Spatio-temporal mixture process estimation to detect population dynamical changes

Solange Pruilh, Anne-Sophie Jannot, Stéphanie Allassonnière

Spatio-temporal mixture process estimation to detect population dynamical changes

Solange Pruilh[1,2], Anne-Sophie Jannot[2,3], and Stéphanie Allassonnière[1,2]

[1]Center for applied mathematics - Ecole Polytechnique
[2]HeKA, Centre de Recherche des Cordeliers, University of Paris, INRIA, INSERM, Sorbonne University
[3]Department of Statistics, Medical Informatic and Public Health, Hôpital européen Georges-Pompidou, AP-HP

Abstract

Population monitoring is a challenge in many areas such as public health and ecology. We propose a method to model and monitor population distributions over space and time, in order to build an alert system for spatio-temporal data changes. Assuming that mixture models can correctly model populations, we propose a new version of the Expectation-Maximization (EM) algorithm to better estimate the number of clusters and their parameters at the same time. This algorithm is compared to existing methods on several simulated datasets. We then combine the algorithm with a temporal statistical model, allowing for the detection of dynamical changes in population distributions, and call the result a spatio-temporal mixture process (STMP). We test STMPs on synthetic data, and consider several different behaviors of the distributions, to adjust this process. Finally, we validate STMPs on a real data set of positive diagnosed patients to coronavirus disease 2019. We show that our pipeline correctly models evolving real data and detects epidemic changes.

Keywords: Gaussian Mixture Model, EM algorithms, spatio-temporal data

1 Introduction

The rapid growth of health information systems has led to the availability of a real-time spatio-temporal follow up of patients affected by a given disease with a high precision. A remaining challenge is to develop methods to use this data to improve public health strategies and to transform this observed data into actionable decision support systems.

Spatial models are based on the characterization of individuals by their geographical location (place of birth, place at the time of diagnosis, place of residence, etc.). All together, these people are building up a population. The temporal component is very important in disease monitoring, therefore requiring consideration of the population distribution as evolving over time. The association of spatial and temporal components for a disease yields a spatio-temporal distribution. One actionable decision-aid support system that could improve health management using such data is real-time highlighting of new or evolving clusters of patients, i.e. a specific sub-group of patients which will evolve differently, while the rest of the population remains stable. This would be particularly useful to rapidly identify a new contamination source for a transmissible disease, as soon as the first affected cases are present in health information systems.

1.1 Related works and motivation

1.1.1 Spatio-temporal statistical analyses in epidemiology

Spatio-temporal statistical analyses are already present in research in epidemiology and are mainly based on statistical tests, coupled, or not, with space-time kernel density estimation, as presented by Kirby et al. [1]. Scan statistic methods proposed in [2, 3] are reference methods for many studies. They propose to detect spatial and/or temporal clusters from aggregated data (discrete in space

and time) using sliding windows to compare cases and reference populations. Another scan statistic method is proposed in [4] in the absence of population-at-risk. In both cases, these methods require to fix several parameters on the considered sliding window (e.g. minimal area and minimal temporal size are two examples of the various parameters). Moreover, cases/controls studies are subject among other things to selection and expensive efforts to find a proper control group among other things and are not feasible in all situations [5]. In addition, these studies are prone to several biases [6]. As it is usually difficult to sample a control group from a reference population distribution, the ensuing comparison between cases and controls is exposed to false differences due to inadequate sampling of the control group [6]. Another important issue is that these methods do not provide a statistical modelling of the population over the whole space and time.

1.1.2 Estimation algorithms for mixture models

Different from looking at data in a sub window of the space, mixture models are another class of models to spatially model data in statistics. Mixture models come with strong advantages. First, they are flexible as one can set the probability distribution function (pdf) of each cluster depending on the type of observations (scalars, vectors, positive measures, etc). Second, the results are interpretable because subjects can be attributed to estimated classes a posteriori which enables one to distinguish homogeneous groups in the whole set. Third, they do not rely on a population reference distribution estimation, unlike scan statistics methods: they only rely on cases distribution. Last, these mixture models are parametric and well understood.

When data are multivariate real valued observations, the usual probability density for each cluster is the multivariate Gaussian distribution. This is particularly relevant when considering geographical data (mapped as lying on the real plane).

To perform the estimation of Gaussian mixture parameters, given the number of clusters (i.e. classes), the classic algorithm is the Expectation-Maximization (EM) algorithm introduced by Dempster et al. [7]. The estimate obtained with the EM algorithm is deterministic and highly dependent on the initialization step. Moreover, the construction of the sequence ensures that the critical points are maxima, but could be either global or local ones. To avoid sensibility to initial values and selection of a wrong local maximum, several strategies rely on repetitions of a random initialisation step or initialisation with K-means algorithm [8]. Recently, Lartigue et al. introduced an annealing E-step to better stride the support and become almost independent from the initialization [9]. However, this method requires to set the temperature profile which may be time-consuming.

Finding an optimal number of components K is not an objective directly included in the original EM algorithm. This objective is often based on a model selection step, which requires a collection of estimated models [10, 11, 12]. The well-known criteria for model selection are the Akaike Information Criterion (AIC) [10], and the Bayesian Informative Criterion (BIC) [11]. They have been proved to be adequate for selecting K, but they are asymptotic criteria, and can select under- (for AIC) or over-adjusted (for BIC) models. Non-asymptotic approaches have been proposed, such as the slope heuristic criterion, introduced by Birgé and Massart [12] and implemented by Baudry et al. [13]. It provides an optimal penalty of the log-likelihood, and thus an optimal model, but also requires a linear behavior of the log-likelihood. On the other hand, Baudry and Celeux [8] proposed to introduce a recursive initialisation which consists in using the K components solution to initialize the $K+1$ components mixture. However, their full process requires several GMM estimations, with a varying number of components K, leading to expensive computations.

Subsequently, the last decade has seen the emergence of methods aiming to simultaneously overcome the need for a collection of models, find the optimal number of classes, and avoid bad local maxima [14, 15, 16, 17, 18, 19]. Several methods rely on a minimum message length criterion [20, 21] which penalises the cost function [15, 19]. These methods force parameter space exploration to obtains several models. However they have the default to not stop before reaching a minimal number of clusters fixed in advance. This forced estimation of an internal collection of models is also present in [14], where Derman and Le Pennec combine the slope heuristic criterion for model selection [12] with a dynamical change of the number of components inside the EM algorithm.

Another dynamical algorithm is the step-wise split-and-merge EM algorithm [16, 17]. With split and merge criteria based on Kullback-Leibler divergence or correlation coefficient, these methods explore dynamically the parameters space by forcing clusters to merge together (or split apart). But they may rely on independent split and merge movements or several runs of the EM algorithms, implying computational issues. On the contrary, in the work of Yang et al. [18], the number of components is estimated in a single-run EM algorithm with a reasonably low computational time. This solution is named Robust EM algorithm. But this algorithm can reach incorrect local maxima as we will see below.

The temporal component to monitor the population distribution is absent of these different procedures using EM algorithms, and the epidemiological models presented previously also cannot meet the criteria for estimating, monitoring and modelling population dynamics over time. As a consequence, these drawbacks prevent us from directly using the presented algorithms to obtain correct approximations of population dynamic and to monitor them.

1.2 Contributions

In this paper, we propose a complete pipeline named spatio-temporal mixture process (STMP). This pipeline infers population distribution and highlights temporal population distribution differences as a first step towards a decision support and alert system for spatio-temporal analysis of the evolution of a population. STMP can be used to initiate a detailed analysis of the environment for example if the pathology may depend on environmental causes. The STMP can also allow to focus on effects of decisions in specific areas where changes are happening, as we have faced with the COVID-19 pandemic and successive lockdowns for example. Within the proposed STMP, we combine a mixture model with reliable estimation and temporal monitoring of this model. This pipeline will create a temporal process with two mixture models, one time-depending and one totally independent. The adequacy of population dynamic to either of these two models will determine if an alert should be raised or not.

As a module to our STMP, we will introduce an adaptation of the EM algorithm [7] to take into account a temporal dependency during a mixture model evolution. Finally, we will also propose an improvement of the Robust EM algorithm [18]. We will suggest changes to obtain a more automatic algorithm to avoid overlapping components, observed with the Robust EM algorithm on real data tests. This modified version of Robust EM algorithm is compared to the original one (and other selection model criteria) to show that on synthetic data, there is no loss of performances and on real data we outperform the state of the art algorithm.

To finish designing our STMP, we will perform experiments on synthetic data. And we will study the behaviour of our pipeline in different situations to produce a robust monitoring. Applying our process to a dataset of COVID-19 cases from the Paris area, we will demonstrate the adequacy of a mixture model evolving over time and the consistency of the alert response to population epidemic changes.

2 Notations and reminders on mixture models and estimation algorithms

We assume for our future application in Section 4 and 5 that the population is modeled as a Gaussian Mixture Model (GMM). In this section, we first recall the GMM definition. Then we detail the Robust EM algorithm, one of the algorithms used to fit GMM. These methods are the basic elements on which we build our STMP pipeline described in Section 3.1.

2.1 The Gaussian Mixture Model

In order to describe a Gaussian Mixture Model, we consider a set of observations denoted $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ with $\mathbf{x}_i \in \mathbb{R}^d$. Let $\mathcal{N}_d(\cdot|\mu_k, \Sigma_k)$ be the probability density function (pdf) of the Gaussian density of dimension d with mean μ_k and covariance matrix Σ_k. To write the GMM in its complete

form we introduce latent variables $(z_i)_{i=1,\dots,n}$, such as each z_i is following a categorical distribution of parameter π. This information is then encoded as a K-dimensional binary variable $\mathbf{z}_i$ for each $i \in \{1, \dots, n\}$ with $z_i^k = 1$ if data x_i belongs to cluster k, 0 otherwise.
Then the complete model writes :

$$\begin{cases} z_i & \sim \quad \mathcal{C}ategorical(\pi_1, \dots, \pi_k)\,, \\ x_i | z_i^k = 1 & \sim \quad \mathcal{N}_d(\mu_k, \Sigma_k)\,. \end{cases} \tag{1}$$

The whole issue with GMM is twofold. The first challenge is to estimate the number of components K in the model. Then, given this estimated K, the second issue is how to estimate the vector of parameters θ, containing the Gaussian distributions parameters and the mixture proportions π. All this has to be performed from the observed data only.

2.2 The Expectation-Maximization algorithm

The most popular algorithm to estimate a GMM is the Expectation-Maximization (EM) algorithm [7] as it has been introduced for that purpose. The general principle is to produce a sequence of parameters $(\hat{\theta}^p)_{p\in\mathbb{N}}$ which converges towards the set of critical points of the observed likelihood, which is for a GMM on a set of observations $\mathbf{x}$:

$$p_\theta(\mathbf{x}) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k \mathcal{N}_d(x_i | \mu_k\, \Sigma_k) \right] . \tag{2}$$

The EM algorithm alternates between an expectation step, and a maximisation step which updates the mixture parameters, until a convergence criterion is met. The detailed equations are given in Appendix A.1.

As the EM algorithm presents several drawbacks detailed in Section 1, and that we expect our framework to have a single run to estimate the data distribution at a given time step, we turn to the more "dynamical" algorithms where estimation and selection of the model are performed at the same time [15, 19, 17, 16, 18].

In the next part, we will detail a recent dynamical algorithm proposed by Yang et al. [18], which answers almost all issues and is the base of our proposition.

2.3 The Robust EM algorithm

As mentioned previously, the unknown number of clusters in GMM is a main issue. The authors of [18] go deeper into looking dynamically for the best number of components in the mixture. Their Robust EM adjusts the EM mixture objective function, by adding a criterion based on the entropy of the mixture proportions π_k. Non-informative proportions are given by a high entropy. Consequently, the penalty added to the likelihood is given by the negative entropy. Starting from the complete log-likelihood, the objective function to maximize in the M-step with this entropy-based penalty is therefore:

$$\begin{aligned} \mathcal{L}'(\theta, \mathbf{x}, \mathbf{z}) = & \sum_{i=1}^{n} \sum_{k=1}^{K} \tau_i^k log(\pi_k \mathcal{N}(x_i | \mu_k, \Sigma_k)) \\ & + \beta \sum_{i=1}^{n} \sum_{k=1}^{K} \pi_k log \pi_k, \text{with } \beta \geq 0\,. \end{aligned} \tag{3}$$

With this new criterion to maximise, the update equation of components proportions π inside the EM algorithm becomes:

$$\hat{\pi}_k^{(new)} = \hat{\pi}_{k,MLE} + \beta \hat{\pi}_k^{(old)} \left(\ln \hat{\pi}_k^{(old)} - \sum_{s=1}^{K} \hat{\pi}_s^{(old)} \ln \hat{\pi}_s^{(old)} \right) \tag{4}$$

with $\hat{\pi}_{k,MLE}$ obtained by maximisation of the original objective function (without penalisation) (see Section A.1 Eq.(9)), and $\hat{\pi}_k^{(old)}$ being the component weight estimate of previous iteration. The equations to estimate the means $\hat{\mu}_k$ and the covariance matrices $\hat{\Sigma}_k$ in Robust EM remain unchanged. These parameters are estimated at each maximisation step by Eq.(10) and Eq.(11) with the new component weights from Eq.(4).

As we can see, a new hyperparameter β comes as a penalty weight in Eq.(3). It helps to control the competition between clusters. Acting on the evolution of proportions with β enables one to check at each iteration that all the components proportions are above a given threshold, and therefore to delete those of proportion $\pi_k < \frac{1}{n}$. This is the annihilation part in their process. A specific dynamic is imposed on β. This parameter is set to zero when the cluster number K is stable, i.e not decreasing for a time period p_{min}. This is important to avoid oscillating parameters, and so to reach a maximum. A limitation is that they fixed this time limit to $p_{min} = 60$ iterations, without any attempt to adapt it to different use cases. This algorithm is however robust to initialization as, to start with, each data point is the center of its own component, which yields the initial number of class K^0 to be n, the sample size.

Although efficient, entropy-based penalisation [18] does not prevent from having several components with similar parameters, meaning that two cluster may be superimposed. In the Robust EM algorithm [18], competition and instability of component proportions do not avoid ending up with a local maximum of this type. The coefficient β is usually not high enough to trigger removal of one of the superimposed clusters. As the competition is not guaranteed at each iteration, we suggest improvements of the Robust EM algorithm in the next section. We also present a temporal process which, combined with estimation algorithms, will provide efficient detection of population dynamical changes.

3 Method: Spatio-temporal mixture model with efficient estimation algorithms for dynamical change detection

In this section, we describe our general pipeline for temporal evolution modelling of a population including a distribution change detection, named STMP. Then, we introduce modifications on the Robust EM algorithm to escape local maxima characterised by "overlapping clusters". Finally, we detail another adaptation of the EM algorithm in order to constrain the estimation of GMM parameters. This enables to propose a close variant of a given distribution which, since estimated, highly depends on samples. The STMP pipeline and the estimation algorithms are generic enough to apply on different mixture models by using different estimation algorithms.

3.1 A spatio-temporal mixture process (STMP) with dynamical change detection

We consider that the time period is discretized and the time steps are given by $t = 1, \ldots, T$. At each time step, denote the data vector $\mathbf{X}^{(t)} = (\mathbf{X}_1^{(t)}, \ldots, \mathbf{X}_{n_t}^{(t)})$ with $\mathbf{X}_i^{(t)} \in \mathbb{R}^d$. We assume that this data is sampled from a statistical time dependent model. We model the data at each time step t by a mixture of probability distributions, parametrized by a vector $\theta^{(t)}$, characterizing the current model $M^{(t)}$.

At each time t, we observe a new vector $\mathbf{X}^{(t)}$, independent of the previous one $\mathbf{X}^{(t-1)}$. Given this new sample, we want to evaluate if the previous model $M^{(t-1)}$, defined as a mixture model estimated on $\mathbf{X}^{(t-1)}$, is likely to fit the new set $\mathbf{X}^{(t)}$. We make the assumption that the distribution of the underlying global population does not change over time. This is in line with the difficulties related to the use of reference populations presented in Section 1 and coherent with our targeted applications. We design our model to monitor population evolution over time in particular for either short time period of time like the COVID-19 analysis where the population was constrained to strict locked-down or hard traveling restrictions, or longer period of time where the study focuses on longer time steps as well, with aggregated data. Thus the model does not require any datasets other than

the vectors $\mathbf{X}^{(t)}$ for each time step t.

However, as $M^{(t-1)}$ depends on the data set at time $t-1$, it suffers from estimation variability, which means that the true model is likely close to but not equal to $M^{(t-1)}$. To deal with this uncertainty, we estimate a constrained model (or candidate model) M' to fit $\mathbf{X}^{(t)}$ where M' is an adjustment of $M^{(t-1)}$, given by θ' close to $\theta^{(t-1)}$. Through this adaptation of $M^{(t-1)}$, we indirectly keep track of the estimated model at previous time. However if at time t the data set $\mathbf{X}^{(t)}$ is sampled from a very different distribution, M' should not be able to fit $\mathbf{X}^{(t)}$. In this situation, we would like our process to detect this shift in population dynamic, and propose an alternative model more representative of the new data.

In order to do this, we propose to also estimate an alternative model, M^a only from the dataset $\mathbf{X}^{(t)}$. We do not make any assumption on a previous time step dependence to estimate this model leading to a parameter vector θ^a only driven by $\mathbf{X}^{(t)}$.

With these two estimated models in hands, we are now able to track changes of the population distribution, and determine whether there is a modification in the population geographical spreading. Our proposed warning system is defined as follows. If at time t, the model M', close to $M^{(t-1)}$, is not adapted to describe $\mathbf{X}^{(t)}$, we keep the independent model M^a as the new description of the current population and raise an alert. The aim is now to define the decision rule to select either model and to raise the alert or not as a result.

A simple way to quantify goodness of fit of a statistical model to the data is its likelihood. The likelihoods of estimated mixture models M' and M^a, given by $p_{\theta'}(\mathbf{X}^{(t)})$ and $p_{\theta^a}(\mathbf{X}^{(t)})$ respectively, are used to define a decision rule in our process, called the likelihood ratio or Bayes factor.

As the alternative model is unconstrained, $p_{\theta^a}(\mathbf{X}^{(t)})$ is the maximum value of the likelihood of the data without assumption. On the other hand, $p_{\theta'}(\mathbf{X}^{(t)})$ is the maximum value of the likelihood when the parameters θ' are restricted to stay in a neighbourhood of $\theta^{(t-1)}$. In the case where the constrained model M', fits well the new data set, the alternative model is likely to be similar and to have a similar likelihood. Therefore, the likelihood ratio will be close to one. On the other hand, if the new data set is sampled from a very different distribution from $M^{(t-1)}$, then the constrained model will have a likelihood that is lower than the alternative model which by design will be able to better fit the new point cloud. Therefore, there should be a notification when this ratio is far above one.
Finally, we define the ratio as follows:

$$r_t(M', M^a) = \frac{p_{\theta^a}(\mathbf{X}^{(t)})}{p_{\theta'}(\mathbf{X}^{(t)})}\,. \tag{5}$$

In order to accept or reject the alternative model at time t, we define a threshold τ such that if $r_t(M', M^a) \geq \tau$, the alternative model is selected and an alert is raised. The detailed behaviour of this likelihood ratio depending on the population evolution will be studied in Subsection 4.3. In particular, this empirical study allows us to set the threshold τ and highlight its properties in particular its low dependence w.r.t the sample size.

With all these elements in hand, our space-time complete pipeline, named Space-Time Mixture Process (STMP), executes at each time t the following steps:

1. Estimate models M' and M^a based on respectively $(M^{(t-1)}, \mathbf{X}^{(t)})$ and $(\mathbf{X}^{(t)})$,
2. Compute likelihood ratio $r_t(M', M^a)$ as in Eq.(5),
3. If $r_t(M', M^a) \geq \tau$, raise an alert and set $M^{(t)} = M^a$. Else set $M^{(t)} = M'$.

Note that this pipeline is very versatile with respect to the chosen distributions in the mixture model as well as the estimation algorithms used in first step. Depending on the dataset, the model is able to handle any type of pdfs.

We now describe the two algorithms that we use to perform the candidate and alternative model estimations.

3.2 The Modified Robust EM algorithm: tackling superimposed clusters

In Section 2, we have highlighted two weaknesses of the Robust EM algorithm by [18]. First, the minimal number of iterations (named p_{min}) before setting $\beta = 0$ is too small, which means that the algorithm is untimely stopped in its exploration. Then, the algorithm is stuck in local maxima as soon as the convergence condition ($\|\mu^{(p)} - \mu^{(p-1)}\| < \tau$ where $\tau > 0$ is a threshold) is satisfied, which stops the algorithm too early, revealing aberrant clusters. These aberrant clusters are here superimposed clusters, which means that at least two clusters are sharing very similar (or exactly equal) parameters values. This corresponds to local maxima which can be analysed only by post-processing the results, and it is particularly observable on real and scattered data.

To avoid this local maximum issue inside the estimation algorithm (avoid post-processing analysis), we propose slight modifications of the Robust EM algorithm, by incorporating an online verification step of superimposed clusters. We consider that two clusters i and j are superimposed if

$$\|\mu_i - \mu_j\|_2 + \|\Sigma_i - \Sigma_j\|_F < \epsilon \tag{6}$$

for some small $\epsilon > 0$. Note that requiring equality in Eq. (6) is numerically too strong and would rarely happen. We check Condition (6) when the algorithm has reached the convergence condition (Algorithm 2, line 1). As long as there are overlapping clusters we force the estimation to continue, as we will see now.

Inside Algorithm 2, the "stop-competition" part is the moment in the algorithm where $\beta = 0$ if the component number is stable for at least 100 iterations and if the actual iteration number p is greater than p_{min} (Algorithm 2, line 2). At that point in the algorithm, if we set $\beta = 0$ too early, it slows down the competition between clusters, and may prevent components from disappearing. If there are no overlapped clusters and stability conditions are fulfilled then we set $\beta = 0$. Otherwise, we proceed as follows: we first increase p_{min} by increment of 50 iterations (Algorithm 2, line 3). By increasing p_{min}, the algorithm has more iterations to try to annihilate some components. Since increasing p_{min} indefinitely can lead to a "stable" configuration where β adopts a cyclical behaviour and loops on it, we then check the proximity condition (6) again. If Eq. (6) is still true for some clusters, we merge these clusters. The weight of the fused clusters is the sum of the weights of the overlapping ones. The means and covariance matrices being almost equal, this fusion of components does not change much the likelihood. This makes the algorithm jump to another configuration with almost the same likelihood and enables it to explore this new region of interest. Other steps of the algorithm stay identical to the original Robust EM, as presented in Subsection 2.3. The full modified Robust EM algorithm is summarized in Algorithm 2.

3.3 The Constrained EM algorithm: former parameter based estimation

We name Constrained EM (C-EM) a slight variation of original EM algorithm [7] where the parameters are restricted to a neighbourhood of a given vector of parameters denoted θ^0. In particular, we introduce constraints on the estimated components proportions $(\pi_k)_{1 \leq k \leq K}$. Moreover, when the cluster means are involved, restrictions are also put on these means. The initialization of our C-EM algorithm is given by the parameter vector θ^0 as well. The idea behind C-EM algorithm is to obtain estimated parameters highly driven by the initial parameters vector θ^0 but updated on data $\mathbf{X}$. Because the parameters of our dynamical modeling are estimated empirically, the estimation suffers from the uncertainty given by the sampling. This means that the estimated parameters at time $t-1$ may not be the perfect description of the data set and a new independent sample from the same ground truth distribution will lead to a slightly different estimated parameter vector and a slightly different likelihood. Therefore, we consider that a newly independent estimated mixture and the given estimated one may both come from the same ground truth. For this reason, the C-EM enables us to give a chance to the previously estimated model to explain the data distribution. Otherwise, forcing the comparison of $M^{(t-1)}$ with M^a will always be in favor of M^a. With this parameters-dependency, the newly estimated parameters could be incorporated in our temporal process as a time-dependent estimate.

From now on, we propose the details of this algorithm for distributions where the cluster means and covariances are to be estimated. This will be the case in our disease progression use case where the model is a mixture of Gaussian distributions. We now detail constraints we impose on parameter estimations inside an EM-like algorithm to estimate GMM. We name $\hat{\pi}^c$, $\hat{\mu}^c$ and $\hat{\mathbf{\Sigma}}^c$ the constrained proportions, means and covariance matrices obtained through the C-EM algorithm. As in the original EM algorithm, $\hat{\pi}^p$ and $\hat{\mu}^p$ vectors are estimated at iteration p of C-EM following equations (9) and (10). We then add a third step in the estimation algorithm to obtain $\hat{\pi}^c$ and $\hat{\mu}^c$.

The constrains in the C-EM algorithm always imply θ^0, the initial parameter vector at $p = 0$, as we want to restrict the parameters estimation. The initial parameter vector contains $(\pi_k^0)_k$, $(\mu_k^0)_k$ and $(\Sigma_k^0)_k$ the covariance matrices providing information about the anisotropy we allow for the uncertainty on the means parameters to adapt locally. Components proportions are probability weights and live in $[0, 1]$, so we simply constrain component proportion of cluster k, $\hat{\pi}_k^p$ (at iteration p), to vary inside $[\pi_k^0 \pm 0.1]$. This means each proportion varies by at most 10We also avoid proportions to become null to avoid the artificial death of a cluster in the mixture. Constrained mean $\hat{\mu}_k^c$ of the component k at iteration p with the C-EM algorithm is a projection of estimated $\hat{\mu}_k^p$ on a rectangular space centered on μ_k^0 and of length and width given by ellipse axis of the covariance matrix Σ_k^0 (square roots of the eigenvalues of Σ_k^0).

These constraints are written here for each iteration p:

$$\begin{cases} \hat{\pi}_k^c & = min(max(\pi_k^0 - 0.1, \hat{\pi}_k^p), \pi_k^0 + 0.1)\,, \\ \hat{\mu}_k^c & = \mathcal{P}_{rect(\mu_k^0, \Sigma_k^0)}(\hat{\mu}_k^p)\,. \end{cases} \tag{7}$$

Note that the algorithm can converge to final parameters where one covariance matrix is degenerated, reflecting the aim of the algorithm to delete one component of the mixture model. In the original EM algorithm, implementations usually include a regularisation on the covariance matrices, in order to avoid singular ones. As we want to determine when the estimated candidate model does not correspond to the data, we remove this regularisation from the C-EM algorithm. Therefore, we raise an alert when one or more covariance matrices become singular. We add this condition as an alert in STMP detailed in Subsection 3.1, before the calculation of the ratio r_t (Eq.(5)).

In addition to this, as the covariance matrices are not constrained in the C-EM algorithm, we introduce a condition to check these parameters a posteriori. From the C-EM algorithm, covariance matrices are freely estimated, but they can evolve far away from initial covariances matrices Σ_k^0, thus missing the time link. We introduce an already existing similarity measure between final estimated $\hat{\Sigma}_k^c$ in C-EM and Σ_k^0 the initial covariance matrices. We use the cosine similarity, also introduced as the correlation matrix distance by [22] on correlation matrices. We adopt their formulation and apply it on covariance matrices instead of correlation matrices. Bounded between 0 and 1, this coefficient measures orthogonality between two matrices and is useful to evaluate whether the spatial structure of the clusters have significantly changed. Low values reflects high similarity while high values reflects orthogonality, and so on dissimilarities. As $\hat{\Sigma}_k^c$ should be similar to Σ_k^0, we only tolerate a value of 0.1 or less, in order to introduce flexibility and sampling error tolerance inside STMP. For higher values, showing dissimilarities between $\hat{\Sigma}_k^c$ and Σ_k^0, we also raise an alert in STMP detailed in Subsection 3.1.

In STMP, θ^0 will be the estimated parameter vector from the previous time step $t - 1$ of the pipeline, which corresponds to $\theta^{(t-1)}$. We obtain at time t an estimated parameter depending on estimated parameter at time $t-1$, but allowing some adaptation of the model to the newly observed data $\mathbf{X}^{(t)}$. Finally, we should not forget that the C-EM is constrained by initial parameters θ^0, including a fixed number of clusters K^0. It is not possible in C-EM to merge clusters based on their properties, as this would violate the imposed constraints. If the model estimated by C-EM is not correctly fitting data $X^{(t)}$, this will be detected inside STMP.

3.4 Application of the STMP on Gaussian Mixtures Models

To conclude this section, our new process is fully described in Algorithm 1, combining the temporal process described in Subsection 3.1 with the C-EM to estimate M' (Subsection 3.3), and the modified Robust EM to estimate M^a (Subsection 3.2) on GMMs.

Algorithm 1: The Spatio-Temporal Mixture Process (STMP)

input : For $t = 0, \dots, T$: data $\mathbf{X}^{(t)}$

$\theta^0 \leftarrow$ `ModifiedRobustEM(`$\mathbf{X}^0$`)`
$\theta^{(t)} \leftarrow \theta^0$
for $t = 1, \dots, T$ **do**
  $\theta^{(t-1)} \leftarrow \theta^{(t)}$
  $\theta' \leftarrow$ `C-EM(`$\mathbf{X}^{(t)}$, $\theta^{(t-1)}$,*maxiterations=5*`)`
  $\theta^a \leftarrow$ `ModifiedRobustEM(`$\mathbf{X}^{(t)}$`)`
  $r_t \leftarrow \frac{p_{\theta^a}(\mathbf{X}^{(t)})}{p_{\theta'}(\mathbf{X}^{(t)})}$
  if *(*$\exists$ *singular* $\hat{\Sigma}'_k \subset \theta'$*) or (*$\exists$ *cos_similarity*$(\hat{\Sigma}'_k, \hat{\Sigma}_k^{(t-1)}) > 0.1$ *)* **then**
    alert $\leftarrow$ True
    $\theta^{(t)} \leftarrow \theta^a$
  else if $r_t \geq \tau$ **then**
    alert $\leftarrow$ True
    $\theta^{(t)} \leftarrow \theta^a$
  else
    $\theta^{(t)} \leftarrow \theta'$
  end
end

As in the following applications we will only consider geographical data, in $\mathbb{R}^2$, we use Gaussian Mixture Models to represent these data. The GMM parameters are estimated with the presented algorithms, and the likelihoods are computed with Eq. (2). Recall that the pseudo-code 1 shows the STMP with all our propositions, which could be used with different mixture models. The adaptation of the estimation algorithms may also be used to fit with other distributions.

4 Experiments on synthetic data

This section is dedicated to the experimental validations therefore focused on synthetic data. First, we present comparisons of our Modified Robust EM with other EM-based algorithms and selection criteria. The comparisons are conducted on two mixture distributions from existing benchmarks. Second, we present all the experiments that are tested on our complete pipeline. We study the estimated likelihood ratio for different behaviors of the population distribution (characterized by the experiments) and the resulting performances of the pipeline. By analyzing these performances we can fix a threshold conditioning the raise of an alert in all situations. Then, we focus on the validation of STMP, given by Algorithm 1. Finally, we also present experiments on the number of points n in the data sample, and how it affects each step of STMP.

4.1 Comparisons of the Modified Robust EM with other EM-based algorithms and selection criteria

We compare here our Modified Robust EM with several EM-based methods mentioned in Subsection 1.1.2. First, we run the original EM algorithm, which is based on the a priori knowledge of the number of clusters. As in practice we do not know the true number of components, we estimate several models and select the best one based on either Bayesian Information Criterion (BIC) [11] or Integrated Completed Likelihood (ICL) [23], two of the most commonly used criteria in model selection. Several mixture estimates are therefore obtained by running an EM algorithm for a range of values of K from K_{min} to K_{max}. The initialization issue is treated by starting from 10 random small K-means runs and then keeping the solution with the highest likelihood as the initialization of

the EM algorithm. Second, we also compared our method with the original Robust EM algorithm (REM) [18] and Figueiredo and Jain's method [15] (called FJ method from here). The FJ method requires to fix an initial number of clusters $K_{initial}$. Originally $K_{initial}$ was "far from the true number of components" but not too high (around $K_{initial} = 30$ in several experiments of [15]), but in order to approach the behavior of the Robust EM and Modified Robust EM we use $K_{initial} = n$. The methods are computed on 100 different data sets generated for each of the defined mixture distributions. The methods are then compared based on their capacity to estimate the correct number of components and when this number is correct, to estimate the parameters of the mixture models. They are also compared in terms of the computational cost given by the number of iterations.

First, we compare the different methods on their ability to estimate the correct number of components. From a first mixture given by Fig.1a (with $n = 400$ points), REM was 95 percent successful in identifying the four components, close to the 99 percent of our method, against 51 percent for the FJ algorithm and 63 and 61 percent for EM-BIC and EM-ICL respectively. From a second mixture given by Fig.1b (with $n = 400$ points), all methods had more difficulty in identifying the four clusters. EM-BIC and EM-ICL were the most performant with 52 and 54 percent of successful estimation of the number of components, against 46 percent for our method MREM, and then 37 percent for the REM and 36 for the FJ method.
Then, we compare the estimated parameter precision over runs with successful component estimation. For each of the two defined mixtures, we computed the relative distance between the true and the estimated parameters. From these mean relative errors (Table 1 and Table 2), all the values are of the same range. It appears that FJ method, EM-BIC and EM-ICL have slightly lower errors than REM and MREM on the first mixture (Fig.1a), but slightly higher errors than REM and MREM on the second mixture. This shows the importance of capturing the correct number of clusters, which is the goal of our algorithm. However, this implies for model selection criteria to have an average guess of the data heterogeneity and to run the estimation algorithm for each of the possible number of components and for several initializations each time.

Finally, we compare the mean number of iterations for executions with each mixture distribution.

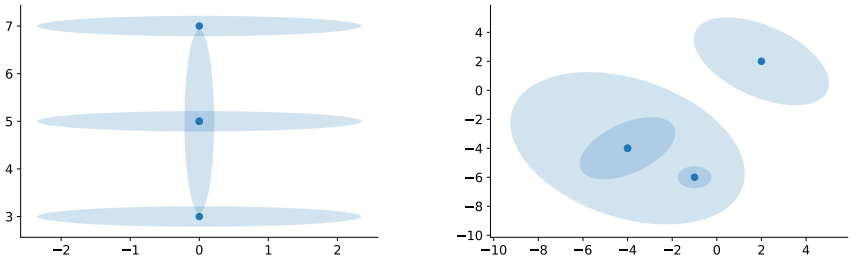
	REM	FJ	EM-BIC	EM-ICL	MREM
$\hat{\pi}_0$	0.0851 (0.0547)	**0.0670** (0.0460)	0.0683 (0.0465)	0.0683 (0.0465)	0.0870 (0.0615)
$\hat{\pi}_1$	0.0988 (0.0740)	0.0678 (0.0513)	**0.0667** (0.0514)	**0.0667** (0.0514)	0.0990 (0.0737)
$\hat{\pi}_2$	0.0937 (0.0719)	**0.0741** (0.0660)	0.0742 (0.0654)	0.0742 (0.0654)	0.0935 (0.0724)
$\hat{\pi}_3$	0.0888 (0.0749)	**0.0680** (0.0689)	0.0681 (0.0682)	0.0681 (0.0682)	0.0894 (0.0731)
$\hat{\mu}_0$	0.0305 (0.0230)	0.0250 (0.0206)	**0.0248** (0.0205)	**0.0248** (0.0205)	0.0308 (0.0231)
$\hat{\mu}_1$	**0.0167** (0.0111)	0.0186 (0.0120)	0.0185 (0.0119)	0.0185 (0.0119)	0.0187 (0.0129)
$\hat{\mu}_2$	0.0116 (0.0082)	0.0121 (0.0078)	0.0122 (0.0078)	0.0122 (0.0078)	**0.0115** (0.0082)
$\hat{\mu}_3$	0.0187 (0.0122)	0.0207 (0.0123)	0.0210 (0.0123)	0.0210 (0.0123)	**0.0169** (0.0104)
$\hat{\Sigma}_0$	0.1052 (0.0737)	**0.1030** (0.0799)	**0.1030** (0.0792)	**0.1030** (0.0792)	0.1062 (0.0734)
$\hat{\Sigma}_1$	0.7164 (0.5245)	0.4811 (0.5133)	**0.4730** (0.5117)	**0.4730** (0.5117	0.6087 (0.5353)
$\hat{\Sigma}_2$	0.1121 (0.0777)	0.1075 (0.0725)	**0.1071** (0.0719)	**0.1071** (0.0719)	0.1128 (0.0786)
$\hat{\Sigma}_3$	1.0568 (0.8288)	0.7403 (0.8096)	**0.7275** (0.8070)	**0.7275** (0.8070)	0.9110 (0.8541)

Table 1: Mean (standard deviation) relative errors for the estimates parameters of GMM within dataset from Fig.1a. The absolute-value norm is used for proportions, the euclidean norm is used for means, and the Frobenius norm for covariances.

The mean number of iterations on first and second mixture is respectively of 83 and 137 iterations for REM, against 95 and 185 for our method, 170 and 222 for BIC/ICL which used $K_{min} = 2$ and $K_{max} = 6$, and 730 and 711 for FJ method. The number of iterations is slightly higher with our method than the REM one because we put a "soft" condition on convergence to stop the algorithm. The number of iterations is very high for the FJ method because of the initial number of components, which is high here. But it was originally fixed to a lower number by the authors of [15], and needed to be fixed arbitrarily.

	REM	FJ	EM-BIC	EM-ICL	MREM
$\hat{\pi}_0$	**0.1427** (0.1393)	0.1665 (0.1795)	0.1645 (0.1775)	0.1645 (0.1775)	0.1430 (0.1605)
$\hat{\pi}_1$	**0.1281** (0.1399)	0.1650 (0.1694)	0.1628 (0.1676)	0.1628 (0.1676)	0.1472 (0.1452)
$\hat{\pi}_2$	**0.0405** (0.0364)	0.0709 (0.0501)	0.0704 (0.0495)	0.0704 (0.0495)	0.0600 (0.0479)
$\hat{\pi}_3$	0.1477 (0.0991)	0.1461 (0.1053)	0.1468 (0.1040)	0.1468 (0.1040)	**0.1268** (0.030)
$\hat{\mu}_0$	0.0761 (0.1174)	0.0669 (0.1720)	0.0661 (0.1697)	0.0661 (0.1697	**0.0391** (0.0236)
$\hat{\mu}_1$	**0.0368** (0.0265)	0.0455 (0.0356)	0.0458 (0.0351)	0.0458 (0.0351)	0.0650 (0.1150)
$\hat{\mu}_2$	0.0589 (0.0325)	0.0603 (0.0224)	0.0626 (0.0261)	0.0626 (0.0261)	**0.0561** (0.0300)
$\hat{\mu}_3$	**0.0117** (0.0071)	0.0141 (0.0063)	0.0140 (0.0063)	0.0140 (0.0063)	0.0121 (0.0067)
$\hat{\Sigma}_0$	3.7634 (2.1204)	1.9392 (2.4028)	1.8895 (2.3888)	1.8895 (2.3888)	**1.4402** (2.019)
$\hat{\Sigma}_1$	0.7021 (0.3135)	0.4211 (0.3314)	0.4139 (0.3297)	0.4139 (0.3297)	**0.3611** (0.3156)
$\hat{\Sigma}_2$	0.1213 (0.0611)	0.1253 (0.0620)	0.1277 (0.0627)	0.1277 (0.0627)	**0.1102** (0.0469)
$\hat{\Sigma}_3$	0.3597 (0.1584)	0.3640 (0.1616)	0.3582 (0.1631)	0.3582 (0.1631)	**0.3460** (0.1855)

Table 2: Mean (standard deviation) relative errors for the estimates parameters of GMM within dataset from Fig.1b. The absolute-value norm is used for proportions, the euclidean norm is used for means, and the Frobenius norm for covariances.



(a) A Gaussian mixture with 4 crossed components, defined in [18].

(b) A Gaussian mixture with 4 overlapping components, defined initially in [15].

Figure 1: Two Gaussian mixtures defined in [15] and [18].

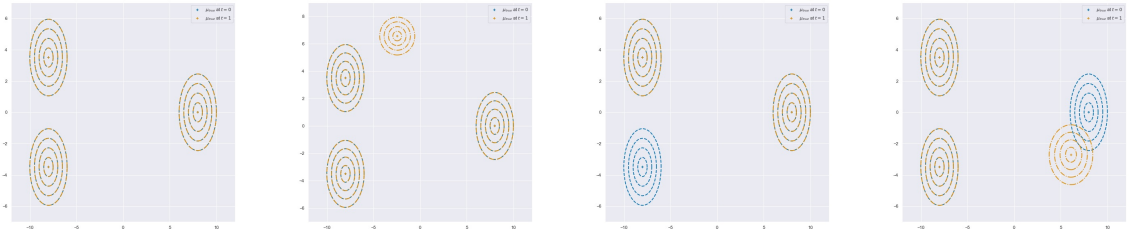
Note that we have provided a narrow range of values including the correct one for model selection criteria with BIC and ICL. The EM algorithm failed with higher number of components as the algorithm tended to remove one cluster by cancelling its proportion and degenerating the covariance matrix. Our Modified Robust EM shows no loss of performances compared to the Robust EM on synthetic data, and solve Robust EM problems on real data as we will see later.

4.2 Description of the experimental setups to calibrate STMP

All the experiments are done on a two time steps configuration (only $t = 0$ and $t = 1$). We consider the following situation where we have a Gaussian mixture distribution with three clusters at initial time ($t = 0$). One cluster is isolated on the far right hand side, and the two others are on the left hand side. This is the basic structure that all initial distributions (at $t = 0$) will follow. Different positions of left hand side clusters are represented in Figure 3, for Setup F. (Far.), Setup M. (Moderate.) and Setup C. (Close.).

From this initial Gaussian mixture, various changes are done at time $t = 1$ considering:

- (Case I.) : no evolution at $t = 1$, clusters are properly distinct (corresponds to Setup F. at $t = 0$ and $t = 1$).
- (Case II.) : the emergence of one new cluster leading to a distribution with four clusters at



(a) Case I. (b) Case II. (c) Case III. (d) Case IV.

Figure 2: Description of Gaussian mixture distributions for Cases I. to IV. (from Table 9). Blue centers and covariance ellipses correspond to Gaussian Mixture parameters at $t = 0$, orange ones to Gaussian Mixture parameters at $t = 1$. Note that when both elements are superimposed, the centers only appear orange and the ellipses have mixed colors dotted lines.

time $t = 1$.

- (Case III.) : the disappearance of one cluster among the existing three initially present.
- (Case IV.) : the movement of one initial cluster, which corresponds to moving centers and changing proportions and covariances.
- (Case V. and Case VI.) : no evolution at $t = 1$, as Case I., but here the two left hand side clusters are slightly interfering (Setup M.) for Case V., and finally these two clusters are very closed and barely identifiable without enough samples (Setup C.) for Case VI. .
- (Case VII. to Case IX.) : from initial Setup F. or M. at $t = 0$, there is a movement of the two left hand side clusters to Setup M. or C. at $t = 1$.

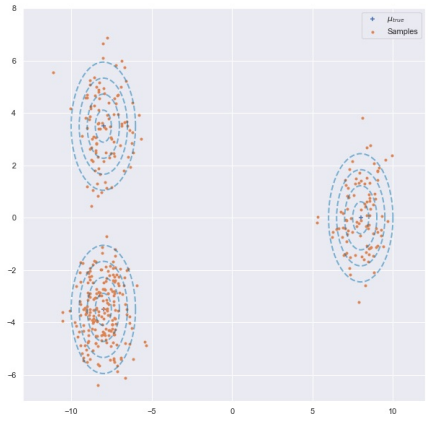
We denote $K^{(0)}_{true}$ the number of components in the mixture distribution at $t = 0$, $K^{(1)}_{true}$ in the mixture distribution at $t = 1$. A case is finally characterised by its mixture parameters at $t = 0$ and at $t = 1$ and we represent all cases in Table 9. In addition, Figure 2 and Figure 3 give a simple representation of Cases I. to IV. and of Setup F., M. and C. involved in Cases V. to IX.

We obtain Gaussian mixture distributions with parameters $\theta^{(0)}$ and $\theta^{(1)}$ from described cases. For each case I. to IX. , given the two distributions, we can sample $n_0 = n_1 = n$ points, which form our data sets $\mathbf{X}^{(0)}$ and $\mathbf{X}^{(1)}$. The sampling step, for any of the cases presented above, is executed S times and followed by execution of our STMP on each set of sampled data. It produces S different resulting processes, for each experiment (see Table 9). This enables us to analyze the behavior of STMP and likelihood ratio across runs and evolution cases.

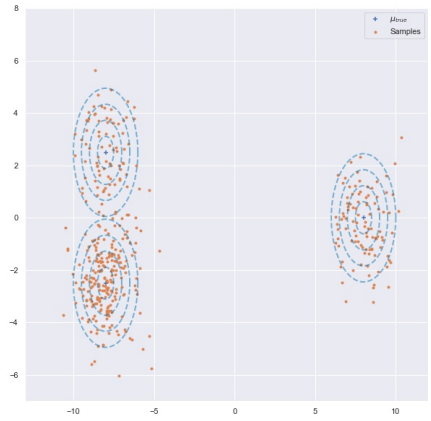
4.3 Estimation of the alert threshold in STMP

As motivated in Subsection 3.1, the likelihood ratio is a good indicator of how well the alternative model M^a at time t is fitting data $\mathbf{X}^{(t)}$ against the model M'. In case of no evolution of the distribution from $t = 0$ to $t = 1$, both M^a and M' should fit the data correctly, leading to a likelihood ratio around one. Of course, as said previously, due to the sampling of the distribution, it cannot be equal to one exactly. Thus the goal of the following study is to introduce an empirical threshold of adequacy, over which the alternative model M^a is definitely considered as the best model explaining current data and an alert is raised. With all the experiments above, we study the behavior of our STMP according to the alert threshold τ involved in Algorithm 1. It is important to fix this threshold in order to raise meaningful alerts and reach a correct performance.

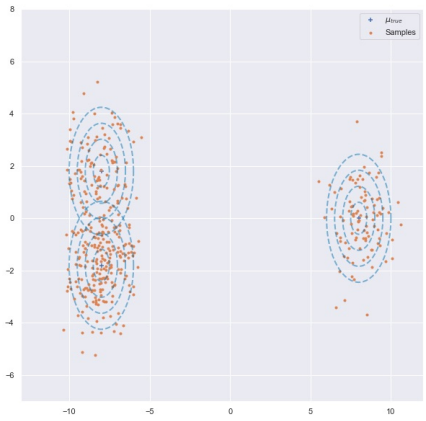
As said previously, we run S sampling steps for each case distributions, here fixed to $S = 100$ runs. We obtain S pairs of datasets $(X^{(0)}, X^{(1)})$. For each pair we compute the theoretical likelihood



(a) Setup F. (Far)



(b) Setup M. (Middle)



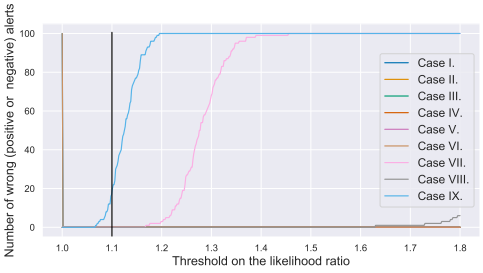
(c) Setup C. (Close)

Figure 3: Gaussian mixture distributions for Setups F., M. and C. involved in Cases presented in Table 9 with an example of sampled data sets. Blue crosses correspond to μ_k and ellipses to covariance matrices Σ_k. Orange points are samples.

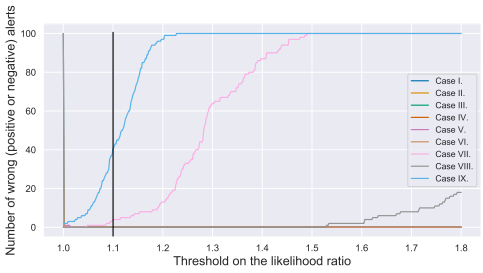
ratio

$$r_1^*(M^{(0)}, M^{(1)}) = \frac{p_{\theta^1}(\mathbf{X}^{(1)})}{p_{\theta^0}(\mathbf{X}^{(1)})},$$

implying the true parameters of models $M^{(0)}$ and $M^{(1)}$. This provides a "theoretical" value of r, only depending on the observation sets. We then account for the number of wrong alerts, depending on the value of the likelihood ratio threshold τ. Figure 4 represents this behavior, with one curve by case explained in Subsection 4.2. As the dataset $X^{(1)}$ is sampled from the truth model $M^{(1)}$, the theoretical likelihood ratio should be almost one modulo the variability of the data if $M^{(0)} = M^{(1)}$. In contrary, this theoretical ratio should quickly diverge from one if $X^{(1)}$ is not corresponding to the model $M^{(0)}$. This explains that we obtain 100 % of correct alerts on the majority of the case experiments (Fig. 4), as the computed theoretical likelihood ratios are really higher than tested values of the threshold. The cases which are critical for the choice of the threshold are Case VII. and Case IX (Fig. 4). They imply slight differences of the distributions between $t = 0$ and $t = 1$, so the



(a) Experiments with datasets of size $n = 400$.



(b) Experiments with datasets of size $n = 100$.

Figure 4: For each Case is presented the number of false alerts (positive or negative) on theoretical likelihood ratios, over $S = 100$ runs according to the considered threshold τ. The filled black vertical line is the final selected threshold. Note that except for Cases IX. and VII. the other curves are superimposed for a threshold superior to one.

theoretical likelihood ratio values stay relatively close to one. Therefore correct alerts are not raised for a threshold over about 1.06 when considering experiments with datasets of size $n = 400$ points (Fig. 4a) for these two cases. With $n = 100$ points, we clearly see that theoretical likelihood ratios are globally lower. For a same value of the threshold the number of false negative alerts increases (Fig. 4b). This provides us an intuition on the level of variations that our model can detect.

While the best possible performance would be obtained with a threshold at 1.05 (Fig. 4a) if we only consider theoretical ratios results, the study of the threshold involving the estimated models M' and M^a is less optimal. The computation of the likelihood ratio in the complete pipeline implies uncertainty on sampled data and on estimated parameters θ' and θ^a. This estimated ratio is defined by Equation (5) between M' and M^a at $t = 1$. We study the performance of the pipeline with these estimated ratios values, by accounting for the number of wrong alerts over $S = 100$ runs as before. The corresponding results, with these estimated likelihood ratios, are given in Figure 5. In Table 10 we retrieve the number of alerts per case for different threshold values and for different dataset sizes. For Case I., Case V. and Case VI., the population distribution is the same at $t = 0$ and $t = 1$, but

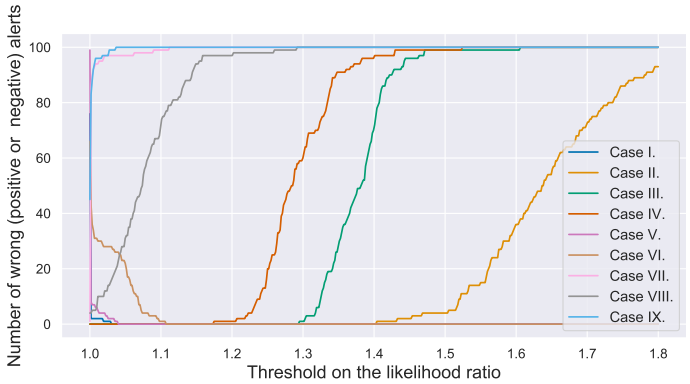


Figure 5: For each case is presented the number of false alerts (positive or negative) depending on estimated likelihood ratios given by (5), over $S = 100$ runs according to the considered threshold τ. Datasets are of size $n = 400$.

M^a and M' are not estimated by the same algorithm, and the likelihood ratios are depending on the sampled data. However, the model M' should still be accepted as the two mixture distributions are very close. From Figure 5, we observe that a threshold of 1.0 is not appropriate, as a high number of alerts is raised for these cases, where we should have zero alert. Increasing the threshold allows for model and data variability to be taken into account, and avoid false positive alerts.

On the other hand, if we set a too high threshold τ, there is a risk of not detecting all important changes. We clearly see for Cases II., III., IV., and VIII. that the number of true positive alerts is affected by a too high threshold. If we go above $\tau = 1.2$ we see an important decrease for Case IV., and later for the Cases II. and III. . Case VIII., which corresponds to a move from Setup F. to Setup C. is affected earlier by the likelihood ratio threshold, as the proximity of two clusters (Figure 3c) affects the estimation of mixture parameters and so on the likelihood ratio. It leads us to set a threshold relatively closed to one. As on the theoretical likelihood ratio study, we observe here that slight movements corresponding to Case VII. and Case IX. lead to incorrect alerts for a threshold over one. The estimated likelihood ratio values stay relatively close to one because the model M' can adapt to data $\mathbf{X}^{(1)}$. The evolving distributions are not detected.

Therefore when applied to a specific problem, one has to know that the relocation of one cluster may be detected if it relates to the variance of the estimated clusters. Otherwise, these displacements may be considered as normal variability of the discretization of the distributions. Note that this alert criterion may be adapted given a specific problem with the constrains that are imposed to the candidate model. Finally, we see from analysis of the theoretical ratios and the estimated ratios that we need to make a compromise. The optimistic theoretical likelihood ratios would lead us to take a threshold very close to one. But the obtained values with the estimated models contain more uncertainty that we cannot ignore and require to select a larger threshold. To obtain good performances of our pipeline we fix the threshold to $\tau = 1.1$. We obtain a balance between false negative and false positive alerts, that we want to maintain as low as possible, considering all possible situations.

4.4 Performances of STMP on synthetic data

4.4.1 Performances of the Modified REM algorithm within STMP

We present here results of the estimation of GMM parameters with the Modified Robust EM algorithm at $t = 0$ and $t = 1$ in STMP experiments on synthetic data. All experimental frameworks described in Subsection 4.2 are tested, with $n_0 = n_1 = n = 400$ points.

For each run of each experiment, we check if the number of estimated clusters at $t = 0$ or $t = 1$ with Modified REM is correct. We report the correctly estimated K rates in Table 3. We consider that our STMP correctly estimates K over time if and only if $\hat{K}^{(0)} = K^{(0)}_{true}$ and $\hat{K}^{a} = K^{(1)}_{true}$, with $\hat{K}^{(0)}$ and $\hat{K}^{a}$ estimated by Modified Robust EM at $t = 0$ and $t = 1$ respectively. In brief, the correctly estimated K number is given by the intersection of correctly estimated $\hat{K}^{(0)}$ and $\hat{K}^{a}$.

Cases I. to IV. give high rates, explained by the correct separation of the clusters as seen in Figure 2. On experiments with configurations bringing closer two clusters (Cases V. to IX.), we obtain high rate (over 90%) for static and well-enough separated clusters (Setup F., Setup M.). This score is also high for displacement from Setup F. to Setup M. (Case VII.) .
When we consider moving clusters which are getting too close this score decreases. The global score of STMP executions involving at least one Setup C. distribution is affected by the superposition of two clusters. The correct proportions are not bigger than 54%. By looking at estimated $\hat{K}^{(0)}$ and $\hat{K}^{a}$ in Table 3,the Modified REM algorithm estimates at least 30 over 100 times two classes with samples from Setup C. distribution. Although these estimates are incorrect, they lead to understandable results, as samples from the two left hand side clusters can be confused (see Figure 3c). An example of wrong estimated parameters for Setup C. is presented in Figure 6, which confirms the interpretability of the results.

Thereafter, we compute estimation errors for means and covariances matrices on experiments with correctly estimated number of components K (see Table 4). It confirms that these estimated Gaussian mixtures are correctly estimated by the Modified Robust EM inside our pipeline STMP. We also notice a poorer average estimate of GMM parameters for datasets from Setup C. As said previously, this parametrization implies that two clusters are mixed up. Estimates of Setup C. models present a slightly higher average Euclidean distance between the true means and the estimated ones. For covariance matrices errors, computed with Frobenius norm, the average errors are less contrasted, but we observe the highest error for M^a estimate in Case VIII. (two clusters are closed to each other

Experiment	Proportion of right estimated number of components (% for $\hat{K}^{(0)} = 2$ and $\hat{K}^a = 2$)
Case I.	**96**%
Case II.	**98**%
Case III.	**100**%
Case IV.	**100**%
Case V.	**92**%
Case VI.	42% (**30**%,**32**%)
Case VII.	**99**%
Case VIII.	54% (0%,**34**%)
Case IX.	54% (0%,**37**%)

Table 3: Proportion of correctly estimated number of components among $S = 100$ runs. At each execution, the estimation is correct iff : $\hat{K}^a = K^{(1)}_{true}$ and $\hat{K}^{(0)} = K^{(0)}_{true}$. Configurations are described in Table 9.

at $t = 1$).

	$M^{(0)}$		M^a	
Case	$\hat{\mu}$	$\hat{\Sigma}$	$\hat{\mu}$	$\hat{\Sigma}$
Case I.	1.5 (0.8)	15.4 (7.1)	1.5 (0.9)	14.3 (6.8)
Case II.	1.5 (0.9)	14.4 (7.3)	1.7 (0.9)	16.4 (6.7)
Case III.	1.6 (0.9)	14.3 (6.9)	1.2 (0.7)	11.0 (4.8)
Case IV.	1.5 (0.8)	14.0 (6.7)	1.5 (0.8)	13.6 (5.8)
Case V.	1.7 (1.2)	16.6 (12.2)	1.7 (0.9)	15.6 (7.8)
Case VI.	4.2 (16.6)	22.8 (22.0)	3.2 (3.8)	23.6 (25.5)
Case VII.	1.5 (0.9)	14.8 (6.6)	1.6 (1.0)	15.8 (7.8)
Case VIII.	1.5 (0.9)	14.3 (8.2)	3.7 (4.6)	29.0 (30.3)
Case IX.	1.7 (1.0)	16.2 (8.5)	2.9 (3.7)	24.8 (29.9)

Table 4: Mean (standard deviation) relative errors (expressed as a percentage) for the estimated means and covariance matrices within each case, over all runs having correctly estimated $\hat{K}$ inside STMP. The Euclidean norm is used for means, and the Frobenius norm for covariances.

4.4.2 Performances of STMP as an alert system

We have defined in the previous subsection the threshold to alert the user that there may be a population dynamical change between two given time points. As explained in Subsection 3.3, there is also an alert when a component tries to disappear (leading to degenerated covariance matrix) when estimated by the C-EM. And when covariance matrices estimated by the C-EM are too different from the previous step ones. With these warning systems, we defined a whole pipeline, named STMP, to monitor the dynamic of the population and raise alerts when reasonable changes occur. We are now demonstrating the performances of STMP. Using an alert threshold of $\tau = 1.1$, we obtain the

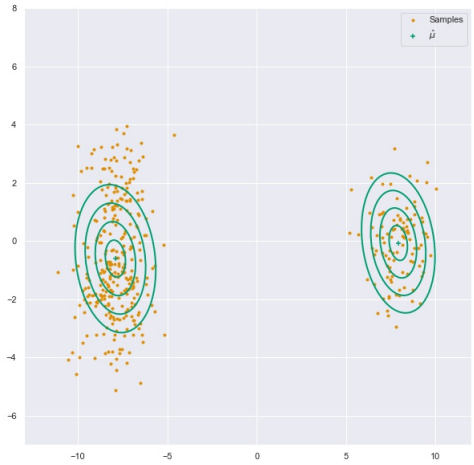


Figure 6: An estimated GMM with $\hat{K} = 2 \neq K_{true} = 3$ for a Setup C. distribution. The centers and covariances are represented in green. Orange points are samples.

following alert rates, that we can retrieve in the Figure 5. For Case I., Case V. and Case VI. we obtain 98% true negative alerts respectively. For Cases II. to IV. we obtain a true positive alert rate of 100%, detecting all changes in population distribution with our STMP.

STMP does not raise an alert when the distributions differ barely in time. This is due to our likelihood ratio threshold fixed to $\tau = 1.1$. The true positive number of alerts is of 2% for Case VII. and 1% for Case IX.. In contrary, the bigger movement in Case VIII. leads to a true positive number of 29%. This brings us to the problem that STMP can not raise an alert when GMM are hard to estimate correctly, as here. This experiment involves the Setup C., which is complex to estimate for EM algorithms.

Last but not least, our proposed method is computationally efficient with a very low computational time. All experiments on datasets of size $n = 400$ are performed with an average execution time of 1.33s. We recover average execution time by case type in Table 5. Fast execution was also a criterion leading the construction of our method, and satisfying for our future applications.

Experiment	Average computation time over $S = 100$ runs (std)
Case I.	1.24s (0.19)
Case II.	1.14s (0.18)
Case III.	1.55s (0.50)
Case IV.	1.35s (0.41)
Case V.	1.33s (0.14)
Case VI.	1.45s (0.32)
Case VII.	1.23s (0.10)
Case VIII.	1.31s (0.27)
Case IX.	1.35s (0.20)

Table 5: Average (and standard deviation) computation time of the different case experiments, with $n_0 = n_1 = 400$ points.

4.4.3 Effects of the data set size on STMP

In previous explained experiments on synthetic data, we fixed the data set size to $n = 400$. Afterwards, we study the impact of the number of points for Cases I. to IX. described previously, with $n \in \{100, 200, 400\}$. With the same true distributions as in Figures 2 and 3, we perform $S = 100$ runs of our process with data samples of size $n = 200$ and $n = 100$ at each time step.

As expected, decreasing n decreases the proportion of good estimated $\hat{K}$ over the 100 runs and inherently the quality of estimation of parameters K (Table 6). For $n = 200$ points and cases with Gaussian clusters not too close to each other the Modified Robust EM algorithm gives a high rate of correct estimation of K. For Cases I. to V. and Case VII., the rates are between 76% and 92%, allowing us to be confident in the estimates. For cases implying the Setup C. the estimated GMM are worse, because two true Gaussians are almost overlapping. With $n = 100$, it becomes complicated to properly estimate a GMM even with well defined clusters: the best alert rate is 61% and the worst is 8%. Therefore, we must be aware that decrease the number of samples decreases the proportion of good estimated $\hat{K}$ and inherently the quality of estimation of parameters θ in our Modified Robust EM. But overall, the STMP performance is less affected by changes of data sets size (Table 10). As

Experiment	Proportions of right estimated number of components with $n = 400$	Proportions of right estimated number of components with $n = 200$	Proportions of right estimated number of components with $n = 100$
Case I.	**96%**	**90%**	61%
Case II.	**98%**	**87%**	46%
Case III.	**100%**	**92%**	61%
Case IV.	**100%**	**87%**	59%
Case V.	**92%**	76%	42%
Case VI.	42%	20%	8%
Case VII.	**99%**	**81%**	51%
Case VIII.	54%	29%	22%
Case IX.	54%	34%	29%

Table 6: Proportion of right estimated number of components among $S = 100$ runs. At each execution, the estimation is correct iff : $\hat{K}^a = K^{(1)}$ and $\hat{K}^{(0)} = K^{(0)}$.

we saw in Subsection 4.4.2, we reach 98% true negative alerts for data sets of size $n = 400$. For data sets of size $n = 200$ we have 19 false positive alerts, and for $n = 100$ we have 56 false positive alerts. For Cases II. to IV. the proportion of success is 100% for all n values (Table 10). It even raises more alerts with fewer points on Case VII. to Case IX., due to overlapping Gaussian components which are estimated as one single component. For example if a $t = 1$ we are in Setup C. (Fig 3c), as two Gaussians components are hardly separable the pipeline will estimated one cluster for the two components, and raise an alert as it is evolving away from the estimated distribution at $t = 0$ (which could be Setup F. or M.).

Even if the Modified Robust EM becomes less accurate with smaller data sets, our pipeline still produces interpretable and meaningful results. The decrease of performance with smaller data sets should be improved inside the Modified Robust EM.

5 Application of STMP on a real life use case

In this section we demonstrate the relevance of STMP with GMM on real epidemiological data from the COVID-19 in Paris, France.

5.1 Presentation of the data set

AP-HP (Assistance Publique des Hopitaux de Paris) is the largest hospital entity in Europe with 39 hospitals (22,474 beds) mainly located in the greater Paris area with 1.5 M hospitalizations per

year (10% of all hospitalizations in
France). Since 2014, the AP-HP has deployed an analytics platform based on a clinical data repository, aggregating day-to-day clinical data from 8.8 million patients captured by clinical databases. An "EDS-COVID" database stemmed from this initiative. The AP-HP COVID database retrieved electronic health records from all AP-HP facilities and aggregated them into a clinical data warehouse. The clinical data warehouse allows a large set of data to be retrieved in real time to deeply characterize hospitalized patients, including their residential address. New patients who tested positive by polymerase chain reaction (PCR) as being infected by SARS-CoV-2 from the 24th of february to the 10th of may 2020 (weeks 9 to 19), in one of the AP-HP hospitals and living in Paris constitutes the dataset for this study. During this time period, tests availability outside public hospital facilities were very limited, and therefore we can consider in this study that this sample constitutes a representative sample of patients having been positively tested during this period. To preserve privacy, residential addresses were extracted at the IRIS level, which is a geographical division in France of residential units of 2000 inhabitants on average.

For each patient we have two pieces of information: the week he/she was diagnosed positive, and his/her place of residence at the IRIS level. We therefore use a week as a time step t in our process. Beginning from the first week (week 9), which corresponds to the beginning of pandemic in France, we apply our STMP, keeping at each time t one of the models M^a or M' according to the criterion defined in Subsection 3.1 with threshold τ given in Section 4. We have 5621 positive diagnosed patients over all weeks and all Paris IRIS areas. Table 7 informs us that the number of cases per week is not homogeneous, as in first weeks, few cases living in Paris were detected positive.

Week	Number of positive diagnosed people per week
9	5
10	18
11	272
12	965
13	1666
14	1297
15	695
16	366
17	209
18	114
19	14

Table 7: Distribution of positive diagnosed people to COVID-19 over weeks.

5.2 Comparison of the Robust EM and the Modified Robust EM on the real data set

As described in Section 3.2, Robust EM [18] has a convergence problem, revealed on real data sets. Even if it is dynamical, it can be stuck in an incorrect local maximum involving overlapping clusters. This phenomenon has been detected on the real dataset of COVID-19 positive cases in Paris area. We compare here the estimated GMMs by original Robust EM and by our Modified REM, which is correcting this overlapping effect (Subsection 3.2).

On all weeks except the 13^{rd} week, the Robust EM presents no overlapping cluster. It returns acceptable estimated mixture models. As there are no abnormalities in the estimation process, our Modified Robust EM returns similar results. It is illustrated by Figures 7a and 7b showing estimations on 12^{th} week for both algorithms. On the 13^{rd} week, the Robust EM algorithm presents overlapping clusters. The final number of classes is 11, but the Figure 7c shows us that there are only nine clusters. We can only detect superimposed clusters by doing a post-processing analysis,

which consists of checking the estimated parameters. Table 11 gives these estimated parameters for both the Robust EM [18] and the Modified REM. From this table we see that there are two pairs of superimposed clusters with mixture estimation by Robust EM. By executing our Modified REM on the same week, independently of the other time steps, we obtain nine clusters (Figure 7d and Table 11), confirming that if we merge redundant clusters, we obtain a stable solution, accepted by the algorithm. Our Modified REM solves the problem of superimposed clusters. This avoids having to consider post-processing inside STMP, which would require an user action at each time step. It also allows to answer a specific problem, to correctly model COVID-19 data in space and time.

5.3 Results

The aim here is to underline presence or absence of temporal constancy in COVID-19 data. A temporal constancy would suggest that the population distribution was stable at the peak of the pandemic. This is in line with epidemiological studies that where showing a "peak" around these weeks after the first propagation phase (9^{th} to 12^{th} weeks) (see weekly reports of Public Health Institution [24] Page.7 Figure 8.).

We use the fixed alert threshold $\tau = 1.1$ in our pipeline, as estimated by previous experiments in Section 4. On the first week (9^{th} week), as the number of cases is very low, the initial Modified Robust EM converges towards the removal of all clusters. The final parameters corresponds here to the initial ones, so we observe on Figure 8a initial high variances and that each case is its proper cluster. The 10^{th} week is still presenting a low number of scattered cases, which are modelled by two global clusters, geographically distributed on both sides of the river Seine. As from 10^{th} week to 11^{th} week (and 11^{th} week to 12^{th} week) the number of cases is highly increasing, the model accepts new estimated parameters θ^a. Our STMP reveals that the GMM estimated on 12^{th} week with $\hat{K}^{(12)} = 10$ was accepted on 13^{th} and 14^{th} weeks. As a reminder, 12^{th} week represents the beginning of the lockdown and 13^{th} week represents the peak of the pandemic, in terms of new positive cases. This means that C-EM executed across 13^{th} and 14^{th} weeks fits very well the new data set each week with a source model estimated on 12^{th} week. Even if the number of cases changes over time, STMP is able to detect a constant distribution. This is consistent with the patients distribution on 12^{th},13^{th} and 14^{th} weeks as we can see on Figures 8 and 9. On 15^{th} week, STMP rejects the hypothesis that the patients data set is approximated by the mixture law estimated on previous weeks. The alternative model M^a is accepted. Parameters $\theta^{(15)}$ on 15^{th} week are newly estimated, evolving too far from $\theta^{(14)}$, estimated parameters of 14^{th} week. It can be interpreted as the strong decrease of new positive cases such as the disappearance of large clusters from previous weeks and the detection of large and global clusters, corroborated by the Figure 9a. On the 16^{th} week, these three clusters from 15^{th} week, large and non-informative, are accepted by STMP. On the following weeks (17^{th},18^{th} and 19^{th} weeks), the number of cases is still decreasing. As on first weeks, the small number of cases leads to accept totally new estimated parameters θ^a each week, without link with previous weeks.

From Table 8, the likelihood ratio values are globally distant from our defined threshold $\tau = 1.1$, leaving no doubt about the choice of best parameters $\theta^{(t)}$ at each time step t. Only on 15^{th} week the likelihood ratio value is smaller that our defined threshold while the temporal-dependent model M' is rejected. This reject is due to large variations in the covariance matrices during the C-EM stage. The model M' fits the new data set by excessively moving the covariance parameters herited from $M^{(14)}$. There is no likelihood ratio value in the last week. This ratio is incalculable due to the "empty class phenomenon". The model M' tries to remove a component which leads to an early stop of the estimation process of this model. This triggers the inevitable choice of the alternative model and raises an alert.

From the mathematical and algorithmic point of view we obtain interesting results, showing that C-EM can over time sufficiently model evolving real-world data with a relatively stable and high size.

5.4 Interpretations

The results obtained with STMP on this COVID-19 data set are coherent with public health policy and COVID-19 transmission patterns during this time period. Lockdown in France took place from the 17th of march (beginning of 12^{th} week) to the 1st of June. As it takes about two weeks to go from contamination to cytokine storm, no evolution in clusters was expected from week 12 to 14. Thereafter a decrease in the number of clusters was expected, associated with a moving distribution. Moreover, estimated clusters concentrate closed to Paris periphery, which are low-income neighbourhoods, known to favour COVID-19 transmission. The reject on $15th$ week of the previous time step model can be interpreted as the effect of the lockdown, and we can observe the natural barrier formed by the Seine, as people can only move in a perimeter of one kilometre. The numerous clusters during 18^{th} week are residual clusters not solved by the lockdown. They mainly correspond to concentrated positive cases areas, whereas in the rest of the city there are few and scattered cases.

6 Conclusion

We have proposed a complete and generic pipeline for modeling evolution of population distribution, and detecting abnormal changes in this distribution. This STMP was combined with new EM algorithm variants. Our application on public health data shows that this STMP well models population distributions, and raises meaningful alerts.

The STMP for monitoring population distributions and the algorithms to estimate the models are two independent objects. This enables future directions of our work when integrating covariables following non-Gaussian distributions in the mixture. We will still be able to use our proposed algorithms as they are blind to the distributions in the mixture.

On the other hand, the performance of the EM algorithms depends on the data set sizes. In future work we will try to temperate the Modified Robust EM as proposed by [25] to improve estimations in unstable situations.

Last, the decision rule was here empirically fixed. In future work this decision rule will be modeled as an acceptation probability, taking advantage of Monte Carlo Markov Chains theory.

Week	Estimated number of classes $\hat{K}$ by M^a	Estimated number of classes $\hat{K}$ by M'	Likelihood Ratio r	Accepted model
9	**5**	None	None	$M^{(0)}$
10	**2**	5	1.383	M^a
11	**4**	2	1.376	M^a
12	**10**	4	1.171	M^a
13	9	**10**	1.088	M'
14	5	**10**	0.950	M'
15	**3**	10	0.87	M^a
16	5	**3**	1.080	M'
17	**4**	3	1.307	M^a
18	**9**	4	2.215	M^a
19	**3**	9	computationally invalid	M^a

Table 8: Results of our process on positive diagnosed people in AP-HP hospitals with a time step being a week

Conflict of interest statement

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Acknowledgments

The authors thank the EDS APHP Covid consortium integrating the APHP Health Data Warehouse team as well as all the APHP staff and volunteers who contributed to the implementation of the EDS-COVID database and operating solutions for this database. The authors particularly thank Mélodie Bernaux (DST), Nicolas Paris, Ali Bellamine and Christel Daniel (DSI-WIND).

Fundings

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article:

- This work was supported by a grant from Région Île-de-France.
- This work was supported by a grant of Paris Artificial Intelligence Research Institute : ANR-19-P3IA-0001 - PRAIRIE IA - Paris Artificial Intelligence Research Institute (2019).

Ethics approval and consent to participate

This study was approved by the institutional review board (authorization number IRB 00011591) from the Scientific and Ethical Committee from the AP-HP.

References

[1] Russell S Kirby, Eric Delmelle, and Jan M Eberth. Advances in spatial epidemiology and geographic information systems. *Annals of epidemiology*, 27(1):1–9, 2017.

[2] Martin Kulldorff. A spatial scan statistic. *Communications in Statistics-Theory and methods*, 26(6):1481–1496, 1997.

[3] Martin Kulldorff, William F Athas, Eric J Feurer, Barry A Miller, and Charles R Key. Evaluating cluster alarms: a space-time scan statistic and brain cancer in los alamos, new mexico. *American journal of public health*, 88(9):1377–1380, 1998.

[4] Martin Kulldorff, Richard Heffernan, Jessica Hartman, Renato Assunção, and Farzad Mostashari. A space–time permutation scan statistic for disease outbreak detection. *PLoS Medicine*, 2(3):e59, Feb 2005.

[5] Paul Elliott and Daniel Wartenberg. Spatial epidemiology: Current approaches and future challenges. *Environmental Health Perspectives*, 112(9):998–1006, Jun 2004.

[6] David L. Sackett. Bias in analytic research. *Journal of Chronic Diseases*, 32(1–2):51–63, Jan 1979.

[7] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[8] Jean-Patrick Baudry and Gilles Celeux. EM for mixtures: Initialization requires special care. *Statistics and Computing*, 25(4):713–726, July 2015.

[9] Thomas Lartigue, Stanley Durrleman, and Stéphanie Allassonnière. Deterministic approximate em algorithm; application to the riemann approximation em and the tempered em. *arXiv preprint arXiv:2003.10126*, 2020.

[10] Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle,[w:] proceedings of the 2nd international symposium on information, bn petrow, f. *Czaki, Akademiai Kiado, Budapest*, 1973.

[11] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[12] Lucien Birgé and Pascal Massart. Minimal penalties for gaussian model selection. *Probability theory and related fields*, 138(1-2):33–73, 2007.

[13] Jean-Patrick Baudry, Cathy Maugis, and Bertrand Michel. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470, 2012.

[14] Esther Derman and Erwan Le Pennec. Clustering and model selection via penalized likelihood for different-sized categorical data vectors. *arXiv preprint arXiv:1709.02294*, 2017.

[15] Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):381–396, 2002.

[16] Haixian Wang, Bin Luo, Quan bing Zhang, and Sui Wei. Estimation for the number of components in a mixture model using stepwise split-and-merge em algorithm. *Pattern Recognition Letters*, 25(16):1799–1809, 2004.

[17] Baibo Zhang, Changshui Zhang, and Xing Yi. Competitive em algorithm for finite mixture models. *Pattern recognition*, 37(1):131–144, 2004.

[18] Miin-Shen Yang, Chien-Yo Lai, and Chih-Ying Lin. A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, 45(11):3950–3961, 2012.

[19] Martin HC Law, Mario AT Figueiredo, and Anil K Jain. Simultaneous feature selection and clustering using mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 26(9):1154–1166, 2004.

[20] Chris S Wallace and Peter R Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society: Series B (Methodological)*, 49(3):240–252, 1987.

[21] Chris S. Wallace and David L. Dowe. Minimum message length and kolmogorov complexity. *The Computer Journal*, 42(4):270–283, 1999.

[22] M. Herdin, N. Czink, H. Ozcelik, and E. Bonek. Correlation matrix distance, a meaningful measure for evaluation of non-stationary mimo channels. In *2005 IEEE 61st Vehicular Technology Conference*, volume 1, pages 136–140 Vol. 1, May 2005.

[23] Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a Mixture Model for Clustering with the Integrated Classification Likelihood. Technical Report RR-3521, INRIA, October 1998.

[24] Santé Publique France. Point épidémiologique hebdomadaire du 25 juin 2020. Technical report, Santé Publique France, jun 2020.

[25] Stéphanie Allassonnière and Juliette Chevallier. A New Class of EM Algorithms. Escaping Local Minima and Handling Intractable Sampling. working paper or preprint, June 2019.

A Appendices

A.1 Equations for mixture parameters estimation in the original EM algorithm [7]

The EM algorithm alternates between the two following steps (until convergence criterion is met). At the p-th iteration of the algorithm, the update equations are given by:

- E-step : Compute the conditional expectation of the complete log-likelihood. Latent variables z_i^k are discrete, so their conditional expectations are given by

$$\begin{aligned} p_\theta(z_i^k = 1|x_i) &= \frac{\pi_k \mathcal{N}_d(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}_d(x_i|\mu_j, \Sigma_j)} \\ &= \tau_i^k(\theta)\,. \end{aligned} \tag{8}$$

- M-step : Update the parameter estimates:

$$\hat{\pi}_{k,MLE}^p = \frac{1}{n}\sum_{i=1}^n \tau_i^k\,, \tag{9}$$

$$\hat{\mu}_{k,MLE}^p = \frac{\sum_{i=1}^n \tau_i^k x_i}{\sum_{i=1}^n \tau_i^k}\,, \tag{10}$$

$$\hat{\Sigma}_{k,MLE}^p = \frac{\sum_{i=1}^n \tau_i^k (x_i - \mu_i)^\top (x_i - \mu_i)}{\sum_{i=1}^n \tau_i^k}\,. \tag{11}$$

A.2 Pseudo-Code of the Modified Robust EM presented in Section 3

Algorithm 2: Modified Robust EM

Initialization : data set $\mathbf{X} \in \mathbb{R}^{n \times d}$, $K^0 \leftarrow n$, $\varepsilon > 0$
$p \leftarrow 0$, $\beta^0 \leftarrow 1$
$\pi_k^0 \leftarrow 1/n$, $\mu^0 \leftarrow \mathbf{X}$
$\Sigma_k^0 \leftarrow d^2_{k(\lceil \sqrt{K^{\text{initial}}} \rceil)} \mathbf{I}_d$ with
$D_k = \text{sort}\left\{ d_{ki}^2 = \|x_i - \mu_k\|^2 : d_{ki}^2 > 0, \quad i \neq k, \quad 1 \leq i \leq n \right\} = \{d^2_{k(1)}, \ldots, d^2_{k(n)}\}$;
Compute $\tau_i^{k,0}$ with (8)
$p \leftarrow 1$
Compute μ_k^p with (10)
while $\max_{1 \leq k \leq K^p} \|\mu_k^{p+1} - \mu_k^p\| > \varepsilon$ *or Eq.* (6) *is verified for some clusters* **do**
  Compute π_k^p by (4)
  $\pi_{(1)}^{EM} \leftarrow \underset{1 \leq k \leq K^p}{max} \pi_k^{p,EM}$, $\pi_{(1)}^{(old)} \leftarrow \underset{1 \leq k \leq K^p}{max} \pi_k^{(old)}$
  $E \leftarrow \sum_{k=1}^{K^p} \pi_k^{(old)} ln \pi_k^{(old)}$
  $\beta^p \leftarrow \min \left\{ \frac{\sum_{k=1}^{K^{p-1}} \exp\left(-\eta n \left| \pi_k^p - \pi_k^{(old)} \right| \right)}{K^{p-1}}, \frac{\left(1 - \pi_{(1)}^{EM}\right)}{\left(-\pi_{(1)}^{(old)} E\right)} \right\}$
  Update class number K^{p-1} to K^p by deleting classes with $\pi_k^p < 1/n$, then adjust π_k^p and $\tau_i^{k,p-1}$
  if $K^{p-1} \neq K^p$ **then**
    $p_{component} \leftarrow 1$ `/* variable to keep in memory the number of iterations with a stable number of components */`
  end

  if $p \geq p_{min}$ *and* $p_{component} \geq 100$ **then**
2   **if** *no superimposed clusters (Eq.*(6) *false)* **then**
      $\beta^p = 0$
3   **else if** *superimposed clusters and* $p_{component} < 200$ **then** `/* give more time to the algorithm to converge */`
      $p_{min} \leftarrow p_{min} + 50$
4   **else** merge superimposed clusters
      adjust π^p, μ^p, Σ^p and τ^{p-1}
    end
  end
  Compute Σ_k^p with (11) and $\Sigma_k^p = (1 - \gamma)\Sigma_k + \gamma Q$ with
  $\gamma = 0.0001, \; Q = d^2_{\min} \mathbf{I}_d, \; d^2_{\min} = \min\{\{d_{ij}^2 : d_{ij}^2 = \|x_i - x_j\|^2 > 0, \; 1 \leq i, j \leq n\}$
  Compute $\tau_i^{k,p}$ with (8)
  Compute μ_k^{p+1} with (10)
  $p \leftarrow p + 1$
  $p_{component} \leftarrow p_{component} + 1$
end

A.3 Supplementary analyses of Section 4

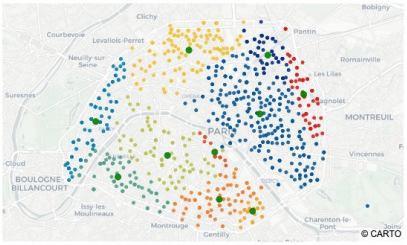
Study case reference	Description at $t=0$	Description at $t=1$	Number of clusters $K_{true}^{(0)}$	Number of clusters $K_{true}^{(1)}$	Parameters at $t=0$	Parameters at $t=1$
Case I.	Setup F.	Same distributions (Setup F.)	3	3	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$
Case II.	Setup F.	Emergence of a cluster	3	4	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=\pi_2=\pi_3=\pi_4=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0), \mu_4=(-2.45,6.57)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0, \Sigma_4=\begin{pmatrix}0.88 & 0.\\ 0. & 0.48\end{pmatrix}$
Case III.	Setup F.	Vanishing of a cluster	3	2	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=\pi_2=0.5$,$\mu_1=(-8,3.5), \mu_2=(8,0)$,$\Sigma_1=\Sigma_2=\Sigma_0$
Case IV.	Setup F.	Changing a cluster	3	3	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=\pi_2=\pi_3=1/3$,$\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(6.09,-2.71)$,$\Sigma_1=\Sigma_2=\Sigma_0, \Sigma_3=\begin{pmatrix}1.36 & 0.\\ 0. & 0.92\end{pmatrix}$
Case V.	Setup M.	Setup M.	3	3	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-2.5), \mu_2=(-8,2.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=0.5, \pi_2=\pi_3=0.25$,$\mu_1=(-8,-2.5), \mu_2=(-8,2.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$
Case VI.	Setup C.	Setup C.	3	3	$\pi_1=0.5, \pi_2=\pi_3=0.25$,$\mu_1=(-8,-1.8), \mu_2=(-8,1.8), \mu_3=(8,0)$,$\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=0.5, \pi_2=\pi_3=0.25$,$\mu_1=(-8,-1.8), \mu_2=(-8,1.8), \mu_3=(8,0)$,$\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$
Case VII.	Setup F.	Setup M.	3	3	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-2.5), \mu_2=(-8,2.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$
Case VIII.	Setup F.	Setup C.	3	3	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-3.5), \mu_2=(-8,3.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-1.8), \mu_2=(-8,1.8), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$
Case IX.	Setup M.	Setup C.	3	3	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-2.5), \mu_2=(-8,2.5), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$	$\pi_1=0.5, \pi_2=\pi_3=0.25$, $\mu_1=(-8,-1.8), \mu_2=(-8,1.8), \mu_3=(8,0)$, $\Sigma_1=\Sigma_2=\Sigma_3=\Sigma_0$

Table 9: Different cases of data distributions changes from one time point to the next one (here only considering $t=0$ and $t=1$). Note that $\Sigma_0=\begin{pmatrix}1. & 0.\\ 0. & 1.5\end{pmatrix}$

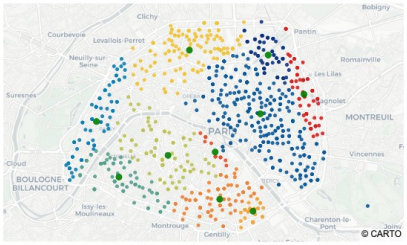
	Case	Case I.	Case II.	Case III.	Case IV.	Case V.	Case VI.	Case VII.	Case VIII.	Case IX.
n	Threshold									
400	1.00	78%	**100**%	**100**%	**100**%	99%	67%	**99**%	**96**%	55%
	1.05	**2**%	**100**%	**100**%	**100**%	**2**%	21%	4%	69%	*1%*
	1.10	**2**%	**100**%	**100**%	**100**%	**2**%	**2**%	2%	29%	1%
	1.15	**2**%	**100**%	**100**%	**100**%	**2**%	**1**%	1%	7%	1%
	1.20	**2**%	**100**%	**100**%	**99**%	**2**%	**1**%	1%	4%	1%
	1.25	**2**%	**100**%	**100**%	**92**%	**2**%	**1**%	1%	4%	*1%*
	1.30	**2**%	**100**%	**100**%	**81**%	**2**%	**1**%	1%	2%	1%
	1.35	**2**%	**100**%	**99**%	73%	**2**%	**1**%	1%	2%	1%
	1.40	**2**%	**100**%	**98**%	72%	**2**%	**1**%	1%	2%	1%
200	1.00	80%	**100**%	**100**%	**100**%	91%	55%	**98**%	**94**%	58%
	1.05	19%	**100**%	**100**%	**100**%	22%	38%	27%	71%	26%
	1.10	19%	**100**%	**100**%	**100**%	18%	20%	16%	47%	21%
	1.15	18%	**100**%	**100**%	**100**%	16%	18%	14%	38%	19%
	1.20	17%	**100**%	**100**%	**96**%	13%	17%	14%	30%	19%
	1.25	17%	**100**%	**100**%	**91**%	13%	17%	13%	29%	19%
	1.30	17%	**100**%	**100**%	**81**%	13%	17%	13%	28%	19%
	1.35	17%	**100**%	**99**%	75%	13%	17%	13%	26%	19%
	1.40	17%	**100**%	**98**%	71%	13%	17%	13%	26%	19%
100	1.00	78%	**100**%	**100**%	**100**%	89%	74%	**96**%	**96**%	**84**%
	1.05	62%	**100**%	**100**%	**100**%	72%	71%	75%	**89**%	*74%*
	1.10	56%	**100**%	**100**%	**100**%	70%	63%	70%	**80**%	71%
	1.15	56%	**100**%	**100**%	**100**%	69%	54%	70%	77%	67%
	1.20	54%	**100**%	**100**%	**99**%	64%	51%	68%	72%	64%
	1.25	54%	**100**%	**100**%	**97**%	60%	51%	67%	68%	63%
	1.30	53%	**100**%	**100**%	**93**%	60%	51%	64%	67%	63%
	1.35	53%	**100**%	**98**%	**89**%	58%	49%	63%	66%	62%
	1.40	53%	**100**%	**97**%	**86**%	58%	49%	63%	66%	62%

Table 10: Number of alerts raised by our STMP for each experiment ($S = 100$ runs) on data sets of n points. For each size n and each Case is provided the number of alerts for different values of the alert threshold.

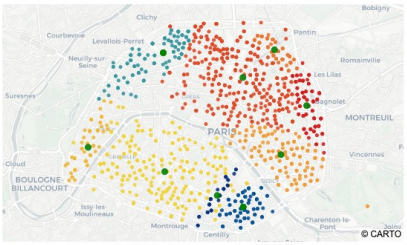
A.4 Results on the COVID-19 data set of Section 5



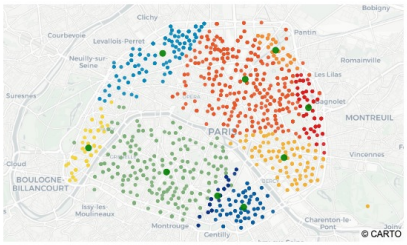
(a) Results of the original Robust EM on week 12.



(b) Results of our Modified Robust EM on week 12.



(c) Results of the original Robust EM on week 13. We have here overlapping clusters.



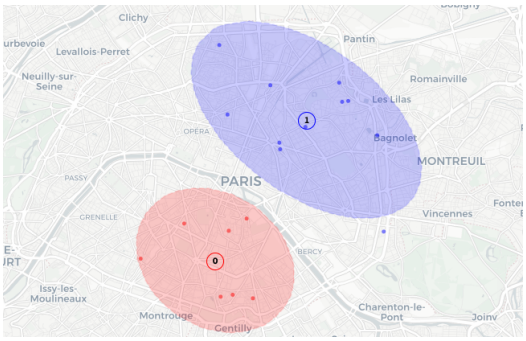
(d) Results of our Modified Robust EM on week 13.

Figure 7: Estimated GMM labels and centers by Robust EM [18] and Modified Robust EM on COVID-19 data set on weeks 12 and 13. Green dots are centers of the clusters.

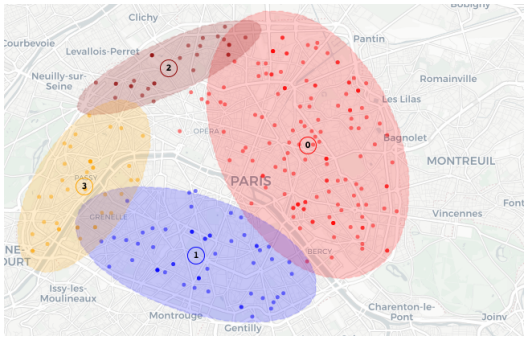
Parameters	Robust EM [18]	Modified Robust EM
$\hat{\pi}$	$\begin{pmatrix} 0.02 & \mathbf{0.0355} & \mathbf{0.0355} & \boldsymbol{0.0494} & \boldsymbol{0.0494} & 0.1992 \\ 0.0389 & 0.1049 & 0.0577 & 0.3463 & 0.0631 & \end{pmatrix}$	$\begin{pmatrix} 0.0155 & 0.0651 & 0.0932 & 0.2086 & 0.0341 \\ 0.0896 & 0.0525 & 0.3924 & 0.049 & \end{pmatrix}$
$\hat{\mu}$	$\begin{pmatrix} 2.3487 & 48.8308 \\ \mathbf{2.3646} & \mathbf{48.8262} \\ \mathbf{2.3646} & \mathbf{48.8262} \\ \boldsymbol{2.3158} & \boldsymbol{48.8882} \\ \boldsymbol{2.3158} & \boldsymbol{48.8882} \\ 2.3166 & 48.8404 \\ 2.27 & 48.8502 \\ 2.3875 & 48.8473 \\ 2.3836 & 48.8894 \\ 2.3644 & 48.8782 \\ 2.4032 & 48.867 \end{pmatrix}$	$\begin{pmatrix} 2.3486 & 48.8307 \\ 2.3646 & 48.8262 \\ 2.3151 & 48.8879 \\ 2.3174 & 48.8403 \\ 2.2697 & 48.8499 \\ 2.3892 & 48.8461 \\ 2.3841 & 48.8892 \\ 2.3655 & 48.8775 \\ 2.4042 & 48.8661 \end{pmatrix}$
$\hat{\Sigma}$	$\begin{pmatrix} 3.6945e-05 & 3.2396e-05 \\ 3.2396e-05 & 3.2901e-05 \\ \hline \mathbf{6.924e\text{-}05} & \mathbf{1.184e\text{-}05} \\ \mathbf{1.184e\text{-}05} & \mathbf{2.077e\text{-}05} \\ \hline \mathbf{6.924e\text{-}05} & \mathbf{1.184e\text{-}05} \\ \mathbf{1.184e\text{-}05} & \mathbf{2.077e\text{-}05} \\ \hline \boldsymbol{0.00032407} & \boldsymbol{0.00011124} \\ \boldsymbol{0.00011124} & \boldsymbol{5.9566e\text{-}05} \\ \hline \boldsymbol{0.00032407} & \boldsymbol{0.00011124} \\ \boldsymbol{0.00011124} & \boldsymbol{5.9566e\text{-}05} \\ \hline 0.0005979 & -8.6696e-05 \\ -8.6696e-05 & 9.8369e-05 \\ \hline 4.2497e-05 & 4.0846e-05 \\ 4.0846e-05 & 6.039e-05 \\ \hline 0.00025143 & -7.2942e-05 \\ -7.2942e-05 & 6.2798e-05 \\ \hline 6.2128e-05 & -3.2058e-05 \\ -3.2058e-05 & 2.2364e-05 \\ \hline 0.0004057 & -0.00010182 \\ -0.00010182 & 0.00013259 \\ \hline 3.8073e-05 & -4.0277e-05 \\ -4.0277e-05 & 8.9835e-05 \end{pmatrix}$	$\begin{pmatrix} 3.6889e-05 & 3.244e-05 \\ 3.244e-05 & 3.2399e-05 \\ \hline 6.842e-05 & 1.1486e-05 \\ 1.1486e-05 & 2.0443e-05 \\ \hline 0.0003229 & 0.00011263 \\ 0.00011263 & 6.0603e-05 \\ \hline 0.00063988 & -9.7086e-05 \\ -9.7086e-05 & 9.9909e-05 \\ \hline 3.9314e-05 & 3.8337e-05 \\ 3.8337e-05 & 5.7455e-05 \\ \hline 0.00022363 & -5.9831e-05 \\ -5.9831e-05 & 5.3155e-05 \\ \hline 6.0678e-05 & -3.2255e-05 \\ -3.2255e-05 & 2.264e-05 \\ \hline 0.00043225 & -0.00011348 \\ -0.00011348 & 0.00014621 \\ \hline 3.1265e-05 & -3.6572e-05 \\ -3.6572e-05 & 8.4985e-05 \end{pmatrix}$

Table 11: Estimated parameters by Robust EM [18] and modified Robust EM on week 13 of the COVID-19 dataset. These estimations were performed independently of previous time steps.

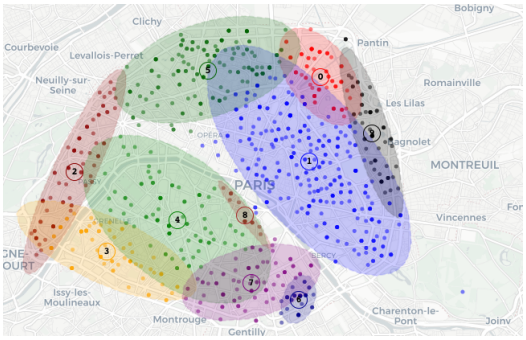
(a) Week 9 (There are only 5 cases, each case is center of its own cluster.)



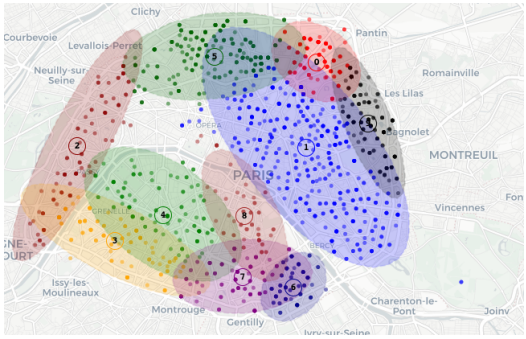
(b) Week 10



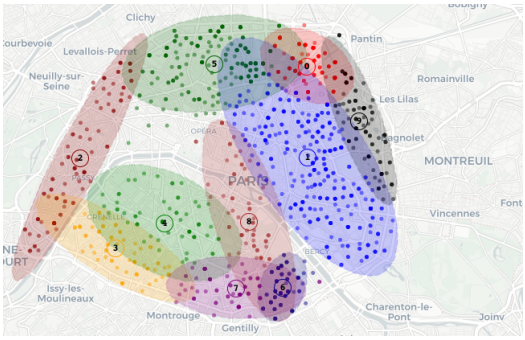
(c) Week 11



(d) Week 12

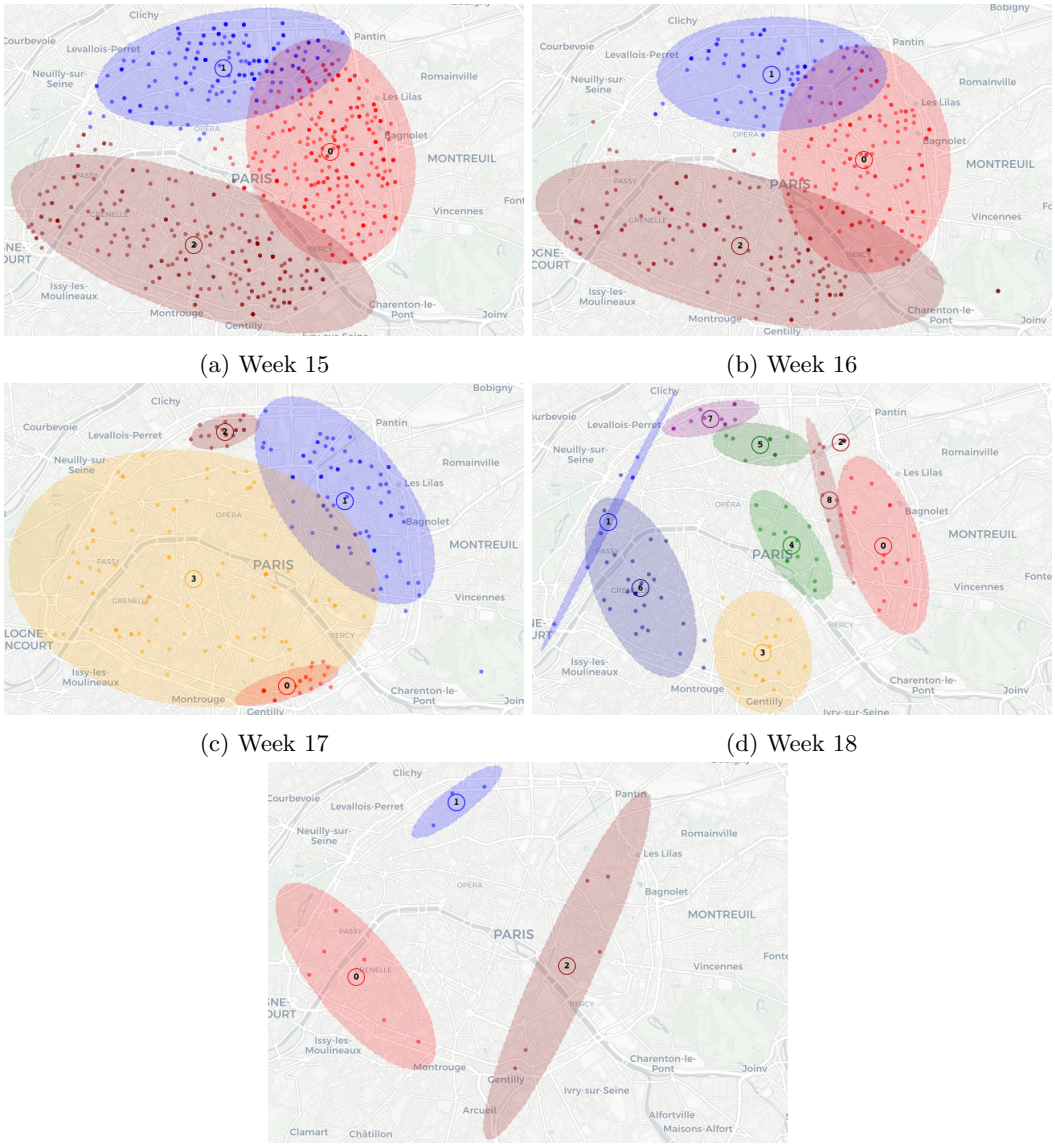


(e) Week 13



(f) Week 14

Figure 8: Estimated Gaussian Mixture Models on COVID-19 dataset per week (weeks 9 to 14).



(a) Week 15

(b) Week 16

(c) Week 17

(d) Week 18

(e) Week 19

Figure 9: Estimated Gaussian Mixture Models on COVID-19 dataset per week (weeks 15 to 19).