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]INSERM U1138, University Paris Descartes, 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 or 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. Assuming that mixture models can correctly model populations, we propose new versions of the Expectation-Maximization algorithm to overcome drawbacks of the existing ones. We then combine these algorithms with a temporal structure, allowing to detect dynamical changes in population distributions, and name it a spatio-temporal mixture process (STMP). We test STMP on synthetic data, and consider several different behaviors of the distributions, to adjust this process. Finally, we validate STMP on a real data set of positive diagnosed patients to corona virus disease 2019. We show that our pipeline correctly model evolving real data.

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

1 Introduction

1.1 Background

The rapid growth of health information systems has led to the availability of 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 these data to improve public health strategies and to transform these observed data into actionable decision-aid tools.

A spatial model is 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 to consider the population distribution as evolving along time. The association of spatial and temporal components for a disease yields a spatio-temporal distribution. One actionable decision-aid tool 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 identically distributed. This would be particularly useful to rapidly identify a new contamination source for transmissible disease, as soon as the first affected cases are present in health information systems.

1.2 Related Works

1.2.1 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 in [1]. Scan statistics methods proposed in [2, 3] are reference methods for many studies. They propose detection of spatial and/or temporal clusters from aggregated data (discrete in space and time) using sliding windows. They exhaustively scan the space and time in order to seek significant spatio-temporal clusters. The hypothesis that the incidence rate is higher inside the windows than in the studied region is tested using Monte Carlo methods to simulate likelihood ratio distributions.They develop different statistics using a known underlying population at risk, or cases/controls. In absence of population-at-risk, the authors of [4] estimate the expected number of cases. In both case, their methods require to fix several parameters on the considered sliding window (minimal area and minimal temporal size are two examples of the various parameters). An important issue is that these methods do not provide a modelling of the population over the whole space.

An alternative way is to use mixture models. By using a finite mixture of distributions, we model each point as belonging to each of the subgroups (clusters) with a certain probability. 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, it is interpretable because subjects can be attributed to estimated classes a posteriori which enables to distinguish homogeneous groups in the whole set. Third, they do not rely on controls distribution estimation or observation, unlike scan statistics methods. Moreover, by using mixture models, we make assumption here that the distribution of global population is not changing across time. We are working on the temporal evolution of the cases distribution. Last, these mixture models are parametric and well understood.

1.2.2 Estimation algorithms of mixture models and their issues

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

But several issues arise when using such a model. First, one has to set the number of components in the mixture. To solve this burden, model selection criteria have been proposed, notably by [5, 6, 7]. However, they rely on finite collection of estimations with increasing complexity.

Then, one has to be able to estimate the parameters of this particular model given the data base. To perform the estimation of Gaussian mixture parameters, given that we fix the number of clusters, the leading algorithm is the Expectation-Maximization algorithm introduced by Dempster et al. [8].

The choice of initial parameters is a major issue for the EM algorithm, as its solution is deterministic and highly dependent on this initial choice. The construction of the sequence ensures that the critical points are maxima, but could be both global or local ones. As Baudry and Celeux [9] pointed out, several strategies exist to avoid sensibility to initial values and selection of a bad local maximum. Easy ways to address this issue are to use small-EM algorithms as initialisation of a long run or to execute the EM algorithm several times with different random initialization procedures. When the problem is high-dimensional, these methods are not convenient because the parameters space to explore becomes too large. Another suggestion was to initialise with k-means

algorithm, which is also a clustering method. The stability is improved but the k-means algorithm has to be initialized which switches the problem without solving it. Recently, Lartigue et al. [10] introduced an annealing E-step to better stride the support and become almost independent from the initialization.

On the other hand, Baudry and Celeux [9] proposed to introduce a recursive initialisation which consists in using the K components solution to initialize the $K+1$ components mixture. Although interesting, their full process requires several GMM, with a varying number of components, leading to expensive computational time. By using this recursive initialization strategy, Baudry and Celeux also acted on the second burden of the EM algorithm: the choice of the number of components. Selecting the optimal number of components is highly dependent on which finite collection of models we consider and which selection criterion is used.

To select the best model in a finite collection, the well known criteria based on maximum likelihood are the Akaike Information Criterion (AIC)[5], or the Bayesian Informative Criterion (BIC) [6].BIC has been proved to be adequate for selecting K, but it is an asymptotic criterion and requires to run the estimation for several given possible K.

To overcome the problems of these asymptotic criteria, non-asymptotic approaches have been proposed, as the slope heuristic criterion. It was introduced by Birgé and Massart [7], and Baudry et al. [11] proposed a framework to calibrate it. This criterion assumes that there exists an optimal constant, which, associated with the model dimension, provides an optimal penalty of the log-likelihood. By computing a regression with obtained log-likelihood values in the estimated models collection, the optimal constant is then obtained with the slope. Drawbacks of this method are the required linear behavior of the log-likelihood and a large enough finite set of estimated models.

Trying to solve both issues together is a recurrent objective, which led to original methods in the past decades. These methods perform estimation and selection of the model at the same time [12, 13, 14, 15, 16, 17] .

A recent idea, proposed in [12], combines the slope heuristic criterion for model selection [7] with a dynamical change of the number of components inside the EM algorithm. The aim was to avoid convergence towards local maxima at the boundary of the parameters space and a too restrictive initialization. They introduced a annihilation step which deletes components based on a data-depending threshold and iterate between it and the EM algorithm until all components proportions are above a chosen threshold. The final estimated model is saved in a collection and this process is repeated several times for different initial K. Form the estimated models collection they select the best model with the slope heuristic criterion. As described here, their introduced method is also based on estimation of a finite collection of models. Moreover they have to run several time the EM algorithm to estimate only one model of the collection. This leads to high computation time, and to the risk that the real model does not belong to the finite set of models.

In [13] and later [17] a minimum message length criterion [18, 19] is developed to penalise the cost function, originally based on the log-likelihood in the EM algorithm. With this introduced penalisation, clusters may be annihilated if they are non-informative.This step prevents the algorithm from approaching the boundary of parameter space, and acts as a model selection process. They also try to address the initialization problem by beginning their algorithm with a large number of components. Their complete algorithm has the particularity to not stop before reaching a minimal number of clusters and it forces parameter space exploration to obtain several models.

Another dynamical algorithm is the step-wise split-and-merge EM algorithm, based on the construction of split and merge criteria [14, 15]. The authors of [14] based their split and merge criteria on Kullback-Leibler divergence and correlation coefficient respectively, while in [15] they used the local Kullback-Leibler divergence to measure distance between a local density and model density of each component for both split and merge criteria. In [15] they free themselves from the choice of

a criteria threshold, and introduce an acceptance probability for freshly new computed parameters after the split-merge step, which avoids too frequent and unstable moves.

In [16], Yang et al. also considered a dynamical algorithm, where the number of components is estimated in a single-run EM algorithm at a not too-high computational time. Moreover, the framework remains close to the EM algorithm one, but still presents the problem of falling into local maximums.

These different variants of the EM algorithm require a high computational cost, the tuning of several parameters or cannot avoid undesirable local maxima involving for example superimposed clusters. In addition the temporal component to monitor the population distribution is absent of these procedures, and the epidemiological models presented above also cannot meet the criteria for estimating, monitoring and modelling population dynamics over time. As a consequence, these drawbacks prevent us from directly using these different methods to obtain correct approximations of population dynamic and to monitor them.

1.3 Contributions

In this paper, we propose a pipeline named spatio-temporal mixture process (STMP) to infer population distribution and to highlight 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.

With the proposed STMP, we combine reliable estimation and modeling of mixture models and temporal monitoring of these models. 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 to take into account a temporal dependency inside a mixture model. Finally, we will also propose an adaptation of the Robust EM algorithm in [16] to overcome the EM algorithm drawbacks that are model selection and efficient estimation. Even if this Robust EM was shown to be effective to tune parameters as the number of components in the mixture, and estimate the means and cluster covariances, it can output overlapping components. We will suggest changes to obtain a more flexible algorithm and avoid these overlapping components.

To finish designing our STMP, we will perform experiments on synthetic data and we will study the behaviour of our pipeline in different situations. We will then test our algorithm on a COVID19 data set from Paris area, showing the adequacy of a mixture model evolving over time and the consistency of the alert response to population dynamical changes.

2 Notations and reminders on mixture models

We assume for our future application in Section 4 and 5 that the population is modeled as a Gaussian mixture. As we will associate here our pipeline with Gaussian Mixture Models (GMMs), we first recall the GMM definition and then the classical methods introduced in the literature to estimate GMM parameters. In particular, we focus on the question of estimating the number of clusters and the mixture in a single-run algorithm. These methods are the basic elements on which we build our pipeline STMP described in Section 3.1.

2.1 Summary on Gaussian mixture models

Let us 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, then the observations are assumed to be sampled from the following probability distribution:

$$\sum_{k=1}^{K} \pi_k \mathcal{N}_d(\cdot|\mu_k\, \Sigma_k)\,, \tag{1}$$

where $(\pi_k)_{1\leq k\leq K}$ is the mixing proportion sequence and satisfies $0 < \pi_k < 1$, $\forall\ k = 1, \ldots, K$ and $\sum_{k=1}^{K} \pi_k = 1$. The parameter is given by $\theta = (\pi_1, \ldots, \pi_K, \mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K)$ with means $(\mu_k)_{k=1,\ldots,K} \in \mathbb{R}^{d\times K}$ and covariance matrices $(\Sigma_k)_{k=1,\ldots,K} \in \mathcal{S}^{++}_{d\times d}(\mathbb{R})$, the set of symmetric positive definite matrices in d dimensions.

A GMM can be written as a hierarchical model. Let us introduce latent variables $(\mathbf{z}_1, \ldots, \mathbf{z}_n)$ such that $\mathbf{z}_i = \{0, 1\}^K$, $z_i^k = 1$ if data x_i belongs to cluster k, 0 otherwise. Then the complete model writes :

$$\left\{ \begin{array}{lcl} \mathbf{z}_i & \sim & \mathcal{M}ult(1, \pi_1, \ldots, \pi_k)\,, \\ x_i|z_i^k = 1 & \sim & \mathcal{N}_d(\mu_k, \Sigma_k)\,. \end{array} \right. \tag{2}$$

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

In the following subsection, we recall the Expectation-Maximization algorithm and its variants to solve both above mentioned issues. We also highlight their drawbacks which prevent us from using them as is in our STMP.

2.2 Estimation of GMM parameters with EM-like algorithms

The most popular algorithm to estimate a GMM is the Expectation-Maximization (EM) algorithm [8] 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. The observed likelihood writes for GMMs on a set of observations $\mathbf{x}$:

$$p_\theta(\mathbf{x}) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k \mathcal{N}_d(x_i|\mu_k\, \Sigma_k) \right]\,. \tag{3}$$

The EM algorithm alternates between the two following steps (until convergence criterion is met). Suppose we are at the p-th iteration of the algorithm, we have:

- 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

$$p_\theta(z_i^k = 1|x_i) = \frac{\pi_k \mathcal{N}_d(x_i|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}_d(x_i|\mu_j, \Sigma_j)} = \tau_i^k(\theta)\,. \tag{4}$$

- M-step : Update the parameter estimates:

$$\hat{\pi}^p_{k,MLE} = \frac{1}{n} \sum_{i=1}^{n} \tau_i^k\,, \tag{5}$$

$$\hat{\mu}_{k,MLE}^{p} = \frac{\sum_{i=1}^{n} \tau_i^k x_i}{\sum_{i=1}^{n} \tau_i^k}\,, \tag{6}$$

$$\hat{\Sigma}_{k,MLE}^{p} = \frac{\sum_{i=1}^{n} \tau_i^k (x_i - \mu_i)^\top (x_i - \mu_i)}{\sum_{i=1}^{n} \tau_i^k}\,. \tag{7}$$

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[13, 17, 15, 14, 16].

In the next section, we will detail a recent dynamical algorithm proposed by [16], which answers almost all issues and is the base of our proposition.

2.3 The Robust EM algorithm

As mentioned above, the unknown number of clusters in GMM is a main problem. The authors of [16] 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, leading to minimise this entropy.Starting from the complete log-likelihood, the objective function to maximize in the M-step with this entropy-based penalty is:

$$\mathcal{L}'(\theta, \mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_i^k log(\pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)) + \beta \sum_{i=1}^{n}\sum_{k=1}^{K} \pi_k log \pi_k, \text{with } \beta \geq 0\,. \tag{8}$$

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

$$\hat{\pi}_k^{(new)} = \hat{\pi}_{k,MLE} + \beta \hat{\pi}_k^{(old)} \left(\ln \hat{\pi}_k^{(old)} - \sum_{s=1}^{K} \hat{\pi}_s^{(old)} \ln \hat{\pi}_s^{(old)} \right) \tag{9}$$

with $\hat{\pi}_{k,MLE}$ given by (5), and $\hat{\pi}_k^{(old)}$ being the component weight estimate of previous iteration. Equations to estimate the means $\hat{\mu}_k$ and the covariance matrices $\hat{\Sigma}_k$ in Robust EM are still given by Eq.(6) and Eq.(7) with the new component weights from Eq.(9).

As we can see, new parameter β comes as a penalty weight in Eq.(8). It helps to control the competition between clusters. This parameter is enhanced at each iteration to increase the entropy weight in Eq.(9) if proportions at previous iteration were too close. And reciprocally β is reduced automatically to undercut the gap between the different proportions. 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 $\pi_k \leq \frac{1}{n}$. This is the annihilation part in their process.

Another dynamic is imposed to β which is fixed to zero when the cluster number K is stable, i.e not decreasing for a long period. This is important to not obtain oscillating parameters, and so to reach a maximum. From their implementation and experiments, they fixed this time limit to $p_{min} = 60$ iterations, without any attempt to adapt it to different use cases.

This algorithm is 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. Starting with higher number of mixture components than the true value is a solution to the initialization problem,

allowing to escape local maxima in some situations where components are very heterogeneously distributed in space.

Although efficient, entropy-based penalisation [16] or equivalently Dirichlet-type prior [13] do not prevent from having several components with similar parameters, meaning that two cluster may be superimposed.

In the Robust EM algorithm ([16]), competition and instability of component proportions do not avoid to finish in a wrong 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 by [16] in the next section. We also present a temporal process which combined with estimation algorithms will provide efficient detection of population dynamical changes.

3 Methods: combine a spatio-temporal mixture process and estimation algorithms

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

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 $t = 1...T$. At each time step, the data vector is $\mathbf{X}^{(t)} = (\mathbf{X}_1^{(t)}, \ldots, \mathbf{X}_{n_t}^{(t)})$ with $\mathbf{X}_i^{(t)} \in \mathbb{R}^d$. We assume that these data are 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 set $\mathbf{X}^{(t)}$, 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 $\mathbf{X}^{(t)}$.

However, as $M^{(t-1)}$ depends on the data set at time $t-1$, it suffers from the estimation variability, which means that the true model is likely to be close but not necessarily exactly this one. 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 $\theta^{(t-1)}$. Behind this adaptation of $M^{(t-1)}$, we indirectly keep track of the estimated model at previous time, so we obtain a temporal chain from $M^{(0)}$ to $M^{(T)}$ through computation of $M^{(t)} = M'$. 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 from the only data set $\mathbf{X}^{(t)}$. 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 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, named likelihood ratio or also known as 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 far 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 away from one.

Finally, we define the ratio as follows:

$$r_t(M', M^a) = \frac{p_{\theta^a}(\mathbf{X}^{(t)})}{p_{\theta'}(\mathbf{X}^{(t)})}\,. \tag{10}$$

In order to accept or reject the alternative model at time t, we define a threshold τ such that if $r_t(M', M^a) > \tau$, 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 4.2. In particular, this empirical study allows us to set the threshold τ and highlight its properties in particular its independence w.r.t the sample size.

With all these elements in hand, our space-time complete pipeline, named Space-Time Mixture Model (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.(10),
3. If $r_t(M', M^a) > \tau$, raise an alert and set $M^{(t)} = M^a$. Else set $M^{(t)} = M'$.

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[16]. 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 ($\|\mu^{(p)} - \mu^{(p-1)}\| < threshold$) 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.

To avoid this local maximum problem, we propose slight modifications of the Robust EM algorithm, by incorporating a verification step of superimposed clusters. We consider that two clusters i and j are superimposed if

$$\|\mu_i - \mu_j\|_2 + \|\Sigma_i - \Sigma_j\|_F < \epsilon \tag{11}$$

for some small $\epsilon > 0$. Note that requiring equality in Eq. (11) is numerically too strong and would barely happen.

We check Condition (11) 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} (line 2). At that point in the algorithm, if we set $\beta = 0$ too early, it slows down the competition between clusters, and may block components from disappearing. If there is 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 closeness condition Eq. (11) again. If Eq. (11) is still true for some clusters, we merge these clusters. The weight of the fused cluster 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 the likelihood. This makes the algorithm jump to another configuration with 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: incorporating former parameters

We name Constrained EM (C-EM) a slight variation of original EM algorithm [8] 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. 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 (5) and (6). We then add a third step in the estimation algorithm to obtain $\hat{\pi}^c$ and $\hat{\mu}^c$. Covariance matrices $\hat{\mathbf{\Sigma}}^p$ are estimated with Eq. (7), and correspond to $\hat{\mathbf{\Sigma}}^c$, without any direct constraint but we will see right after a tested condition at the end of the C-EM algorithm.

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. 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 $[\pi_k^0 \pm 0.1]$. It will represent π_k^0 more or less 10% of all proportions. We also avoid proportions to become null to avoid an 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:

$$\begin{cases} \hat{\pi}_k^c & = min(max(\pi_k^0 - 0.1, \hat{\pi}_k^p), \pi_k^0 + 0.1)\,, \\ \hat{\mu}_k^c & = \mathcal{P}_{rect(\mu_k^0, \Sigma_k^0)}(\hat{\mu}_k^p)\,. \end{cases} \tag{12}$$

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 $\mathbf{X}^{(t)}$.

In the use case presented later, we use Gaussian distribution for each cluster. This will be detailed and motivated in Section 5. In this setting the constrains are the one presented above on the mixture probabilities and the means of the Gaussian distributions (Eq.(12)).

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 see when the estimated candidate model does not correspond to 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.(10)).

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 C-EM algorithm, covariance matrices are freely estimated, but they can evolve far away from initial covariances matrices Σ_k^0, so 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 [20] 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.

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).

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 next applications we will only consider geographical data, in $\mathbf{R}^2$, 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. (3).

Remember 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. We first present all the experiments that are tested on our pipeline.

To demonstrate and validate our complete pipeline, using both our modified Robust EM and our C-EM, we then study the estimated likelihood ratio for different behaviors of the population distribution (characterized by the experiments), and the resulting performances of the pipeline. By studying these performances we can fix a threshold conditioning the raise of the alert in all situations.

Finally, in Subsection 4.3 we focus on the validation of STMP,given by Algorithm 1. We evaluate the alert performances and computation times for all experiments.

In Appendix A.2.1 we present the estimation results of the Gaussian mixture models inside our pipeline. And in Appendix A.2.2 we include experiments on the number of points n in the data

sample, and how it affects each step of STMP.

4.1 Description of experimental setups

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 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 if not 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 write $K^{(0)}_{true}$ 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 2. 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 (from Table 2). This enables us to analyze the behavior of STMP and likelihood ratio across runs and evolution cases.

4.2 Estimation of the alert threshold

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 correctly the data, 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.

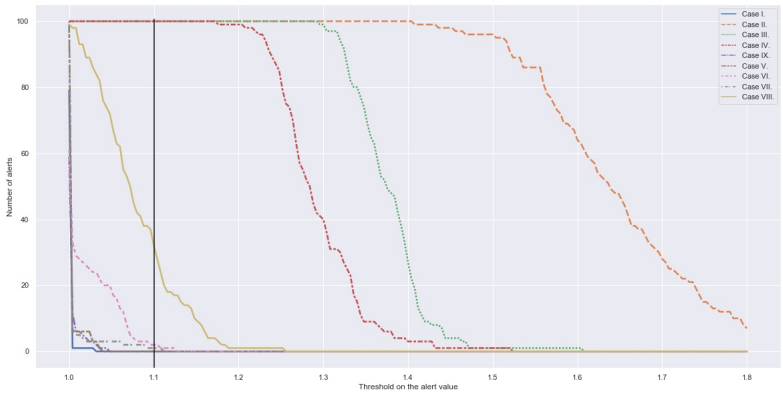


Figure 1: For each Case is presented the number of raised alerts over $S = 100$ runs according to the considered threshold τ. The filled black vertical line is the final selected threshold.

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 we said previoulsy, we run several times our STMP on different sampled data for each case distribution. This number of runs S is here fixed to $S = 100$. For each run of STMP on each case, we look at the computed likelihood ratio defined by Equation (10)between M' and M^a at $t = 1$. We then account for the number of raised alerts, depending on the value of the likelihood ratio threshold. Figure 1 represents this behavior, with one curve by case explained in Subsection 4.1.

For Case I., Case V. and Case VI., the population distribution is the same at $t = 0$ and $t = 1$, but 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. Therefore, it requires to take into account this uncertainty when selecting our threshold alert. From Figure 1, 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 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.

We observe that for population moves from Setup F. to Setup M. (Case VII.) or from Setup M. to Setup C. (Case IX.), corresponding to slight movements, alerts are not raised for a threshold over one. The likelihood ratio values stay relatively close to one because the model M' can adapt to data $\mathbf{X}^{(1)}$. This provides us an intuition on the level of variations that our model can detect.

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, to make a compromise over all our experiments and obtain a good performance 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.

4.3 Performances of the STMP on synthetic data

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 following alert rates, that we can retrieve in the Figure 1.

For Cases I. and V. we obtain 99 and 98 true negative alerts respectively, and Case VI. raises eight false positive alerts. 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 IX. .

In contrary, the bigger movement in Case VIII. leads to a true positive number of 32%. 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 are performed with an average execution time of 1.70s. From Table 3 we recover average execution time by case type. Fast execution was also a criterion leading the construction of our method, and satisfying for our future applications.

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 COVID19 in Paris, France.

5.1 Presentation of the data set

For this use case, we included all positive diagnosed patients to COVID19 living in Paris city. These data were collected in AP-HP (Assistance Publique des Hopitaux de Paris) which 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). For each patient we have two pieces of information: the week he/she was diagnosed positive, and his/her place of residence. Patients were aggregated at the scale of an IRIS area, which is the smallest geographical division in France with 2000 inhabitants in average per area.

New positive diagnosed patients were aggregated by IRIS and by week over 11 weeks (from weeks 9 to 19 of the year 2020). We therefore use a week as the 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 8 informs us that the number of cases per week is not homogeneous, as in first weeks, few cases living in Paris were detected positive.

5.2 Performances of STMP

The aim here is to underline presence or absence of temporal constancy in data, which suggests 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 (weeks 9 to 12) (see weekly reports of Public Health Institution[21] Page.7 Figure 8.).

We still use a fixed alert threshold of $\tau = 1.1$ in STMP, estimated by previous experiments. Our STMP reveals that a GMM, estimated by our modified Robust EM on week 13 with $\hat{K}^{(13)} = 7$, was accepted on weeks 14 and 15. As a reminder, week 13 represents the peak of the pandemic, in terms of new positive cases. This means that C-EM executed across weeks 14 and 15 fits very well the new data set each week with a source model estimated on week 13. Even if the number of cases changes over time, STMP is able to detect a constant underlying distribution. It is consistent with the patients distribution on weeks 13, 14 and 15 as we can see on Figures 5, 6, 7 and 8.

On 16^{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 $\theta^{(16)}$ on 16^{th} week are newly estimated, evolving too far from $\theta^{(15)}$, parameters on 15^{th} week. It can be interpreted with the decrease of new positive cases such as the disappearance of large clusters from previous weeks and the detection of many smaller clusters, corroborated by the Figures 6b and 8b. In addition, from Table 8, the number of cases is starting to get weak again.

On the following weeks (weeks 17,18 and 19), the number of cases is still decreasing, and 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 1, 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 week 13 the likelihood ratio value is smaller that our defined threshold while the temporal-dependent model M' is rejected. This is due to high changing covariance matrices during the C-EM stage. The model M' fits the new dataset by excessively moving the covariance parameters herited from $M^{(12)}$. Finally, the ratio values on next weeks confirm that a GMM with 7 clusters is adapted to the data distribution on these weeks.

An important and interesting result stemming from this analysis is the highlight of small clusters closed to Paris periphery: on weeks 13 to 15, which have a not so high number of clusters, we observe relatively small clusters on the edges of Paris area. These small clusters are even more striking on weeks 12 or 16 where the number of clusters is large. These areas are low-income neighbourhoods which are known to favour COVID-19 epidemics.

We can observe absence of likelihood ratio value on the last week. We cannot compute this ratio, due to the "empty class phenomenon". The model M' tries to remove a component which leads to an early stop of the estimation process. This triggers the inevitable choice of the alternative model and raises an alert.

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

Table 1: Results of our process on positive diagnosed people in AP-HP hospitals with a time step being a week

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	1.390	M^a
11	**5**	2	1.471	M^a
12	**12**	5	1.214	M^a
13	**7**	12	1.079	M^a
14	5	**7**	0.976	M'
15	7	**7**	1.028	M'
16	**12**	7	1.362	M^a
17	**6**	12	1.254	M^a
18	**9**	6	1.760	M^a
19	**3**	9	computationally invalid	M^a

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 raise 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 [22] 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.

Declaration of Competing Interest

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.

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] 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.

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

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

[8] 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.

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

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

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

[22] 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 Appendix

A.1 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 (4)
$p \leftarrow 1$
Compute μ_k^p with (6)
while $\max_{1 \leq k \leq K^p} \|\mu_k^{p+1} - \mu_k^p\| > \varepsilon$ *or Eq.* (11) *is verified for some clusters* **do**
  Compute π_k^p by (9)
  $\pi_{(1)}^{EM} \leftarrow \underset{1 \leq k \leq K^p}{max} \pi_k^{p,EM}$, $\pi_{(1)}^{(old)} \leftarrow \underset{1 \leq k \leq K^p}{max} \pi_k^{(old)}$
  $E \leftarrow \sum_{k=1}^{K^p} \pi_k^{(old)} ln \pi_k^{(old)}$
  $\beta^p \leftarrow \min \left\{ \frac{\sum_{k=1}^{K^{p-1}} \exp\left(-\eta n \left| \pi_k^p - \pi_k^{(old)} \right| \right)}{K^{p-1}}, \frac{\left(1 - \pi_{(1)}^{EM}\right)}{\left(-\pi_{(1)}^{(old)} E\right)} \right\}$
  Update class number K^{p-1} to K^p by deleting classes with $\pi_k^p \leq 1/n$, 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**
    if *no superimposed clusters (Eq.*(11) *false)* **then**
      $\beta^p = 0$
    else if *superimposed clusters and* $p_{component} < 200$ **then** `/* give more time to the algorithm to converge */`
      $p_{min} \leftarrow p_{min} + 50$
    else merge superimposed clusters
      adjust π^p, μ^p, Σ^p and τ^{p-1}
    end
  end
  Compute Σ_k^p with (7) 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 (4)
  Compute μ_k^{p+1} with (6)
  $p \leftarrow p + 1$
  $p_{component} \leftarrow p_{component} + 1$
end

A.2 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 2: Different cases of data distributions changes from one time point to the next one (here only considering $t = 0$ and $t = 1$). Note that $\Sigma_0 = \begin{pmatrix} 1. & 0. \\ 0. & 1.5 \end{pmatrix}$

Described Experiment	Average computation time over $S = 100$ runs (std)
Case I.	1.76s (0.61)
Case II.	1.73s (0.51)
Case III.	1.40s (0.31)
Case IV.	1.58s (0.36)
Case V.	1.63s (0.20)
Case VI.	1.85s (0.49)
Case VII.	1.79s (0.29)
Case VIII.	2.04s (0.64)
Case IX.	1.51s (0.20)

Table 3: Average (and standard deviation) computation time of different experiments. Each execution of our method was performed after sampling $n_0 = n_1 = 400$ points.

A.2.1 Results on the estimation of the number of components K inside our pipeline

We present here results on the estimation of GMM parameters with the modified Robust EM algorithm at $t = 0$ and $t = 1$ in our STMP experiments. All experimental frameworks described in Subsection 4.1 are tested, all with $n_0 = n_1 = n = 400$ points, the biggest number of samples we considered.

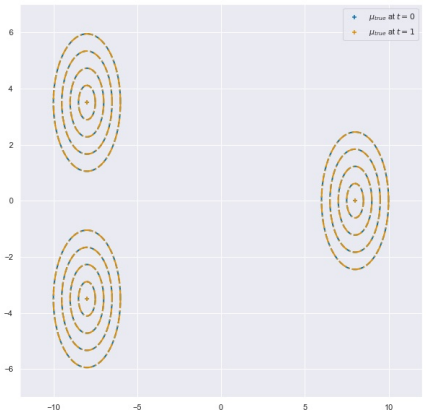
For each run of each experiment, we check here if the number of estimated clusters by our modified Robust EM at $t = 0$ or $t = 1$ is correct. We report the correctly estimated K rate in Table 4.

We use the modified Robust EM twice in our STMP: to estimate the initial model at $t = 0$ and then the alternative model at $t = 1$. We have $\hat{K}^{(0)}$ and $\hat{K}^a$ components estimated for $M^{(0)}$ and M^a respectively. We decide that our STMP correctly estimates K over time if and only if $\hat{K}^{(0)} = K^{(0)}_{true}$ and $\hat{K}^a = K^{(1)}_{true}$. As an example, in Table 4, for Case I. (same distribution at $t = 0$ and $t = 1$), over $S = 100$ runs, 98 runs of our STMP give both correct estimated $\hat{K}^{(0)}$ at $t = 0$ and $\hat{K}^a$ at $t = 1$. In brief, the correctly estimated K number is given by the intersection of correctly estimated $\hat{K}^{(0)}$ and $\hat{K}^a$.

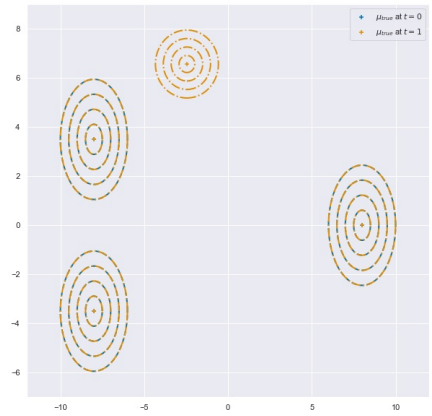
The Cases I. to IV. give good K estimates, explained by the correct separation of the clusters as seen in Figure 2. On experiments with configurations bringing closer two clusters (Cases V. to IX.), we obtain high rate (over 90%) for static and well-enough separated clusters (Setup F., Setup M.). This score is also high for displacement from Setup F. to Setup M. .

But this score decreases when we consider moving clusters which are getting too close. In Setup C., it becomes harder for our modified Robust EM to differentiate the two merging clusters, which lead to worst scores. The global score of STMP executions involving at least one Setup C. distribution is affected by this, the correct proportions are not bigger than 57%. If we look at the estimates $\hat{K}^{(0)}$ and $\hat{K}^a$ in Table 4, the Modified Robust EM algorithm often estimates two classes with samples from Setup C. distribution. This behavior happens at least 30 over 100 times for each experiment. But this incorrect estimation leads 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 4, which confirms the interpretability of the results.

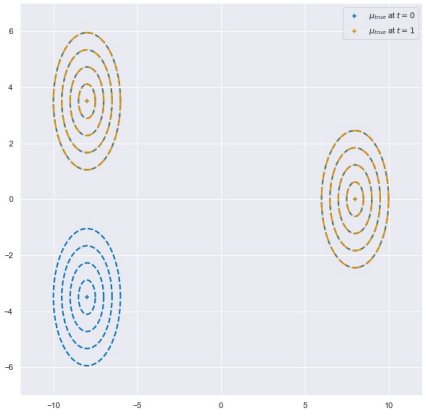
Thereafter, we can compute the estimation errors for means and covariances matrices on experiments with correctly estimated number of components K (see Table 5). This allows us to confirm that these estimated Gaussian mixtures are correctly estimated by the modified Robust EM inside our pipeline STMP. We also notice a poorer average estimate of GMM parameters for data sets from



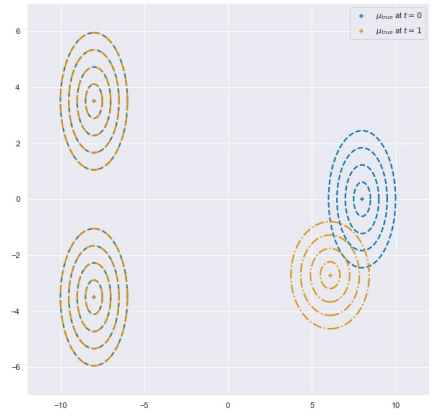
(a) Case I.



(b) Case II.

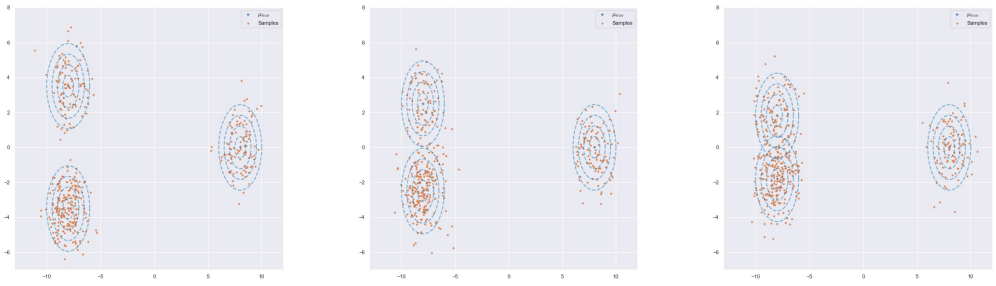


(c) Case III.



(d) Case IV.

Figure 2: Description of Gaussian mixture distributions for Cases I. to IV. (from Table 2). Blue centers and covariance ellipsis 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.



(a) Setup F. (Far) (b) Setup M. (Moderate) (c) Setup C. (Close)

Figure 3: Gaussian mixture distributions for Setups F., M. and C. involved in Cases presented in Table 2 with an example of sampled data sets. Blue crosses correspond to μ_k and ellipsis to covariance matrices Σ_k. Orange points are samples.

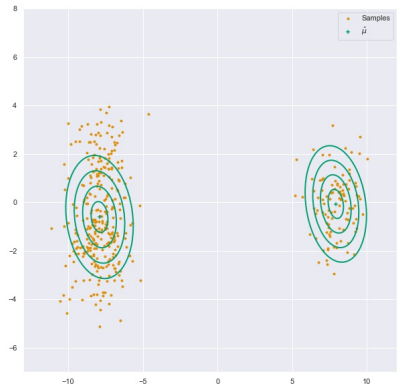


Figure 4: An estimated GMM with $\hat{K} = 2 \neq K_{true} = 3$ for a Setup C. distribution. The centers and covariances are represented in green. Orange points are samples.

Experiment	Proportion of correctly estimated number of components (% for $\hat{K}^{(0)} = 2$, $\hat{K}^a = 2$)
Case I.	**98** %
Case II.	**96** %
Case III.	**100** %
Case IV.	**100** %
Case V.	**94** %
Case VI.	40 % (**30** %,**34** %)
Case VII.	**100** %
Case VIII.	57 % (0 %,**34** %)
Case IX.	57 % (0 %,**36** %)

Table 4: Proportion of right 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 2.

Setup C. As said previously, this parametrization implies that two clusters are mixed up. In Table 5 we clearly see a slight higher average euclidean distance between the true means and the estimated ones for Setup C. models. For covariance matrices errors, simply estimated with Frobenius norm, the average errors are less contrasted, but we observe the highest error for estimation of M^a in Case II. (the GMM with an emerging cluster).

	$M^{(0)}$		M^a	
Case	$\hat{\mu}$	$\hat{\Sigma}$	$\hat{\mu}$	$\hat{\Sigma}$
Case I.	2.0 (1.0)	15.0 (7.0)	2.0 (1.0)	14.0 (7.0)
Case II.	1.0 (1.0)	14.0 (7.0)	2.0 (1.0)	33.0 (31.0)
Case III.	2.0 (1.0)	14.0 (7.0)	1.0 (1.0)	11.0 (5.0)
Case IV.	2.0 (1.0)	14.0 (7.0)	2.0 (1.0)	20.0 (13.0)
Case V.	1.73 (1.24)	16.3 (11.58)	1.68 (0.97)	15.82 (8.19)
Case VI.	2.94 (3.71)	23.80 (25.55)	3.47 (4.35)	24.82 (28.84)
Case VII.	1.49 (0.92)	14.80 (6.63)	1.61 (0.95)	15.69 (7.77)
Case VIII.	1.61 (0.82)	15.2 (8.67)	3.05 (3.86)	24.24 (27.15)
Case IX.	1.68 (0.97)	16.0 (8.48)	3.02 (3.66)	25.39 (25.86)

Table 5: Mean (standard deviation) relative errors (expressed as a percentage) for the estimated means and covariance matrices within each case, over all runs (for each case) having correctly estimated $\hat{K}$ inside STMP. The euclidean norm is used for means, and the Frobenius norm for covariances.

A.2.2 Effects of the data set size on estimation of Gaussian mixtures and on STMP

In previous explained experiments on synthetic data, we fixed the data set size $n = 400$. In this part we study effect of a varying $n \in \{100, 200, 400\}$ in experiments Cases I. to IV. described previously. With the same true distributions as in Figure 2, we perform $S = 100$ runs of our process with data samples of size $n = 200$ and $n = 100$ at each time step.

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.	98 %	88 %	65 %
Case II.	96 %	87 %	51 %
Case III.	100%	94 %	63 %
Case IV.	100 %	89 %	62 %

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 $\hat{K}^{'} = K^{(0)}$.

Experiment	Data sets size $n = 400$	Data sets size $n = 200$	Data sets size $n = 100$
Case I.	**1**	21	56
Case II.	**100**	**100**	**100**
Case III.	**100**	**100**	**100**
Case IV.	**100**	**100**	**100**

Table 7: Number of alerts raised by our STMP for each experiment ($S = 100$ runs) on data sets of n points.

As expected, decreasing the number of samples decreases the proportion of good estimated $\hat{K}$ and inherently the quality of estimation of parameters θ (Table 6). For $n = 200$ points, the modified Robust EM algorithm still gives high rates, between 87% and 94%, allowing to be confident in the estimates. A data set of 100 points begins to be very limited to properly estimate a GMM: the best alert rate is 65% and the worst is 51%. Therefore, we must be aware that accuracy of estimated GMM in our process quickly decreases with the data set size.

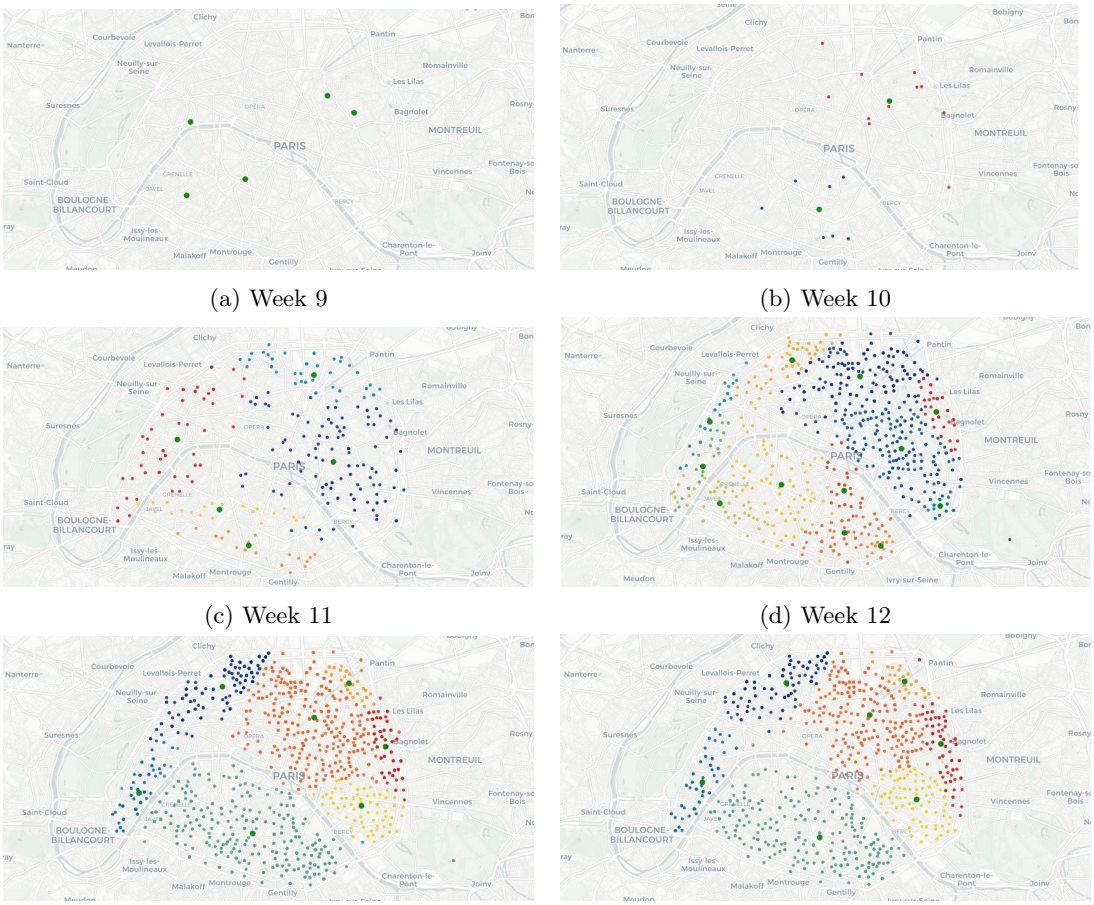
We can now look at the performance of our pipeline, depending on the data set size. For the Case I., we want to obtain zero alert. As we saw in Subsection 4.3, we almost reach it for data sets of size $n = 400$. For data sets of size $n = 200$ we have 21 false positive alerts, and for $n = 100$ we have 56 false positive alerts. For Cases II. to IV. the proportion of success is 100% for all n values (Table 7).

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

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 8: Distribution of positive diagnosed people to COVID19 over weeks.

A.3 Results on the COVID19 data set of Section 5



(a) Week 9

(b) Week 10

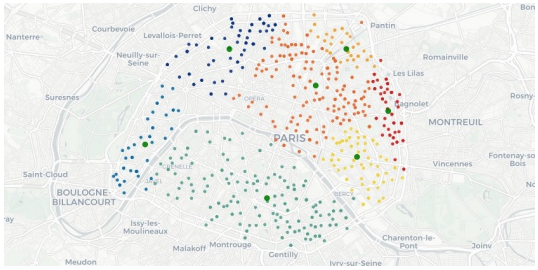
(c) Week 11

(d) Week 12

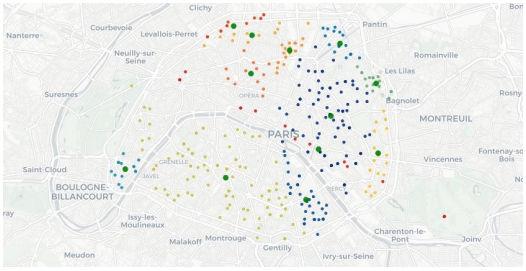
(e) Week 13

(f) Week 14

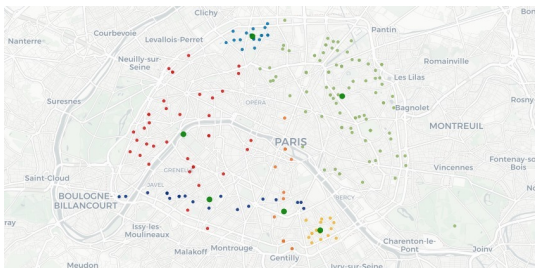
Figure 5: Estimated GMM parameters on Covid19 data set per week (weeks 9 to 14). Green dots are centers of the clusters.



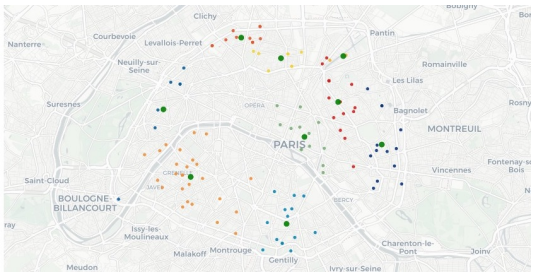
(a) Week 15



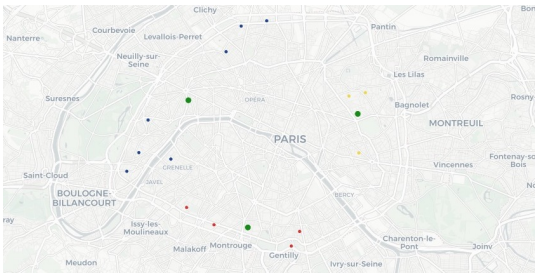
(b) Week 16



(c) Week 17

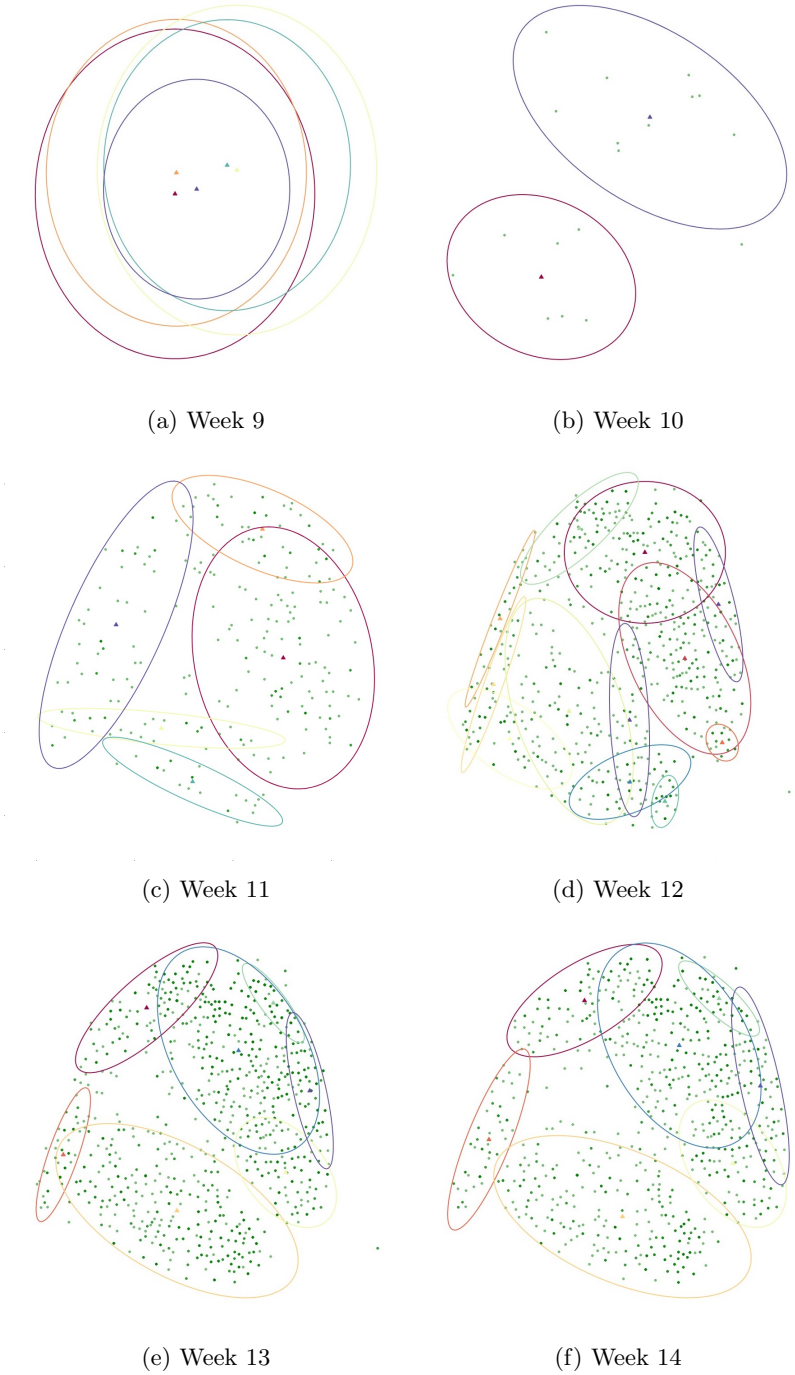


(d) Week 18



(e) Week 19

Figure 6: Estimated GMM parameters on Covid19 data set per week (weeks 15 to 19). Green dots are centers of the clusters.



(a) Week 9

(b) Week 10

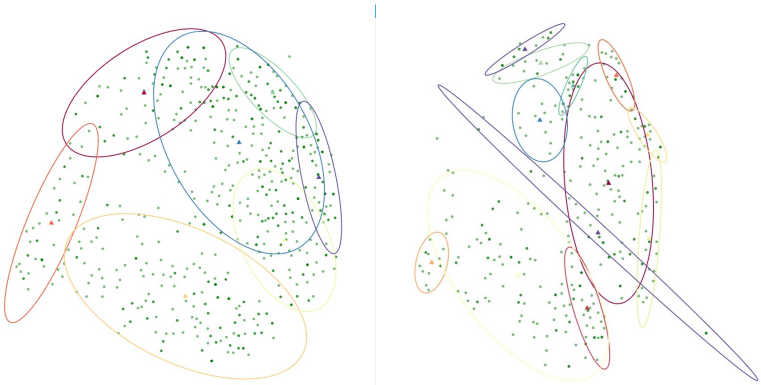
(c) Week 11

(d) Week 12

(e) Week 13

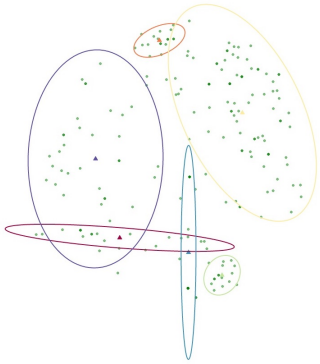
(f) Week 14

Figure 7: Estimated GMM parameters on Covid19 data set per week (weeks 9 to 14). Triangles and ellipses are stimated parameters. Green dots are patients.

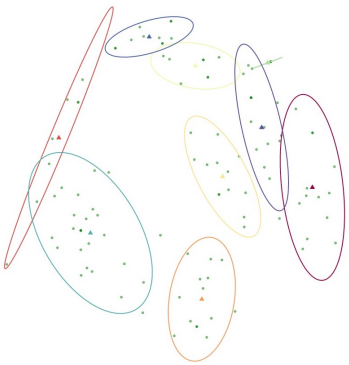


(a) Week 15

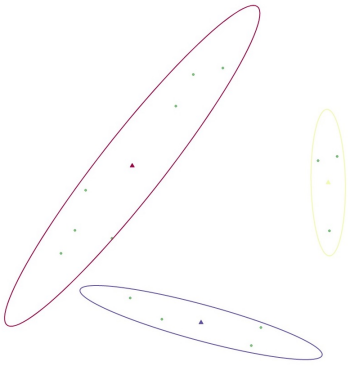
(b) Week 16



(c) Week 17



(d) Week 18



(e) Week 19

Figure 8: Estimated GMM parameters on Covid19 data set per week (weeks 15 to 19). Triangles and ellipses are stimated parameters. Green dots are patients.