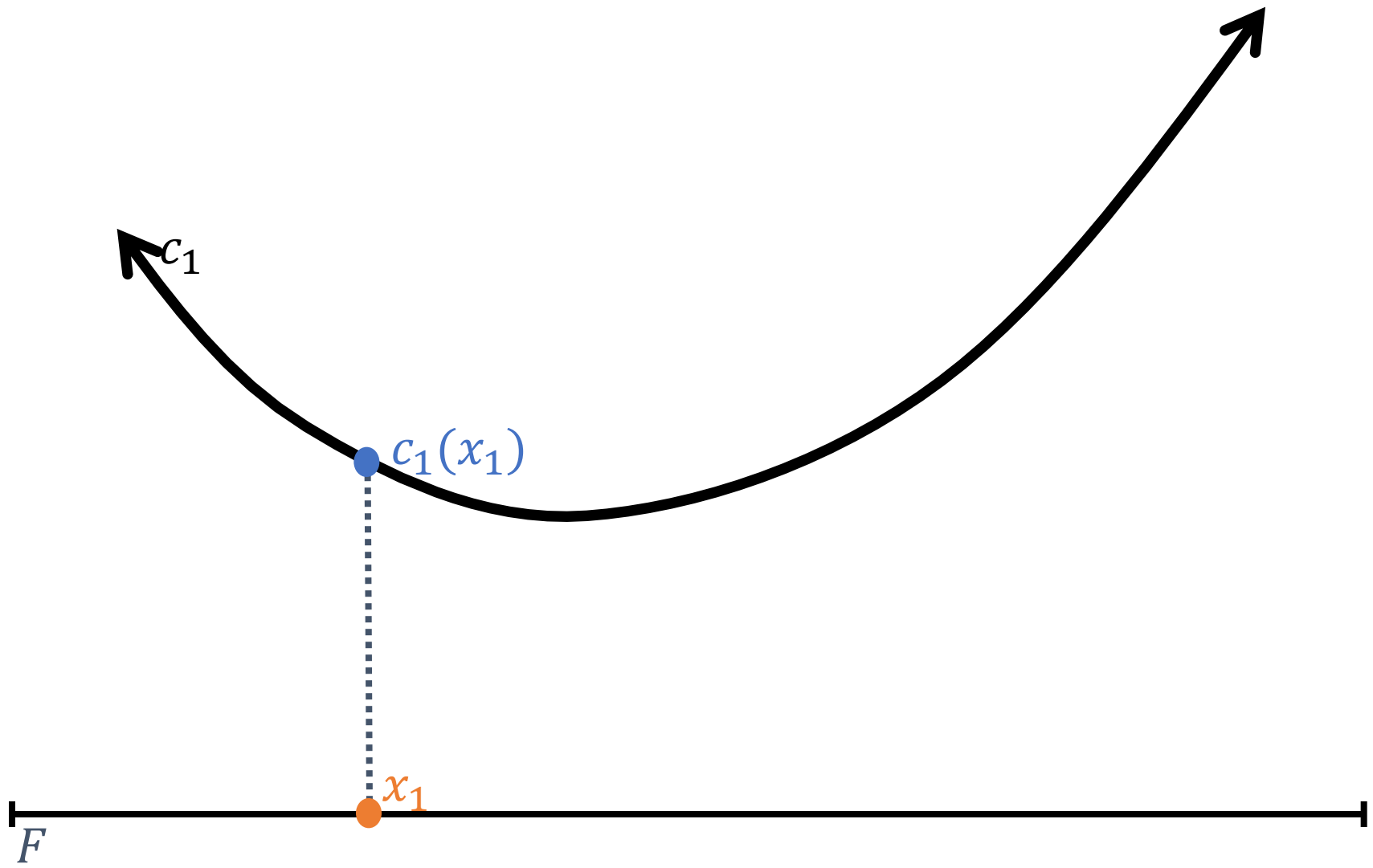


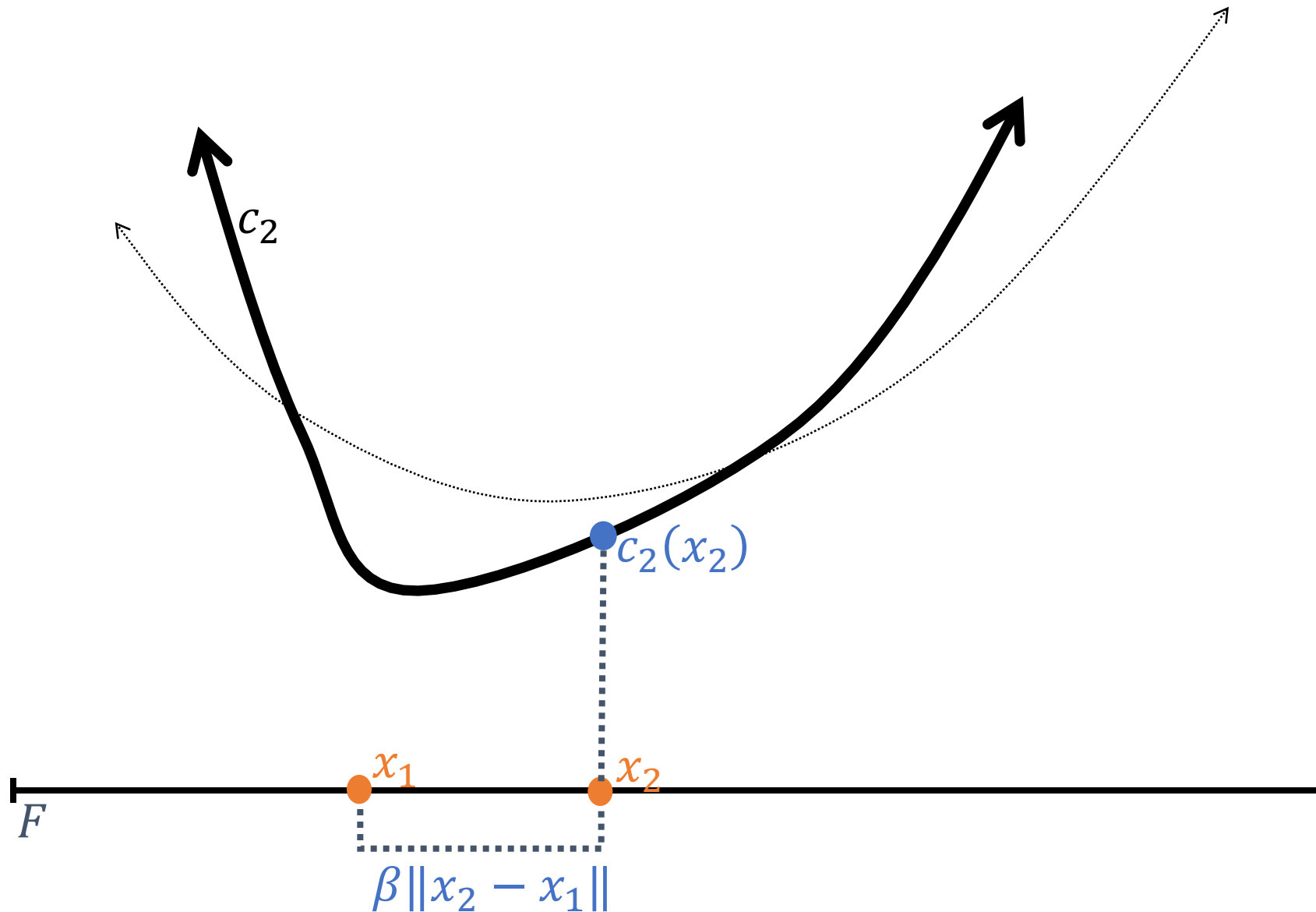
Online Convex Optimization Using Predictions

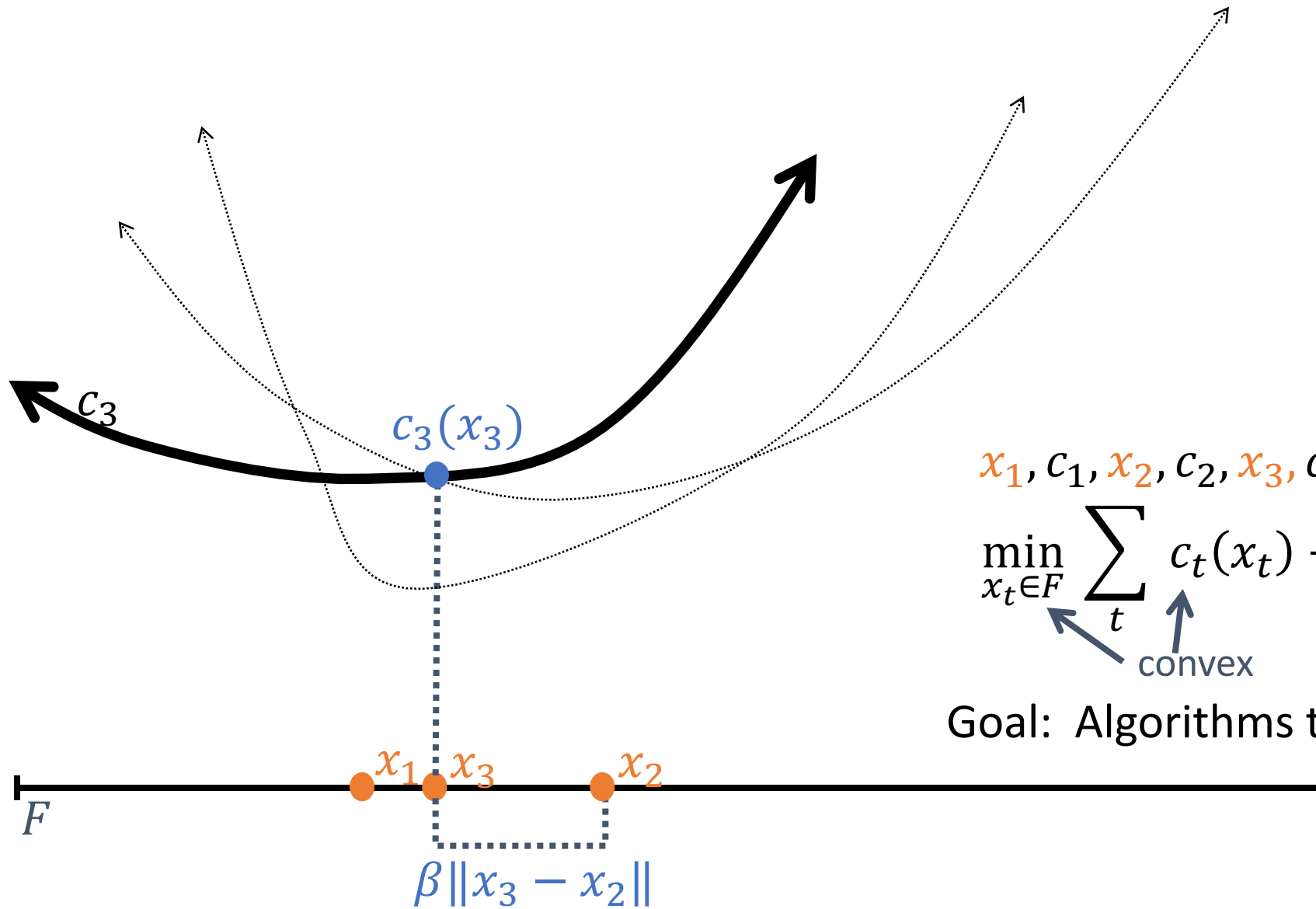
Niangjun Chen

Joint work with Anish Agarwal, Lachlan Andrew, Siddharth Barman, and Adam Wierman









$x_1, c_1, x_2, c_2, x_3, c_3 \dots \leftarrow$ online

$$\min_{x_t \in F} \sum_t c_t(x_t) + \underbrace{\beta \|x_t - x_{t-1}\|}_{\text{switching cost}}$$

convex

Goal: Algorithms to minimize cost

Lots of applications ...

Dynamic capacity management in data centers [Tu et al. 2013]

Power system generation/load scheduling [Lu et al. 2013]

Portfolio management [Cover 1991][Boyd et al. 2012]

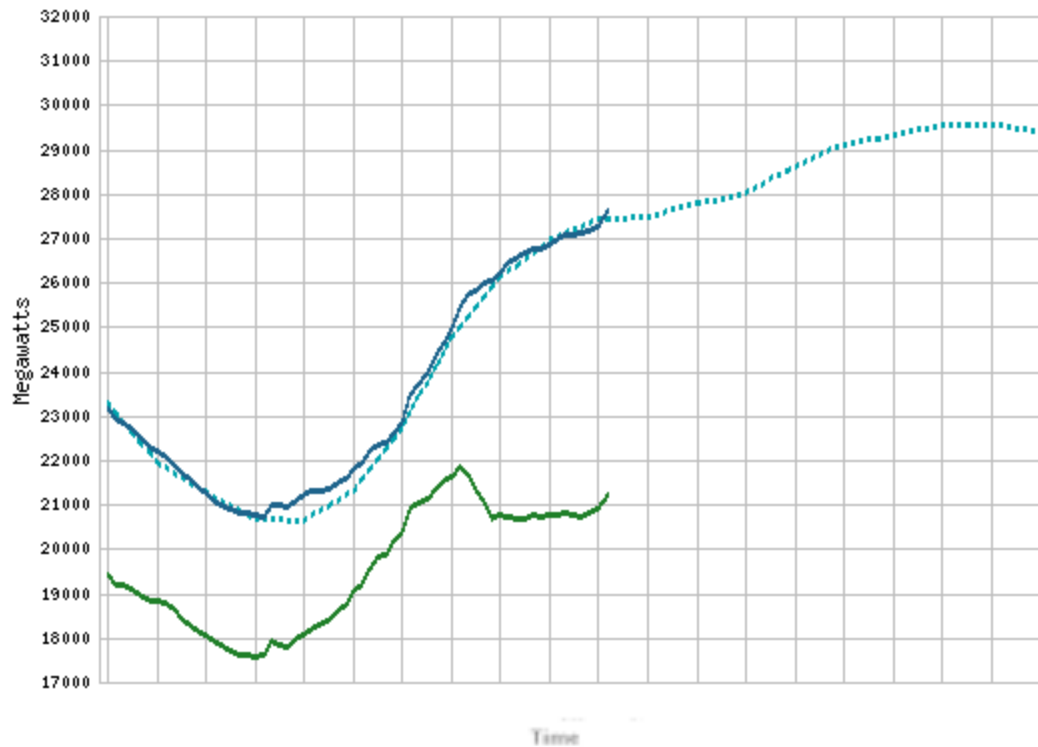
Video streaming [Sen et al. 2000][Liu et al. 2008]

Network routing [Bansal et al. 2003][Kodialam et al. 2003]

Geographical load balancing [Hindman et al. 2011] [Lin et al. 2012]

...

In most applications, predictions are crucial



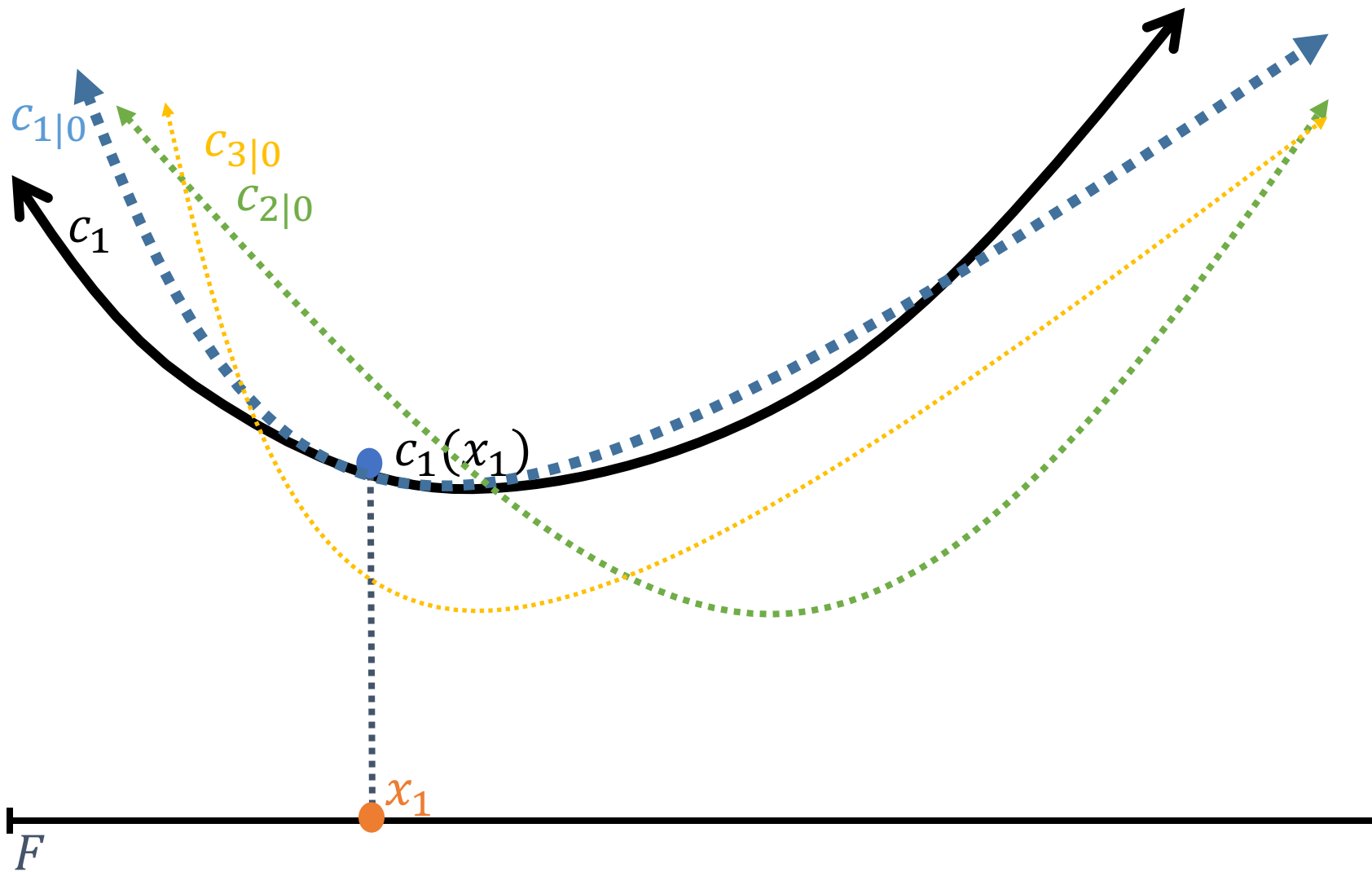
Dow Stock Market Trend Forecast to Jan 2015

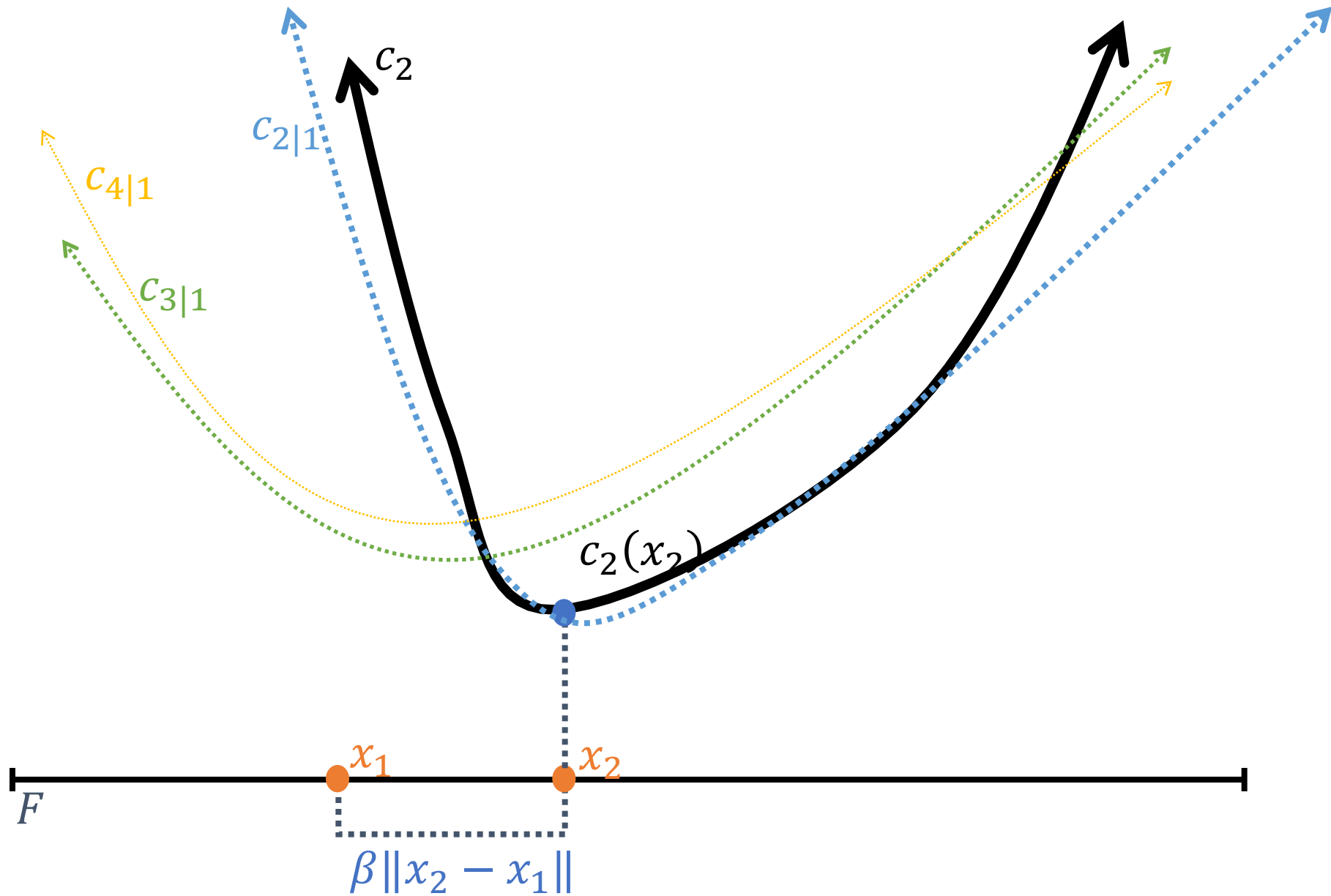
By Nadeem Walayat

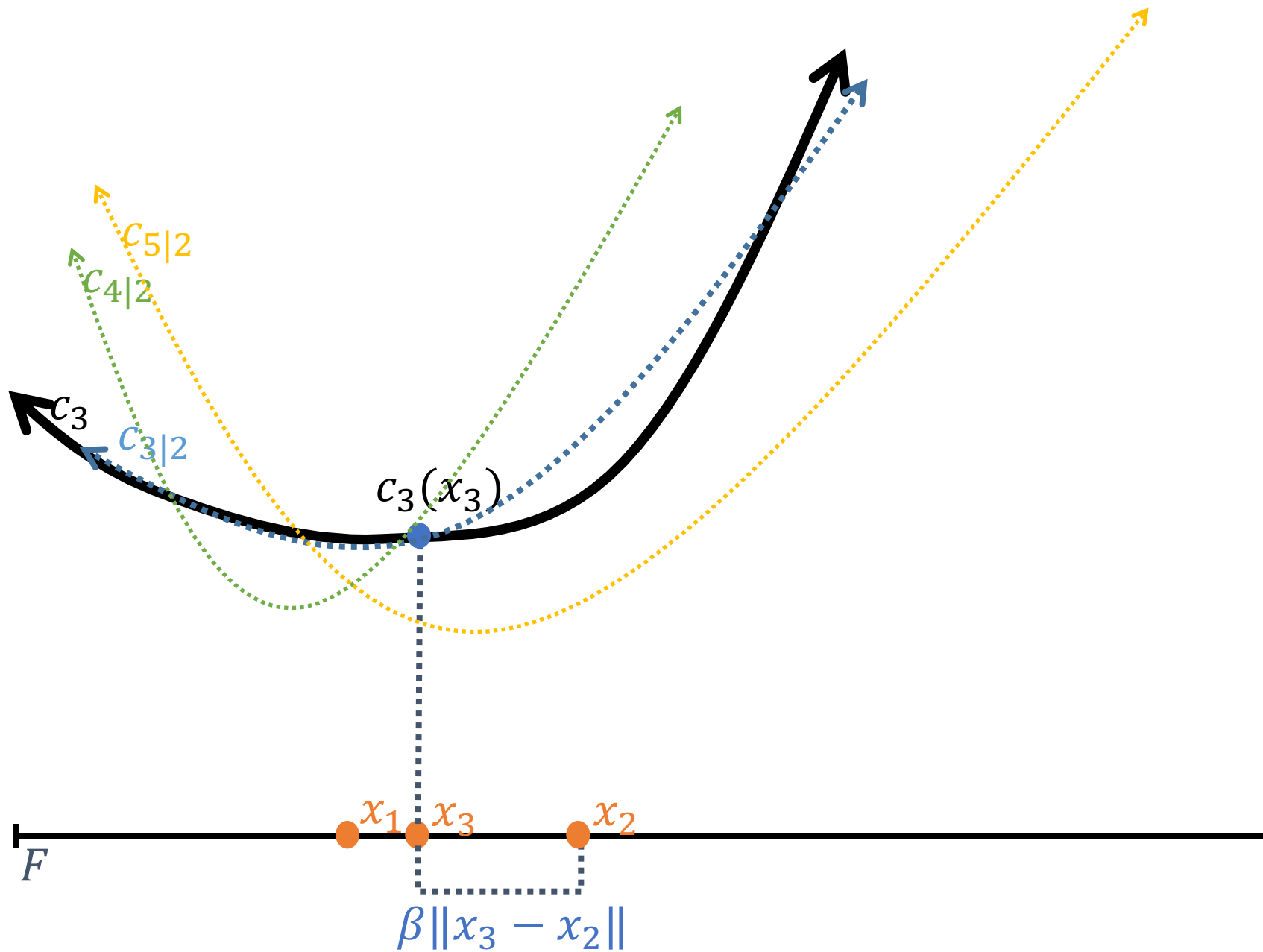


But we do not have a good understanding about how (imperfect) predictions impact online algorithm design

This talk: Online Convex Optimization Using Predictions







Online convex optimization using predictions

$x_1, y_1, x_2, y_2, x_3, y_3 \dots$ ← online

$$\min_{x_t \in F} \sum_t c(x_t, y_t) + \beta \|x_t - x_{t-1}\|$$

convex
switching cost

e.g. online tracking cost

$$c(x_t, y_t) = \frac{1}{2} \|y_t - Kx_t\|^2$$

Given prediction of y_t at time τ , $y_{t|\tau}$

Time	Information Available				Decision
1	$y_{1 0}$	$y_{2 0}$	$y_{3 0}$...	x_1

How do algorithms model prediction noise?

➤ **Learning and Algorithms: Perfect lookahead model**

(Near) perfect lookahead for w time steps and then adversarial

Both too optimistic and pessimistic

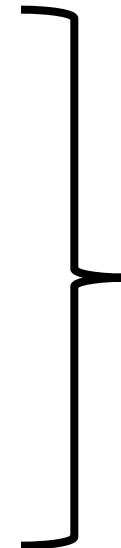


Worst case analysis

➤ **Control and Signal Processing: Stochastic model**

Assume a stochastic process and derive optimal predictor

Too sensitive to assumptions



Average case analysis

➤ **Systems Design: Numeric evaluation**

Test predictor given historic traces

No guarantee for performance

Our contribution: a general and tractable model for prediction

Key message: prediction allows

1. Overcoming “impossibility” results for OCO with minimal structural assumption
2. Mixture of average case and worst case analysis

Outline

1. **Background : regret and competitive ratio**

OCO without prediction

OCO with worst case prediction

2. Our prediction noise model

3. Algorithm design

4. OCO with stochastic prediction noise

Two communities, two metrics

Online Learning

$$\text{Regret}(\text{Alg}) = \sup_y [\text{Cost}(\text{Alg}) - \text{Cost}(\text{STA})]$$

Goal: sublinear regret

Online Algorithm

$$\text{Competitive ratio}(\text{Alg}) = \sup_y \left[\frac{\text{Cost}(\text{Alg})}{\text{Cost}(\text{OPT})} \right]$$

Goal: constant competitive ratio

Real applications want both

Guarantees without prediction

➤ Sublinear regret?

Yes, [Kivinen & Vempala 2002] [Bansal et al. 2003]
[Zinkevich 2003] [Hazan et al. 2007] [Lin et al. 2012] ...

➤ Constant CR?

Yes, but only for scalar case

[Blum et al. 1992] [Borodin et al. 1992][Blum & Burch 2000]
[Lin et al. 2011][Lin et al. 2012] ...

➤ Sublinear regret *and* constant CR?

Not even in scalar case! [Andrew et al. 2013]

Guarantees with prediction

1st cut, perfect lookahead:

$$y_{t|\tau} = y_t \text{ for any time } t \leq \tau + w$$

➤ Sublinear regret?

Yes, [Kivinen & Vempala 2002] [Bansal et al. 2003]

[Zinkevich 2003] [Hazan et al. 2007] [Lin et al. 2012] ...

➤ Constant CR?

Yes in general [Lin et al. 2013]

➤ Sublinear regret *and* constant CR?

Not without a lot of prediction [Chen et al. 2015]

Theorem:

An online algorithm with perfect lookahead requires unbounded lookahead window w to simultaneously achieve sublinear regret and a constant competitive ratio.

$w = \omega(1)$ as T grows

We may be using the wrong prediction model

Outline

1. Background : regret and competitive ratio
OCO without prediction
OCO with worst case prediction
2. **Our prediction noise model**
3. Algorithm design
4. OCO with stochastic prediction noise

What do we want in a prediction noise model?

- Predictions are “refined” as time goes forward
- Predictions are more noisy as you look further ahead
- Prediction errors can be correlated
- Should be general enough to incorporate detailed models

A more realistic prediction noise model

$$y_t = y_{t|\tau} + \underbrace{\sum_{s=\tau+1}^t f(t-s)e(s)}_{\text{prediction error}}$$

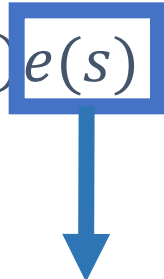
Realization that algorithm is trying to track

Prediction for time t given to algorithm at time τ

A more realistic prediction noise model

$$y_t = y_{t|\tau} + \sum_{s=\tau+1}^t f(t-s) \boxed{e(s)}$$

Per-step noise



How much uncertainty is there one step ahead?

$$y_t - y_{t|t-1} = f(0)e(t)$$

where $e(t)$ are white, mean zero (unbiased)

$$\text{and } f(0)=1, \mathbb{E}e(t)e(t)^T = R_e$$

A more realistic prediction noise model

$$y_t = y_{t|\tau} + \sum_{s=\tau+1}^t \text{Weighting factor } f(t-s) \epsilon(s)$$

How important is the noise at time $t - s$ for the prediction of t ?

A more realistic prediction noise model

$$y_t = y_{t|\tau} + \underbrace{\sum_{s=\tau+1}^t f(t-s)e(s)}_{\text{prediction error}}$$

- Predictions are “refined” as time goes forward
- Predictions are more noisy as you look further ahead

$$\mathbb{E} \left\| y_t - y_{t|\tau} \right\|^2 = \sigma^2 \sum_{s=0}^{t-\tau-1} \|f(s)\|^2$$

- Prediction errors can be correlated
- Form of errors matches many classic models

A more realistic prediction noise model

$$y_t = y_{t|\tau} + \underbrace{\sum_{s=\tau+1}^t f(t-s)e(s)}_{\text{prediction error}}$$

This form of prediction error matches what occurs in

- Prediction of a wide-sense stationary process using a [Weiner filter](#)
- Prediction of a linear dynamical system using a [Kalman filter](#)

A more realistic prediction noise model

$$y_t = y_{t|\tau} + \sum_{s=\tau+1}^t f(t-s)e(s)$$

Key observation: No assumption about y_t or how predictions are made

➔ Allows adversarial analysis using stochastic prediction noise

$$\mathbf{Regret}(\mathbf{Alg}) = \sup_y \mathbb{E}_e \text{cost}(\mathbf{Alg}) - \text{cost}(\text{STA})$$

$$\mathbf{Competitive Ratio}(\mathbf{Alg}) = \sup_y \mathbb{E}_e \frac{\text{cost}(\mathbf{Alg})}{\text{cost}(\text{Opt})}$$


Outline

1. Background : regret and competitive ratio
OCO without prediction
OCO with worst case prediction
2. Our prediction noise model
- 3. Algorithm design**
4. OCO with stochastic prediction noise

A natural suggestion: Model Predictive Control (MPC)

$$\begin{array}{c} \boxed{y_{t+1|t}, y_{t+2|t}, \dots, y_{t+w|t}} \quad y_{t+w+1|t}, y_{t+w+2|t}, \dots \\ \downarrow \\ x_{t+1}, x_{t+2}, \dots, x_{t+w} = \operatorname{argmin} \left\{ \sum_{s=t+1}^{t+w} \frac{1}{2} \|y_{s|t} - Kx_t\|^2 + \beta \|x_t - x_{t-1}\|_1 \right\} \end{array}$$

A natural suggestion: Model Predictive Control (MPC)

$$y_{t+1|t}, y_{t+2|t}, \dots, y_{t+w|t}, y_{t+w+1|t}, y_{t+w+2|t}, \dots$$
$$\boxed{y_{t+2|t+1}, y_{t+3|t+1}, \dots, y_{t+w+1|t+1}, y_{t+w+2|t+1}, y_{t+w+3|t+1}, \dots}$$

$$x_{t+2}, x_{t+3}, \dots, x_{t+w+1}$$

A natural suggestion: Model Predictive Control (MPC)

$$\begin{aligned} & y_{t+1|t}, y_{t+2|t}, \dots, y_{t+w|t}, y_{t+w+1|t}, y_{t+w+2|t}, \dots \\ & \quad y_{t+2|t+1}, y_{t+3|t+1}, \dots, y_{t+w+1|t+1}, y_{t+w+2|t+1}, y_{t+w+3|t+1}, \dots \\ & \quad \quad \boxed{y_{t+3|t+2}, y_{t+4|t+2}, \dots, y_{t+w+2|t+2}, y_{t+w+3|t+2}, y_{t+w+4|t+2}, \dots} \\ & \quad \quad \quad \downarrow \\ & \quad \quad \quad x_{t+3}, x_{t+4}, \dots, x_{t+w+2} \end{aligned}$$

But MPC doesn't work well in this setting ...

A more stable alternative: Averaging Fixed Horizon Control (AFHC)

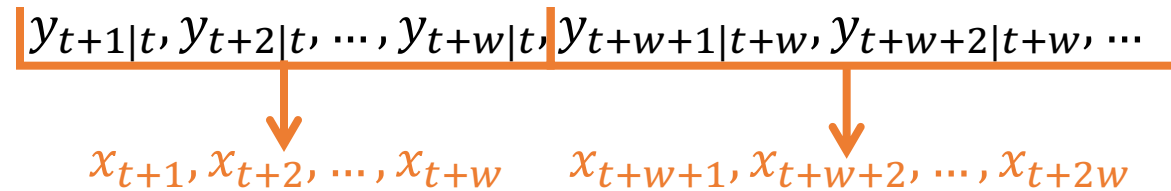
Fixed Horizon Control (FHC)

$$y_{t+1|t}, y_{t+2|t}, \dots, y_{t+w|t}$$

$$x_{t+1}, x_{t+2}, \dots, x_{t+w} = \operatorname{argmin} \left\{ \sum_{s=t+1}^{t+w} \frac{1}{2} \|y_{s|t} - Kx_t\|^2 + \beta \|x_t - x_{t-1}\|_1 \right\}$$

A more stable alternative: Averaging Fixed Horizon Control (AFHC)

Fixed Horizon Control (FHC)

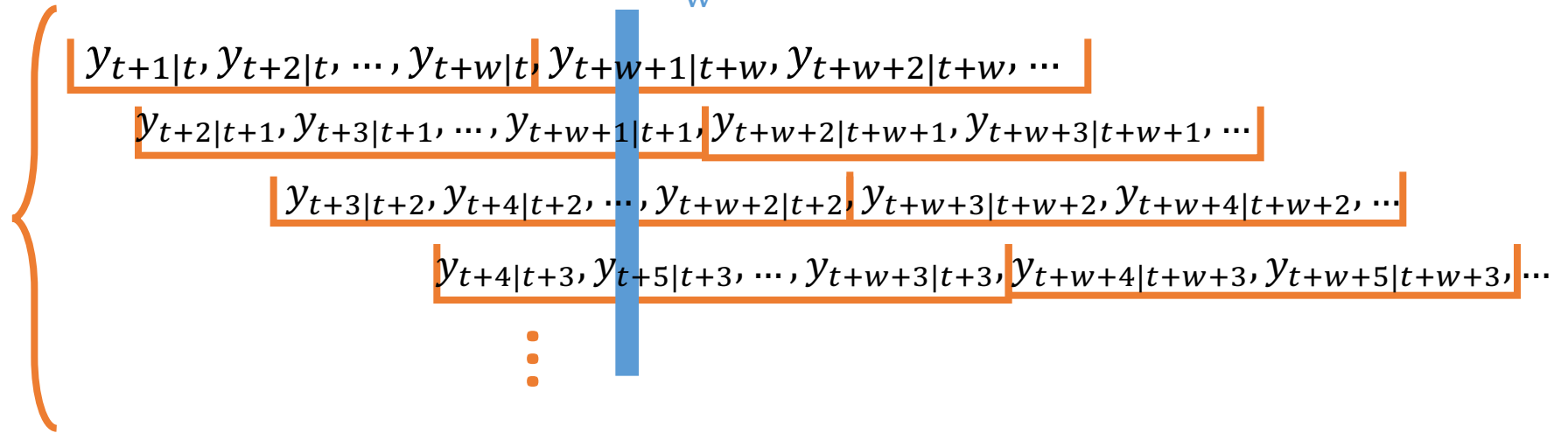


A more stable alternative: Averaging Fixed Horizon Control (AFHC)

Average choices of FHC algorithms

$$x_{AFHC} = \frac{1}{w} \sum_{k=1}^w x_{FHC}^{(k)}$$

w FHC algorithms

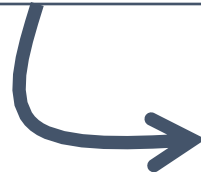


Outline

1. Background : regret and competitive ratio
OCO without prediction
OCO with worst case prediction
2. Our prediction noise model
3. Algorithm design
4. **OCO with stochastic prediction noise**

Theorem: AFHC(w) with $w = O(1)$ has sublinear regret and is constant competitive (in expectation) when $cost(OPT) = \Omega(T)$, and $cost(STA) \geq \alpha_1 T - o(T)$.

Theorem: AFHC(w) with $w = O(1)$ has sublinear regret and is constant competitive (in expectation) when $\text{cost}(OPT) = \Omega(T)$, and $\text{cost}(STA) \geq \alpha_1 T - o(T)$.

 How tight is this condition?

Theorem: Any online algorithm that chooses action independent of $e(t)$ has cost at least $\left\|R_e^{1/2}\right\|^2 T + o(T)$

No online algorithm can do well if $\text{cost}(OPT) \in o(T)$ or $\text{cost}(STA) \leq \left(\left\|R_e^{1/2}\right\|^2 - \gamma\right) T$ for some $\gamma > 0$.

Theorem: **AFHC**(w) with $w = O(1)$ has sublinear regret and is constant competitive (in expectation) when $cost(OPT) = \Omega(T)$, and $cost(STA) \geq \alpha_1 T - o(T)$.

How to choose w ?

Lemma: $\sup_y \mathbb{E}[cost(AFHC) - cost(OPT)] \leq \frac{T}{w} F(w) + g(\beta, K, w)$

Cumulative prediction error over w timesteps

We can compute the optimal lookahead w

Theorem: AFHC(w) with $w = O(1)$ has sublinear regret and is constant competitive (in expectation) when $\text{cost}(OPT) = \Omega(T)$, and $\text{cost}(STA) \leq \alpha_1 T - o(T)$.

How likely is large deviation from expected performance for AFHC?



Theorem: When $e(t)$ is independent, sub-Gaussian for all t , for sufficiently large u ,
 $\exists a, b, c > 0$ such that

$$\mathbb{P}(\text{cost}(\text{AFHC}) - \text{cost}(\text{Opt}) > t + \mu) \leq c \cdot \exp\left(-\frac{t^2}{a + bt}\right)$$

Intuition: the competitive difference of AFHC is a “smooth” function of $e(t)$

Our contribution: a general and tractable model for prediction

Key message: prediction allows

1. Overcoming “impossibility” results for OCO with minimal structural assumption

AFHC can achieve sublinear regret and constant CR

2. Balance between average case and worst case analysis

Concentration of AFHC around its mean performance

Online Convex Optimization Using Predictions

Niangjun Chen

Joint work with Anish Agarwal, Lachlan Andrew, Sid Barman, and Adam Wierman