What is the best online algorithm for PCA?

Jiazhong Nie
Wojciech Kotlowski
Manfred K. Warmuth

University of California - Santa Cruz

July 5, 2013
Online PCA

- Batch PCA

Project onto subspace

Before see next point, learner picks random subspace
Then suffers compression loss
Keep uncertainty over subspaces with density matrix

Two main algorithms: EG and GD
Online PCA

Batch PCA

Online PCA
- Before see next point, learner picks random subspace
- Then suffers compression loss
- Keep uncertainty over subspaces with density matrix
- Two main algorithms: EG and GD
EG seems best for online PCA

- Known regret bound for EG:

\[
\sqrt{k \ln \frac{n}{k}} \text{ loss of best } k \text{ subspace}
\]
EG seems best for online PCA

- Known regret bound for EG:
  \[ \sqrt{k \ln \frac{n}{k}} \text{ loss of best } k \text{ subspace} \]

- Only logarithmic dependence on the dimension \( n \)
- EG seems best for simplex-like constraints
- Mathematically elegant
  - Quantum Relative Entropy, matrix log, matrix exponential
  - But computationally expensive
What about GD for PCA?

Motivated by the squared Frobenius norm $\|W\|_F^2 = \sum_{1 \leq i, j \leq n} (w_{i,j})^2$.

$$W_{t+1} = \inf_{W \text{ dens. matrix w. eigenvals } \leq \frac{1}{n-k}} \|W - W_t\|_F^2 + \eta \text{ tr}(WW_t^t),$$
What about GD for PCA?

Motivated by the squared Frobenius norm \(|W|^2_F = \sum_{1 \leq i,j \leq n} (w_{i,j})^2|.

\[ W_{t+1} = \inf_{\text{W dens. matrix}} \| W - W_t \|^2_F + \eta \text{tr}(Wx_t x_t^\top), \]

Rewrite into two steps:

Unconstrained min.: \( \hat{W}_t = W_t - \eta x_t x_t^\top \)

Projection: \( W_{t+1} = \inf_{W \succeq 0} \| W - \hat{W}_t \|^2_F \) with eigenvalues \( \leq \frac{1}{n-k} \) and \( \text{tr}(W) = 1 \).
Regret bound for GD

\[
\sqrt{2 \frac{k(n-k)}{n} T} \quad \text{if } n \gg k \\
\approx \sqrt{kT} \quad \text{if } n \gg k
\]
Regret bound for GD

\[ \sqrt{\frac{2k(n-k)}{n}} T \]

\[ n \gg k \]

\[ \sqrt{kT} \]

- No dependence on \( n \)?
- Better than \( EG \)?
Improved regret bound for EG for the PCA case

New: Exploit the fact that instances are sparse

\[
\sqrt{2(n-k) \log \frac{n}{n-k}} \leq \frac{k}{(n-k)} \leq \frac{n-k}{n} T \leq \sqrt{2k T} + k
\]
Improved regret bound for EG for the PCA case

New: Exploit the fact that instances are sparse

$$\sqrt{2(n - k) \log \frac{n}{n - k}} \leq \frac{k}{(n-k)} \leq \frac{n-k}{n} T \leq \sqrt{2k T} + k$$

Same bound as GD - but not better?
Improved regret bound for EG for the PCA case

New: Exploit the fact that instances are sparse

\[
\sqrt{2(n-k) \log \frac{n}{n-k}} \leq \frac{k}{(n-k)} \leq \frac{n-k}{n} T \leq \sqrt{2k T} + k
\]

- Same bound as GD - but not better?
- Are these Quantum Relative Entropies unnecessary?
EG’s revenge

**Budget bound** for EG instead of **time bound**

\[ \leq \sqrt{2k} \text{ loss of best } k \text{ subspace} + k \]

EG’s regret is optimal

Regret of GD

\[ \geq k \sqrt{\text{loss of best } k \text{ subspace}} \]
EG’s revenge

**Budget bound** for EG instead of **time bound**

\[ \leq \sqrt{2k} \text{ loss of best } k \text{ subspace} + k \]

EG’s regret is optimal

**Regret of GD**

\[ \geq k \sqrt{\text{loss of best } k \text{ subspace}} \]

**EG is better than GD by a factor of** \( \sqrt{k} \)
EG’s revenge

**Budget bound** for EG instead of **time bound**

\[ \leq \sqrt{2k} \text{ loss of best } k \text{ subspace} + k \]

EG’s regret is optimal

Regret of GD

\[ \geq k \sqrt{\text{loss of best } k \text{ subspace}} \]

**EG is better than GD by a factor of** \( \sqrt{k} \)

**Synopsis:**

- Time bounds are crude - focus on high loss case
EG’s revenge

Budget bound for EG instead of time bound

\[ \leq \sqrt{2k} \text{ loss of best } k \text{ subspace} + k \]

EG’s regret is optimal

Regret of GD

\[ \geq k \sqrt{\text{loss of best } k \text{ subspace}} \]

EG is better than GD by a factor of \( \sqrt{k} \)

Synopsis:
- Time bounds are crude - focus on high loss case
- Budget bounds are more refined - can focus on small loss case
GD seems to be hurt by “forgetting”

Alternate **Incremental Off-line GD**:

\[
W_{t+1} = \inf_{\begin{array}{l}
  W \text{ dens. matrix} \\
  \text{w. eigenvals } \leq \frac{1}{n-k}
\end{array}} \quad \frac{\|W - W_0\|_F^2}{\sum_{q=1}^t \text{tr}(Wx_qx_q^\top)}
\]

Divergence from first parameter

all data up to trail \(t\)
GD seems to be hurt by “forgetting”

Alternate **Incremental Off-line** GD:

\[
W_{t+1} = \inf_{W \text{ dens. matrix} \quad \text{w. eigenvals} \leq \frac{1}{n-k}} \left( \|W - W_0\|_F^2 + \eta \sum_{q=1}^{t} \text{tr}(Wx_qx_q^\top) \right)
\]

- **Incremental Offline** GD might have a budget bound as good as EG?