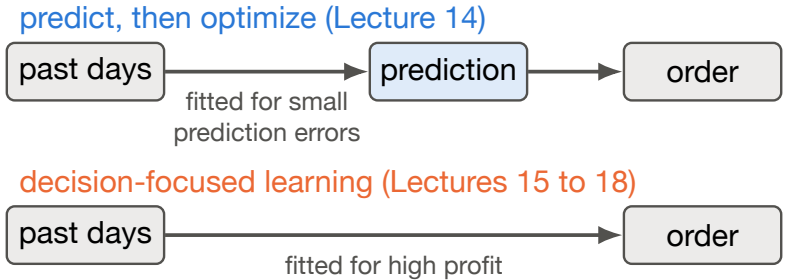
Learning the policy directly

208424 Lecture 15

Lecture 14: predict, then optimize

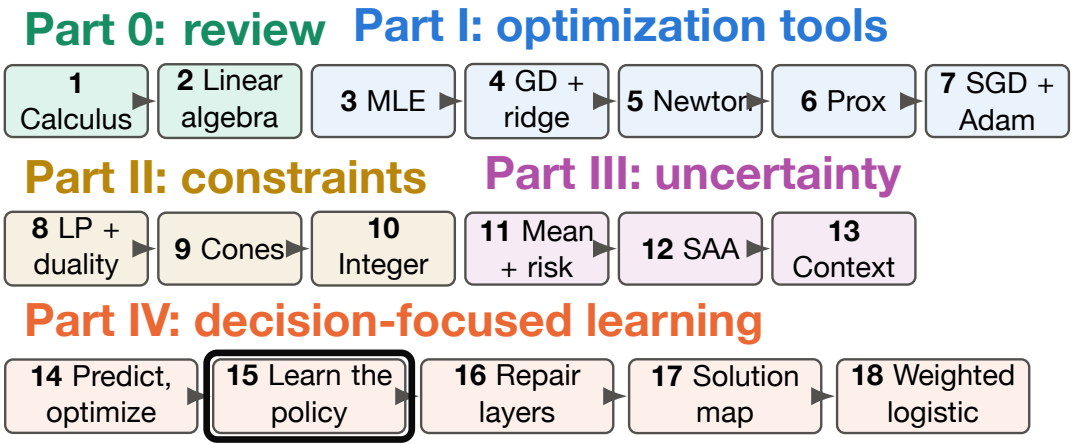


Good predictions did not always give good orders.

Method 4 learns the policy from past days.

Today, we learn policies by simulating past days.

Course map


Part 0: review
Part I: optimization tools
1 Calculus
2 Linear algebra
3 MLE
4 GD + ridge
5 Newton
6 Prox
7 SGD + Adam
Part II: constraints
Part III: uncertainty
8 LP + duality
9 Cones
10 Integer
11 Mean + risk
12 SAA
13 Context
Part IV: decision-focused learning
14 Predict, optimize
15 Learn the policy
16 Repair layers
17 Solution map
18 Weighted logistic

Lecture 15: Learning the policy directly



Photo: Kenta Odajima, CC BY-SA 4.0, Wikimedia Commons

A café learns a policy from its past days.

Unsold drinks can be sold on the next day.

A policy on three past days

policy
$x = 60 + 15z$

$z = 2,\ D = 70$	$z = 6,\ D = 170$	$z = 8,\ D = 170$
order 90 profit 3800	order 150 profit 9000	order 180 profit 10000

We know z and D for every past day.

So we know what any policy would have earned.

For this policy, the average profit is 7600 baht.

The average profit of a policy

$$\hat{V}(b_0, b_1) = \underbrace{\frac{1}{n}}_{\text{average over the } n \text{ past days}} \sum_{i=1}^{n} \underbrace{\pi_i(x_i)}_{\pi_i(x) = 80 \min(D_i, x) - 20x \text{ with } x_i = b_0 + b_1 z_i}$$

We write $\hat{V}(b_0, b_1)$ for the average profit.
Simulating the past days with a new policy is called counterfactual simulation.

The optimal order in hindsight

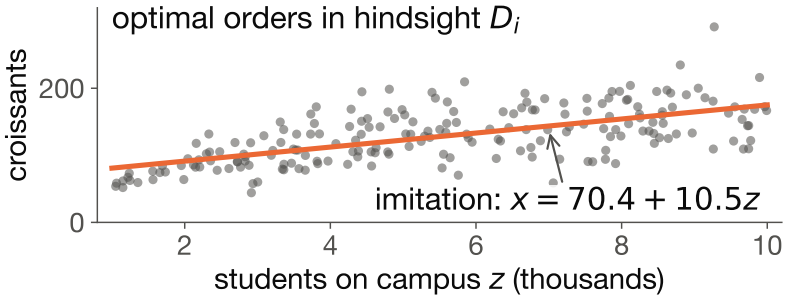
optimal in hindsight	$z = 2,\ D = 70$	$z = 6,\ D = 170$	$z = 8,\ D = 170$
	order 70 profit 4200	order 170 profit 10200	order 170 profit 10200

After day i, we know the demand D_i.

The optimal order in hindsight is exactly D_i.

This order earns $60D_i$ baht.

Imitation

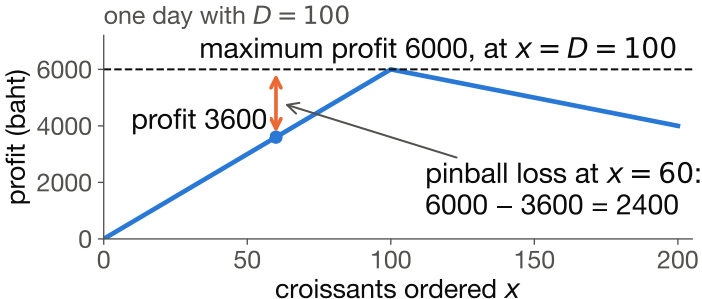


Imitation fits a policy to the optimal orders.

With least squares, imitation fits the average.

So imitation orders the predicted average.

Profit and the pinball loss

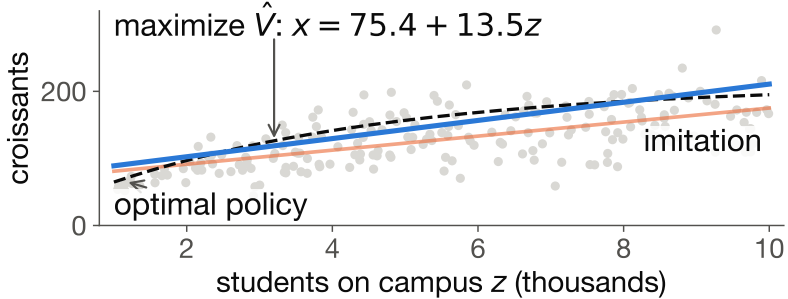


If $D = 100$, the maximum profit is 6000 baht.

The pinball loss is the profit lost

compared with the optimal order in hindsight.

Maximizing the average profit

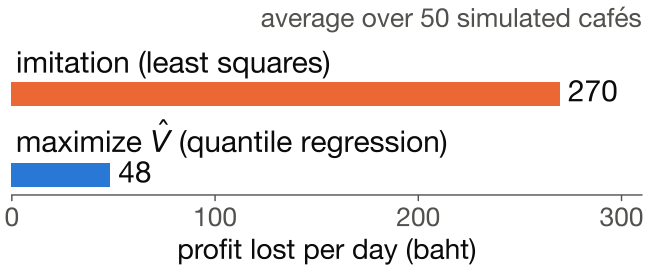


Method 4 chooses b_0 and b_1 to maximize $\hat{V}$.

Maximizing $\hat{V}$ minimizes the average pinball loss.

The fitted policy is $x(z) = 75.4 + 13.5z$.

Profit lost on new days

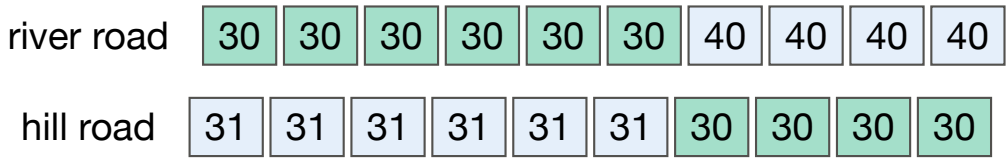


Imitation loses 270 baht per day.

Maximizing $\hat{V}$ loses only 48 baht per day.

Imitation ignores the costs of 60 and 20 baht.

Imitation for a rider



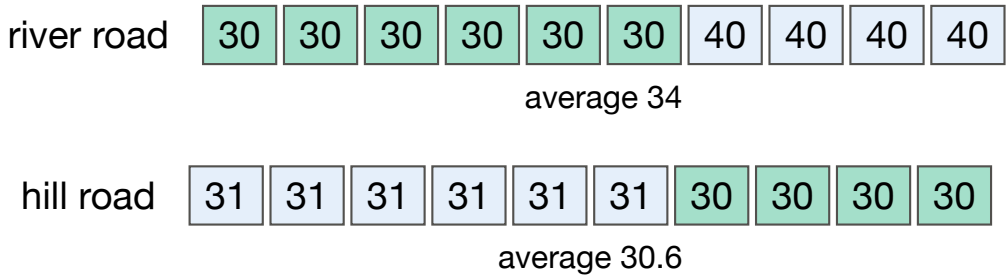
minutes on 10 past trips (the faster road of each trip is shaded)

Imitation chooses the road that is faster more often.

The river road was faster on 6 of the 10 trips.

So imitation chooses the river road.

The average travel time

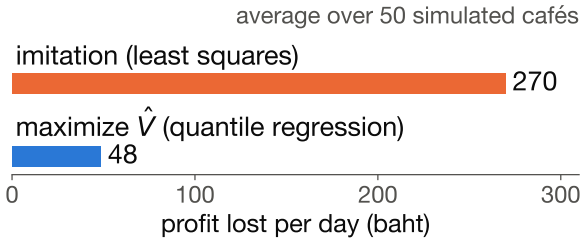


The river road takes 34 minutes on average.

The hill road takes 30.6 minutes on average.

Imitation loses 3.4 minutes per trip.

Summary: imitate or maximize



Imitation ignores how much a mistake costs.

Maximizing $\hat{V}$ uses the profit of every past day.

Checkpoint 1

north road	20	20	20	20	20	20	20	50	50	50
south road	25	25	25	25	25	25	25	25	25	25

minutes on 10 past trips

Which road does imitation choose?

Which road is faster on average?

Checkpoint 1

north road	20	20	20	20	20	20	20	50	50	50	average 29
south road	25	25	25	25	25	25	25	25	25	25	average 25

minutes on 10 past trips

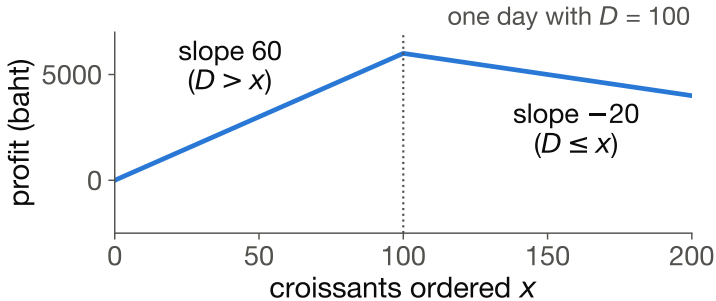
Which road does imitation choose?

Which road is faster on average?

Imitation chooses the north road.

The south road is faster on average ($25 < 29$).

The slope of one day's profit (Lecture 12)

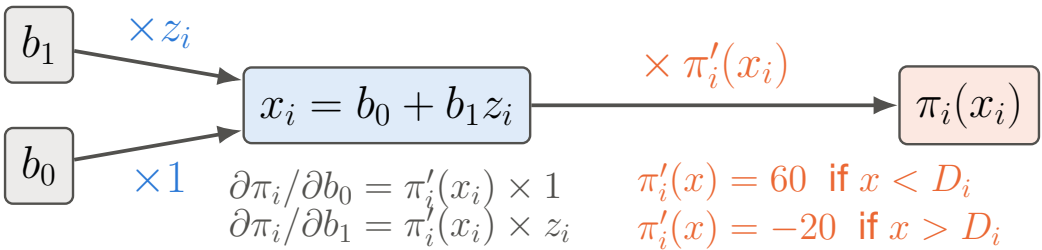


If $D > x$, one more croissant adds 60 baht.

If $D \leq x$, one more croissant loses 20 baht.

Unlike an LP, SGD needs only this slope.

The chain rule (Lecture 1)

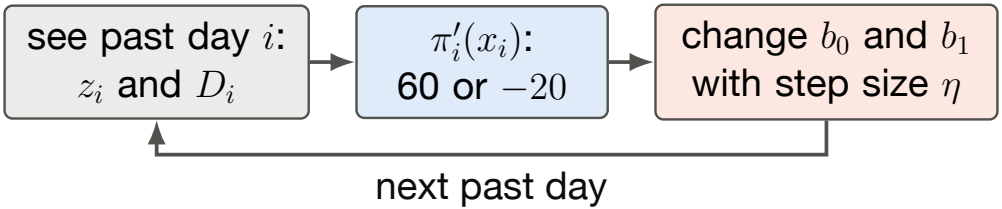


When b_1 increases by 1, the order x_i increases by z_i.

By the chain rule, we multiply the derivatives.

So $\partial \pi_i / \partial b_1$ is $60 z_i$ or $-20 z_i$.

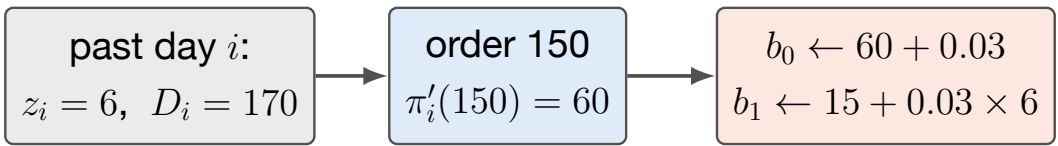
SGD for a linear policy



$$b_0 \leftarrow b_0 + \eta \times \pi'_i(x_i)$$
$$b_1 \leftarrow b_1 + \eta \times \pi'_i(x_i) \times z_i$$

We add the derivative because we maximize profit.

One SGD step

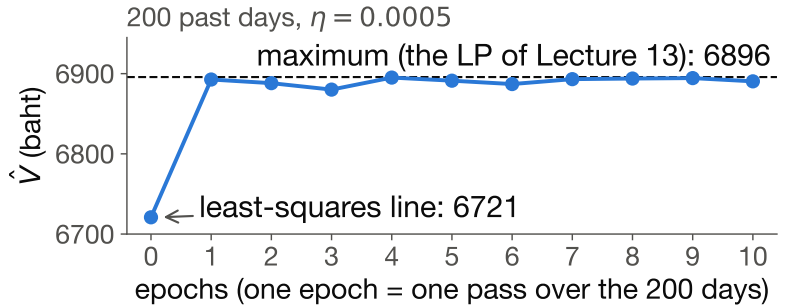


The policy $x = 60 + 15z$ orders 150 for $z = 6$.

Since $D_i = 170 > 150$, $\pi_i'(150) = 60$.

So b_0 becomes 60.03, and b_1 becomes 15.18.

SGD on the 200 past days

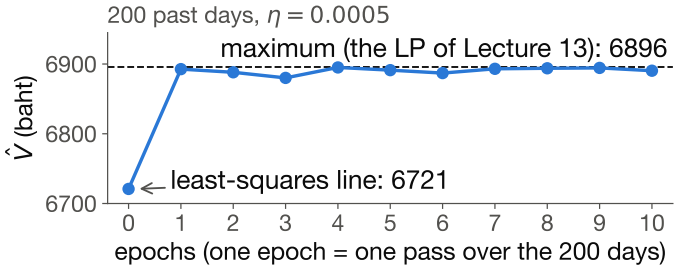


SGD starts at the least-squares line.

After one epoch, $\hat{V}$ is close to its maximum.

Then $\hat{V}$ moves up and down just below 6896.

Summary: SGD for a linear policy



SGD uses the derivative π_i' of one day's profit.

The chain rule gives $\partial \pi_i / \partial b_0$ and $\partial \pi_i / \partial b_1$.

Checkpoint 2

policy: $x = 60 + 15z$
step size $\eta = 0.0005$

SGD now uses a past day i with $z_i = 4$ and $D_i = 100$.

What are the new b_0 and b_1?

Checkpoint 2

policy: $x = 60 + 15z$
step size $\eta = 0.0005$

$$\eta \times (-20) = -0.01$$
$$b_0 \leftarrow 60 - 0.01$$
$$b_1 \leftarrow 15 - 0.01 \times 4$$

SGD now uses a past day i with $z_i = 4$ and $D_i = 100$.

What are the new b_0 and b_1?

The order is $120 > 100$, so $\pi_i'(120) = -20$.

So b_0 becomes 59.99, and b_1 becomes 14.96.

Bottled drinks



Photo: Kenta Odajima, CC BY-SA 4.0, Wikimedia Commons

a bottle costs 20 baht
and sells for 80 baht

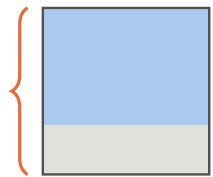
The café also sells bottled drinks.

Unsold bottles stay in the fridge for later days.

Keeping a bottle costs 2 baht per night.

The order-up-to level

level $S(z) = 100$



bought: $100 - 30 = 70$

left from yesterday: 30

Each evening, the café sees tomorrow's z.

The café buys bottles up to the level $S(z)$.

The level $S(z)$ is called the order-up-to level.

One bottle too few or too many

one bottle too few

-60 baht

one bottle too many

-2 baht per night

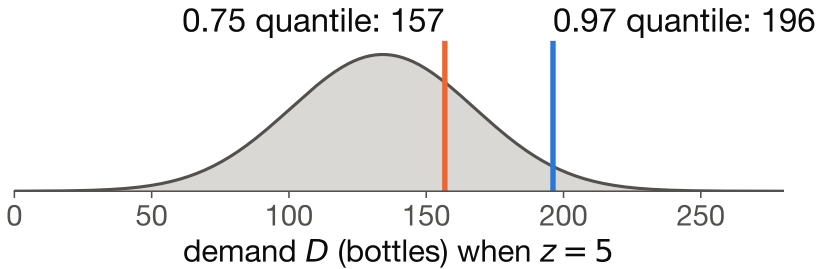
(sold on a later day)

One bottle too few loses a profit of 60 baht.

One bottle too many costs only 2 baht for one night.

For croissants, one too many loses 20 baht.

The 0.97 quantile for drinks



For drinks, the quantile is $60/(60+2) \approx 0.97$.

The nurses of Lecture 13 used the same formula.

Both quantiles use the true demand distribution.

Three past days

	level $S(z)$	left from yesterday	bought	demand D	sold	left tonight
day 1, $z = 6$	100	0	100	70	70	**30**
day 2, $z = 4$	80	**30**	50	90	80	0
day 3, $z = 8$	120	0	120	110	110	10

day 2: sold out (80 bottles in the fridge)

The policy is $S(z) = 40 + 10z$.

The 30 bottles left on day 1 are sold on day 2.

So we must simulate the days in order.

The profit of each day

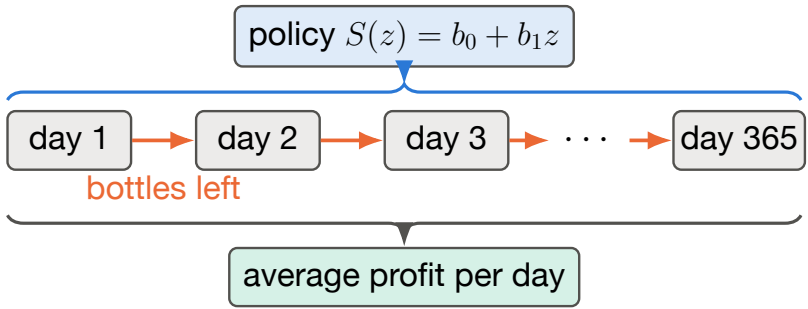
	bought	sold	left tonight	profit
day 1	100	70	30	**3540**
day 2	50	80	0	5400
day 3	120	110	10	6380

profit $= 80 \times$ sold $- 20 \times$ bought $- 2 \times$ left tonight

The profit on day 1 is

$80 \times 70 - 20 \times 100 - 2 \times 30 = 3540$ baht.

Counterfactual simulation of a past year

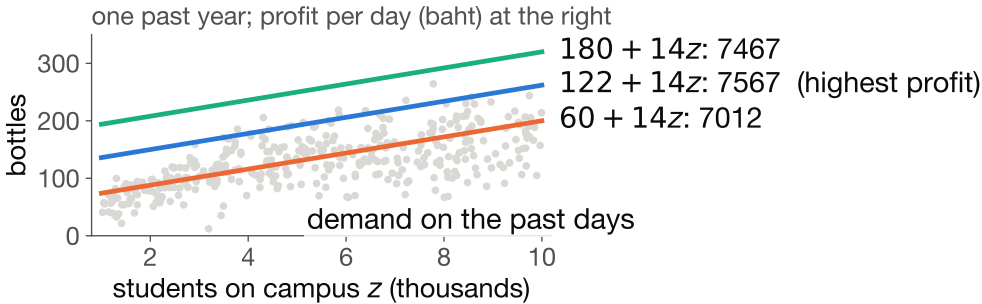


The café has z and D for 365 past days.

We simulate the year again with a new policy $S(z)$.

The simulation gives the average profit of $S(z)$.

Three lines on the past year

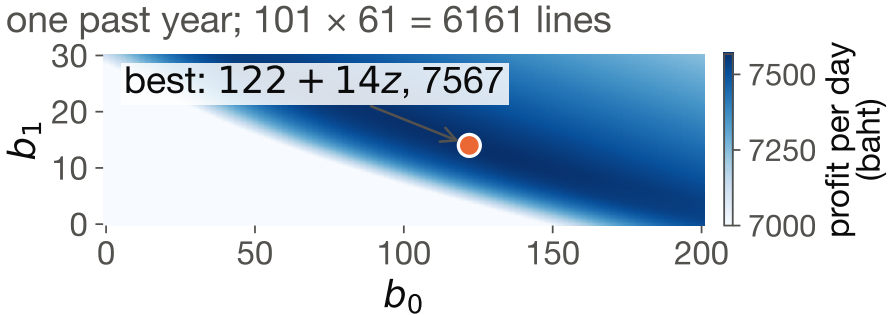


Each line $S(z) = b_0 + b_1 z$ is one policy.

We simulate the past year once for each line.

We keep the line with the highest profit.

A grid search

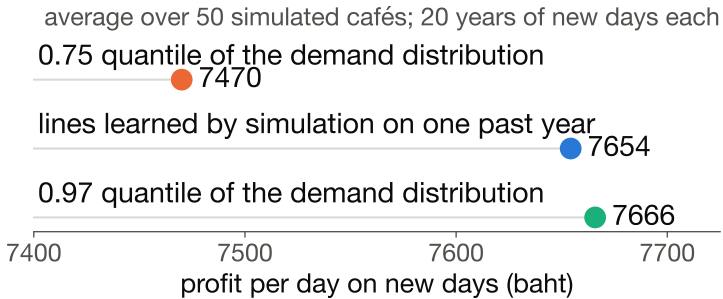


We try $b_0 = 0, 2, \ldots, 200$ and $b_1 = 0, 0.5, \ldots, 30$.

Trying every pair (b_0, b_1) is called a grid search.

The best line on this year is $S(z) = 122 + 14z$.

Profit on new days



Lines learned by simulation earn 184 baht per day more than the 0.75 quantile.

These lines earn 12 baht less than the 0.97 quantile.

When can we simulate past days?

a sold-out day:
80 bottles, 80 sold
demand: 80 or more?

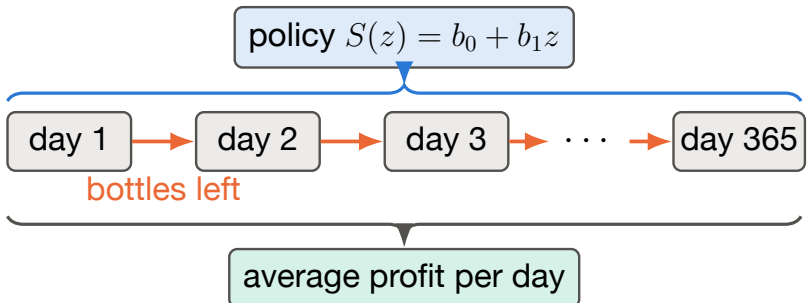
our orders must not
change the demand

Simulation needs the demand D of every past day.

On a sold-out day, the sales do not show D.

Also, our orders must not change the demand.

Summary: simulating past days



We simulate the past days again, in order.

From one past year, simulation found a good policy.

Checkpoint 3

empty fridge before day 1

level $S = 100$ every evening

day 1: $D = 70$
day 2: $D = 80$

How many bottles does the café buy for day 2?

How many bottles are left after day 2?

Checkpoint 3

empty fridge before day 1	level $S = 100$ every evening	day 1: $D = 70$ day 2: $D = 80$

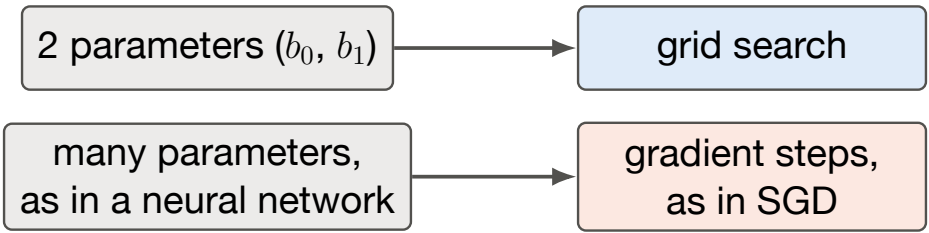
How many bottles does the café buy for day 2?

How many bottles are left after day 2?

After day 1, 30 bottles are left, so the café buys 70.

The café sells 80, so $100 - 80 = 20$ bottles are left.

Policies with many parameters



A grid search works for two parameters, b_0 and b_1.

For many parameters, we take gradient steps.

The next two studies use gradient steps (Lecture 7).

Inventory at Amazon (Lecture 14)

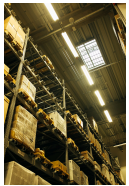


Photo (a warehouse, not Amazon's): Shixart1985, CC BY 2.0, Wikimedia Commons

Madeka et al. (2022)

Researchers at Amazon built a simulator.

In a test on past data, their policy earned more profit than a newsvendor policy.

Groceries in Ecuador

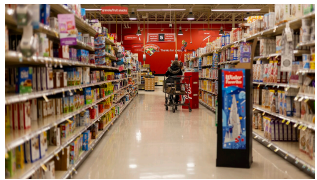
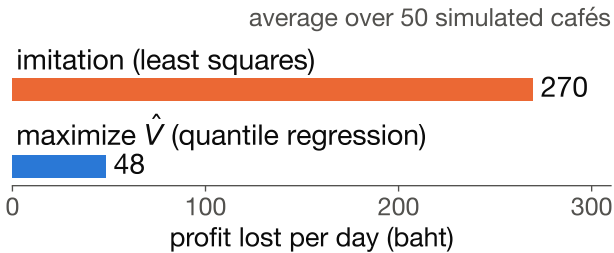


Photo (a store in the United States): Tessa Bury, CC BY 4.0, Wikimedia Commons

Alvo et al. (2025)

The data came from 54 grocery stores in Ecuador. Policies trained by simulation earned more profit than predict, then optimize did.

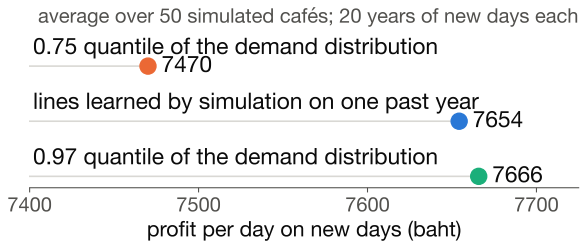
Lecture summary



Maximizing $\hat{V}$ loses much less than imitation.

SGD learns b_0 and b_1 with the chain rule.

Lecture summary (continued)

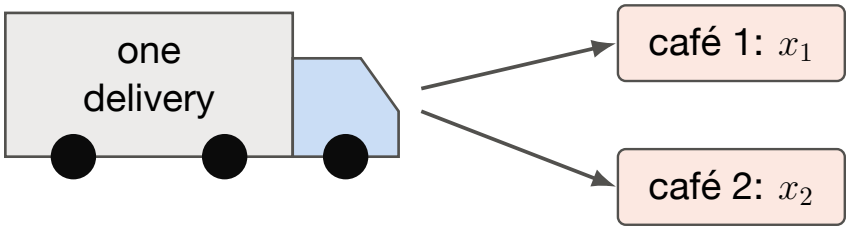


We simulate the past days again, in order.

For drinks, simulation found a good policy.

Simulation needs the demand of every past day.

Next lecture



In Lecture 16, two cafés share one truck.

Each café has its own linear policy.

On busy days, the orders may not fit in the truck.

Exercises

1 east road | 15 | 15 | 15 | 15 | 40 | west road | 18 | 18 | 18 | 18 | 18 |

2 policy $x = 50 + 20z$, $\eta = 0.0005$; past day $z = 5$, $D = 200$

3 level $S = 100$; empty fridge before day 1
demand on three days: 60, 120, 80

1. Which road does imitation choose?
2. Find b_0 and b_1 after one SGD step.
3. Find how many bottles the café buys each day.