DEVOIR DE FIN DE MODULE

MODELE ARMA

SUJET

PREVISION DU NOMBRE DE VOYAGEURS

ETUDIANT : **BATONGA Elisabeth-Lucie**

MD4 2022 – 2023

PROFESSEUR : **Hakim HORAIRY**

ECOLE HETIC

13 / 06 / 2023

Table des matières

1. Les modèles utilisés pour prévoir des séries temporelles

Une série temporelle ou une série chronologique est une suite d'observation faites au cours du temps.

Autrement dit, des séries chronologiques sont des données associées à des indices temporels de tout ordre de grandeur: seconde, minute, heure, jour, mois, année, etc.

L'étude des séries temporelles a pour objectif de déterminer le trend et la saisonnalité au cours du temps et de savoir si le processus est linéaire ainsi d'estimer les modèles ARMA et ARIMA et faire des prévisions à la fin.

Exemples de modèles:

- Le modèle AR
- Le modèle MA
- Le modèle ARMA (AutoRegressive Moving Average)
- Le modèle ARIMA (Autoregressive integrated moving average. Ce modèle supprime les effets saisonniers pour dérivation)
- Le modèle SARIMA (seasonal autoregressive integrated moving average : permet de prédire une tendance en intégrant des effets de saisonnalité.)

1.1. Le modèle AR (Auto-Régressif)

Son but est de prédire les valeurs futures en fonction des valeurs présentes. Donc on essaie de prédire une variable d'intérêt en fonction de ses observations passées.

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

Un modèle autorégressif d'ordre *p*, en abrégé AR(*p*), s'écrit :

où $\varphi_1, \ldots, \varphi_p$ sont les paramètres du modèle, c est une constante et ε_t un bruit blanc. La constante est bien souvent omise dans la littérature, le processus étant alors dit centré.

1.2. Le modèle MA (Moyenne mobile)

On essaie de prévoir une variable stochastique en fonction de différentes choses qui peuvent se produire dans différentes périodes.

$$X_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

où les $\theta_1, \ldots, \theta_q$ sont les paramètres du modèle et $\varepsilon_t, \varepsilon_{t-1}, \ldots$ sont encore une fois des termes d'erreur.

1.3. Le modèle ARMA

Le modèle ARMA est un « outil » polyvalent pour l'analyse des séries temporelles. Il permet ainsi de modéliser les dépendances temporelles, de faire des prévisions, de détecter les tendances et d'évaluer la qualité du modèle.

L'utilisation des termes AR & MA pour modéliser les processus temporels : les termes AR capturent les relations entre les observations passées et la valeur actuelle, tandis que les termes MA modélisent les erreurs résiduelles.

$$X_t = \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

où les $\boldsymbol{\varphi_i}$ **et** $\boldsymbol{\theta_i}$ sont les paramètres du modèle et les $\boldsymbol{\varepsilon_i}$ les termes d'erreur.

2. Implémentation du modèle ARMA : Python

L'implémentation du modèle ARMA se fait en plusieurs étapes qui sont :

- Identification de p et q du modèle
- Entraînement du modèle
- Prédictions

2.1. Identification de p et q du modèle

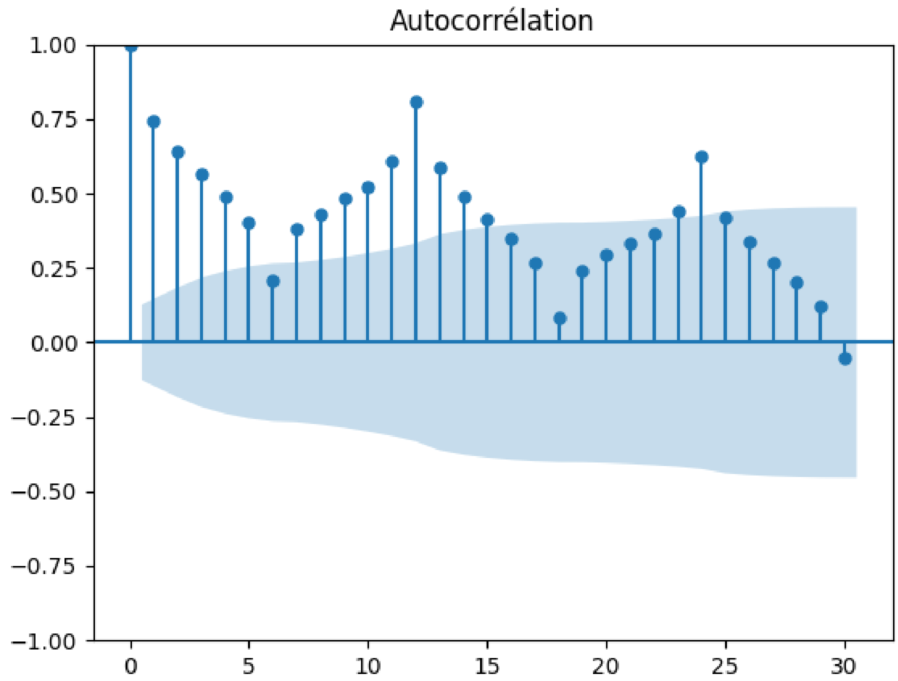
Cette étape implique deux aspects importants :

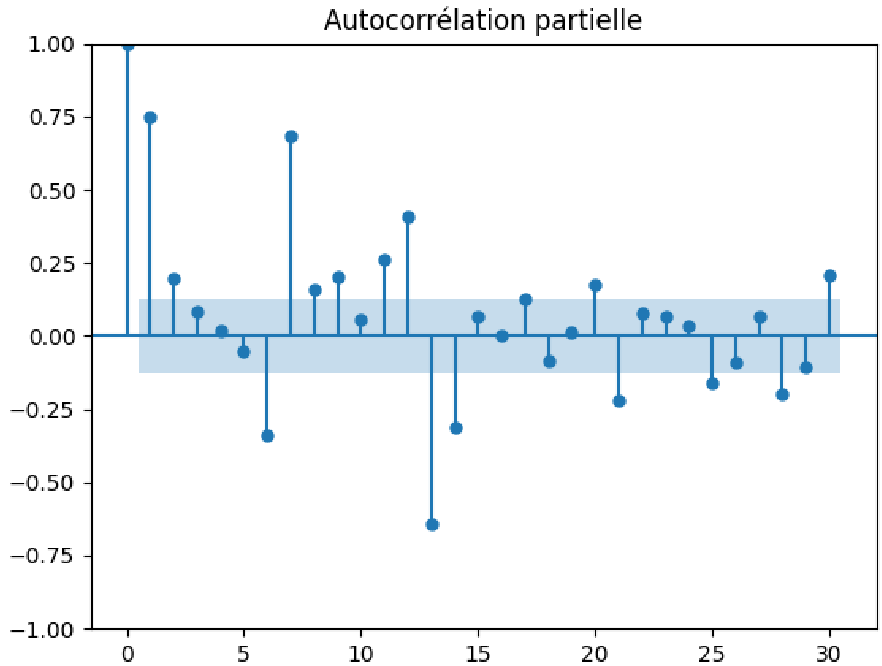
a) Analyse des données temporelles : Il est essentiel de visualiser et d'analyser les données temporelles pour identifier les tendances, les saisons ou les motifs récurrents. Cela peut être fait en traçant des graphiques de la série chronologique et en observant les comportements spécifiques des données.

b) Estimation des ordres p et q : L'autocorrélation et l'autocorrélation partielle sont des outils statistiques couramment utilisés pour estimer les ordres p (nombre de termes autorégressifs) et q (nombre de termes moyens mobiles) du modèle ARMA.

L'autocorrélation mesure la corrélation entre une observation et ses observations précédentes, tandis que l'autocorrélation partielle mesure la corrélation entre une observation et ses observations précédentes en éliminant l'influence des observations intermédiaires.

En identifiant les valeurs significatives de l'autocorrélation et de l'autocorrélation partielle, on peut estimer les ordres p et q appropriés pour le modèle ARMA.





2.2. Entraînement du modèle

Pour les exigences du projet, nous utilisons la descente de gradient pour l'entrainement de notre modèle.

- **La descente de gradient**

Le **gradient** est une généralisation de la **dérivée** pour les fonctions de plusieurs variables.
Plus précisément, le gradient d'une fonction $f(x_1, x_2, \ldots, x_n)$ est un vecteur dont chaque composante est la **dérivée partielle** de la fonction par rapport à une des variables x_i.

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$$

Une façon de visualiser cela est d'imaginer des courbes d'équipotentielles.
Ces courbes représentent les ensembles de points où la fonction prend la même valeur.
Ce dessous un exemple :

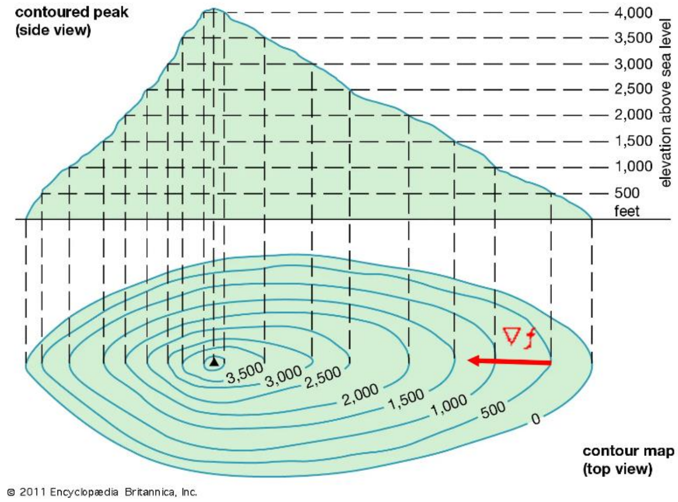


figure 4 : courbe equi-altitude [2]

Le gradient de la fonction est alors perpendiculaire à chaque courbe d'équipotentielles

Inégalité de la descente de gradient

Soit :

- f une fonction de $\mathbb{R}^n \rightarrow \mathbb{R}$
- $\alpha > 0$, le Learning Rate

On a :

$$f(x - \alpha\nabla f(x)) < f(x)$$

Cela veut dire qu'à partir d'une entrée (un vecteur que l'on passe en argument de f), si l'on se déplace dans le sens opposé du gradient de la fonction f, on obtient une image par f plus faible.
Ainsi pour **minimiser** une fonction on peut imaginer une **heuristique itérative** qui consiste à **démarrer** à partir d'une **entrée aléatoire**, actualiser l'entrée en **se déplaçant dans le sens opposé du gradient** un certain nombre de fois.

2.3. Prédictions

Prédiction des valeurs futures : Le modèle ARMA permet de prédire les valeurs futures en utilisant les valeurs passées et les résidus du modèle. Les prédictions peuvent être générées pour une ou plusieurs périodes futures.

Évaluation de la performance : Pour évaluer la performance du modèle, les valeurs prédites peuvent être comparées aux valeurs réelles à l'aide de métriques d'évaluation telles que l'erreur quadratique moyenne (MSE) :

$$\mathrm{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(Y_i - \hat{Y_i} \right)^2 .$$

Pour les prédictions avec le modèle ARMA, nous utilisons la fonction **arma_model _forecast.**

Voilà comment se présente cette fonction :

```
# calcul les prédictions du modèle ARMA
       def arma_model_forecast(feature):
           ar_term = np.dot(self.parameters[:p], feature)
           ma_term = np.dot(self.parameters[p:p+q], self.residuals[-q:])
           return ar_term + ma_term + self.parameters[-1]
```

Cette fonction **arma_model_forecast** est appelée et appliquée à chaque feature du jeu de données. Ainsi, on a :

```
arma_model_forecast = lambda feature: np.dot(self.parameters[:p], feature) + np.dot(self.parameters[p:p+q], self.residuals

train_predictions = list(map(arma_model_forecast, self.t_train_arma))
validation_predictions = list(map(arma_model_forecast, self.t_validation_arma))
test_predictions = list(map(arma_model_forecast, self.t_test_arma))
```

Conclusion

Le modèle ARMA est un outil puissant qui permet de modéliser et de prédire les phénomènes temporels de manière précise. En combinant les termes autorégressifs et les termes moyens mobiles, il capture les structures complexes des données.

Pour utiliser efficacement le modèle ARMA, il est essentiel de bien comprendre les termes AR et MA, ainsi que les différentes étapes clés telles que l'identification du modèle, l'estimation des paramètres, le diagnostic et l'évaluation.