EECS 598-006: Prediction, Learning and Games **Fall 2013**

Lecture 2: Weighted Majority Algorithm

Lecturer: Jacob Abernethy *Scribe: Petter Nilsson*

2.1 Course announcements

- A GSI will (likely) be recruited to the course.
- For now, office hours are Tuesday 1pm-2pm.
- Homework 1 is about to be posted, due 9/23. Group work is encouraged for homework.

2.2 Predictive sorting revisited

Lecture 1 ended by studying the problem of predicting the outcomes of games played among k teams, where a game between two of these k teams is played in every round. Assume the outcomes y_s, $s = 1 \dots t-1$ in $t-1$ rounds have been observed for pairs of teams $(i_1, j_1), \dots, (i_{t-1}, j_{t-1})$. Further assume that there exists a perfect ranking $\pi^* \in S_k$ determining the outcomes y_s of each game. Without loss of generality (WLOG) assume $\pi^*(i_s) < \pi^*(j_s)$ for $s = 1, \dots, t-1$. In round t, the pair (i_t, j_t) is observed.

The Halving algorithm makes a prediction $\hat{y}_t$ of the outcome y_t by comparing the number of members of the following sets:

$$C_t^< := \{\pi \in S_k \mid \pi(i_s) < \pi(j_s) \; \forall s = 1, \dots, t-1 \text{ and } \pi(i_t) < \pi(j_t)\}, \tag{2.1}$$

$$C_t^> := \{\pi \in S_k \mid \pi(i_s) < \pi(j_s) \; \forall s = 1, \dots, t-1 \text{ and } \pi(i_t) > \pi(j_t)\}. \tag{2.2}$$

The prediction follows majority vote, i.e.

$$\hat{y}_t = \begin{cases} < & \text{if } |C_t^<| \geq |C_t^>|, \\ > & \text{otherwise.} \end{cases} \tag{2.3}$$

The naive way to determine $\hat{y}_t$ is to enumerate S_k and eliminate permutations that are inconsistent with past outcomes. This is however not efficient since $|S_k| = k!$.

The problem of computing the quantitites $|C_t^<|$ and $|C_t^>|$ is in fact $\#P$-hard. The outcomes can be recorded in a directed acyclic graph (DAG), and the problem is equivalent to finding the number of topological orderings of that graph, which is known to be $\#P$-complete [1].
(Challenge Problem) *Continuation from last lecture*: Is the problem of determining the *largest* of these two sets also $\#P$-complete, or is there an efficient algorithm?

2.3 Review of basics in Convex Analysis

Definition 2.1 *A function $f : \mathbb{R}^n \to R$ is* convex *if and only if any of the following conditions hold:*

(a) For all $x, y \in \mathbb{R}^n$,

$$\frac{1}{2}f(x) + \frac{1}{2}f(y) \geq f(\frac{x+y}{2}). \tag{2.4}$$

(b) *For all $x, y \in \mathbb{R}^n$ and $\alpha \in [0,1]$,*

$$(1-\alpha)f(x) + \alpha f(y) \geq f\left((1-\alpha)x + \alpha y\right). \tag{2.5}$$

(c) *For all random variables X, Jensen's inequality*

$$\mathbb{E}f(X) \geq f(\mathbb{E}(X)) \tag{2.6}$$

is satisfied.

Remark 2.2 *As seen in* (2.6)*, Jensen's inequality can actually be used to* define *convexness.*

(Exercise) Prove that (2.4)-(2.6) are equivalent.

Proposition 2.3 *A sufficient condition for f to be convex is that*

$$\nabla^2 f \succeq 0, \tag{2.7}$$

where $\succeq 0$ stands for positive semi-definiteness, i.e. $\nabla^2 f$ has non-negative eigenvalues.

Proposition 2.4 *If f is convex and differentiable, then for all $x_0, \delta \in \mathbb{R}^n$,*

$$f(x_0 + \delta) \geq f(x_0) + \delta \cdot \nabla f(x_0). \tag{2.8}$$

Remark 2.5 *Equation* (2.8) *could also be used to define convexness. However, in the case of non-differentiability the concept of subgradient is required.*

2.3.1 Approximations

The following inequalities will be used in the course. The functions are plotted in Figure 2.1 for a visual interpretation of these inequalities.

(1) Upper bound on the exponential function obtained by linearizing e^x in 0:

$$e^x \geq 1 + x. \tag{2.9}$$

(1') Lower bound on the logarithm function, derived by using the log operator on (2.9):

$$\log(1+x) \leq x. \tag{2.10}$$

(2) This inequality follows from the convexity of $f(z) = e^{\alpha z}$ and is obtained by inserting 0 and 1 as evaluation points in (2.5):

$$e^{\alpha x} \leq 1 + (e^{\alpha} - 1)x \quad \text{for } x \in [0,1]. \tag{2.11}$$

(3) Finally, a harder inequality:

$$-\log(1-x) \leq x + x^2 \quad \text{for } x \in [0, 1/2]. \tag{2.12}$$

(Challenge Problem) *Logarithm inequality*: Prove the inequality (2.12).

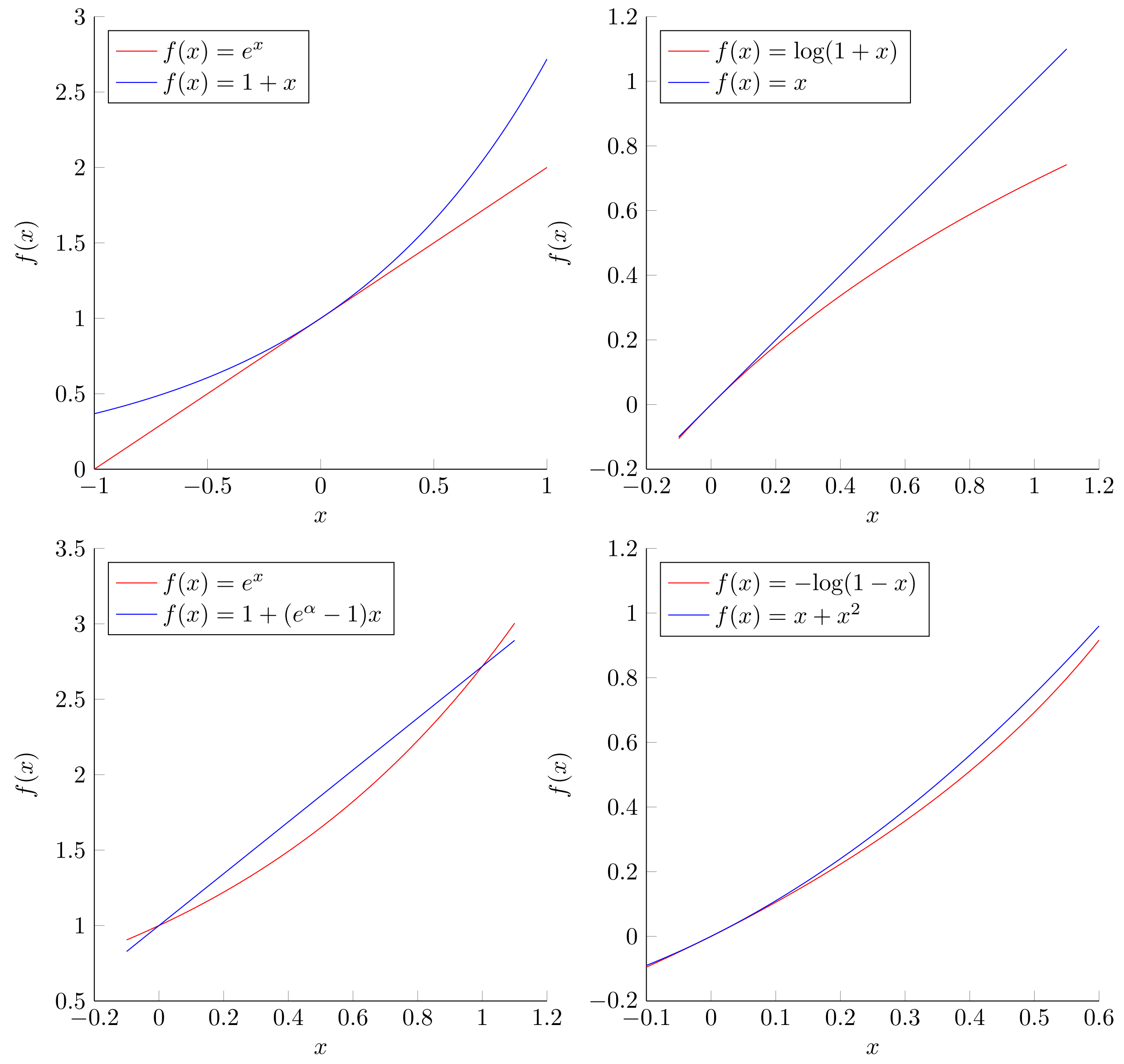

Figure 2.1: Plots of the functions in the inequalities (2.9) - (2.12).

2.4 Back to Online Learning

The drawback of the Halving algorithm presented last time is the strong assumption that a perfect ‘expert’ exists. How can this assumption be removed? The answer is to use weights for the ‘experts’, based on past performance. This leads to the WEIGHTED MAJORITY ALGORITHM (WMA) outlined below. This algorithm was first proposed 1994 by Littlestone and Warmuth in [2].

Algorithm 1: WEIGHTED MAJORITY ALGORITHM

```
input: $N$ experts predicting the outcomes, a parameter $\epsilon > 0$.
$w_i^1 \longleftarrow 1$   for all $i = 1, \dots, N$   (weights initialized to 1 for all experts)
for rounds t=1,2,... do
    $f_i^t \in \{0,1\}$   (obtain prediction of expert $i$, $i = 1, \dots, N$)
    $\hat{y}_t \longleftarrow \text{round}\left(\frac{\sum_i w_i^t f_i^t}{\sum_i w_i^t}\right)$   (compute prediction of WMA)
    Outcome $y_t$ is revealed
    $w_i^{t+1} \longleftarrow w_i^t (1-\epsilon)^{\mathbb{1}[f_i^t \neq y_t]}$   (expert weights are updated according to their predictions)
end
```

The function round : $[0,1] \to \{0,1\}$ above is defined as

$$\text{round}(x) = \begin{cases} 1 & \text{if } x \in [0, 1/2], \\ 0 & \text{if } x \in (1/2, 1]. \end{cases} \tag{2.13}$$

Theorem 2.6 *For all experts i, it holds that*

$$M_T(\mathit{WMA}) \leq \frac{2 \log N}{\epsilon} + 2(1+\epsilon) M_T(\mathit{expert}\ i), \tag{2.14}$$

where $M_T(\mathit{WMA})$ (resp. $M_T(\mathit{expert}\ i)$) is the number of mistakes of the algorithm (resp. expert i) up to round T.

Proof Idea: The proof is similar to the that of the bound on the Halving algorithm, but instead of ‘remaining number of experts’, ‘total weight’ is used.

Proof: Define total weight at time t as

$$\Phi_t = \sum_{i=1}^{N} w_i^t. \tag{2.15}$$

Evidently, we have that

$$\Phi_1 = N. \tag{2.16}$$

Furthermore, it holds that

$$\Phi_T \geq w_i^T = (1-\epsilon)^{M_T(\text{expert } i)}, \tag{2.17}$$

since the weight of expert i will have been decreased $M_T(\text{expert } i)$ times.

Suppose that the algorithm makes a false prediction at time t. Then more than half of the weight Φ_t must have been on experts who made false predictions. Thus, their weight would decrease by a factor $1-\epsilon$. This leads to the following result:

Lemma 2.7 *If the algorithm makes a mistake at time t, the following inequality holds:*

$$\Phi_{t+1} \leq \Phi_t \left(1 - \frac{\epsilon}{2}\right). \tag{2.18}$$

(Exercise) Prove this lemma rigorously.

By repeatedly applying the lemma,

$$\Phi_T \leq \Phi_1 \left(1 - \frac{\epsilon}{2}\right)^{M_T(\text{WMA})} . \tag{2.19}$$

Using (2.16), (2.17) and (2.19) one then obtains

$$(1-\epsilon)^{M_T(\text{expert } i)} \leq N \left(1 - \frac{\epsilon}{2}\right)^{M_T(\text{WMA})} . \tag{2.20}$$

Apply the negative logarithm to get

$$\underbrace{-\log(1-\epsilon)}_{\leq \epsilon + \epsilon^2} M_T(\text{expert } i) \geq -\log(N) + \underbrace{\left(-\log(1 - \frac{\epsilon}{2})\right)}_{\geq \epsilon/2} \cdot M_T(\text{WMA}), \tag{2.21}$$

where the new inequalities result from using (2.12)[1] and (2.10), respectively. By inserting these and rearranging the terms, the inequality stated in the theorem is derived. ■

2.5 Next lecture

The Weighted Majority Algorithm will be discussed further in the next lecture; a variant of the algorithm will be presented which gets rid of the factor 2 in Theorem 2.6.

References

[1] Brightwell, G., and Winkler, P. Counting linear extensions is #p-complete. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing* (New York, NY, USA, 1991), STOC '91, ACM, pp. 175–181.

[2] Littlestone, N., and Warmuth, M. The weighted majority algorithm. *Information and Computation 108*, 2 (1994), 212 – 261.

[1]To use this inequality, ε must be less than 0.5 (or actually 0.68)