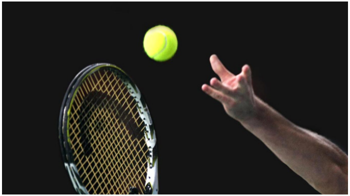
Projet dans le cadre du cours de Machine Learning

Prévision de victoire lors d'un tournoi de tennis (Dubaï)



Joël Cohen, Victorin Martin

Objectifs

Prévoir les vainqueurs des matchs de tennis se déroulant lors d'un tournoi du circuit ATP (ici à Dubaï), en se basant sur des prédictions d'experts.

Introduction

1. Prédiction avec experts ($N = 7$ experts indicés $j \in \{1, \ldots, 7\}$)

Introduction

1. Prédiction avec experts ($N = 7$ experts indicés $j \in \{1, \ldots, 7\}$)
2. 21 matchs ($t \in \{1, \ldots, 21\}$)

Nos experts

3. Classement technique ATP.

Nos experts

1. Classement technique ATP.
2. Classement “The Race” ATP (année en cours)

Nos experts

1. Classement technique ATP.
2. Classement "The Race" ATP (année en cours)
3. Dernière confrontation

Nos experts

1. Classement technique ATP.
2. Classement "The Race" ATP (année en cours)
3. Dernière confrontation
4. Ensemble des confrontations

Nos experts

1. Classement technique ATP.
2. Classement "The Race" ATP (année en cours)
3. Dernière confrontation
4. Ensemble des confrontations
5. Dernière confrontation sur la surface concernée

Nos experts

1. Classement technique ATP.
2. Classement “The Race” ATP (année en cours)
3. Dernière confrontation
4. Ensemble des confrontations
5. Dernière confrontation sur la surface concernée
6. Site de pari sportif

Nos experts

1. Classement technique ATP.
2. Classement “The Race” ATP (année en cours)
3. Dernière confrontation
4. Ensemble des confrontations
5. Dernière confrontation sur la surface concernée
6. Site de pari sportif
7. “Marcel” (féru de tennis)

Cadre

1. Deux joueurs : $Y = \{0, 1\}$

Cadre

1. Deux joueurs : $Y = \{0, 1\}$
2. Prédictions dans $X = [0, 1]$ (CL convexe d'experts)

$$l_1(x_t, y_t) = |x_t - y_t| \qquad \text{(fonction de pertes)}$$

Cadre

1. Deux joueurs : $Y = \{0, 1\}$
2. Prédictions dans $X = [0, 1]$ (CL convexe d'experts)

$$l_1(x_t, y_t) = |x_t - y_t| \qquad \text{(fonction de pertes)}$$

3. Prédictions dans $X = \{0, 1\}$ (tirage aléatoire)

$$l_2(x_t, y_t) = \mathbb{1}_{x_t \neq y_t} \qquad \text{(fonction de pertes)}$$

L'algorithme des "poids exponentiels"

1. $f_{j,t}$ prévision de l'expert j pour le match t (gagnant/perdant)

L'algorithme des “poids exponentiels”

1. $f_{j,t}$ prévision de l'expert j pour le match t (gagnant/perdant)
2. $\mu_{j,t}$ poids de l'expert j pour le match t (fiabilité)

L'algorithme des "poids exponentiels"

1. $f_{j,t}$ prévision de l'expert j pour le match t (gagnant/perdant)
2. $\mu_{j,t}$ poids de l'expert j pour le match t (fiabilité)
3. Initialement,

$$\mu_1 = (\mu_{1,1}, \dots, \mu_{N,1}) = \left(\frac{1}{N}, \dots, \frac{1}{N}\right)$$

L'algorithme des "poids exponentiels"

1. $f_{j,t}$ prévision de l'expert j pour le match t (gagnant/perdant)
2. $\mu_{j,t}$ poids de l'expert j pour le match t (fiabilité)
3. Initialement,

$$\mu_1 = (\mu_{1,1}, \dots, \mu_{N,1}) = \left(\frac{1}{N}, \dots, \frac{1}{N}\right)$$

4. pour $t \geq 2$,

$$\mu_{j,t} = \frac{\exp\left(-\eta \sum_{s=1}^{t-1} l(f_{j,s}, y_s)\right)}{\sum_{k=1}^{N} \exp\left(-\eta \sum_{s=1}^{t-1} l(f_{j,k}, y_k)\right)}$$

Cas $X = [0, 1]$ (doux)

1. Prévision sous la forme $\hat{p}_t = \sum_{k=1}^{N} \mu_{j,t} f_{j,t}$

Cas $X = [0, 1]$ (doux)

1. Prévision sous la forme $\hat{p}_t = \sum_{k=1}^{N} \mu_{j,t} f_{j,t}$
2. Perte du statisticien (nous)

$$\hat{L}_n = \sum_{t=1}^{n} l_1(\hat{p}_t, y_t)$$

Cas $X = [0, 1]$ (doux)

1. Prévision sous la forme $\hat{p}_t = \sum_{k=1}^{N} \mu_{j,t} f_{j,t}$
2. Perte du statisticien (nous)

$$\hat{L}_n = \sum_{t=1}^{n} l_1(\hat{p}_t, y_t)$$

3. Pertes des experts

$$L_{j,n} = \sum_{t=1}^{n} l_1(f_{j,t}, y_t)$$

Cas $X = [0, 1]$ (doux)

1. Prévision sous la forme $\hat{p}_t = \sum_{k=1}^{N} \mu_{j,t} f_{j,t}$
2. Perte du statisticien (nous)

$$\hat{L}_n = \sum_{t=1}^{n} l_1(\hat{p}_t, y_t)$$

3. Pertes des experts

$$L_{j,n} = \sum_{t=1}^{n} l_1(f_{j,t}, y_t)$$

4. Regret

$$R_n = \hat{L}_n - \min_{j \in \{1 \dots N\}} L_{j,n}$$

Cas $X = [0, 1]$ (doux)

1. Comment minimiser le regret R_n ?

Cas $X = [0, 1]$ (doux)

1. Comment minimiser le regret R_n ?
2. On a toujours

$$\sup R_n \leq \frac{\ln N}{\eta} + \frac{\eta n}{8}$$

Cas $X = [0, 1]$ (doux)

1. Comment minimiser le regret R_n ?
2. On a toujours
$$\sup R_n \leq \frac{\ln N}{\eta} + \frac{\eta n}{8}$$
3. Ici $\|l_1\|_\infty \leq 1$, donc calibrage optimal de η avec :
$$\eta_n = \sqrt{\frac{8 \ln N}{n}}$$

Cas $X = [0, 1]$ (doux)

1. Comment minimiser le regret R_n ?
2. On a toujours
$$\sup R_n \leq \frac{\ln N}{\eta} + \frac{\eta n}{8}$$
3. Ici $\|l_1\|_\infty \leq 1$, donc calibrage optimal de η avec :
$$\eta_n = \sqrt{\frac{8 \ln N}{n}}$$
4. On obtient alors la borne :
$$\sup R_n \leq \sqrt{\frac{n \ln N}{2}}$$

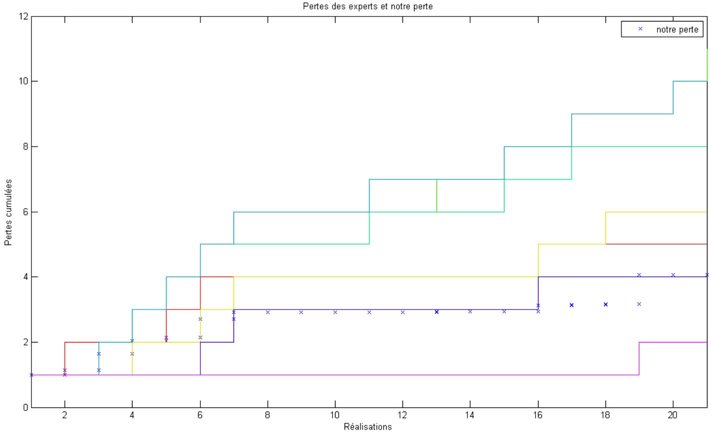
Cas $X = \{0, 1\}$ (dur)

1. Prédictions dans $X = \{0, 1\}$

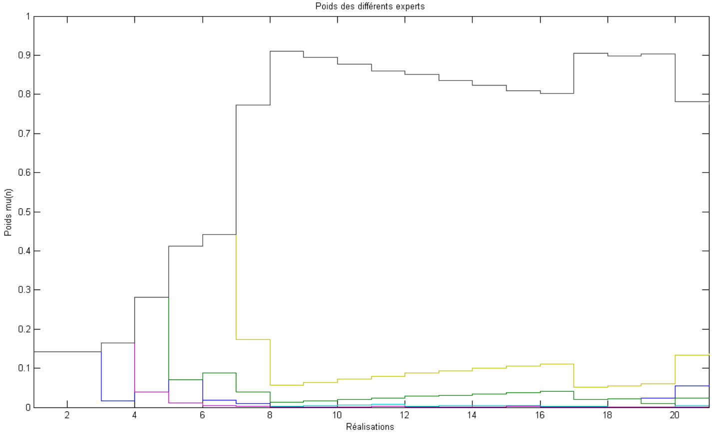
Cas $X = \{0, 1\}$ (dur)

1. Prédictions dans $X = \{0, 1\}$
2. $\Longrightarrow$ tirage aléatoire en fonction des poids $\mu_{j,t}$ et des prévisions $f_{j,t}$ (selon une loi de Bernouilli)

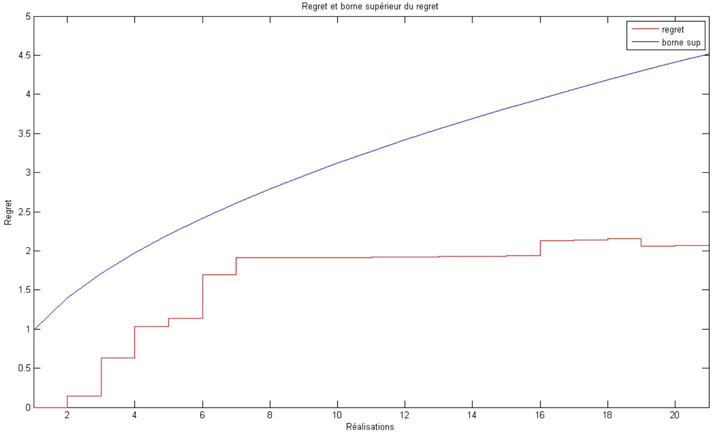
Notre perte (douce) contre celles des experts



Poids des différents experts



Regret (doux) et sa borne supérieur



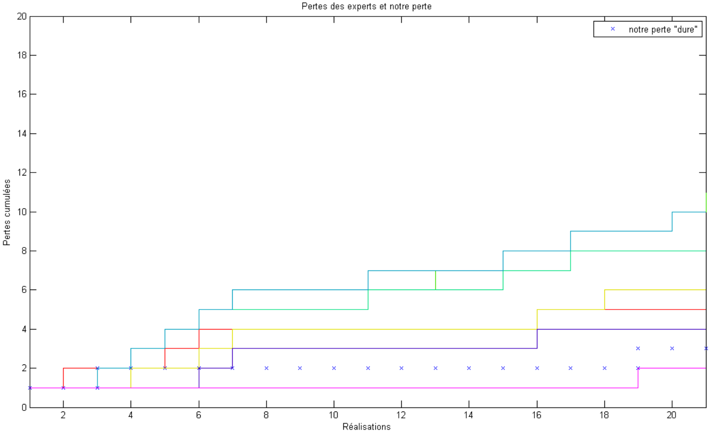
Observations

1. Prépondérance de “Marcel” (expert humain) : poids de 80% ($7^{\text{ième}}$ prévision)

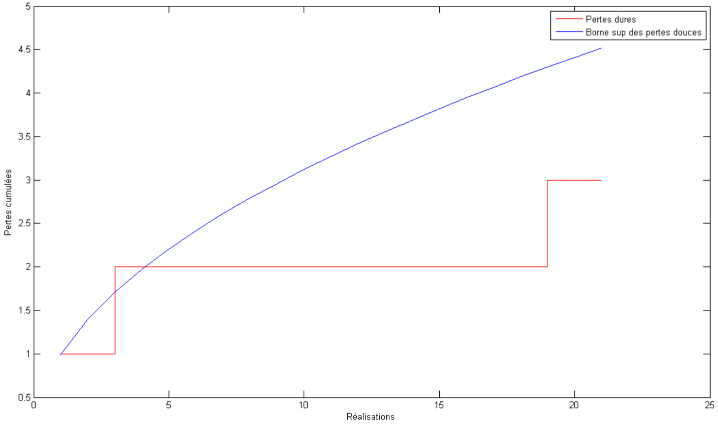
Observations

1. Prépondérance de "Marcel" (expert humain) : poids de 80% ($7^{\text{ième}}$ prévision)
2. Le regret reste bien inférieur à la borne supérieure "ìdéale".

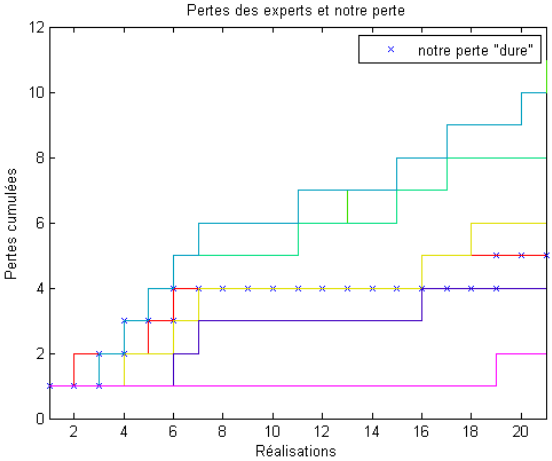
Notre perte (dure) contre celles des experts (essai 1)



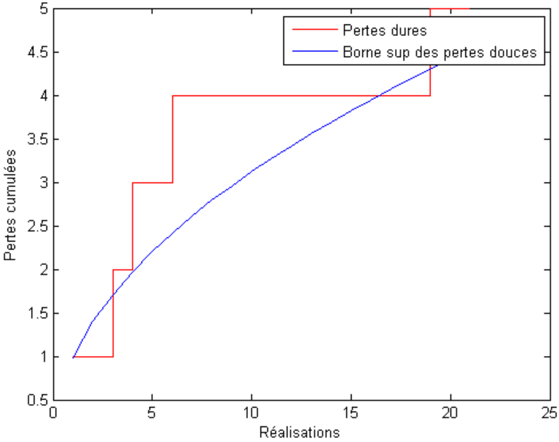
Regret (dur) et borne supérieure du regret (doux) (essai 1)



Notre perte (dure) contre celles des experts (essai 2)



Regret (dur) et borne supérieure du regret (doux) (essai 2)



Observations

1. La borne supérieure du regret n'est pas respectée en général (les poids ne sont jamais strictement nuls)

Conclusion