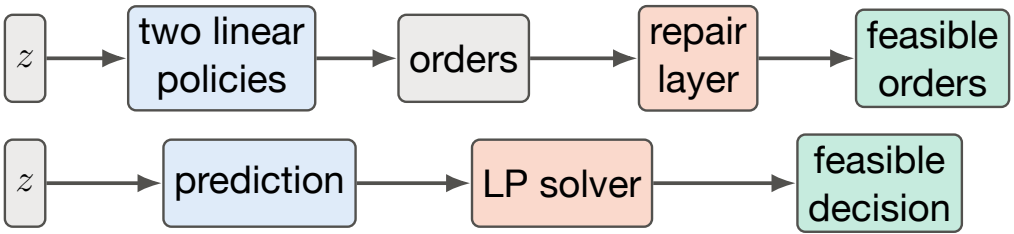
Plug-in policies and the solution map

208424 Lecture 17

Lecture 16: repair layers

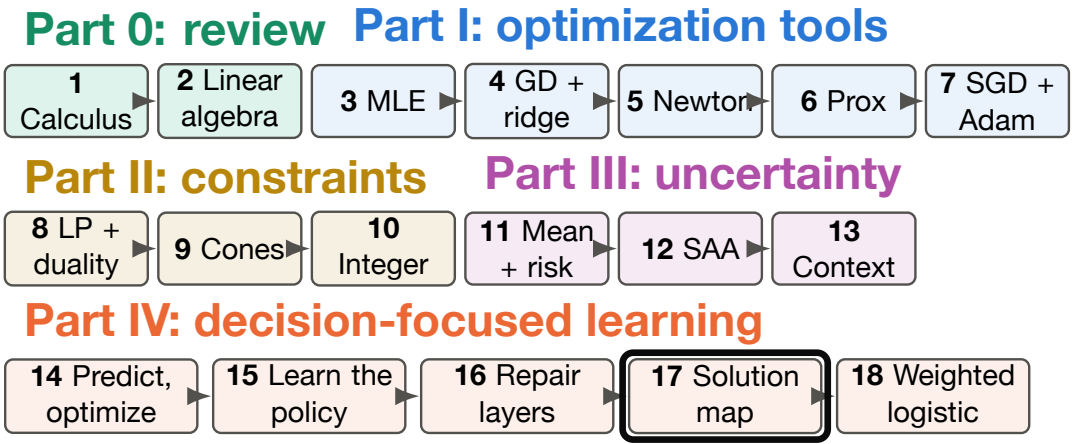


In Lecture 16, two cafés shared one truck.

A repair layer made the orders feasible.

Today, a solver chooses a feasible decision.

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 17:
Plug-in policies and the solution map



Photo: Dr. Manavpreet Kaur, CC BY-SA 4.0, Wikimedia Commons

A bakery predicts its profits, then solves an LP.

We study how the plan changes with the prediction.

The profit per croissant changes





profit per croissant: $c_1 = ?$ profit per sandwich: $c_2 = 40$

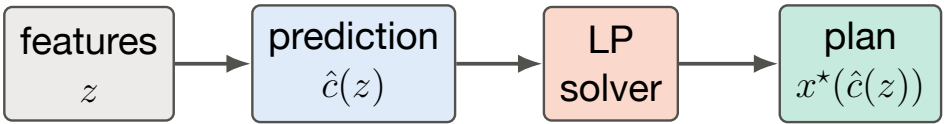
Photos (Wikimedia Commons): Petr Kratochvil, CC0; Andy Li, CC0.

In Lecture 8, the profits were $c = (c_1, c_2) = (50, 40)$.

Now c_1 depends on the price of butter.

The bakery must plan before c_1 is known.

A plug-in policy



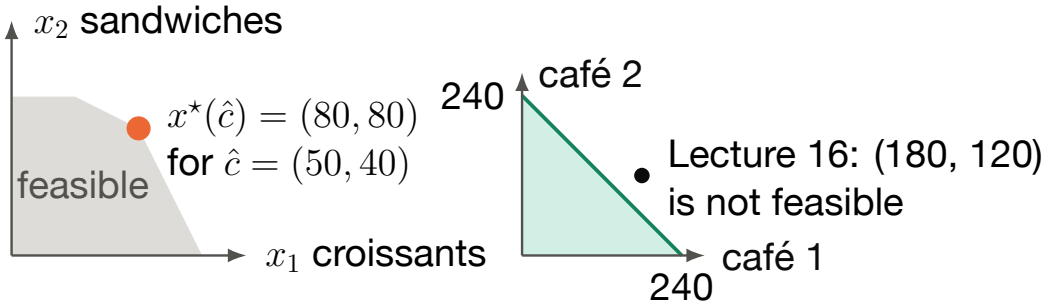
$\hat{c}(z) = (\hat{c}_1(z), 40)$

$x^\star(c)$: the feasible x that maximizes $c^\top x$

A plug-in policy predicts, then optimizes as if the prediction were true.

For an LP, the policy is $x(z) = x^\star(\hat{c}(z))$.

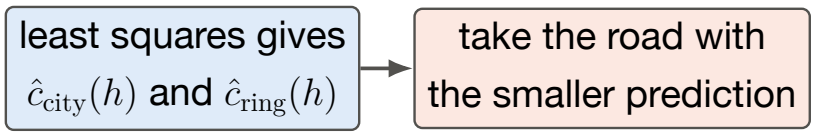
Plans from a solver are feasible



A solver gives only feasible plans.

So a plug-in policy needs no repair layer.

The rider in Lecture 14

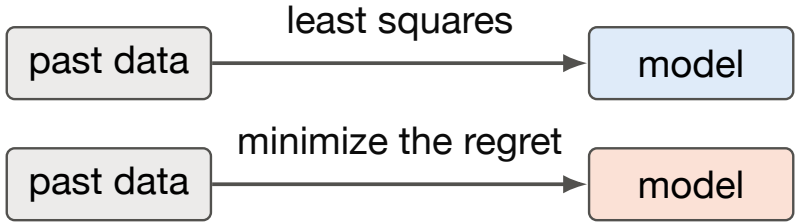


In Lecture 14, least squares predicted the times.

The rider took the road with the smaller prediction.

So “predict, then optimize” is a plug-in policy.

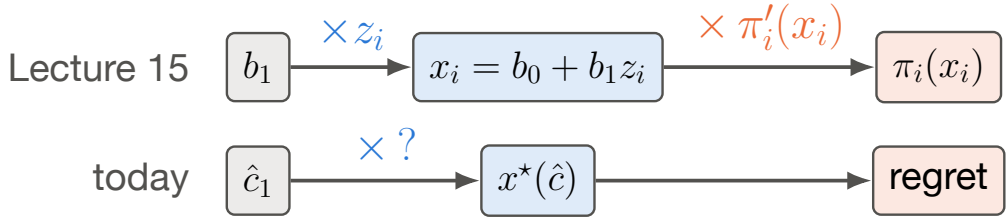
Training for regret



Least squares minimizes the sum of squared errors.

A plug-in policy trained for regret fits the model by minimizing the average regret on past data.

Can SGD train for regret?

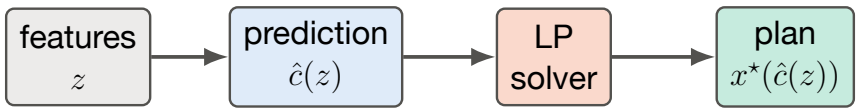


In Lecture 15, SGD used the chain rule.

Can SGD also train a plug-in policy for regret?

The answer depends on how $x^\star(\hat{c})$ changes with $\hat{c}$.

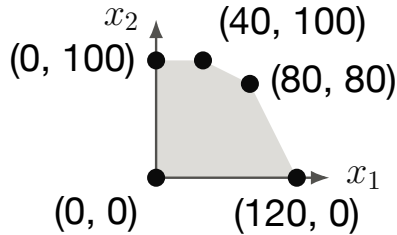
Summary: plug-in policies



A plug-in policy predicts, then optimizes.

For an LP, the solver's plan is always feasible.

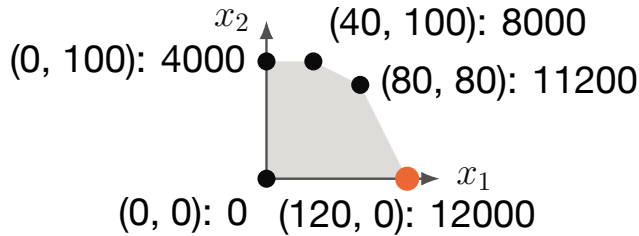
Checkpoint 1



The bakery predicts $\hat{c} = (100, 40)$.

Which plan does the solver give?

Checkpoint 1



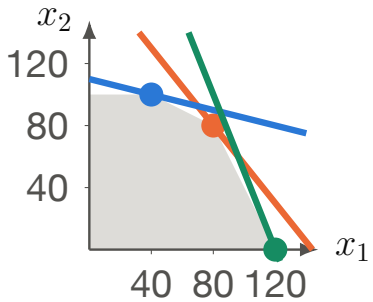
The bakery predicts $\hat{c} = (100, 40)$.

Which plan does the solver give?

At the vertices, $100x_1 + 40x_2$ is largest at $(120, 0)$.

So the solver gives the plan $(120, 0)$.

The optimal plan for three values of c_1



lines of equal profit

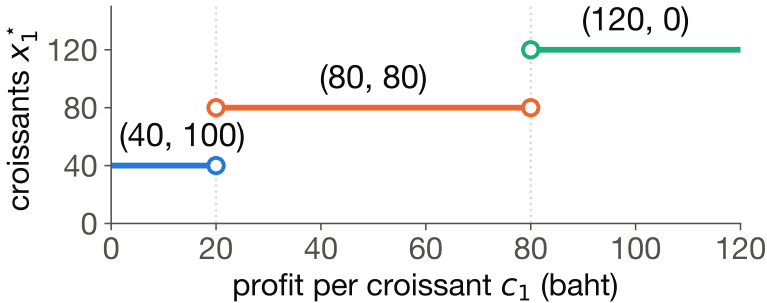
- $c_1 = 50$
- $c_1 = 10$
- $c_1 = 100$

For $c_1 = 50$, the optimal plan is $(80, 80)$.

For $c_1 = 10$, the optimal plan is $(40, 100)$.

For $c_1 = 100$, the optimal plan is $(120, 0)$.

The solution map

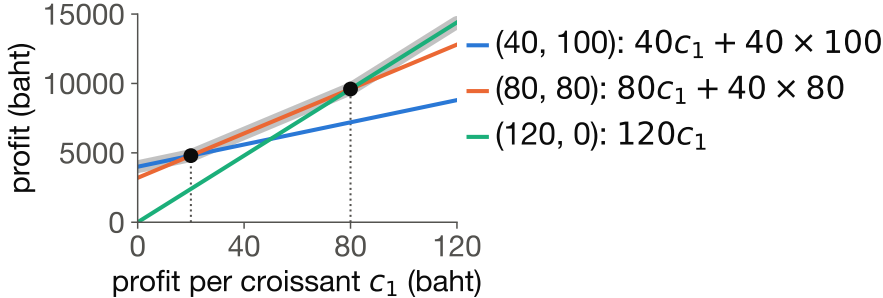


The solution map gives $x^\star(c)$ for every c.

For $c_1 > 0$, $x^\star$ changes only at $c_1 = 20$ and 80.

Such a function is called piecewise constant.

The profit of each vertex

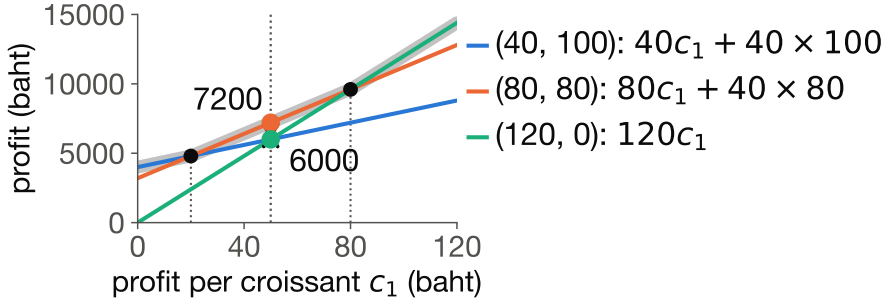


The profit of a vertex is linear in c_1.

The optimal plan is the vertex with the highest line.

The highest line changes at $c_1 = 20$ and $c_1 = 80$.

The regret of a plan

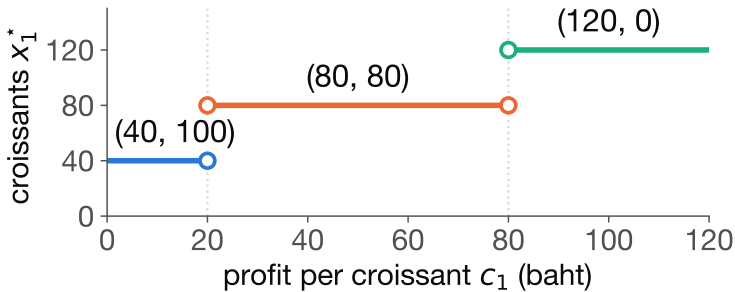


The regret is $c^\top x^\star(c) - c^\top x^\star(\hat{c})$, with the true c.

With $\hat{c}_1 = 100$ and $c_1 = 50$, the plan is $(120, 0)$.

So the regret is $50 \times 80 + 40 \times 80 - 50 \times 120 = 1200$.

Summary: the solution map



The bakery's solution map is piecewise constant.

The regret is $c^\top x^\star(c) - c^\top x^\star(\hat{c})$.

Checkpoint 2

prediction: $\hat{c}_1 = 50$

true value: $c_1 = 90$

$c_2 = 40$

The prediction is $\hat{c}_1 = 50$, and the true c_1 is 90.

What is the regret of the solver's plan?

Checkpoint 2

prediction: $\hat{c}_1 = 50$

true value: $c_1 = 90$

$c_2 = 40$

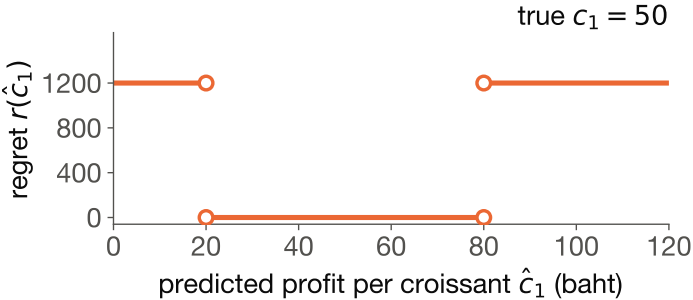
The prediction is $\hat{c}_1 = 50$, and the true c_1 is 90.

What is the regret of the solver's plan?

The optimal plan for $c_1 = 90$ is $(120, 0)$.

So the regret is $90 \times 120 - (90 \times 80 + 40 \times 80) = 400$.

Regret against the prediction

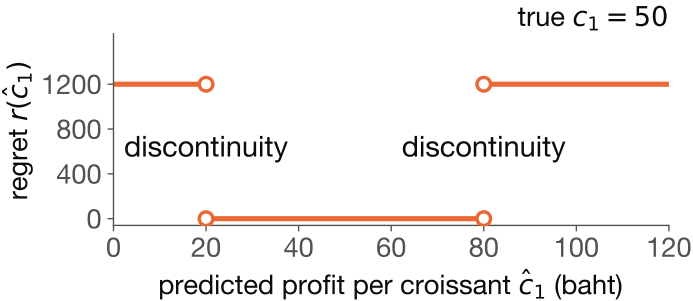


We write $r(\hat{c}_1)$ for the regret with the prediction $\hat{c}_1$.

With $c_1 = 50$, $r(\hat{c}_1) = 1200$ if $0 < \hat{c}_1 < 20$ or $\hat{c}_1 > 80$,

and $r(\hat{c}_1) = 0$ if $20 < \hat{c}_1 < 80$.

Discontinuities

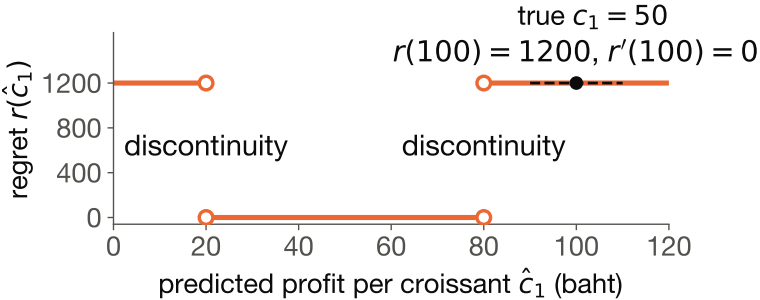


At $\hat{c}_1 = 20$ and 80, $r(\hat{c}_1)$ changes suddenly.

These points are called discontinuities.

So r is piecewise constant in $\hat{c}_1$.

The derivative of the regret

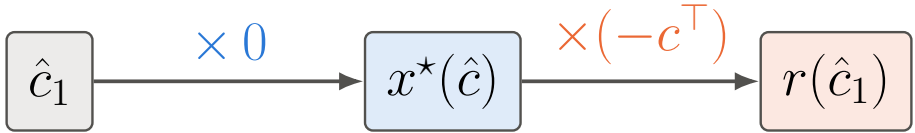


For $\hat{c}_1 > 0$, $r'(\hat{c}_1)$ is 0, except at $\hat{c}_1 = 20$ and 80.

At $\hat{c}_1 = 20$ and 80, r' does not exist.

Since $r'(100) = 0$, gradient descent stays at 100.

The chain rule for the bakery

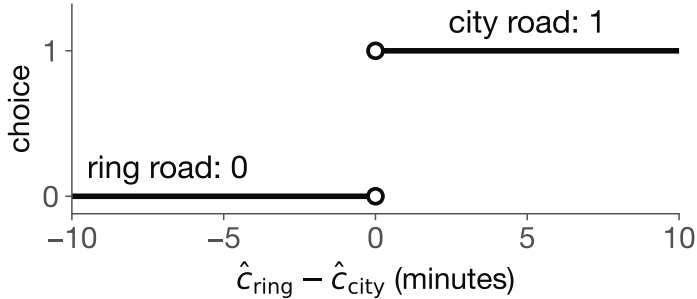


By the chain rule, $r'(\hat{c}_1) = -c_1\, dx_1^\star/d\hat{c}_1 - c_2\, dx_2^\star/d\hat{c}_1$.

Between the discontinuities, $dx_1^\star/d\hat{c}_1 = dx_2^\star/d\hat{c}_1 = 0$.

So $r' = 0$, and SGD cannot train the prediction.

The rider's choice

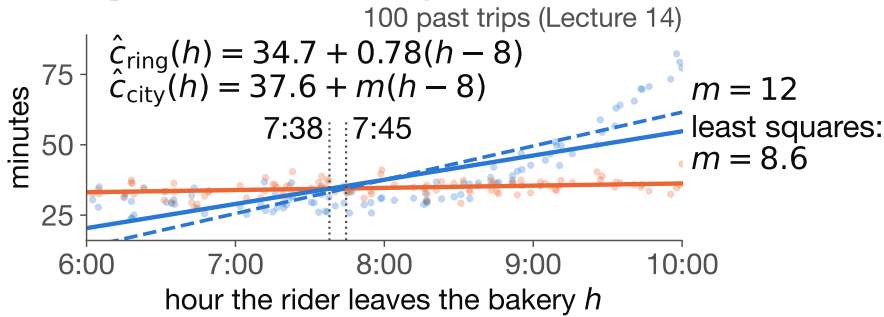


The rider takes the city road if $\hat{c}_{\text{ring}}(h) - \hat{c}_{\text{city}}(h) > 0$.

So the rider's choice is also piecewise constant.

Next, we change the slope of the city road's line.

The slope of the city road's line

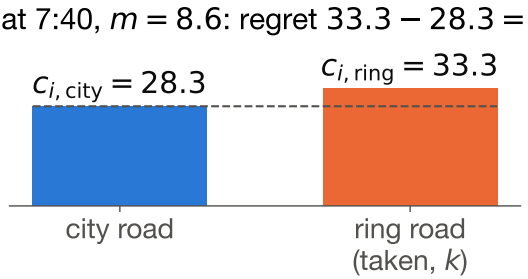


The city road's line is $\hat{c}_{\text{city}}(h) = 37.6 + m(h - 8)$.

The switch hour is the h at which $\hat{c}_{\text{city}}(h) = \hat{c}_{\text{ring}}(h)$.

With $m = 12$, the switch hour moves to 7:45.

The regret of a past trip

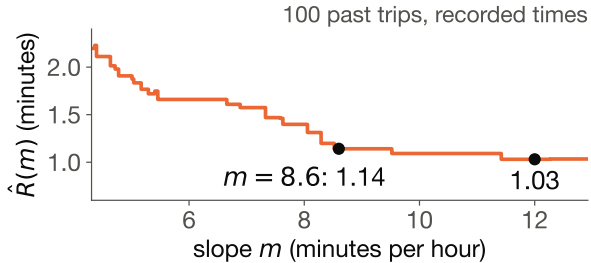


On trip i, the recorded minutes are $c_{i,\text{city}}$ and $c_{i,\text{ring}}$.

We define $\text{regret}_i = c_{i,k} - \min(c_{i,\text{city}}, c_{i,\text{ring}})$,

where k is the road that the rider took.

The average regret on 100 past trips

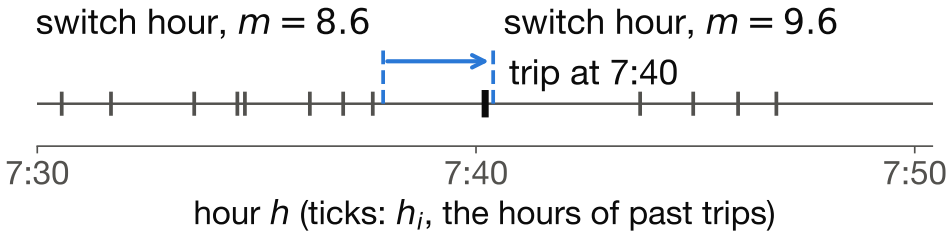


$$\hat{R}(m) = \frac{1}{100} \sum_{i=1}^{100} \text{regret}_i(m)$$

The choice on trip i depends on m.

The average $\hat{R}(m)$ is piecewise constant in m.

The switch hour and the past trips

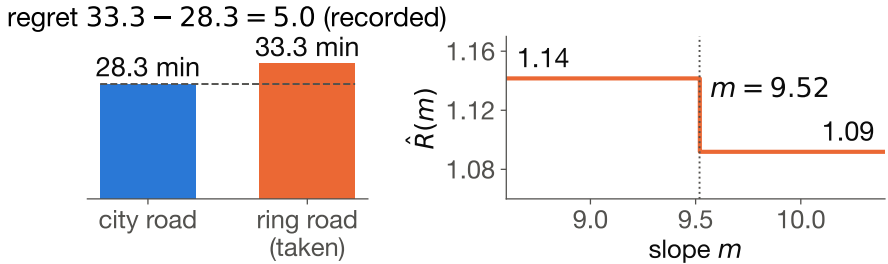


As m increases, the switch hour moves later.

Trip i changes road when the switch hour passes h_i.

If the switch hour passes no trip, $\hat{R}$ stays the same.

The trip at 7:40

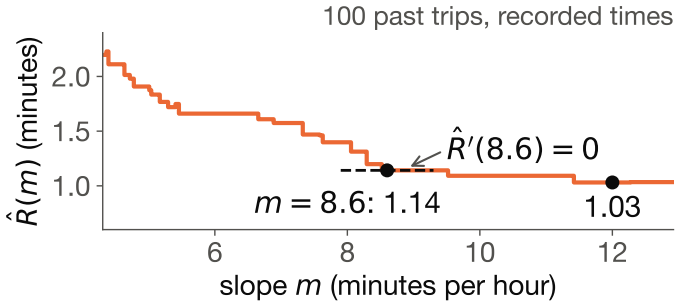


At $m \approx 9.52$, the switch hour passes the trip at 7:40.

The trip's regret decreases from 5.0 to 0.

So $\hat{R}$ decreases by $5.0/100 = 0.05$, from 1.14 to 1.09.

Gradient descent at $m = 8.6$

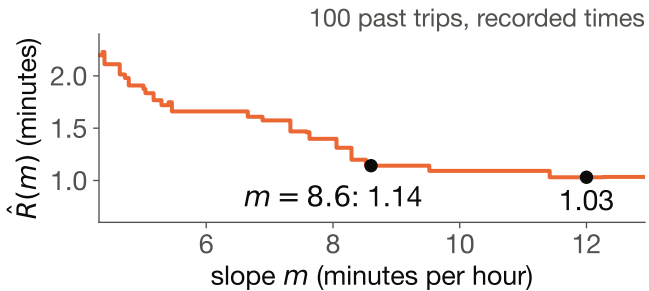


At $m = 8.6$, $\hat{R}'(8.6) = 0$.

So gradient descent stays at $m = 8.6$, with $\hat{R} = 1.14$.

Gradient descent never reaches $\hat{R} = 1.03$.

Summary: piecewise constant regret



The regret on past trips is piecewise constant.

The derivative $\hat{R}'(m)$ is 0 wherever it exists.

Checkpoint 3

prediction: $\hat{c}_1 = 50$

true value: $c_1 = 90$

$r(50) = 400$

What is $r'(50)$ when $c_1 = 90$?

Checkpoint 3

prediction: $\hat{c}_1 = 50$

true value: $c_1 = 90$

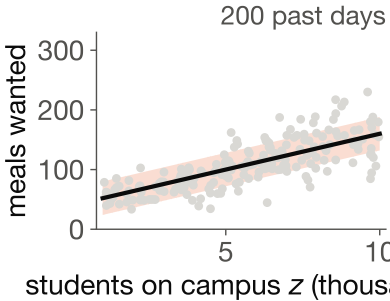
$r(50) = 400$

What is $r'(50)$ when $c_1 = 90$?

Every $\hat{c}_1$ between 20 and 80 gives the plan $(80, 80)$.

So $r(\hat{c}_1) = 400$ near $\hat{c}_1 = 50$, and $r'(50) = 0$.

A normal model for the canteen

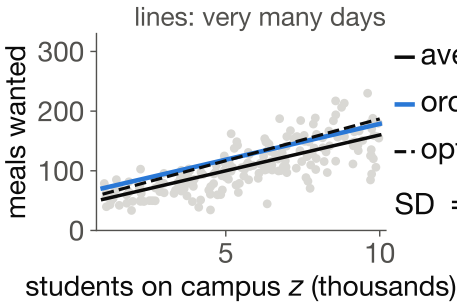


The model assumes that the demand is normal.

The average is $a_0 + a_1 z$, and the SD is constant.

The 0.75 quantile is $x(z) = a_0 + a_1 z + 0.67 \times$ SD.

Fitting the model by least squares

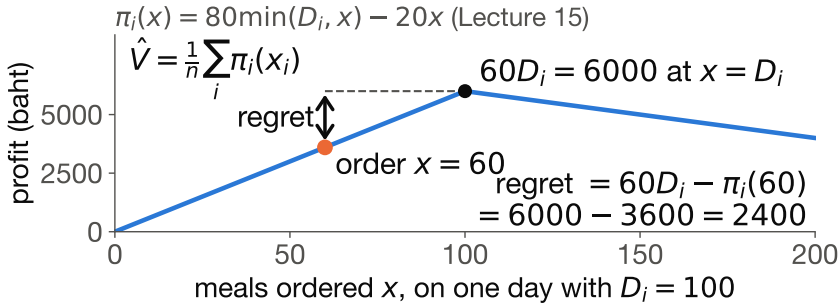


With very many days, least squares gives $40 + 12z$.

The SD of the errors is about 28, so $x(z) \approx 59 + 12z$.

This order is Lecture 14's predict, then optimize.

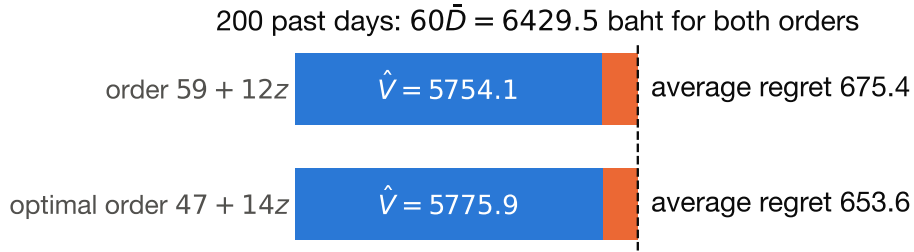
The regret of one day



In hindsight, the optimal order D_i earns $60D_i$.

So the regret of the order x_i is $60D_i - \pi_i(x_i)$.

The average regret

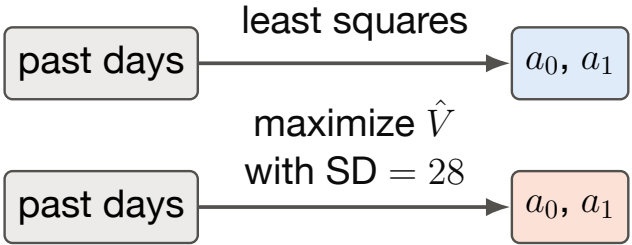


The average regret is $60\bar{D} - \hat{V}$, where $\bar{D} = \frac{1}{n}\sum_i D_i$.

The orders do not change $\bar{D}$.

So maximizing $\hat{V}$ minimizes the average regret.

Training the model for regret



$$\hat{V} = \frac{1}{n} \sum_{i=1}^{n} \pi_i(x_i)$$

Least squares minimizes the sum of squared errors.

Training for regret uses the same SD $= 28$

and chooses a_0 and a_1 to maximize $\hat{V}$.

A linear policy, as in Lecture 15

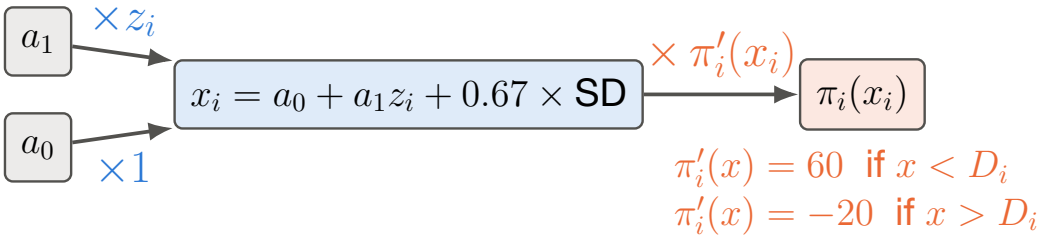
$$x(z) = \underbrace{(a_0 + 0.67 \times \text{SD})}_{b_0} + \underbrace{a_1}_{b_1} z$$

The order is a linear policy $b_0 + b_1 z$,

with $b_0 = a_0 + 0.67 \times$ SD and $b_1 = a_1$.

So maximizing $\hat{V}$ over a_0, a_1 is Lecture 15's method.

Partial derivatives for the canteen

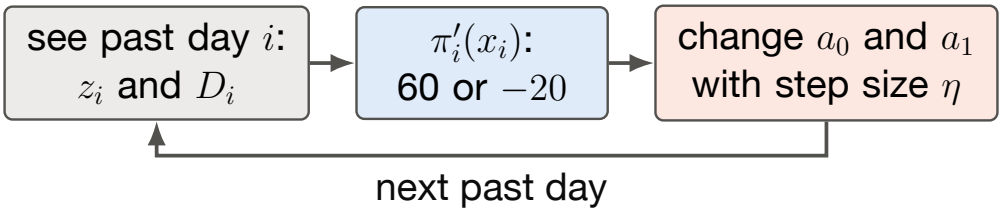


By the chain rule, $\partial \pi_i / \partial a_0 = \pi_i'(x_i) \times 1$

and $\partial \pi_i / \partial a_1 = \pi_i'(x_i) \times z_i$.

The factor $\partial x_i / \partial a_1 = z_i$ is not 0.

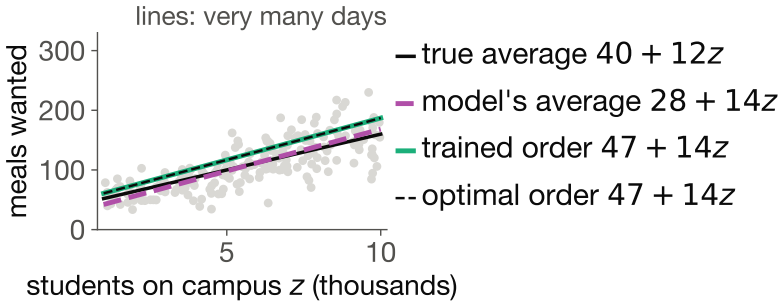
SGD for a_0 and a_1



$$a_0 \leftarrow a_0 + \eta \times \pi'_i(x_i)$$
$$a_1 \leftarrow a_1 + \eta \times \pi'_i(x_i) \times z_i$$

We add the derivative because we maximize profit.

The model trained for regret

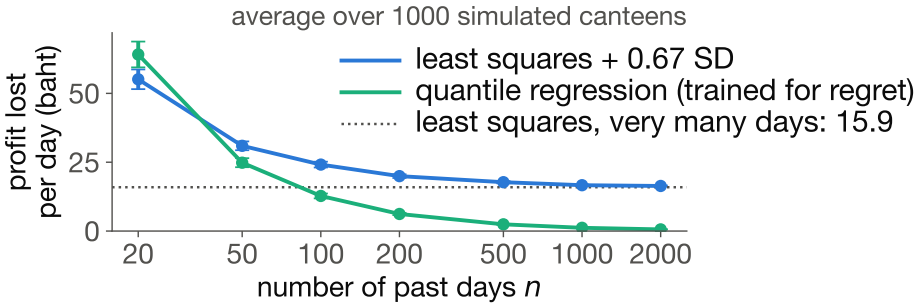


The trained model gives the optimal order $47 + 14z$.

The line $28 + 14z$ has larger errors than $40 + 12z$.

A line with one SD cannot describe this demand.

Profit lost in Lecture 14's canteen

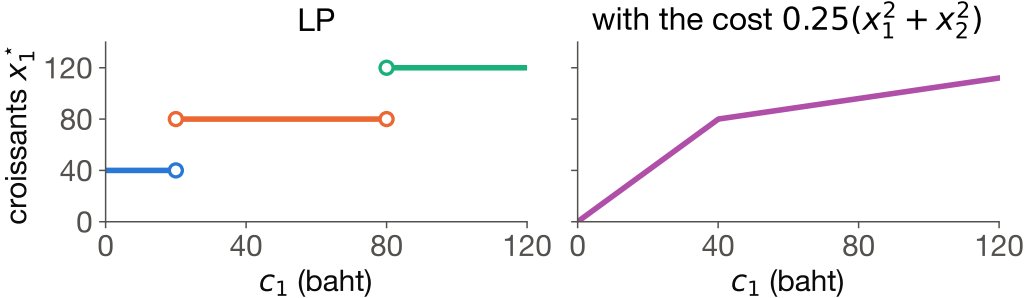


For large n, quantile regression also gives $47 + 14z$.

For $n \geq 50$, quantile regression loses less profit.

At $n = 2000$, least squares loses 16.4 baht per day.

A solution map without discontinuities



Now the profit is $c_1 x_1 + 40 x_2 - 0.25(x_1^2 + x_2^2)$.

So $dx_1^\star / dc_1 = 2$ if $0 < c_1 < 40$ and 0.4 if $40 < c_1 < 120$.

This derivative is not 0, so SGD can train $\hat{c}_1$.

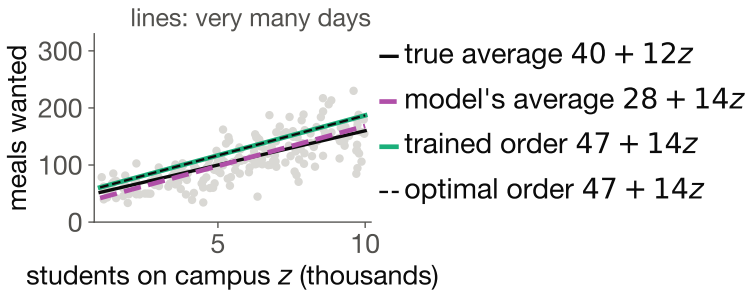
Power grids (Donti et al., 2017)



Photo: Foto3821, CC0, Wikimedia Commons

A neural network predicts the electricity demand.
Least squares gave the smaller prediction errors,
but training for regret gave a 38.6% lower cost.

Summary: training for regret



SGD can train for regret, since $\partial x_i / \partial a_1 = z_i$ is not 0.

Worse predictions can give better orders.

Checkpoint 4

$a_0 = 40,\ a_1 = 12$
SD $= 28,\ \eta = 0.0005$
day i: $z_i = 5,\ D_i = 80$

SGD uses past day i to change a_0 and a_1.

What are the new a_0 and a_1?

Checkpoint 4

$a_0 = 40,\ a_1 = 12$
SD $= 28,\ \eta = 0.0005$
day i: $z_i = 5,\ D_i = 80$

$$x_i = 40 + 12 \times 5 + 0.67 \times 28 \approx 118.8$$
$$\eta \times (-20) = -0.01$$
$$a_0 \leftarrow 40 - 0.01,\ a_1 \leftarrow 12 - 0.01 \times 5$$

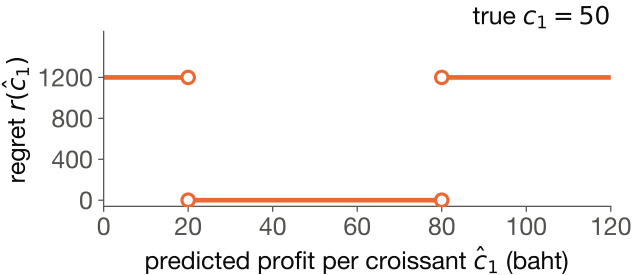
SGD uses past day i to change a_0 and a_1.

What are the new a_0 and a_1?

Since $x_i = 118.8 > 80$, $\pi_i'(x_i) = -20$.

So a_0 becomes 39.99, and a_1 becomes 11.95.

Lecture summary

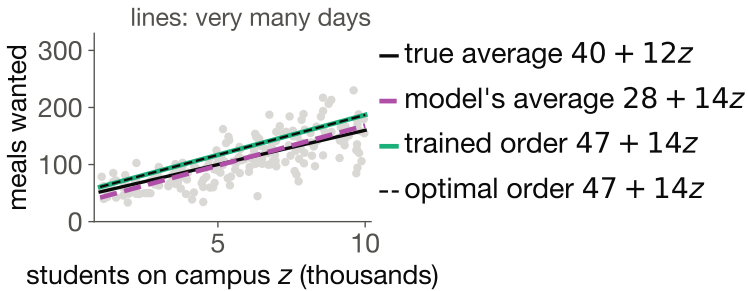


A plug-in policy predicts, then optimizes.

For an LP, the regret is piecewise constant in $\hat{c}$.

Where the derivative of the regret exists, it is 0.

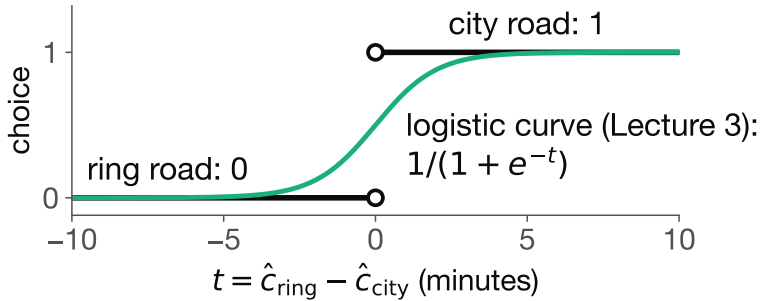
Lecture summary (continued)



For the canteen, $\partial x_i / \partial a_1 = z_i$ is not 0.

So SGD can train the model for regret.

Next lecture



Lecture 18 uses a logistic curve for the choice.

The logistic curve has a derivative that is not 0.

Exercises

1 the bakery: $\hat{c}_1 = 10$, true $c_1 = 100$, $c_2 = 40$

2
$$\hat{c}_{\text{city}}(h) = 38 + m(h - 8)$$
$$\hat{c}_{\text{ring}}(h) = 35 + (h - 8)$$

3 $a_0 = 40$, $a_1 = 12$, SD $= 28$, $\eta = 0.0005$
past day i: $z_i = 8$, $D_i = 160$

1. Find the regret of the plan for $\hat{c}_1 = 10$.
2. Find the switch hour for $m = 7$.
3. Find a_0 and a_1 after one SGD step.