More KKT and Duality Interpretations and Applications

Yinyu Ye
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

Chapters 3.5-3.6, 11.7-11.8

Two-Person Zero-Sum Matrix Game

$$\begin{pmatrix} 3 & -1 & -3 \\ -3 & 1 & 4 \end{pmatrix} = P$$

P is the payoff matrix of a two-person, ”Column” and ”Row”, zero-sum game.
Player Column Player chooses column(s) to maximize the payoff to Column
Player Row chooses row(s) to minimize the payoff to Column

Pure Strategy: Each player chooses a single column (row).

Mixed or Randomized Strategy: Each player randomly chooses columns (rows) strategies with a fixed probability distribution.

Nash Equilibrium: No player can alter its probability distribution to achieve better expected payoff.

Two-Person Zero-Sum Matrix Game II

$$\begin{pmatrix} 3 & -1 & -3 \\ -3 & 1 & 4 \end{pmatrix}$$

Player Column Player: probabilities x_1 to choose column 1, x_2 to choose column 2, and x_3 to choose column 3. Then the expected payoff is

$3x_1 - x_2 - 3x_3$ if Player Row chooses row 1

$-3x_1 + x_2 + 4x_3$ if Player Row chooses row 2

Thus, Player Column would

$$\begin{array}{ll} \text{maximize}_{(x1,x2,x3)} & \min\{3x_1 - x_2 - 3x_3,\ -3x_1 + x_2 + 4x_3\} \\ \text{s.t.} & X_1+x_2+x_3 = 1,\ (x_1,x_2,x_3) \geq 0 \end{array}$$

which can be cast as a linear program

$$\begin{array}{lll} \text{maximize}_{(x1,x2,x3,v)} & v & \\ \text{s.t.} & -3x_1 + x_2 + 3x_3 + v \leq 0 & y_1 \\ & 3x_1 - x_2 - 4x_3 + v \leq 0 & y_2 \\ & x_1 + x_2 + x_3 = 1, & u \\ & (x_1,x_2,x_3) \geq 0 & \end{array}$$

Two-Person Zero-Sum Matrix Game III

$$\begin{pmatrix} 3 & -1 & -3 \\ -3 & 1 & 4 \end{pmatrix}$$

Then, the dual of the linear program

$$\begin{aligned} \text{minimize}_{(y1,y2,u)} \quad & u \\ \text{s.t.} \quad & u - (3y_1 - 3y_2) \geq 0 \\ & u - (-y_1 + y_2) \geq 0 \\ & u - (-3y_1 + 4y_2) \geq 0 \\ & y_1 + y_2 = 1, \ (y_1, y_2) \geq 0 \end{aligned}$$

Interpretations:

Player Row: probabilities y_1 to choose row 1, y_2 to choose row 2. Then the expected payoff to Player Column is

$3y_1 - 3y_2$ if Player Column chooses column 1

$-y_1 + y_2$ if Player Column chooses column 2

$-3y_1+4y_2$ if Player Column chooses column 3;

and Player Row does

$$\begin{aligned} \text{minimize}_{(y1,y2)} \quad & \max\{3y_1 - 3y_2, -y_1 + y_2, -3y_1+4y_2\} \\ \text{s.t.} \quad & y_1+y_2 = 1, \ (y_1,y_2) \geq 0 \end{aligned}$$

Robust Portfolio Management I

Two stocks with return rate 0.5 each, and stock 2 has more variance, and the two are negatively correlated

$$\begin{aligned} \min \quad & (x_1)^2 + 2(x_2)^2 - 2x_1x_2 - 0.5x_1 - 0.5x_2 \\ \text{s.t.} \quad & x_1 + x_2 = 1 \end{aligned}$$

This is a convex optimization problem

FONC are sufficient : set the (partial) derivatives of LF

$$L(x_1, x_2, y) = (x_1)^2 + 2(x_2)^2 - 2x_1x_2 - 0.5x_1 - 0.5x_2 - y(x_1 + x_2 - 1)$$

to zeros

$$2x_1 - 2x_2 - 0.5 - y = 0,\ \ 4x_2 - 2x_1 - 0.5 - y = 0$$

$$\Rightarrow x_2 = 0.5 + y \Rightarrow x_1 = 0.75 + 1.5y \Rightarrow$$

$$\max \quad \phi(y) = -0.3125 - 0.25y - 1.25y^2$$

Application: Robust Portfolio Management II

$$\begin{array}{ll}\min & (x_1)^2 + 2(x_2)^2 - 2x_1x_2 - \mu_1 x_1 - \mu_2 x_2 \\ \text{s.t.} & x_1 + x_2 = 1,\end{array}$$

But the return two return rates are uncertain, and they are in the range $\mu_1+\mu_2=1,\ (\mu_1)^2+(\mu_2)^2 \le 1$

$$\begin{array}{ll}\min & (x_1)^2 + 2(x_2)^2 - 2x_1x_2 + \\ & \left\{\begin{array}{ll}\max_{\mu 1,\mu 2} & -x_1\mu_1 - x_2\mu_2, \\ \text{s.t.} & \mu_1 + \mu_2 = 1, (\mu_1)^2 + (\mu_2)^2 \le 1\end{array}\right\} \\ \text{s.t.} & x_1 + x_2 = 1,\end{array}$$

The inner problem is maximization, representing the decision makers' complete risk reverse attitude

$$\begin{array}{ll}\min & (x_1)^2 + 2(x_2)^2 - 2x_1x_2 + \\ & \left\{\begin{array}{ll}\min_{y1,y2} & \dfrac{(-x_1 - y_1)^2 + (-x_2 - y_1)^2}{4y_2} + y_1 + y_2, \\ \text{s.t.} & y_1 \text{ free}, y_2 \ge 0\end{array}\right\} \\ \text{s.t.} & x_1 + x_2 = 1,\end{array}$$

Replacing the inner problem by its dual (see next slide).

The Dual of the Inner Problem

$$\max_{x1,x2}\ c_1x_1 + c_2x_2$$
$$\text{s.t.}\quad x_1 + x_2 = 1, \qquad \wedge\ y_1\text{: free}$$
$$(x_1)^2 + (x_2)^2 \le 1, \quad \wedge\ y_2 \ge 0$$

Primal

$$L(x_1, x_2, y) = c_1x_1 + c_2x_2 - y_1(x_1 + x_2 - 1) - y_2((x_1)^2 + (x_2)^2 - 1),$$

$$\begin{pmatrix} c_1 - y_1 - 2y_2x_1 \\ c_2 - y_1 - 2y_2x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\phi(y) = \frac{(c_1 - y_1)^2 + (c_2 - y_1)^2}{4y_2} + y_1 + y_2,$$

$$\min\ \phi(y),\ \text{s.t.}\ \ y_1 \text{ free}, y_2 \ge 0$$

Dual

Application: Robust Portfolio Management III

$$\min \quad (x_1)^2 + 2(x_2)^2 - 2x_1x_2 + \left\{ \begin{array}{ll} \min_{y1,y2} & \frac{(-x_1 - y_1)^2 + (-x_2 - y_1)^2}{4y_2} + y_1 + y_2, \\ \text{s.t.} & y_1 \text{ free}, y_2 \geq 0 \end{array} \right\}$$
$$\text{s.t.} \quad x_1 + x_2 = 1,$$

The objectives of the outer and inner problems are now aligned, so that we can combine them into a joint single layer problem

$$\min \ (x_1)^2 + 2(x_2)^2 - 2x_1x_2 + \frac{(x_1 + y_1)^2 + (x_2 + y_1)^2}{4y_2} + y_1 + y_2$$
$$\text{s.t.} \ \ x_1 + x_2 = 1, y_1 \text{ free}, y_2 \geq 0$$

$$\min \ (x_1)^2 + 2(x_2)^2 - 2x_1x_2 + \sqrt{(x_1 + y_1)^2 + (x_2 + y_1)^2} + y_1$$
$$\text{s.t.} \ \ x_1 + x_2 = 1, y_1 \text{ free}$$

Now, given x_1 and x_2, how to find minimizer y_1?

$$\min \quad y_1 + r$$
$$\text{s.t.} \sqrt{(x_1 + y_1)^2 + (x_2 + y_1)^2} \leq r, y_1 \text{free},$$

$$= -\min\{x_1, x_2\}$$

Application: Robust Portfolio Management IV

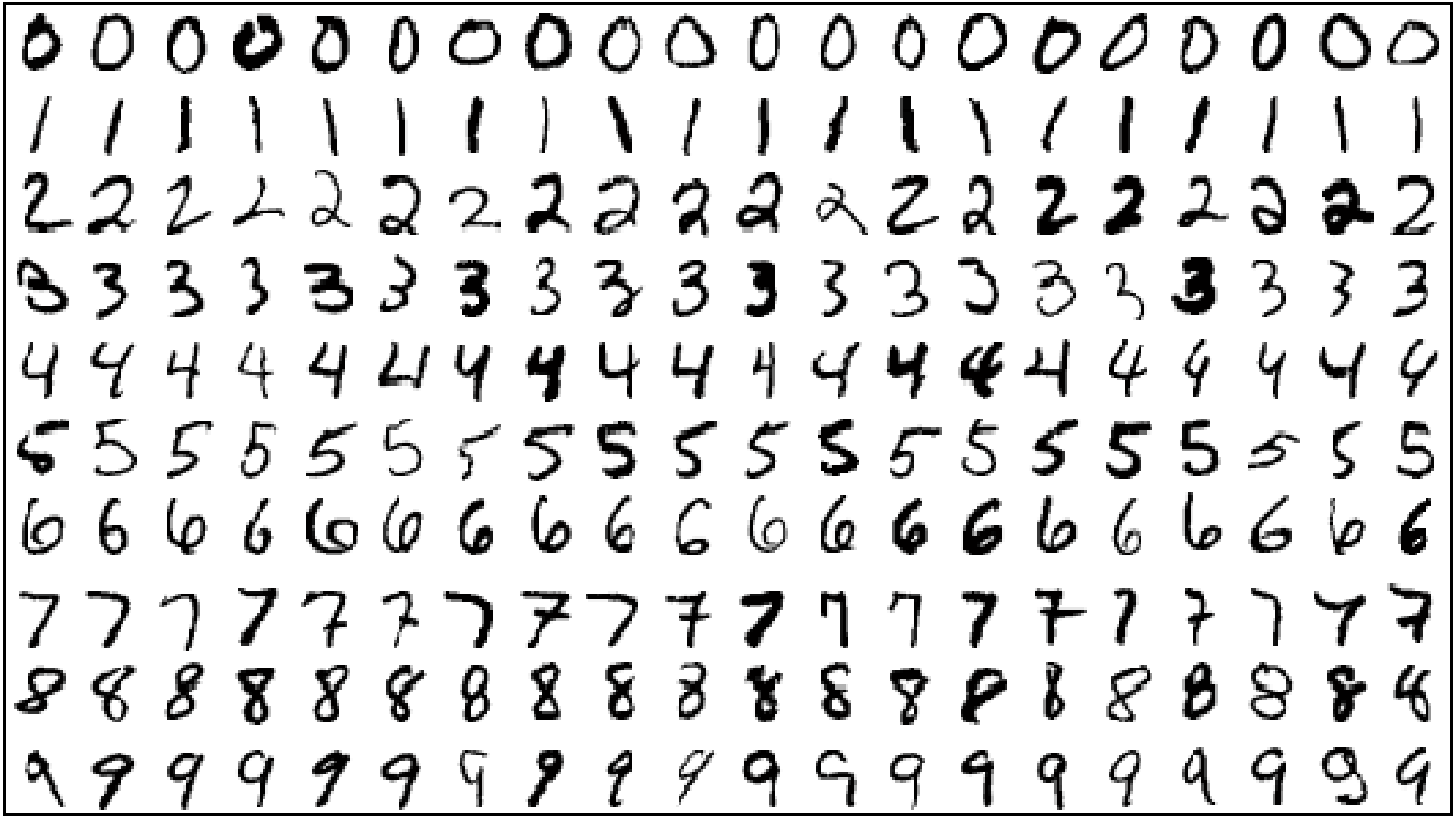
$$\min\ (x_1)^2 + 2(x_2)^2 - 2x_1x_2 - \min\{x_{1,}x_2\}$$
$$\text{s.t.}\ \ x_1 + x_2 = 1.$$

$$\min\ (x_1)^2 + 2(x_2)^2 - 2x_1x_2 - z$$
$$\text{s.t.}\ \ x_1 + x_2 = 1,\ \ x_1 \geq z,\ \ x_2 \geq z$$

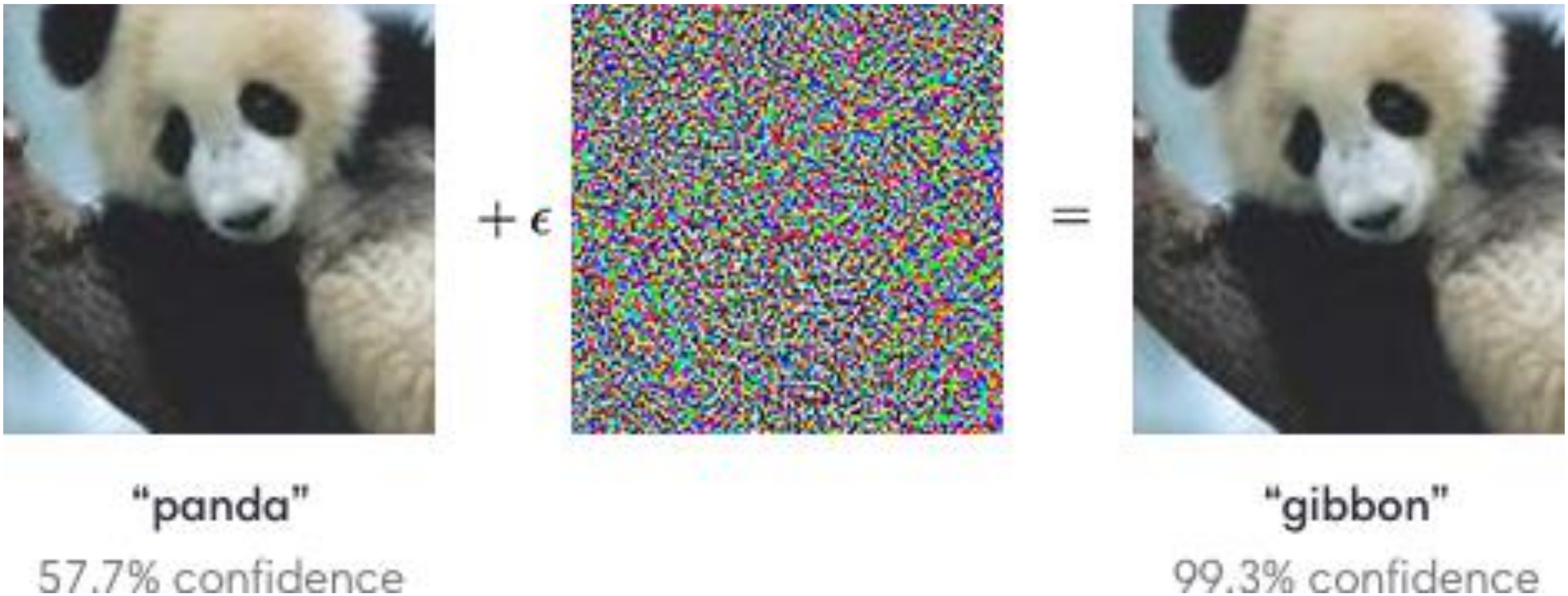
This is a convex optimization problem

$$\text{FONC are sufficient : try } z = x_2$$
$$2x_1 - 2x_2 - y = 0,\ \ 4x_2 - 2x_1 - 1 - y = 0$$
$$\Rightarrow 6x_2 - 4x_1 - 1 = 0 \Rightarrow x_2 = \frac{1}{2} \Rightarrow x_1 = \frac{1}{2}$$

- Deep Learning Based on Sample Writings

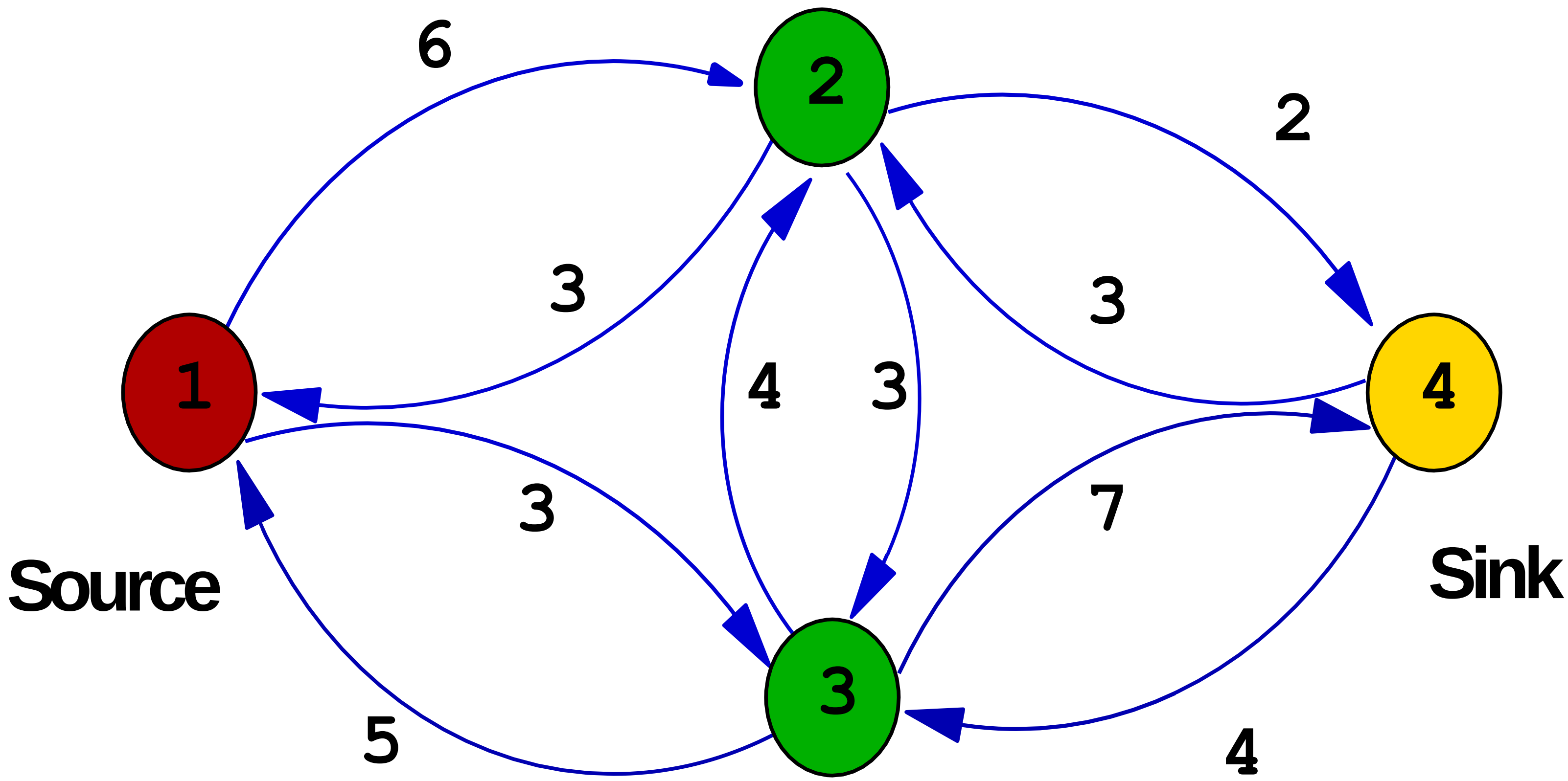


Sample-Based Learning May be Vulnerable



[Goodfellow et al. 14]

Look at a Max-Flow Problem



The Primal Formulation

Let x_{ij} be the flow rate from node i to node j. Then the problem can be formulated as

$$
\begin{array}{llll}
\max & x_{41} & & \\
\text{s.t.} & x_{21} + x_{31} + x_{41} - x_{12} - x_{13} & = 0, & y_1 \\
 & x_{12} + x_{32} + x_{42} - x_{21} - x_{23} - x_{24} & = 0, & y_2 \\
 & x_{13} + x_{23} + x_{43} - x_{31} - x_{32} - x_{34} & = 0, & y_3 \\
 & x_{24} + x_{34} - x_{41} - x_{42} - x_{43} & = 0, & y_4 \\
 & x_{ij} & \leq k_{ij},\ \forall (i, j) \in A, & z_{ij} \\
 & x_{ij} & \geq 0,\ \forall (i, j) \in A. &
\end{array}
$$

Corresponding Dual variables

The Dual of Max-Flow: the Min-Cut Problem

$$\min \quad \sum_{(i,j)\in A} k_{ij} z_{ij}$$

$$\begin{aligned}
\text{s.t.} \quad & y_1 - y_4 = 1, \\
& -y_1 + y_2 + z_{12} \geq 0, \\
& -y_1 + y_3 + z_{13} \geq 0, \\
& \ldots \\
& -y_2 + y_4 + z_{24} \geq 0, \\
& -y_3 + y_4 + z_{34} \geq 0, \\
& z_{ij} \geq 0, \ \forall (i,j) \in A.
\end{aligned}$$

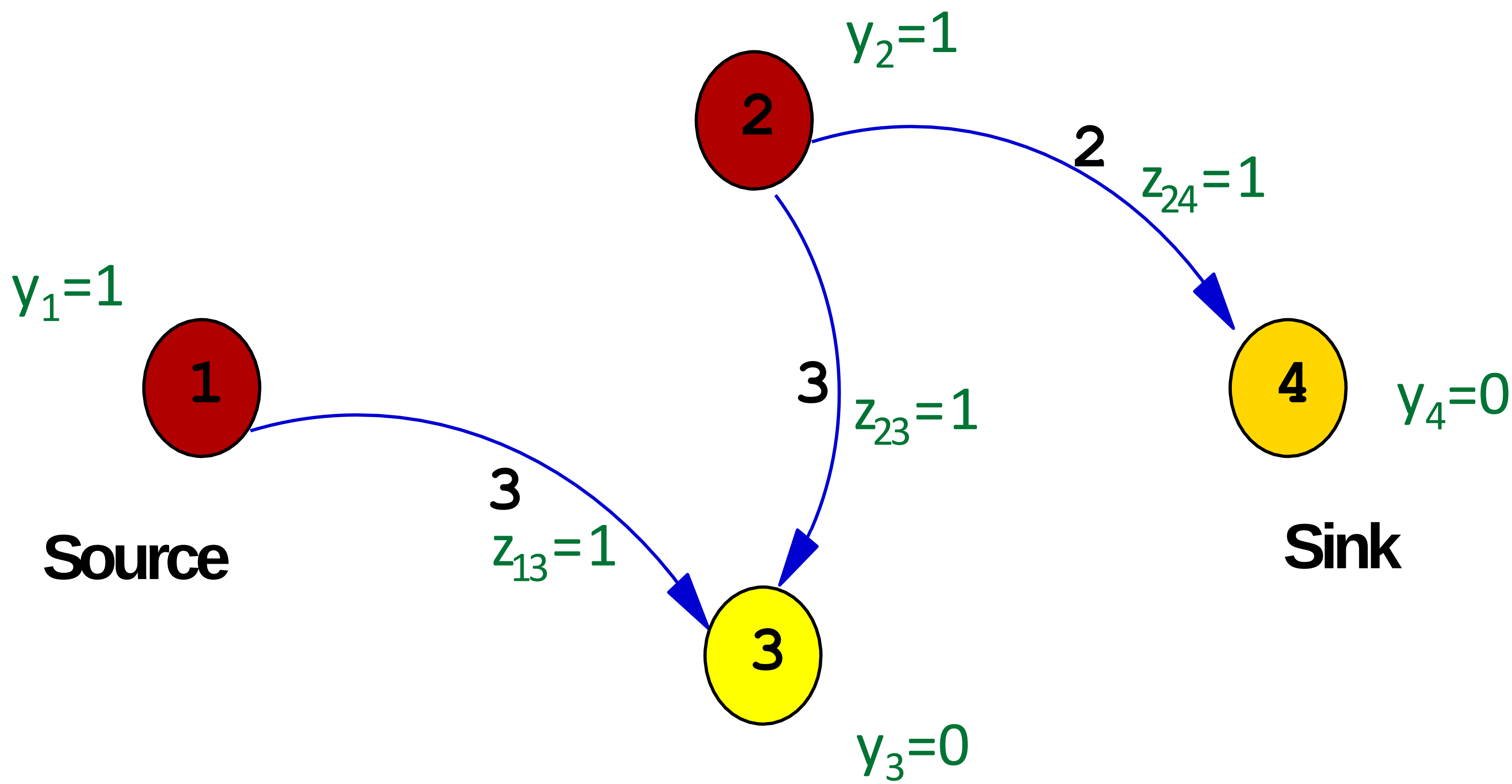
Corresponding Primal variables

x_{41}, x_{12}, x_{13}, …, x_{24}, x_{34}

y_i: node potential value; wlog set $y_4 = 0$ so that $y_1 = 1$ and at optimality for all other y_i:

$$y_i = \begin{cases} 1 & \text{if } i \text{ is on the source side} \\ 0 & \text{if } i \text{ is not in the source side} \end{cases} \quad \text{and} \quad z_{ij} = \begin{cases} 1, & \text{if } y_i=1 \text{ and } y_j=0 \\ 0 & \text{otherwise} \end{cases}$$

The Min-Cut Solution: Min-Cut Value=8



All other $z_{ij}=0$

Production Collaborative Game

Consider a finite set G of manufacture firms each of whom has operations that have representations as production linear programs: maximize the firm's profit $\mathbf{c}^T\mathbf{x}$ subject to the resource consumption constraint $A\mathbf{x} \le \mathbf{b}^i$, where $\mathbf{b}^i$ is the resource vector owned by firm I, that is,

$$V^i := \max \mathbf{c}^T\mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} \le \mathbf{b}^i, \quad \mathbf{x} \ge \mathbf{0}$$

For example

$$\begin{array}{llll} \max & x_1 & +2x_2 & \\ \text{s.t.} & x_1 & & \le b_1 \\ & & x_2 & \le b_2 \\ & x_1 + & x_2 & \le b_3 \\ & x_1 & x_2 & \ge 0 \end{array}$$

Let three firms A, B and C and each own resource vector:

$$\mathbf{b}^A = (1;\ 0;\ 0.5), \quad \mathbf{b}^B = (0;\ 1;\ 0.5), \quad \mathbf{b}^C = (0;\ 0;\ 0.5)$$

Alliances and Core in the Production Game

An (sub-)alliance is a subset of the firms to pool their resources together to find the optimal production mix. For any subset S of the firms, the quarantined pay-off, V^S, to the firms in S is the optimal objective value of the linear program:

$$V^S := \max \mathbf{c}^T\mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} \le \mathbf{b}^S := \textstyle\sum_{i \text{ in } S} \boldsymbol{b}^i, \;\; \mathbf{x} \ge \mathbf{0}$$

The Grand Alliance is the set G including all firms, and its total profit is exactly

$$V^G := \max \mathbf{c}^T\mathbf{x} \quad \text{s.t.} \quad A\mathbf{x} \le \mathbf{b}^S := \textstyle\sum_{i \text{ in } G} \boldsymbol{b}^i, \;\; \mathbf{x} \ge \mathbf{0}$$

The core of the grand alliance is set of payments, z^i, from the Grand Alliance to firm I, such that

$$\textstyle\sum_{i \text{ in } G} z^i = v^G \qquad \text{and} \qquad \sum_{i \text{ in } S} \boldsymbol{z}^i \ge v^S \;\; \textit{for all subset } S \textit{ of } G$$

In other words, no subgroup can do better by deserting the grand coalition to form their own sub-alliance.

Answer: Use the Shadow Prices of the Grand Production

Theorem *The core of the grand alliance of the LP production game exists. Moreover, one specific core allocation is to allocate each firm the value of its resource vector evaluated at the optimal shadow prices of the grand alliance production linear program.*

The pooled resources of the grand alliance are

$$\mathbf{b}^A + \mathbf{b}^B + \mathbf{b}^C = (1;\ 1;\ 1.5)$$

so it is the production problem we solved earlier with the maximal profit 2.5 and the optimal shadows prices

$$\boldsymbol{y}^T = (0 \quad 1 \quad 1).$$

Thus, the profit allocation to

$$\text{A:} \quad = \boldsymbol{y}^T\mathbf{b}^A = 0\cdot1+1\cdot0+1\cdot0.5=0.5$$
$$\text{B:} \quad = \boldsymbol{y}^T\mathbf{b}^B = 0\cdot0+1\cdot1+1\cdot0.5=1.5$$
$$\text{C:} \quad = \boldsymbol{y}^T\mathbf{b}^C = 0\cdot0+1\cdot0+1\cdot0.5=0.5$$

which is a core allocation.

The proof is entirely based on the LP weak and strong duality theorems.

The Dual of the Information Market Problem

The ith order is given as triple ($\mathbf{a}_i \in R^m, \pi_i \in R_+, q_i \in R_+$):

$$\mathbf{a}_i = (a_{i1}, a_{i2}, \ldots, a_{im})$$

is the betting indication row vector where each component is either 1 or 0, where 1 is winning state and 0 is non-winning state;

π_i is the bidding price for one share of such a contract, and

q_i is the maximum number of shares the bidder like to own.

A contract /share on an order is a paper agreement so that on maturity it is worth a notional \$1 dollar if the order includes the winning state and worth \$0 otherwise.

Let x_i be the number of units awarded to the ith order.

A Risk-Free Mechanism of Market Maker

Corresponding Dual Variables

$$
\begin{aligned}
\max \quad & \pi^T \mathbf{x} - x_{n+1} \\
\text{s.t.} \quad & A^T \mathbf{x} - \mathbf{1} \cdot x_{n+1} \le \mathbf{0} && \quad \boldsymbol{p} \\
& \mathbf{x} \le \mathbf{q} && \quad \boldsymbol{s} \\
& \mathbf{x} \ge \mathbf{0} \\
& x_{n+1} \text{ free}
\end{aligned}
$$

where $\mathbf{1}$ is the vector of all ones.

$\pi^T \mathbf{x}$: the revenue amount can be collected.

x_{n+1} : the worst-case cost (amount need to pay to the winners).

The Dual: Regression with "Under-Bid" Filtering

$$\begin{array}{lll} \min & \boldsymbol{q}^T \boldsymbol{s} & \\ \text{s.t.} & A\boldsymbol{p} + \boldsymbol{s} & \geq \boldsymbol{\pi}, \\ & -\boldsymbol{1}^T \boldsymbol{p} & = -1, \\ & (\boldsymbol{p}, \boldsymbol{s}) & \geq 0. \end{array}$$

$\boldsymbol{p}_j$: the shadow/dual price of state j;

$\boldsymbol{a}_i\boldsymbol{p}$: the ith order unit cost at prices $\boldsymbol{p}$;

$\boldsymbol{s}_{\mathrm{j}}$: the unit profit from the jth order ($\boldsymbol{s}=\max\{\boldsymbol{0}, \boldsymbol{\pi}-A\boldsymbol{p}\}$)

The dual problem is to minimize the total "Regression Loss" collected from the (competitive or high-bid) orders, $\boldsymbol{q}^T \boldsymbol{s}$.

ReLu-Regression for Probability Distribution/Information

$$\begin{array}{lll} \min & \boldsymbol{q}^T \max\{\boldsymbol{0}, \boldsymbol{\pi}\text{-}A\boldsymbol{p}\} & \\ \text{s.t.} & \boldsymbol{1}^T \boldsymbol{p} & = 1, \\ & \boldsymbol{p} & \geq 0 \end{array}$$

$\boldsymbol{p}_j$: the shadow-price/probability estimation of state j;

$\boldsymbol{a}_i\boldsymbol{p}$: the ith order unit cost at prices $\boldsymbol{p}$;

$\boldsymbol{\pi}_i$: the ith order bidding price;

$\boldsymbol{q}_i$: the ith order quantity limit;

The dual problem is to minimize the total weighted discrepancy among the competitive bidders such that all winners' betting beliefs $\boldsymbol{\pi}$ are fully utilized, while under-bidders (outliers) would be automatically removed from the estimation.

The World Cup Betting Example

Orders Filled

Order	Price Limit	Quantity Limit	Filled	Argentina	Brazil	Italy	Germany	France
1	0.75	10	5	1	1	1		
2	0.35	5	5				1	
3	0.40	10	5	1		1		1
4	0.95	10	0	1	1	1	1	
5	0.75	5	5		1		1	

State Prices

	Argentina	Brazil	Italy	Germany	France
Price	0.20	0.35	0.20	0.25	0.00

Online Retail Sell

- **There is a fixed selling period or number of buyers; and there is a fixed inventory of goods**
- **Customers come and require a bundle of goods and make a bid**
- **Decision: To sell or not to sell to each individual customer?**
- **Objective: Maximize the revenue.**

Bid #	$100	$30	….	…	…	Inventory
Decision	x1	x2				
Pants	1	0	….	…	…	100
Shoes	1	0				50
T-Shirts	0	1				500
Jackets	0	0				200
Hats	1	1	…	…	…	1000

On-Line Retailer Linear Programming

- Off-line Problem is an (0,1) linear program that can be relaxed as LP
- But now trader/Bidders come one by one sequentially,
- The retailer has to make the decision as soon as an order arrives with the arrived combinatorial order/bid $(\mathbf{a}_k, \pi_k)$
- The retailer faces a dilemma:
  - **To sell or not to sell – this is the decision**
- Optimal Policy or Online Algorithm?

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} \pi_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \forall i = 1, \ldots, m \\
& x_j = \{0 \; or \; 1\} \quad \forall j = 1, \ldots, n
\end{aligned}
$$

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{n} \pi_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{n} a_{ij} x_j \le b_i \quad \forall i = 1, \ldots, m \\
& 0 \le x_j \le 1 \quad \forall j = 1, \ldots, n
\end{aligned}
$$

Off-Line LP Relaxation

CSC of Off-Line Retailer Linear Programming

- Let the optimal solution be $\boldsymbol{x}^*$ and the optimal shadow piece be $\boldsymbol{y}^*$
- Then from the CSC conditions:

  $x_j^* = 1$ if $\pi_j > \mathbf{a}_j^T \boldsymbol{y}^*$

  $x_j^* = 0$ if $\pi_j < \mathbf{a}_j^T \boldsymbol{y}^*$

  $x_j^* =$ fraction if $\pi_j = \mathbf{a}_j^T \boldsymbol{y}^*$

- If we know $\boldsymbol{y}^*$, the online decision would be easy!

$$
\begin{array}{lll}
\max & \sum_{j=1}^{n} \pi_j x_j & \\
\text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j \le b_i & \forall i = 1, \ldots, m \\
 & 0 \le x_j \le 1 & \forall j = 1, .., n
\end{array}
$$

Off-Line LP Relaxation

Online Algorithm and Price-Mechanism

- **Learn "ideal" itemized optimal prices**
- **Use the prices to price each bid**
- **Accept if it is a over bid, and reject otherwise**

Bid #	$100	$30	….	…	…	Inventory	Price?
Decision	x1	x2					
Pants	1	0	….	…	…	100	45
Shoes	1	0				50	45
T-Shirts	0	1				500	10
Jackets	0	0				200	55
Hats	1	1	…	…	…	1000	15

Such ideal prices exist, and they are shadow/dual prices of the offline LP

How to Learn the Shadow Prices Sequentially?

- Sequential Linear Programming Mechanism (SLPM)
  - Solving the LP based on immediately past several periods' data and use the resulted optimal shadow prices to make decision for the next period orders; and repeat when the current period is over.

- The shadow prices are updated periodically and being used to make online decisions for the next period.

Wait for Data from 1 to εn

- Set $x_j=0$ for $j=1,\ldots,\varepsilon n$.

- Solve LP:

$$\begin{array}{lll} \max & \sum_{j=1}^{\varepsilon n} \pi_j x_j & \\ \text{s.t.} & \sum_{j=1}^{\varepsilon n} a_{ij} x_j \leq \varepsilon b_i & \forall i \\ & 0 \leq x_j \leq 1 & \forall j \end{array}$$

- Let $\boldsymbol{p}^1$ be the optimal shadow price vector and use it to make online decision for orders from $\varepsilon n+1$ to $2\varepsilon n$.

Now Use All Data from 1 to 2εn

- Now solve LP:

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{2\varepsilon n} \pi_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{2\varepsilon n} a_{ij} x_j \le 2\varepsilon b_i \qquad \forall i \\
& 0 \le x_j \le 1 \qquad \forall j
\end{aligned}
$$

- Let $\boldsymbol{p}^2$ be the optimal shadow price vector and use it to make online decision for orders from $2\varepsilon n+1$ to $4\varepsilon n$.

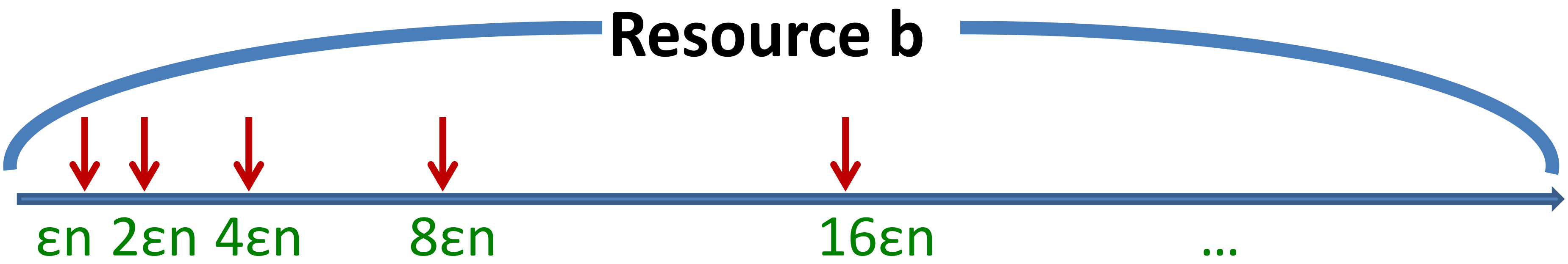
Now Use All Data from 1 to 4εn

- Now solve LP:

$$
\begin{array}{lll}
\max & \sum_{j=1}^{4\varepsilon n} \pi_j x_j & \\
\text{s.t.} & \sum_{j=1}^{4\varepsilon n} a_{ij} x_j \leq 4\varepsilon b_i & \forall i \\
 & 0 \leq x_j \leq 1 & \forall j
\end{array}
$$

- Let $\boldsymbol{p}^3$ be the optimal shadow price vector and use it to make online decision for orders from *4εn+1* to *8εn*.

Use Observed Data: Decisioning while Learning



Resources allocated at each update point is proportional to the number of customers already arrived.

Theorem: Let the orders come randomly and let

$$\min_i \{ b_i \} \geq m\log(n)/\varepsilon^2.$$

Then
the expected online revenue ≥ (1 – ε) the offline revenue.

On the other hand, if $\min_i \{ b_i \} < \log(m)/\varepsilon^2$.
then no mechanism/algorithm can achieve the (1 – ε) guarantee.

Adaptively Update Prices after every Batch of Orders using the Remaining Average Inventory

- Now solve LP:

$$
\begin{aligned}
\max \quad & \sum_{j=1}^{k} \pi_j x_j \\
\text{s.t.} \quad & \sum_{j=1}^{k} a_{ij} x_j \leq (k/(n-k+1)) bk_i && \forall i \\
& 0 \leq x_j \leq 1 && \forall j
\end{aligned}
$$

- Here $\boldsymbol{b}^k$ is the remaining inventory before the next batch of orders arrive.

Update Prices by the Gradient Method

1: Initialize $\mathbf{p}_1 = \mathbf{0}, \ \mathbf{b}_1 = \mathbf{b}$

2: For $k = 1, ..., n$

3: Decide the kth Order $x_k = \begin{cases} 1, & \pi_k > \mathbf{a}_k^\top \mathbf{p}_k \\ 0, & \text{otherwise} \end{cases}$

4: Update Shadow Prices:

$$\begin{aligned} \mathbf{p}_{k+1} &= \mathbf{p}_k + \alpha_k \left(\mathbf{a}_k x_k - \frac{\mathbf{b}_k}{n-k+1} \right) \\ \mathbf{p}_{k+1} &= \mathbf{p}_{k+1} \vee \mathbf{0} \\ \text{stepsize } \alpha_k &= \frac{1}{\sqrt{n}} \text{ or } \frac{1}{\sqrt{k}} \end{aligned}$$

5: Update Remaining Inventory: $\mathbf{b}_{k+1} = \mathbf{b}_k - \mathbf{a}_k x_k$