Exercises

Optimization Methods in Finance

Fall 2010

Sheet 2

Note: This is just one way, a solution could look like. We do not guarantee correctness. It is your task to find and report mistakes.

Exercise 2.1 (*)
Consider again the simple setting, where we have $N$ experts that (over a time horizon of $T$ units) predict a binary event ($y_j^t \in \{0, 1\}$) and a forecaster tries to predict the events so that he is not making significantly more mistakes than the best of the experts. Consider the following strategies:

- **Strategy 1:** The forecaster chooses at any time $t$ the prediction $\hat{p}_t$ of the expert $j$ who made the least number of mistakes so far (i.e. $\hat{p}_t = y_j^t$ where $j = \text{argmin}\{m_j\}$ and $m_j = |\{t' < t | y_j^{t'} \neq z_{t'}\}$ is the number of mistakes, which were made by expert $j$ in time $1, \ldots, t-1$). If several experts have the same minimal number of mistakes, we choose that one with a smaller index $j$.

- **Strategy 2:** The forecaster chooses the prediction of expert $j$ with probability

$$\frac{t - m_j}{\sum_{j=1}^{N} (t - m_j)}$$

(i.e. proportional to the number of correct predictions; say in the first iteration, we choose an expert uniformly at random).

Show that both strategies can be much worse (say for $T \gg N$ and suitable $\varepsilon$) than the weighted majority experts algorithm (Algorithm 2 from the lecture).

Solution:

For strategy 1, imagine we have only 2 experts. Expert 1 predicts correct on even days and expert 2 predicts correct on odd days. After odd days, we decide for expert 2 who will be wrong next time. After even days, both expert have the same number of mistakes, hence we decide for expert 1 (which again will be wrong). Hence the forecaster will always be wrong, while both experts are correct half of the time.

For strategy 2, we imagine to have an expert 1 that is always correct. The other experts are all correct exactly on odd days, on even days they are all wrong. Then the probability to choose expert 1 tends to $\frac{t}{t+(N-1)t/2} \leq \frac{2}{N+1}$. Hence, on even days, we will be wrong with probability essentially $1 - \frac{2}{N+1}$. In other words, while there is a perfect expert, the forecaster makes wrong predictions essentially half of the times.
Exercise 2.2 (*)
Consider again the setting with $N$ experts and loss vectors $\ell^t \in [0, 1]^N$. Let $T$ be the number of iterations, $\hat{L}$ be the forecaster's loss and $L_j$ be the loss of expert $j$. In the lecture we saw the bound

$$E[\hat{L}] \leq \frac{\ln(N)}{\epsilon} + (1 + \epsilon)L^j.$$ 

Observe that this just bounds the average loss of the forecaster. Can you give a concentration bound statement of the form $Pr[\hat{L} > (1 + \ldots) \cdot L^j + \ldots] \leq \ldots$. Here the following theorem (a.k.a. Azuma’s Inequality) might be helpful (which you may use without proving it):

*Let $0 = X_0, X_1, \ldots, X_n$ be a sequence of random variables with increment $Y_t := X_t - X_{t-1}$. Here $Y_t := Y_t(X_0, \ldots, X_{t-1})$ might arbitrarily depend on $X_0, \ldots, X_{t-1}$, but always $|Y_t| \leq 1$ and $E[Y_t] = 0$. For $\lambda \geq 0$ one has $Pr[X_n \geq \lambda \sqrt{n}] \leq e^{-\lambda^2/2}$.*

Solution:
Let $Y_t := \hat{\ell}^t - \hat{\ell}^{t-1}$ be the deviation from the average loss in step $t$ and $X_t = \sum_{j=0}^t Y_t$ be the cumulated loss. Note that $E[Y_t] = E[\hat{\ell}^t] - \hat{\ell}^{t-1} = 0$ and $|Y_t| \leq 1$ hence $X_0, \ldots, X_T$ is a martingale. Hence

$$Pr[\bigg\lfloor \frac{X_T}{E[\hat{L}]} \bigg\rfloor \geq \lambda \sqrt{T}] \leq e^{-\lambda^2/2}$$

which implies that

$$Pr[\hat{L} \geq E[\hat{L}] + \lambda \sqrt{T}] \leq e^{-\lambda^2/2}$$

$$\leq \frac{\ln(N)}{\epsilon} + (1 + \epsilon)L^j$$

hence

$$Pr \left[ \hat{L} \geq \frac{\ln(N)}{\epsilon} + \lambda \sqrt{T} + (1 + \epsilon)L^j \right] \leq e^{-\lambda^2/2}$$

Exercise 2.3 (*)
Recall that a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is convex, if $dom(f)$ is a convex set and for all $x, y \in dom(f)$ and $0 \leq \lambda \leq 1$ one has $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$. Prove that if $f_1, \ldots, f_n : K \rightarrow \mathbb{R}$ are convex, $\lambda_1, \ldots, \lambda_n \geq 0$, then also $\sum_{i=1}^n \lambda_i f_i(x)$ is convex.

Solution:
Let $\lambda \in [0, 1]$ and $f(x) := \sum_{i=1}^n \lambda_i f_i(x)$. Then for any $x, y \in K$

$$f(\lambda x + (1-\lambda)y) = \sum_{i=1}^n \lambda_i f_i(\lambda x + (1-\lambda)y) \leq \sum_{i=1}^n \lambda_i (\lambda f_i(x) + (1-\lambda)f_i(y))$$

$$= \lambda \left( \sum_{i=1}^n \lambda_i f_i(x) \right) + (1-\lambda) \left( \sum_{i=1}^n \lambda_i f_i(y) \right) = \lambda f(x) + (1-\lambda)f(y)$$
Exercise 2.4 (*)
Let \( y \in \mathbb{R}^n \) be a vector with \( y_i > 0 \) for all \( i = 1, \ldots, n \) and \( x \in \Sigma^n \). Prove
\[
\| \nabla (-\ln(y^T x)) \|_\infty \leq \max_{i,j} \left| \frac{y_i}{y_j} \right|
\]
Note: The gradient is w.r.t. \( x \) as variable.

Solution:
Note that
\[
dx i (-\ln(y^T x)) = -\ln \left( \sum_{j \neq i} y_j x_j + x_i y_i \right) = -\frac{y_i}{\sum_{j \neq i} y_j x_j + x_i y_i} = -\frac{y_i}{y^T x}
\]
Hence
\[
\| \nabla (-\ln(y^T x)) \|_\infty = \max_{i,j} \left| \frac{\text{max}_i y_i}{y^T x} \right| \leq \max_{i} \frac{\text{max}_i y_i}{\sum_{j=1}^n x_j \min_j y_j} = \max_{i,j} \left| \frac{y_i}{y_j} \right|
\]

Exercise 2.5 (one practical bonus point)
Recall the example from the lecture

<table>
<thead>
<tr>
<th></th>
<th>Stock A</th>
<th>Stock B</th>
<th>Money Market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>2</td>
<td>1.5</td>
<td>1</td>
</tr>
<tr>
<td>Stable</td>
<td>1.2</td>
<td>1.7</td>
<td>1.3</td>
</tr>
<tr>
<td>Down</td>
<td>0.8</td>
<td>1.2</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Implement the presented algorithm to determine an optimum row strategy. Choose \( \varepsilon := 0.1, \delta := 0.2 \) and run the algorithm for \( T = 100 \) iterations.

The details for the submission are as follows:

1. You can implement the algorithm in one of the programming languages C/C++/Java/Pascal/Basic/Matlab (you can choose your favourite one).

2. Your submission should contain your (compilable) code together with an output of the algorithm, which states \( t, w^t, p^t, j_t \) for all iterations \( t = 0, \ldots, 100 \).

3. Send the files till 20.10.10 to thomas.rothvoss@epfl.ch

4. You can work in groups up to 3 people (you need only one submission per group).
Solution:

The sequence of strategies $p^0, \ldots, p^T$ in the simplex $\Sigma^n$ can be visualized as follows (we draw the strategy $p^t$ every 100th iteration).

We can easily convince ourselves that $Q = \left(\frac{2}{7}, 0, \frac{5}{7}\right)$ is an optimal strategy for the row-player and $(0, \frac{4}{7}, \frac{3}{7})$ is an optimal column strategy since

$$
\begin{align*}
(2/7 & \quad 0 \quad 5/7) \cdot \begin{pmatrix} 2 & 1.5 & 1 \\ 1.2 & 1.7 & 1.3 \\ 0.8 & 1.2 & 1.4 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 4/7 \\ 3/7 \end{pmatrix} \\
&= (2/7 \quad 0 \quad 5/7) \cdot \left(9/7, 107/70, 9/7\right) \\
&= (8/7 \quad 9/7 \quad 9/7) \cdot (0, 4/7, 3/7)
\end{align*}
$$

In other words, if the row player starts playing, the (maximizing) column player has no incentive to deviate from his strategy and vice versa. The payoff is 9/7.