Pianificazione Automatica e sistemi di Supporto delle Decisioni

Matteo Aprile
Professore: Giampaolo Ghiani, Enamuele Manni

INDICE

I. DEFINIZIONI - 22/23.09.22

Ci occuperemo di 2 tipi di scenari:

- usare algoritmi a supporto delle decisioni
- usare algoritmi che sostituiscono completamente l'uomo

A. Business Analytics

Disciplina che utilizza dati, statistiche, modelli matematici per aiutare a prendere delle decisioni in base a dei dati.

Possiamo racchiudere i suoi passaggi in:

1) descriptive analytics: capire **cosa sia successo nel passato** tramite i dati disponibili
2) predictive analytics: cercare di **fare delle previsioni** in base ai dati già disponibili
3) prescriptive analytics: **creare un piano di azione** per poter massimizzare il KPI (Key Performance Indicator)

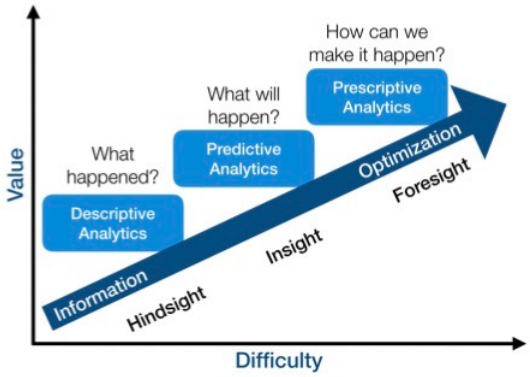

Figura 1. Fasi della business analytics

B. Decisioni

Rappresenta la scelta di un elemento tra piu' soluzioni dopo aver ponderato le opzioni.

Possiamo avere più casi d'uso:

- simplest case: abbiamo **poche alternative** quindi una semplice scelta
- multple criteria: abbiamo **più metri di paragone** delle performance, quindi si dovranno tenere in conto:
  - **soluzioni migliori** di altre (dette di Pareto)
  - **vincoli** dovuti dai clienti o da casi logistici da gestire (es: spedizioni)
  - ottimizzazioni matematiche
  - **conflitti tra i vincoli**
- incertezze e rischi:
  - **decisioni operative**: di **breve periodo reversibili** e **limitate** a "n" persone del team
  - **decisioni tattiche**: **coinvolge una parte dell'organizzazione** per un medio periodo

- **decisioni strategiche**: di **lungo periodo non reversibili** e **coinvolgono denaro**
- **decisioni strutturate**: hanno una **procedura di risoluzione specifica**
- **decisioni non strutturate**: richiedono **creatività** ed **esperienza** in un dato settore

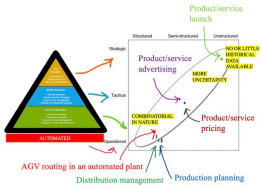

Figura 2. Diagonale decisionale

C. Business Intelligence (BI)

Usato per indicare un sistema dedicato alla raccolta di dati e alla loro elaborazione al fine di un reporting, infatti per "Inteligence" si intende investigazione. Venivano usati su dati atomici per avere delle conoscenze approfondite in un determinato business.

D. Data visualization

Consiste nel prendere dati e plottare un grafico, ma in realtà ora si ha una trattazione più metodologica, cioè se visualizzare in modo statico o meno i dati.

E. Decision Support Systems (DSS)

Si indicava un sistema computerizzato dotato di un sistema di "data managment" per creare un modello di ottimizzazione, fornendo un feedback tramite un'interfaccia. Ora indica una varietà di sistemi per visualizzare i dati in larga misura o meno.

F. Operations Research (OR)

Attivita organizzative per portare avanti un sistema logistico. Per "research" si indica la ricerca delle operation per conseguire dei risultati, avremo come sottocategorie:

- **ottimizzazione matematica**
- **queueing theory**: **studio matematico delle linee in attesa** il limite è che funzionano solo con sistemi semplici e con richieste di servizio in ordine stocastico
- **simulazione**: per usarle è **necessario generare dei numeri randomici quindi inconveniente** (bisogna fare un analisi statistica dei risultati dalle quali si farà una **stima**
- **game theory**: decisioni con **più players**

G. Agents

È un sistema che si muove in un environment (ambiente), ha dei sensori tramite i quali percepisce alcuni aspetti del mondo che lo circonda quindi si crea una rappresentazione del mondo circostante che può vedere. È capace di influenzare l'ambiente tramite degli attuatori come ruote o braccia (intendiamo anche agenti software).

Possiamo classificarli come:

- agenti autonomi: se è concepito in modo tale che **tramite un'istruzione sintetica raggiunge un goal sviluppando le azioni per raggiungerlo** In realtà può anche non essere una sequenza di azioni dato che **potrebbero esserci degli imprevisti**
- agenti intelligenti: se
- **impara dall'esperienza**
- crea una **rappresentazione dell'ambiente** che lo circonda e **ci ragiona sopra** per un possibile risultato delle proprie azioni
- **si adatta ad un ambiente mutevole**

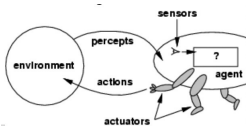

Figura 3. Schematizzazione di un agente e sue caratteristiche

H. Artificial Intelligence (AI)

Comprende tante sottodiscipline:

- automated reasoning: legato alla **rappresentazione del mondo** e **come raggionare su di essa** ma anche calcolandone le probabilità
- automated planning: usato in ambienti industriali
- automated learning
- natural language processing: **sviluppare agenti software** per fare sintesi di testi, scrivere automaticamente articoli, chat bot, ecc
- perception: visione artificiale
- manipuliation: avere un **agente che può modificare l'agente** circostante

I. Machine Learning (ML)

Consiste nell'apprendimento automatico e quindi lo sviluppo degli agenti che apprendo tramite la loro esperienza pregressa. Ci sarà allora una fase di training. Una delle possibili architetture che permette di farlo sono le Neural Networks prima avevano solo 2/3 neuroni, ora ne hanno vari strati il che fornisce delle prestazioni impressionanti

J. Deep Leaning

Si basa sull'apprendimento automatico con reti neurale tramite un gran numero di strati di neuroni.

K. Data Mining (DM)

Usare metodi di Machine Learning per estrarre manualmente dei pattern dai dati, cioè una regolarita' o un trend. È quindi la parte nobile del knowledge discovery in db, dato che i dati sono in genere disponibili su db o da altre piattaforme.

La sequenza nella quale interviene è:

1) **prendere** i dati
2) trovare i vari **target**
3) **preprocessare** i dati
4) trasformare i dati tramite il **data mining**
5) trovare dei **patterns** (dopo il data mining)

II. Software solutions and languages for AP and DSS - 29.09.22

A. Decisioni operative/strutturate

Sono una classe importante, si possono prendere tramite una procedura standard che può seguire un manuale o delle normative, automatizzata o no. Queste decisioni di breve periodo si collocano in basso a destra in figura 2.

Non essendo decisioni dove possiamo solo supportare, allora si possono andare a codificare in un linguaggio di programmazione procedurale come C++, Java, ecc...

Potremo avere un approccio:

- procedurale: dove devo far **generare delle azioni** in seguito di un obiettivo
- dichiarativo: si divide in:
  1) **modellazione** del problema
  2) **descrivo tramite un linguaggio di modellazione** (modelling language) che è un linguaggio di programmazione matematico come AMPL, oppure in linguaggi come python con Amply e Pulp
  3) **solver of the shelf**, che ci darà delle istruzioni per il nostro contesto

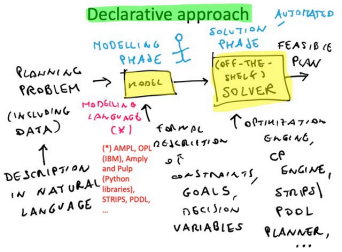

Figura 4. procedura implementata

Il che è utile dato che per agire su un problema bastera' cambiare il modello senza cambiare solver, dovrò solo cambiare il modello. È la soluzione più economico e flessibile ma è meno performante se in ambienti realtime devo prendere soluzioni in tempi molto stretti. Quindi in questi casi servono approcci procedurali.

Per sistemi che devono prendere soluzioni nel breve, si usa **C, C++, C#**.

B. Decisioni non strutturate o destrutturate

In questo caso non possiamo automatizzare, quindi:

1) tiro fuori i **dati aggregati**
2) si creano statistiche con **modelli di ottimizzazione**

Si usano degli **spreadsheet** che però non riescono a gestire big data e tendono a generare errori.

Il linguaggio più usato è **Python** ma non è la soluzione più efficiente per tutte quelle applicazioni dove il tempo di calcolo è importante.

C. Decisioni semistrutturate

Vogliamo solo valutare le prestazione di un sistema. Un esempio sono i sistemi che presentano un comportamento random per motivi:

1) i **server hanno un tempo di risposta** che possiamo modellare
2) le richieste del sistema **arrivano in maniera stocastica**

Si usano, in questo caso, metodi simulativi tramite dei Visual Interactive Modellling System, per simulare la rete per la quale passano le informazioni e i server ognuno con diverse proprietà di ciascun linker.

III. Introduzione all'ottimizzazione matematica - 30.09.22

A. Introduzione

Partiamo da un insieme di formule ed equazioni che modelleranno il problema. Con questo modello proviamo a trovare una soluzione al nostro problema attraverso algoritmi o risolutori. L'output è una soluzione per il nostro modello da implementare nel mondo reale.

B. Ingredienti principali

Gli ingredienti principali sanno:

- **dati** del problema
- variabili: dette anche var decisionali: scelte da fare in merito al problema. rappresentano quindi le scelte, quello su cui il decisore può intervenire
- vincoli: equazioni che definiscono i valori che le variabili possono assumere
- funzione obietivo: sarà una formula che rappresenta una misura di tipo quantitativo per capire quando è buona la soluzione che abbiamo ottenuto. quindi dovremo ottimizzare questo valore in base al contesto

Parleremo di programmazione lineare con modelli matematici o relazioni lineari, dato che molti problemi reali si rifanno a modelli lineari, per quanto essi possano essere complessi.

C. Descrizione del problema

Proviamo a risolvere un problema di mix di produzione, cioè un sistema con un impianto con 2 stabilimenti in cui:

1) nel primo: diamo le materie prime e vengono realizzati i componenti in uscita
2) nel secondo: diamo i componenti realizzati che vengono assemblati per creare il prodotto finito

Figura 5. Catena tra i due stabilimenti

Supponendo di voler realizzare 2 prodotti A, B con un differente profitto. Determinare il mix di produzione, cioè quante unità di A e B produrre la prossima settimana. Saranno presenti dei vincoli creati dalle risorse come i macchinari o gli addetti che potranno lavorare un numero di ore finito.

Stab	A	B	Addetti
1	4 ore	2 ore	10
2	2 ore	4 ore	10

Tabella I
TABELLA DELLE ORE DI LAVORO

A	B
15k	10k

Tabella II
TABELLA DEL PROFITTO /PALLET

A	B
40	120

Tabella III
TABELLA DEL PROFITTO EURO/PALLET

D. Dati del problema

- ore di lavoro:
- ogni addetto lavora 40 ore/settimana
- profitto /pallet:
- richiesta del prodotto nella prossima settimana:

E. Descrizione del problema con un modello matematico

Per effettuare una modellazione faremo:

1) identificare le variabili decisionali:
   - x_A: # di pallet di prodotto A da realizzare
   - x_B: # di pallet di prodotto B da realizzare
2) definire la funzione obbiettivo (FO), per massimizzare il profitto
3) definire i vincoli espressi come uguaglianza o disuguaglianza
   - vincolo 1: capacità produttiva dello stab 1 $4x_A+2x_B$ che non può superare $40*10$ cioè ore disponibili ogni settimana per un addetto * numero di addetti:
   $$4x_A + 2x_B <= 400$$
   - vincolo 2: capacità produttiva dello stabilimento 2 $2x_A + 4x_B$ che non può superare $40 * 10$ cioè ore disponibili ogni settimana per un addetto * numero di addetti:
   $$4x_A + 2x_B <= 400$$
   - vincolo 3: vincolo sulla richiesta di A:
   $$x_A <= 40$$
   - vincolo 4: vincolo sulla richiesta di B:
   $$x_B <= 120$$
   - vincolo 5: vincolo di non-negatività:
   $$x_A, x_B >= 0$$

Nella forma completa il modello complessivo è:

$$MAX = z = 15x_A + 10x_B$$

sottoposto ai vincoli (sv):

- $4x_A + 2x_B <= 400$
- $2x_A + 4x_B <= 400$
- $x_A <= 40$
- $x_B <= 120$
- $x_A, x_B >= 0$

F. Risolvere il modello matematico

Rappresentiamo sul piano cartesiano tutte le soluzioni ammissibili cercando quella che massimizza il nostro risultato

Impostiamo delle rette per ogni vincolo:

- presa $4x_A + 2x_B <= 400$ poniamo = 0, a turno, x_A e x_B: (200, 100)
- presa $2x_A + 4x_B <= 400$ poniamo = 0, a turno, x_A e x_B: (100, 200)
- presa $x_A <= 40$: (40, 0)
- presa $x_B <= 120$: (0, 120)

Avremo allora una regione ammissibile dove valgono tutti i vincoli e nella quale dovrebbe essere presente la nostra soluzione ammissibile. Per trovare il punto che rende massima la funzione z usiamo il metodo del gradiente:

$$\nabla z = \begin{bmatrix} \frac{dz}{dx_A} \\ \frac{dz}{dx_B} \end{bmatrix} = \begin{bmatrix} 15 \\ 10 \end{bmatrix}$$

dove ∇z sarà la massima crescita che viene rappresentata tramite (15, 10).

Tracciando una retta perpendicolare (curve di livello) alla retta del gradiente avremo valori sempre buoni ma più bassi di quelli sul gradiente, a patto che siano validi. Troveremo in fine il punto massimo che consente di massimizzare, cioè il più estremo alla regione ammissibile sarà il nostro punto.

Seguendo la retta del gradiente troviamo che la soluzione ottimale si trova nell'intersezione tra le rette del vincolo 2 con il 3: $x_A = 40$ $2 * 40 + 4X_B = 400$ quindi $x_B = 80$.

La soluzione ottimale sarà:

$$\begin{cases} x_A = 40 \\ 2x_A + 4x_B = 400 \end{cases} \quad \begin{cases} x_A = 40 \\ x_B = 80 \end{cases}$$

Si nota che lo stabilimento 2 viene saturato e quello 1 no, dal fatto che la soluzione giace sulla retta del vincolo per il quale si satura.

G. Terminologia

Possiamo avere altre forme di modelli di PL:

- fo da minimizzare
- vincoli di ugualianza
- vincoli >=
- variabili negative
- variabili non vincolate

Terminologie da sapere:

- soluzione: quella di output
- soluzione ammissibile: soluzione, se esiste, che **soddisfa tutti i vincoli**
- soluzione inammissibile: se **viola almeno un vincolo**

- regione ammissibile: tutti i punti che rispettano i vincoli
- prob inammissibile: **regione ammissibile vuota**
- prob ammissibile:
  - soluzione ottima singola
  - soluzioni multiple
  - fo illimitata

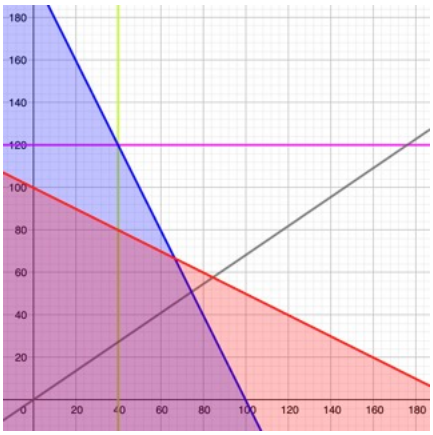

Figura 6. Rappresentazione grafica esempio

H. Implementazione in Python

```
import pulp as p

# 1. creazione del modello
model = p.LpProblem("ProductMix", p.LpMaximize)

# 2. definisco le variabili decisionall
x_A = p.LpVariable("x_A", cat="Continuous", lowBound
    =0)
x_B = p.LpVariable("x_B", cat="LpContinuous",
    lowBound=0)

# 3. definisco la funzione obiettivo in funzione
    delle variabili decisionali
model += 15 * x_A + 10 * x_B

# 4. definire i vincoli
model += 4 * x_A + 2 * x_B <= 400
model += 2 * x_A + 4 * x_B <= 400
model += x_A <= 40
model += x_B <= 120

# 5. ricolvere il problema
model.solve()

# print della soluzione
print("next week produce {} pallets of A".format(x_A
    .varValue))
print("next week produce {} pallets of B".format(x_B
    .varValue))
```

I. Esercitazione

1) Massimizzare la f.o. $z = 8x_1 + 6x_2$, con i vincoli:
   - $x_1 <= 5$
   - $x_2 <= 7$
   - $4x_1 + 3x_2 <= 29$
   - $x_1, x_2 >= 0$

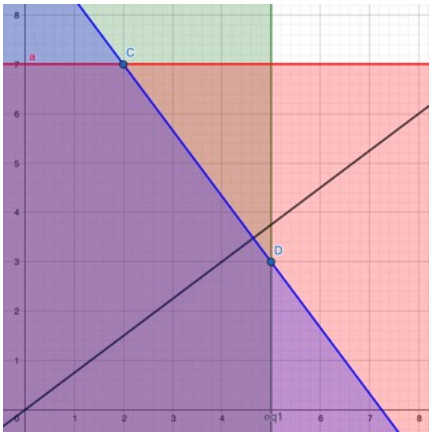

Figura 7. Rappresentazione grafica esempio 1

$$\nabla z = \begin{bmatrix} \frac{dz}{dx_A} \\ \frac{dz}{dx_B} \end{bmatrix} = \begin{bmatrix} 8 \\ 6 \end{bmatrix}$$

Abbiamo che esiste una curva di livello coincidente con lo spigolo CD, quindi abbiamo delle soluzioni ottime multiple:

- vertice C, prendiamo allora vincolo 2 e 3:

$$x_2 = 7 \rightarrow x_1 = 2$$

- vertice D, prendiamo allora vincolo 1 e 3:

$$x_1 = 5 \rightarrow x_2 = 3$$

- punti del segmento CD

Quindi $z = 58$

2) Minimizziamo la f.o. $z = 25x_1 + 22x_2$, con i vincoli:
   - $x_1 + x_2 >= 5$
   - $3x_1 + 2x_2 >= 12$
   - $3x_1 + 6x_2 >= 18$
   - $x_1, x_2 >= 0$

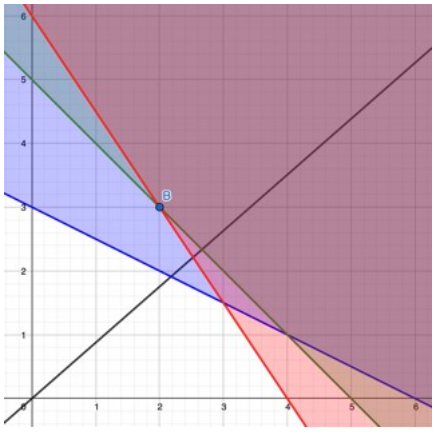

Figura 8. Rappresentazione grafica esempio 2

$$\nabla z = \begin{bmatrix} \frac{dz}{dx_A} \\ \frac{dz}{dx_B} \end{bmatrix} = \begin{bmatrix} 25 \\ 22 \end{bmatrix}$$

Per poter minimizzare, tracciando la curva di livello, trovando che la soluzione ottima si troverà dal punto B dato dall'intersezione dei vincoli 1 e 2:

$$x_1 = 2, x_2 = 3$$

Quindi $z = 116$

3) Massimizziamo la f.o. $z = 2x_1 + x_2$, con i vincoli:
   - $x_1 - x_2 <= 1$
   - $2x_1 + x_2 >= 6$
   - $x_2 >= 6$
   - $x_1, x_2 >= 0$

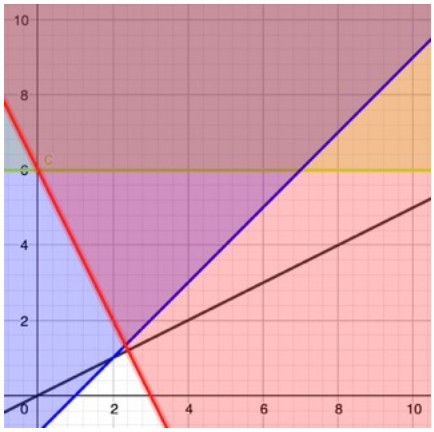

Figura 9. Rappresentazione grafica esempio 3

$$\nabla z = \begin{bmatrix} \frac{dz}{dx_A} \\ \frac{dz}{dx_B} \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Non raggiungeremo la regione ammissibile, quindi il problema non ammette una soluzione ottima.

4) Minimizziamo la f.o. $z = -2x_1 + 3x_2$, con i vincoli:
   - $x_1 - 2x_2 >= -2$
   - $2x_1 - x_2 <= 3$
   - $x_2 >= 4$
   - $x_1, x_2 >= 0$

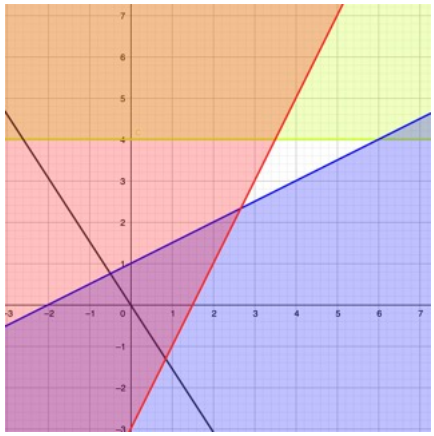

Figura 10. Rappresentazione grafica esempio 4

$$\nabla z = \begin{bmatrix} \frac{dz}{dx_A} \\ \frac{dz}{dx_B} \end{bmatrix} = \begin{bmatrix} -2 \\ 3 \end{bmatrix}$$

La regione ammissibile vuota e per tanto i problema inammissibile, quindi non esiste un punto che soddisfa contemporaneamente tutti i vincoli.

5) L'azienda vuole decide oltre al piano di produzione anche la giusta riallocazione degli addetti (10-10) tra i due reparti. Le variabili decisionali sono:
   - x_A, x_B: i prodotti
   - n_p: # addetti allocati al reparto produzione
   - n_a: #addetti allocati al reparto assemblaggio

   Quindi andremo ad aggiungere il vincolo per cui $n_p + n_a = 20$.
   Perciò avremo: $z = 15x_A + 10x_B$, con vincoli:
   - $4x_A + 2x_B <= 40n_p$
   - $2x_A + 4x_B <= 40n_a$
   - $x_A <= 40$
   - $x_B <= 120$
   - $n_a + n_p = 20$
   - $x_A, x_B, n_a, n_p >= 0$

RIFERIMENTI BIBLIOGRAFICI

[1] https://en.wikibooks.org/wiki/LaTeX/Hyperlinks