# Predictions for individual sequences

Reda Ouhamma

March 1, 2019

## 1 Part 1: Theory

**Q1, a** if we denote: $g(x) = log(1+x) - x + x^2$

$$\text{We have} \quad g'(x) = \frac{1}{1+x} - 1 + 2x = \frac{x(1+2x)}{1+x}$$

$$\text{And} \quad g'(x) \leq 0 \Leftrightarrow x \in [-\tfrac{1}{2}, 0] \Rightarrow g(x) \geq g(0) = 0 \quad \forall x \geq -\tfrac{1}{2}$$

**Q1, b** We have:

$$\begin{aligned}
log(W_{T+1}) &= log(\sum_{j=1}^{K} w_{T+1}(j)) \geq log(w_{T+1}(k)) \\
&\geq log(\prod_{s=1}^{T}(1+\eta(k)(p_s l_s - l_s(k))) \geq \sum_{s=1}^{T} log((1+\eta(k)(p_s l_s - l_s(k))) \\
&\geq \sum_{s=1}^{T} \eta(k)(p_s l_s - l_s(k)) - \eta(k)^2 (p_s l_s - l_s(k))^2 \quad \text{Using Q1} \\
&\geq \eta(k) \sum_{s=1}^{T} (p_s l_s - l_s(k)) - \eta(k)^2 \sum_{s=1}^{T} (p_s l_s - l_s(k))^2
\end{aligned}$$

Hence the inequality for all $k$.

**Q1, c** We have :

$$\begin{aligned}
W_{t+1} &= \sum_{j=1}^{K} w_{t+1}(j) = \sum_{j=1}^{K} w_t(j)(1+\eta(j)(p_t l_t - l_t(j))) \\
&= \sum_{j=1}^{K} w_t(j) + \sum_{j=1}^{K} w_t(j)(\eta(k)(p_t l_t - l_t(j))) \\
&= W_t + p_t l_t \sum_{j=1}^{K} w_t(j)\eta(j) - \sum_{j=1}^{K} w_t(j)\eta(j) l_t(j)) \\
&= W_t + p_t l_t \sum_{j=1}^{K} w_t(j)\eta(j) - \sum_{j=1}^{K} p_t(j) l_t(j)) \sum_{j=1}^{K} w_t(j)\eta(j) \\
&= W_t + p_t l_t \sum_{j=1}^{K} w_t(j)\eta(j) - p_t l_t \sum_{j=1}^{K} w_t(j)\eta(j) \\
&= W_t
\end{aligned}$$

Then:

$$\forall t \geq 1 \quad log(W_t) = log(W_1) = K$$

**Q1 - d** We rearrange the terms :

$$\sum_{s=1}^{T} (p_s l_s - l_s(k) \leq \frac{log(K)}{\eta} + \eta_k \sum_{s=1}^{T} (p_s l_s - l_s(k))$$

Then we optimize and find that:

$$\eta(k) = \sqrt{\tfrac{\log K}{\sum_{s=1}^{T} (p_s l_s - l_s(k))^2}}$$

achieves :

$$\sum_{s=1}^{T} (p_s l_s - l_s(k) \leq 2\sqrt{log(K) \sum_{s=1}^{T} (p_s l_s - l_s(k))^2}$$

**Q2 - a**

$$\begin{aligned}
\forall t \geq 1 \quad \tilde{p}_t \tilde{l}_t &= \sum_{k=1}^{K} \tilde{p}_t(k)(1_{k \in A_t} l_t(k) + 1_{k \notin A_t} p_t l_t) \\
&= \sum_{k=1}^{K} \tilde{p}_t(k) 1_{k \in A_t} l_t(k) + p_t l_t \sum_{k=1}^{K} \tilde{p}_t(k) 1_{k \notin A_t} \\
&= \sum_{k=1}^{K} p_t(k) l_t(k) \sum_{k=1}^{K} \tilde{p}_t(k) 1_{k \in A_t} + p_t l_t \sum_{k=1}^{K} \tilde{p}_t(k) 1_{k \notin A_t} \\
&= p_t l_t \sum_{k=1}^{K} \tilde{p}_t(k) = p_t l_t
\end{aligned}$$

Then :

$$\forall k \in \mathcal{X} \quad \tilde{p}_t \tilde{l}_t - \tilde{l}_t(k) = (p_t l_t - l_t(k)) 1_{k \in A_t}$$

**Q2 - b**

$$\begin{aligned}
R_T(k) = \sum_{t=1}^{T} (p_t l_t - l_t(k)) 1_{k \in A_t} &= \sum_{t=1}^{T} \tilde{p}_t \tilde{l}_t - \tilde{l}_t(k) \\
&\leq 2 \sqrt{log(K) \sum_{t=1}^{T} (\tilde{p}_t \tilde{l}_t - \tilde{l}_t(k))^2} \\
&= 2 \sqrt{log(K) \sum_{t=1}^{T} ((p_t l_t - l_t(k)) 1_{k \in A_t})^2} \\
&\leq 2 \sqrt{log(K) T_k} \qquad \text{because} \quad |p_t l_t - l_t(k)| \leq 1
\end{aligned}$$

## 2 Part 2: Experiments

**Q3** We chose this loss function because it is differentiable, convex and lipschitz which means that we have theoretical guarantees on all the online algorithms we've seen in class.

**Q5, a** The following plots show the performance of the algorithms with standard $\eta$ as recommended in the class materials. And for the prod algorithm, we compare the performance for different values of the latter to specify the best value.

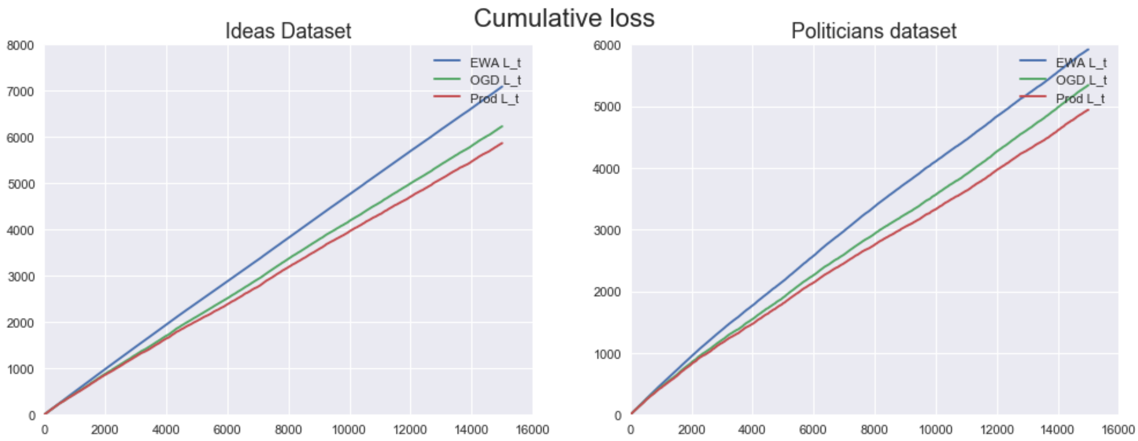



Figure 1: Cumulative loss on Ideas and Politicians dataset

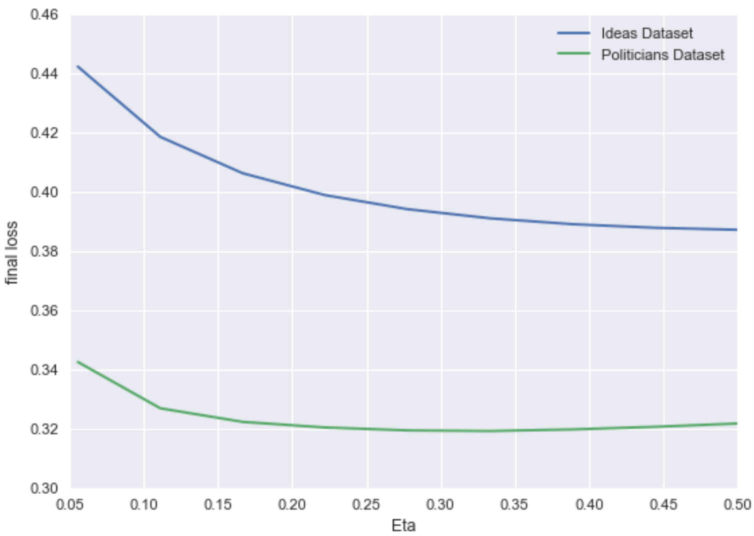
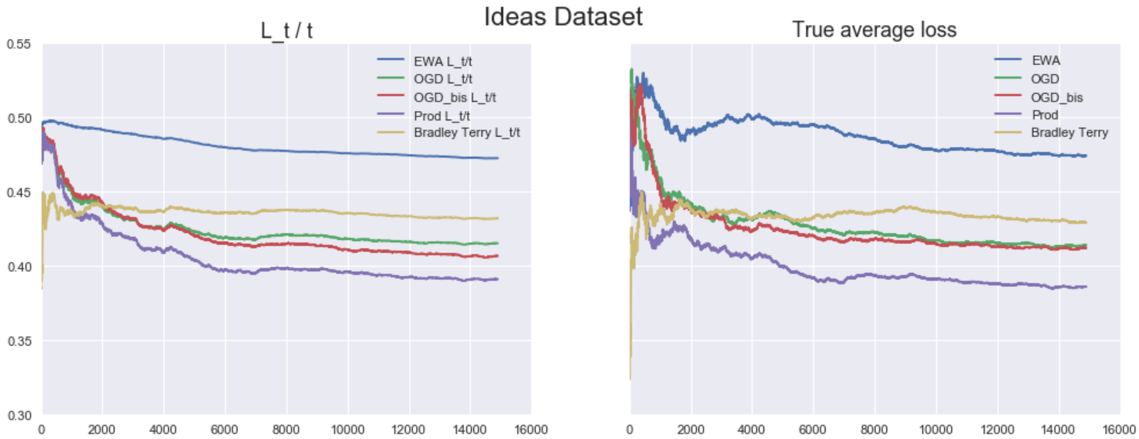
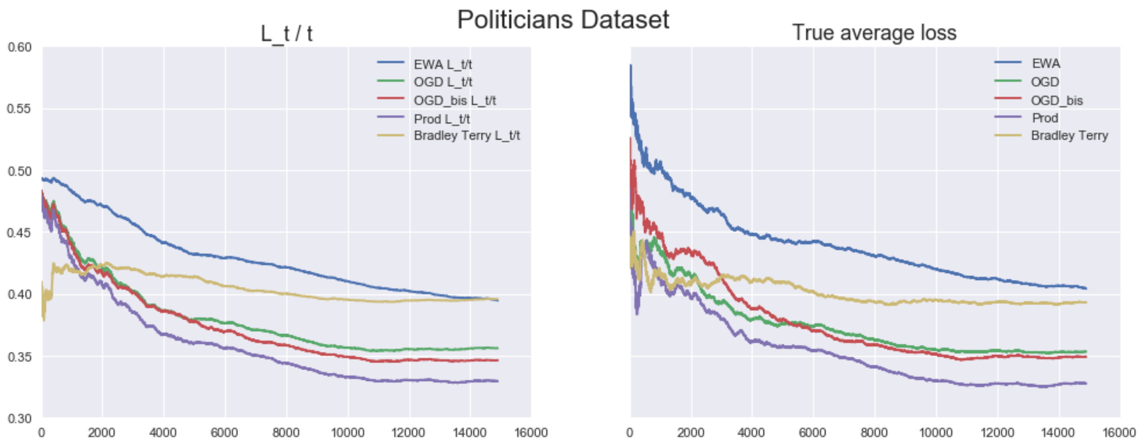



Figure 2: Grid search for eta for Prod

**Rest** These graphs show the average expected loss and the true loss for: EWA, OGD, Prod and two others which I will explain later on.




First of all, we can clearly see that all algorithms beat random predictions. Second, Prod algorithm seems to perform better then all the rest for both datasets.

To improve performance, I tried the standard Bradley Terry model but it only seems to outperform EWA, I think that a good change would be to add the Bradley Terry predictions as an expert and feed it to the algorithms at hand.

Another improvement I've read about is to use to OGD algorithm but at the update step, they recommend not updating the probabilities for which the experts are sleeping, this seems to slightly enhance OGD as we can see in the OGD_bis line.