Introduction au traitement de données RADAR

Léo Monnier

Laboratoire d'intelligence artificielle THALES DMS
Université Technologique de Troyes

January 30, 2025

Overview

Introduction

Les sujets traités au laboratoire IA TDMS

- Les physics informed neural networks (PINN).
- Le désentrelacement.
- La recherche d'architecture de deep learning.
- Les transformers.
- L'ontologie.
- L'algorithmie quantique.
- La guerre electronique.
- Le traitement du signal radar.

Le RADAR

Qu’est ce qu’un RADAR?

Le radar (RAdio Detection And Ranging, en anglais) est un dispositif technologique utilisé pour détecter, localiser et suivre des objets à distance, généralement dans l’air, sur terre ou en mer. Son fonctionnement repose sur l’émission et la réception d’ondes radio

Figure: Figure RBE2 AESA

Principe de fonctionnement

1. Émission d'ondes radio : Le radar émet un faisceau d'ondes radio dans une direction spécifique à l'aide d'une antenne. Ces ondes se déplacent à la vitesse de la lumière (environ 300 000 km/s).
2. Réflexion des ondes : Lorsqu'une onde radio rencontre un objet (comme un avion, un bateau ou une voiture), une partie de l'énergie est réfléchie vers l'antenne du radar.
3. Réception et analyse : Le radar capte les ondes réfléchies (échos) grâce à une antenne réceptrice. En mesurant le délai entre l'émission et la réception des ondes, il peut calculer la distance entre le radar et l'objet détecté.
4. Traitement des données :
   - Distance (portée) : Déduite du temps que mettent les ondes à revenir.
   - Direction (azimut) : Déterminée par l'orientation de l'antenne.
   - Vitesse de l'objet : Calculée grâce à l'effet Doppler, qui mesure le changement de fréquence des ondes réfléchies en fonction du mouvement relatif de l'objet.

Types de radars

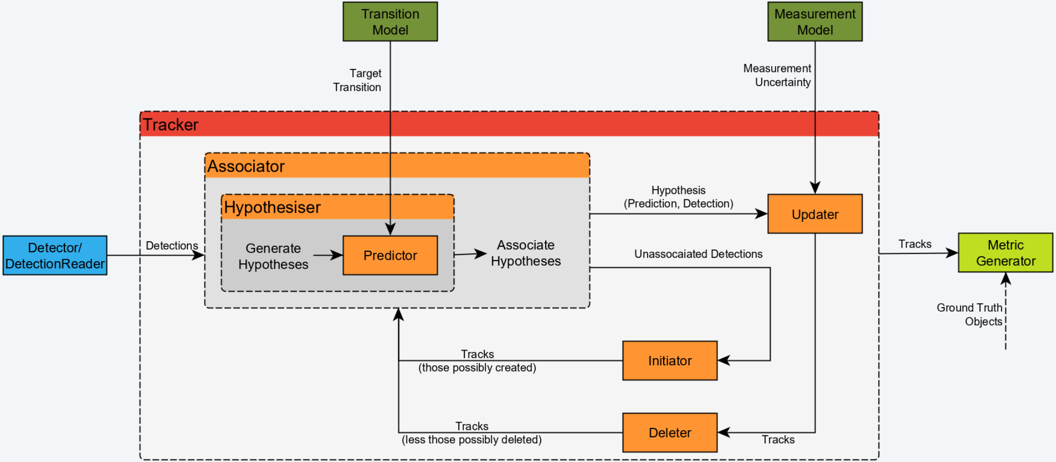
- Radar primaire : Détecte les objets passivement, en analysant uniquement les ondes réfléchies.
- Radar secondaire : Utilisé notamment en aviation, il interagit avec des transpondeurs installés dans les avions pour obtenir des informations précises comme l'identité ou l'altitude.

Applications

1. Militaire :
   - Détection d’avions, missiles ou navires ennemis.
   - Surveillance du champ de bataille.
2. Aviation :
   - Contrôle du trafic aérien.
   - Gestion des décollages et atterrissages.
   - visualisation en vol
3. Navigation maritime :
   - Éviter les collisions en mer.
   - Localiser des obstacles ou d’autres navires.
4. Météorologie :
   - Observation des précipitations (pluie, neige, etc.).
   - Prévision des tempêtes et cyclones.
5. Surveillance routière :
   - Mesure de la vitesse des véhicules pour identifier les excès de vitesse.

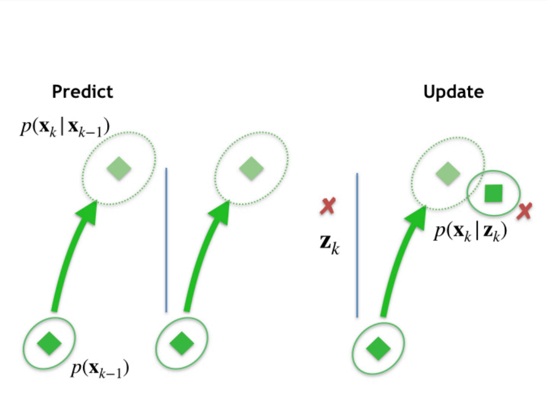
Le Tracking

Qu’est ce qu’un tracker?



L’idée du filtre de kalman

Idée générale



Principe Mathématique

Soit $p(\mathbf{x}_k)$, la distribution de probabilité sur $\mathbf{x}_k \in \mathbb{R}^n$, représentant un état caché à un instant discret k. Le terme k est le plus souvent interprété comme un pas de temps, mais il peut aussi représenter tout autre indice séquentiel (c'est pourquoi nous n'utilisons pas t). La mesure associée est donnée par $\mathbf{z}_k \in \mathbb{R}^m$.

Notre objectif est d'inférer $p(\mathbf{x}_k)$, étant donné une séquence de mesures $\mathbf{z}_1, ..., \mathbf{z}_k$ (notée $\mathbf{z}_{1:k}$). En général, $\mathbf{z}$ peut inclure du bruit et de fausses alarmes. Nous reporterons ces complications à des tutoriels ultérieurs et supposerons, pour le moment, que toutes les mesures sont générées par une cible et que la détection de la cible à chaque pas de temps est certaine ($p_d = 1$ et $p_{fa} = 0$).

Prediction

Nous faisons l'hypothèse **markovienne** selon laquelle :

$$p(\mathbf{x}_k) = \int_{-\infty}^{\infty} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1})d\mathbf{x}_{k-1}$$

Cela signifie que la distribution de l'état d'un objet à l'instant k peut être prédite uniquement à partir de son état à l'instant précédent $k-1$. Si notre connaissance de $p(\mathbf{x}_{k-1})$ est informée par une série de mesures jusqu'à l'instant $k-1$, nous pouvons écrire :

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int_{-\infty}^{\infty} p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}$$

C'est ce qu'on appelle l'**équation de Chapman-Kolmogorov**.

Prediction dans stonesoup

Dans *Stone Soup*, ce processus est appelé **prédiction**, et l'objet qui l'effectue est un `Predictor`. Un prédicteur a besoin d'un **modèle de transition d'état**, c'est-à-dire une fonction qui réalise :

$$\mathbf{x}_{k|k-1} = f(\mathbf{x}_{k-1}, \mathbf{w}_k)$$

où $\mathbf{w}_k$ est un terme de bruit. *Stone Soup* propose des modèles de transition dérivés de la classe `TransitionModel`.

Mesure

On suppose qu'une mesure capteur est générée par un processus stochastique représenté par une fonction :

$$\mathbf{z}_k = h(\mathbf{x}_k, \boldsymbol{\nu}_k)$$

où $\boldsymbol{\nu}_k$ est un bruit.
L'objectif du processus de mise à jour est de générer **l'estimation d'état a posteriori**, à partir de la prédiction et de la mesure.

Mesure

Cela se fait grâce à la **règle de Bayes** :

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}$$

où :

- $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ est la sortie de l'étape de prédiction,
- $p(\mathbf{z}_k|\mathbf{x}_k)$ est la **vraisemblance**,
- $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ est **l'évidence**.

Dans *Stone Soup*, ce calcul est réalisé par la classe `Updater`. Les *Updaters* utilisent une classe `MeasurementModel`, qui modélise l'effet de $h(\cdot)$.

Les composants d'un filtre de kalman

Predicteur

Un prédicteur est utilisé pour prédire un nouvel état donné un état a priori et un modèle de transition. De plus, un modèle de contrôle peut être utilisé pour modéliser une influence externe sur l'état.

$$\mathbf{x}_{k|k-1} = f_k(\mathbf{x}_{k-1}, \nu_k) + b_k(\mathbf{u}_k, \eta_k) \quad (1)$$

où $\mathbf{x}_{k-1}$ est l'état a priori, $f_k(\mathbf{x}_{k-1})$ est la fonction de transition, $\mathbf{u}_k$ est le vecteur de contrôle, $b_k(\mathbf{u}_k)$ l'entrée de contrôle, et ν_k et η_k représentent respectivement le bruit des modèles de transition et de contrôle.

Prédicteur de Kalman

Ici, les modèles de transition et de contrôle doivent être linéaires :

$$f_k(\mathbf{x}_{k-1}, \nu_k) = F_k \mathbf{x}_{k-1} + \nu_k, \quad \nu_k \sim \mathcal{N}(0, Q_k)$$

$$b_k(\mathbf{u}_k, \eta_k) = B_k(\mathbf{u}_k + \eta_k), \quad \eta_k \sim \mathcal{N}(0, \Gamma_k).$$

Updater

Un updater est utilisé pour mettre à jour l'état prédit en utilisant une mesure et un modèle de mesure. Le modèle d'observation général est :

$$\mathbf{z} = h(\mathbf{x}, \sigma) \tag{2}$$

où $\mathbf{x}$ est l'état, ω le bruit de mesure et $\mathbf{z}$ la mesure obtenue.

Kalman updater

Les mise à jour de Kalman supposent que :

$$h(\mathbf{x}) = H\mathbf{x}$$

avec un bruit additif

$$\sigma = \mathcal{N}(0, R).$$

`update` appelle d'abord la fonction `predict_measurement` qui procède en calculant la mesure prédite, la covariance de l'innovation et la covariance croisée de la mesure,

$$\mathbf{z}_{k|k-1} = H_k \mathbf{x}_{k|k-1}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

Kalman updater

Le gain de Kalman est ensuite calculé comme suit,

$$K_k = \Upsilon_k S_k^{-1}$$

et la moyenne et la covariance de l'état posteriori sont,

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + K_k(\mathbf{z}_k - H_k \mathbf{x}_{k|k-1})$$

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T$$

Associer des trajectoires

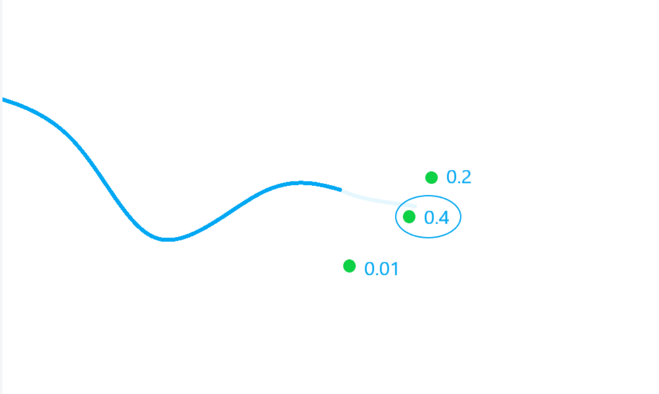
Le problème du bruit

Le suivi d'une cible est souvent compliqué par la présence de détections qui ne sont pas associées à la cible d'intérêt. Celles-ci peuvent provenir du bruit généré par le capteur, de réflexions sur des objets intermédiaires ou d'effets environnementaux. Nous les désignons collectivement sous le terme de *fouillis* lorsqu'elles n'ont qu'une valeur parasite et doivent être filtrées.
Une métrique de distance appropriée pour les états décrits par des distributions gaussiennes est la distance de Mahalanobis. Celle-ci quantifie la distance d'un point par rapport à une distribution donnée. Dans le cas d'un point $\mathbf{x}$, et d'une distribution avec une moyenne μ et une matrice de covariance Σ, la distance de Mahalanobis de $\mathbf{x}$ par rapport à la distribution est donnée par :

$$d_M(\mathbf{x}) = \sqrt{(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)}$$

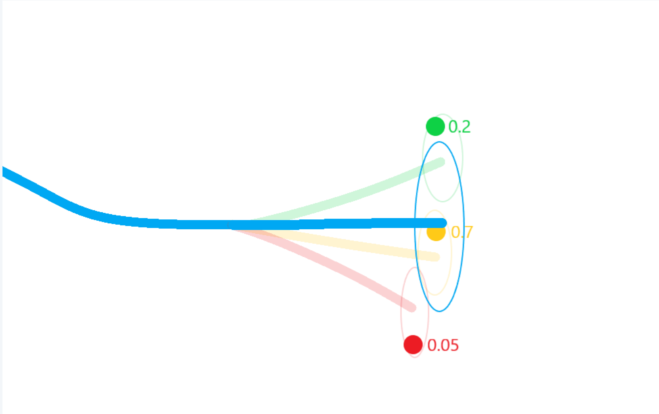
Le problème du bruit

Exemple d'association basique.



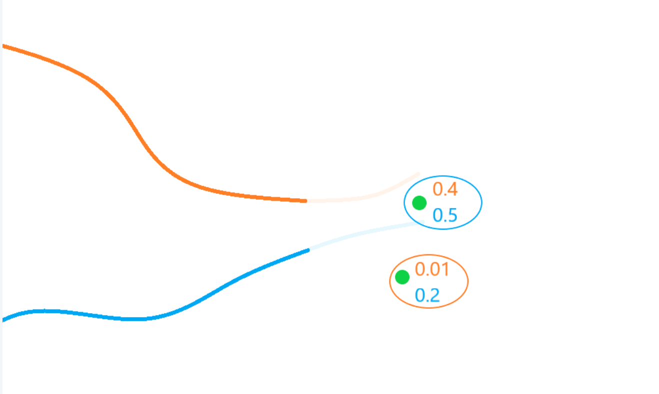
Le problème du bruit

Exemple par association probabilistique.



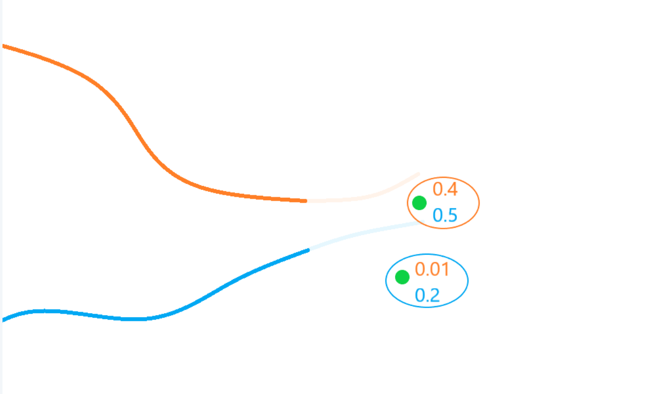
Le cas pour plusieurs trajectoires

Association avec Nearest Neigbours



Le cas pour plusieurs trajectoires

Association Global Nearest Neigbours



Le cas pour plusieurs trajectoires

Par calcul des probabilités jointes