

Un algorithme EM récursif pour le SLAM

Sylvain LE CORFF, Gersende FORT, Eric MOULINES

LTCI, TELECOM ParisTech & CNRS
46, rue Barrault, 75634 Paris Cedex 13, France
prenom.nom@telecom-paristech.fr

Résumé – Nous proposons un algorithme de type Expectation Maximisation (EM) récursif pour l’inférence dans les modèles de Chaînes de Markov cachées (HMM) à espace d’état général. Cet algorithme permet de traiter les situations où les observations arrivent séquentiellement et où le modèle complet n’est pas de type exponentiel. Nous appliquons cet algorithme au problème de Localisation et Cartographie Simultanées (SLAM) lorsque les modèles d’état et d’observations sont non linéaires et non gaussiens. Pour ce faire, nous modélisons la carte comme un paramètre du HMM. L’originalité de cette contribution à la résolution du SLAM réside à la fois (i) dans la formulation du problème de cartographie comme un problème d’estimation de paramètres dans un HMM; et (ii) dans l’approche algorithmique retenue, basée sur l’algorithme EM.

Abstract – We propose a new online Expectation Maximization (EM)-based algorithm for inference in general state space Hidden Markov models (HMM). This algorithm applies when observations are sequential and when the complete likelihood is not in the exponential distribution class. This algorithm is applied to the Simultaneous Localisation and Mapping (SLAM) problem when the state and observation models are non linear and non gaussian. To that goal, the map is seen as a parameter of the HMM. The novelty of our approach for the SLAM problem is (i) in the expression of the mapping problem as an inference problem in HMM and (ii) the algorithm used for solving inference, based on EM.

1 Introduction

L’algorithme EM [1] est un algorithme très répandu pour la résolution de problèmes inverses dans des modèles à données cachées. Cet algorithme a tout d’abord été proposé pour l’estimation “hors ligne” i.e. dans le cas où toutes les observations sont disponibles avant de lancer la procédure. Néanmoins, l’intérêt pour des procédures de résolution de problèmes inverses “en ligne” est croissant du fait par exemple de la taille importante des données qui empêche leur stockage et/ou de la nécessité d’une réponse en temps réel. Ainsi, des procédures de type *EM récursif* i.e. des algorithmes de type EM capables d’intégrer les nouvelles observations au fur et à mesure de leur disponibilité, ont été récemment proposées (par [2] dans le cas d’observations i.i.d.; par [3, 4] pour les HMM à espace d’état fini; par [5] pour les modèles de Markov cachés généraux, lorsque la vraisemblance complète est une distribution exponentielle).

Nous proposons une autre extension de ces méthodes au cas des HMM à espace d’état général qui permet aussi de traiter les HMM dont la vraisemblance complète n’est pas exponentielle. Dans ce contexte de HMM non linéaire, non nécessairement gaussien et à espace d’état général, la principale difficulté est de définir l’étape E de l’algorithme EM récursif. Pour ce faire, nous combinons (i) une approximation (locale) de la loi jointe par une distribution dans la famille exponentielle; et (ii) un algorithme de calcul récursif de l’espérance de la statistique suffisante sous la loi a posteriori, méthode adaptée de [6].

Ces extensions sont motivées par la problématique du SLAM : un robot se déplace et observe des amers; la carte (i.e. la po-

sition des amers et leurs nombres) ainsi que la trajectoire du robot sont inconnues et doivent être reconstruites à l’aide des observations. Dans les approches classiques, le problème du SLAM est modélisé via un HMM tel que la carte et la position du robot sont intégrées à l’état caché. Les premières réponses ont été apportées pour des HMM linéaires gaussiens (via le filtre de Kalman; puis l’algorithme de Kalman étendu pour l’approximation de HMM par un modèle linéaire gaussien). Néanmoins ces algorithmes souffrent d’inconsistance notamment du fait de l’existence d’une composante fixe (la carte) dans l’état caché [7, 8]. Récemment, [9] a proposé de voir la carte comme un paramètre du HMM, l’état caché se réduisant à la pose du robot. La carte est alors estimée au sens du maximum de vraisemblance récursif résolu en pratique par un algorithme de gradient stochastique, et la distribution de la pose est approchée par une méthode particulière. Nous nous plaçons dans le même modèle que [9] et résolvons le problème inverse en appliquant l’algorithme EM récursif pour les HMM à espace d’état général. Nous obtenons une estimation ponctuelle de la carte - via un algorithme itératif -, et une approximation particulière de la distribution a posteriori de la pose.

En section 2, nous présentons les algorithmes EM en ligne pour les HMM puis exposons notre nouvel algorithme (section 2.3). Nous considérons l’application au SLAM en section 3.

2 Algorithme de type EM récursif pour l'inférence dans les HMM

Étant données des observations $Y_{1:T} = (Y_1, \dots, Y_T)$, on ajuste un modèle de Chaînes de Markov cachées dépendant d'un paramètre $\theta \in \Theta \subseteq \mathbb{R}^\ell$, en déterminant le meilleur modèle dans cette famille au sens du maximum de vraisemblance. Les modèles HMM sont définis par la donnée (a) d'une loi initiale de l'état caché, que l'on va considérer indépendante de θ pour simplifier l'exposé, (b) d'un noyau de transition $m_\theta(x, x')d\lambda(x')$ de l'état caché et, (c) d'une densité conditionnelle de l'observation étant donné l'état caché $g_\theta(x, y)d\nu(y)$. Notons $\{(X_t, Y_t)\}_{t \geq 1}$ un HMM défini par ces noyaux de transition, à valeurs dans des espaces généraux $\mathbb{X} \times \mathbb{Y}$.

2.1 EM hors ligne - Cas des HMM exponentiels

Les HMM sont des modèles à données latentes et pour ces modèles, l'algorithme EM permet le calcul du maximum de vraisemblance ? Il consiste à définir de façon itérative une suite $\{\theta_n\}_{n \geq 0}$, qui converge vers un point stationnaire de la vraisemblance. Chaque itération comprend deux étapes :

Étape E Calcul de

$$Q_T(\theta, \theta_n) \stackrel{\text{def}}{=} \mathbb{E}_{\theta_n} \left[\sum_{t=1}^T \log \{ m_\theta(X_{t-1}, X_t) g_\theta(X_t, Y_t) \} \middle| Y_{1:T} \right]$$

où $\mathbb{E}_{\theta}[\cdot | Y_{1:T}]$ désigne l'espérance sous la loi conditionnelle des données manquantes $X_{1:T}$ sachant $Y_{1:T}$ pour la valeur θ du paramètre.

Étape M Mise à jour de l'estimée courante :

$$\theta_{n+1} \in \operatorname{argmax}_{\theta \in \Theta} Q_T(\theta, \theta_n).$$

Excepté dans des modèles HMM simples, la quantité $Q_T(\cdot, \theta_n)$ n'est pas calculable explicitement car la loi conditionnelle $\phi_{1:T|T}^\theta$ de $X_{1:T}$ sachant $Y_{1:T}$ pour la valeur θ du paramètre (i.e. la loi de *lissage*) n'est pas connue explicitement. Néanmoins, du fait du caractère markovien et donc récursif de la dynamique de construction du HMM, il existe une relation de récurrence liant la distribution $\phi_{1:T-1|T-1}^\theta$ à $\phi_{1:T|T}^\theta$, relation que l'on peut exploiter pour construire une approximation particulière de la loi d'intérêt $\phi_{1:T|T}^\theta$ (voir e.g. [10]) ; dans ce cas, l'étape E est approchée par une méthode de Monte Carlo Séquentielle.

L'étape M nécessite de calculer la quantité intermédiaire $\theta \mapsto Q_T(\theta, \theta_n)$ pour toutes les valeurs de θ , c'est-à-dire calculer une infinité d'espérances sous la loi conditionnelle étant donnée la valeur courante du paramètre. Dans les applications réelles, cela est impossible à mettre en oeuvre sauf dans le cas où la vraisemblance complète est une distribution exponentielle : il existe des fonctions $\psi : \Theta \rightarrow \mathbb{R}^d$ et $S : \mathbb{X} \times \mathbb{X} \times \mathbb{Y} \rightarrow \mathbb{R}^d$ telles que

$$\log \{ m_\theta(x, x') g_\theta(x', y) \} = \langle S(x, x', y), \psi(\theta) \rangle,$$

où $\langle \cdot, \cdot \rangle$ désigne le produit scalaire. Dans ce cas, l'étape E consiste à calculer la quantité

$$\bar{S}_{1:T}(\theta_n, Y_{1:T}) \stackrel{\text{def}}{=} \mathbb{E}_{\theta_n} \left[\sum_{t=1}^T S(X_{t-1}, X_t, Y_t) \middle| Y_{1:T} \right],$$

et l'étape M devient $\theta_{n+1} \in \operatorname{argmax}_{\theta} Q_T(\theta, \theta_n)$ où

$$Q_T(\theta, \theta_n) = \langle \bar{S}_{1:T}(\theta_n, Y_{1:T}), \psi(\theta) \rangle.$$

Nous qualifierons ces modèles de *HMM exponentiels*.

2.2 Algorithme EM en ligne pour des HMM exponentiels

Pour résoudre un problème d'estimation par maximum de vraisemblance récursif, il faut introduire des procédures "en ligne" qui ne nécessitent ni le stockage des observations, ni l'utilisation des observations sur plusieurs itérations successives. Ces méthodes doivent être capables d'intégrer la nouvelle information disponible Y_{T+1} aux quantités disponibles à l'instant T . Ainsi, l'algorithme *EM en ligne* consiste à définir une suite $\{\theta_T\}_{T \geq 0}$ définie par

$$\theta_{T+1} \in \operatorname{argmax}_{\theta \in \Theta} Q_{T+1}(\theta, \theta_T).$$

Cette méthode n'est récursive que si le calcul de $Q_{T+1}(\theta, \theta_T)$ est récursif. Observons que dans les HMM exponentiels,

$$\begin{aligned} & \bar{S}_{1:T+1}(\theta_T, Y_{1:T+1}) \\ &= \mathbb{E}_{\theta_T} \left[\mathbb{E}_{\theta_T} \left[\sum_{t=1}^T S(X_{t-1}, X_t, Y_t) \middle| Y_{1:T}, X_{T+1} \right] \middle| Y_{1:T+1} \right] \\ & \quad + \mathbb{E}_{\theta_T} \left[S(X_T, X_{T+1}, Y_{T+1}) \middle| Y_{1:T+1} \right] \end{aligned} \quad (1)$$

de sorte que le calcul de $Q_{T+1}(\theta, \theta_T)$ nécessite de connaître une approximation (i) de l'espérance conditionnelle

$$\mathbb{E}_{\theta_T} \left[\sum_{t=1}^T S(X_{t-1}, X_t, Y_t) \middle| Y_{1:T}, X_{T+1} \right]$$

et (ii) de la loi de filtrage. L'espérance conditionnelle se calcule récursivement (voir e.g. [5, 4]). La loi de filtrage à l'instant $(T + 1)$ peut aussi être approchée récursivement par des méthodes de Monte Carlo séquentielles (SMC, voir e.g. [10]). Néanmoins, le calcul de ces lois relatives à l'instant $(T + 1)$ pour la valeur θ_T du paramètre nécessite la connaissance de ces lois relatives à l'instant T pour la même valeur θ_T du paramètre. Dans l'itération de l'EM, la propagation de ces lois de T vers $T + 1$ ne peut donc pas être menée à l'aide des quantités disponibles à l'instant T qui reposent toutes sur la valeur courante du paramètre θ_{T-1} . Il faut donc introduire des approximations de propagation : substituer les lois relatives à l'instant T pour la valeur θ_T du paramètre par les lois relatives à l'instant T pour la valeur θ_{T-1} .

Ces approximations de propagation de loi et la décomposition (1) sont à l'origine des algorithmes *EM en ligne pour les HMM exponentiels* proposés par [4] pour les HMM à espace d'état fini, et par [5, 6] pour les HMM à espace d'état général.

2.3 Nouvel algorithme : EM en ligne pour les HMM

L'algorithme que nous proposons étend les procédures proposées par [5, 6] ; cette extension indique comment exploiter les procédures pour des HMM exponentiels lorsque le modèle HMM considéré n'est pas exponentiel. Cette extension, motivée par la problématique du SLAM (voir section 3) mime ce qui est proposé dans la littérature SLAM probabiliste pour passer d'un modèle HMM non linéaire (et gaussien) à un modèle linéaire (gaussien) afin de résoudre le SLAM par des techniques de Kalman.

La solution que nous proposons repose sur une approximation de la quantité $\theta \mapsto \log\{m_\theta(x, x')g_\theta(x', y)\}$ par un modèle exponentiel. Cette approximation est "locale" au sens où elle est faite à chaque itération de l'algorithme autour de la valeur courante du paramètre. Par suite, notre algorithme consiste à définir une suite $\{\theta_T\}_{T \geq 0}$ de façon itérative, en répétant les trois étapes suivantes. Étant donnée la valeur courante du paramètre θ_T ,

Étape d'approximation par modèle exponentiel : approcher $\log\{m_\theta(x_{t-1}, x_t)g_\theta(x_t, Y_t)\}$ par $\langle \mathcal{S}(x_{t-1}, x_t, Y_t; \theta_T), \psi(\theta) \rangle$ et en déduire une approximation de la log-vraisemblance complète $\sum_{t=1}^{T+1} \log\{m_\theta(x_{t-1}, x_t)g_\theta(x_t, Y_t)\}$ de la forme :

$$\left\langle \sum_{t=1}^{T+1} \mathcal{S}(x_{t-1}, x_t, Y_t; \theta_T), \psi(\theta) \right\rangle .$$

Étape E de type SMC : approcher l'espérance conditionnelle

$$\mathbb{E}_{\theta_T} \left[\sum_{t=1}^{T+1} \mathcal{S}(X_{t-1}, X_t, Y_t; \theta_T) \middle| Y_{1:T+1} \right]$$

en utilisant la procédure itérative décrite dans la section 2.2. Notons \tilde{S}_{T+1} cette approximation.

Étape M : mettre à jour le paramètre

$$\theta_{T+1} \in \arg\max_{\theta \in \Theta} \langle \tilde{S}_{T+1}, \psi(\theta) \rangle .$$

Nous donnons dans la section suivante des exemples d'approximation par modèle exponentiel de type quadratique. Puisque la première étape est une approximation locale, on peut substituer l'étape M par une étape de maximisation locale, l'intuition étant que l'approximation du modèle est d'autant meilleure que l'on est dans un voisinage du paramètre courant.

3 Application au SLAM

Un robot évolue dans un environnement (plan) inconnu. A chaque instant t , il observe q_t amers ; $Y_t = (Y_{t,i}, i \in \mathcal{A}_t)$ collecte les q_t mesures de distances et d'angles d'observation de ces amers. Il se déplace selon une dynamique markovienne guidée par des contrôles u_t (connus), et indépendante de la position des amers. L'état caché $X_t \in \mathbb{R}^2 \times [0, 2\pi]$ collecte la pose du robot à l'instant t (coordonnées cartésiennes) et son orientation. L'objectif est d'estimer la position des amers i.e. pour chaque amer i , le vecteur $\theta_i \in \mathbb{R}^2$ de ses coordonnées

cartésiennes ; et la pose du robot. On modélise ces observations par un HMM de la forme

$$\begin{aligned} X_t &= f(X_{t-1}, V_t, u_t) \\ \forall i \in \mathcal{A}_t, Y_{t,i} &= h(X_t, \theta_i) + W_{t,i}, \end{aligned}$$

où $\{(V_t, W_{t,i}), t \geq 0, i \in \mathcal{A}_t\}$ sont des bruits centrés i.i.d. de loi connue. Avec ce choix de modèle, la transition de l'état caché $m_\theta(x, x')$ est indépendante de θ ; par suite, la quantité intermédiaire de l'EM s'exprime uniquement en fonction de $\log g_\theta(X_t, Y_t)$ - seuls les termes dépendant de θ comptant dans l'étape M - et la structure exponentielle du modèle ne repose que sur la forme de la loi conditionnelle $\log g_\theta$. Dans les applications numériques, on prend pour modèle $(x, v \in \mathbb{R}^3, \tau, u \in \mathbb{R}^2)$

$$\begin{aligned} f(x, v, u) &= x + \begin{pmatrix} (u_1 + v_1) \delta \cos(x_3 + u_2 + v_2) \\ (u_1 + v_1) \delta \sin(x_3 + u_2 + v_2) \\ (u_1 + v_1) \delta B^{-1} \sin(u_2 + v_2) + v_3 \end{pmatrix}, \\ h(x, \tau) &= \begin{pmatrix} \sqrt{(\tau_1 - x_1)^2 + (\tau_2 - x_2)^2} \\ \arctan\left(\frac{\tau_2 - x_2}{\tau_1 - x_1} - x_3\right) \end{pmatrix}, \end{aligned}$$

où δ est le temps entre deux mesures successives et B est l'empattement. On envisage successivement deux lois pour le bruit d'observation $W_{t,i}$: tout d'abord un modèle gaussien puis un modèle de mélange de deux lois gaussiennes pour définir un modèle plus robuste aux observations aberrantes. Dans les deux cas, les lois des bruits sont connues. Avec ces choix, le modèle n'est pas de type "HMM exponentiel" ; on applique donc l'algorithme décrit en section 2.3. Pour ce faire, on utilise l'approximation quadratique de $\theta \mapsto \log g_\theta(x, y)$ issue de l'approximation à l'ordre 1 de $\theta \mapsto h(x, \theta)$ autour de l'estimée courante θ_T : $h(x, \theta) \approx h(x, \theta_T) + \langle (\theta - \theta_T), \nabla_\theta h(x, \theta_T) \rangle$. Nous suivons en cela la démarche utilisée dans les algorithmes basés sur des extensions du filtre de Kalman (e.g. l'EKF).

L'application de l'algorithme présenté en section 2.3 permet de répondre à la problématique du SLAM : on dispose à chaque instant T (i) d'une estimée ponctuelle de la carte θ_T ; et (ii) d'une approximation particulière de la loi de filtrage à l'instant T i.e. une approximation particulière de la pose du robot à T . Dans les simulations ci-dessous, on résume cette loi a posteriori par la valeur moyenne (pondérée) des particules pour proposer une estimée ponctuelle de la pose du robot à chaque instant T .

Les figures 1 à 4 présentent l'application de cet algorithme sur données simulées et sur données réelles¹ (Figure 4). Pour les données simulées, les bruits V_t et $W_{t,i}$ sont gaussiens resp. de variance $Q = \text{diag}(\sigma_v^2, \sigma_\phi^2)$ où $\sigma_v = 0.5\text{m.s}^{-1}$ et $\sigma_\psi = \frac{\pi}{60}\text{rad}$; et $R = \begin{pmatrix} \sigma_r^2 & \rho \\ \rho & \sigma_b^2 \end{pmatrix}$, où $\sigma_r = 0.5\text{m}$, $\sigma_b = \frac{\pi}{60}\text{rad}$ et $\rho = 0.01$. Pour mettre en oeuvre l'algorithme, le nombre d'amers et les matrices R, Q sont supposés connus. La loi de proposition du filtre particulière est une approximation gaussienne de la loi a priori $m(x, x')$. L'expérience a été réalisée avec $N = 100$ particules. La figure 1 représente la vraie trajectoire du robot (tirets), la vraie position des amers (étoiles) et l'estimation de

¹<http://www.cas.kth.se/SLAM>

la position courante du robot (points) avec le nuage de particules. La figure 2 compare notre méthode (trait plein) aux algorithmes itératifs *fastSLAM* [11] (pointillés) et *marginalSLAM* [9] (points) pour l'estimation de la position d'un amer : on trace l'erreur d'estimation moyenne (sur 100 répliquions de Monte Carlo) en fonction du nombre d'itérations. La figure 3 donne le résultat de l'expérience sur 100 répliquions de Monte Carlo. La vraie position des amers (points) est comparées aux estimations moyennes fournies à la fin de la trajectoire (ici, $T = 1492$) par le *marginalSLAM* (étoiles) et par notre algorithme (croix). De même, la vraie trajectoire du robot (trait plein) est comparée aux trajectoires moyennes sur les 100 répliquions pour le *marginalSLAM* (pointillés) et pour notre algorithme (tirets). Ces figures illustrent l'intérêt de notre algorithme, tant en qualité d'estimation qu'en vitesse de convergence.

Enfin, la figure 4 montre le résultat obtenu sur données réelles : on trace les vraies données (amers en points ; trajectoire en trait plein) et les estimées (amers en étoiles ; trajectoire en pointillés). Ces résultats sont obtenus en considérant un mélange de gaussiennes comme bruit d'observation, de poids $\omega_0 = 0.8$ et $\omega_1 = 0.2$ et de variance respectives R et $5R$, R étant donné plus haut. A la différence du cas "données simulées", le nombre total d'amers est inconnu : l'estimation est commencée avec une carte vide et, pour chaque nouvelle observation reçue, un processus d'association basé sur la vraisemblance est utilisé. Si aucun amer n'est suffisamment vraisemblable, un nouvel amer est créé dans notre carte.

Références

- [1] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38 (with discussion), 1977.
- [2] O. Cappé and E. Moulines, "On-line expectation-maximization algorithm for latent data models," *J. Roy. Statist. Soc. B*, vol. 71, no. 3, pp. 593–613, 2009.
- [3] G. Mongillo and Denève S., "Online learning with Hidden Markov Models," *Neural Computation*, vol. 20, pp. 1706–1716, 2008.
- [4] O. Cappé, "Online EM algorithm for hidden Markov models," 2011, To appear in *J. Comput. Graph. Statist.*
- [5] O. Cappé, "Online sequential Monte Carlo EM algorithm," in *IEEE Workshop Statist. Signal Process. (SSP)*, 2009.
- [6] P. Del Moral, A. Doucet, and S. S. Singh, "Forward smoothing using sequential monte carlo," arXiv :1012.5390v1, 2010.
- [7] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the EKF-Slam algorithm," in *IEEE International Conference on Intelligent Robots and Systems*, 2006, pp. 3562–3568.
- [8] S.J. Julier and J.K. Uhlmann, "A counter example to the theory of simultaneous localization and map building," in *IEEE international conference on robotics and automation*. Citeseer, 2001, vol. 4, pp. 4238–4243.
- [9] R. Martinez-Cantin, "Active map learning for robots : insights into statistical consistency," PhD Thesis, 2008.
- [10] O. Cappé, E. Moulines, and T. Rydén, *Inference in Hidden Markov Models*, Springer, 2005.
- [11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0 : An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proceedings of the Sixteenth IJCAI*, Mexico, 2003.

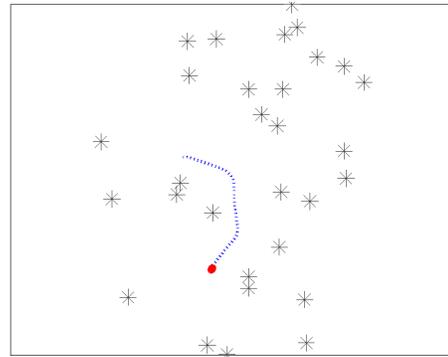


FIG. 1: Données simulées

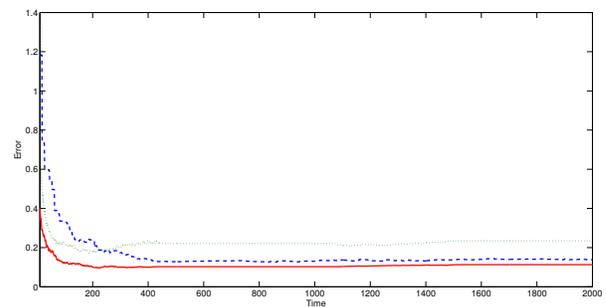


FIG. 2: Estimation de la position d'un amer

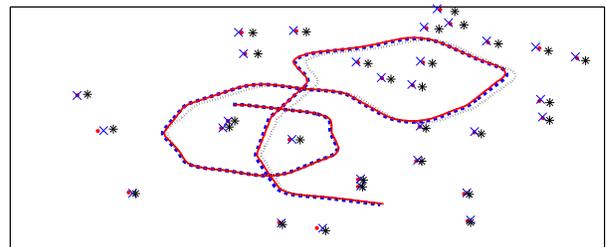


FIG. 3: Estimation de la pose

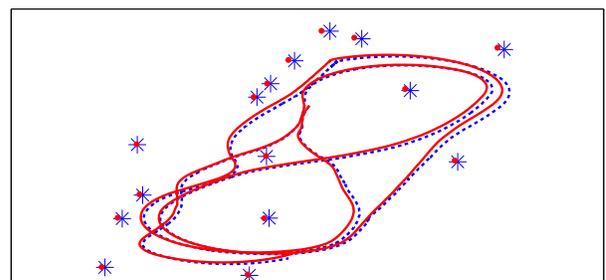


FIG. 4: Données réelles