Data-Driven Decision-Making In Enterprise Applications

Linear Programming II

Rainer Schlosser

Hasso Plattner Institute (EPIC)

May the 4th (be with you), 2020

Decision-Making Using Linear Programming

(Non)linear Decision Problem	→	Formulated as LP Model	→	Standard LP Solver	→	Obtain Optimal Solution
Tough		**Magic**		Easy		Awesome!

Linear Programming II

- Questions regarding last week?

- Today: – Motivation AMPL

  – Example V – Equilibria in Mixed Strategies (Game Theory)

  – Penalty Approaches & Continuous Relaxations

  – Solution Tuning

  – Tricks to Circumvent Non-Linearities

Solving Motivation AMPL

Solving Knapsack Problems using LP via AMPL

- All you need: AMPL, a solver, 10 lines of code

- AMPL translates the problem to the solver, which solves the problem
- Simplex Alg. is fast in general - but can have exponential complexity

- Can we solve our knapsack problem with 1000, 10K, or 100K items?
- What do you think is the solution time?

LP meets Game Theory

Game Theory – “Gefangenendilemma” (Pure NE)

A \ B	Gestehen	Leugnen
Gestehen	-6 / -6	0 / -10
Leugnen	-10 / 0	-2 / -2

What’s the best strategy? Equilibrium in **pure** strategies: “Gestehen” (dominant)

Game Theory – "Papier Stein Schere" (Mixed NE)

		Spieler 2		
		Stein′	Schere′	Papier′
Spieler 1	Stein	0 0	−1 1	1 −1
	Schere	1 −1	0 0	−1 1
	Papier	−1 1	1 −1	0 0

No pure equilibrium. What is the best (mixed) strategy?

Game Theory – "Papier Stein Schere" (Mixed NE)

		1/3	1/3	1/3
		Spieler 2		
		Stein′	Schere′	Papier′
Spieler 1	1/3 Stein	0, 0	1, −1	−1, 1
	1/3 Schere	−1, 1	0, 0	1, −1
	1/3 Papier	1, −1	−1, 1	0, 0

No pure equilibrium. What is the best (mixed) strategy?

Symmetric Intuition: Equilibrium in mixed strategies, i.e., 1/3, 1/3, 1/3

Game Theory – “Papier Stein Schere 2.0”

Spieler 2

Spieler 1		Stein′	Schere′	Papier′
	Stein	0 0	−1 1	2.0 −1
	Schere	1 −1	0 0	−1 1
	Papier	−1 1	1 −1	0 0

←

Asymmetric rewards. Will player 2 play more often “Papier”?

Answer?

Game Theory – "Papier Stein Schere 2.0"

		1/3	1/3	1/3
		Spieler 2		
Spieler 1		Stein′	Schere′	Papier′
1/4	Stein	0 0	−1 1	2.0 −1
5/12	Schere	1 −1	0 0	−1 1
1/3	Papier	−1 1	1 −1	0 0

Asymmetric rewards. Will player 2 play more often "Papier"?

Answer: No. But player 1 plays more "Schere"!

Game Theory – “Papier Stein Schere 2.0”

		Spieler 2		
		1/3	1/3	1/3
Spieler 1		Stein'	Schere'	Papier'
1/4	Stein	0 / 0	1 / −1	−1 / 2.0
5/12	Schere	−1 / 1	0 / 0	1 / −1
1/3	Papier	1 / −1	−1 / 1	0 / 0

Solution Approach: Use Linear Programming to make the competitor *indifferent* in his/her strategies !

LP Model – "Papier Stein Schere 2.0"

Assume payoff $r^{(1)}(i,j)$ for player 1 when playing *i* while the other plays *j*

Assume payoff $r^{(2)}(i,j)$ for player 2 when playing *j* while the other plays *i*

Variables: $x^{(1)}(i)\,, x^{(2)}(j) \in [0,1]$ prob's of players playing options, *i,j=1,…,N*

Solution Approach: P1 makes P2 indifferent in all actions *j=1,…,N*, i.e.,

$$\sum_{i=1,..,N} x^{(1)}(i) \cdot r^{(2)}(i,1) = \sum_{i=1,..,N} x^{(1)}(i) \cdot r^{(2)}(i,2) = \sum_{i=1,..,N} x^{(1)}(i) \cdot r^{(2)}(i,3)$$

and vice versa (P2 makes P1 indifferent in all actions *i=1,…,N*):

$$\sum_{j=1,..,N} x^{(2)}(i) \cdot r^{(1)}(1,j) = \sum_{j=1,..,N} x^{(2)}(i) \cdot r^{(1)}(2,j) = \sum_{j=1,..,N} x^{(2)}(j) \cdot r^{(1)}(3,j)$$

LP Model – “Papier Stein Schere 2.0”

```
param N :=3;                                # number of options

param r1{i in 1..N, j in 1..N} := if i=j then 0 else if (1+i) mod 3
       = j mod 3 then Uniform(0,5) else Uniform(-5,0); # payoffs

param r2{i in 1..N, j in 1..N} := -r1[i,j];             # 2Pers-0sum-game

var x1 {i in 1..N} >= 0;                    # probability P1 playing option i
var x2 {j in 1..N} >= 0;                    # probability P2 playing option j

subject to NB1:             sum{i in 1..N} x1[i] = 1;    # norm player 1
subject to NB2:             sum{j in 1..N} x2[j] = 1;    # norm player 2

subject to NB3{j in 2..N}:  sum{i in 1..N} x1[i]*r2[i,j] # 1 makes 2
                          = sum{i in 1..N} x1[i]*r2[i,1];# indifferent

subject to NB4{i in 2..N}:  sum{j in 1..N} x2[j]*r1[i,j] # 2 makes 1
                          = sum{j in 1..N} x2[j]*r1[1,j];# indifferent

solve;  display x1,x2;                                   # solution
```

Penalty Approaches & Continuous Relaxations

Penalty Formulations for Contraints

Objective: $\max\limits_{x_1,\dots,x_N \in \{0,1\}} \sum\limits_{i=1,\dots,N} u_i \cdot x_i$ Knapsack example

Constraints: $\sum\limits_{i=1,\dots,N} s_i \cdot x_i \leq C$ (One) Hard Constraint

Penalty-Objective: $\max\limits_{x_1,\dots,x_N \in \{0,1\}} \sum\limits_{i=1,\dots,N} u_i \cdot x_i - \alpha \cdot \sum\limits_{i=1,\dots,N} s_i \cdot x_i$ (Soft Constraint)

Constraints: none

Results: Pareto-optimal combinations of "Utility" and "Space"

Continuous Relaxations of Integer Problems

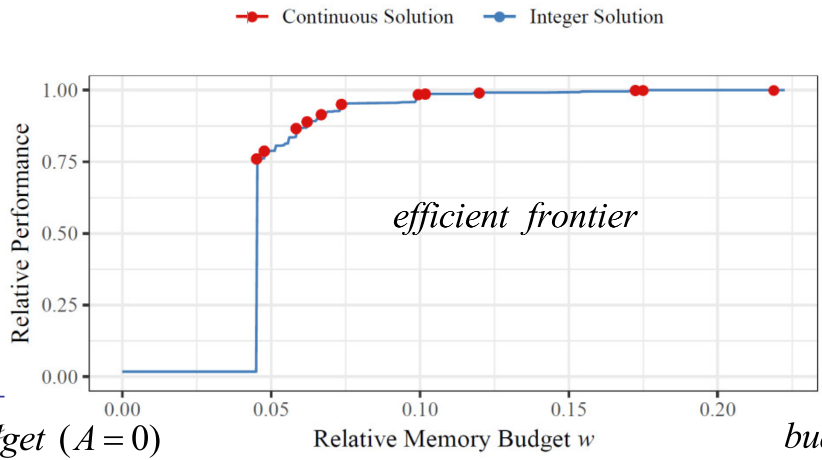
(i) Optimal integer solution (blue): $\min_{\vec{x}\in\{0,1\}^N} F(\vec{x}) \;\; s.t. \;\; M(\vec{x}) \le A \;\Rightarrow\; \vec{x}^*(A) \;\; optimal$

(ii) Continuous relaxation: $\rightarrow \min_{\vec{x}\in[0,1]^N} F(\vec{x}) \;\; s.t. \;\; M(\vec{x}) \le A \;\Rightarrow\; \vec{x}^*(A) \in \{0,1\}^N \; ?$

(iii) Penalty formulation (red): $\min_{\vec{x}\in[0,1]^N} F(\vec{x}) + \alpha \cdot M(\vec{x}) \;\Rightarrow\; \vec{x}^*(\alpha) \in \{0,1\}^N \;\; and$

$\uparrow \qquad Pareto-optimal!$

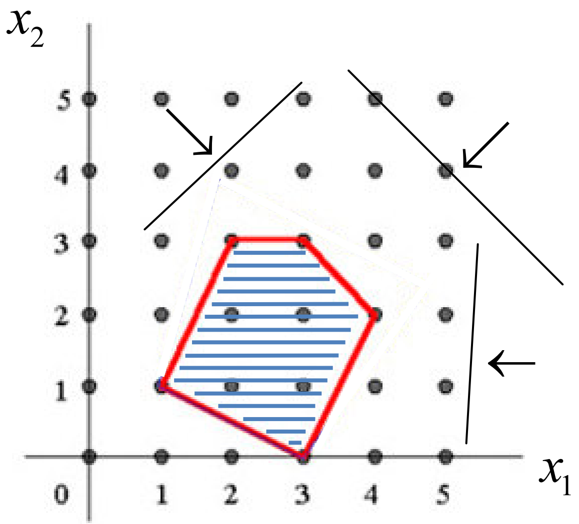
Performance: Runtime saved



no budget $(A = 0)$ $\qquad$ *budget* A

When do Integer & Continuous Solutions Coincide?

maximize $a \cdot x_1 + b \cdot x_2$ s.t. . . . with $x_1, x_2 \in \mathbb{R}$ vs. $x_1, x_2 \in \mathbb{N}$



- **Answer**: The corners of the polygon have to be “integers”!

Solution Tuning

Recall Example IV: Project Assignment Problem

$x_{i,j} \in \{0,1\}$ whether project i, i=1,...,N, is assigned to worker j, j=1,...,N

LP: $$\max_{x_{i,j} \in \{0,1\}^{N \times N}} \sum_{i=1,\dots,N,\ j=1,\dots,N} w_{i,j} \cdot x_{i,j}$$

s.t. $\sum_{i=1,\dots,N} x_{i,j} = 1$ for all j=1,...,N (each worker gets 1 project)

$\sum_{j=1,\dots,N} x_{i,j} = 1$ for all i=1,...,N (each project is assigned)

- Will the allocation always be fair?
- How "outliers" can be avoided?
- Approaches: (i) utility functions, (ii) max min, (iii) multi-objective

Approach (i): Fair Project Assignment (Non-linear)

$x_{i,j} \in \{0,1\}$ whether project i, i=1,...,N, is assigned to worker j, j=1,...,N

NLP:

$$\max_{x_{i,j}\in\{0,1\}^{N\times N}} \sum_{j=1,\ldots,N} u\left(\sum_{i=1,\ldots,N} w_{i,j}\cdot x_{i,j}\right)$$

using, e.g., $u(z):=\ln(z)$, $u(z):=z^{0.6}$, or $u(z):=-e^{-0.1\cdot z}$

s.t.

$$\sum_{i=1,\ldots,N} x_{i,j} = 1 \quad \text{for all } j=1,\ldots,N \quad \text{(each worker gets 1 project)}$$

$$\sum_{j=1,\ldots,N} x_{i,j} = 1 \quad \text{for all } i=1,\ldots,N \quad \text{(each project is assigned)}$$

- Idea: Avoiding low scores is better than including high scores
- Disadvantage (i): Non-linear solver is needed

Approach (ii): Fair Project Assignment (Linear!)

$x_{i,j} \in \{0,1\}$ whether project i, i=1,…,N, is assigned to worker j, j=1,…,N

NLP: $$\max_{x_{i,j} \in \{0,1\}^{N \times N}} \left\{ \min_{j=1,\ldots,N} \sum_{i=1,\ldots,N} w_{i,j} \cdot x_{i,j} \right\}$$, i.e., max poorest guy's reward!

LP: $$\cong \max_{x_{i,j} \in \{0,1\}^{N \times N}, z \in \mathbb{R}} z \quad \text{s.t.} \quad z \le \sum_{i=1,\ldots,N} w_{i,j} \cdot x_{i,j} \quad \text{for all } j=1,\ldots,N$$

$$\sum_{i=1,\ldots,N} x_{i,j} = 1 \quad \text{for all } j=1,\ldots,N \quad \text{(each worker gets 1 project)}$$

$$\sum_{j=1,\ldots,N} x_{i,j} = 1 \quad \text{for all } i=1,\ldots,N \quad \text{(each project is assigned)}$$

- Idea: Optimize the lowest willingness (cf. worst case criteria)
- Disadvantage (ii): Total willingness score can be low

Approach (iii): Fair Project Assignment (Linear!)

$x_{i,j} \in \{0,1\}$ whether project i, i=1,...,N, is assigned to worker j, j=1,...,N

LP: $$\max_{x_{i,j} \in \{0,1\}^{N \times N}, z \in \mathbb{R}} \sum_{i=1,...,N,\ j=1,...,N} w_{i,j} \cdot x_{i,j} + \alpha \cdot z,$$ with parameter $\alpha \geq 0$

s.t. $$z \leq \sum_{i=1,...,N} w_{i,j} \cdot x_{i,j} \quad \forall j$$

$\sum_{i=1,...,N} x_{i,j} = 1$ for all j=1,...,N (each worker gets 1 project)

$\sum_{j=1,...,N} x_{i,j} = 1$ for all i=1,...,N (each project is assigned)

- Idea: Combine both objectives as a weighted sum
- Disadvantage (iii): Suitable weighting factor α has to be determined

Nonlinear Programming Models

- Often *non-linear expressions* are needed within a model

- (–) Linear solvers cannot be used anymore
- (–) NL solvers often cannot guarantee optimality

- (+) So-called "mild" nonlinearities can be expressed linearly
- (+) This is very valuable as we can exploit LP solvers and their optimality

- The price of such transformations is acceptable:

  More variables and constraints

Linearization Tricks

I Linearization of “and” in the Constraints

Objective: $\min_{x_1, x_2 \in \{0,1\}} 2 \cdot x_1 + x_2$

Constraints NL: . . .

$$x_1 = 1 \; and \; x_2 = 1$$ (e.g. needed as joint condition)

Objective: $\min_{x_1, x_2 \in \{0,1\}} 2 \cdot x_1 + x_2$

Constraints LIN: . . .

$$x_1 + x_2 = 2$$

II Linearization of "or" in the Constraints

Objective: $\min_{x_1,x_2\in\{0,1\},x_3\in[0,M]} 2\cdot x_1 + x_2 + x_3$

Constraints NLa: $x_1 = 1 \; or \; x_2 = 1$ (e.g., needed as joint condition)

Constraints NLb: $x_1 = 1 \; or \; x_2 = 0$

Constraints NLc: $x_3 = 0 \; or \; x_3 \geq 3$

Objective: $\min_{x_1,x_2\in\{0,1\},x_3\in[0,M],z\in\{0,1\}} 2\cdot x_1 + x_2 + x_3$

Constraints LINa: $x_1 + x_2 \geq 1$

Constraints LINb: $x_1 + (1 - x_2) \geq 1$

Constraints LINc: $x_3 \leq M\cdot z \,,\quad x_3 \geq 3\cdot z$

III Linearization of "max" in the Objective

Objective NL: $$\min_{x_1,\ldots,x_N \in \mathbb{R}} \left\{ \max_{i=1,\ldots,N} x_i \right\}$$

Constraints: . . .

Objective LIN: $$\min_{x_1,\ldots,x_N \in \mathbb{R},\, z \in \mathbb{R}} z$$

Constraints: . . .

new $z \geq x_i$ for all $i=1,\ldots,N$

IV Linearization of “min” in the Objective

Objective NL: $$\max_{x_1,\dots,x_N \in \mathbb{R}} \left\{ \min_{i=1,\dots,N} x_i \right\}$$

Constraints: . . .

Objective LIN: $$\max_{x_1,\dots,x_N \in \mathbb{R}, z \in \mathbb{R}} z$$

Constraints: . . .

new $z \leq x_i$ for all $i=1,\dots,N$

V Linearization of "min" in the Constraints

Objective: $\min_{x_1, x_2 \in [0,M]} 2 \cdot x_1 + x_2$

Constraints NL: $4 \leq \min(x_1, x_2) \leq 7$

Objective: $\min_{x_1, x_2 \in [0,M], z_1, z_2 \in \{0,1\}} 2 \cdot x_1 + x_2$

Constraint LIN: $4 \leq x_i$ for all i=1,2

new $M \cdot z_i \geq x_i - 7$ for all i=1,2

new $z_1 + z_2 \leq 1$

VI Linearization of "abs" in the Objective

Objective NL: $\min_{x_1, x_2 \in \mathbb{R}} 2 \cdot x_1 + abs(3 - x_2)$

Constraints: . . .

Objective LIN: $\min_{x_1, x_2 \in \mathbb{R}, z \in \mathbb{R}} 2 \cdot x_1 + z$

Constraints: . . .

new $x_2 - 3 \le z$

new $3 - x_2 \le z$

VII Linearization of “abs” in the Constraints

Objective: $\min_{x_1, x_2 \in \mathbb{R}} 2 \cdot x_1 + x_2$

Constraints NL: $abs(3 - x_2) \leq x_1$

Objective LIN: $\min_{x_1, x_2 \in \mathbb{R}, z \in \mathbb{R}} 2 \cdot x_1 + x_2$

Constraints: $z \leq x_1$

new $x_2 - 3 \leq z$

new $3 - x_2 \leq z$

VIII Linearization of “if-then-else”

Objective NL: $$\min_{x_1, x_2 \in \{0,1,2,\ldots,M\}} 2 \cdot x_1 + \left(if\ x_2 \leq 5.5\ \ then\ a\ else\ b \right)$$

Constraints: . . .

Objective LIN: $$\min_{x_1, x_2 \in \{0,1,2,\ldots,M\}, z \in \{0,1\}} 2 \cdot x_1 + b \cdot z + a \cdot (1 - z)$$

Constraints: . . .

new $$x_2 - 5.5 \leq M \cdot z$$

new $$5.5 - x_2 \leq M \cdot (1 - z)$$

IX Linearization of a Product of Binary Variables

Objective: $\min\limits_{x_1, x_2 \in \{0,1\}} 2 \cdot x_1 + x_2$

Constraints NL: including the term: $x_1 \cdot x_2$

Objective: $\min\limits_{x_1, x_2 \in \{0,1\}, z \in \{0,1\}} 2 \cdot x_1 + x_2$

Constraints LIN: include the term z instead, where

$z \leq x_i$, for i=1,2

$z \geq x_1 + x_2 - 1$

X Linearization of a Binary x Continuous Variable

Objective: $$\min_{x_1 \in \{0,1\}, x_2 \in [0,M]} 2 \cdot x_1 + x_2$$

Constraints NL: including the term: $x_1 \cdot x_2$

Objective: $$\min_{x_1 \in \{0,1\}, x_2 \in [0,M], z \in [0,M]} 2 \cdot x_1 + x_2$$

Constraints LIN: include the term z instead, where

$z \leq M \cdot x_1$, for i=1,2

$z \leq x_2$

$z \geq x_2 - (1 - x_1) \cdot M$

Next Week

Homework: Get AMPL. Solve Examples I-V (see code online).

Review the Linearizations I-X!

Outlook:

- Introduction in AMPL
- Implementations of Example I-V
- Play with parameters, randseed, and problem complexity
- Nonlinear Programming and Suitable Solvers

Overview

Week	Dates	Topic	
1	April 27/30	Introduction + Linear Programming	
2	May 4/ (7)	Linear Programming II	
3	May **11**/14	**Exercise Implementations**	
4	May 18	Linear + Logistic Regression	(Thu May 21 “Himmelfahrt”)
5	May 25/28	Dynamic Programming	(Mon June 1 “Pfingstmontag”)
6	June 4	Dynamic Pricing Competition	
7	June 8/11	Project Assignments	
8	June 15/18	Robust + Nonlinear Optimization	
9	June 22/25	Work on Projects: Input/Support	
10	June 29/2	Work on Projects: Input/Support	
11	July 6/9	Work on Projects: Input/Support	
12	July 13/16	Work on Projects: Input/Support	
13	July/Aug	Finish Documentation (Deadline: Aug 31)	