Prediction of Individual Sequences - HW

Abdessamad Ed-dahmouni

February 2019

1 Theory - Sleeping experts

1.1 The prod algorithm

1.1.(a) We consider $f(x) = log(1+x) - x + x^2$, we have:

$$f'(x) = \frac{1}{1+x} - 1 + 2x = \frac{x(2x+1)}{x+1}$$

Which means that f defined on $[-\frac{1}{2}, +\infty[$ reaches its minimum for $x = 0$, which is 0.// This proves that:

$$\forall x \in [-\frac{1}{2}, +\infty[\quad log(1+x) \geq x - x^2$$

1.1.(b) For each $k \in \mathcal{X}$:

$$\begin{aligned}
\log(W_{T+1}) &\geq \log(w_{T+1}(k)) \\
&\geq \sum_{t=1}^{T} \log(1 + \eta(k)(p_t.\ell_t - \ell_t(k))) \\
&\geq \sum_{t=1}^{T} \eta(k)(p_t.\ell_t - \ell_t(k)) - \eta(k)^2(p_t.\ell_t - \ell_t(k))^2 \\
&= \eta(k) \sum_{t=1}^{T} (p_t.\ell_t - \ell_t(k)) - \eta(k)^2 \sum_{t=1}^{T} (p_t.\ell_t - \ell_t(k))^2
\end{aligned}$$

1.1.(c) We consider $t \geq 1$, we have:

$$\begin{aligned}
W_{t+1} &= \sum_{k \in \mathcal{X}} w_{t+1}(k) \\
&= \sum_{k \in \mathcal{X}} w_t(k)(1 + \eta(k)(p_t.\ell_t - \ell_t(k)) \\
&= \sum_{k \in \mathcal{X}} w_t(k) + p_t.\ell_t \sum_{k \in \mathcal{X}} \eta(k) w_t(k) - \sum_{k \in \mathcal{X}} \eta(k) w_t(k) \ell_t(k)
\end{aligned}$$

And since $p_t(k) = \frac{\eta(k) w_t(k)}{\sum_{j \in \mathcal{X}} \eta(j) w_t(j)}$, we have:

$$p_t.\ell_t \sum_{k \in \mathcal{X}} \eta(k) w_t(k) = \sum_{k \in \mathcal{X}} \eta(k) w_t(k) \ell_t(k)$$

This yields:

$$W_{t+1} = \sum_{k \in \mathcal{X}} w_t(k) = W_t$$

And since $W_1 = K$, we deduce:

$$\log(W_{T+1}) = \log(K)$$

1.1.(d)
Using 1.(b) and 1.(c), we get:

$$\sum_{t=1}^{T} p_t.\ell_t - \ell_t(k) \leq \frac{\log(K)}{\eta(k)} + \eta(k) \sum_{t=1}^{T} (p_t.\ell_t - \ell_t(k))^2$$

Optimizing $\eta(k)$, we get for:

$$\eta(k) = \sqrt{\frac{\log(K)}{\sum_{t=1}^{T} (p_t.\ell_t - \ell_t(k))^2}}$$

$$\sum_{t=1}^{T} p_t.\ell_t - \ell_t(k) \leq 2\sqrt{\log(K) \sum_{t=1}^{T} (p_t.\ell_t - \ell_t(k))^2}$$

1.2 Sleeping experts

1.2.(a) First, we will show that $\tilde{p}_t.\tilde{\ell}_t = p_t.\ell_t$:

$$\begin{aligned}
\tilde{p}_t.\tilde{\ell}_t &= \sum_{j \notin A_t} \tilde{p}_t(j) p_t.\ell_t + \sum_{j \in A_t} \tilde{p}_t(j) \ell_t(j) \\
&= p_t.\ell_t \sum_{j \notin A_t} \tilde{p}_t(j) + \sum_{j \in A_t} [p_t(j) \sum_{x \in A_t} \tilde{p}_t(x)] \ell_t(j) \\
&= p_t.\ell_t \sum_{j \notin A_t} \tilde{p}_t(j) + [\sum_{j \in A_t} p_t(j) \ell_t(j)] \sum_{x \in A_t} \tilde{p}_t(x) \\
&= p_t.\ell_t \sum_{j \in \mathcal{X}} \tilde{p}_t(j) \\
&= p_t.\ell_t
\end{aligned}$$

We consider $k \in \mathcal{X}$. Since $\tilde{\ell}_t(k) = \ell_t(k) \mathbb{1}_{\{k \in A_t\}} + p_t.\ell_t \mathbb{1}_{\{k \notin A_t\}}$, we have:

$$\begin{aligned}
\tilde{p}_t.\tilde{\ell}_t - \tilde{\ell}_t(k) &= (p_t.\ell_t - \ell_t(k)) \mathbb{1}_{\{k \in A_t\}} + (p_t.\ell_t - p_t.\ell_t) \mathbb{1}_{\{k \notin A_t\}} \\
&= (p_t.\ell_t - \ell_t(k)) \mathbb{1}_{\{k \in A_t\}}
\end{aligned}$$

1.2.(b) Using 1.1.(d) with $\tilde{p}_t$ and $\tilde{\ell}_t$, we get:

$$\sum_{t=1}^{T} \tilde{p}_t.\tilde{\ell}_t - \tilde{\ell}_t(k) \leq 2\sqrt{\log(K) \sum_{t=1}^{T} (\tilde{p}_t.\tilde{\ell}_t - \tilde{\ell}_t(k))^2}$$

And using the result from 1.2.(a), we have:

$$\begin{aligned}
R_T(k) &= \sum_{t=1}^{T} (p_t.\ell_t - \ell_t(k)) \mathbb{1}_{\{k \in A_t\}} \\
&= \sum_{t=1}^{T} \tilde{p}_t.\tilde{\ell}_t - \tilde{\ell}_t(k) \\
&\leq 2\sqrt{\log(K) \sum_{t=1}^{T} (\tilde{p}_t.\tilde{\ell}_t - \tilde{\ell}_t(k))^2}
\end{aligned}$$

And since $(p_t.\ell_t - \ell_t(k))^2 \leq 1$:

$$\sum_{t=1}^{T} (\tilde{p}_t.\tilde{\ell}_t - \tilde{\ell}_t(k))^2 = \sum_{t=1}^{T} (p_t.\ell_t - \ell_t(k))^2 \mathbb{1}_{\{k \in A_t\}}$$
$$\leq \sum_{t=1}^{T} \mathbb{1}_{\{k \in A_t\}} = T_k$$

Finally:

$$R_T(k) \leq 2\sqrt{\log(K) T_k}$$

2 Experiments – predict votes of surveys

2.3. This loss can be interpreted as $\ell(\hat{y}_t, y_t) = \mathbb{P}(Ber(\hat{y}_t) \neq y_t)$ where $Ber(p)$ is a Bernoulli r.v. with parameter p. This is the expected error of the forecaster at time t.
ℓ also has nice smoothness and convexity properties allowing the use of the algorithms we saw in class.

2.4. See the Jupyter notebook.

2.5.
2.5.(a) Here are the results for different values of η on the two datasets:

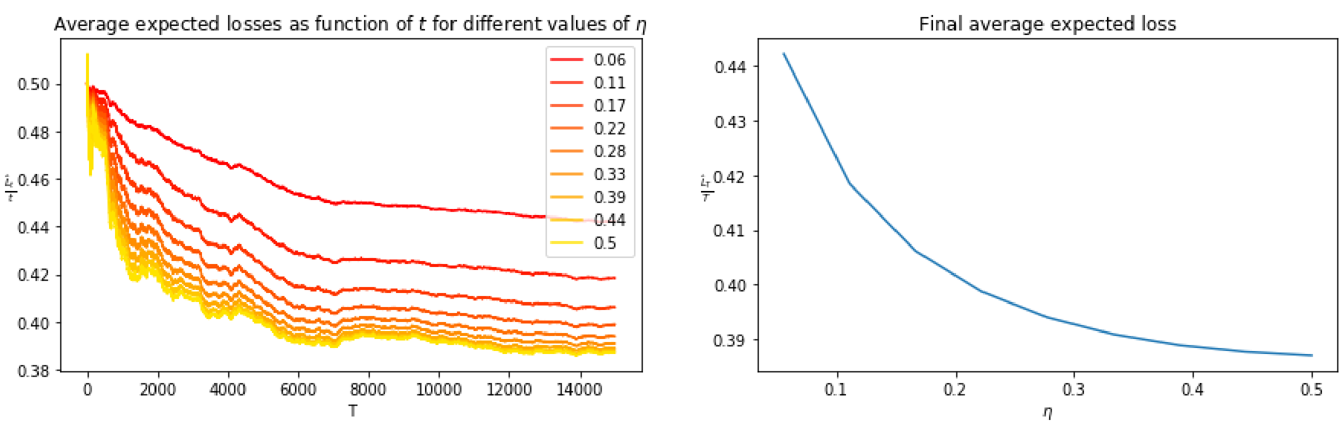

Figure 1: Results of the prod algorithm on ideas votes

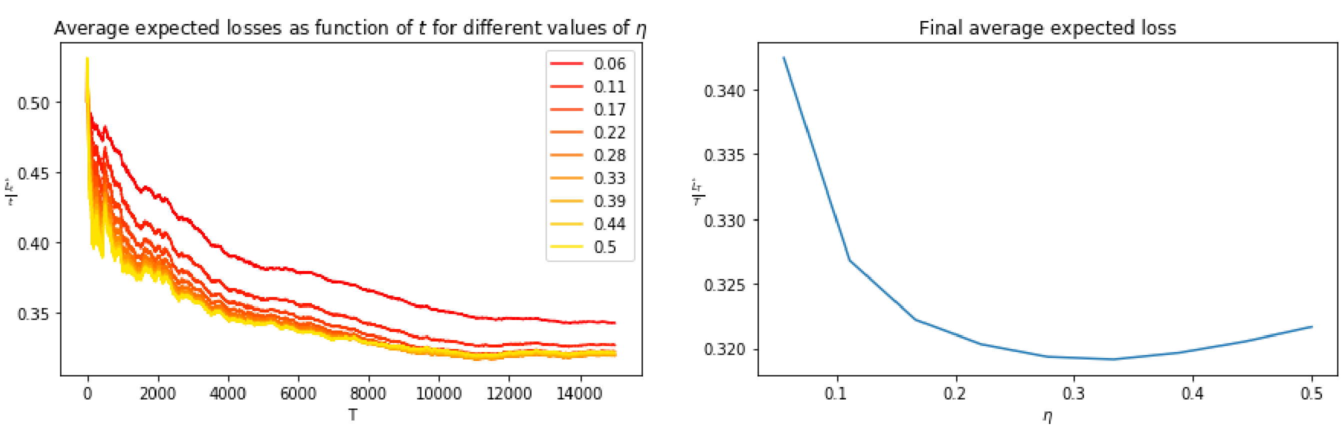

Figure 2: Results of the prod algorithm on politicians votes

2.5.(b) and (c) Here we plot the true average loss alongside the average expected loss for comparison, we notice that the two losses have the same behavior for high values of t and that the true average loss oscillates more for first iterations. For the value of η for each algorithm, we used η values that guarantee the theoretical upper bounds:

- For EWA : $\eta = \sqrt{\frac{log(K)}{T}}$;
- For OGD : $\eta = \frac{D}{G\sqrt{T}}$ with $D = \sqrt{2}$ and $G = \sqrt{K}$;
- For prod algorithm: $\eta = \sqrt{\frac{K\log(K)}{2T}}$ (I approximated $\sum_{t=1}^{T}(\tilde{p}_t.\tilde{\ell}_t - \tilde{\ell}_t(k))^2$ with $\frac{2T}{K}$).

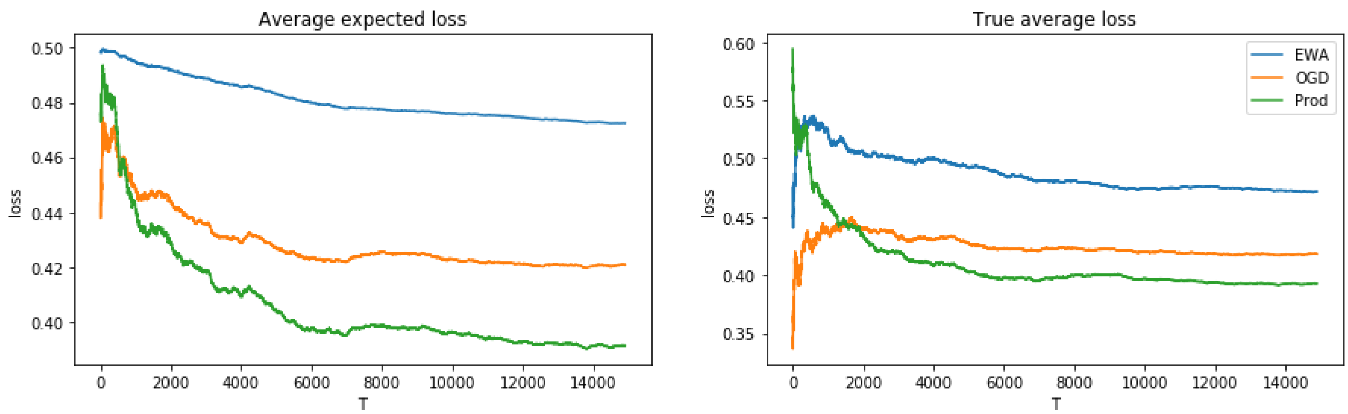

Figure 3: Average expected loss and true average loss of EWA, OGD, and prod algorithm - ideas votes

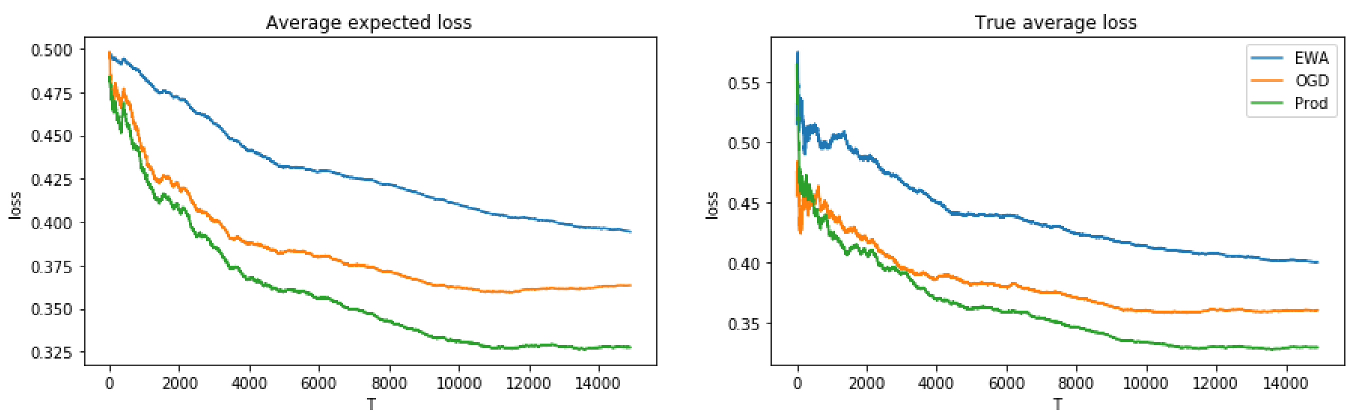

Figure 4: Average expected loss and true average loss of EWA, OGD, and prod algorithm - politicians votes

2.6. I implemented EG algorithm and Bradley-Terry iterative algorithm (from Wikipedia). Here are the results: (I used $\eta = \frac{1}{G}\sqrt{\frac{\log(K)}{T}}$ for EG as outlined in the course).

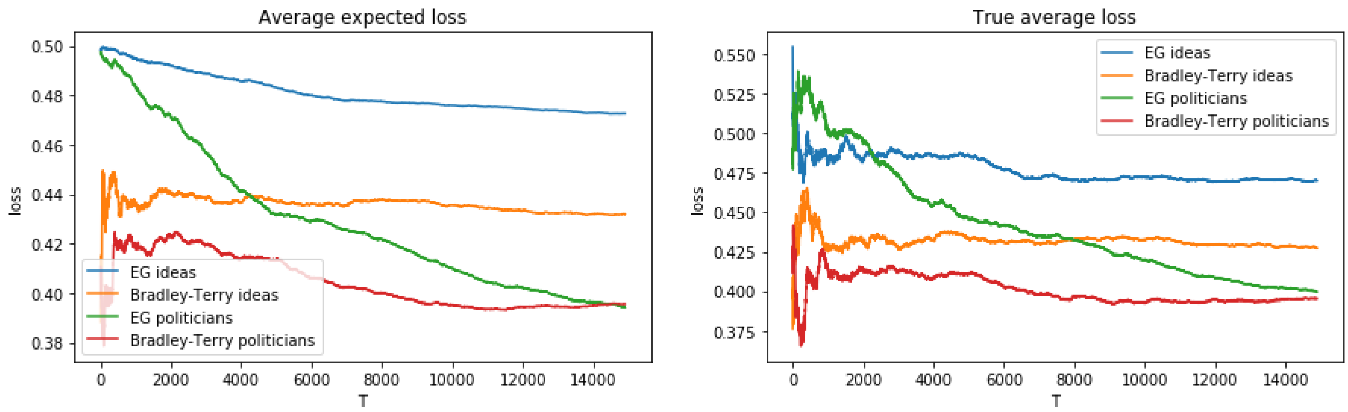

Figure 5: Average expected loss and true average loss of EG and Bradley-Terry model for both datasets