indici calcolati sulle singole estensioni lineari, è importante potere sintetizzare tale distribuzione ed estrarne alcune caratteristiche fondamntali quali la varianza e i quantili. Non è però possibile mantenere i singoli valori degli indici in memoria poiché il loro numero può essere elevatissimo e creare seri problemi a un calcolatore con media disponibilità di memoria.

Sono dunque necessari procedimenti dinamici che aggiornino il valore di varianza e quantili ad ogni iterazione.

Per la varianza è utilizzato il metodo di *Chan et al.* [9] che ad ogni interazione aggiorna la somma dei quadrati nel seguente modo:

$$M_{2,n} = M_{2,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n)$$

Poiché come già detto è necessario suddividere i calcoli in blocchi per limiti tecnici, è necessario aggiornare la somma dei quadrati anche fra un blocco di istruzini e l'altro e non solo all'interno dello stesso blocco. *Chan et al.* generalizzano la formula di sopra per qualsiasi partizione del campione X nei sottoinsiemi X_A e X_B:

$$\delta = \bar{x}_B - \bar{x}_A$$

$$M_{2,X} = M_{2,A} + M_{2,B} + \delta^2 \cdot \frac{n_A n_B}{n_X}$$

È importante menzionare anche che, dato l'elevato numero di calcoli richiesto, tutti gli indici vengono normalizzati solo dopo l'ultimo blocco di iterazioni e dunque anche la varianza risulterebbe non normalizzata. Questo non è un problema perché il metodo di *Chan* aggiorna solamente la somma dei quadrati riducendo notevolmente la mole di divisioni, l'operazione elementare più onerosa computazionalmente.

Per i quantili è invece utilizzato l'algoritmo P^2*-esteso* di *Raatikainen* [28] per stimare m quantili, una generalizzazione dell'algoritmo *P-square* in [21], che fornisce una *stima* sufficientemente accurata dei quantili desiderati e il cui funzionamento è descritto nell'algoritmo 3 in appendice A.

L'algoritmo è piuttosto lungo ma non eccessivamente complesso e si basa sul mantenere una distanza costante predefinita fra i quantili stimati, aggiornando le stime ad ogni

iterazione inserendo il nuovo indice calcolato all'interno della sequenza di $2m + 3$ punti ordinati precedentemente e aggiornando le posizioni degli altri punti per mantenere tale distanza. Al termine della procedura, l'array $\hat{q}$ contiene i quantili desiderati.

Il calcolo dei quantili aggiunge una notevole complessità computazionale dato l'elevato numero di operazioni all'interno dell'algoritmo P^2 e la procedura viene ritenuta per questo non soddisfacente. Non sono però stati trovati in letteratura, per ora, algoritmi più eleganti ed efficienti per questo calcolo dinamico.

Capitolo 5

Implementazione in parsec

parsec [3], che sta per *PARtial orders in Socio-EConomics*, è una libreria per il software statistico R [27] sviluppata dal Dott. Alberto Arcagni con la collaborazione del Dott. Marco Fattore ed una delle pochissime soluzioni software che implementano gli insiemi parzialmente ordinati. Il pacchetto è focalizzato soprattutto sulla misura della severità della povertà e presenta numerose funzioni utili a tale scopo che sono state utilizzate da Fattore e Arcagni in tutte le loro pubblicazioni.

Oltre a un backend in R molto efficace per la gestione delle matrici e per le rappresentazioni grafiche, il pacchetto si avvale in maniera consistente della API C di R, attraverso l'interfaccia *.C* per il campionamento delle estensioni lineari e per il calcolo di misure ed indici su di esse.

La scelta del linguaggio C per questo fine è dettata dalla necessità di ottimizzare le prestazioni quanto più possibile data l'enorme mole di calcoli ripetuti richiesta dagli algoritmi di campionamento uniforme sullo spazio delle estensioni lineari di un poset e dal calcolo degli indici. Ultimamente sono nate soluzioni per interfacciarsi in maniera *seamless* fra R e C++, come la libreria *RCPP*. Si è preferito non utilizzare tale interfaccia nello sviluppo del pacchetto per la maggiore stabilità delle API C native di R e per facilità di sviluppo, a scapito di un leggero peggioramento delle prestazioni.

Nell'ultima versione rilasciata del pacchetto su *CRAN* non è implementato l'algoritmo *ottimo* di Bubley-Dier ma una versione modificata della catena di Karzanov-Khachiyan

(algoritmo 1) che ha complessità di $O(n^5 \log n + n^4 \log \epsilon^{-1})$.

L'obiettivo del lavoro è stato dunque quello di implementare l'algoritmo ottimo di *Bubley-Dier* per il campionamento delle estensioni lineari in linguaggio C oltre a funzioni in R a livello più alto che permettano di calcolare gli indici definiti nella sezione 3.3 sempre avvalendosi di funzioni a basso livello in C.

Una nuova versione del pacchetto comprendente le modifiche e le aggiunte effettuate nel contesto del presente lavoro verrà rilasciata a breve.

5.1 Implementazione dell'algoritmo di Bubley e Dier

Per implementare correttamente l'algoritmo di Bubley e Dier è necessario campionare casualmente dalla distribuzione $f : i \to i(n-i)/K$ dove $K = (n^3 - n)/6$. È dunque necessario implementare in qualche modo l'alias method descritto nella sezione 4.3.

Fortunatamente, la libreria C *ransampl* [36] fornisce una implementazione molto efficace dell'alias method che inoltre non necessità di probabilità normalizzate in ingresso, ottimizzando ulteriormente il processo.

Il vettore di probabilità viene dunque creato in R utilizzando le funzioni vettoriali proprie del linguaggio e viene trasmesso tramite l'interfaccia *.C* alla parte di codice C in input, per poi utilizzare le funzioni della libreria *ransampl* per inizializzare gli ogetti necessari e per generare ad ogni iterazione una nuova estensione lineare.

In alcuni casi particolari, la catena di Karzanov-Khachiyan può richiedere meno iterazioni dell'algoritmo di Bubley-Dier. Si consideri a questo scopo la funzione:

$$f(n, \epsilon) = n^5 \log n + n^4 \log \epsilon^{-1} - n^3 \log n\epsilon^{-1}, n \in \mathbb{N} \setminus \{0, 1\}$$

Ovviamente, $f(n, \epsilon) < 0$ se la catena di Karzanov-Khachyan richiede meno iterazioni rispetto a quella di Bubley-Dier. Fissando $\epsilon = 0.001$ come valore standard, si noti che:

$$f(2, \epsilon) < 0$$

$$\frac{\partial f}{\partial n} > 0 \iff n > 7$$

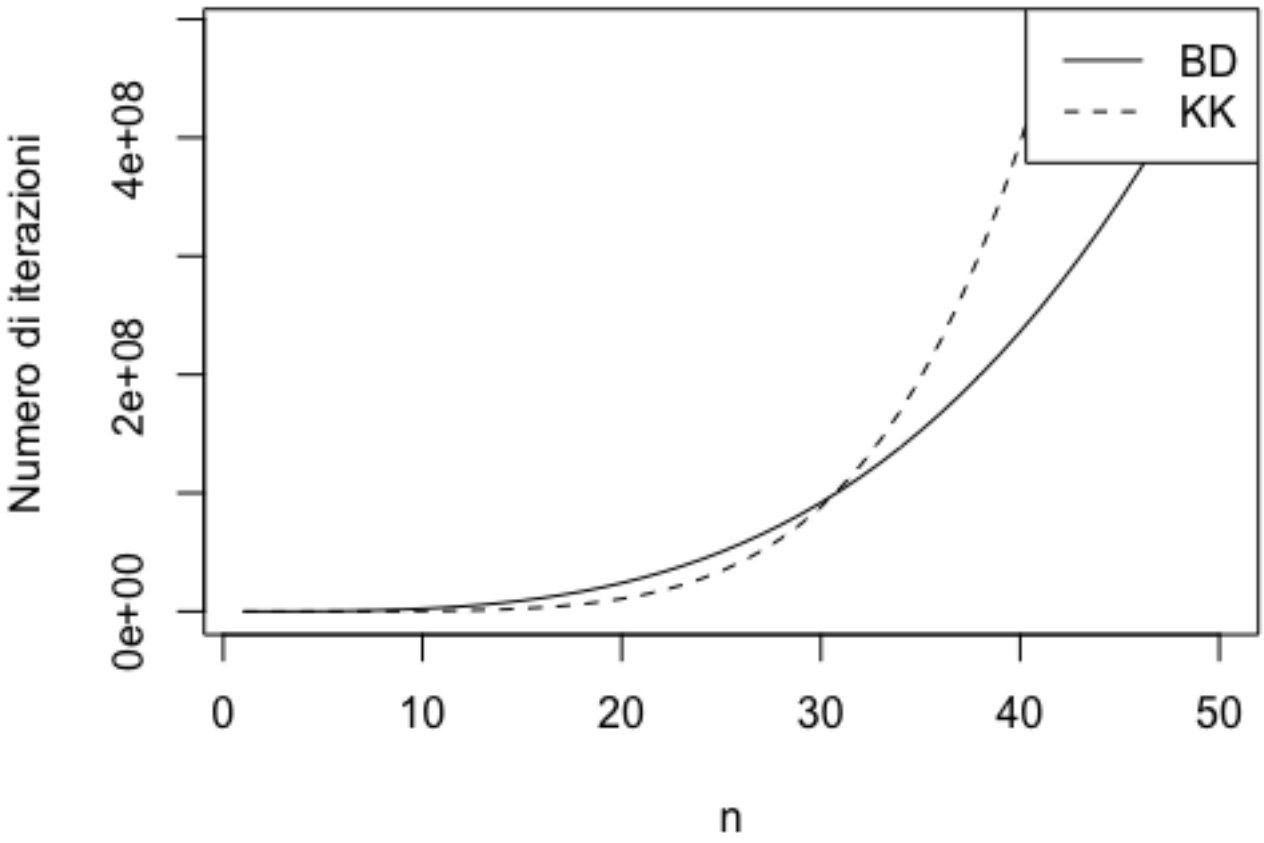

Figura 5.1 – Confronto fra il numero di iterazioni necessarie alle catene di Bubley-Dier (BD) e Karzanov-Khachiyan (KK), dato $\epsilon = 0.001$

Dunque per poset di piccole dimensioni l'algoritmo generalmente meno efficace richiede meno iterazioni. Per $\epsilon = 0.001$ in particolare:

$$f(n, \epsilon) < 0 \iff n > 30$$

dove il risultato è stato ottenuto per via numerica. Il pacchetto dunque, dato ϵ fissato dall'utente, trova $n : f(n, \epsilon) = 0$ per via numerica tramite la funzione *uniroot* presente nella libreria standard *stats* di R, se esiste, e se il numero di profili osservati $n_{oss} < n$,

allora utilizza la catena di Karzanov-Khachiyan per la generazione delle estensioni lineari, altrimenti quella di Bubley-Dier. L'utente esperto può altresì specificare quale catena desidera utilizzare manualmente attraverso un argomento delle funzioni R.

5.2 Implementazione delle misure di disugaglianza sociale

Per implementare le funzioni descritte nella sezione 3.3 tenendo conto della logica di ordinamento degli attributi descritta nella sezione 2.2 è stato dapprima necessario combinare le funzioni già presenti in parsec per ottenere, a partire dal poset osservato e da un ordinamento sulle variabili, l'estensione del poset originale compatibile con l'ordinamento Λ.

Data una matrice di incidenza λ ottenuta dall'ordinamento sulle variabili espresso attraverso la funzione *getlambda* di *parsec*, si estraggono le estensioni lineari del poset Λ attraverso la funzione $LE(\lambda)$.

Successivamente, attraverso la funzione *LE2incidence* si estraggono le estensioni lineari lessicografiche del poset originale compatiibili con l'ordinamento espresso dalla matrice λ.

Si opera in seguito come descritto nella sezione 1.2, moltiplicando elemento per elemento in sequenza le matrici di incidenza associate alle estensioni lessocografiche compatibili, ottenendo l'estensione Π^* del poset Π. Una volta ottenuta questa struttura, è possibile calcolarvici gli indici della sezione 3.3.

Le singole formule per il calcolo degli indici sono definite separatamente all'interno del codice C e inserite in un puntatore. L'indice, o gli indici, da calcolare sono scelti dall'utente tramite un argomento della funzione R che accetta una stringa o un vettore di stringhe indicante uno o più dei nomi predeifiniti degli indicatori implementati che vengono poi convertite in indici corrispondenti alle posizioni delle funzioni all'interno del puntatore nel codice C.

Ad ogni iterazione vengono aggiornate dinamicamente la media, la somma dei quadrati (tramite cui si ricaverà la varianza) e, se esplicitato dall'utente, le stime di quartili e decili. Ognuna di queste misure non è normalizzata: ad ogni indicatore sono associati le rispettive

costanti moltiplicative e additive secondo le quali le misure sono traslate solo al termine di tutte le iterazioni per diminuire il carico computazionale.

Ad ogni iterazione, escluse le prime necessarie per inizializare il calcolo dei quantili, la procedura è dunque la seguente:

1. Si passa allo stato successivo della catena di Bubley-Dier avvalendosi dell'estrazione di un numero casuale tramite l'alias method basandosi su una distribuzione di probabilità definita a priori nella parte di codice in R

2. Se è stata generata una nuova estensione, si calcola l'indice (o gli indici) di riferimento su di essa

3. Si aggiornano le stime della media e della somma dei quadrati

4. Si applica un'iterazione dell algoritmo P^2 di *Raatikainen* (algoritmo 3) aggiornando le stime dei quantili

Nel complesso, la struttura del pacchetto è ritenuta soddisfacente ed il codice risulta ottimizzato per le operazioni richieste. Per motivi di pura ingegneria del software e di logica di sviluppo, si potrebbe provare a sostituire l'interfaccia *.C* utilizzata con l'interfaccia *.Call*, che permette una gestione migliore della comunicazione fra R e C, agevolando la scrittura del codice e permettendo di evitare alcuni artifici necessari per l'utilizzo dell'API *.C*[5] a scapito di una minore intelleggibilità del codice, poiché *.Call* presenta una sintassi elaborata ma dedicata alle operazioni con R. Si rimandano perciò tali sviluppi ad una prossima release del pacchetto.

[5]Tutte le funzioni C chiamate da *.C* devono essere puntatori a *void* e l'unico modo perché ritornino dei valori è dunque modificare i loro input *inplace*.

Capitolo 6

Applicazioni e conclusioni

6.1 Applicazione a dati reali

In questo capitolo verranno mostrati i risultati di un'applicazione delle funzioni e degli algoritmi presentati precentemente su dati provenienti da un censimento effettuato nella Repubblica Democratica del Congo nel 2007 fra i bambini da 0 a 17 anni [26]. Gli stessi dati sono analizzati nell'ottica della misura della severità della povertà in [15]. Ci si prefigge dunque di fornire un'altra chiave di lettura prendendo in esame i fenomeni sociali della disugaglianza e della polarizzazione.
I dati sono costituiti da quattro attributi dicotomici:

- *Water*, che assume valore 1 per i bambini che hanno accesso a una fonte di acqua pulita entro 15 minuti di cammino dalla loro abitazione.

- *Sanitation*, che assume valore 1 per i bambini che hanno accesso a latrine o bagni attrezzati adeguatamente.

- *Shelter*, che assume valore 1 per i bambini che possiedono un pavimento nella loro abitazione, in cui risiedono meno di 5 persone per ogni stanza.

- *Health*, che assume valore 1 per i bambini che hanno accesso a cure mediche entro 15 minuti di cammino dalla loro abitazione.

Le variabili forniscono dunque $2^4 = 16$ profili, elencati nella tabella 6.1 a cui è associata una distribuzione di frequenza per ognuna delle 11 suddivisioni regionali della Repubblica Democratica del Congo[6] ed una ulteriore distribuzione riferita all'aggregato nazionale, riportate nella tabella 6.2.

ID profilo	Water	Sanitation	Shelter	Health
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	0	0
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

Tabella 6.1 – Profili osservati nel censimento

[6] *KSS*: Kinshasa, *BCO*: Bas-Congo, *BDD*: Bandundu, *ETR*: Équateur, *ORT*: Orientale, *NKV*: Nord Kivu, *MNM*: Maniema, *SKV*: Sud Kivu, *KTG*: Katanga, *KOT*: Kasai Orientale, *KOC*: Kasai Occidentale

ID Profilo	National	KSS	BCO	BDD	ETR	ORT	NKV	MNM	SKV	KTG	KOT	KOC
1	0.31	0.03	0.31	0.34	0.57	0.47	0.40	0.19	0.22	0.22	0.17	0.36
2	0.15	0.03	0.24	0.40	0.16	0.08	0.07	0.11	0.10	0.06	0.07	0.22
3	0.03	0.02	0.01	0.11	0.01	0.01	0.01	0.01	0.02	0.01	0.01	0.02
4	0.01	0.03	0.02	0.02	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.01
5	0.14	0.01	0.06	0.07	0.13	0.25	0.08	0.35	0.16	0.10	0.24	0.16
6	0.06	0.01	0.10	0.02	0.05	0.04	0.04	0.17	0.05	0.07	0.11	0.11
7	0.01	0.02	0.03	0.00	0.00	0.00	0.02	0.01	0.03	0.00	0.04	0.01
8	0.01	0.02	0.02	0.00	0.01	0.00	0.02	0.02	0.03	0.00	0.00	0.01
9	0.04	0.04	0.01	0.02	0.01	0.03	0.15	0.03	0.11	0.06	0.02	0.03
10	0.03	0.06	0.02	0.01	0.00	0.01	0.10	0.03	0.03	0.06	0.03	0.06
11	0.02	0.06	0.01	0.00	0.00	0.02	0.03	0.01	0.05	0.02	0.02	0.00
12	0.02	0.12	0.08	0.00	0.00	0.00	0.01	0.00	0.01	0.02	0.01	0.01
13	0.04	0.04	0.00	0.00	0.01	0.04	0.01	0.03	0.04	0.10	0.08	0.00
14	0.04	0.09	0.02	0.00	0.00	0.03	0.01	0.02	0.04	0.08	0.12	0.00
15	0.03	0.12	0.02	0.00	0.01	0.01	0.03	0.00	0.04	0.07	0.05	0.00
16	0.06	0.31	0.06	0.00	0.00	0.01	0.01	0.00	0.07	0.13	0.3	0.00

Tabella 6.2 – Distribuzioni di frequenza dei profili osservati

Poiché le probabilità riferite ad alcune colonne della tabella 6.2 non sommano a 1, e poiché ciò dipende da errori di arrotondamento a monte presenti in [26] e non si hanno a disposizione i dati del rilevamento originale, è stata operata una procedura di normalizzazione delle colonne, che non inficia in alcun modo la bontà delle analisi, mantenendo invariati i rapporti fra le probabilità.

Per utilizzare la procedura sviluppata è necessario anche imporre un ordinamento sulle variabili, ottenendo il poset Λ. La procedura è stata ripetuta con due possibili ordinamenti. Nel primo è stato imposto un ordinamento sulla base di assunzioni sociologiche a priori con il consiglio del Dott. Fattore costruendo $\Lambda_1 = (\mathit{Water} > \mathit{Sanitation}, \mathit{Health} > \mathit{Sanitation}, \mathit{Sanitation} > \mathit{Shelter})$, includendo ovviamente anche le compatibilità indotte dalla transitività della relazione, e ottenendo l'estensione riportata in figura 6.1 del poset $\mathbf{2}^4$. Nel secondo caso, il poset Λ_2 è un'antichain: si assume quindi che le variabili non siano comparabili fra di loro e abbiano la stessa importanza. Il poset possiede quindi solo il consueto ordinamento prodotto. Nella tabella 6.3 vengono mostrati i risultati delle due differenti valutazioni per ciascun aggregato, utilizzando come metrica l'indice di Leti [25].

Oltre alle medie è importante evidenziare come le varianze, non riportate, siano tutte

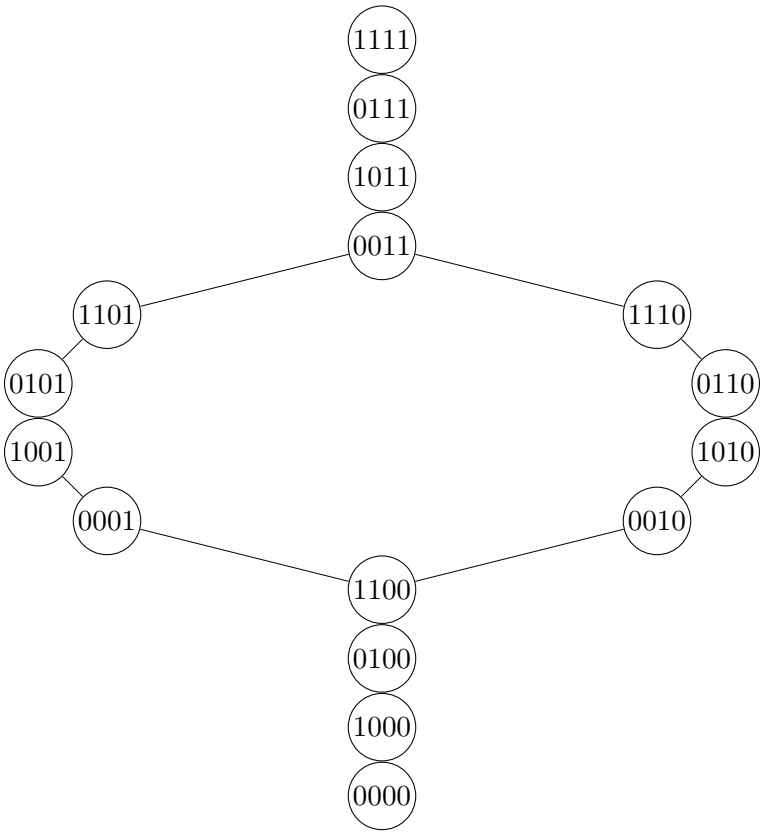

Figura 6.1 – Estensione del poset $\mathbf{2}^4$ ottenuta applicando l'ordinamento Λ_1 alle variabili

Regione	Λ_1	Λ_2
National	0.715	0.685
KSS	0.648	0.648
BCO	0.684	0.720
BDD	0.206	0.274
ETR	0.325	0.323
ORT	0.526	0.442
NKV	0.575	0.566
MNM	0.465	0.477
SKV	0.736	0.717
KTG	0.851	0.823
KOT	0.728	0.697
KOC	0.405	0.441

Tabella 6.3 – Media degli indici di Leti sul poset identificato dalla tabelle 6.1 e 6.2

molto basse e che i decili siano molto vicini fra loro: l'elevatissimo numero di estensioni lineari campionate fa concentrare la distribuzione dell'indice attorno al valore medio in maniera evidente.

Non si commenterà sui valori assoluti assunti dagli indici in quanto sarebbe necessaria una approfondita conoscenza sociologica del fenomeno e si rimandano dunque tali considerazioni a futuri sviluppi del lavoro. Si noti però come le medie variano passando da Λ_1 a Λ_2, dimostrando quindi che le assunzioni a monte sull'ordinamento delle variabili sono sensate.

6.2 Sviluppi futuri e conclusioni

Il lavoro ha fornito basi teoriche e software molto importanti che permettono applicazioni su larga scala, oltre ad essere una delle prime vere applicazione della logica multidimensionale nelll'ambito della polarizzazione sociale.

La procedura potrebbe essere sicuramente affinata per esempio considerando le sole estensioni lessicografiche, come riportato in [15], e lo sviluppo software continuerà certamente andando passo passo con la ricerca e cercando di fare in modo che la complessità computazionale di alcuni dei problemi affrontati non vi sia mai di intralcio, per quanto possibile.

Inoltre, approfondendo la conoscenza sociologica del fenomeno o collaborando con esperti nel settore, grazie alla metodologia sviluppata sarebbe possibile evincere conclusioni molto importanti che potrebbero anche smentire risultati ottenuti con metodologie a carattere unidimensionale. Si considera perciò questo lavoro solo come un punto di inizio della ricerca in tale senso.

La nuova versione di *parsec*, di cui l'autore è *contributor*, verrà rilasciata a breve in maniera open, e se ne auspica un sempre maggiore utilizzo.

Appendice A

Algoritmo P^2

Algoritmo 3 Algoritmo P^2 per la stima dinamica di quantili ai punti $p_1, \ldots, p_m$: $0 < p_1 < p_2 < \cdots < p_m < 1$

procedure INITIALIZATION($X[2m+3], q[2m+3], n[2m+3], f[2m+3], d[2m+3]$)
  $q[i] \leftarrow X[i]$ ▷ $i = 1, \ldots, 2m+3$ altezze
  $n[i] \leftarrow i$ ▷ $i = 1, \ldots, 2m+3$ posizioni effettive
  $f[1] \leftarrow 0$ ▷ incrementi delle posizioni desiderate
  $f[2m+3] \leftarrow 1$
  $f[2i+1] \leftarrow p[i]$ ▷ $i = 1, \ldots, m$
  $f[2i] \leftarrow (f[2i-1] + f[2i+1])/2$ ▷ $i = 1, \ldots, m+1$
  $d[i] \leftarrow 1 + 2(m+1)f[i]$ ▷ $i = 1, \ldots, 2m+3$ posizioni desiderate
end procedure

procedure ESTIMATE($x, q[2m+3], n[2m+3], f[2m+3], d[2m+3]$)
  if $x < q[1]$ **then**
    $k \leftarrow 1$
    $q[1] \leftarrow x$
  else if $x \geq q[2m+3]$ **then**
    $k \leftarrow 2m+2$
    $q[2m+3] \leftarrow x$
  else
    si trovi $k : q[k] \leq x < q[k+1]$
  end if
  $n[i] \leftarrow n[i] + 1$ ▷ $i = k+1, 2m+3$
  $d[i] \leftarrow d[i] + f[i]$ ▷ $i = 1, \ldots, 2m+3$
  for $i = 2, \ldots, 2m+2$ **do**
    $d \leftarrow d[i] - n[i]$ ▷ scostamento dalla posizione desiderata
    $dp \leftarrow n[i+1] - n[i]$ ▷ scostamento dalla posizione successiva
    $dm \leftarrow n[i-1] - n[i]$ ▷ scostamento dalla posizione precedente
    $qp \leftarrow (q[i+1] - q[i])/dp$
    $qm \leftarrow (q[i-1] - q[i])/dm$
    if $(d \geq 1 \wedge dp > 1)$ **then**
      $qt \leftarrow q[i] + ((1-dm)qp + (dp-1)qm)/(dp-dm)$
      if $q[i-1] < qt < q[i+1]$ **then**
        $q[i] \leftarrow qt$
      else
        $q[i] \leftarrow q[i] + qp$
        $n[i] \leftarrow n[i] + 1$
      end if
    else if $(d \leq -1 \wedge dm < -1)$ **then**
      $qt \leftarrow q[i] - ((1+dp)qm + (dm+1)qp)/(dp-dm)$
      if $q[i-1] < qt < q[i+1]$ **then**
        $q[i] \leftarrow qt$
      else
        $q[i] \leftarrow q[i] - qm$
        $n[i] \leftarrow n[i] - 1$
      end if
    end if
  end for
end procedure

48: **function** PSQUARE
49: si inizializzi $X[2m+3]$
50: si conservino in X i primi $2m+3$ indici calcolati
51: si ordinino le $2m+3$ osservazioni in X
52: si inizializzino $q[2m+3], n[2m+3], f[2m+3], d[2m+3]$
53: $INITIALIZATION(X, q, n, f, d)$
54: **for** ogni successivo indice calcolato x **do**
55: $ESTIMATE(x, q, n, f, d)$
56: **end for**
57: si inizializzi $\hat{q}[m]$
58: $\hat{q}[i] \leftarrow q[1+2i]$ ▷ $i = 1, \ldots, m$
59: **return** $\hat{q}$
60: **end function**

Bibliografia

[1] Abul Naga, Ramses H. e Yalcin, Tarik. «Inequality measurement for ordered response health data.» In: *Journal of health economics* 27 6 (2008), pp. 1614–25.

[2] Apouey, Bénédicte H. «Measuring health polarization with self-assessed health data.» In: *Health economics* 16 9 (2007), pp. 875–94.

[3] Arcagni, Alberto e Fattore, Marco. *parsec: Partial Orders in Socio-Economics*. R package version 1.2.0. 2018. URL: https://CRAN.R-project.org/package=parsec.

[4] Berry, Kenneth J. e Mielke Jr., Paul W. «Indices of Ordinal Variation». In: *Perceptual and Motor Skills* 74.2 (1992), pp. 576–578. DOI: 10.2466/pms.1992.74.2.576.

[5] Blair, Julian e Lacy, Michael G. «Statistics of Ordinal Variation». In: *Sociological Methods & Research* 28.3 (2000), pp. 251–280. DOI: 10.1177/0049124100028003001.

[6] Brightwell, Graham e Winkler, Peter. «Counting Linear Extensions is #P-complete». In: *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*. STOC '91. New Orleans, Louisiana, USA: ACM, 1991, pp. 175–181. ISBN: 0-89791-397-3. DOI: 10.1145/103418.103441.

[7] Bubley, Russ e Dyer, Martin. «Faster Random Generation of Linear Extensions». In: *Discrete Math.* 201.1-3 (1999), pp. 81–88. ISSN: 0012-365X. DOI: 10.1016/S0012-365X(98)00333-1.

[8] Bubley, Russ e Dyer, Martin. «Path coupling: A technique for proving rapid mixing in Markov chains». In: *Proceedings 38th Annual Symposium on Foundations of Computer Science*. 1997, pp. 223–231. DOI: 10.1109/SFCS.1997.646111.

[9] Chan, Tony F., Golub, Gene H. e LeVeque, Randy J. «Updating Formulae and a Pairwise Algorithm for Computing Sample Variances». In: *COMPSTAT 1982 5th Symposium held at Toulouse 1982*. A cura di Caussinus, H., Ettinger, P. e Tomassone, R. Heidelberg: Physica-Verlag HD, 1982, pp. 30–41. ISBN: 978-3-642-51461-6.

[10] Cormen, Thomas H. et al. *Introduzione agli algoritmi e strutture dati*. McGraw-Hill, 2001.

[11] De Loof, Karel. «Efficient computation of rank probabilities in posets». Tesi di dott. Universiteit Gent, 2009.

[12] Fattore, Marco. «Functionals and Synthetic Indicators Over Finite Posets». In: *Partial Order Concepts in Applied Sciences*. A cura di Fattore, Marco e Bruggemann, Rainer. Cham: Springer International Publishing, 2017, pp. 71–86. ISBN: 978-3-319-45421-4. DOI: 10.1007/978-3-319-45421-4_5.

[13] Fattore, Marco. «Partially Ordered Sets and the Measurement of Multidimensional Ordinal Deprivation». In: *Social Indicators Research* 128.2 (2016), pp. 835–858. ISSN: 1573-0921. DOI: 10.1007/s11205-015-1059-6.

[14] Fattore, Marco e Arcagni, Alberto. «A Reduced Posetic Approach to the Measurement of Multidimensional Ordinal Deprivation». In: *Social Indicators Research* 136.3 (2018), pp. 1053–1070. ISSN: 1573-0921. DOI: 10.1007/s11205-016-1501-4.

[15] Fattore, Marco e Arcagni, Alberto. «F-FOD: Fuzzy First Order Dominance analysis and populations ranking over ordinal multi-indicator systems». In revisione.

[16] Fattore, Marco e Arcagni, Alberto. «Measuring Multidimensional Polarization with Ordinal Data». In: *Advances in Latent Variables* (Milan, Italy). A cura di Brentari, Eugenio e Carpita, Maurizio. Vita e Pensiero, 2013.

[17] Fleurbaey, Marc e Blanchet, Didier. *Beyond GDP: Measuring Welfare and Assessing Sustainability*. Oxford University Press, 2013. URL: https://EconPapers.repec.org/RePEc:oxp:obooks:9780199767199.

[18] Gigliarano, Chiara e Mosler, Karl. «Constructing indices of multivariate polarization». In: *The Journal of Economic Inequality* 7.4 (2008), p. 435. ISSN: 1573-8701. DOI: 10.1007/s10888-008-9096-x.

[19] Grilli, Leonardo e Rampichini, Carla. «Scomposizione della dispersione per variabili statistiche ordinali». In: *Statistica* 62.1 (2002), pp. 111–116.

[20] Huber, Mark. «Fast perfect sampling from linear extensions». In: *Discrete Mathematics* 306.4 (2006), pp. 420–428. ISSN: 0012-365X. DOI: https://doi.org/10.1016/j.disc.2006.01.003.

[21] Jain, Raj e Chlamtac, Imrich. «The P-Square algorithm for dynamic calculation of percentiles and histograms without storing observations». In: *Communications of the ACM* (1985).

[22] Kobus, Martyna e Miłoś, Piotr. «Inequality decomposition by population subgroups for ordinal data». In: *Journal of Health Economics* 31.1 (2012), pp. 15–21. ISSN: 0167-6296. DOI: https://doi.org/10.1016/j.jhealeco.2011.11.005.

[23] Kvålseth, Tarald O. «Coefficients of Variation for Nominal and Ordinal Categorical Data». In: *Perceptual and Motor Skills* 80.3 (1995), pp. 843–847. DOI: 10.2466/pms.1995.80.3.843.

[24] Leik, Robert K. «A Measure of Ordinal Consensus». In: *The Pacific Sociological Review* 9.2 (1966), pp. 85–90. ISSN: 00308919. URL: http://www.jstor.org/stable/1388242.

[25] Leti, Giuseppe. *Statistica descrittiva*. Strumenti (il Mulino).: Scienze sociali. Il Mulino, 1983. ISBN: 9788815002785. URL: https://books.google.it/books?id=1Z5UAAAACAAJ.

[26] Nanivazo, Malokele. «First Order Dominance Analysis: Child Wellbeing in the Democratic Republic of Congo». In: *Social Indicators Research* 122.1 (2015), pp. 235–255. ISSN: 1573-0921. DOI: 10.1007/s11205-014-0673-z.

[27] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2018. URL: https://www.R-project.org/.

[28] Raatikainen, Kimmo E. E. «Simultaneous Estimation of Several Percentiles». In: *Simulation* 49 (1987), pp. 159–164. ISSN: 0037-5497. DOI: 10.1177/003754978704900405.

[29] Reardon, Sean F. «Measures of ordinal segregation». In: *Occupational and Residential Segregation*, pp. 129–155. DOI: 10.1108/S1049-2585(2009)0000017011.

[30] Schröder, Bernd. *Ordered Sets: An Introduction with Connections from Combinatorics to Topology*. Springer International Publishing, 2016. ISBN: 9783319297880.

[31] Schwarz, Keith. *Darts, Dice, and Coins*. Stanford University. 2011. URL: http://www.keithschwarz.com/darts-dice-coins/ (visitato il 27/09/2018).

[32] Segal, Michael e Kedem, Klara. «Geometric Applications of Posets». In: *Algorithms and Data Structures*. A cura di Dehne, Frank et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 402–415. ISBN: 978-3-540-69422-9.

[33] Sen, Amartya. «Capability and Well-Being». In: *The Quality of Life*. A cura di Nussbaum e Sen. Oxford: Clarendon Press, 1993.

[34] Sen, Amartya. *La diseguaglianza*. Bologna: Il Mulino, 1994.

[35] Vose, Michael D. «A linear algorithm for generating random numbers with a given distribution». In: *IEEE Transactions on Software Engineering* 17.9 (set. 1991), pp. 972–975. ISSN: 0098-5589. DOI: 10.1109/32.92917.

[36] Wuttke, Joachim. Forschungszentrum Juelich GmbH. 2013. URL: http://apps.jcns.fz-juelich.de/doku/sc/ransampl (visitato il 27/09/2018).