Why naive maximization fails

208424 Lecture 18: the winner's curse and overfitting

Lecture 17: we kept the largest estimate

$$\theta \leftarrow \theta + \eta \nabla \hat{V}_{\mathrm{IPS}}(\pi_\theta)$$

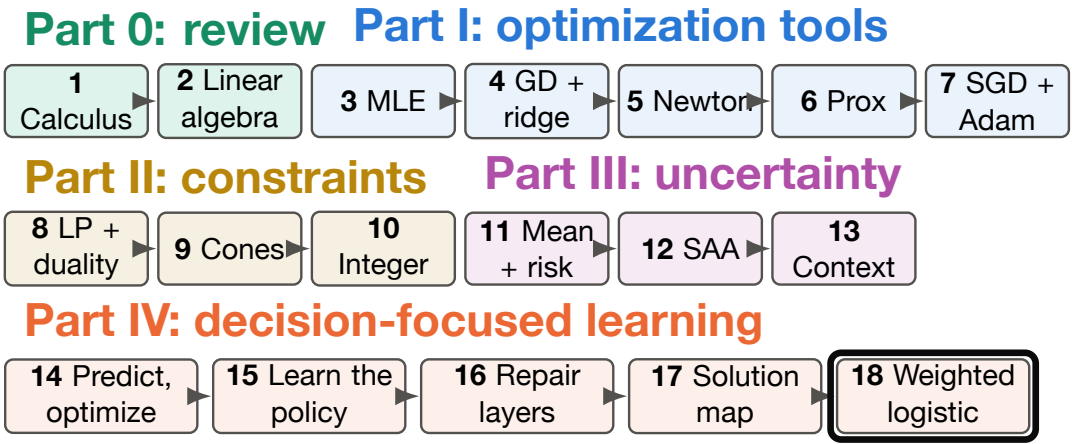
maximize the IPS estimate; keep the maximizing policy

click app, our log: estimate 0.1037, true 0.1039

café, one log: estimate 0.1333 $>$ optimal 0.1311

today: why the maximum is too high; two fixes

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 12: biased past-day profit

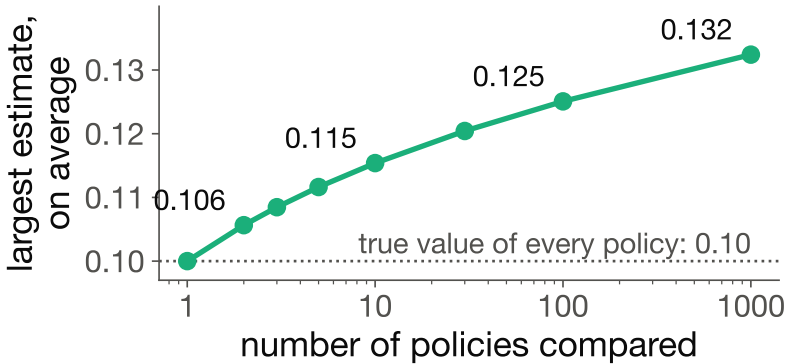
order chosen on 10 past days; baht per day,
averages over many simulated cafés

its profit on those past days	5,310
optimal profit	5,237
its true profit	5,173

it was chosen because it looked best: partly noise

Lecture 12's rule: evaluate it on new days

The largest of many IPS estimates



every policy worth 0.10, SE 0.01: largest too high

SE: the typical random error of an estimate; estimates independent

Why the chosen estimate is too high

$\pi^\star$: the best of the policies we compare, fixed in advance

$\hat{\pi}$: the policy with the largest estimate on our log

on every log: $\hat{V}_{\mathrm{IPS}}(\hat{\pi}) \geq \hat{V}_{\mathrm{IPS}}(\pi^\star)$

over many logs: $\hat{V}_{\mathrm{IPS}}(\pi^\star)$ averages $V(\pi^\star)$

$$\mathbb{E}\left[\hat{V}_{\mathrm{IPS}}(\hat{\pi})\right] \geq V(\pi^\star) \geq \mathbb{E}\left[V(\hat{\pi})\right]$$

Lecture 12: 5,310 $\geq$ 5,237 $\geq$ 5,173

the winner's curse: the winner's estimate is too high,

though each estimate alone is unbiased (Lecture 15)

Lecture 17 with 3% exploration

Lecture 17's click app; the learned policy over 2000 logs

exploration	its estimate minus its true value	logs where it is worse than the past policy
30%	$+0.0001$	0%
3%	$+0.0115$	11.7%

with 3%, an item rarely shown has about 30 rows:
its entry is noisy, and the largest entry is often mostly noise

Summary: the winner's curse

- Let $\hat{V}_{\text{IPS}}(\pi)$: the estimate on one log; $V(\pi)$: the true value; π: policies we compare.
- We want: $\max_\pi V(\pi)$; we take $\hat{\pi}$, the largest $\hat{V}_{\text{IPS}}(\pi)$.
- $\mathbb{E}[\hat{V}_{\text{IPS}}(\hat{\pi})] \geq \max_\pi V(\pi) \geq \mathbb{E}[V(\hat{\pi})]$.
- The gap grows with more policies and higher-variance estimates.
- Worst with many policies and few rows for each.

Checkpoint 1

five policies, all worth 0.10

independent estimates, SE 0.02

the largest of 5 independent $N(0,1)$ numbers averages 1.163

(a) the largest estimate, on average?
(b) the true value of the chosen policy?

Checkpoint 1

five policies, all worth 0.10

independent estimates, SE 0.02

the largest of 5 independent $N(0,1)$ numbers averages 1.163

(a) the largest estimate, on average?

(b) the true value of the chosen policy?

(a) $0.10 + 0.02 \times 1.163 = 0.123$

(b) 0.10: every policy is worth 0.10

A smaller café log

Lecture 17's café, now only 2000 visits

past policy 0.1164; optimal 0.1311

a table of k hour bins: $3k$ parameters

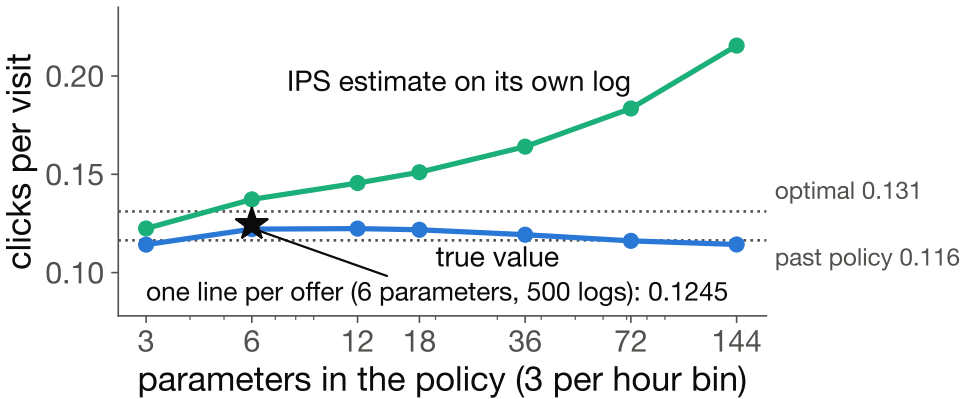
$k = 1$: one offer all day; $k = 48$: 15-minute bins

learned: largest clicks ÷ propensity (Lecture 15) per bin

k bins: 3^k policies; 48 bins: 8×10^{22}

we repeat on 2000 simulated logs

More parameters, higher estimate



2000 logs; 144 parameters: estimate 0.2155, true 0.1143

One bin where B looked best by chance

our café log, 48 bins: 9:45 to 10:00, its most misleading bin

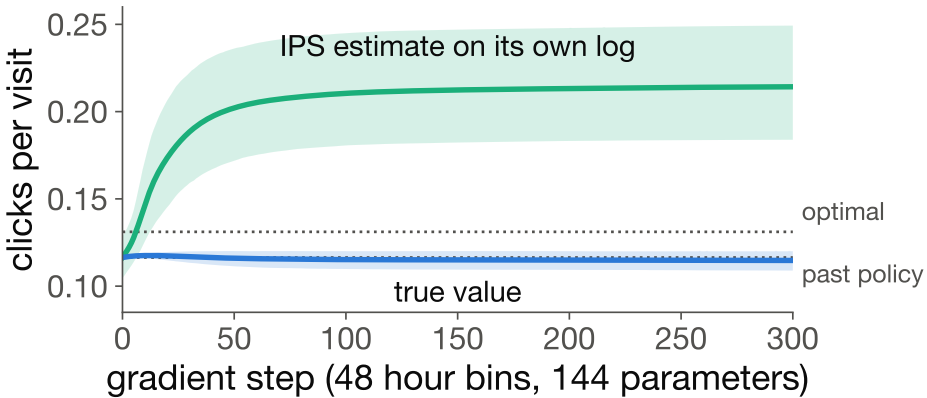
(μ: propensity)

offer	shown	clicks	μ	clicks $\div$ μ	true click probability
A breakfast	26	2	0.8	2.5	0.1375
B coffee	8	3	0.1	**30**	0.1200
C snack	3	0	0.1	0	0.0385

learned: B, from only 3 clicks in 8 tries

on this log, 20 of the 48 bins learned a worse offer

Estimate and true value at each step



2000 logs: true value 0.1175 at step 12, 0.1147 at 300

Two signs that the estimate is too high

48 bins, 2000 logs

1. estimate 0.2155: above every click probability

   click probabilities are at most 0.16: partly noise

   real data: compare with the largest possible outcome

2. average weight 1.10 (Lecture 15: about 1)

   the policy picked offers that got extra rows by chance

Lecture 17's baseline, again

IPS can rise by raising the average weight:

"propensity overfitting" (Swaminathan and Joachims, 2015)

a baseline b subtracts $b \times (\text{average weight} - 1)$

SNIPS divides by the average weight

48 bins: estimate $\div$ average weight $= 0.1955,\ > 0.16$

2000 logs: SNIPS-learned 0.1117, IPS-learned 0.1143

mostly noise in the clicks

Summary: overfitting with many bins

- Let k: the number of hour bins, $3k$ parameters; learned: largest entry per bin.
- We want: a large true value; we get the largest estimate.
- Overfitting: the estimate rises to 0.2155; the true value 0.1224 at 4 bins, 0.1143 at 48.
- Warning signs: estimate above every click probability; average weight above 1.
- Worst with many parameters and few rows for each.

Checkpoint 2

one bin: A shown 40 times (propensity 0.8), 4 clicks

B shown 5 times (propensity 0.1), 1 click

(a) clicks ÷ propensity? learned offer?

(b) true click probabilities 0.14 and 0.12: which offer is better?

Checkpoint 2

one bin: A shown 40 times (propensity 0.8), 4 clicks

B shown 5 times (propensity 0.1), 1 click

(a) clicks $\div$ propensity? learned offer?

(b) true click probabilities 0.14 and 0.12: which offer is better?

(a) A: $4/0.8 = 5$; B: $1/0.1 = 10$: B

(b) A: B's entry comes from 1 click in 5 tries

Evaluate on held-out rows

half A for training, half B held out (1000 visits each)

learn on A, estimate on B

B is held out: the policy was fixed before B was used,
so IPS on B is unbiased (Lecture 15)

Lecture 12: new days; Lecture 4: validation

2000 logs: on B, estimate minus truth 0.0007 or less, every k

Choose the number of bins

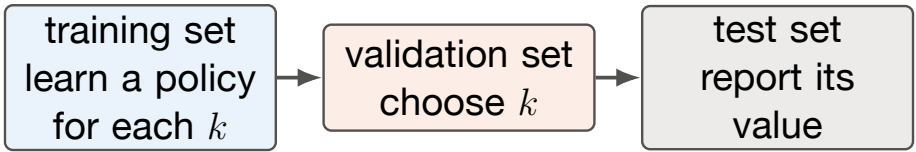
k chosen by the estimate on	k	true	reported
its own log	always 48	0.1143	0.2155
half B	1 to 48	0.1199	0.1404

2000 logs; in row 2 the policies learn on half A only

48 bins can copy any coarser table: never lower

half B picks a better policy, but B's number is too high: $+0.0205$ (slide 6: the largest of 7)

Training, validation and test sets

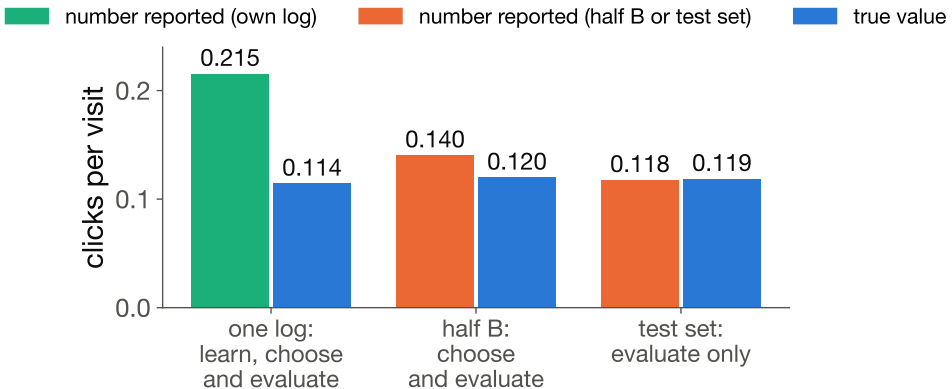


2000 logs, thirds of about 667 visits:
reported 0.1177, true 0.1186: unbiased

one log's test set is small: give its SE too

Lecture 4's validation, plus a part kept for the end

Which number to report?



2000 logs: only the test-set number is unbiased

Summary: evaluate on a test set

- Split the log into training, validation and test sets: learn, choose, evaluate.
- We want: a number for the kept policy, unbiased.
- IPS on the test set is unbiased; give its SE too.
- Choosing on half B makes B's number too high: $+0.0205$ here.
- Costs rows: each part is smaller than the log.

Checkpoint 3

policies for 4 bins and for 48 bins, learned on the training set

their estimates on the validation set: 0.121 and 0.126

(a) which policy do we keep?

(b) is 0.126 an unbiased estimate for it?

(c) where do we get one?

Checkpoint 3

policies for 4 bins and for 48 bins, learned on the training set

their estimates on the validation set: 0.121 and 0.126

(a) which policy do we keep?

(b) is 0.126 an unbiased estimate for it?

(c) where do we get one?

(a) 48 bins: its part-2 estimate is larger

(b) no: chosen for being largest, so likely too high

(c) its estimate on the test set

The variance penalty

$$\text{maximize} \;\; \hat{V}_{\text{IPS}}(\pi) - \lambda \, \text{SE}(\pi)$$

SE: Lecture 15's standard error, the square root of a variance

$\lambda = 1.96$: the lower confidence bound (95%)

pessimism: choose the largest lower bound

a policy learned from a few rows has a large SE

λ is a penalty weight, like Lecture 4's: here fixed at 1.96

Our café log, 4 bins

one log; bins 8–11, 11–14, 14–17, 17–20

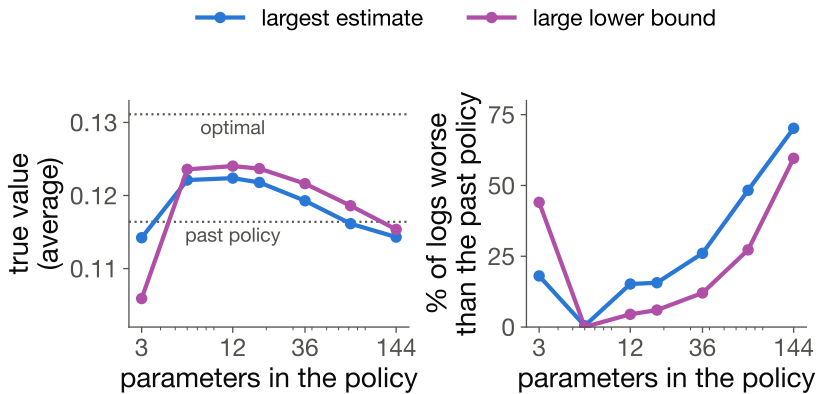
keep the largest	offers	estimate	SE	lower bound	true
estimate	B, A, C, B	0.1462	0.0213	0.1045	0.1130
lower bound	A, A, C, C	0.1325	0.0086	0.1156	0.1240

lower bound $=$ estimate $- 1.96 \times \mathrm{SE}$; search: one bin at a time

coffee in the morning comes from a few rows: a large SE

this log: $+0.011$; 2000 logs: $+0.0016$ on average

Pessimism over 2000 logs



2000 logs; from 6 parameters: slightly better, fewer bad logs

When pessimism fails

slide 28's mechanism: avoid rarely tried offers

1. a good offer, rarely tried

one offer all day: coffee is best, tried 10%

2000 logs: pessimism skips it in 44% (plain: 18%)

2. many parameters

the largest lower bound is too high, too

below the truth: 90% of logs at 4 bins, 0% at 48

one fixed policy: about 97.5%; our log, 48 bins: $0.172 > 0.16$

Summary: the variance penalty

- Let $\mathrm{SE}(\pi)$: Lecture 15's standard error of $\hat{V}_{\mathrm{IPS}}(\pi)$; $\lambda \geq 0$.
- We want: a policy that does well, not one that only looks good on this log.
- Maximize $\hat{V}_{\mathrm{IPS}}(\pi) - \lambda \mathrm{SE}(\pi)$; $\lambda = 1.96$: 95% lower bound.
- 6 to 72 parameters: $+0.0015$ to $+0.0025$ (past policy to optimum: 0.0147); fewer bad policies.
- Fails: rare good items skipped; many parameters: still too high.

Checkpoint 4

policy 1: estimate 0.130, SE 0.020

policy 2: estimate 0.120, SE 0.004

(a) lower bounds?

(b) which policy does pessimism keep? plain IPS?

Checkpoint 4

policy 1: estimate 0.130, SE 0.020

policy 2: estimate 0.120, SE 0.004

(a) lower bounds?

(b) which policy does pessimism keep? plain IPS?

(a) $0.130 - 0.039 = 0.091$; $0.120 - 0.008 = 0.112$

(b) pessimism: policy 2; plain IPS: policy 1

A policy with fewer parameters

Lecture 17's policy: one line per offer, 6 parameters, 300 steps

visits	true	estimate minus true	worse than past policy
50,000 (Lecture 17)	0.1295	$+0.0022$	0%
10,000 (Lecture 17)	0.1274	$+0.0071$	0%
2000	0.1245	$+0.0197$	1.0%
2000, 4-bin table	0.1224	$+0.0232$	15.2%

200, 200, 500 and 2000 logs

fewer visits: larger error

Test set, penalty, fewer parameters

	helps	costs
test set	unbiased	fewer rows to learn
variance penalty	avoids noisy policies	skips rarely tried items
fewer parameters	less noise to fit	needs a good model

always report a number from a test set

Lecture 17's café log: 0.1333 $>$ best 0.1311

Where these fixes are used

learning from logs

variance penalty, λ on rows kept aside

Swaminathan and Joachims (2015)

ad auctions

maximize a lower bound that skips rarely tried settings

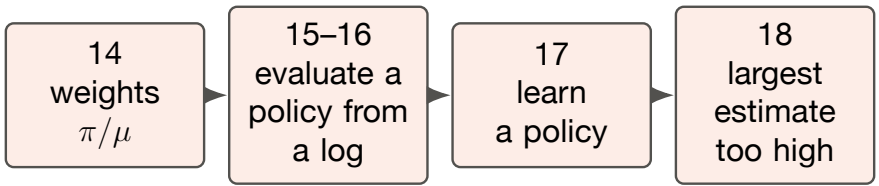
Bottou et al. (2013)

learning from old data only

pessimistic values; simulated robots, games

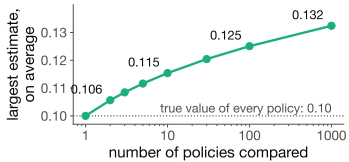
Kumar et al. (2020)

Part IV in one slide



estimates from a log are accurate only for policies near the policy that made the log

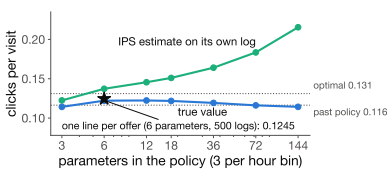
Lecture summary



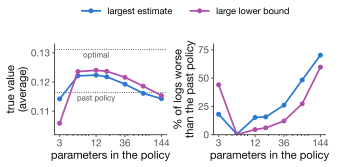
largest estimate: too high



evaluate on a test set



more parameters: overfitting



the variance penalty

Exercise

1. Ten policies worth 0.08, SE 0.01, independent estimates. Largest estimate? (Largest of 10 $N(0,1)$: 1.539.)
2. One bin: A shown 30 times (propensity 0.8), 3 clicks; B 4 times (propensity 0.1), 1 click. Entries? Learned?
3. Policy 1: estimate 0.15, SE 0.03; policy 2: 0.13, SE 0.01. Lower bounds? Which does pessimism keep?
4. We choose k on half B and report B's number. Too high, too low, or unbiased? Why?
5. A learned policy's average weight on its own log is 1.15. What does it suggest?