THROUGH THE LENS OF SKI RENTAL

THREE FLAVORS OF PREDICTIONS IN ONLINE ALGORITHMS

Manish Purohit

Joint work with Ravi Kumar, Zoya Svitkina, and Erik Vee

OUTLINE

- Motivation
- Ski Rental Problem
- The Fortune Cookie
- The Weatherman
- The Constrained Adversary
- Conclusions

TACKLING UNCERTAINTY

Online Algorithms

- Full input is unknown
- Design algorithms for worst-possible future

- Pessimistic
- Cannot exploit patterns / predictability in data.

Machine Learning

- Observe past data
- Build robust models to predict the future

- Highly successful!
- Trained for good average performance
- Not robust to outliers

SKI RENTAL PROBLEM

- Sam recently moved to Colorado
- Renting : $1
- Buying : $B
- Should he rent or should he buy?

- Missing: How often does Sam want to ski?

SKI RENTAL PROBLEM

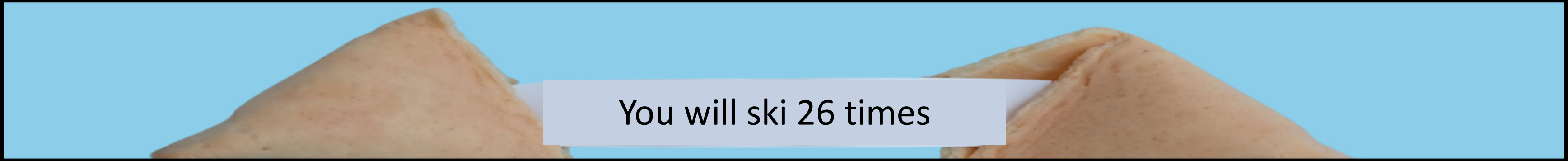
- Sam is very pessimistic and strongly believes "Anything that can go wrong will go wrong"

- Minimizes $\max_x \frac{Alg(x)}{Opt(x)}$

SKI RENTAL PROBLEM

- Minimizes $\max_x \frac{Alg(x)}{Opt(x)}$
- Deterministic Algorithm:
  - Buy on day B-1
  - 2-competitive
- Randomized Algorithm:
  - Sample $i \in \{1, B\}$; $p(i) \propto \left(\frac{b}{b-1}\right)^{i-1}$
  - Buy on day i
  - $\frac{e}{e-1}$-competitive

THE FORTUNE COOKIE



- Notation
  - $y \leftarrow$ predicted number of days
  - $\eta = |x - y| =$ prediction error
- Competitive Ratio
  - Function of the error
  - $\frac{Alg(I)}{Opt(I)} \leq c(\eta(I))$
- Consistency

Algorithm is β-consistent if $c(0) = \beta$

- Robustness

Algorithm is γ-robust if $c(\eta) \leq \gamma$ for all η

ATTEMPT 1

Blind Trust

- If $y \geq b$
  - Buy on day 1
- Else
  - Rent every day

ATTEMPT 1

Blind Trust

- If $y \geq b$
  - Buy on day 1
- Else
  - Rent every day

Analysis

If $(y \geq b$ and $x \geq b)$ or $(y < b$ and $x < b)$

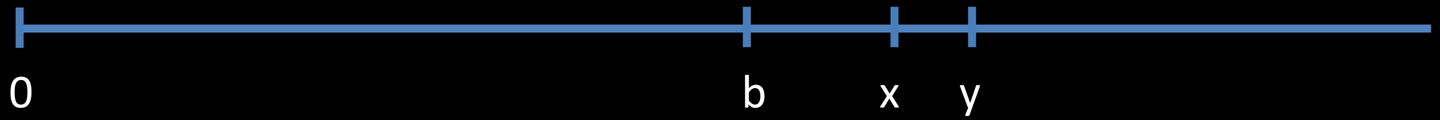
$$ALG = OPT$$

If $(y \geq b$ and $x < b)$

$$ALG = b \leq x + (y - x) = OPT + \eta$$

If $(y < b$ and $x \geq b)$

$$ALG = x \leq b + (x - y) = OPT + \eta$$



ATTEMPT 1

Blind Trust

- If $y \geq b$
  - Buy on day 1
- Else
  - Rent every day

Analysis

If $(y \geq b$ and $x \geq b)$ or $(y < b$ and $x < b)$

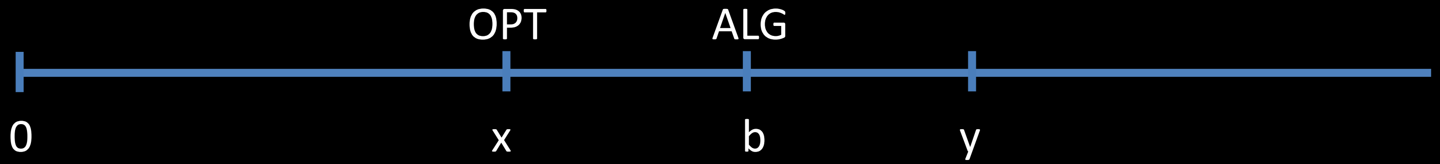
$$ALG = OPT$$

If $(y \geq b$ and $x < b)$

$$ALG = b \leq x + (y - x) = OPT + \eta$$

If $(y < b$ and $x \geq b)$

$$ALG = x \leq b + (x - y) = OPT + \eta$$



ATTEMPT 1

Blind Trust

- If $y \geq b$
  - Buy on day 1
- Else
  - Rent every day

Analysis

If $(y \geq b$ and $x \geq b)$ or $(y < b$ and $x < b)$

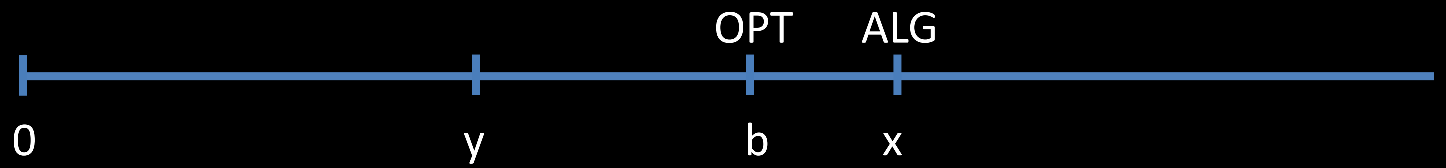
$$ALG = OPT$$

If $(y \geq b$ and $x < b)$

$$ALG = b \leq x + (y - x) = OPT + \eta$$

If $(y < b$ and $x \geq b)$

$$ALG = x \leq b + (x - y) = OPT + \eta$$



ATTEMPT 1

Blind Trust

- If $y \geq b$
  - Buy on day 1
- Else
  - Rent every day

Analysis

If $(y \geq b$ and $x \geq b)$ or $(y < b$ and $x < b)$

$$ALG = OPT$$

If $(y \geq b$ and $x < b)$

$$ALG = b \leq x + (y - x) = OPT + \eta$$

If $(y < b$ and $x \geq b)$

$$ALG = x \leq b + (x - y) = OPT + \eta$$

$$\boxed{ALG \leq OPT + \eta}$$

ATTEMPT 1

Blind Trust

- If $y \geq b$
  - Buy on day 1
- Else
  - Rent every day

1-consistent!

Not Robust

$ALG \leq OPT + \eta$

ATTEMPT 2

Cautious Trust

- Let $\lambda \in (0,1)$ be a hyperparameter

- If $y \geq b$
  - Buy on day $\lceil \lambda b \rceil$
- Else
  - Buy on day $\left\lceil \frac{b}{\lambda} \right\rceil$

Analysis

$$\frac{ALG}{OPT} \leq \min\left\{1 + \lambda + \frac{\eta}{(1-\lambda)\,OPT}, \frac{1+\lambda}{\lambda}\right\}$$

$(1 + \lambda)$-consistent!

😊

$\left(\frac{1+\lambda}{\lambda}\right)$-Robust

😊

ATTEMPT 2

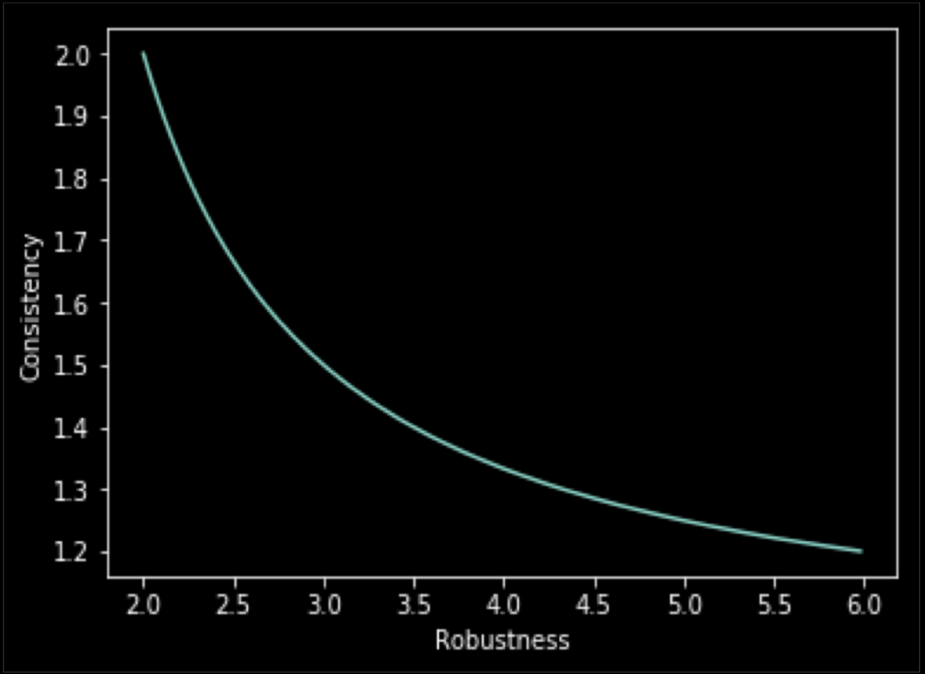
Cautious Trust

- $\lambda \in (0,1)$ gives a tradeoff between consistency and robustness

- Small λ
  - Higher trust in the predictions
  - Better consistency
  - Worse robustness

$(1+\lambda)$-consistent!

$\left(\frac{1+\lambda}{\lambda}\right)$-Robust



ATTEMPT 2

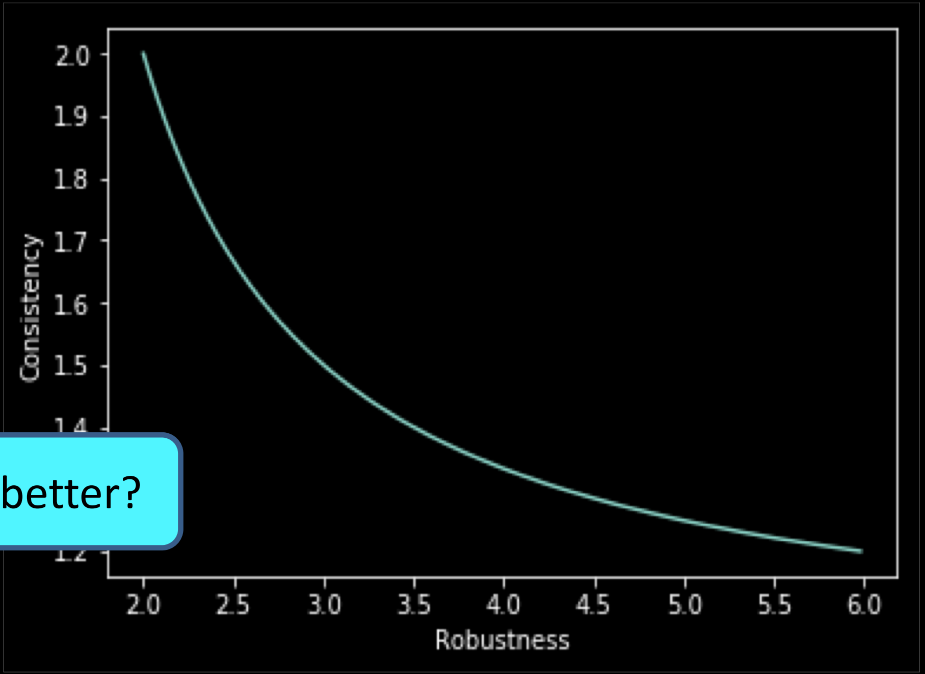
Cautious Trust

- $\lambda \in (0,1)$ gives a tradeoff between consistency and robustness

- Small λ
  - Higher trust in the predictions
  - Better consistency
  - Worse robustness

$(1+\lambda)$-consistent!

$\left(\frac{1+\lambda}{\lambda}\right)$-Robust

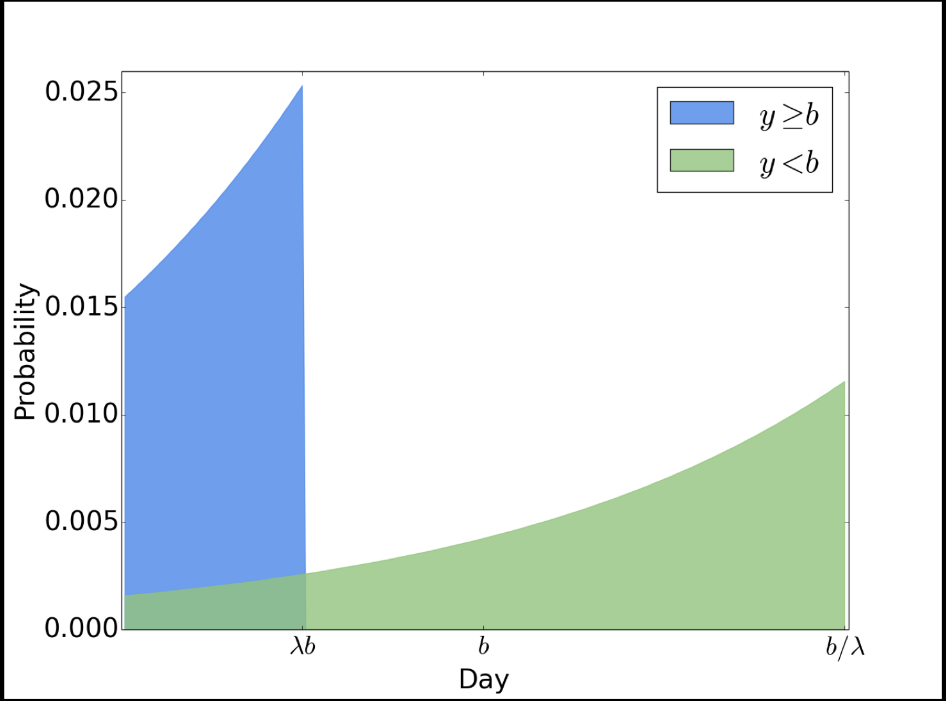


Can we do better?

ATTEMPT 3

- Let's randomize!

- If $y \geq b$
  - $k = \lfloor \lambda b \rfloor$
  - Define $q_i \leftarrow \left(\frac{b-1}{b}\right)^{k-i} \cdot \frac{1}{b(1-(1-1/b)^k)}$
  - Choose $j \in \{1, 2, \dots, k\}$ randomly from distribution defined by q_i.
  - Buy on day j
- Else
  - $\ell = \left\lceil \frac{b}{\lambda} \right\rceil$
  - Define $r_i \leftarrow \left(\frac{b-1}{b}\right)^{\ell-i} \cdot \frac{1}{b(1-(1-1/b)^\ell)}$
  - Choose $j \in \{1, 2, \dots, \ell\}$ randomly from distribution defined by r_i.
  - Buy on day j

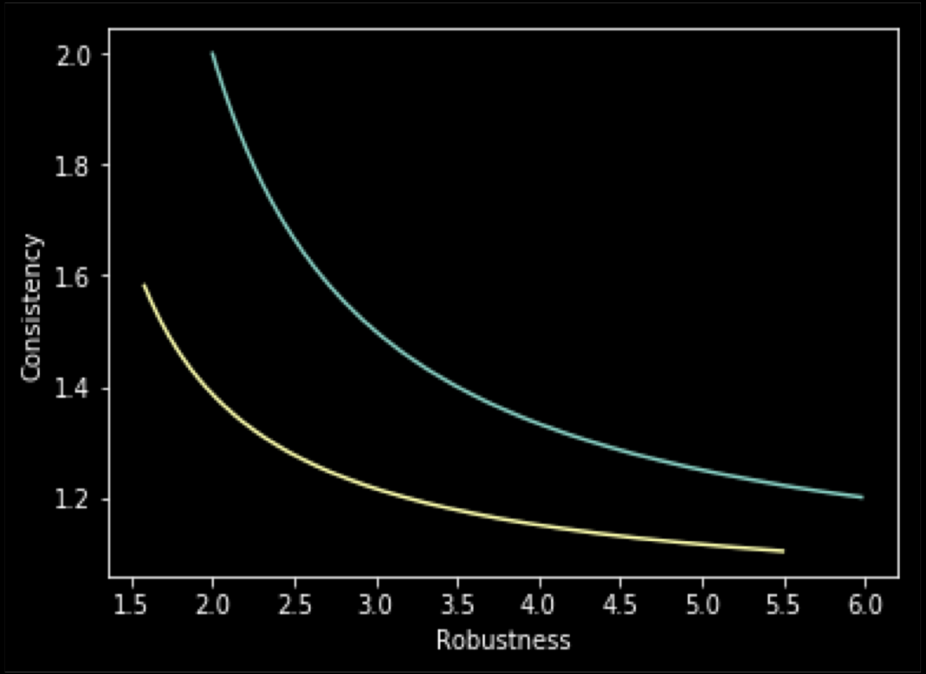


ATTEMPT 3

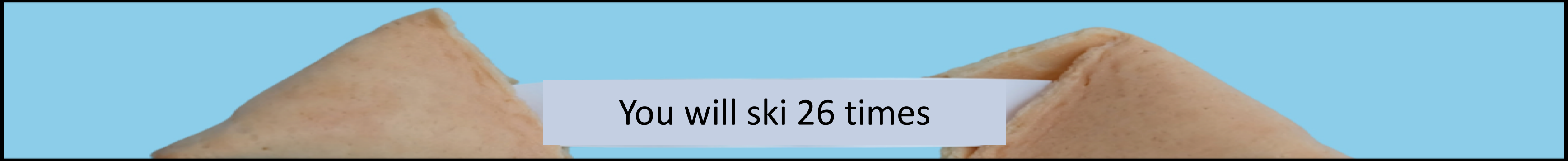
- Let's randomize!
- If $y \geq b$
  - $k = \lfloor \lambda b \rfloor$
  - Define $q_i \leftarrow \left(\frac{b-1}{b}\right)^{k-i} \cdot \frac{1}{b(1-(1-1/b)^k)}$
  - Choose $j \in \{1, 2, \dots, k\}$ randomly from distribution defined by q_i.
  - Buy on day j
- Else
  - $\ell = \left\lceil \frac{b}{\lambda} \right\rceil$
  - Define $r_i \leftarrow \left(\frac{b-1}{b}\right)^{\ell-i} \cdot \frac{1}{b(1-(1-1/b)^\ell)}$
  - Choose $j \in \{1, 2, \dots, \ell\}$ randomly from distribution defined by r_i.
  - Buy on day j

$\left(\frac{\lambda}{1-e^{-\lambda}}\right)$-consistent!

$\left(\frac{1}{1-e^{-\left(\lambda-\frac{1}{b}\right)}}\right)$-Robust



THE FORTUNE COOKIE



Prediction Error

$$\frac{ALG}{OPT} \leq \min\left\{\left(\frac{\lambda}{1-e^{-\lambda}}\right)\left(1+\frac{\eta}{OPT}\right), \frac{1}{1-e^{-\left(\lambda-\frac{1}{b}\right)}}\right\}$$

Consistency

Robustness

OUTLINE

- Motivation
- Ski Rental Problem
- The Fortune Cookie
- The Weatherman
- The Constrained Adversary
- Conclusions

THE WEATHERMAN

- Predictions are backed by a probabilistic guarantee
- The algorithm can utilize these error probabilities to obtain better performance

THE WEATHERMAN FOR SKI RENTAL

- Suppose we train a binary classifier to predict whether Sam will ski for more than b days or not
- $h \leftarrow$ probability of correct prediction
- The algorithm knows h
  (say, by observing performance on validation data)

- What algorithms can we obtain in this setting?

THE WEATHERMAN FOR SKI RENTAL

- If prediction < b days
  - Buy on day b
- If prediction more than b days
  - Buy on day i with probability p_i

Minimize c

subject to

$$\forall d, E[Alg(d)] \leq c \min(b, d)$$

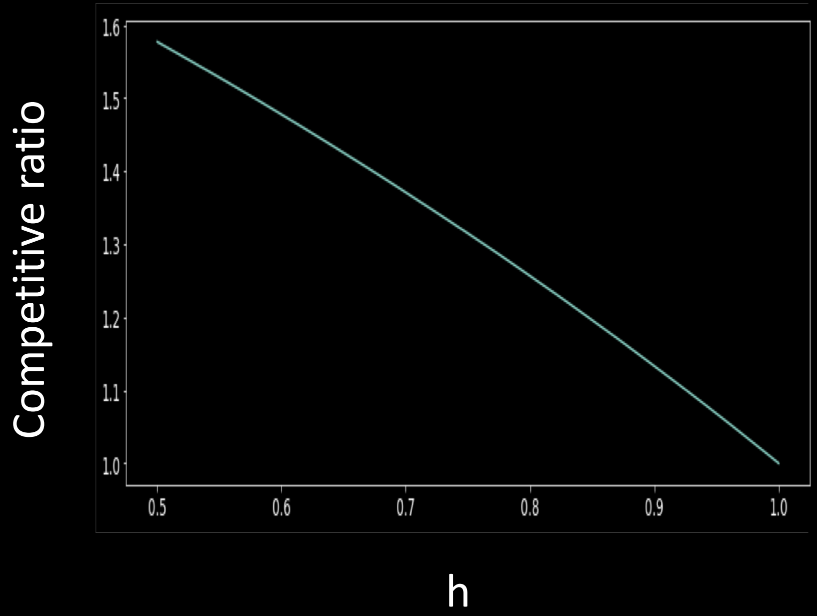
THE WEATHERMAN FOR SKI RENTAL

- If prediction < b days
  - Buy on day b
- If prediction more than b days
  - Buy on day i with probability p_i

Minimize c

subject to

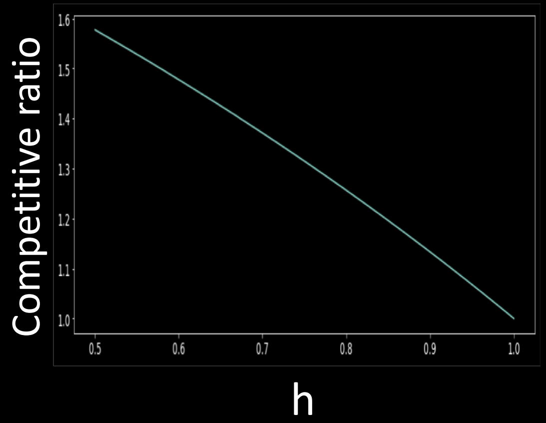
$$\forall d, E[Alg(d)] \leq c \min(b, d)$$



THE WEATHERMAN

competitive ratio= $f(h)$

$f(1) = 1,\ f\left(\frac{1}{2}\right) = \frac{e}{e-1}$



OUTLINE

- Motivation
- Ski Rental Problem
- The Fortune Cookie
- The Weatherman
- The Constrained Adversary
- Conclusions

THE CONSTRAINED ADVERSARY

- Bound the amount of uncertainty
- Make structural assumptions about the online input
- More structure → Better guarantees

THE CONSTRAINED ADVERSARY

- More convenient to work with fractional version of the problem

- Costs 1 to buy skis
- Costs z to rent for z time (fractional)

- Constraint: $x \geq y$
- "Sam knows he'll ski at least five times"

THE CONSTRAINED ADVERSARY

- Let $p_y(z) \leftarrow$ Probability of buying on day z
- Let $q(y) \quad \leftarrow$ Probability of buying on the first day

- Say we enforce $p_y(z) = 0, \forall z > 1$ (Even the deterministic algorithm does that)

- What's the expected algorithm cost for x days?
- $Cost_y(x) = q(y) + \int_0^x (1+z) p_y(z) dz + \int_x^1 x p_y(z) dz$
- Set probabilities so that $\left(\frac{Cost_y(x)}{\min(x,1)}\right)$ is a constant

THE CONSTRAINED ADVERSARY

- Let $p_y(z) \leftarrow$ Probability of buying on day z
- Let $q(y) \leftarrow$ Probability of buying on the first day

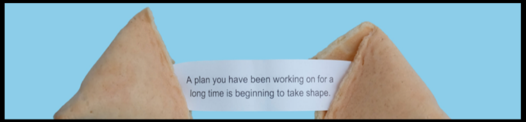
- Say we enforce $p_y(z) = 0, \forall z > 1$ (Even the deterministic algorithm does that)

There exists a randomized algorithm with competitive ratio $\left(\frac{e}{e-(1-y)e^y}\right)$

- Set probabilities so that $\left(\frac{cost_y(x)}{\min(x,1)}\right)$ is a constant

CONCLUSIONS

- The Fortune Cookie
  - Predictions with no error guarantees
  - Competitive ratio = min(consistency, robustness)



- The Weatherman
  - Predicts with error guarantees
  - Competitive ratio = function(error probability)

- The Constrained Adversary (Semi-Online)
  - Structural assumptions about input
  - Improved competitive ratios

THANKS!