Bandit PCA

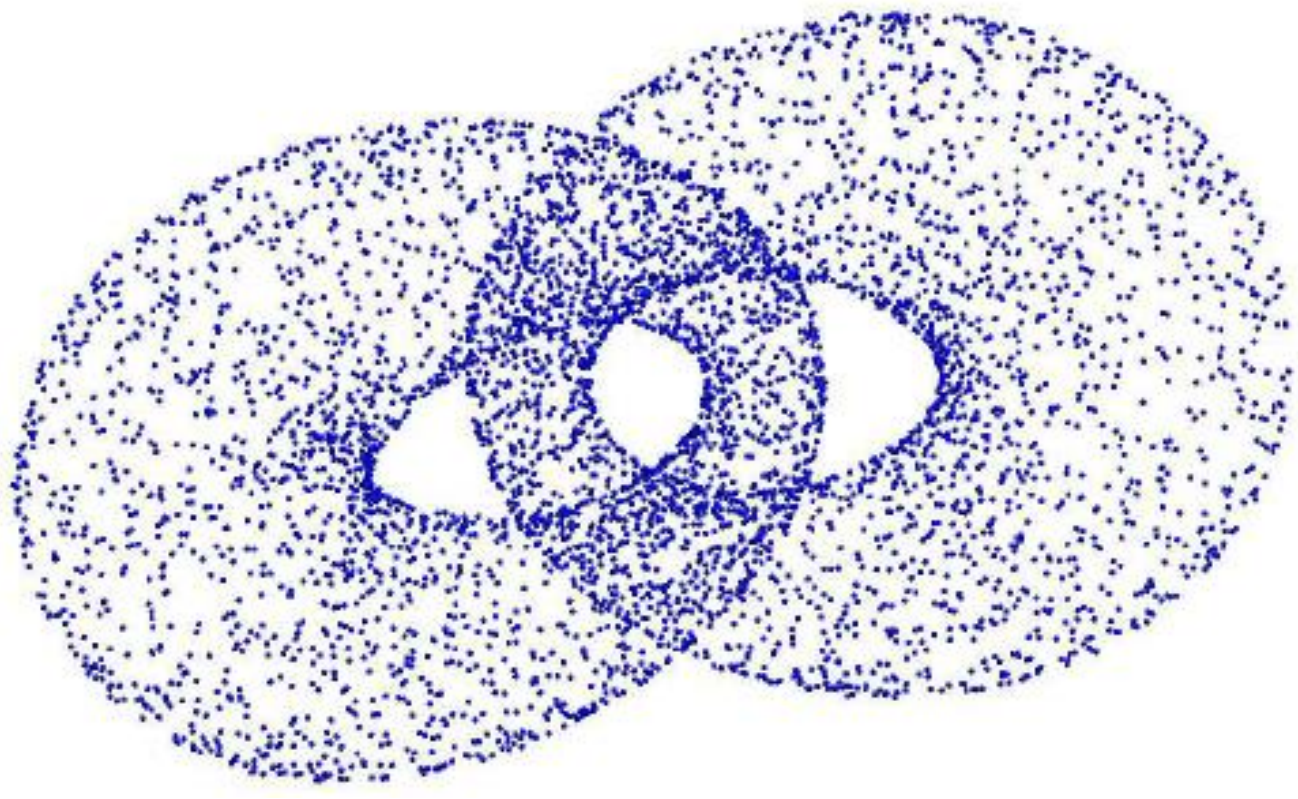
Gergely Neu
Univ. Pompeu Fabra (Barcelona, Spain)

joint work with Wojciech Kotłowski

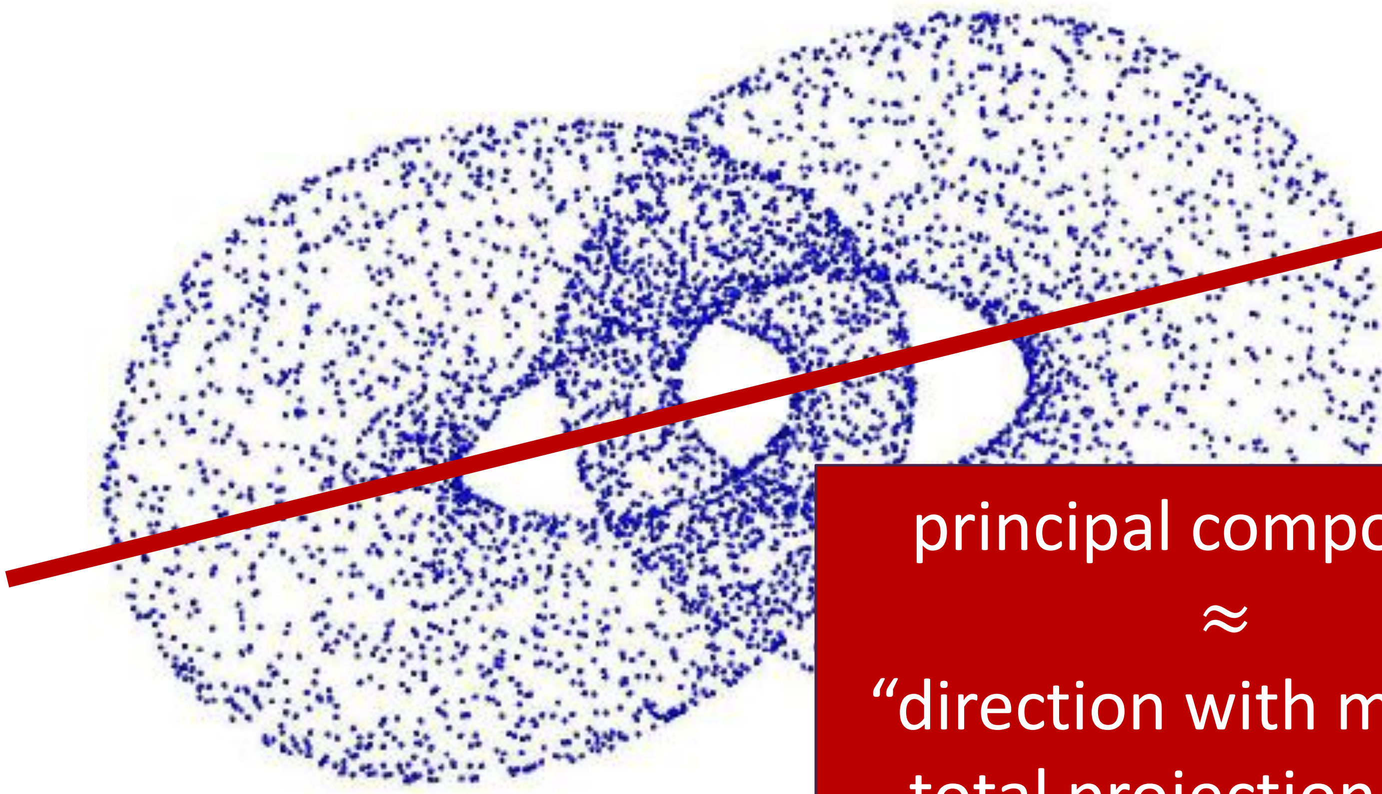
Appetizer

PCA,
bandit PCA,
phase retrieval

Principal component analysis (PCA)



Principal component analysis (PCA)



principal component

≈

“direction with minimal total projection loss”

Bandit PCA

Principal Component Analysis with

- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)

Bandit PCA

Principal Component Analysis with

- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)

$t = 1$

Bandit PCA

Principal Component Analysis with

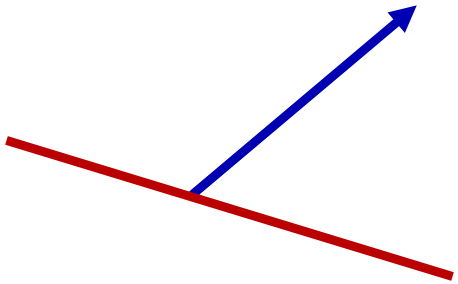
- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)

$t = 1$ environment chooses **hidden** vector x_t

Bandit PCA

Principal Component Analysis with

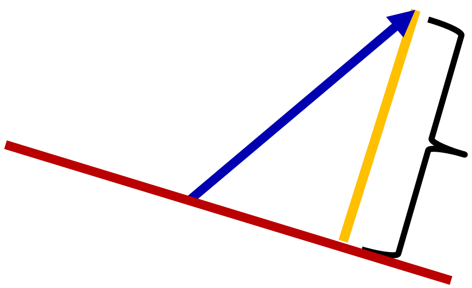
- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)



Bandit PCA

Principal Component Analysis with

- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)

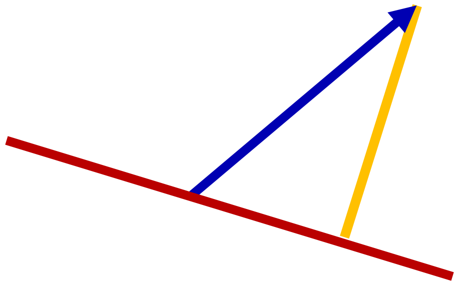


Bandit PCA

Principal Component Analysis with

- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)

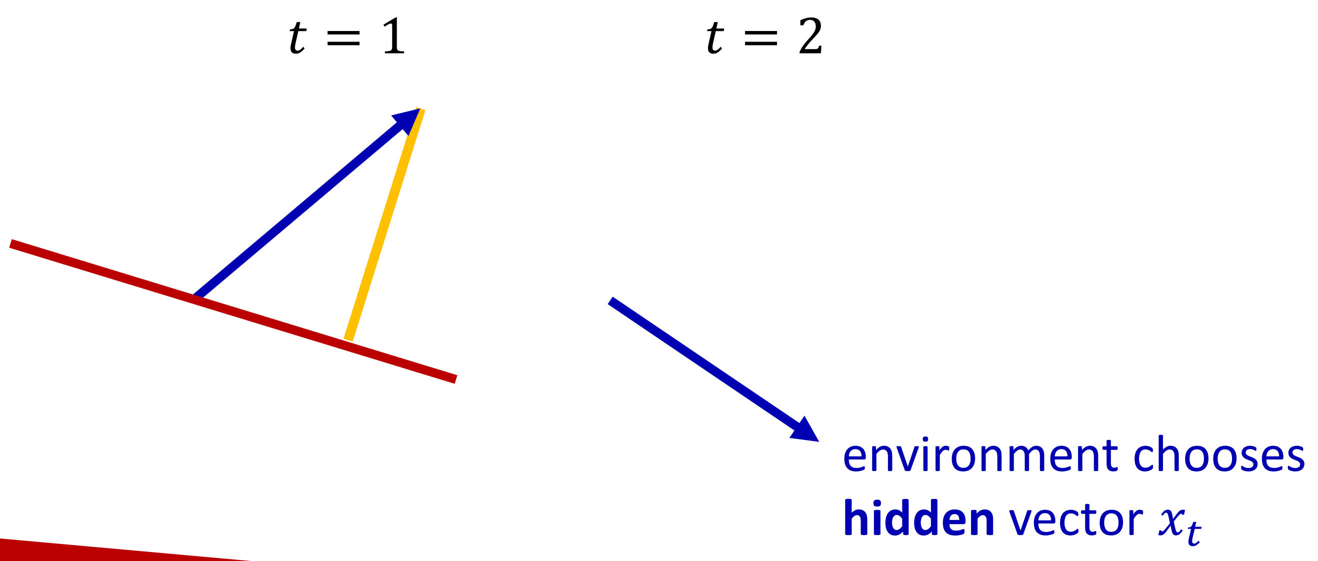
$t = 1$ $t = 2$



Bandit PCA

Principal Component Analysis with

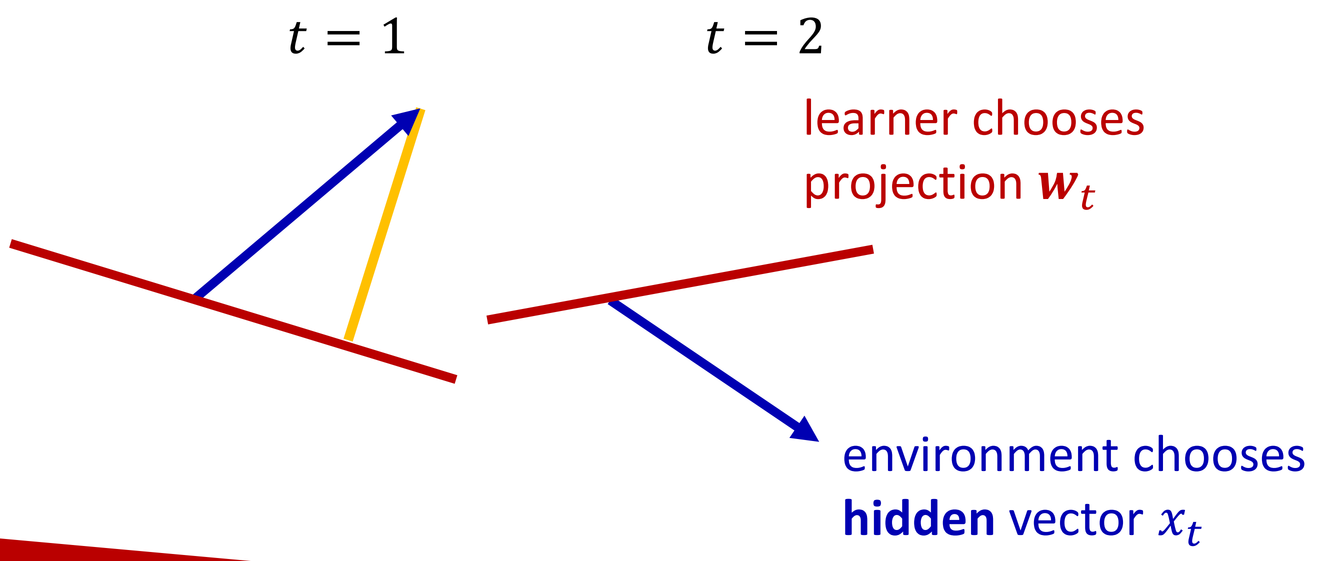
- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)



Bandit PCA

Principal Component Analysis with

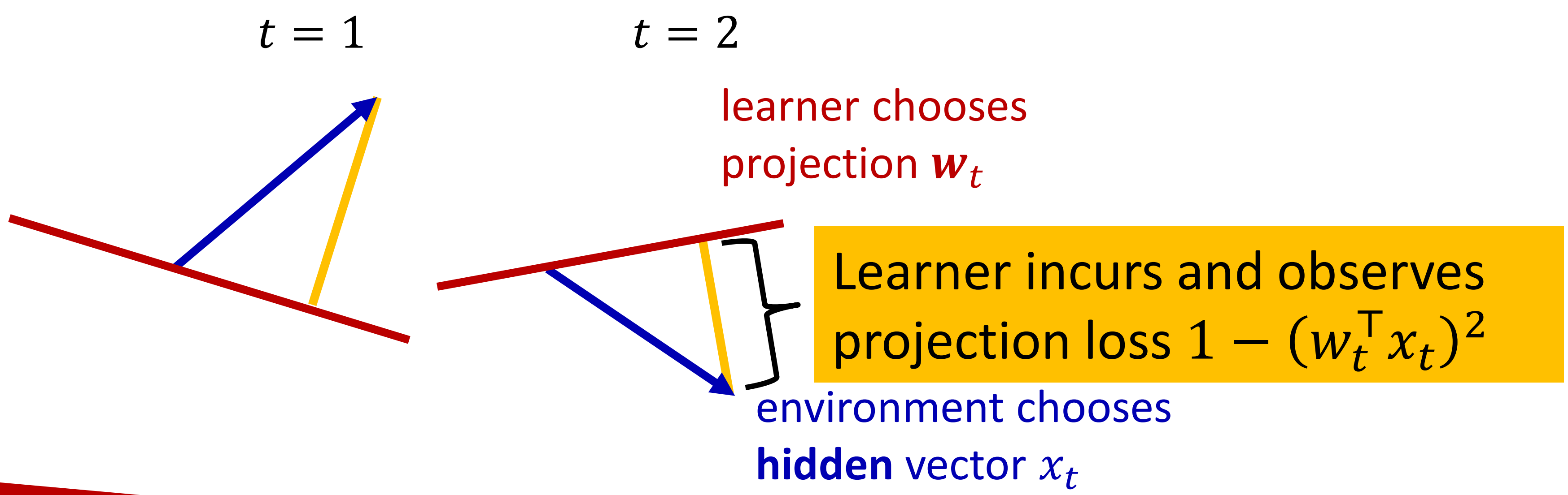
- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)



Bandit PCA

Principal Component Analysis with

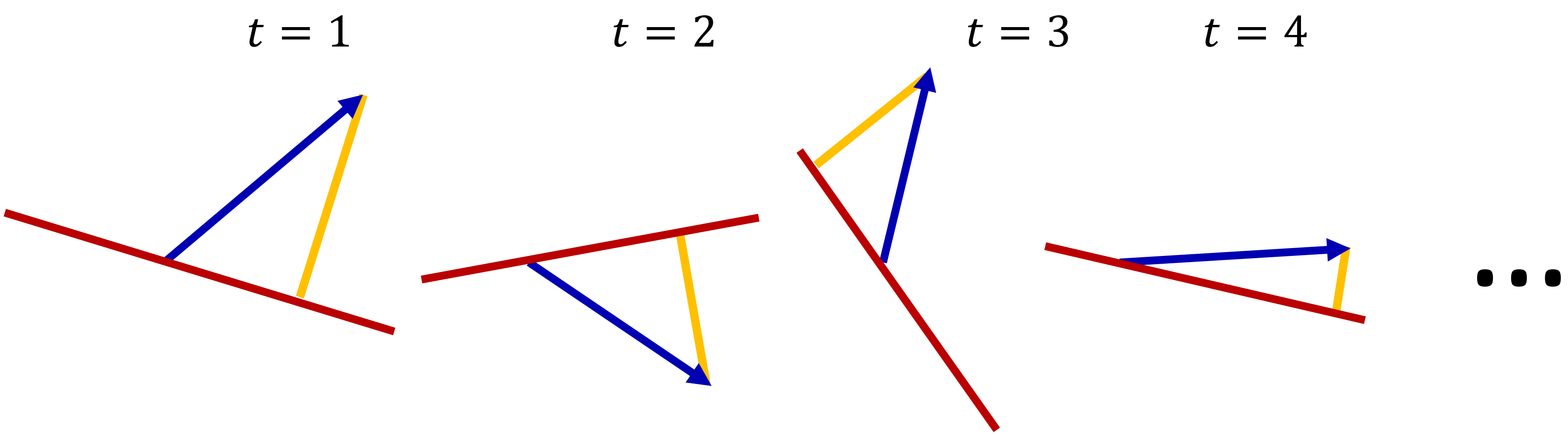
- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)



Bandit PCA

Principal Component Analysis with

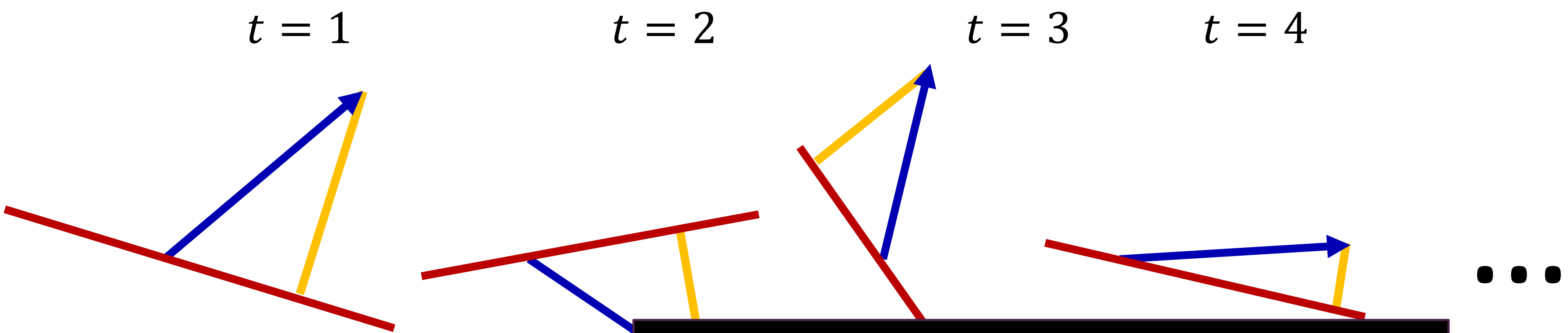
- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)



Bandit PCA

Principal Component Analysis with

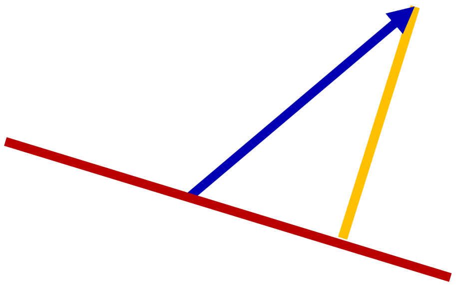
- sequentially chosen projections (online PCA)
- partial observability (bandit PCA)



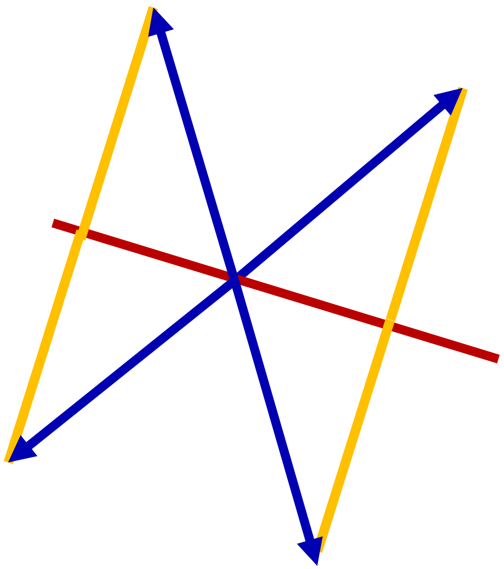
GOAL:

minimize total projection loss

Why is this hard?

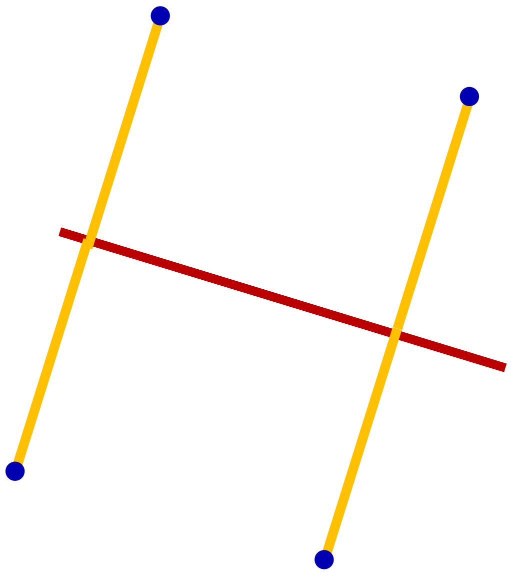


Why is this hard?



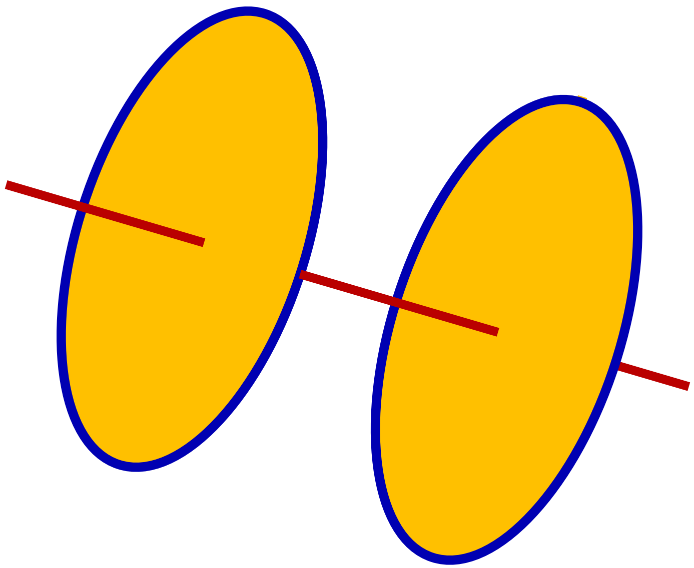
x_t is ambiguous!

Why is this hard?



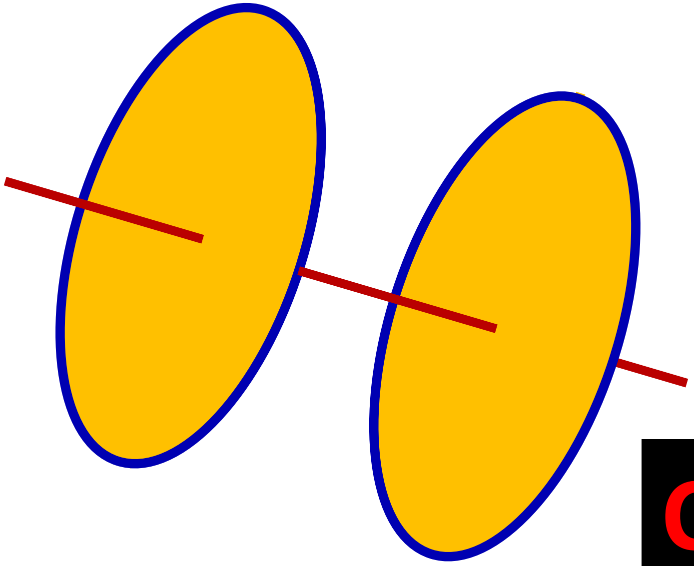
x_t is ambiguous!

Why is this hard?



x_t is ambiguous!
even worse in high dim

Why is this hard?



x_t is ambiguous!
even worse in high dim

Challenge:
How do we reconstruct x_t from a **single** projection?

NB: $\pm x_t$ are impossible to tell apart with any number of projections

Why is this interesting?

Why is this interesting?

Bandit PCA $\approx$ online phase retrieval

Why is this interesting?

Bandit PCA ≈ online phase retrieval

Phase retrieval:

- $\boldsymbol{w}_t \sim \mathcal{N}(0, I_{d\times d})$ i.i.d.
- $\boldsymbol{x}_t = \boldsymbol{x}$ fixed
- Observations:

  $|\boldsymbol{x}^\top \boldsymbol{w}_t|^2$ (+noise)

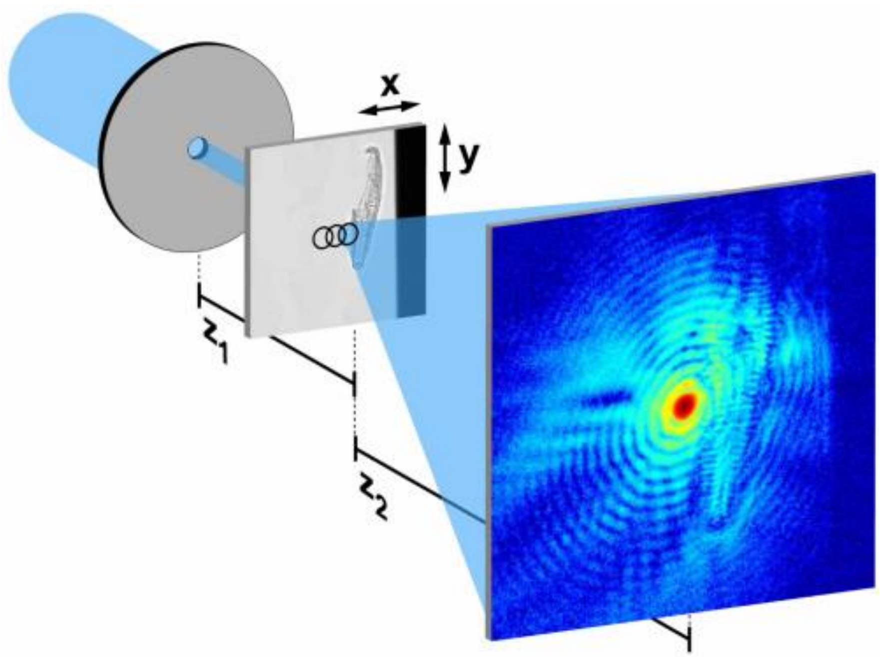
Fienup (1982), Millane (1990)

Why is this interesting?

Bandit PCA ≈ online phase retrieval

Phase retrieval:

- $\boldsymbol{w}_t \sim \mathcal{N}(0, I_{d\times d})$ i.i.d.
- $\boldsymbol{x}_t = \boldsymbol{x}$ fixed
- Observations:
  $|\boldsymbol{x}^\top \boldsymbol{w}_t|^2$ (+noise)
- Applications in
  - *diffractive imaging*
  - *X-ray crystallography*
  - *astronomy…*



Fienup (1982), Millane (1990)

Why is this interesting?

Bandit PCA ≈ online phase retrieval

Phase retrieval:

- $\boldsymbol{w}_t \sim \mathcal{N}(0, I_{d\times d})$ i.i.d.
- $\boldsymbol{x}_t = \boldsymbol{x}$ fixed
- Observations:
  $|\boldsymbol{x}^\top \boldsymbol{w}_t|^2$ (+noise)
- Applications in
  - *diffractive imaging*
  - *X-ray crystallography*
  - *astronomy…*

Fienup (1982), Millane (1990)

Bandit PCA:

- $\boldsymbol{w}_t$ chosen adaptively
- $\boldsymbol{x}_t$ arbitrary
- Observations:
  $|\boldsymbol{x}_t^\top \boldsymbol{w}_t|^2$ (+noise)

Why is this interesting?

Bandit PCA ≈ online phase retrieval

Phase retrieval:

- $\boldsymbol{w}_t \sim \mathcal{N}(0, I_{d\times d})$ i.i.d.
- $\boldsymbol{x}_t = \boldsymbol{x}$ fixed
- Observations:

  $|\boldsymbol{x}^\top \boldsymbol{w}_t|^2$ (+noise)
- Applications in
  - *diffractive imaging*
  - *X-ray crystallography*
  - *astronomy…*

Fienup (1982), Millane (1990)

Bandit PCA:

- $\boldsymbol{w}_t$ chosen adaptively
- $\boldsymbol{x}_t$ arbitrary
- Observations:

  $|\boldsymbol{x}_t^\top \boldsymbol{w}_t|^2$ (+noise)

Applicable in the same settings but with **adaptive measurements**!

Let’s get technical

Classic tricks for online PCA

Bandit PCA – general framework

For $t = 1,2,\ldots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks unit-norm vector $\boldsymbol{w}_t$
- Learner incurs and observes loss $\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$

Generalizes the basic PCA setup with $\boldsymbol{L}_t = \boldsymbol{x}_t \boldsymbol{x}_t^\top$

Bandit PCA – general framework

For $t = 1,2,\ldots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks unit-norm vector $\boldsymbol{w}_t$
- Learner incurs and observes loss $\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$

Generalizes the basic PCA setup with $\boldsymbol{L}_t = \boldsymbol{x}_t \boldsymbol{x}_t^\top$

GOAL:

minimize total expected regret

$$\mathrm{regret}_T = \max_{\boldsymbol{u}:\|\boldsymbol{u}\|=1} \mathbb{E}\left[\sum_{t=1}^{T} (\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t - \boldsymbol{u}^\top \boldsymbol{L}_t \boldsymbol{u})\right]$$

Bandit PCA vs. Online PCA

For $t = 1, 2, \ldots, T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks unit-norm vector $\boldsymbol{w}_t$
- Learner incurs and observes loss $\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$

Bandit PCA vs. Online PCA

For $t = 1,2,\dots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks unit-norm vector $\boldsymbol{w}_t$
- Learner incurs ~~and observes~~ loss $\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$
- Learner observes loss matrix $\boldsymbol{L}_t$

Online PCA

Warmuth and Kuzmin (2006, 2008)

Nie, Kotłowski, Warmuth (2013, 2016)

Bandit PCA vs. Online PCA

For $t = 1,2,\ldots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks unit-norm vector $\boldsymbol{w}_t$
- Learner incurs ~~and observes~~ loss $\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$
- Learner observes loss matrix $\boldsymbol{L}_t$

Online PCA

Warmuth and Kuzmin (2006, 2008)

Nie, Kotłowski, Warmuth (2013, 2016)

Our work:

bandit feedback

$\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$

Nonlinear losses?

Challenge:

the loss $\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$ is quadratic in the decision variable $\boldsymbol{w}_t$!

Nonlinear losses?

Challenge:

the loss $\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$ is quadratic in the decision variable $\boldsymbol{w}_t$!

Solution:

loss is actually linear in the matrix variable $\boldsymbol{w}_t \boldsymbol{w}_t^\top$:

$$\boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t = \mathrm{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$$

Warmuth and Kuzmin (2006)

Non-convex decision sets?

Challenge:

the set of feasible $\boldsymbol{w}\boldsymbol{w}^{\top}$ matrices is not convex (recall $||\boldsymbol{w}|| = 1$)

Non-convex decision sets?

Challenge:

the set of feasible $\boldsymbol{w}\boldsymbol{w}^\top$ matrices is not convex (recall $\|\boldsymbol{w}\| = 1$)

Solution: randomization!

Let $\mathcal{S} = \operatorname{conv}(\boldsymbol{w}\boldsymbol{w}^\top : \|\boldsymbol{w}\| = 1)$ be the set of density matrices $\boldsymbol{W}$ **and note that**

$$\max_{\boldsymbol{u}:\|\boldsymbol{u}\|=1} \boldsymbol{u}^\top \boldsymbol{L}\boldsymbol{u} = \max_{\boldsymbol{U}\in\mathcal{S}} \operatorname{tr}(\boldsymbol{U}\boldsymbol{L})$$

Warmuth and Kuzmin (2006)

Online PCA
= Online linear optimization

For $t = 1,2, \ldots, T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks density matrix $\boldsymbol{W}_t \in \mathcal{S}$
- Learner draws random $\boldsymbol{w}_t$ s.t. $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$
- Learner incurs loss

$$\langle \boldsymbol{w}_t \boldsymbol{w}_t^\top, \boldsymbol{L}_t \rangle \stackrel{\text{def}}{=} \operatorname{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$$

- Learner observes loss matrix $\boldsymbol{L}_t$

Bandit PCA
= Bandit linear optimization

For $t = 1,2,\ldots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks density matrix $\boldsymbol{W}_t \in \mathcal{S}$
- Learner draws random $\boldsymbol{w}_t$ s.t. $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$
- Learner incurs and observes loss

$$\langle \boldsymbol{w}_t \boldsymbol{w}_t^\top, \boldsymbol{L}_t \rangle \stackrel{\text{def}}{=} \operatorname{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$$

Bandit PCA
= Bandit linear optimization

For $t = 1,2,\ldots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks density matrix $\boldsymbol{W}_t \in \mathcal{S}$
- Learner draws random $\boldsymbol{w}_t$ s.t. $\mathbb{E}[\boldsymbol{w}_t\boldsymbol{w}_t^\top] = \boldsymbol{W}_t$
- Learner incurs and observes loss

$$\langle \boldsymbol{w}_t\boldsymbol{w}_t^\top, \boldsymbol{L}_t \rangle \overset{\text{def}}{=} \operatorname{tr}(\boldsymbol{w}_t\boldsymbol{w}_t^\top \boldsymbol{L}_t)$$

Idea:

Apply a generic linear bandit algorithm!

Bandit PCA
= Bandit linear optimization

For $t = 1,2,\ldots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks density matrix $\boldsymbol{W}_t \in \mathcal{S}$
- Learner draws random $\boldsymbol{w}_t$ s.t. $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$
- Learner incurs and observes loss
  $\langle \boldsymbol{w}_t \boldsymbol{w}_t^\top, \boldsymbol{L}_t \rangle \overset{\text{def}}{=} \mathrm{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$

Idea:
Apply a generic linear bandit algorithm!

GeometricHedge guarantees
$$\mathrm{regret}_T = \tilde{O}\left(d^2 \sqrt{T}\right)$$

Dani, Hayes, Kakade (2008),
Bubeck and Eldan (2015)

Bandit PCA
= Bandit linear optimization

For $t = 1,2,\ldots,T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks density matrix $\boldsymbol{W}_t \in \mathcal{S}$
- Learner draws random $\boldsymbol{w}_t$ s.t. $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$
- Learner incurs and observes loss

$$\langle \boldsymbol{w}_t \boldsymbol{w}_t^\top, \boldsymbol{L}_t \rangle \stackrel{\text{def}}{=} \operatorname{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$$

Idea:
Apply a generic linear bandit algorithm!

GeometricHedge guarantees

$$\text{regret}_T = \tilde{O}\left(d^2\sqrt{T}\right)$$

Dani, Hayes, Kakade (2008),
Bubeck and Eldan (2015)

BUT
no polytime implementation is known ☹☹☹

Bandit PCA
= Bandit linear optimization

For $t = 1,2, \dots, T$

- Environment picks secret loss matrix $\boldsymbol{L}_t$
- Learner picks density matrix $\boldsymbol{W}_t \in \mathcal{S}$
- Learner draws random $\boldsymbol{w}_t$ s.t. $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$
-

Idea:

Apply a generic linear bandit ...thm!

Ge

Dani, Hayes, Kakade (2008),
Bubeck and Eldan (2015)

...T
...time
...ntation
is known ☹☹☹

Our contribution:

a fast algorithm with regret

$$O\left(d^{3/2}\sqrt{T \log T}\right)$$

Main course

Algorithm

Main results

Main course

Algorithm

Main results

Online Mirror Descent for bandit PCA

Idea: rely on the good old template

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W}\in\mathcal{S}}\left\{\eta\left\langle \boldsymbol{W}, \widehat{\boldsymbol{L}}_t \right\rangle + D(\boldsymbol{W}\|\boldsymbol{W}_t)\right\}$$

Online Mirror Descent for bandit PCA

Idea: rely on the good old template

$$W_{t+1} = \arg\min_{W \in \mathcal{S}} \left\{ \eta \left\langle W, \widehat{L}_t \right\rangle + D(W \| W_t) \right\}$$

+

Sample w_t so that $\mathbb{E}[w_t w_t^\top] = W_t$

Online Mirror Descent for bandit PCA

loss estimate $\hat{\boldsymbol{L}}_t = ?$

divergence $D = ?$

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W} \in \mathcal{S}} \{ \eta \langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \}$$

+

sampling scheme?

Sample $\boldsymbol{w}_t$ so that $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$

Online Mirror Descent for bandit PCA

loss estimate $\hat{L}_t =?$

divergence $D =?$

$\langle W, \hat{L}_t \rangle + D(W \| W_t)\}$

None of these are obvious at all!

+ sampling scheme?

Sample w_t so that $\mathbb{E}[w_t w_t^\top] = W_t$

sampling scheme?

loss estimate $\hat{L}_t =$?

Sampling scheme?

First thought:

decompose $\boldsymbol{W}_t = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$

and sample w_t so that $\mathbb{P}[\boldsymbol{w}_t = \boldsymbol{u}_i] = \lambda_i$

Warmuth and Kuzmin (2006)

Sampling scheme?

Recall:
$\sum_i \lambda_i = \mathrm{tr}(W) = 1$
$\lambda_i \geq 0$

First thought:

decompose $\boldsymbol{W}_t = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$
and sample w_t so that $\mathbb{P}[\boldsymbol{w}_t = \boldsymbol{u}_i] = \lambda_i$

Warmuth and Kuzmin (2006)

Sampling scheme?

Recall:
$\sum_i \lambda_i = \mathrm{tr}(W) = 1$
$\lambda_i \geq 0$

First thought:

decompose $\boldsymbol{W}_t = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$
and sample w_t so that $\mathbb{P}[\boldsymbol{w}_t = \boldsymbol{u}_i] = \lambda_i$

Warmuth and Kuzmin (2006)

Unbiased: $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$

Sampling scheme?

Recall:
$\sum_i \lambda_i = \mathrm{tr}(W) = 1$
$\lambda_i \geq 0$

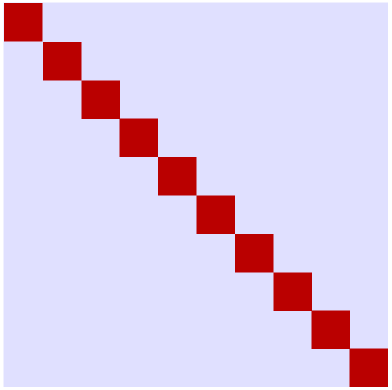
First thought:

decompose $\boldsymbol{W}_t = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$
and sample w_t so that $\mathbb{P}[\boldsymbol{w}_t = \boldsymbol{u}_i] = \lambda_i$

Warmuth and Kuzmin (2006)

Unbiased: $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$

only senses "diagonal" elements of $\boldsymbol{L}_t$ ☹☹



Sampling scheme?

Recall:
$\sum_i \lambda_i = \mathrm{tr}(W) = 1$
$\lambda_i \geq 0$

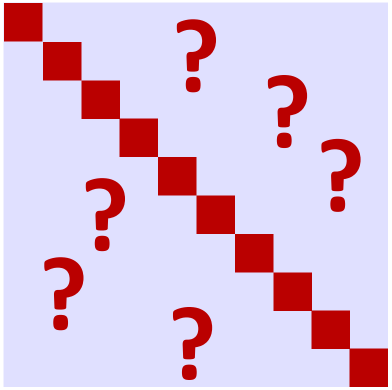
First thought:

decompose $\boldsymbol{W}_t = \sum_i \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\top$
and sample w_t so that $\mathbb{P}[\boldsymbol{w}_t = \boldsymbol{u}_i] = \lambda_i$

Warmuth and Kuzmin (2006)

Unbiased: $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$

only senses "diagonal"
elements of $\boldsymbol{L}_t$ ☹☹



Our key trick:
sparse off-diagonal sampling

Sparse sampling

- sample **two** indices
$$i, j \sim \lambda$$
- if $i = j$, set
$$\boldsymbol{w}_t = \boldsymbol{u}_i$$
- otherwise draw random sign $s \in \{-1, 1\}$ and set
$$\boldsymbol{w}_t = \frac{1}{\sqrt{2}}\left(\boldsymbol{u}_i + s\boldsymbol{u}_j\right)$$

Our key trick:
sparse off-diagonal sampling

Sparse sampling

- sample **two** indices

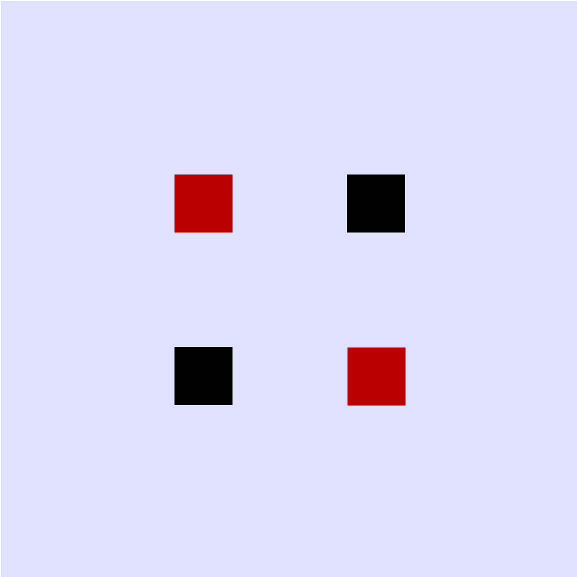
$$i, j \sim \lambda$$

- if $i = j$, set

$$\boldsymbol{w}_t = \boldsymbol{u}_i$$

- otherwise draw random sign $s \in \{-1, 1\}$ and set

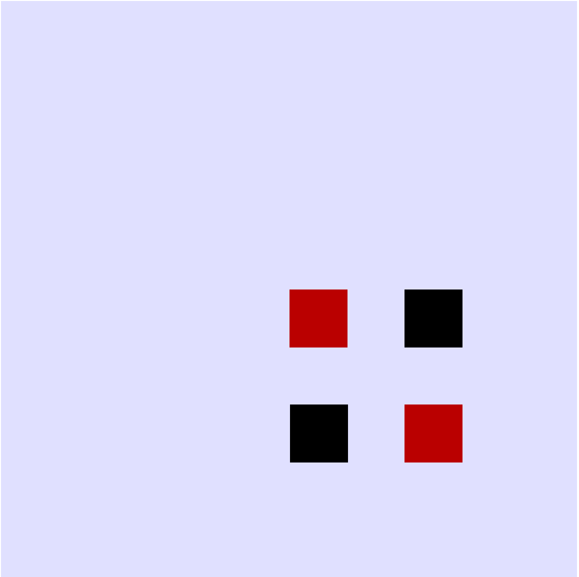
$$\boldsymbol{w}_t = \frac{1}{\sqrt{2}}\left(\boldsymbol{u}_i + s\boldsymbol{u}_j\right)$$



Our key trick: sparse off-diagonal sampling

Sparse sampling

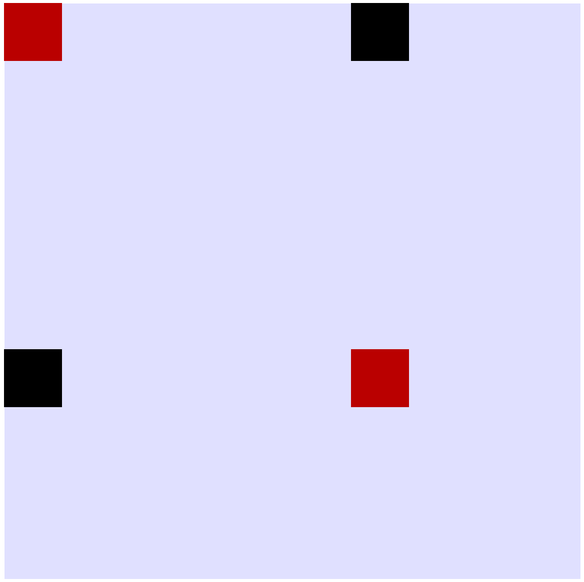
- sample **two** indices
$$i, j \sim \lambda$$
- if $i = j$, set
$$\boldsymbol{w}_t = \boldsymbol{u}_i$$
- otherwise draw random sign $s \in \{-1,1\}$ and set
$$\boldsymbol{w}_t = \frac{1}{\sqrt{2}}\left(\boldsymbol{u}_i + s\boldsymbol{u}_j\right)$$



Our key trick: sparse off-diagonal sampling

Sparse sampling

- sample **two** indices
  $$i, j \sim \lambda$$
- if $i = j$, set
  $$\boldsymbol{w}_t = \boldsymbol{u}_i$$
- otherwise draw random sign $s \in \{-1, 1\}$ and set
  $$\boldsymbol{w}_t = \frac{1}{\sqrt{2}}\left(\boldsymbol{u}_i + s\boldsymbol{u}_j\right)$$



Our key trick:
sparse off-diagonal sampling

Sparse sampling

- sample **two** indices

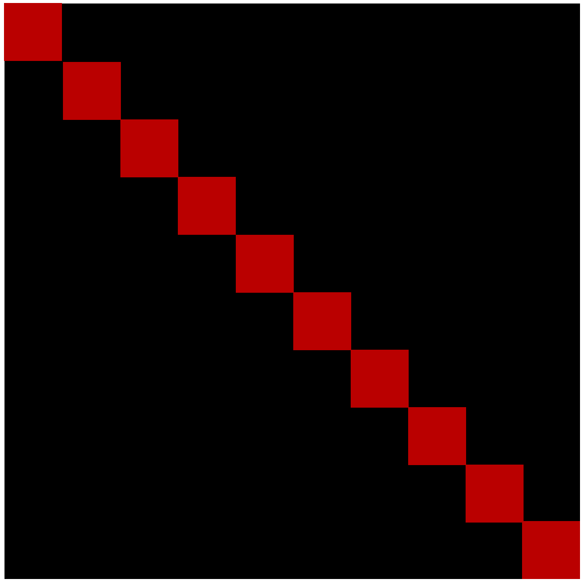
$$i, j \sim \lambda$$

- if $i = j$, set

$$\boldsymbol{w}_t = \boldsymbol{u}_i$$

- otherwise draw random sign $s \in \{-1,1\}$ and set

$$\boldsymbol{w}_t = \frac{1}{\sqrt{2}}\left(\boldsymbol{u}_i + s\boldsymbol{u}_j\right)$$



Unbiased: $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$

Estimating the loss matrix

Sparse sampling

- sample **two** indices
$$i, j \sim \lambda$$
- if $i = j$, set
$$\boldsymbol{w}_t = \boldsymbol{u}_i$$
- otherwise draw random sign $s \in \{-1, 1\}$ and set
$$\boldsymbol{w}_t = \frac{1}{\sqrt{2}}(\boldsymbol{u}_i + s\boldsymbol{u}_j)$$

Loss estimation

- let $\ell = \boldsymbol{w}_t^\top \boldsymbol{L}_t \boldsymbol{w}_t$
- if $i = j$, set
$$\hat{\boldsymbol{L}}_t = \frac{\ell}{\lambda_i^2} \boldsymbol{u}_i \boldsymbol{u}_i^\top$$
- otherwise set
$$\hat{\boldsymbol{L}}_t = \frac{s\ell}{\lambda_i \lambda_j} \left(\boldsymbol{u}_j \boldsymbol{u}_i^\top + \boldsymbol{u}_i \boldsymbol{u}_j^\top\right)$$

Unbiasedness of the estimator

Lemma:

$$\mathbb{E}\left[\hat{\boldsymbol{L}}_t\right] = \boldsymbol{L}_t$$

Unbiasedness of the estimator

Lemma:

$$\mathbb{E}\big[\hat{\boldsymbol{L}}_t\big] = \boldsymbol{L}_t$$

Proof:

$$\mathbb{E}\left[\widetilde{\boldsymbol{L}}\right] = \underbrace{\sum_i \lambda_i^2 \operatorname{tr}(\boldsymbol{u}_i\boldsymbol{u}_i^\intercal \boldsymbol{L})\frac{1}{\lambda_i^2}\boldsymbol{u}_i\boldsymbol{u}_i^\intercal}_{\text{when } I=J}$$

$$+\underbrace{\sum_{i\neq j}\lambda_i\lambda_j\mathbb{E}_s\left[\operatorname{tr}\left(\frac{1}{2}(\boldsymbol{u}_i+s\boldsymbol{u}_j)(\boldsymbol{u}_i+s\boldsymbol{u}_j)^\intercal\boldsymbol{L}\right)\frac{s}{2\lambda_i\lambda_j}(\boldsymbol{u}_i\boldsymbol{u}_j^\intercal+\boldsymbol{u}_j\boldsymbol{u}_i^\intercal)\right]}_{\text{when } I\neq J}$$

$$= \sum_i L_{ii}\boldsymbol{u}_i\boldsymbol{u}_i^\intercal + \frac{1}{4}\sum_{i\neq j}(L_{ii}+L_{jj})\underbrace{\mathbb{E}_s\left[s\right]}_{=0}(\boldsymbol{u}_i\boldsymbol{u}_j^\intercal+\boldsymbol{u}_j\boldsymbol{u}_i^\intercal) + \frac{1}{2}\sum_{i\neq j}L_{ij}(\boldsymbol{u}_i\boldsymbol{u}_j^\intercal+\boldsymbol{u}_j\boldsymbol{u}_i^\intercal)$$

$$= \sum_{ij} L_{ij}\boldsymbol{u}_i\boldsymbol{u}_j^\intercal = \boldsymbol{L},$$

where in the second inequality we used the fact that $s^2 = 1$. ■

divergence $D =$?

What divergence?

First thought:

the usual quantum relative entropy

$$D(\boldsymbol{W} \| \boldsymbol{U}) = \boldsymbol{W} \log(\boldsymbol{W}\boldsymbol{U}^{-1})$$

induced by the quantum entropy $R(\boldsymbol{W}) = \boldsymbol{W} \log \boldsymbol{W}$

What divergence?

First thought:

the usual quantum relative entropy

$$D(\boldsymbol{W} \| \boldsymbol{U}) = \boldsymbol{W} \log(\boldsymbol{W}\boldsymbol{U}^{-1})$$

induced by the quantum entropy $R(\boldsymbol{W}) = \boldsymbol{W} \log \boldsymbol{W}$

a.k.a. “Matrix Hedge”

Warmuth and Kuzmin (2006)

Matrix Hedge for Bandit PCA does not work?

W.K.

June 25, 2018

Consider the adversarial $k = 1$ PCA with bandit feedback. In each trial, the algorithm plays with a rank-one matrix $\boldsymbol{w}_t \boldsymbol{w}_t^\top$ with $\boldsymbol{w}_t \in \mathbb{R}^d$, $\|\boldsymbol{w}_t\| = 1$. Then, nature chooses a symmetric loss matrix $\boldsymbol{L}_t \in \mathbb{R}^{d \times d}$ with eigenvalues bounded in $[0, 1]$, and the algorithm receives and observes loss $\ell_t = \mathrm{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$.

We start with a standard bound on the Matrix Hedge algorithm: for any loss sequence $\widetilde{\boldsymbol{L}}_1, \ldots, \widetilde{\boldsymbol{L}}_T$ such that each $\widetilde{\boldsymbol{L}}_t$ has eigenvalues in the range $[-a, \infty)$, the sequence of density matrices $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_T$ produced by Matrix Hedge with fixed learning rate η has regret against a comparator density matrix $\boldsymbol{U}$ upper-bounded by:

$$\mathrm{regret}_T(\boldsymbol{U}) = \sum_{t=1}^{T} \mathrm{tr}((\boldsymbol{W}_t - \boldsymbol{U})\widetilde{\boldsymbol{L}}_t) \leq \frac{\ln d}{\eta} + \kappa(\eta a)\eta \sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{W}_t \widetilde{\boldsymbol{L}}_t^2),$$

where $\kappa(x) = \frac{e^x - x - 1}{x^2}$. The trick is now to use this bound in the bandit case as follows: in each trial $t = 1, \ldots, T$, the algorithm probabilistically chooses $\boldsymbol{w}_t \boldsymbol{w}_t^\top$ such that $\mathbb{E}_t[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$ (where $\mathbb{E}_t[\cdot]$ denotes the conditional expectation with respect the randomness at trial t, conditioned on all the past); then, the algorithm observes ℓ_t and produced an estimate $\widetilde{\boldsymbol{L}}_t$ of the loss matrix $\boldsymbol{L}_t$, with eigenvalues in $[-a, \infty]$, such that $\mathbb{E}_t[\widetilde{\boldsymbol{L}}_t] = \boldsymbol{L}_t + c_t \boldsymbol{I}$ (the estimate is allowed to be biased by a multiplicity of identity matrix!). The expected regret of the algorithm is given by:

Matrix Hedge for Bandit PCA does not work?

W.K.

June 25, 2018

Doesn't work indeed

☹☹☹

In each trial, the algorithm plays with a
e chooses a symmetric loss matrix $\boldsymbol{L}_t \in$
ives and observes loss $\ell_t = \mathrm{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$.

We start with a standard bound on the Matrix Hedge algorithm: for any loss sequence $\widetilde{\boldsymbol{L}}_1, \ldots, \widetilde{\boldsymbol{L}}_T$ such that each $\widetilde{\boldsymbol{L}}_t$ has eigenvalues in the range $[-a, \infty)$, the sequence of density matrices $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_T$ produced by Matrix Hedge with fixed learning rate η has regret against a comparator density matrix $\boldsymbol{U}$ upper-bounded by:

$$\mathrm{regret}_T(\boldsymbol{U}) = \sum_{t=1}^{T} \mathrm{tr}((\boldsymbol{W}_t - \boldsymbol{U})\widetilde{\boldsymbol{L}}_t) \leq \frac{\ln d}{\eta} + \kappa(\eta a)\eta \sum_{t=1}^{T} \mathrm{tr}(\boldsymbol{W}_t \widetilde{\boldsymbol{L}}_t^2),$$

where $\kappa(x) = \frac{e^x - x - 1}{x^2}$. The trick is now to use this bound in the bandit case as follows: in each trial $t = 1, \ldots, T$, the algorithm probabilistically chooses $\boldsymbol{w}_t \boldsymbol{w}_t^\top$ such that $\mathbb{E}_t[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$ (where $\mathbb{E}_t[\cdot]$ denotes the conditional expectation with respect the randomness at trial t, conditioned on all the past); then, the algorithm observes ℓ_t and produced an estimate $\widetilde{\boldsymbol{L}}_t$ of the loss matrix $\boldsymbol{L}_t$, with eigenvalues in $[-a, \infty]$, such that $\mathbb{E}_t[\widetilde{\boldsymbol{L}}_t] = \boldsymbol{L}_t + c_t \boldsymbol{I}$ (the estimate is allowed to be biased by a multiplicity of identity matrix!). The expected regret of the algorithm is given by:

Matrix Hedge for Bandit PCA does not work?

W.K.

June 25, 2018

Doesn't work indeed ☹☹☹

In each trial, the algorithm plays with a ... e chooses a symmetric loss matrix $\boldsymbol{L}_t \in$ ... ives and observes loss $\ell_t = \operatorname{tr}(\boldsymbol{w}_t \boldsymbol{w}_t^\top \boldsymbol{L}_t)$.

We start with a standard bound on the Matrix Hedge algorithm: for any loss sequence $\widetilde{\boldsymbol{L}}_1, \ldots, \widetilde{\boldsymbol{L}}_T$ such that each $\widetilde{\boldsymbol{L}}_t$ has eigenvalues in the range $[-a, \infty)$, the sequence of density matrices $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_T$ produced by Matrix Hedge with fixed learning rate η has regret against a comparator density matrix $\boldsymbol{U}$

$$\text{regret}_T(\boldsymbol{U}) = \sum_{t=1}^{T} \operatorname{tr}((\boldsymbol{W}_t - \boldsymbol{U})\widetilde{\boldsymbol{L}}_t) \leq \frac{\ln d}{\eta} + \kappa(\eta a)\eta \sum_{t=1}^{T} \operatorname{tr}(\boldsymbol{W}_t \widetilde{\boldsymbol{L}}_t^2),$$

This bound is virtually useless

(for complicated reasons)

where $\kappa(x) = \frac{e^x - x - 1}{x^2}$. The trick is now to use th... in the bandit case as follows: in each trial $t = 1, \ldots, T$, the algorithm probabilistically ... (where $\mathbb{E}_t[\cdot]$ denotes the conditional expectation with res... past); then, the algorithm observes ℓ_t and produce... ues in $[-a, \infty]$, such that $\mathbb{E}_t[\widetilde{\boldsymbol{L}}_t] = \boldsymbol{L}_t + c_t \boldsymbol{I}$ (the e... entity matrix!). The expected regret of the algorit...

The right divergence

$$D(\boldsymbol{W} \| \boldsymbol{U}) = \mathrm{tr}(\boldsymbol{W}\boldsymbol{U}^{-1}) - \log\det(\boldsymbol{W}\boldsymbol{U}^{-1}) - d$$

The Bregman divergence induced by

$$R(\boldsymbol{W}) = -\log\det\boldsymbol{W}$$

a.k.a. Stein's loss (James and Stein, 1967)

The right divergence

$$D(\boldsymbol{W} \| \boldsymbol{U}) = \mathrm{tr}(\boldsymbol{W}\boldsymbol{U}^{-1}) - \log\det(\boldsymbol{W}\boldsymbol{U}^{-1}) - d$$

The Bregman divergence induced by
$R(\boldsymbol{W}) = -\log\det\boldsymbol{W}$
a.k.a. Stein's loss (James and Stein, 1967)

The matrix generalization of the trendy "log-barrier" regularizer $-\sum_i \log p_i$
(Foster et al., 2016, Agarwal et al., 2017, Bubeck et al. 2018, Wei and Luo, 2018, Luo et al., 2018, …)

Implementing the update

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W} \in \mathcal{S}} \left\{ \eta \left\langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \right\rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

Implementing the update

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W}\in\mathcal{S}} \left\{ \eta \left\langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \right\rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

= by the classic decomposition

$$\widetilde{\boldsymbol{W}}_{t+1} = \arg\min_{\boldsymbol{W}} \left\{ \eta \left\langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \right\rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

$$\boldsymbol{W}_{t+1} = \arg\min_{W\in\mathcal{S}} D\left(\boldsymbol{W} \| \widetilde{\boldsymbol{W}}_{t+1}\right)$$

Implementing the update

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W}\in\mathcal{S}} \left\{ \eta \left\langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \right\rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

= by the classic decomposition

$$\widetilde{\boldsymbol{W}}_{t+1} = \left(\boldsymbol{W}_t^{-1} + \eta \hat{\boldsymbol{L}}_t \right)^{-1}$$

$$\boldsymbol{W}_{t+1} = \text{renormalize}\left(\widetilde{\boldsymbol{W}}_{t+1} \right)$$

Implementing the update

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W}\in\mathcal{S}} \left\{ \eta \left\langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \right\rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

= by the classic decomposition

$$\widetilde{\boldsymbol{W}}_{t+1} = \left(\boldsymbol{W}_t^{-1} + \eta \hat{\boldsymbol{L}}_t \right)^{-1}$$
$$\boldsymbol{W}_{t+1} = \text{renormalize}\left(\widetilde{\boldsymbol{W}}_{t+1} \right)$$

takes $\mathcal{O}(d^3)$ time in general...

Online Mirror Descent for bandit PCA

loss estimate $\hat{\boldsymbol{L}}_t$ =?

divergence D =?

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W}\in\mathcal{S}} \left\{ \eta \langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

+

sampling scheme?

Sample $\boldsymbol{w}_t$ so that $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$

Online Mirror Descent for bandit PCA

loss estimate $\hat{\boldsymbol{L}}_t =?$

divergence $D =?$

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W} \in \mathcal{S}} \{ \eta \langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \}$$

+

sampling scheme?

Sample $\boldsymbol{w}_t$ so that $\mathbb{E}[\boldsymbol{w}_t \boldsymbol{w}_t^\top] = \boldsymbol{W}_t$

Main course

Algorithm

Main results

Regret bound

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W}\in\mathcal{S}}\left\{\eta\left\langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t\right\rangle + D(\boldsymbol{W}\|\boldsymbol{W}_t)\right\}$$

Regret bound

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W} \in \mathcal{S}} \left\{ \eta \langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

Mirror descent regret bound:

$$\text{regret}_T \leq \frac{D(\boldsymbol{U}^* \| \boldsymbol{W}_1)}{\eta} + \sum_{t=1}^{T} \mathbb{E}\left[\langle \boldsymbol{W}_t - \widetilde{\boldsymbol{W}}_t, \hat{\boldsymbol{L}}_t \rangle\right]$$

Regret bound

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W} \in \mathcal{S}} \left\{ \eta \langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

Mirror descent regret bound:

$$\text{regret}_T \leq \frac{D(\boldsymbol{U}^* \| \boldsymbol{W}_1)}{\eta} + \sum_{t=1}^{T} \eta \mathbb{E}\left[\langle \boldsymbol{W}_t^2, \hat{\boldsymbol{L}}_t^2 \rangle \right]$$

Regret bound

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W} \in \mathcal{S}} \left\{ \eta \langle \boldsymbol{W}, \hat{\boldsymbol{L}}_t \rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t) \right\}$$

Mirror descent regret bound:

$$\text{regret}_T \leq \frac{D(\boldsymbol{U}^* \| \boldsymbol{W}_1)}{\eta} + \sum_{t=1}^{T} \eta \mathbb{E}\left[\langle \boldsymbol{W}_t^2, \hat{\boldsymbol{L}}_t^2 \rangle \right]$$

Regret bound

$$W_{t+1} = \arg\min_{W \in \mathcal{S}} \{ \eta \langle W, \hat{L}_t \rangle + D(W \| W_t) \}$$

Mirror descent regret bound:

$$\text{regret}_T \leq \frac{D(U^* \| W_1)}{\eta} + \sum_{t=1}^{T} \eta \mathbb{E}\left[\langle W_t^2, \hat{L}_t^2 \rangle \right]$$

Theorem

$$\text{regret}_T \leq \frac{d \log T}{\eta} + \eta d \sum_{t=1}^{T} \| L_t \|_F^2$$

Regret bound

$$\boldsymbol{W}_{t+1} = \arg\min_{\boldsymbol{W}\in\mathcal{S}} \{\eta\langle \boldsymbol{W}, \widehat{\boldsymbol{L}}_t\rangle + D(\boldsymbol{W} \| \boldsymbol{W}_t)\}$$

Mirror descent regret bound:

$$\text{regret}_T \leq \frac{D(\boldsymbol{U}^* \| \boldsymbol{W}_1)}{\eta} + \sum_{t=1}^{T} \eta\mathbb{E}\left[\langle \boldsymbol{W}_t^2, \widehat{\boldsymbol{L}}_t^2\rangle\right]$$

Corollary: let $r = \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{L}_t\|_F^2$ and $\eta = \sqrt{\log T / rT}$, then

$$\text{regret}_T = \mathcal{O}\left(d\sqrt{rT\log T}\right)$$

Regret bounds: TL;DR

For rank-1 losses:

$$\text{regret}_T = \mathcal{O}\left(d\sqrt{T\log T}\right)$$

In general:

$$\text{regret}_T = \mathcal{O}\left(d^{3/2}\sqrt{T\log T}\right)$$

Regret bounds: **TL;DR**

For rank-1 losses:

$$\text{regret}_T = \mathcal{O}\left(d\sqrt{T\log T}\right)$$

In general:

$$\text{regret}_T = \mathcal{O}\left(d^{3/2}\sqrt{T\log T}\right)$$

+ another sampling & estimation method that gives

$$\text{regret}_T = \tilde{\mathcal{O}}\left(d^{3/2}\sqrt{L_T^*\log T} + d^3\log T\right)$$

with L_T^* being the total loss of the best projection

Lower bound on the regret

Theorem

There is a problem instance on which any algorithm will suffer

$$\text{regret}_T = \Omega\left(d\sqrt{T/\log T}\right)$$

Proof ideas from

Auer, Cesa-Bianchi, Freund, Schapire (2002)

Cohen, Hazan, Koren (2017)

Lower bound on the regret

Theorem

There is a problem instance on which any algorithm will suffer

$$\text{regret}_T = \Omega\left(d\sqrt{T/\log T}\right)$$

Corollary

Bandit PCA is strictly harder than the multi-armed bandit problem where the minimax regret is $\Theta\left(\sqrt{dT}\right)$

Dessert | Fast implementation

Runtime

$$\widetilde{\boldsymbol{W}}_{t+1} = \left(\boldsymbol{W}_t^{-1} + \eta \hat{\boldsymbol{L}}_t\right)^{-1}$$
$$\boldsymbol{W}_{t+1} = \mathrm{renormalize}\left(\widetilde{\boldsymbol{W}}_{t+1}\right)$$

takes $\mathcal{O}(d^3)$ time in general…

Runtime

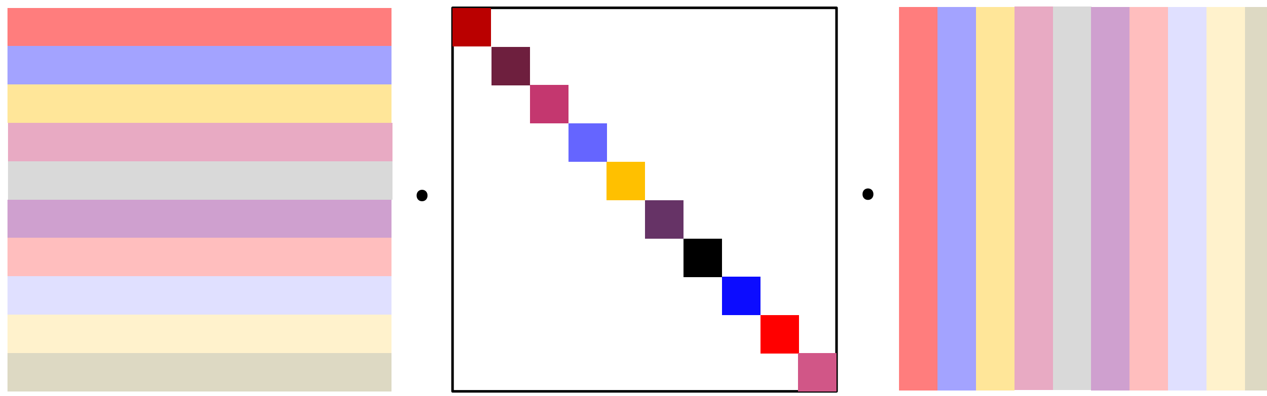
$$\widetilde{\boldsymbol{W}}_{t+1} = \left(\boldsymbol{W}_t^{-1} + \eta \hat{\boldsymbol{L}}_t\right)^{-1}$$
$$\boldsymbol{W}_{t+1} = \text{renormalize}\left(\widetilde{\boldsymbol{W}}_{t+1}\right)$$

takes $\mathcal{O}(d^3)$ time in general…

BUT ONLY
$\mathcal{O}(d)$ TIME
IN OUR CASE!!

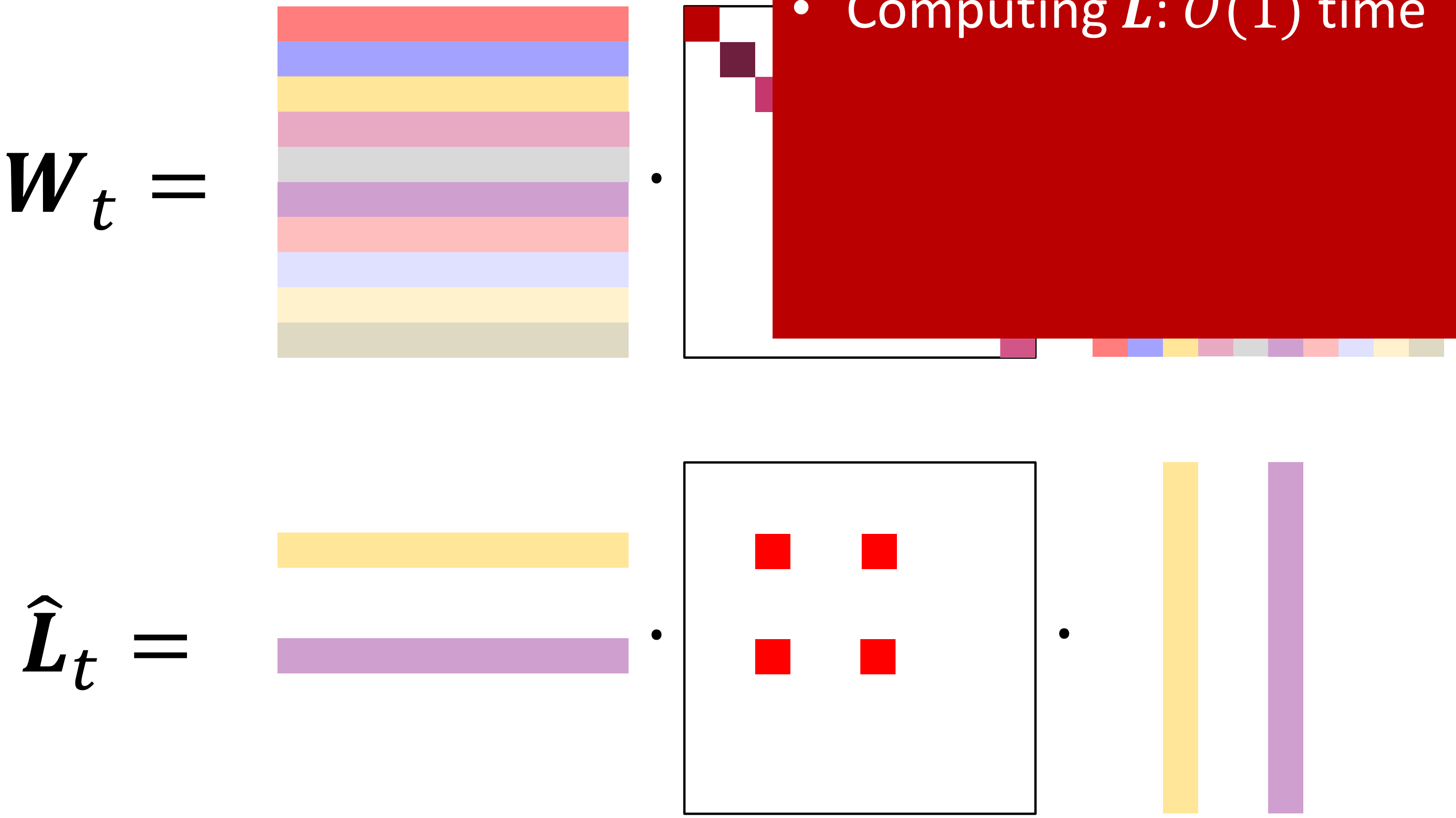
Updating in $\mathcal{O}(d)$ time

$\boldsymbol{W}_t =$



Updating in $\mathcal{O}(d)$ time

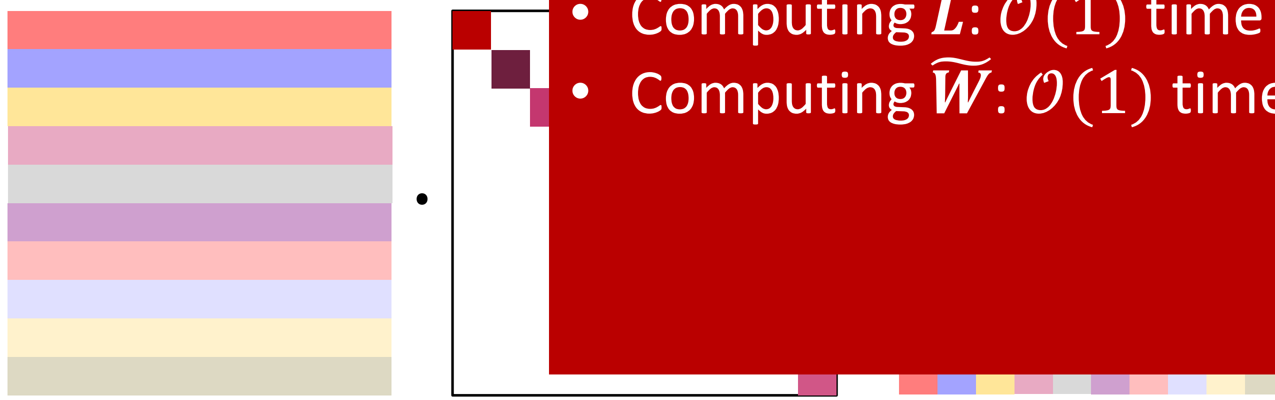
- Computing $\hat{\boldsymbol{L}}$: $\mathcal{O}(1)$ time



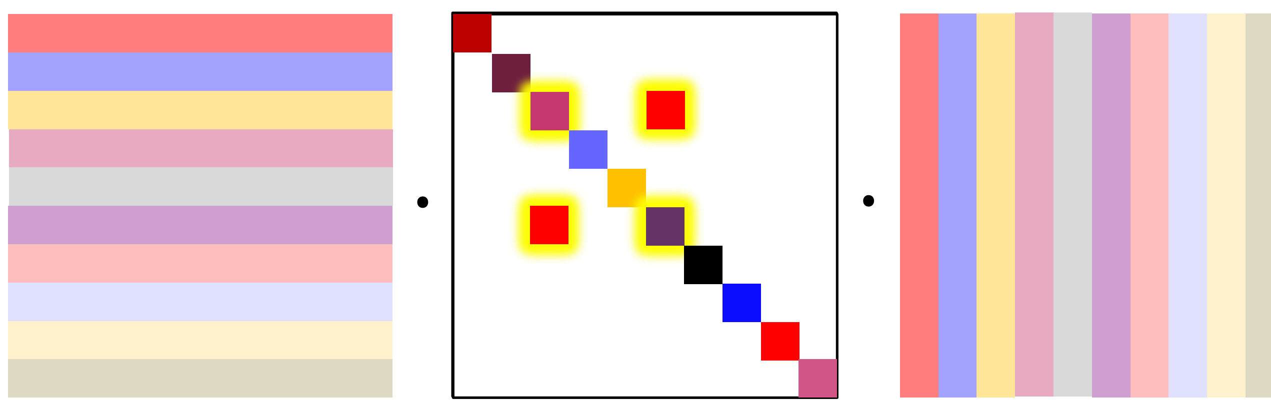
Updating in $\mathcal{O}(d)$ time

- Computing $\widehat{\boldsymbol{L}}$: $\mathcal{O}(1)$ time
- Computing $\widetilde{\boldsymbol{W}}$: $\mathcal{O}(1)$ time

$\boldsymbol{W}_t =$



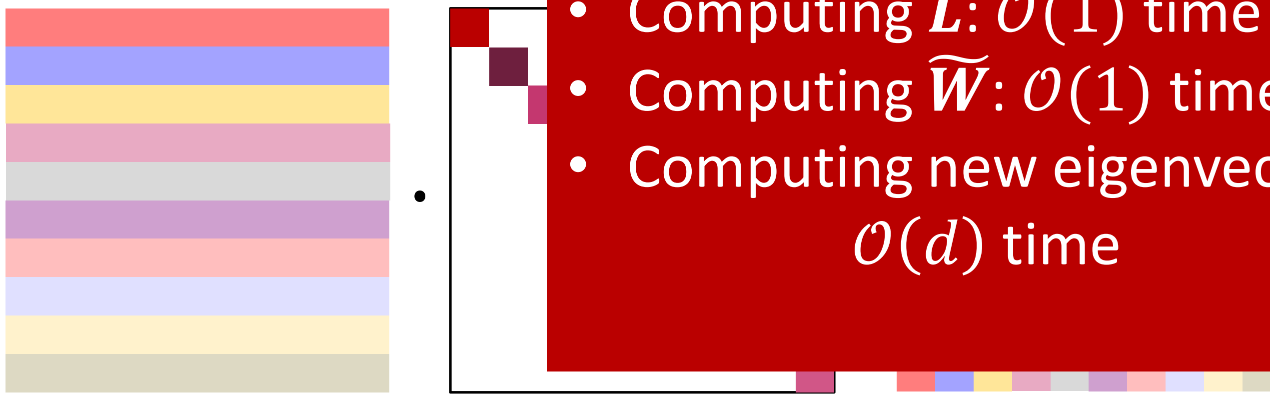
$\widetilde{\boldsymbol{W}}_{t+1} =$



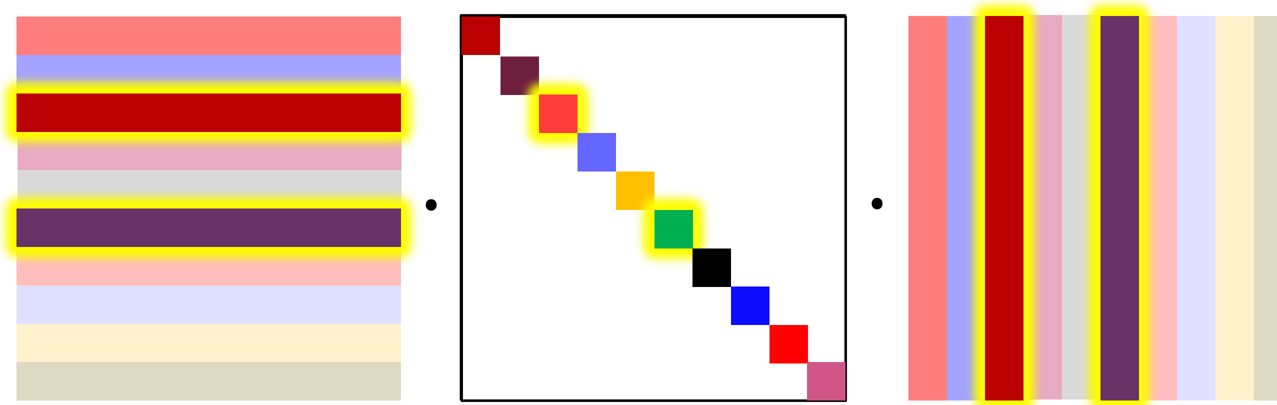
Updating in $\mathcal{O}(d)$ time

- Computing $\widehat{\boldsymbol{L}}$: $\mathcal{O}(1)$ time
- Computing $\widetilde{\boldsymbol{W}}$: $\mathcal{O}(1)$ time
- Computing new eigenvectors: $\mathcal{O}(d)$ time

$\boldsymbol{W}_t =$



$\widetilde{\boldsymbol{W}}_{t+1} =$

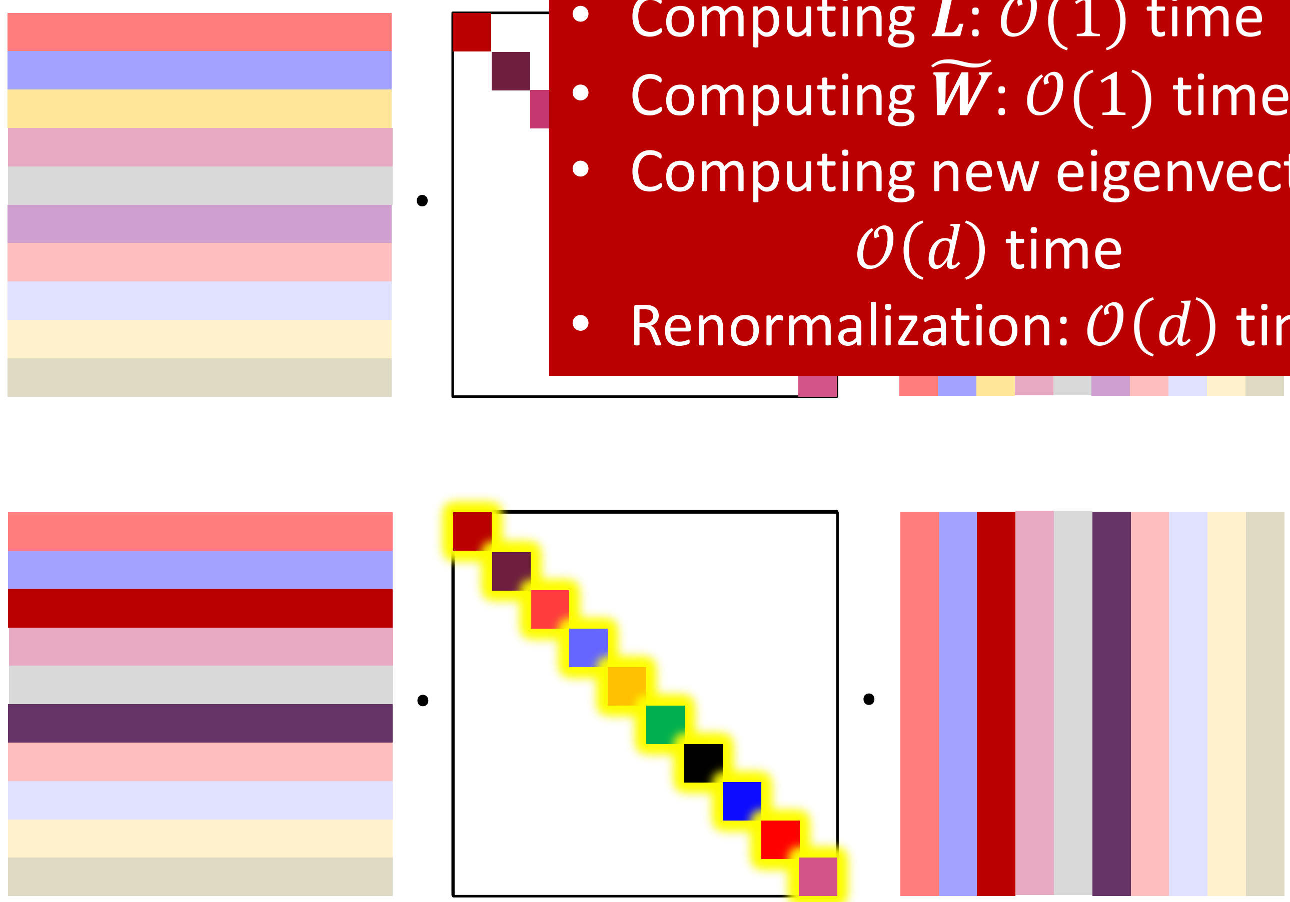


Updating in $\mathcal{O}(d)$ time

- Computing $\widehat{\boldsymbol{L}}$: $\mathcal{O}(1)$ time
- Computing $\widetilde{\boldsymbol{W}}$: $\mathcal{O}(1)$ time
- Computing new eigenvectors: $\mathcal{O}(d)$ time
- Renormalization: $\mathcal{O}(d)$ time

$\boldsymbol{W}_t =$

$\widetilde{\boldsymbol{W}}_{t+1} =$



Updating in $\mathcal{O}(d)$ time

1) time
(1) time
eigenvectors:
ne
$\mathcal{O}(d)$ time

$W_t =$

UPDATING TAKES LESS TIME THAN READING THE FULL LOSS MATRIX L_t!!!

$\widetilde{W}_{t+1} =$

Summary & open problems

	Previous best	Our work
Runtime	no polytime?	d
Upper bound	$d^2\sqrt{T}$	$d^{3/2}\sqrt{T}$
Lower bound	$\sqrt{dT}$	$d\sqrt{T}$

Summary & open problems

	Previous best	Our work
Runtime	no polytime?	d
Upper bound		$d^{3/2}\sqrt{T}$
Lower bound		$d\sqrt{T}$

Still a gap of $\sqrt{d}$

☹☹☹☹

Summary & open problems

	Previous best	Our work
Runtime	no polytime?	d
Upper bound		$d^{3/2}\sqrt{T}$
Lower bound		$d\sqrt{T}$

Still a gap of $\sqrt{d}$
☹☹☹☹

+ possible improvements:

- high-probability bounds
- better bounds for i.i.d. data
- …

Summary & open problems

	Previous best	Our work
Runtime	no polytime?	d
Upper bound		$d^{3/2}\sqrt{T}$
Lower bound		$d\sqrt{T}$

Still a gap of $\sqrt{d}$
☹☹☹☹

+ possible improvements:
- high-probability bounds
- better bounds for i.i.d. data
- …

New hammer for matrix prediction problems?

Thanks!