On-line variance minimization in $O(n^2)$ per trial?

Elad Hazan
Technion

Satyen Kale
Yahoo

Manfred K. Warmuth
UCSC

Open problem at COLT 2010
June 28, 2010
Since COLT 03, eight open problems

- On-line variance minimization in $O(n^2)$ per trial?
Synopsis of open problems

- Since COLT 03, **eight open problems**
- **None solved**
Since COLT 03, **eight open problems**
- None solved
- Had to solve **one myself** (this COLT)
Since COLT 03, eight open problems
None solved
Had to solve one myself (this COLT)
- with some brilliant co-authors
Synopsis of open problems

- Since COLT 03, **eight open problems**
- **None solved**
- Had to solve **one myself** (this COLT)
  - with some **brilliant co-authors**
- So far - **no money lost**
For trial $t = 1, \ldots,$

- Predict with a distribution $\mathbf{w}_{t-1}$ over $n$ experts
- Receive loss vector $\ell_t \in [0, 1]^n$
- Incur loss $\mathbf{w}_{t-1} \cdot \ell_t$
For trial $t = 1, \ldots,$

- Predict with a distribution $\mathbf{w}_{t-1}$ over $n$ experts
- Receive loss vector $\ell_t \in [0, 1]^n$
- Incur loss $\mathbf{w}_{t-1} \cdot \ell_t$

Hedge algorithm

$$w_{t,i} = \frac{e^{-\eta \ell_{t,i}}}{\sum e^{-\eta \ell_{t,i}}}$$
For trial $t = 1, \ldots$, 
- Predict with a distribution $w_{t-1}$ over $n$ experts 
- Receive loss vector $\ell_t \in [0, 1]^n$ 
- Incur loss $w_{t-1} \cdot \ell_t$

Hedge algorithm

$$w_{t,i} = \frac{e^{-\eta \ell_{<t,i}}}{\sum e^{-\eta \ell_{<t,i}}}$$

Regret bounds

$$\sum_{t} w_{t-1} \cdot \ell_t - \inf_{i} \ell_{\leq t,i} \leq \sqrt{2\ell^* \ln n + \ln n}$$
Loss matrices instead of loss vectors

- Symmetric loss matrices $L$
- Experts replaced by dyads $uu^T$
- Loss of $uu^T$ is $\text{tr}(uu^T L) = uL u^T$ which variance of random variable with covariance $L$ in direction $u$
Loss matrices instead of loss vectors

- Symmetric loss matrices $L$
- Experts replaced by dyads $uu^\top$
- Loss of $uu^\top$ is $\text{tr}(uu^\top L) = uLu^\top$ which variance of random variable with covariance $L$ in direction $u$
- Uncertainty over experts is probability distribution
- Uncertainty over dyads is density matrix $W$, i.e. eigenvalues nonnegative and sum to one
Matrix version of same problem \[KW\]

For trial \(t = 1, \ldots\),

- Predict with an \(n\)-dimensional density matrix \(W_{t-1}\)
- Receive loss matrix \(L_t\) which is symmetric and has eigenvalues in \([0,1]\)
- Incur loss \(W_{t-1} \cdot L_t\)
Matrix version of same problem [KW]

For trial $t = 1, \ldots,$

- Predict with an $n$-dimensional density matrix $\mathbf{W}_{t-1}$
- Receive loss matrix $\mathbf{L}_t$ which is symmetric and has eigenvalues in $[0,1]$
- Incur loss $\mathbf{W}_{t-1} \cdot \mathbf{L}_t$

Matrix Hedge algorithm

$$W_t = \frac{\exp(-\eta \mathbf{L}_{<t})}{\text{tr}(\exp(-\eta \mathbf{L}_{<t}))}$$
Matrix version of same problem \[KW\]

For trial \( t = 1, \ldots \),
- Predict with an \( n \)-dimensional density matrix \( W_{t-1} \)
- Receive loss matrix \( L_t \) which is symmetric and has eigenvalues in \([0,1]\)
- Incur loss \( W_{t-1} \cdot L_t \)

Matrix Hedge algorithm

\[
W_t = \frac{\exp(-\eta L_{<t})}{\text{tr}(\exp(-\eta L_{<t}))}
\]

Regret bounds

\[
\sum_t \text{tr}(W_{t-1} L_t) - \inf_u u^\top L_{\leq T} u \leq \sqrt{2\ell^* \ln n + \ln n}
\]

Expert setting recovered when all matrices \( L_t \) are diagonal
The issue of time

Matrix Hedge implementation requires eigendecomposition of current total loss matrix $L_t$:  

<table>
<thead>
<tr>
<th></th>
<th>input size</th>
<th>update time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector Hedge</td>
<td>$n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Matrix Hedge</td>
<td>$n^2$</td>
<td>$O(n^3)$</td>
</tr>
</tbody>
</table>

Open: Is there an algorithm for the matrix case with $O(n^2)$ update time?

Regret remains $\sqrt{2\ell^* \ln n} + \ln n$?

When rank of loss matrices small, then many tricks.

Elad Hazan
Technion

Satyen Kale
Yahoo

Manfred K. Warmuth
UCSC

On-line variance minimization in $O(n^2)$ per trial?

Open problem at COLT 2010

June 28, 2010 6
The issue of time

Matrix Hedge implementation requires eigendecomposition of current total loss matrix $L_t$: 

$$
\begin{array}{ccc}
\text{input size} & \text{update time} \\
\text{Vector Hedge} & n & O(n) \\
\text{Matrix Hedge} & n^2 & O(n^3)
\end{array}
$$

OPEN: Is there an algorithm for the matrix case with 

- $O(n^2)$ update time
- Regret remains $\sqrt{2\ell^* \ln n + \ln n}$?

When rank of loss matrices small, then many tricks
Can we use FPL?

Vector case

- Add random loss vector $r$ to $\ell_{t}$
- Predict with $\arg\min_i (\ell_{t,i} + r)$ \[KV\]
- With proper choice of $r$, FPL simulates Vector Hedge: \[K,KW\]

$$E(\arg\min_i \ell_{t,i} + r) = \frac{e^{-\eta \ell_{t,i}}}{\sum e^{-\eta \ell_{t,i}}}$$

Matrix case:

Add random loss matrix $R$ to $L_{t}$

Predict with $\arg\min_u u^\top (L_{t} + R)u$

$\arg\min_u u^\top (L_{t} + R)u$ is eigenvector with minimum eigenvalue

Takes $O(n^2)$ time to compute $\arg\min$
Can we use FPL?

Vector case
- Add random loss vector $\mathbf{r}$ to $\ell_{<t}$
- Predict with $\text{argmin}_i(\ell_{<t,i} + \mathbf{r})$ [KV]
- With proper choice of $\mathbf{r}$, FPL simulates Vector Hedge: [K,KW]

$$E(\text{argmin}_i \ell_{<t,i} + \mathbf{r}) = \frac{e^{-\eta \ell_{<t,i}}}{\sum e^{-\eta \ell_{<t,i}}}$$

Matrix case:
- Add random loss matrix $\mathbf{R}$ to $\mathbf{L}_{<t}$
- Predict with $\text{argmin}_u \mathbf{u}^\top (\mathbf{L}_{<t} + \mathbf{R}) \mathbf{u}$
- $\text{argmin}_u \mathbf{u}^\top (\mathbf{L}_{<t} + \mathbf{R}) \mathbf{u}$ is eigenvector with minimum eigenvalue
- Takes $O(n^2)$ time to compute argmin
What $R$?

- Decompose $L_{<t} = UDU^T$
- Add right vector perturbation $r$ to diagonal, i.e. choose $R = U \operatorname{diag}(r) U^T$:
  \[
  L_{<t} + R = U (D + r) U^T
  \]
What $\mathbf{R}$?

- Decompose $L_{<t} = \mathbf{U} \mathbf{D} \mathbf{U}^\top$
- Add right vector perturbation $\mathbf{r}$ to diagonal, i.e. choose $\mathbf{R} = \mathbf{U} \operatorname{diag}(\mathbf{r}) \mathbf{U}^\top$:

$$L_{<t} + \mathbf{R} = \mathbf{U} (\mathbf{D} + \mathbf{r}) \mathbf{U}^\top$$

Can simulate Matrix Hedge
- Regret $\sqrt{2\ell^* \ln n + \ln n}$
- But $O(n^3)$ time to choose $\mathbf{R}$
- No advantage over Matrix Hedge :-(

Elad Hazan Technion
Satyen Kale Yahoo
Manfred K. Warmuth UCSC

On-line variance minimization in $O(n^2)$ per trial?

Open problem at COLT 2010 June 28, 2010
A suboptimal choice of $\mathbf{R}$

- Pick $\mathbf{R} = \mathbf{U} \text{diag}(\mathbf{r}) \mathbf{U}^\top$,  
  - where $\mathbf{U}$ is random orthogonal  
  - $\mathbf{r}$ exponentially distributed entries  
- $O(n^3)$ preprocessing
- Use same perturbation $\mathbf{R}$ in each trial  
  - $O(n^2)$ per trial
- Suboptimal regret $\sqrt{\ell^* n}$
Open problem

Is there a different per trial choice of random matrix $R$ such that

- $R$ can be found in $O(n^2)$ time
- Predict with minimum eigendirection of perturbed total loss matrix
- Regret remains $\sqrt{2\ell^* \ln n + \ln n}$

What is needed:
- Perturbation matrix $R$ must "listen" to spectrum of total loss matrix
- But no time to decompose
- Would speed up all applications of Matrix Exponentiated Gradient algorithm

$100 for solving it
Open problem

Is there a different per trial choice of random matrix $R$ such that

- $R$ can be found in $O(n^2)$ time
- Predict with minimum eigendirection of perturbed total loss matrix
- Regret remains $\sqrt{2\ell^* \ln n} + \ln n$

What is needed:

- Perturbation matrix $R$ must “listen” to spectrum of total loss matrix
- But no time to decompose

Would speed up all applications of Matrix Exponentiated Gradient algorithm

$100$ for solving it