On-line PCA

Manfred Warmuth Dima Kuzmin

University of California - Santa Cruz

Aug 1, 2006 - MLSS Taipei
Last update - August 1, 2006

Outline

Outline

Predicting as well as the best expert

	E_1	E_2	E_3	$\cdots$	E_n	prediction	true label	loss
day 1	1	1	0	$\cdots$	0	0	1	1
day 2	1	0	1	$\cdots$	0	1	0	1
day 3	0	1	1	$\cdots$	1	1	1	0
day t	$x_{t,1}$	$x_{t,2}$	$x_{t,3}$	$\cdots$	$x_{t,n}$	$\hat{y}_t$	y_t	$\lvert y_t - \hat{y}_t \rvert$

Master Algorithm

For $t = 1$ To T Do

Get instance	$\mathbf{x}_t \in \{0,1\}^n$
Predict	$\hat{y}_t \in \{0,1\}$
Get label	$y_t \in \{0,1\}$
Incur loss	$\lvert y_t - \hat{y}_t \rvert$

Goal of expert setting

$$\underbrace{\sum_{t=1}^{T} |y_t - \hat{y}_t|}_{\text{loss}_{\text{alg}}} \quad \sim \quad \underbrace{\inf_i \sum_{t=1}^{T} |y_t - x_{t,i}|}_{\text{loss}_{\text{best}}}$$

- Master maintains probability vector $\boldsymbol{\omega}_t$ on experts
  Simple prediction: $\hat{y}_t = \boldsymbol{\omega}_t \cdot \mathbf{x}_t$

$$\underbrace{|y_t - \boldsymbol{\omega}_t \cdot \mathbf{x}_t|}_{\text{loss of alg}} = \underbrace{\sum_i \omega_{t,i} \overbrace{|y_t - x_{t,i}|}^{\lambda_{t,i}}}_{\text{expected loss of experts}}$$

Dot loss [FS]

Learning on-line

- Pick expert i based on probability vector $\boldsymbol{\omega}_t$
- Receive loss vector $\boldsymbol{\lambda}_t$
- Incurr loss $\lambda_{t,i}$ and expected loss $\boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t$
- Update $\boldsymbol{\omega}_t$

Goal

$$\text{loss}_{\text{alg}} = \sum_t \boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t \quad \sim \quad \text{loss}_{\text{best}} = \inf_i \sum_t \lambda_{t,i}$$

The works [LW,FS]

- Maintain probability vector

$$\omega_{t+1,i} = \frac{\omega_{t,i} e^{-\eta \lambda_{t,i}}}{Z_t}$$

- Motivation

$$\boldsymbol{\omega}_{t+1} = \inf_{\boldsymbol{\omega} \in \text{simplex}} \overbrace{\sum_i \omega_i \ln \frac{\omega_i}{\omega_{t,i}}}^{\Delta(\boldsymbol{\omega}, \boldsymbol{\omega}_t)} + \eta \, \boldsymbol{\omega} \cdot \boldsymbol{\lambda}_t$$

- Bound

$$\boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t \leq \frac{\eta \, \boldsymbol{v} \cdot \boldsymbol{\lambda}_t + \Delta(\boldsymbol{v}, \boldsymbol{\omega}_{t+1}) - \Delta(\boldsymbol{v}, \boldsymbol{\omega}_t)}{1 - e^{-\eta}}$$

Total loss bound

- Summing over trials

$$\overbrace{\sum_{t=1}^{T} \boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t}^{\text{loss}_{\text{alg}}} \leq \frac{\eta \overbrace{\sum_{t=1}^{T} \boldsymbol{\upsilon} \cdot \boldsymbol{\lambda}_t}^{\text{loss}_{\boldsymbol{\upsilon}}} + \overbrace{\Delta(\boldsymbol{\upsilon}, \boldsymbol{\omega}_{T+1}) - \Delta(\boldsymbol{\upsilon}, \boldsymbol{\omega}_1)}^{\leq \log n}}{1 - e^{-\eta}}$$

- Tuning η

$$\text{loss}_{\text{alg}} \leq \text{loss}_{\text{best}} + \sqrt{2 \text{loss}_{\text{best}} \log n} + \log n$$

Alternates to the softmin

Form of weights [V]

$$\omega_i \sim e^{-\eta \, \text{loss}_i}$$

As $\eta \to \infty$ all weight placed on min. loss expert

Follow the perturbed leader [KV]

- Add random perturbation to total loss of each expert
- Choose expert with minimum perturbed loss

Deterministic algorithm off by factor of at least two

Outline

Variance

- So far, small expected loss: $\boldsymbol{\omega} \cdot \boldsymbol{\lambda}$
- Now, symmetric positive definite matrix $\mathbf{C}$ is covariance matrix of some random vector $\mathbf{p} \in \mathbb{R}^n$

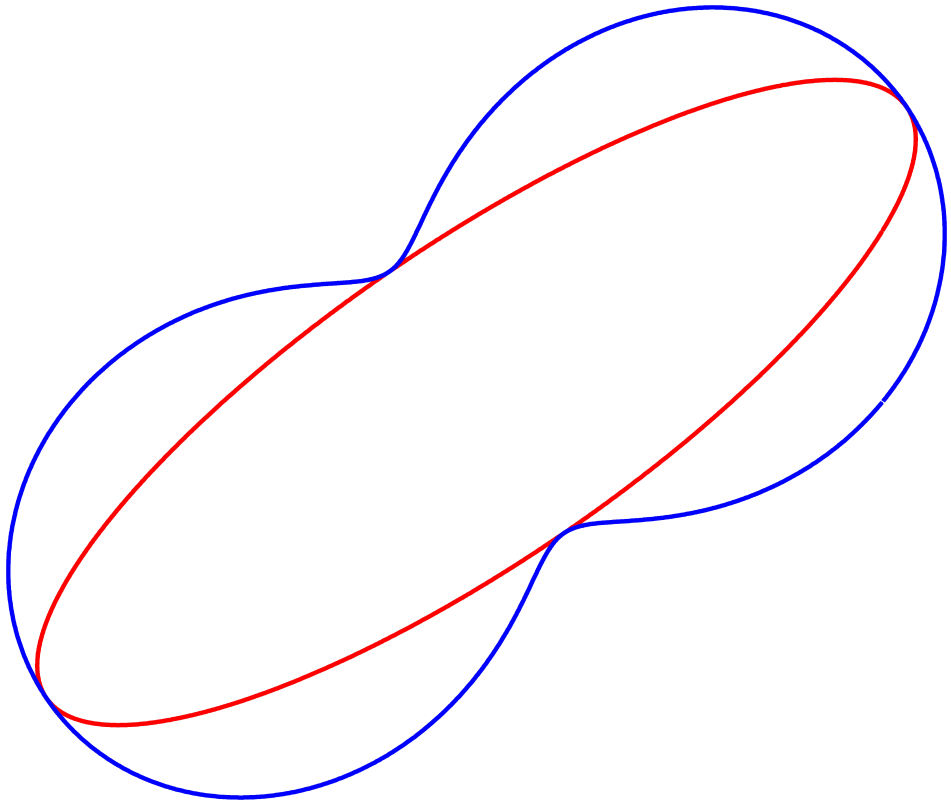
$$\mathbf{C} = \mathbb{E}\left((\mathbf{p} - \mathbb{E}(\mathbf{p})(\mathbf{p} - \mathbb{E}(\mathbf{p}))^\top\right)$$

- The variance along any vector $\mathbf{u}$ is

$$\begin{aligned}
\mathbb{V}(\mathbf{p}^\top \mathbf{u}) &= \mathbb{E}(\left(\mathbf{p}^\top \mathbf{u} - \mathbb{E}(\mathbf{p}^\top \mathbf{u})\right)^2) \\
&= \mathbb{E}(\left((\mathbf{p}^\top - \mathbb{E}(\mathbf{p}^\top))\,\mathbf{u}\right)^2) \\
&= \mathbf{u}^\top \mathbb{E}\left((\mathbf{p} - \mathbb{E}(\mathbf{p})(\mathbf{p} - \mathbb{E}(\mathbf{p}))^\top\right)\mathbf{u}
\end{aligned}$$

Outline

Variance of unit vectors



The ellipse is plot of vector $\mathbf{Cu}$ for unit vector $\mathbf{u}$

The outer figure eight is direction $\mathbf{u}$ times the variance $\mathbf{u}^\top \mathbf{Cu}$

At eigenvectors variance touches ellipse

Variance minimization problem

On-line learning problem

- Pick a vector unit vector $\mathbf{w}_t$
- Receive a covariance matrix $\mathbf{C}_t$
- Loss is variance along vector $\mathbf{w}_t$

$$\mathbf{w}_t^\top \mathbf{C}_t \mathbf{w}_t$$

Goal: Achieve variance close to variance of shortest axis picked in hindsight

$$L_{\text{best}} = \inf_{\mathbf{u}} \ \mathbf{u}^\top \left(\sum_t \mathbf{C}_t \right) \mathbf{u}$$

Mixtures of directions/dyads = density matrix

- Unit ball not a convex set
- Our algorithm will pick a direction $\mathbf{w}_i$ with probability ω_i
- Expected variance

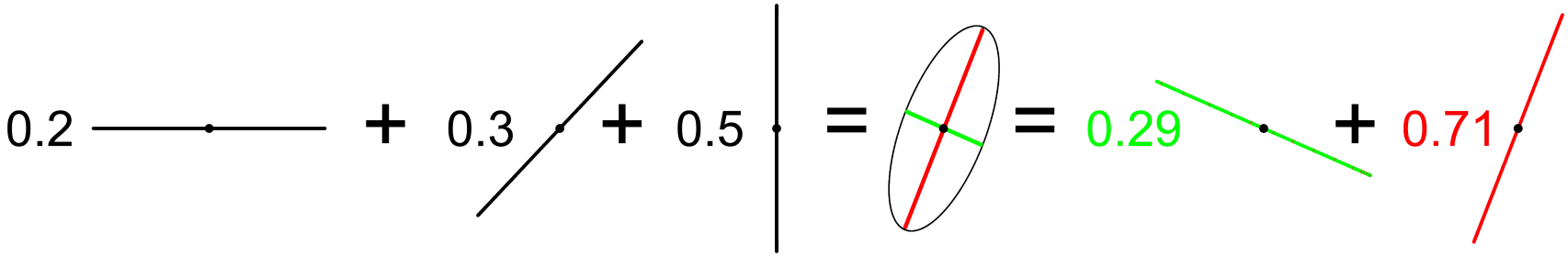
$$\underbrace{\sum_i \omega_i \overbrace{\mathbf{w}_i^\top \mathbf{C}\, \mathbf{w}_i}^{\text{var.in.dir.}\mathbf{w}_i}}_{\text{expected variance}} = \sum_i \omega_i \operatorname{tr}(\mathbf{C}\, \mathbf{w}_i \mathbf{w}_i^\top) = \operatorname{tr}(\mathbf{C} \underbrace{\sum_i \omega_i\, \mathbf{w}_i \mathbf{w}_i^\top}_{\text{density matrix } \mathbf{W}})$$

$\mathbf{w}\mathbf{w}^\top$ for unit $\mathbf{w}$ is called a **dyad**

- Symmetric positive definite matrix of rank one
- Trace one: $\operatorname{tr}(\mathbf{w}\mathbf{w}^\top) = \mathbf{w}^\top \mathbf{w} = \|\mathbf{w}\|_2^2 = 1$
- Projection matrix onto direction $\mathbf{w}$

Density matrices

- Convex combinations of dyads
- Symmetric positive definite matrices of trace one
- Eigenvalues form probability vector
- Many mixtures lead to the same matrix:



- Can always be written as a convex combination of n dyads corresponding to eigenvectors

Diagonal case: $\sum_i \omega_i \, \mathbf{e}_i \mathbf{e}_i^\top$

Variance minimization with density matrices

Setup

- Parameter: density matrix $\mathbf{W}_t = \sum_i \omega_{t,i}\, \mathbf{w}_{t,i}\mathbf{w}_{t,i}^\top$
- Pick direction $\mathbf{w}_{t,i}$ with probability $\omega_{t,i}$
- Covariance matrix $\mathbf{C}_t$ is obtained
- Incur variance $\mathbf{w}_t^\top \mathbf{C}_t \mathbf{w}_t$ and expected variance

$$\sum_i \omega_{t,i}\, \mathbf{w}_{t,i}^\top \mathbf{C}_t \mathbf{w}_{t,i} = \mathrm{tr}(\mathbf{W}_t \mathbf{C}_t)$$

- Update $\mathbf{W}_t$

Goal: Do as well as best density matrix
- single dyad corresponding to smallest eigenvalue of $\sum_t \mathbf{C}_t$

Expert setting retained as diagonal case

$$\omega_t \cdot \boldsymbol{\lambda}_t = \mathrm{tr}\Big(\underbrace{\begin{pmatrix} \omega_{t,1} & 0 & 0 & 0 \\ 0 & \omega_{t,2} & 0 & 0 \\ 0 & 0 & \omega_{t,3} & 0 \\ 0 & 0 & 0 & \omega_{t,4} \end{pmatrix}}_{\text{diagonal } \mathbf{W}_t} \underbrace{\begin{pmatrix} \lambda_{t,1} & 0 & 0 & 0 \\ 0 & \lambda_{t,2} & 0 & 0 \\ 0 & 0 & \lambda_{t,3} & 0 \\ 0 & 0 & 0 & \lambda_{t,4} \end{pmatrix}}_{\text{diagonal } \mathbf{C}_t} \Big)$$

Previous setup

- Pick expert i based on probability vector $\boldsymbol{\omega}_t$
- Receive loss vector $\boldsymbol{\lambda}_t$
- Incurr loss $\lambda_{t,i}$ and expected loss $\boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t$
- Update $\boldsymbol{\omega}_t$

Expert i corresponds to dyad $\mathbf{e}_i \mathbf{e}_i^\top$

In matrix setting continuously many dyads $\mathbf{w}\mathbf{w}^\top$

Deriving the algorithm

$$\mathbf{W}_{t+1} = \underset{\mathbf{U}\ \text{dens. mat.}}{\arg\inf} \overbrace{\underbrace{\operatorname{tr}(\mathbf{U}(\log \mathbf{U} - \log \mathbf{W}_t))}_{\text{quantum relative entropy}}}^{\Delta(\mathbf{U},\mathbf{W}_t)} + \eta \underbrace{\operatorname{tr}(\mathbf{U}\mathbf{C}_t)}_{\text{expected variance}}$$

$$\mathbf{W}_{t+1} = \frac{\overbrace{\exp(\overbrace{\log \mathbf{W}_t}^{\text{symmetric}} - \eta \overbrace{\mathbf{C}_t}^{\text{symmetric}})}^{\text{symmetric positive definite}}}{\operatorname{tr}(\exp(\log \mathbf{W}_t - \eta \mathbf{C}_t))}$$

log, **exp** are matrix versions of logarithm and exponential [TRW]

Bound generalizes

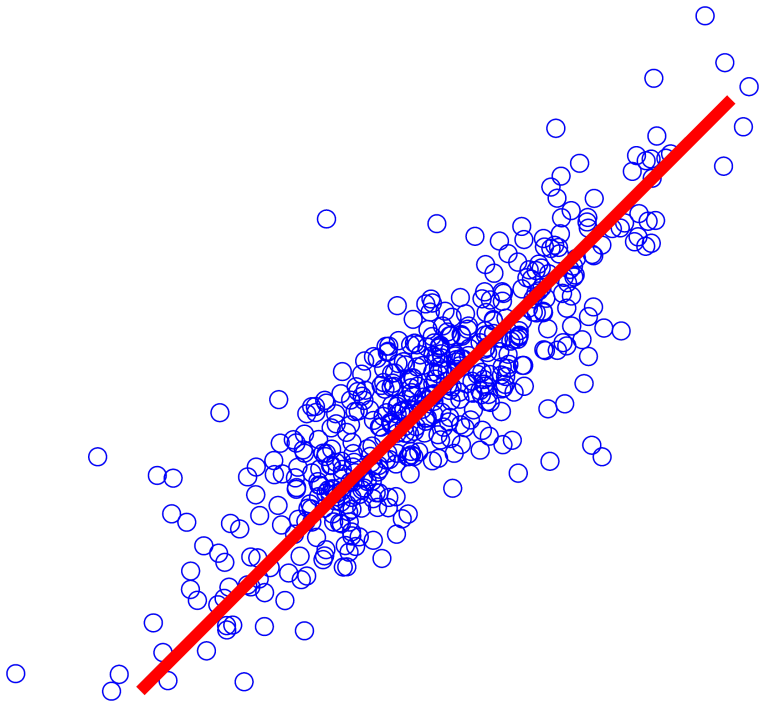
$$\mathrm{tr}(\mathbf{W}_t\mathbf{C}_t) \;\leq\; \frac{\eta\,\mathrm{tr}(\mathbf{U}\mathbf{C}_t) + \Delta(\mathbf{U},\mathbf{W}_{t+1}) - \Delta(\mathbf{U},\mathbf{W}_t)}{1-e^{-\eta}}$$

$$\mathrm{loss}_{\mathrm{alg}} \leq \mathrm{loss}_{\mathrm{best}} + \sqrt{2\mathrm{loss}_{best}\log n} + \log n$$

Assumption: max. eigenvalue of $\mathbf{C}_t \;\leq 1$

Outline

PCA



- On-line projection of data into low-dimensional subspace
- Best subspace in hindsight: k top eigenvectors of data covariance matrix

Rewrite quadratic loss as linear loss

Want rank k projection matrix $\mathbf{P}$ that minimizes total square loss

$$||\underbrace{\mathbf{P}}_{k}\mathbf{x} - \mathbf{x}||_2^2 = ||\mathbf{P}\mathbf{x} - P\mathbf{x} - (\mathbf{I} - \mathbf{P})\mathbf{x}||_2^2 = \mathrm{tr}(\underbrace{(\mathbf{I} - \mathbf{P})}_{n-k}\ \underbrace{\mathbf{x}\mathbf{x}^\top}_{C})$$

Want to choose $n - k$ dimensional subspace of minimum variance

Projection matrices $\mathbf{P}$ are symmetric positive matrices with eigenvalues in $\{0, 1\}$: $\mathbf{P}^2 = \mathbf{P}$

So far

- Variance of alg. close to variance of smallest axis chosen in hindsight
- Minimizing variance along one directions equivalent to maximizing variance along remaining $n-1$ directions
- For PCA: Maximize variance along k directions or minimize variance along $n-k$ directions
- Idea: Do it first in expert setting

Minimizing loss of $m = n - k$ experts

- Pick set of m experts $\{i_1, \ldots, i_m\}$ based on probability vector $\boldsymbol{\omega}_t$
- Receive loss vector $\boldsymbol{\lambda}_t$
- Loss is total loss of the m experts $\lambda_{i_1} + \ldots + \lambda_{i_m}$
  and expected loss $m\, \boldsymbol{\omega}_t \cdot \mathbf{x}_t$
- Update $\boldsymbol{\omega}_t$

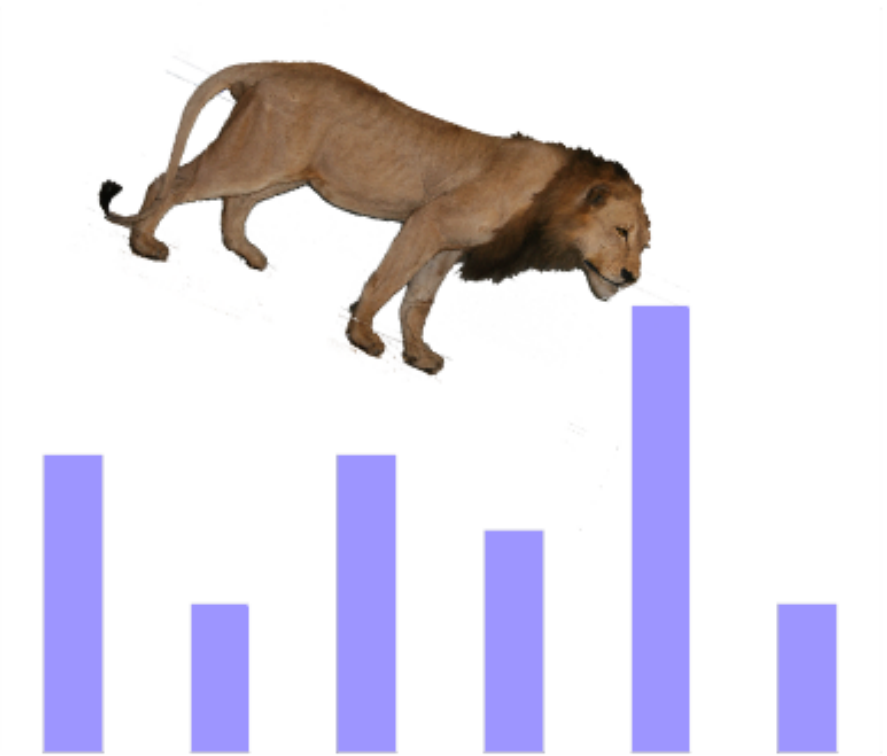
Goal: Total (expected) loss of alg. close to total loss of best expert

$$\text{loss}_{\text{alg}} = \sum_t \boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t \quad \sim \quad \inf_{\{i_1,\ldots,i_m\}} \sum_t \sum_j \lambda_{t,i_j}$$

Minimizing loss $\boldsymbol{\lambda}$ on m experts
equivalent to maximizing gain $-\boldsymbol{\lambda}$ on $n - m$ experts

New trick: cap weights

Super predator algorithm

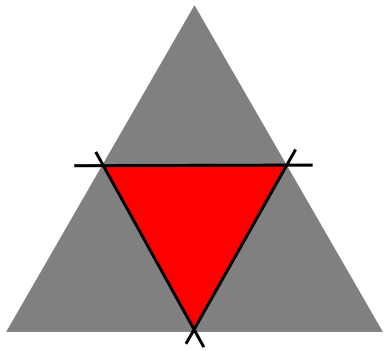


Preserves variety

Weights $\leq \frac{1}{m}$

$$\hat{\omega}_{t,i} = \frac{\omega_{t,i} e^{-\eta \lambda_{t,i}}}{Z}$$

$$\boldsymbol{\omega}_{t+1} = \inf_{\boldsymbol{\omega} \text{ in capped simplex}} \Delta(\boldsymbol{\omega}, \widehat{\boldsymbol{\omega}}_t)$$



expected loss of alg

$$\leq \text{loss of best } m \text{ set} + \sqrt{2 \text{loss of best } m \text{ set } m \log n} + m \log n$$

Why capping?

- m sets encoded as probability vectors $(0, \frac{1}{m}, 0, 0, \frac{1}{m}, 0, \frac{1}{m})$ called m-corners
- The convex hull of the m-corners $=$ capped probability simplex
- We can effectively decompose any capped probability vector $\boldsymbol{\omega}$ as convex combination of n m-corners

$$\boldsymbol{\omega} = \sum_j \alpha_j \mathbf{r}_j$$

- Choose m-corner $\mathbf{r}_j$ with probability α_j

Alternates to capping

- Follow the perturbed leader: cheap but inferior bounds
- Dynamic programming: more expensive

Lift sets of expert alg. to matrices

- Pick $n-k$ dimensional subspace based on density matrix $\underbrace{\mathbf{W}_t}_{n-k}$
- Choose complementary subspace $\underbrace{\mathbf{P}_t}_{k}$
- Receive instance $\mathbf{x}_t$
- Incur loss $||\mathbf{P}_t\mathbf{x}_t - \mathbf{x}_t||_2^2 = \mathrm{tr}(\underbrace{(\mathbf{I} - \mathbf{P}_t)}_{n-k}\, \mathbf{x}_t\mathbf{x}_t^\top)$

  and expected loss $(n-k)\,\mathrm{tr}(\mathbf{W}_t\, \mathbf{x}_t\mathbf{x}_t^\top)$
- Update $\underbrace{\mathbf{W}_t}_{n-k}$
  - Exponential update
  - Cap eigenvals to $\leq \frac{1}{n-k}$

Update and Winnow-like bound

$$\widehat{\mathbf{W}}_t = \frac{\exp(\log \mathbf{W}_t - \eta\, \mathbf{x}_t \mathbf{x}_t^\top)}{\mathrm{tr}(\exp(\log \mathbf{W}_t - \eta\, \mathbf{x}_t \mathbf{x}_t^\top))}$$

$$\mathbf{W}_{t+1} = \inf_{\substack{\mathbf{W} \text{ dens.matrix} \\ \text{w.eigenvals } \leq \frac{1}{n-k}}} \Delta(\mathbf{W}, \hat{\mathbf{W}}_t)$$

expected loss of alg

$$\leq \text{loss of best } k \text{ subspace} + \sqrt{2 \text{ loss of best } k \text{ subspace } k \log n} + k \log n$$

Two families again

Regularize with $||\mathbf{W} - \mathbf{W}_1||_2^2$ [C]

- $\mathbf{W} =$ lin. comb. of $\mathbf{x}_t\mathbf{x}_t^\top$
- Fast and kernelizable

Regularize with quantum relative entropy

- $\mathbf{W} = \frac{\mathbf{exp}(\text{lin. comb. of } \mathbf{x}_t\mathbf{x}_t^\top)}{Z}$
- Predict with random projection matrix
- Regret bounds instead of filtering loss

Key insight: Mixtures of experts generalize density matrices

Outline

What's next?

- Shifting methodology from expert setting carries over
- Experiments
- Survey on "The Blessing and Curse of the Multiplicative Updates"
  - Adapt quickly
  - Loss of variety
  - Connections to Biology
- Work out probability calculus for density matrices