Follow the leader with Dropout perturbations - Additive versus multiplicative noise

Manfred K. Warmuth

UCSC

June 11, 2015, UC Berkeley

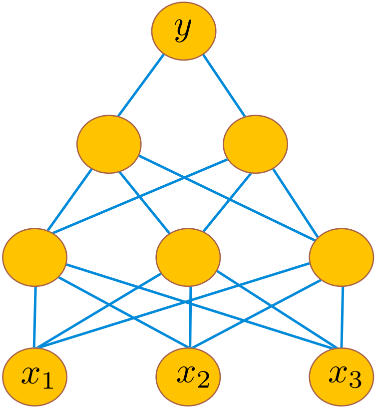
Joint work with Tim Van Erven and Wojciech Kotłowski

Universiteit Leiden

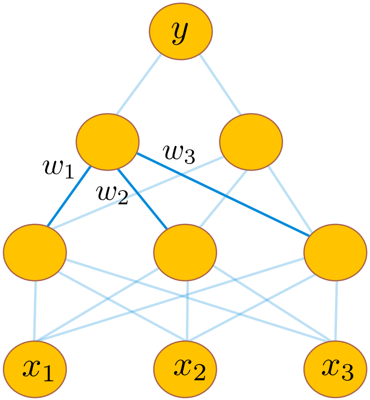
Major insights from [Devroye, Lugosi, Neu 2013]

Outline

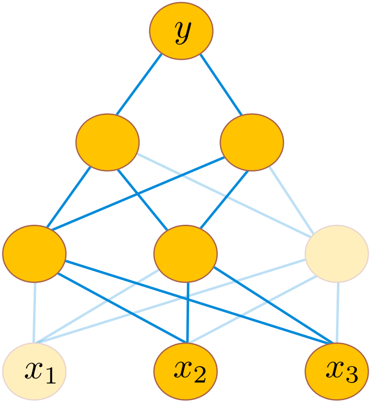
Feed forward neural net



Weights parameters - sigmoids at internal nodes



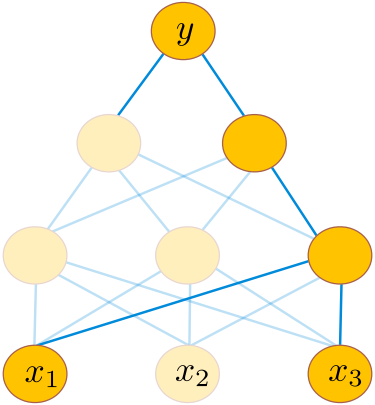
Dropout training



- 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 image recognition & speech recognition
- Why does it work?

  [Wagner, Wang, Liang 2013]
  [Helmbold, Long 2014]

What are we doing?

Prove bounds for dropout

- single neuron
- linear loss

Outline

Online learning with expert

	E_1	E_2	E_3	$\ldots$	E_n	$prediction$	label	loss
day 1	0	1	0	$\ldots$	0	0	1	1

Online learning with expert

	E_1	E_2	E_3	...	E_n	$prediction$	label	loss
day 1	0	1	0	...	0	0	1	1
day 2	1	1	0	...	0	1	1	0

Online learning with expert

	E_1	E_2	E_3	$\dots$	E_n	$prediction$	label	loss
day 1	0	1	0	$\dots$	0	0	1	1
day 2	1	1	0	$\dots$	0	1	1	0
notation	x_1	x_1	x_2	$\dots$	x_n	$\widehat{y}$	y	$\|\widehat{y} - y\|$

Online learning with expert

	E_1	E_2	E_3	$\dots$	E_n	*prediction*	label	loss
day 1	0	1	0	...	0	0	1	1
day 2	1	1	0	...	0	1	1	0
notation	x_1	x_1	x_2	$\dots$	x_n	$\widehat{y}$	y	$\lvert\widehat{y} - y\rvert$
scope	$\in [0,1]$			$\dots$		$\in [0,1]$	$\in \{0,1\}$	$\in [0,1]$

Online learning with expert

	E_1	E_2	E_3	$\dots$	E_n	*prediction*	label	loss
day 1	0	1	0	...	0	0	1	1
day 2	1	1	0	...	0	1	1	0
notation	x_1	x_1	x_2	$\dots$	x_n	$\widehat{y}$	y	$\lvert\widehat{y} - y\rvert$
scope	$\in [0, 1]$			$\dots$		$\in [0, 1]$	$\in \{0, 1\}$	$\in [0, 1]$

- Algorithm maintains probability vector $\mathbf{w}$:
  - prediction $\widehat{y} = \mathbf{w} \cdot \mathbf{x}$

Online learning with expert

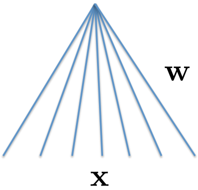
	E_1	E_2	E_3	$\dots$	E_n	*prediction*	label	loss
day 1	0	1	0	$\dots$	0	0	1	1
day 2	1	1	0	$\dots$	0	1	1	0
notation	x_1	x_1	x_2	$\dots$	x_n	$\widehat{y}$	y	$\lvert\widehat{y}-y\rvert$
scope	$\in [0,1]$			$\dots$		$\in [0,1]$	$\in \{0,1\}$	$\in [0,1]$

- Algorithm maintains probability vector $\mathbf{w}$:
  - prediction $\widehat{y} = \mathbf{w} \cdot \mathbf{x}$
- Loss linear because label $y \in \{0,1\}$
- $\underbrace{|\overbrace{\mathbf{w} \cdot \mathbf{x}}^{\widehat{y}} - y|}_{\text{loss of alg.}} = \sum_i \; w_i \; \underbrace{|x_i - y|}_{\text{loss of expert } i}$

Outline

Predicting with expert advice

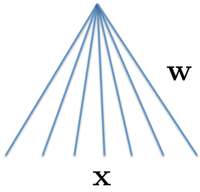
$\hat{y} = \mathbf{w} \cdot \mathbf{x}$ loss $|\hat{y} - y|$



On-line learning

Predicting with expert advice

$\hat{y} = \mathbf{w} \cdot \mathbf{x}$ loss $|\hat{y} - y|$



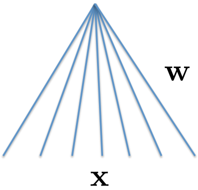
trial t

- get advice vector $\mathbf{x}_t$
- predict $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$
- get label y_t
- exp. losses $|x_{t,i} - y_t|$
- alg. loss $|\widehat{y}_t - y_t|$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

On-line learning

Predicting with expert advice

$\hat{y} = \mathbf{w} \cdot \mathbf{x}$ loss $|\hat{y} - y|$

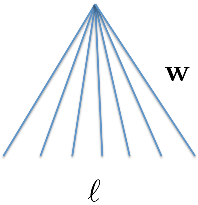


trial t

- get advice vector $\mathbf{x}_t$
- predict $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$
- get label y_t
- exp. losses $|x_{t,i} - y_t|$
- alg. loss $|\widehat{y}_t - y_t|$
- update $\mathbf{w}_t \rightarrow \mathbf{w}_{t+1}$

Hedge setting

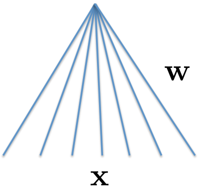
loss $\mathbf{w} \cdot \ell$



On-line learning

Predicting with expert advice

$\hat{y} = \mathbf{w} \cdot \mathbf{x}$ loss $|\hat{y} - y|$

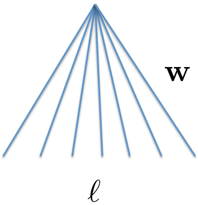


trial t

- get advice vector $\mathbf{x}_t$
- predict $\widehat{y}_t = \mathbf{w}_t \cdot \mathbf{x}_t$
- get label y_t
- exp. losses $|x_{t,i} - y_t|$
- alg. loss $|\widehat{y}_t - y_t|$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

Hedge setting

loss $\mathbf{w} \cdot \ell$



trial t

-
- predict $\mathbf{w}_t$
- get loss vector $\boldsymbol{\ell}_t$
- exp. losses $\ell_{t,i}$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ or predict with random expert $\widehat{i_t}$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ — or predict with random expert $\widehat{i_t}$
- get loss vector $\boldsymbol{\ell}_t$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$ — or alg. expected loss $\mathbb{E}\left[\mathbf{e}_{\widehat{i_t}} \cdot \boldsymbol{\ell}_t\right] = \underbrace{\mathbb{E}\left[\mathbf{e}_{\widehat{i_t}}\right]}_{\mathbf{w}_t} \cdot \boldsymbol{\ell}_t$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ — or predict with random expert $\widehat{i_t}$
- get loss vector $\boldsymbol{\ell}_t$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$ — or alg. expected loss $\mathbb{E}\left[\mathbf{e}_{\widehat{i_t}} \cdot \boldsymbol{\ell}_t\right] = \underbrace{\mathbb{E}\left[\mathbf{e}_{\widehat{i_t}}\right]}_{\mathbf{w}_t} \cdot \boldsymbol{\ell}_t$
- update $\mathbf{w}_t \to \mathbf{w}_{t+1}$

Predicting with a random expert

trial t

- predict $\mathbf{w}_t$ — or predict with random expert $\widehat{i}_t$
- get loss vector $\boldsymbol{\ell}_t$
- alg. loss $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$ — or alg. expected loss $\mathbb{E}\left[\mathbf{e}_{\widehat{i}_t} \cdot \boldsymbol{\ell}_t\right] = \underbrace{\mathbb{E}\left[\mathbf{e}_{\widehat{i}_t}\right]}_{\mathbf{w}_t} \cdot \boldsymbol{\ell}_t$
- update $\mathbf{w}_t \rightarrow \mathbf{w}_{t+1}$

weights are implicit

Only works for linear loss

How do we measure performance

Worst-case **regret**

$$\underbrace{\sum_{t=1}^{T} \mathbf{w}_t \cdot \boldsymbol{\ell}_t}_{\text{total expected loss of alg}} \quad - \quad \underbrace{\inf_i \ell_{\leq T,i}}_{\text{loss } \ell^* \text{ of best expert}}$$

Should be logarithmic in # of experts n

Outline

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$					

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

FL $\quad \widehat{i}_t = \operatorname{argmin}_i \ \ell_{\leq t-1,i}$ $\quad$ ties broken uniformly

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

FL	$\widehat{i}_t = \mathrm{argmin}_i \ \ell_{\leq t-1,i}$	ties broken uniformly
FPL(η)	$\widehat{i}_t = \mathrm{argmin}_i \ \ell_{\leq t-1,i} + \frac{1}{\eta}\xi_{t,i}$	indep. additive noise

Main algorithms

	E_1	E_2	E_3	E_4	E_5
	0	1	0	0	1
	1	1	0	1	1
day $t-1$	0	0	1	1	1
$\ell_{\leq t-1,i}$	1	2	1	2	3

FL	$\widehat{i}_t = \operatorname{argmin}_i \ \ell_{\leq t-1,i}$	ties broken uniformly
FPL(η)	$\widehat{i}_t = \operatorname{argmin}_i \ \ell_{\leq t-1,i} + \frac{1}{\eta}\xi_{t,i}$	indep. additive noise
Hedge(η)	$w_i = \frac{e^{-\eta \ell_{\leq t-1,i}}}{Z}$	Weighted Majority algorithm for pred. with Expert Advice Soft min

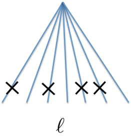
Dropout

	E_1	E_2	E_3	E_4	E_5
	0	$\not{1}$	0	0	$\not{1}$
	1	1	0	1	1
day $t-1$	0	0	$\not{1}$	$\not{1}$	1
$\widehat{\ell}_{\leq t-1,i}$					

Dropout

	E_1	E_2	E_3	E_4	E_5
	0	~~1~~	0	0	~~1~~
	1	1	0	1	1
day $t-1$	0	0	~~1~~	~~1~~	1
$\widehat{\ell}_{\leq t-1,i}$	1	1	0	1	2

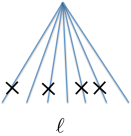
$\widehat{\ell}_{t,i} = \beta_{t,i}\ell_{t,i},$ where $\beta_{t,i}$ iid Bernoilli



Dropout

	E_1	E_2	E_3	E_4	E_5
	0	$\not{1}$	0	0	$\not{1}$
	1	1	0	1	1
day $t-1$	0	0	$\not{1}$	$\not{1}$	1
$\widehat{\ell}_{\leq t-1,i}$	1	1	0	1	2

$\widehat{\ell}_{t,i} = \beta_{t,i}\ell_{t,i},$ where $\beta_{t,i}$ iid Bernoilli



FL on dropout $\widehat{i}_t = \operatorname*{argmin}_i \widehat{\ell}_{\leq t-1,i}$

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(η) and Hedge(η) 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(η) and Hedge(η) 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(η) and Hedge(η) 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

How good?

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

- FL is bad
- FPL(η) and Hedge(η) 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(η) and Hedge(η) 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 $\boldsymbol{\ell}_t \quad \longrightarrow \quad$ loss matrices $\mathbf{L}_t$
- Prob. vectors $\mathbf{w}_t \quad \longrightarrow \quad$ density matrices $\mathbf{W}_t$
- Hedge $w_{t,i} = \frac{e^{-\eta \ell_{\leq t-1,i}}}{Z} \quad \longrightarrow \quad$ Matrix Hedge

$$\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1}\right)}{Z'}$$

- Matrix Hedge $O(n^3)$ per update

Our path to dropout

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

$$\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1}\right)}{Z'}$$

- Matrix Hedge $O(n^3)$ per update
- FL minimum eigenvector calculation of $\mathbf{L}_{\leq t-1}$: $O(n^2)$

Our path to dropout

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

$$\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1}\right)}{Z'}$$

- 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?

Our path to dropout

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

$$\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1}\right)}{Z'}$$

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

$$\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1}\right)}{Z'}$$

- 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
- Proof techniques break down
  - settled for vector case and independent multiplicative noise
  $=$ dropout

Our path to dropout

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

$$\mathbf{W}_t = \frac{\mathbf{exp}\left(-\eta \mathbf{L}_{\leq t-1}\right)}{Z'}$$

- 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
- 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(η) relative entropy

What regularization?

Hedge(η)	relative entropy
FPL(η)	additive $\frac{1}{\eta}$ log exponential noise $=$ Hedge(η)

What regularization?

Hedge(η)	relative entropy
FPL(η)	additive $\frac{1}{\eta}$ log exponential noise $=$ Hedge(η)
FL on dropout	tricky

Feed forward NN	[Wagner, Wang, Liang 2013]
Logistic regression	[Helmbold, Long 2014]
Linear loss case	[ALST 2014]

Outline

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: $\leq \frac{T}{n}$

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: $\leq \frac{T}{n}$

regret: $\geq \underbrace{T}_{nL^*} - \underbrace{\frac{T}{n}}_{L^*} = (n-1)L^*$

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: $\leq \frac{T}{n}$

regret: $\geq \underbrace{T}_{nL^*} - \underbrace{\frac{T}{n}}_{L^*} = (n-1)L^*$

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: $\leq \frac{T}{n}$

  regret: $\geq \underbrace{T}_{nL^*} - \underbrace{\frac{T}{n}}_{L^*} = (n-1)L^*$

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

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Our analysis of dropout

Unit rule

- Adversary forces more regret by splitting loss vectors into units

$$\begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Swapping rule

											$\ell_{\leq T,i}$
E_1	1	1	1	1	1	1	**1**	1	1		9
E_2	1	1	1	1	1	1	1	1			8
E_3	1	1	1	**1**	1	1	1	1	1	1	10
E_4	1	1	1	1	1	1					6

Our analysis of dropout

Unit rule

- Adversary forces more regret by splitting loss vectors into units

$$\begin{pmatrix}1\\0\\1\\1\end{pmatrix} \longrightarrow \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

Swapping rule

											$\ell_{\leq T,i}$
E_1	1	1	1	1	1	1	**1**	1	1		9
E_2	1	1	1	1	1	1	1	1			8
E_3	1	1	1	**1**	1	1	1	1	1	1	10
E_4	1	1	1	1	1	1					6

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

Worst-case pattern

```
1  1  1  1  1
 1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1
   1  1  1  1  1  1  1  1  1
    1  1  1  1  1  1  1  1  1
```

Cost per sweep

Assume we have s leaders

Cost per sweep

Assume we have s leaders

$$
\begin{array}{l} s \text{ leader get unit} \\ \text{ignore non-leaders} \end{array}
\left\{
\begin{array}{l}
1 \\
\;1 \\
\;\;1 \\
\;\;\;1 \\
\;\;\;\;1
\end{array}
\right.
$$

Cost per sweep

Assume we have s leaders

s leader get unit
ignore non-leaders

$$\left\{ \begin{array}{l} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right.$$

FL

$$\frac{1}{s} + \frac{1}{s - 1} + \frac{1}{s - 2} + \frac{1}{s - 3} + \ldots + \frac{1}{\underbrace{s - s - 2}_{2}} + \frac{1}{\underbrace{s - s - 1}_{1}}$$

$$\approx \ln s$$

Cost per sweep

Assume we have s leaders

s leader get unit
ignore non-leaders

$$\begin{cases} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{cases}$$

FL

$$\frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \frac{1}{\underbrace{s-s-2}_{2}} + \frac{1}{\underbrace{s-s-1}_{1}}$$

$$\approx \ln s$$

Dropout

$$\frac{1}{s} + \frac{1}{s-1/2} + \frac{1}{s-2/2} + \frac{1}{s-3/2} + \ldots + \frac{1}{s-(s-2)/2} + \frac{1}{s-(s-1)/2}$$

Cost per sweep

Assume we have s leaders

s leader get unit
ignore non-leaders
$$\begin{cases} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{cases}$$

FL

$$\frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \underbrace{\frac{1}{s-s-2}}_{2} + \underbrace{\frac{1}{s-s-1}}_{1}$$

$$\approx \ln s$$

Dropout

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

$$\approx 2\,(\ln 2s \; - \; \ln s) = 2 \ln 2$$

Cost per sweep

Assume we have s leaders

$$s \text{ leader get unit} \quad \text{ignore non-leaders} \quad \left\{ \begin{array}{l} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{array} \right.$$

FL

$$\frac{1}{s} + \frac{1}{s-1} + \frac{1}{s-2} + \frac{1}{s-3} + \ldots + \underbrace{\frac{1}{s-s-2}}_{2} + \underbrace{\frac{1}{s-s-1}}_{1}$$

$$\approx \ln s$$

Dropout

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

$$\approx 2\left(\ln 2s - \ln s\right) = 2 \ln 2$$

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

FL

- One sweep

$$\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} \cancel{+1} \approx (\ln n) - 1$$

- Optimal

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

FL

- One sweep

$$\frac{1}{n} + \frac{1}{n-1} + \ldots + \frac{1}{2} \cancel{+1} \approx (\ln n) - 1$$

- Optimal

Dropout

- # of leaders reduced by half in each sweep
- $\approx \log_2 n$ sweeps times $\leq 2 \ln 2 = 1.386$

  $=========================$

  $2 \ln n$

Overview of proof for noisy case

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

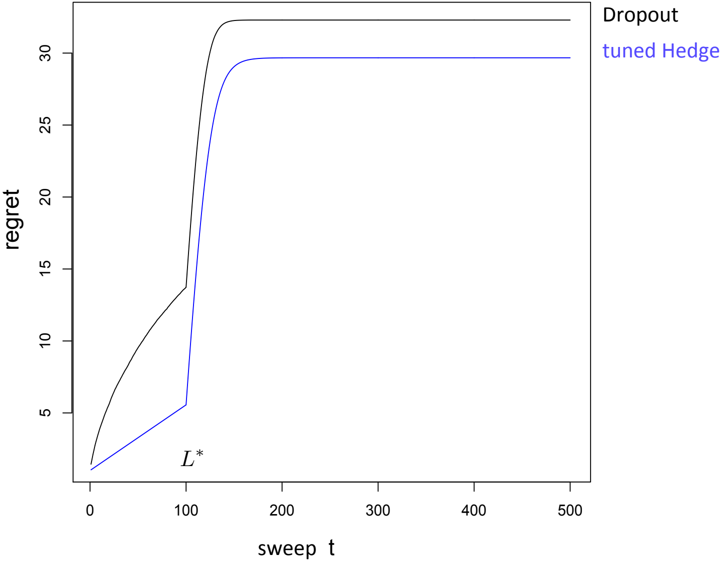
Overview of proof for noisy case

- 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(η) and FPL(η) cost per sweep constant and dependent on η

Dropout versus Hedge



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

[Lugosi, Neu 2014] dense counter example

0	1^*
1	0
1	0
1	0
1	0
1	0
$\frac{n-1}{n}$	$\frac{1}{2}$

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?

0	1
1	0
1	0
1	0
1	0
1	0
$\frac{n-1}{n}$	$\frac{1}{\frac{n-1}{2}}$

How does dropout ovoid this example?

0	1
1	0
1	0
1	0
1	0
1	0
$\frac{n-1}{n}$	$\frac{1}{\frac{n-1}{2}}$

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

sparse counter example

0	1^*
$\frac{1}{n-1}$	0
$\frac{1}{n-1}$	0
$\frac{1}{n-1}$	0
$\frac{1}{n-1}$	0
$\frac{1}{n-1}$	0
$\frac{n-1}{n^2}$	$\frac{1}{2}$

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