Follow the leader with Dropout perturbations - Additive versus multiplicative noise

Manfred K. Warmuth

Google Mountain View
August 25, 2015

Joint work with Tim Van Erven and Wojciech Kotłowski

Major insights from [Devroye, Lugosi, Neu 2013]
1. What is dropout?

2. Learning from expert advice

3. Hedge setting

4. The algorithms

5. Proof methods
Feed forward neural net
Weights parameters - sigmoids at internal nodes
Dropout training

- Stochastic gradient descent
- Randomly remove every hidden/input node with prob. $\frac{1}{2}$ before each gradient descent update

[Hinton et al. 2012]
Dropout training

- Very successful in e.g. image classification, speech recognition
- Many people trying to analyse why it works
  - [Wager, Wang, Liang, 2013]
  - [Helmbold, Long, 2014]

Why does it work?

- [Wagner, Wang, Liang, 2013]
- [Helmbold, Long, 2014]
What are we doing?

Prove bounds for dropout
- single neuron
- linear loss
1. What is dropout?

2. Learning from expert advice

3. Hedge setting

4. The algorithms

5. Proof methods
## Online learning with expert

<table>
<thead>
<tr>
<th>day 1</th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>...</th>
<th>$E_n$</th>
<th>prediction</th>
<th>label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### Online learning with expert predictions

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>...</th>
<th>$E_n$</th>
<th>prediction</th>
<th>label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>day 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Online learning with expert

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>...</th>
<th>$E_n$</th>
<th>prediction</th>
<th>label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>day 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>notation</td>
<td>$x_1$</td>
<td>$x_1$</td>
<td>$x_2$</td>
<td>...</td>
<td>$x_n$</td>
<td>$\hat{y}$</td>
<td>$y$</td>
<td>$</td>
</tr>
</tbody>
</table>
# Online learning with expert

<table>
<thead>
<tr>
<th>day 1</th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$\ldots$</th>
<th>$E_n$</th>
<th>prediction</th>
<th>label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$\ldots$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$\ldots$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Notation

- $x_1$, $x_2$, $\ldots$, $x_n$ are the input features.
- $\hat{y}$ is the prediction.
- $y$ is the label.
- $|\hat{y} - y|$ is the loss.

### Scope

- $x_1$, $x_2$, $\ldots$, $x_n$ are in $[0, 1]$.
- $\hat{y}$ is in $[0, 1]$.
- $y$ is in $\{0, 1\}$.
- $|\hat{y} - y|$ is in $[0, 1]$. 

Algorithm maintains probability vector $w$: 

\[
\text{prediction} = w \cdot x
\]

Loss is linear because label $y$ is in $\{0, 1\}$.

\[
|\hat{y} - y| = \sum_i w_i |x_i - y|
\]
Online learning with expert

<table>
<thead>
<tr>
<th>day 1</th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>...</th>
<th>$E_n$</th>
<th>prediction</th>
<th>label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>day 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

| notation | $x_1$ | $x_1$ | $x_2$ | ... | $x_n$ | $\hat{y}$ | $y$ | $|\hat{y} - y|$ |
| scope    | $\in [0, 1]$ | $\in [0, 1]$ | $\in [0, 1]$ | $\in \{0, 1\}$ | $\in [0, 1]$ |

- Algorithm maintains probability vector $\mathbf{w}$:
  - prediction $\hat{y} = \mathbf{w} \cdot \mathbf{x}$
## Online learning with expert

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>...</th>
<th>$E_n$</th>
<th>prediction</th>
<th>label</th>
<th>loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>day 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

### Notation
- $x_1, x_2, \ldots, x_n$
- $\hat{y}$
- $y$
- $|\hat{y} - y|$
- $\in [0, 1]$
- $\in \{0, 1\}$

### Algorithm
- Maintains probability vector $w$:
  - Prediction $\hat{y} = w \cdot x$

### Loss
- Linear because label $y \in \{0, 1\}$
  - $|\hat{y} - y|$
  - $\sum_i w_i |x_i - y|$
Outline

1. What is dropout?
2. Learning from expert advice
3. Hedge setting
4. The algorithms
5. Proof methods
On-line learning

Predicting with expert advice

\[ \hat{y} = \mathbf{w} \cdot \mathbf{x} \quad \text{loss } |\hat{y} - y| \]
On-line learning

Predicting with expert advice

\[ \hat{y} = w \cdot x \quad \text{loss} \ |\hat{y} - y| \]

trial \( t \)
- get advice vector \( x_t \)
- predict \( \hat{y}_t = w_t \cdot x_t \)
- get label \( y_t \)
- exp. losses \( |x_{t,i} - y_t| \)
- alg. loss \( |\hat{y}_t - y_t| \)
- update \( w_t \rightarrow w_{t+1} \)
Predicting with expert advice

\[ \hat{y} = w \cdot x \quad \text{loss} \mid \hat{y} - y \mid \]

Hedge setting

\[ \text{loss} \ w \cdot \ell \]

trial \( t \)
- get advice vector \( x_t \)
- predict \( \hat{y}_t = w_t \cdot x_t \)
- get label \( y_t \)
- exp. losses \( \mid x_{t,i} - y_t \mid \)
- alg. loss \( \mid \hat{y}_t - y_t \mid \)
- update \( w_t \rightarrow w_{t+1} \)
On-line learning

Predicting with expert advice
\[ \hat{y} = w \cdot x \quad \text{loss} \quad |\hat{y} - y| \]

Hedge setting
\[ \text{loss} \quad w \cdot \ell \]

trial \( t \)
- get advice vector \( x_t \)
- predict \( \hat{y}_t = w_t \cdot x_t \)
- get label \( y_t \)
- exp. losses \( |x_{t,i} - y_t| \)
- alg. loss \( |\hat{y}_t - y_t| \)
- update \( w_t \rightarrow w_{t+1} \)

trial \( t \)
- predict \( w_t \)
- get loss vector \( \ell_t \)
- exp. losses \( \ell_{t,i} \)
- alg. loss \( w_t \cdot \ell_t \)
- update \( w_t \rightarrow w_{t+1} \)
Predicting with a random expert

trial $t$
- predict $w_t$ or predict with random expert $\hat{i}_t$
Predicting with a random expert

- predict $w_t$
- get loss vector $\ell_t$
- alg. loss $w_t \cdot \ell_t$

or predict with random expert $\hat{i}_t$

or alg. expected loss $\mathbb{E} \left[ e_{\hat{i}_t} \cdot \ell_t \right] = \mathbb{E} \left[ e_{\hat{i}_t} \right] \cdot \ell_t$

weights are implicit

Only works for linear loss
Predicting with a random expert

trial \( t \)
- predict \( w_t \)  
- get loss vector \( \ell_t \)
- alg. loss \( w_t \cdot \ell_t \)  
  or alg. expected loss \( \mathbb{E} \left[ e_{\hat{i}_t} \cdot \ell_t \right] = \mathbb{E} \left[ e_{\hat{i}_t} \right] \cdot \ell_t \)
- update \( w_t \rightarrow w_{t+1} \)
Predicting with a random expert

- predict $w_t$
- get loss vector $l_t$
- alg. loss $w_t \cdot l_t$
- update $w_t \rightarrow w_{t+1}$

Weights are implicit

Only works for linear loss
Worst-case regret

$$\sum_{t=1}^{T} w_t \cdot \ell_t - \inf_{i \leq T} \ell^*$$

total expected loss of alg

loss $\ell^*$ of best expert

Should be logarithmic in $\#$ of experts $n$
1. What is dropout?
2. Learning from expert advice
3. Hedge setting
4. The algorithms
5. Proof methods
### Main algorithms

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$E_4$</th>
<th>$E_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t-1$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$t-1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

day $t - 1$  

$l_{\leq t-1,i}$
Main algorithms

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$E_4$</th>
<th>$E_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 0 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 0 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>day $t - 1$</td>
<td>0 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \ell_{t-1,i} \]

|       | 1 2 1 2 3 |
|-------|-------|-------|-------|-------|
| $\ell_{t-1,i}$ |       |       |       |       |
Main algorithms

\[
\begin{array}{ccccc}
E_1 & E_2 & E_3 & E_4 & E_5 \\
0 & 1 & 0 & 0 & 1 \\
1 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{ccccc}
\text{day } t - 1 & 0 & 0 & 1 & 1 & 1 \\
\ell_{\leq t-1,i} & 1 & 2 & 1 & 2 & 3 \\
\end{array}
\]

\[\hat{i}_t = \arg\min_i \ell_{\leq t-1,i}\] ties broken uniformly

FL
Main algorithms

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$E_4$</th>
<th>$E_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\begin{align*}
\text{day } t - 1 & \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \\
\ell_{\leq t-1,i} & \quad 1 \quad 2 \quad 1 \quad 2 \quad 3
\end{align*}

FL \quad \hat{i}_t = \arg\min_i \ell_{\leq t-1,i} \quad \text{ties broken uniformly}

FPL(\eta) \quad \hat{i}_t = \arg\min_i \ell_{\leq t-1,i} + \frac{1}{\eta} \xi_{t,i} \quad \text{indep. additive noise}
Main algorithms

<table>
<thead>
<tr>
<th></th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
<th>$E_4$</th>
<th>$E_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>day 0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$\ell_{\leq t-1,i}$</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**FL**

$$\hat{i}_t = \arg\min_i \ell_{\leq t-1,i}$$

ties broken uniformly

**FPL($\eta$)**

$$\hat{i}_t = \arg\min_i \ell_{\leq t-1,i} + \frac{1}{\eta} \xi_{t,i}$$

indep. additive noise

**Hedge($\eta$)**

$$w_i = \frac{e^{-\eta \ell_{\leq t-1,i}}}{Z}$$

Weighted Majority algorithm for pred. with Expert Advice

Soft min
Dropout

\[
\begin{array}{cccccc}
E_1 & E_2 & E_3 & E_4 & E_5 \\
0 & \chi & 0 & 0 & \chi \\
1 & 1 & 0 & 1 & 1 \\
\text{day } t - 1 & 0 & 0 & \chi & \chi & 1 \\
\end{array}
\]

\[\hat{\ell}_{\leq t-1,i}\]
\[ \hat{\ell}_{t,i} = \beta_{t,i} \ell_{t,i}, \quad \text{where } \beta_{t,i} \text{ iid Bernoulli} \]

\begin{array}{cccccc}
E_1 & E_2 & E_3 & E_4 & E_5 \\
0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 \\
\hline
day t - 1 & 0 & 0 & 1 & 1 & 1 \\
\hline
\hat{\ell}_{\leq t-1,i} & 1 & 1 & 0 & 1 & 2 \\
\end{array}
## Dropout

\[
\begin{array}{cccccc}
E_1 & E_2 & E_3 & E_4 & E_5 \\
0 & \checkmark & 0 & 0 & \checkmark \\
1 & 1 & 0 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{cccccc}
day \ t - 1 & 0 & 0 & \checkmark & \checkmark & 1 \\
\end{array}
\]

\[
\hat{\ell}_{\leq t-1,i} = 1 \ 1 \ 0 \ 1 \ 2
\]

\[
\hat{\ell}_{t,i} = \beta_{t,i} \ell_{t,i}, \quad \text{where} \ \beta_{t,i} \ \text{iid Bernoulli}
\]

\[
\begin{array}{ccccc}
\times & \times & \times & \times & \ell
\end{array}
\]

**FL on dropout**

\[
\hat{i}_t = \arg \min_{i} \hat{\ell}_{\leq t-1,i}
\]
How good?

Optimal worst case regret: $\sqrt{L^* \ln n} + \ln n$
How good?

Optimal worst case regret: $\sqrt{L^* \ln n + \ln n}$

- FL is bad
- FPL($\eta$) and Hedge($\eta$) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop
How good?

Optimal worst case regret: \( \sqrt{L^* \ln n} + \ln n \)

- FL is bad
- FPL(\( \eta \)) and Hedge(\( \eta \)) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop
- FL on dropout requires no tuning
How good?

Optimal worst case regret: $\sqrt{L^* \ln n + \ln n}$

- FL is bad
- FPL($\eta$) and Hedge($\eta$) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop
- **FL on dropout** requires no tuning
  - dropout better noise for achieving optimal worst case regret
  - additive noise needs tuning - multiplicative noise does not
- in iid case when gap between 1st and 2nd: $\log n$ regret
Optimal worst case regret: $\sqrt{L^* \ln n + \ln n}$

- FL is bad
- FPL($\eta$) and Hedge($\eta$) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop
- FL on dropout requires no tuning
  - dropout better noise for achieving optimal worst case regret
    - additive noise needs tuning - multiplicative noise does not
  - in iid case when gap between 1st and 2nd: $\log n$ regret
- In the meantime
  - new fancy algorithms by
    Haipeng Luo, Rob Schapire & Tim van Erven, Wouter Koolen
How good?

Optimal worst case regret: $\sqrt{L^* \ln n + \ln n}$

- FL is bad
- FPL($\eta$) and Hedge($\eta$) achieve optimal regret with tuning
  - fancy tunings: AdaHedge and Flipflop
- FL on dropout requires no tuning
  - dropout better noise for achieving optimal worst case regret
  - additive noise needs tuning - multiplicative noise does not
  - in iid case when gap between 1st and 2nd: $\log n$ regret

In the meantime
- new fancy algorithms by Haipeng Luo, Rob Schapire & Tim van Erven, Wouter Koolen
- also no tuning, many other advantages
Our path to dropout

- Loss vectors $\ell_t \rightarrow$ loss matrices $L_t$
- Prob. vectors $w_t \rightarrow$ density matrices $W_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq t-1,i}}}{Z} \rightarrow$ Matrix Hedge

$$W_t = \frac{\exp(-\eta L_{\leq t-1})}{Z'}$$

- Matrix Hedge $O(n^3)$ per update
Our path to dropout

- Loss vectors $\ell_t \rightarrow$ loss matrices $L_t$
- Prob. vectors $w_t \rightarrow$ density matrices $W_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell \leq t-1, i}}{Z} \rightarrow$ Matrix Hedge

$$W_t = \frac{\exp(-\eta L \leq t-1)}{Z'}$$

- Matrix Hedge $O(n^3)$ per update
- FL minimum eigenvector calculation of $L_{\leq t-1}$: $O(n^2)$
Our path to dropout

- Loss vectors $\ell_t \rightarrow$ loss matrices $L_t$
- Prob. vectors $w_t \rightarrow$ density matrices $W_t$
- Hedge $w_{t,i} = \frac{e^{-\eta\ell_{\leq t-1,i}}}{Z} \rightarrow$ Matrix Hedge

\[ W_t = \frac{\exp(-\eta L_{\leq t-1})}{Z'} \]

- Matrix Hedge $O(n^3)$ per update
- FL minimum eigenvector calculation of $L_{\leq t-1}$: $O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?
Our path to dropout

- Loss vectors $\mathbf{\ell}_t \rightarrow$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \rightarrow$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq t-1,i}}}{Z} \rightarrow$ Matrix Hedge

$$\mathbf{W}_t = \exp\left(-\eta \mathbf{L}_{\leq t-1}\right)$$

- Matrix Hedge $O(n^3)$ per update
- FL minimum eigenvector calculation of $\mathbf{L}_{\leq t-1}$: $O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?
- Follow the skipping leader:
  - Drop entire loss $\mathbf{L}_t$ with probability $\frac{1}{2}$
  - = Online Bagging
Our path to dropout

- Loss vectors $\ell_t \rightarrow$ loss matrices $L_t$
- Prob. vectors $w_t \rightarrow$ density matrices $W_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq t-1,i}}}{Z} \rightarrow$ Matrix Hedge
  \[ W_t = \frac{\exp(-\eta L_{\leq t-1})}{Z'} \]
- Matrix Hedge $O(n^3)$ per update
- FL minimum eigenvector calculation of $L_{\leq t-1}$: $O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?
- **Follow the skipping leader:**
  - Drop entire loss $L_t$ with probability $\frac{1}{2}$
    = Online Bagging
- Proof techniques break down
  - settled for vector case and independent multiplicative noise
    = dropout
Our path to dropout

- Loss vectors $\ell_t \rightarrow$ loss matrices $L_t$
- Prob. vectors $w_t \rightarrow$ density matrices $W_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq t-1,i}}}{Z} \rightarrow$ Matrix Hedge
  $$W_t = \frac{\exp(-\eta L_{\leq t-1})}{Z'}$$
- Matrix Hedge $O(n^3)$ per update
- FL minimum eigenvector calculation of $L_{\leq t-1}$: $O(n^2)$
- Is there $O(n^2)$ perturbation with optimal regret bound?

- **Follow the skipping leader:**
  - Drop entire loss $L_t$ with probability $\frac{1}{2}$
  - = Online Bagging
- Proof techniques break down
  - settled for vector case and independent multiplicative noise
  - = dropout
- **Follow the skipping leader** has linear regret [Lugosi, Neu2014]
What regularization?

Hedge(\(\eta\)) relative entropy
What regularization?

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hedge($\eta$)</td>
<td>relative entropy</td>
</tr>
<tr>
<td>FPL($\eta$)</td>
<td>additive $\frac{1}{\eta}$ log exponential noise $\Rightarrow$ Hedge($\eta$)</td>
</tr>
</tbody>
</table>
What regularization?

\begin{align*}
\text{Hedge}(\eta) & \quad \text{relative entropy} \\
\text{FPL}(\eta) & \quad \text{additive } \frac{1}{\eta} \log \text{exponential noise } = \text{Hedge}(\eta)
\end{align*}

\text{FL on dropout} \quad \text{tricky}

\begin{align*}
\text{Feed forward NN} & \quad \text{[Wagner, Wang, Liang 2013]} \\
\text{Logistic regression} & \quad \text{[Helmbold, Long 2014]} \\
\text{Linear loss case} & \quad \text{[ALST 2014]}
\end{align*}
Outline

1. What is dropout?
2. Learning from expert advice
3. Hedge setting
4. The algorithms
5. Proof methods
Simple algorithms

Any deterministic alg. (such as FL) has huge regret

- For $T$ trials: give algorithm’s expert a unit of loss
- Loss of alg.: $T$ loss of best: $\leq \frac{T}{n}$
Any deterministic alg. (such as FL) has huge regret

- For $T$ trials: give algorithm’s expert a unit of loss
- Loss of alg.: $T/L^*$  loss of best: $\leq \frac{T}{n}$

\[
\text{regret: } \geq \left( \frac{T}{nL^*} - \frac{T}{n} \right) = (n - 1)L^* 
\]
Simple algorithms

Any deterministic alg. (such as FL) has huge regret

- For $T$ trials: give algorithm’s expert a unit of loss
- Loss of alg.: $T$ loss of best: $\leq \frac{T}{n}$

$$\text{regret: } \geq \left( \frac{T}{nL^*} \right) - \left( \frac{T}{L^*} \right) = (n - 1)L^*$$

Recall optimum regret: $\sqrt{L^* \ln n} + \ln n$

FL with random ties
Simple algorithms

Any deterministic alg. (such as FL) has huge regret

- For $T$ trials: give algorithm’s expert a unit of loss
- Loss of alg.: $T$ loss of best: $\leq \frac{T}{n}$

\[
\text{regret: } \geq \frac{T}{nL^*} - \frac{T}{nL} = (n-1)L^*
\]

Recall optimum regret: $\sqrt{L^* \ln n + \ln n}$

FL with random ties

- Give every expert one unit of loss
  - iterate $L^* + 1$ times
- Loss per sweep $\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} + 1 \approx \ln n$
- Loss of alg.: $(L^* + 1) \ln n$ loss of best: $L^*$

regret: $L^* \ln n$
Our analysis of dropout

**Unit rule**

- Adversary forces more regret by splitting loss vectors into units

\[
\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}
\]
Our analysis of dropout

Unit rule

- Adversary forces more regret by splitting loss vectors into units

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

Swapping rule

$$\ell_{\leq T,i}$$

<p>| | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$$E_1$$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$E_2$$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$E_3$$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$$E_4$$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Our analysis of dropout

**Unit rule**
- Adversary forces more regret by splitting loss vectors into units

\[
\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

**Swapping rule**
\[\ell_{\leq T,i}\]

<table>
<thead>
<tr>
<th></th>
<th>(E_1)</th>
<th>(E_2)</th>
<th>(E_3)</th>
<th>(E_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1 1 1</td>
<td>1 1 1 1 1 1</td>
</tr>
<tr>
<td>(\ell)</td>
<td>9</td>
<td>8</td>
<td>10</td>
<td>6</td>
</tr>
</tbody>
</table>

- 1’s occur in some order
- Worst case: 1 before 1
- Otherwise adversary benefits from swapping
Worst-case pattern

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Assume we have \( s \) leaders
Assume we have $s$ leaders

\[
\begin{align*}
\text{$s$ leader get unit} & \\
\text{ignore non-leaders} & \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}
\end{align*}
\]
Cost per sweep

Assume we have $s$ leaders

$s$ leader get unit
ignore non-leaders

\[ FL = \frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \frac{1}{s-s-2} + \frac{1}{s-s-1} \]

\[ \approx \ln s \]
Cost per sweep

Assume we have \( s \) leaders

\[ s \text{ leader get unit } \]
\[ \text{ignore non-leaders} \]

\[
\begin{align*}
\text{FL} & = \frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \frac{1}{s-s-2} + \frac{1}{s-s-1} \\
& \approx \ln s
\end{align*}
\]

Dropdown

\[
\begin{align*}
\frac{1}{s} + \frac{1}{s-1/2} + \frac{1}{s-2/2} + \frac{1}{s-3/2} + \ldots + \frac{1}{s-(s-2)/2} + \frac{1}{s-(s-1)/2}
\end{align*}
\]
Cost per sweep

Assume we have \( s \) leaders

\[
\begin{cases}
1 \\
1 \\
1 \\
1
\end{cases}
\]

\( s \) leader get unit
ignore non-leaders

\[
\frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \frac{1}{s-s-2} + \frac{1}{s-s-1}
\]

\( \approx \ln s \)

**FL**

\[
\frac{2}{2s} + \frac{2}{2s-1} + \frac{2}{2s-2} + \frac{2}{2s-3} + \ldots + \frac{2}{2s-(s-2)} + \frac{2}{2s-(s-1)}
\]

\( \approx 2 (\ln 2s - \ln s) = 2 \ln 2 \)

**Dropout**
Cost per sweep

Assume we have \( s \) leaders

\[
\begin{align*}
\text{s leader get unit} & \quad 1 \\
\text{ignore non-leaders} & \quad 1 \\
\end{align*}
\]

\[
\text{FL} \quad \frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \frac{1}{s-s-2} + \frac{1}{s-s-1}
\]

\[\approx \ln s\]

**Dropout**

\[
\frac{2}{2s} + \frac{2}{2s-1} + \frac{2}{2s-2} + \frac{2}{s-3} + \ldots + \frac{2}{2s-(s-2)} + \frac{2}{2s-(s-1)}
\]

\[\approx 2 (\ln 2s - \ln s) = 2 \ln 2\]
$L^* = 0$ - one expert incurs no loss

FL

- One sweep

\[
\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} \approx (\ln n) - 1
\]

- Optimal
$L^* = 0$ - one expert incurs no loss

**FL**

- One sweep

\[
\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} \approx (\ln n) - 1
\]

- Optimal

**Dropout**

- # of leaders reduced by half in each sweep

\[
\approx \log_2 n \text{ sweeps times } \leq 2 \ln 2 = 1.386
\]

\[
\leq \ln n
\]
Overview of proof for noisy case

- Focus on first $L$ sweeps
- Only occurs constant regret if number of leaders $> 1$
Overview of proof for noisy case

- Focus on first $L$ sweeps
- Only occurs constant regret if number of leaders $> 1$
- Prob. that number of leaders $> 1$ is at most $\sqrt{\frac{\ln n}{q+1}}$ for sweep $q$
Overview of proof for noisy case

- Focus on first $L$ sweeps
- Only occurs constant regret if number of leaders $> 1$

Prob. that number of leaders $> 1$ is at most $\sqrt{\frac{\ln n}{q+1}}$ for sweep $q$

For Hedge$(\eta)$ and FPL$(\eta)$ cost per sweep constant and dependent on $\eta$
Dropout versus Hedge

![Graph showing comparison between Dropout and tuned Hedge](image)

- Dropout
- Tuned Hedge

regret vs sweep t

$L^*$
Outlook

- Combinatorial experts
- Matrix case
- Where else can dropout perturbations be used?
- Dropout for convex losses
- Dropout for neural nets
Outlook

- Combinatorial experts
- Matrix case
- Where else can dropout perturbations be used?
- Dropout for convex losses
- Dropout for neural nets
- Privacy
Iterate this pattern \( n \) times:

\[
\sum_{i=1}^{n} \left( \frac{n - i}{n - i + 1} + \frac{1}{2} \right)
\]

\[
\approx n - \ln n + \frac{n}{2}
\]

\( L^* = n \): Follow the Scipping Leader has linear regret
How does dropout ovoid this example?

$$\begin{array}{c|c}
0 & 1 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
\hline
\frac{n-1}{n} & \frac{1}{n-1} \\
\end{array}$$

It leaves the adversary clueless as to who the leader is, i.e., privacy against adversary.
How does dropout ovoid this example?

\[
\begin{array}{c|c}
0 & 1 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
1 & 0 \\
\end{array}
\]

\[
\begin{array}{c|c}
\frac{n-1}{n} & \frac{1}{2} \\
\end{array}
\]

It leaves the adversary clueless as to who the leader is i.e. privacy against adversary
sparse counter example

\[
\begin{array}{c|c}
0 & 1^* \\
\frac{1}{n-1} & 0 \\
\frac{1}{n-1} & 0 \\
\frac{1}{n-1} & 0 \\
\frac{1}{n-1} & 0 \\
\frac{1}{n^2} & \frac{1}{2} \\
\end{array}
\]

Iterate this pattern \( n \) times:

\[
\sum_{i=1}^{n} \left( \frac{n-i}{(n-i+1)^2} + \frac{1}{2} \right)
\]

\[
= \sum_{i=1}^{n} \left( \frac{1}{n-i+1} - \frac{1}{(n-i+1)^2} + \frac{1}{2} \right)
\]

\[
\approx \ln n - O(1) + \frac{n}{2}
\]

\( L^* = \ln n \): Follow the Scipping Leader has linear regret