On-line PCA

Manfred K. Warmuth   Dima Kuzmin

University of California - Santa Cruz

Nov 14, 2007

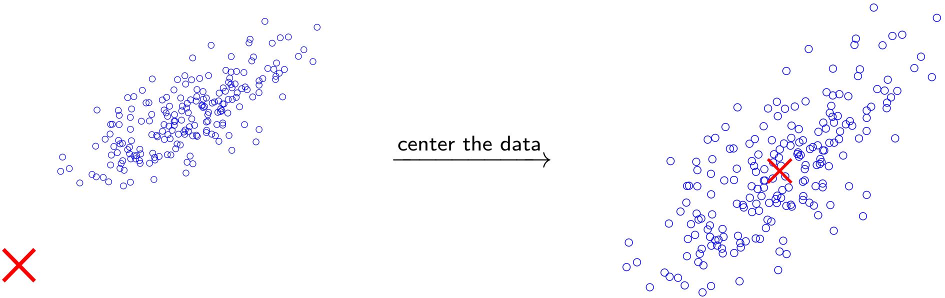
Outline

1. Batch PCA and why do we want to do it on-line
2. Expert setting
3. Variance minimization on the unit sphere
4. On-line PCA
5. What's next?

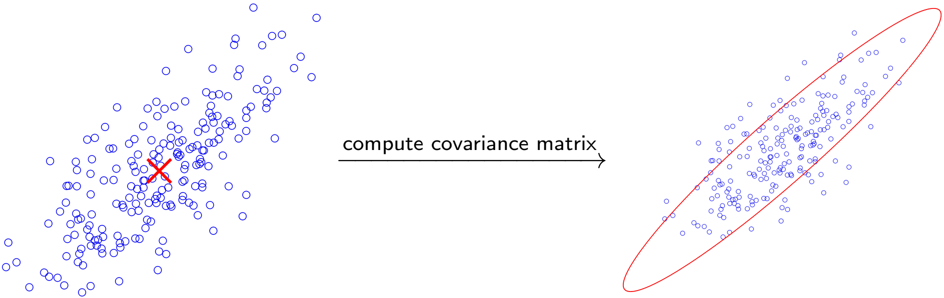
Outline

1. Batch PCA and why do we want to do it on-line
2. Expert setting
3. Variance minimization on the unit sphere
4. On-line PCA
5. What's next?

Step 1 of batch PCA



Step 2: summarize data



Final step: dimensionality reduction

Objective of batch PCA

$$\inf_{\text{center } \mathbf{c}} \quad \inf_{k\text{-dim. proj. matrix } \mathbf{P}} \sum_t || \underbrace{\mathbf{P}(\mathbf{x}_t - \mathbf{c})}_{\text{compressed}} - \underbrace{(\mathbf{x}_t - \mathbf{c})}_{\text{uncompressed}} ||_2^2$$

Solution:

$\mathbf{c}^* =$ average point

$\mathbf{P}^* =$ subspace spanned by k longest axes of covariance matrix $\sum_t (\mathbf{x}_t - \mathbf{c}^*)(\mathbf{x}_t - \mathbf{c}^*)^\top$

$\mathbf{x}_t, \mathbf{c}$	$n \times 1$
$\mathbf{P}$	$n \times n$
covariance matrix	$n \times n$

Why on-line?

- Data points produced on-line
- Data changes over time
- Want to exploit the sequential nature of the data

How do we do it?

- Lift methods from expert setting of on-line learning to matrix setting
- Before: probability vector expresses uncertainty over best expert
- Now: density matrix expresses uncertainty over best subspace

What do we gain?

- On-line algorithms for PCA that work provably well when data shifts with time
- Version of the algorithms that exploit shifts back to previously used subspaces
- New generalization of softmin/max
- The same bounds - matrix case comes for free
- Algorithms are expensive ???

Outline

1 Batch PCA and why do we want to do it on-line

2 Expert setting

3 Variance minimization on the unit sphere

4 On-line PCA

5 What's next?

Expert setting [FS]

Learning on-line

- Pick expert i based on probability vector $\boldsymbol{\omega}_t$
- Receive loss vector $\boldsymbol{\lambda}_t \in [0,1]^n$
- Incur loss $\lambda_{t,i}$ and expected loss $\boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t$
- Update $\boldsymbol{\omega}_t$

Goal

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

Example

Example:

- $\mathbf{w}_1 = (1/3, 1/3, 1/3)^\top$ - distribution on experts
- Pick an expert according to w_1, say $i = 2$
- Recieve losses for all experts $\boldsymbol{\lambda}_1 = (1, 0, 1)^\top$
- Incur loss $\lambda_{1,2} = 0$, expected loss is $\mathbf{w}_1 \cdot \boldsymbol{\lambda}_1 = 2/3$
- Update $\mathbf{w}_1$ to $\mathbf{w}_2$

Follow the leader

FL alg

- Maintain vector of total losses $\boldsymbol{\lambda}_{<t}$ of all experts
- At trial t choose minimum component of $\boldsymbol{\lambda}_{<t}$

Adversary can force

$$\text{loss}_{\text{FL}} \geq n \ \text{loss}_{\text{best}}$$

Its strategy for picking loss vectors

- Chosen expert (leader) incurs one unit of loss
- Rest incur no loss
- After T trials, $\text{loss}_{\text{FL}} = T$ and $\text{loss}_{\text{best}} = \lfloor \frac{T}{n} \rfloor$

Softmin with exponential weights [LW,FS]

- WMR: Choose expert based on probability vector

$$\omega_{t,i} = \frac{\overbrace{\omega_{1,i} e^{-\eta\, \lambda_{<t,i}}}^{\text{softmin}}}{Z_t} \qquad \omega_{t+1,i} = \frac{\omega_{t,i} e^{-\eta\, \lambda_{t,i}}}{Z'_t}$$

- Motivation

$$\boldsymbol{\omega}_{t+1} = \operatorname*{arg\,inf}_{\sum_i \omega_i = 1} \left(\overbrace{\sum_i \omega_i \ln \frac{\omega_i}{\omega_{t,i}}}^{\Delta(\boldsymbol{\omega}, \boldsymbol{\omega}_t)} + \eta\, \boldsymbol{\omega} \cdot \boldsymbol{\lambda}_t \right)$$

- Bound

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

Total loss bound

- Summing over trials

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

- Tuning η

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

Expert algorithms

- Deterministic FL alg

$$\text{loss}_{\text{FL}} \geq n \, \text{loss}_{\text{best}}$$

- WMR with softmin weights

$$\omega_{t,i} \sim e^{-\eta \, \lambda_{<t,i}}$$

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

$$\text{loss}_{\text{WMR}} \quad \leq \quad \text{loss}_{\text{best}} + \text{lower order term}$$

- Follow the perturbed leader (FPL) [KV]
  - Add random perturbation to total loss $\boldsymbol{\lambda}_{<t}$
  - Choose expert with minimum perturbed loss

Overview

comparator	batch	on-line
best single expert	min	softmin w. exponential weights or FPT
best direction PCA w. $k = n - 1$	min eigenvalue	softmin eigenvalue w. matrix exponentials
PCA w. $k < n - 1$	bottom segment of $n - k$ eigenvalues	softmin eigenvalue w. matrix exponentials and projections

Outline

1 Batch PCA and why do we want to do it on-line

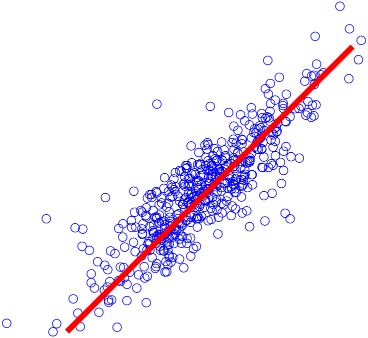
2 Expert setting

3 Variance minimization on the unit sphere

4 On-line PCA

5 What's next?

Finding direction of largest variance on-line



What set of experts and loss?

	set of experts	summary of losses
so far	n experts	n dimensional vector $\boldsymbol{\lambda}$
	loss of expert i	λ_i
new	unit ball in n dimensions	symmetric positive definite matrix $\mathbf{C}$
	loss/cost of direction $\mathbf{u}$	$\mathbf{u}^T\mathbf{C}\mathbf{u}$

Variance interpretation

- Interpret **C** as covariance matrix of some random vector $\mathbf{x} \in \mathbb{R}^n$

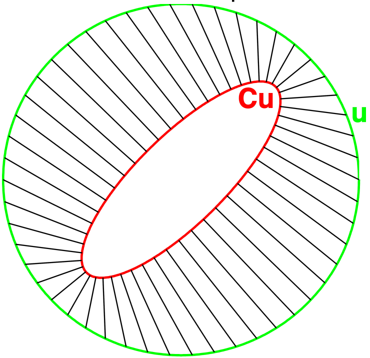
$$\mathbf{C} = \mathbb{E}\left((\mathbf{x} - \mathbb{E}(\mathbf{x})(\mathbf{x} - \mathbb{E}(\mathbf{x}))^\top\right)$$

- The variance along any vector **u** is

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

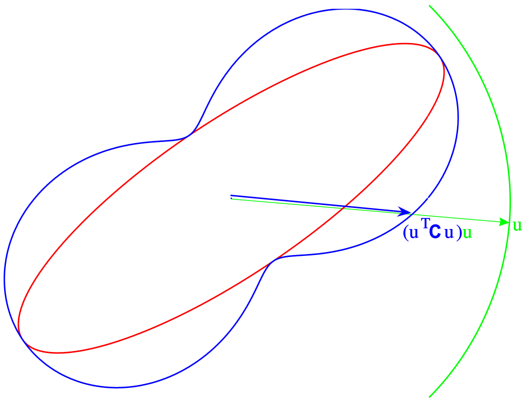
Ellipses

- We illustrate symmetric matrices as ellipses
  - affine transformations of the unit sphere:



- Ellipse $= \{\mathbf{Cu} \; : \; \|\mathbf{u}\|_2 = 1\}$
- Dotted lines connect point **u** on unit sphere with point **Cu** on ellipse

Variance of unit vectors



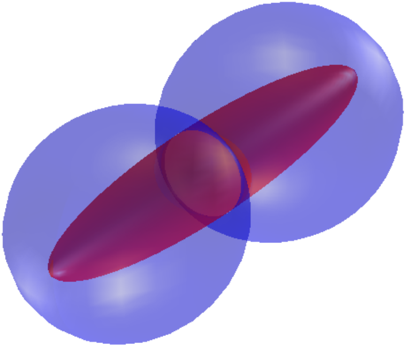
The ellipse is plot of vector $\mathbf{Cu}$ for unit vector $\mathbf{u}$

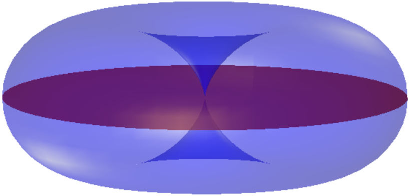
The outer figure eight is variance $\mathbf{u}^\top \mathbf{Cu}$ times direction $\mathbf{u}$

At eigenvectors variance touches ellipse

For uncentered PCA, $\mathbf{C} = \mathbf{x}\mathbf{x}^\top$. In this case $\mathbf{u}^\top \mathbf{x}\mathbf{x}^\top = (\mathbf{u} \cdot \mathbf{x})^2$

3 dimensional variance plots





Variance minimization problem

On-line learning problem

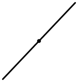
- Pick a vector unit vector $\mathbf{w}_t$
- Receive a covariance matrix $\mathbf{C}_t$
- Loss is variance along vector $\mathbf{w}_t$

$$\mathbf{w}_t^\top \mathbf{C}_t \mathbf{w}_t = \mathrm{tr}(\mathbf{C}_t \mathbf{w}_t \mathbf{w}_t^\top)$$

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

$$\begin{aligned} L_{\text{best}} &= \inf_{\mathbf{u}} \mathbf{u}^\top \left(\sum_t \mathbf{C}_t \right) \mathbf{u} \\ &= \mathrm{tr}\left(\left(\sum_t \mathbf{C}_t \right) \mathbf{u}\mathbf{u}^\top \right) \end{aligned}$$

Mixtures of directions/dyads = density matrix

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

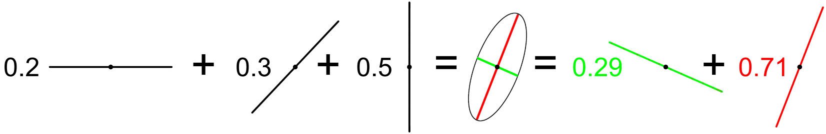
- Symmetric positive definite matrix of rank one
- Trace one: $\operatorname{tr}(\mathbf{w}\mathbf{w}^\top) = \mathbf{w}^\top\mathbf{w} = \|\mathbf{w}\|_2^2 = 1$
- Projection matrix onto direction $\mathbf{w}$
- Dyads as experts instead of directions

Algorithm maintains mixture of dyads

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

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 be written as convex combination (not unique) of n eigendyads

Diagonal case

- $\sum_i \omega_i \, \mathbf{e}_i \mathbf{e}_i^\top$
- Fixed eigensystem - decomposition unique

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

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

Previous setup

- Pick expert i based on probability vector $\boldsymbol{\omega}_t$
- Receive loss vector $\boldsymbol{\lambda}_t$
- Incur 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

$$\mathbf{W}_t = \overbrace{\frac{\exp(\log \mathbf{W}_1 - \eta \mathbf{C}_{<t})}{\mathrm{tr}(\exp(\log \mathbf{W}_1 - \eta \mathbf{C}_{<t}))}}^{\text{softmin}} \qquad \mathbf{W}_{t+1} = \frac{\overbrace{\exp(\overbrace{\log \mathbf{W}_t}^{\text{symmetric}} - \eta \overbrace{\mathbf{C}_t}^{\text{symmetric}})}^{\text{symmetric positive definite}}}{\mathrm{tr}(\exp(\log \mathbf{W}_t - \eta \mathbf{C}_t))}$$

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

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

Bound generalizes

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

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

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

Outline

1. Batch PCA and why do we want to do it on-line
2. Expert setting
3. Variance minimization on the unit sphere
4. On-line PCA
5. What's next?

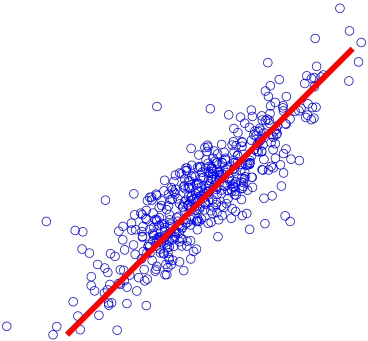
Recall batch Case

- Data sample $\mathbf{x}_1, \ldots, \mathbf{x}_m$ from $\mathbb{R}^n$
- Pick rank k projection matrix $\mathbf{P}$ and offset $\mathbf{c}$
- Minimize total quadradic approximation error:

$$\inf_{\text{center } \mathbf{c}} \; \inf_{k\text{-dim. proj. matrix } \mathbf{P}} \sum_t \| \underbrace{\mathbf{P}(\mathbf{x}_t - \mathbf{c})}_{\text{compressed}} - \underbrace{(\mathbf{x}_t - \mathbf{c})}_{\text{uncompressed}} \|_2^2$$

$$\begin{aligned} \mathbf{c}^* &= \text{average point} \\ \mathbf{P}^* &= \text{subspace spanned by } k \text{ longest axes} \\ &\quad \text{of covariance matrix } \sum_t (\mathbf{x}_t - \mathbf{c}^*)(\mathbf{x}_t - \mathbf{c}^*)^\top \end{aligned}$$

On-line 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

Assume $\mathbf{c} = 0$ for now

$$
\begin{aligned}
||\underbrace{\mathbf{P}}_{k}\mathbf{x} - \mathbf{x}||_2^2 &= ||(\mathbf{P} - \mathbf{I})\mathbf{x}||_2^2 \\
&= \mathbf{x}^\top (\mathbf{I} - \mathbf{P})^2 \mathbf{x} \\
&\overset{\mathbf{I}-\mathbf{P} \text{ proj.matr.}}{=} \operatorname{tr}(\underbrace{(\mathbf{I} - \mathbf{P})}_{n-k}\underbrace{\mathbf{x}\mathbf{x}^\top}_{C})
\end{aligned}
$$

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

Projection matrices are symmetric positive matrices
with eigenvalues in $\{0, 1\}$

$$\mathbf{P}^2 = \mathbf{P}, \quad (\mathbf{I} - \mathbf{P})^2 = \mathbf{I} - \mathbf{P}$$

So far

- Variance of alg. close to variance of smallest axis chosen in hindsight
- Minimizing variance along one direction 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 $m\, \boldsymbol{\omega}_t \cdot \boldsymbol{\lambda}_t$
- Update $\boldsymbol{\omega}_t$

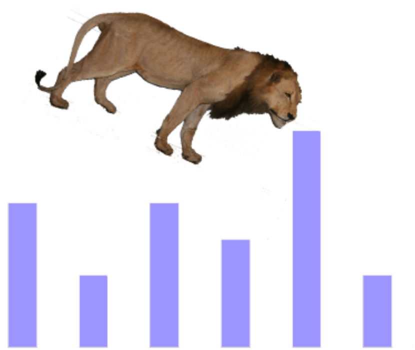
Goal: Total (expected) loss of alg. close to total loss of best expert

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

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

New trick: cap weights

Super predator algorithm

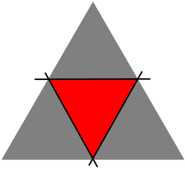


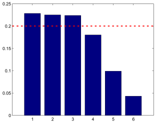
Preserves variety

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

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

$$\boldsymbol{\omega}_{t+1} = \inf_{\omega_i \leq \frac{1}{m}} \Delta(\boldsymbol{\omega}, \widehat{\boldsymbol{\omega}}_t)$$





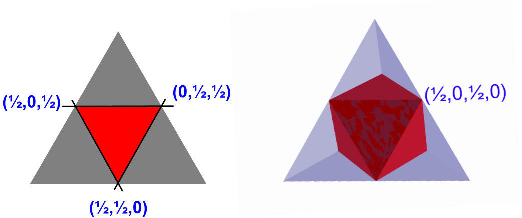
Cap and rescale rest

expected loss of alg

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

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



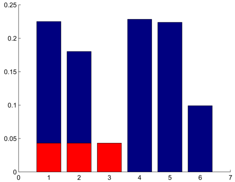
- We can effectively decompose any capped probability vector $\boldsymbol{\omega}$ as convex combination of n of the $\binom{n}{m}$ m-corners

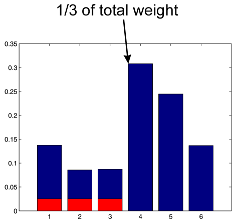
$$\boldsymbol{\omega} = \sum_{j=1}^{n} \alpha_j \mathbf{r}_j$$

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

Mixture construction

Invariant: $0 \leq \omega_i \leq \frac{|\boldsymbol{\omega}|}{m}$





In each iteration the boundary set $\{i : w_i \text{ is } 0 \text{ or } \frac{|\boldsymbol{\omega}|}{m}\}$ increases

- Boundary set never looses a component
- Either the smallest in the new corner
  or the largest of the remaining is added

Alternates to capping

- Follow the perturbed leader
    - Cheap but inferior bounds
- Dynamic programming
    - One weight per m-corner
    - More expensive to compute
    - Weaker bounds so far

Lift sets of expert alg. to matrices

- Pick $n-k$ dimensional subspace based on capped 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

$$\widehat{\mathbf{W}}_t = \frac{\exp(\log \mathbf{W}_t - \eta\, \mathbf{x}_t \mathbf{x}_t^\top)}{\mathrm{tr}(\exp(\log \mathbf{W}_t - \eta\, \mathbf{x}_t \mathbf{x}_t^\top))} \qquad \mathbf{W}_{t+1} = \inf_{\substack{\mathbf{W} \text{ dens.matrix} \\ \text{w.eigenvals } \leq \frac{1}{n-k}}} \Delta(\mathbf{W}, \hat{\mathbf{W}}_t)$$

- Generalization of soft min to soft min $n - k$

expected loss of alg

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

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

Overview again

comparator	batch	on-line
best single expert	min	softmin w. exponential weights or FPT
best direction PCA w. $k = n - 1$	min eigenvalue	softmin eigenvalue w. matrix exponentials
PCA w. $k < n - 1$	bottom segment of $n - k$ eigenvalues	softmin eigenvalue w. matrix exponentials and projections

Main techniques

- Density matrices to express uncertainty over directions
- Matrix Exponentiated Gradient Update
- Capping

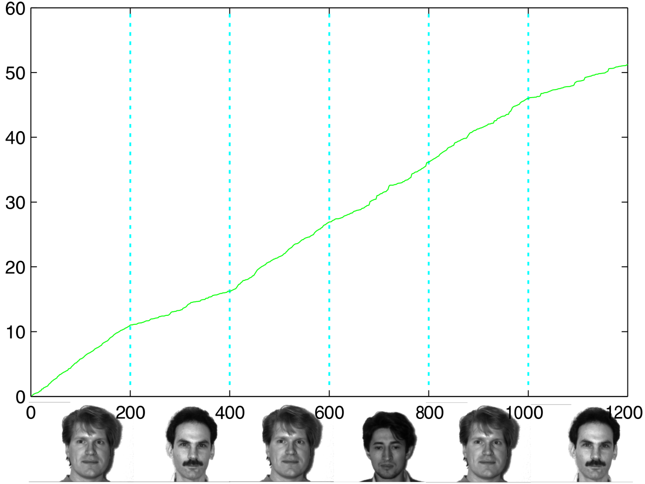
Outline

1. Batch PCA and why do we want to do it on-line
2. Expert setting
3. Variance minimization on the unit sphere
4. On-line PCA
5. What's next?

What's next?

- Generalize to centered case ✓
- Kernelize the algorithm ✓
- Generalize to asymmetric subspaces $\mathbf{P} = \sum_{i=1}^{k} \mathbf{u}_i \mathbf{v}_i^\top$ ✓
  - Use SVD instead of eigendecomposition
- Shifting methodology from expert setting carries over ✓
- Serious experiments (✓)
- Work out probability calculus for density matrices ✓
- Use soft min d as loss - generalization of logistic regression
- Survey on "The Blessing and Curse of the Multiplicative Updates"
  - Adapt quickly
  - Loss of variety
  - Connections to Biology

Loss of offline comparator



Additional loss of online algorithm