Supplementary Materials: Deterministic Approximate EM algorithm;
Application to the Riemann approximation EM and the tempered EM

Thomas Lartigue, **Stanley Durrleman** and **Stéphanie Allassonnière**

1. Introduction

In these supplementary materials, we give a more extensive experimental study of the tempered EM algorithm (tmp-EM). In Section 2, we perform an in depth experimental study of the behaviour and performances of tmp-EM to demonstrate that it solves the issues raised about the EM. In particular, we illustrate on synthetic data, in the GMM case, that tmp-EM consistently reaches better values of the likelihood than the unmodified EM, in addition to better estimating the GMM parameters. We demonstrate that, as intended, tmp-EM is able to escape bad initialisations, unlike EM, and that more diverse configurations are explored during the procedure before reaching convergence. We confirm these observation on real data from the scikit learn library [1]. Finally, in Section 3, we test the tmp-EM within a more complex pipeline: the Independent Factor Analysis model [2] with a hidden GMM. We illustrate that, with tmp-EM, the identified sources are cleaner, more stable looking, and closer to the real ones when those are known.

2. Experiments on tmp-EM with Mixtures of Gaussian

In this section, we present more detailed experiments analysing the tempered EM and comparing it to the regular EM. As in the main paper, we focus on likelihood maximisation within the Gaussian Mixture Model. From the optimisation point of view, we demonstrate that tmp-EM does not fall in the first local maximum like EM does but instead consistently finds better one. From the machine learning point of view, we illustrate how tmp-EM is able to better identify the real GMM parameters even when they are ambiguous and when the initialisation is voluntarily tricky.

The only constraints on the temperature profile is that $T_n \longrightarrow 1$ and $T_n > 0$. We use two different temperature profiles. First, a decreasing exponential: $T_n = 1 + (T_0 - 1)\, \exp(-r.n)$. We call it the "simple" profile, it works most of the time. Second, we examine the capabilities of a profile with oscillations in addition to the main decreasing trend. These oscillations are meant to momentarily increase the convergence speed to "lock-in" some of the most obviously good decisions of the algorithm, before re-increasing the temperature and continuing the exploration on the other, more ambiguous parameters. Those two regimes are alternated in succession with gradually smaller oscillations, resulting in a multi-scale procedure that "locks-in" gradually harder decisions. The formula is taken from [3]: $T_n = th(\frac{n}{2r}) + (T_0 - b\frac{2\sqrt{2}}{3\pi})\, a^{n/r} + b\, sinc(\frac{3\pi}{4} + \frac{n}{r})$. The profile used, as well as the values of the hyper-parameters are specified for each experiment. The hyper parameters are chosen by grid-search.

For the sake of comparison, the following Experiment 1 and 2 are similar to the experiments of [3] on the tmp-SAEM.

2.1. Experiment 1: 6 clusters

We start by demonstrating the superior performance of the tempered EM algorithm on an example mixture of $K = 6$ gaussians in dimension $p = 2$. The real parameters can be visualised on Figure S1, where the real centroids are represented by black crosses and confidence ellipses help visualise the real covariance matrices. In addition, 500 points were simulated in order to illustrate, among other things, the weights of each class. To quantify the ability of each EM method to increase the likelihood and recover the true parameters,

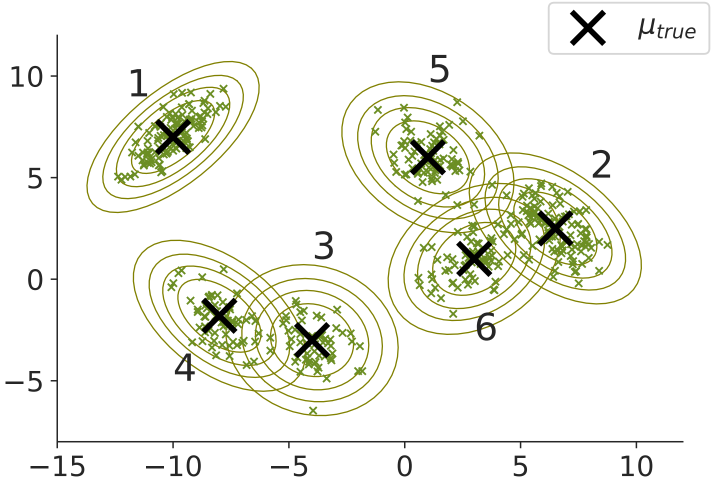


Figure S1. 500 sample points from a Mixture of Gaussians with 6 classes. The true centroid of each Gaussian are depicted by black crosses, and their true covariance matrices are represented by the confidence ellipses of level 0.8, 0.99 and 0.999 around the centre.

we generate from this model 20 different datasets with $n = 500$ observations. For each of these datasets, we make 200 EM runs, all of them starting from a different random initialisation. To initialise the mixture parameters, we select uniformly 6 data points to act as centroids. In each run, EM and tmp-EM start with the same initialisation. The number K of clusters is known by the algorithms. For this experiment, the simple tempering profile is used with parameters $T_0 = 50$ and $r = 2$.

2.1.1. Illustrative

First, we observe on the left of Figure S2, one example of the final states of the EM algorithm. The observations can be seen in green, the initial centroids are represented by blue crosses, and the parameters $\{\hat{\mu}_k\}_{k=1}^K$ and $\left\{\widehat{\Sigma}_k\right\}_{k=1}^K$ estimated by the EM are represented in orange. In this EM run, one of the estimated clusters became degenerated and, as counterpart, two different real clusters were fused as one by the method. On the right of Figure S2, we observe the final state of the tmp-EM on the same dataset, from the same initialisation. This time all the clusters were properly identified.

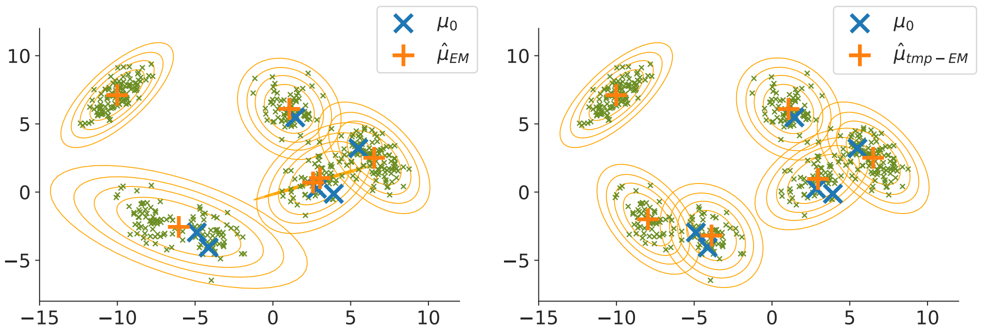


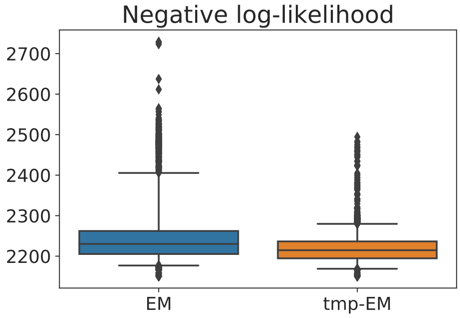
Figure S2. EM and tmp-EM final states on the same simulation with the same initialisation. tmp-EM positioned correctly the estimated centroides, whereas the regular EM made no distinction between the two bottom classes and ended up with a degenerate class instead.

2.1.2. Quantitative

To demonstrate the improvements made by tempering, we present aggregated quantitative results over all the simulated datasets and random initialisations.

Likelihood maximisation

EM and tmp-EM are optimisation methods whose target function is the likelihood of the estimated mixture parameters. We represent on Figure S3 the empirical distribution of the negative log-likelihoods reached at the end of the two methods, EM in blue, tmp-EM in orange. On those boxplots, the coloured "box" at the centre contains 50% of the distribution, hence it is delimited by the 0.25 and 0.75 quantiles. The median of the distribution is represented by an horizontal black line inside the box. The space between the whiskers on the other end, contain 90% of the distribution, its limits are the 0.05 and 0.95 quantiles. The table provides the numeric values of these statistics.



	EM	tmp-EM
median	2230	**2215**
mean	2247	**2219**
std	70	**35**
Q5	2177	**2169**
Q25	2205	**2195**
Q75	2262	**2236**
Q95	2405	**2280**

Figure S3. Empirical distribution of the negative log-likelihood reached by the EM algorithms. EM is blue and tmp-EM in orange. The boxplot allow us to identify the quantiles 0.05, 0.25, 0.5, 0.75 and 0.95 of each distribution, as well as the outliers. Their numeric values can be found in the table, the better ones being in **bold**. tmp-EM is better overall.

We note that the negative log-likelihood reached by tmp-EM is lower on average (higher likelihood) than what EM obtains. Moreover, tmp-EM also has a lower variance, its standard deviation being approximately half of the std of EM. More generally, we observe that the distribution of the final loss of tmp-EM is both shifted towards the lower values and less variable. In particular, each of the followed quantiles are lower for tmp-EM, and both the difference Q95-Q5 (space between whiskers) and Q75-Q25 (size of the box) are lower for tmp-EM. This illustrates that it obtains better, more consistent results on our synthetic example.

Parameter recovery

The EM algorithm is an optimisation procedure. Stricto sensu, the optimised metric - the likelihood - should be the only criterion for success. However, in the case of the Mixture of Gaussians, the underlying Machine Learning stakes are always very visible. Hence we dedicate time to assess the relative success parameter recovery of EM and tmp-EM.
The quality of parameter recovery is always dependent on the number of observation. The larger n, the more the likelihood will describe an actual ad-equation with the real parameters behind the simulation. Additionally, as n grows, the situation becomes less and less ambiguous, until all methods yield either the exact same, or at least very similar solutions, with all of them being fairly close to the truth. All of our simulation are done with $n = 500$ data points. Not a very large number, but since the lowest weight of our $K = 6$ classes is around 0.09, it is sufficient for all the classes to be guaranteed to contain several points. The three families of parameters in a GMM are the weights $\{\pi_k\}_{k=1}^{K}$, the averages (centroids positions) $\{\mu_k\}_{k=1}^{K}$ and the covariance matrices $\{\Sigma_k\}_{k=1}^{K}$ of the K classes. We evaluate the error made on μ with the relative different in squared norm 2:

$\frac{\|\hat{\mu}_k - \mu_k\|_2^2}{\|\mu_k\|_2^2}$. For Σ, we compute the KL divergence between the real matrices and the estimates $KL(\Sigma_k, \widehat{\Sigma}_k) = \frac{1}{2}\left(\ln \frac{|\Theta_k|}{|\widehat{\Theta}_k|} + tr(\Sigma_k \widehat{\Theta}_k) - p\right)$, with $\Theta := \Sigma^{-1}$ for all those matrices. Finally, the analysis on π is harder to interpret and less interesting, but reveals the same trend, with lower errors for the tempering.

The error on the averages μ_k is usually the most informative and easy to interpret metric, quantifying how well each methods position the class centres. Figure S4 and Table S1 represent the distribution of the relative error $\frac{\|\hat{\mu}_k - \mu_k\|_2^2}{\|\mu_k\|_2^2}$. The results of tmp-EM are much better with average and median errors often being orders of magnitude below the errors of EM, with similar or lower variance. The other quantiles of the tmp-EM distribution are also either equivalent to or order of magnitudes below the corresponding EM quantiles. The largest errors happen on Class 3 and 6, two of the ambiguous ones, but are always noticeably smaller and less variable with the tempering.

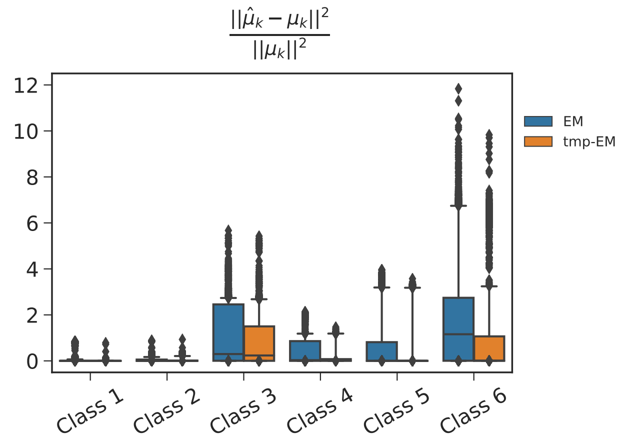


Figure S4. Empirical distribution of the relative error in squared norm 2 $\frac{\|\hat{\mu}_k - \mu_k\|_2^2}{\|\mu_k\|_2^2}$ between the real centroid positions in μ and the estimations by the EM algorithms.

Table S1. Quantiles and other statistics describing the empirical distribution of the relative error in squared norm 2 $\frac{\|\hat{\mu}_k - \mu_k\|_2^2}{\|\mu_k\|_2^2}$ between the real centroid positions in μ and the estimations by the EM algorithms. The error of tmp-EM is always closer to 0 with lower variance (with the exception of class 2 where the variance is similar).

Cl.		mean	std	Q5	Q25	Q50	Q75	Q95
1	EM	0.024	0.119	6.10^{-6}	6.10^{-5}	2.10^{-4}	0.002	0.065
	tmp-EM	**0.002**	**0.014**	**6.10^{-6}**	**4.10^{-5}**	**1.10^{-4}**	**4.10^{-4}**	**0.005**
2	EM	0.038	**0.066**	5.10^{-5}	2.10^{-4}	0.001	0.057	**0.169**
	tmp-EM	**0.032**	0.070	**5.10^{-5}**	**2.10^{-4}**	**5.10^{-4}**	**0.013**	0.210
3	EM	0.971	1.153	4.10^{-4}	0.004	0.297	2.467	2.736
	tmp-EM	**0.743**	**1.072**	**3.10^{-4}**	**0.003**	**0.235**	**1.500**	**2.681**
4	EM	0.310	0.487	7.10^{-5}	8.10^{-4}	0.031	0.859	**1.158**
	tmp-EM	**0.287**	**0.476**	**3.10^{-5}**	**5.10^{-4}**	**0.025**	**0.076**	1.188
5	EM	0.735	1.248	8.10^{-5}	5.10^{-4}	0.002	0.814	3.191
	tmp-EM	**0.432**	**1.054**	**6.10^{-5}**	**4.10^{-4}**	**7.10^{-4}**	**0.002**	**3.180**
6	EM	1.940	2.828	7.10^{-4}	0.005	1.158	2.743	6.744
	tmp-EM	**0.807**	**1.735**	**4.10^{-4}**	**0.002**	**0.010**	**1.066**	**3.243**

The KL divergences $KL(\Sigma_k, \widehat{\Sigma}_k)$ assess whether each the covariances Σ_k of each class are properly replicated. Note that since the computation of the KL divergence involves the matrix inverse $\widehat{\Theta}_k = \widehat{\Sigma}_k^{-1}$, the outliers cases where a class vanishes in an EM have to be removed: they correspond to pathological, non invertible matrices. Figure S5 and Table S2 describe the distribution of the KL divergence. The Figure is cropped and does not show some of the very rare, most upper outliers (less than 1%). Overall, the results are similar to what we get on μ: in terms of average KL and median KL, tmp-EM is better than EM, being either similar on some classes and much better on others. Its standard deviation is also lower - sometimes by one order of magnitude - on all classes except Class 4. The other quantiles are also overall better, with one exception on Q95 of class 4.

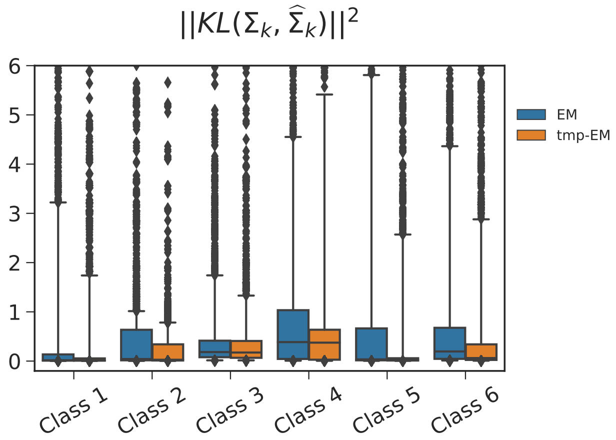


Figure S5. Empirical distribution of the KL divergence $KL(\Sigma_k, \widehat{\Sigma}_k)$ between each covariance matrix estimated by the EMs and the real covariance matrices Σ.

Table S2. Quantiles and other statistics describing the empirical distribution of the KL divergence $KL(\Sigma_k, \widehat{\Sigma}_k)$ between each covariance matrix estimated by the EMs and the real covariance matrices Σ. On every class but the 4th, the deviation of tmp-EM is closer to 0 with lower or similar variance.

Cl.		mean	std	Q5	Q25	Q50	Q75	Q95
1	EM	2.741	39.879	0.003	0.009	0.017	0.136	3.222
	tmp-EM	**0.845**	**8.683**	**0.003**	**0.008**	**0.013**	**0.055**	**1.745**
2	EM	0.852	**9.006**	0.004	0.015	0.042	0.636	1.015
	tmp-EM	**0.412**	9.072	0.004	**0.011**	**0.027**	**0.34**	**0.782**
3	EM	1.185	14.636	0.015	0.078	0.183	0.414	1.742
	tmp-EM	**0.648**	**4.435**	**0.014**	**0.066**	**0.174**	**0.408**	**1.331**
4	EM	**2.008**	**13.156**	0.008	0.043	0.386	1.034	**4.553**
	tmp-EM	2.998	20.1	**0.006**	**0.028**	**0.374**	**0.637**	5.468
5	EM	1.772	12.175	0.005	0.015	0.035	0.664	5.813
	tmp-EM	**0.791**	**7.088**	**0.005**	**0.011**	**0.026**	**0.058**	**2.57**
6	EM	2.909	59.913	0.012	0.045	0.195	0.676	4.371
	tmp-EM	**2.072**	**25.898**	**0.008**	**0.023**	**0.062**	**0.34**	**2.883**

Conclusion

We saw that tmp-EM achieved better average and median results with lower variances both on likelihood maximisation and parameter recovery for every Class (with very rare exceptions). A more global look at the overall distributions confirms this trend: the error of tmp-EM are more centred on 0 with less spread than EM. This indicates that the tempering allows the EM algorithm to avoid falling into the first local maximum available and consistently find better ones. From the Machine Learning point of view, we highlighted that with our GMM parameters and $n = 500$ observations, it was able to better identify the different centroids, despite their ambiguity than the regular EM procedure. Table S3 presents a comparative synthesis of the results of EM and tmp-EM.

Table S3. Synthetic table focusing solely on the average and standard deviation (in parenthesis) of the losses and parameter reconstruction errors made by EM and tmp-EM. We note that the likelihood reached is higher with lower variance, and similarly, the parameter metrics on almost every class are better with lower variance for tmp-EM.

Metric	class	EM	tmp-EM
$-\ln p_{\hat{\theta}}$		2 247.08 (69.62)	**2 218.80 (35.21)**
$\frac{\ln p_{\theta_0} - \ln p_{\hat{\theta}}}{\ln p_{\theta_0}}$		0.12 (0.04)	**0.13 (0.04)**
$\frac{\hat{\pi}_k - \pi_k}{\pi_k}$	1	–0.19 (0.36)	**–0.17 (0.29)**
	2	0.11 (0.57)	**0.04 (0.33)**
	3	0.56 **(0.81)**	**0.45 (0.83)**
	4	**0.10 (0.57)**	0.10 **(0.43)**
	5	–0.08 (0.48)	**–0.02 (0.31)**
	6	–0.20 (0.43)	**–0.13 (0.40)**
$\frac{\lVert\hat{\mu}_k - \mu_k\rVert^2}{\lVert\mu_k\rVert^2}$	1	0.02 (0.12)	**2.10^{-3} (0.01)**
	2	0.04 **(0.07)**	**0.03 (0.07)**
	3	0.97 (1.15)	**0.74 (1.07)**
	4	0.31 (0.49)	**0.29 (0.48)**
	5	0.73 (1.25)	**0.43 (1.05)**
	6	1.94 (2.83)	**0.81 (1.74)**
$KL(\Sigma, \widehat{\Sigma})$	1	2.74 (39.88)	**0.84 (8.68)**
	2	0.85 **(9.01)**	**0.41 (9.07)**
	3	1.18 (14.64)	**0.65 (4.44)**
	4	**2.01 (13.16)**	3.00 (20.10)
	5	1.77 (12.17)	**0.79 (7.09)**
	6	2.91 (59.91)	**2.07 (25.90)**

2.2. *Experiment 2: 3 clusters*

In this section, we will assess the capacity of tmp-EM to escape from sub-optimal local maxima near the initialisation. The experimental protocol is the same as in the main paper. Let us recall it here. We confront the algorithm to situations where the true classes have increasingly more ambiguous positions, combined with initialisations designed to be hard to escape from. Even though we still follow the log-likelihood as a critical metric, for illustrative purposes we put more emphasis in this section on visualising whether the clusters were properly identify and following the paths in the 2D space of the estimated centroids towards their final values during the EM procedures.

The setup is the following: we have three clusters of similar shape and same weight. One is isolated and easily identifiable. The other two are next to one another, in a more ambiguous configuration. Figure S6 represents the three, gradually more ambiguous configurations.

We use two different initialisation types to reveal the behaviours of the two EMs. The first - which we call "barycenter" - puts all three initial centroids at the centre of mass of all the observed data points. However, none of the EM procedures would move from this initial state if the three GMM centroids were at the exact same position, hence we actually apply a tiny perturbation to make them all slightly distinct. The blue crosses on Figure S7 represent a typical barycenter initialisation. With this initialisation method, we assess whether the EM procedures are able to correctly estimate the positions of the three clusters, despite the ambiguity, when starting from a fairly neutral position, providing neither direction nor misdirection. On the other hand, the second initialisation type - which we call "2v1" - is voluntarily misguiding the algorithm by positioning two centroids on the isolated right cluster and only one centroid on the side of the two ambiguous left clusters. The blue crosses on Figure S8 represent a typical 2v1 initialisation. This initialisation is intended to assess whether the methods are able to escape the potential well in which they start and make theirs centroids traverse the empty space between the left and right clusters to reach their rightful position. For each of the three parameter families represented on Figure S6, 1000 datasets with 500 observations each are simulated, and the two EMs are ran with both the barycenter and the 2v1 initialisation. In the case of tmp-EM, the oscillating temperature profile is used with parameters $T_0 = 5$, $r = 2$, $a = 0.6$, $b = 20$ for the barycenter initialisation, and $T_0 = 100$, $r = 1.5$, $a = 0.02$, $b = 20$ for the 2v1 initialisation. Although in the case of 2v1, the oscillations are not critical, and the simple temperature profile with $T_0 = 100$ and $r = 1.5$ works as well. We have two different sets of tempering hyper-parameters values, one for each of the two very different initialisation types. However, these values then remain the same for the three different parameter families and for every data generation within them. Underlining that the method is not excessively sensitive to the tempering parameters. The experiment with 6 clusters in Section 2.1, already demonstrated that the same hyper parameters could be kept over different initialisation (and different data generations as well) when they were made in a non-adversarial way, by drawing random initial centroids uniformly among the data points.

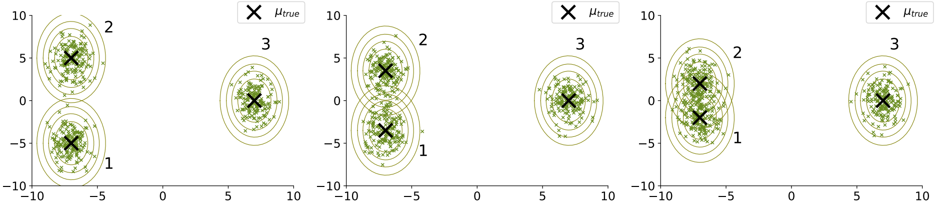


Figure S6. 500 sample points from a Mixture of Gaussians with 3 classes. The true centroid of each Gaussian are depicted by black crosses, and their true covariance matrices are represented by the confidence ellipses of level 0.8, 0.99 and 0.999 around the centre. There are three different versions of the true parameters. From left to right: the true μ_k of the two left clusters (mu_1 and mu_2) are getting closer while everything else stays identical.

2.2.1. Illustrative

First we illustrate on unique examples how tmp-EM is able to avoid falling for the tricky initialisations we set up.
As previously stated, the focus will be less on the likelihood optimisation for these illustrative examples. Indeed, they are meant to demonstrate that tmp-EM is able to cross the gaps and put the clusters in the right place even with the disadvantageous initialisation. The more relevant metric to assess success in this task is the error on μ (and in a lesser way, the error on Σ). One reason why the likelihood looses its ability to discriminate between failure and success in escaping the traps set by the initialisations is that there may not be a big likelihood gap between being completely wrong and mostly right. For instance placing two centroids (one of which is linked to an empty class) on the isolated left cluster and putting only one where the two ambiguously close clusters are could have a decent likelihood while being blatantly wrong.

On Figure S7, we represent the results of each EM after convergence for every of the three parameter set, when the start at the barycenter of all data points (blue crosses). The estimated means and covariance matrices of the GMM are represented by orange crosses and confidence ellipses respectively. In those examples, tmp-EM correctly identified the real clusters whereas EM put two centroids on the right, where only the isolated cluster stands, and only one on the left, where the two ambiguous clusters are. Figure S8 shows similar results, with the same conventions in the case of the "2v1" initialisation.

These different outcomes are exactly what one would expect: unlike the classical EM, tmp-EM is by design supposed to avoid the local minima close to the initialisation by

taking a more exploratory stance during its first steps. To demonstrate that point, we detail in Figure S9 to S12 the paths taken by the estimated centroids by tmp-EM in those simulations. The paths of the regular EM are straightforward convergences towards their final positions, and are not represented in these supplementary materials. Figure S9 represents the paths of the three cluster centroids during the iterations of tmp-EM. The parameter family is the least ambiguous (the two left cluster are well separated) with the "barycenter" initialisation. On Figure S10, the initialisation is "2v1" instead. The two following Figures, S11 and S12, also features the initialisations "barycenter" and "2v1" respectively, but with the most ambiguous parameter set, where the two left clusters are very close to one another. These graphs are made of several rows of figures, each row representing a step in the EM procedure. In order to make the Figures informative, the number of steps between each row is not fixed, instead the most interesting steps are represented. Convergence is always achieved within 20 to 50 steps, so there are never big differences between the step gaps anyway. The first row is always the initial stage without any EM step, and the last one is the stage after convergence. Each of the three columns corresponds to one of the three centroids estimated by the EM procedure and represents its evolution in the 2D space, from initialisation to convergence. The corresponding estimated covariance matrix is represented by confidence ellipses. For each of the centroids, the observed data points are coloured accordingly to their (un-tempered) posterior probability of belonging to the associated class at this stage of the the algorithm. Plain blue being a low probability while bright green is a high probability.
We make the following observations on the steps taken by tmp-EM: with a "barycenter" initialisation (Figure S9 and S11), the three centroids gradually converge towards their final position (which correspond to true class centres in these cases) without too much hesitation. We also note that, since the three initial points are slightly distinct, there appears to be preferences at the very beginning, with each class having different high probability points right at the initialisation stage. However those preferences are not respected after a couple EM step, we generally see the centroids directing themselves towards different points than their initial favoured ones. This can be attributed to the tempering reshuffling the positions and preferences at the beginning. The "2v1" initialisation illustrates this phenomenon more clearly and in doing so, showcases the true power of the tempering. The very first steps after this very adversarial initialisation are not very remarkable: the single centroid on the left solidifies its position at the centre of the two ambiguous clusters, while the two centroids on the right try to share the single cluster they started in. However, very quickly this status quo is shattered and every estimated centroid jumps to a completely different position. On both Figure S10 and S12 we see the positions being completely reversed with the lonely centroid moving from the two left clusters to the isolated right one whereas the two close centroids make the inverse trip to reach the two clusters on the left. This jump is an indication that the tempering flattened the likelihood enough to allow each centroid to escape their potential wells. Effectively redoing the initialisation and allowing itself to start from more favourable positions. This behaviour is unattainable with the classical EM.

2.2.2. Quantitative

The quantitative analysis can be found in the main paper.

2.3. Experiment on real data: Wine recognition dataset

To further validate tmp-EM, we compare it once more to the unmodified EM, this time on real observations from the scikit learn [1] classification data base "Wine" [4]. This dataset contains $p = 13$ chemical measurements of $n = 178$ wines each belonging to one of $K = 3$ families. Despite being in high dimension, this dataset is known as not very challenging (the classes are separable) and useful for testing new methods. We expect the unmodified EM to perform quite well already. For tmp-EM, we use the simple decreasing temperature profile, with no oscillations, the tempering parameters are $T_0 = 100$, $r = 4$. Table S4 shows the result of 500 runs of the EMs from different random initial points. We focus on the

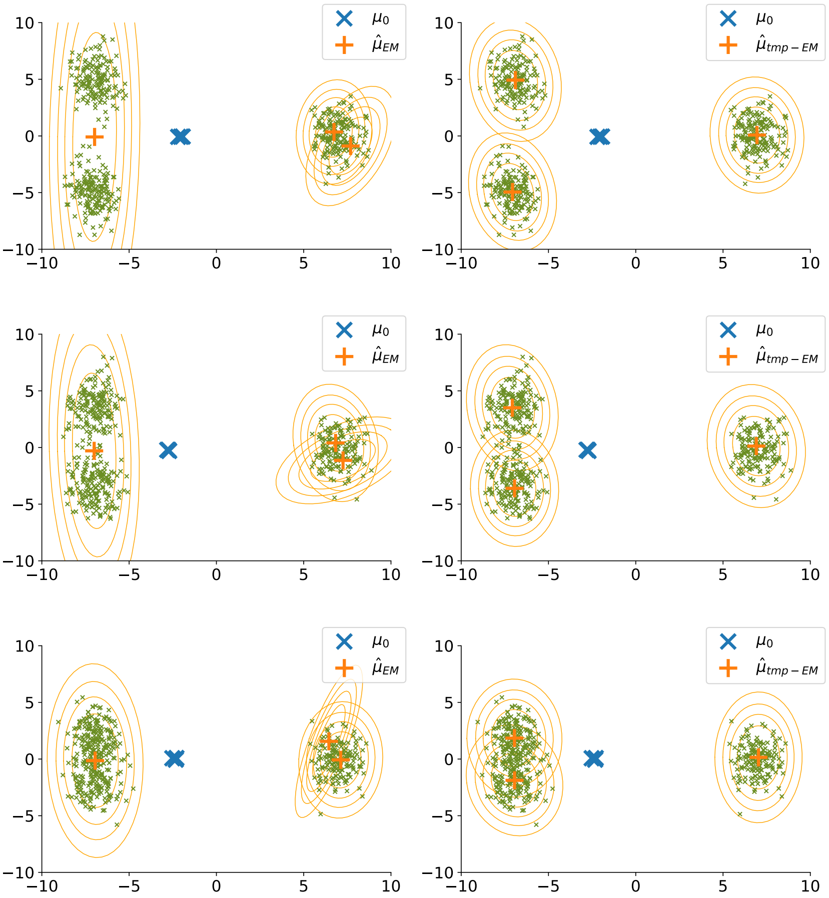


Figure S7. Typical final positioning of the centroids by EM (left column) and tmp-EM (right column) **when the initialisation is made at the barycenter of all data points** (blue crosses). The three rows represent the three gradually more ambiguous parameter sets. Each figure represents the positions of the estimated centroids after convergence of the EM algorithms (orange cross), with their estimated covariance matrices (orange confidence ellipses). In each simulation, 500 sample points were drawn from the real GMM (small green crosses). In those example, tmp-EM managed to correctly identify the position of the three real centroids.

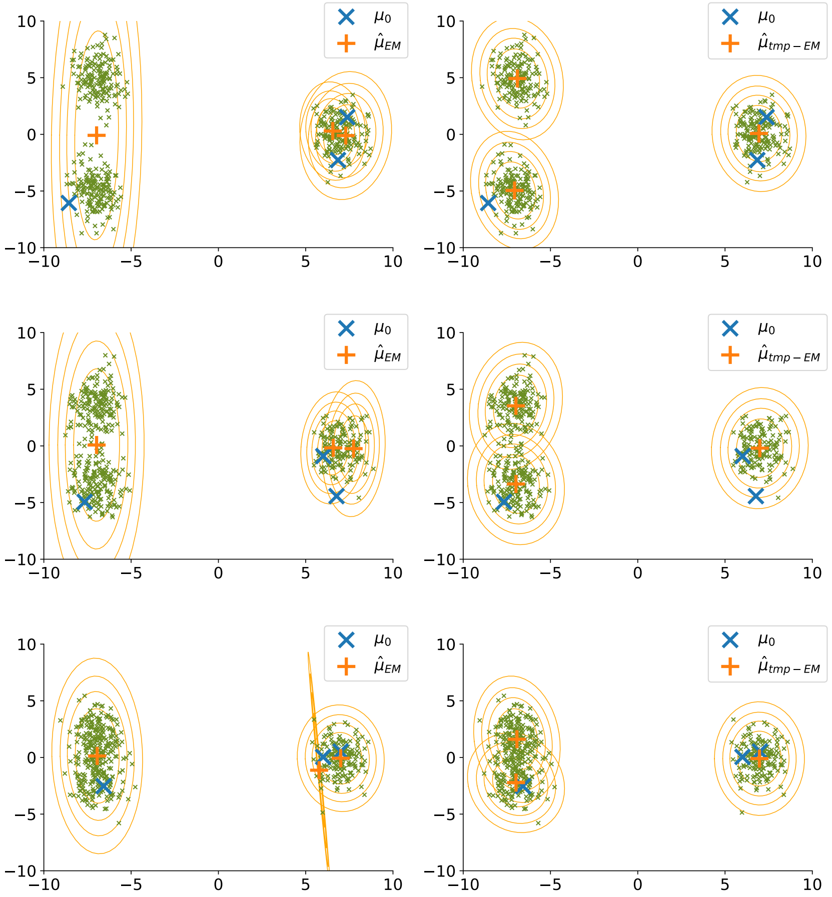


Figure S8. Typical final positioning of the centroids by EM (left column) and tmp-EM (right column) **when the initialisation is made by selecting two points in the isolated cluster and one in the lower ambiguous cluster** (blue crosses). The three rows represent the three gradually more ambiguous parameter sets. Each figure represents the positions of the estimated centroids after convergence of the EM algorithms (orange cross), with their estimated covariance matrices (orange confidence ellipses). In each simulation, 500 sample points were drawn from the real GMM (small green crosses). In those examples, although EM kept two centroids on the isolated cluster, tmp-EM managed to correctly identify the position of the three real centroids.

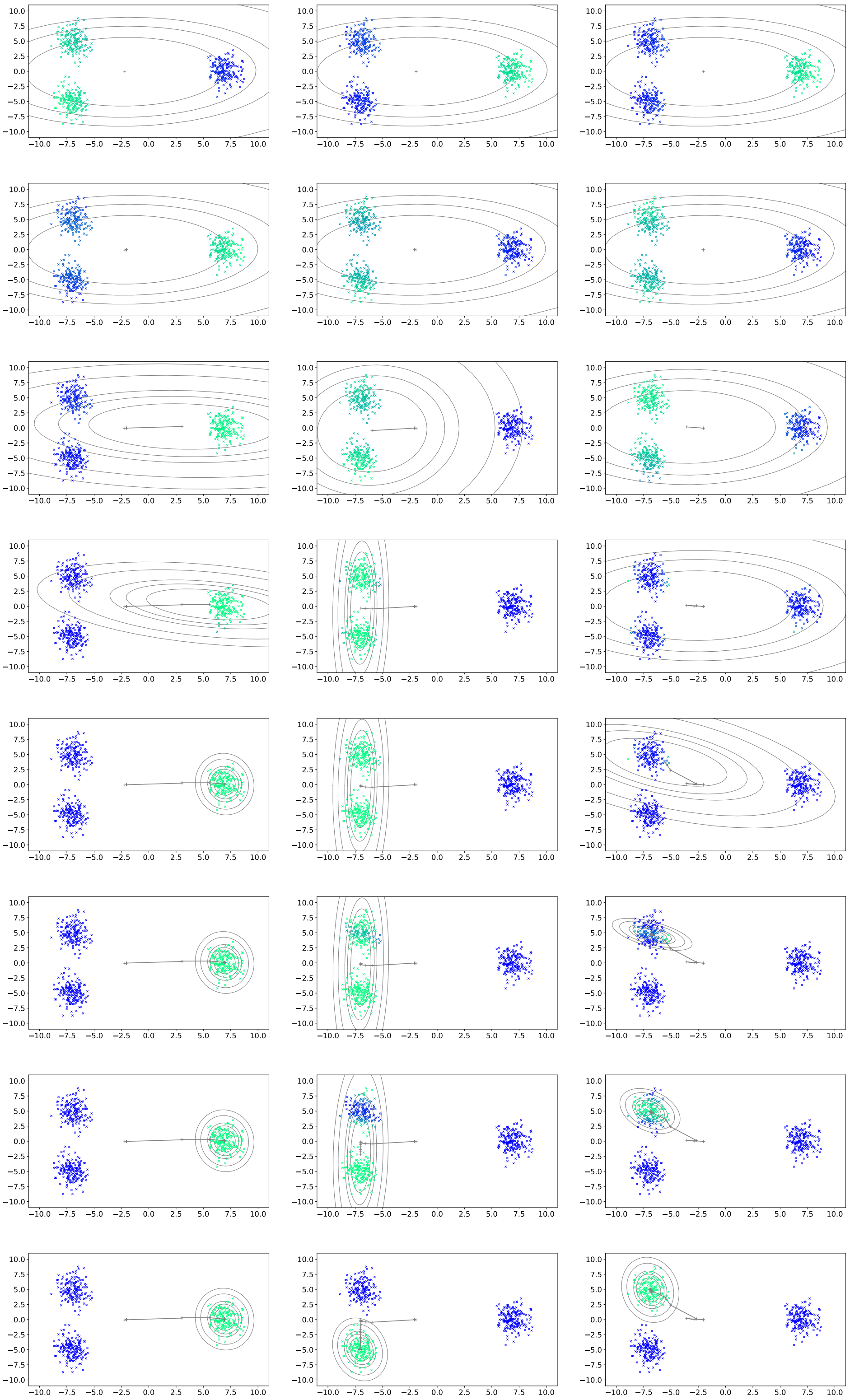


Figure S9. Paths of the centroids for tmp-EM with the "barycenter" initialisation. Parameter set 1 (least ambiguous).

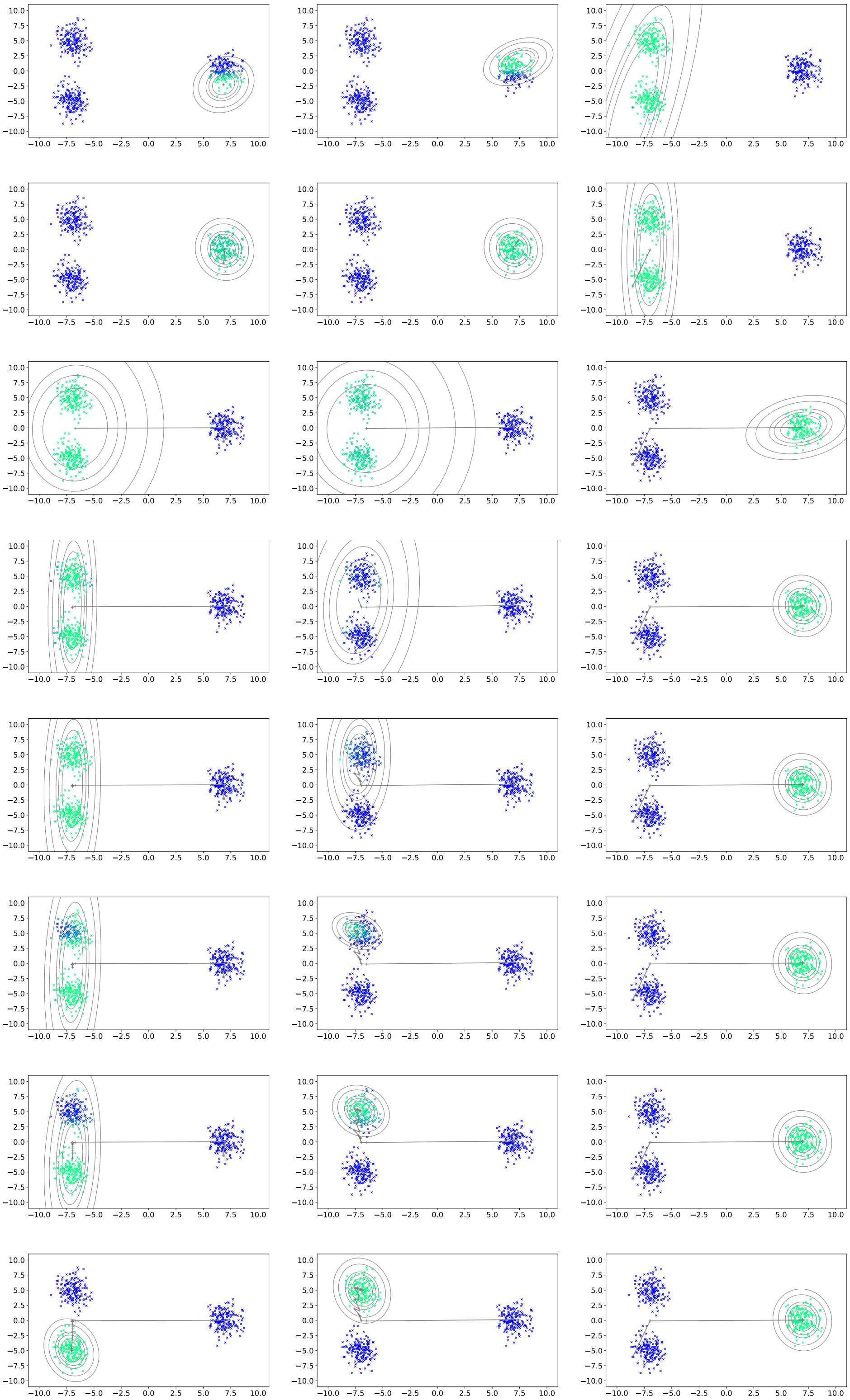


Figure S10. Paths of the centroids for tmp-EM with the "2v1" initialisation. Parameter set 1 (least ambiguous).

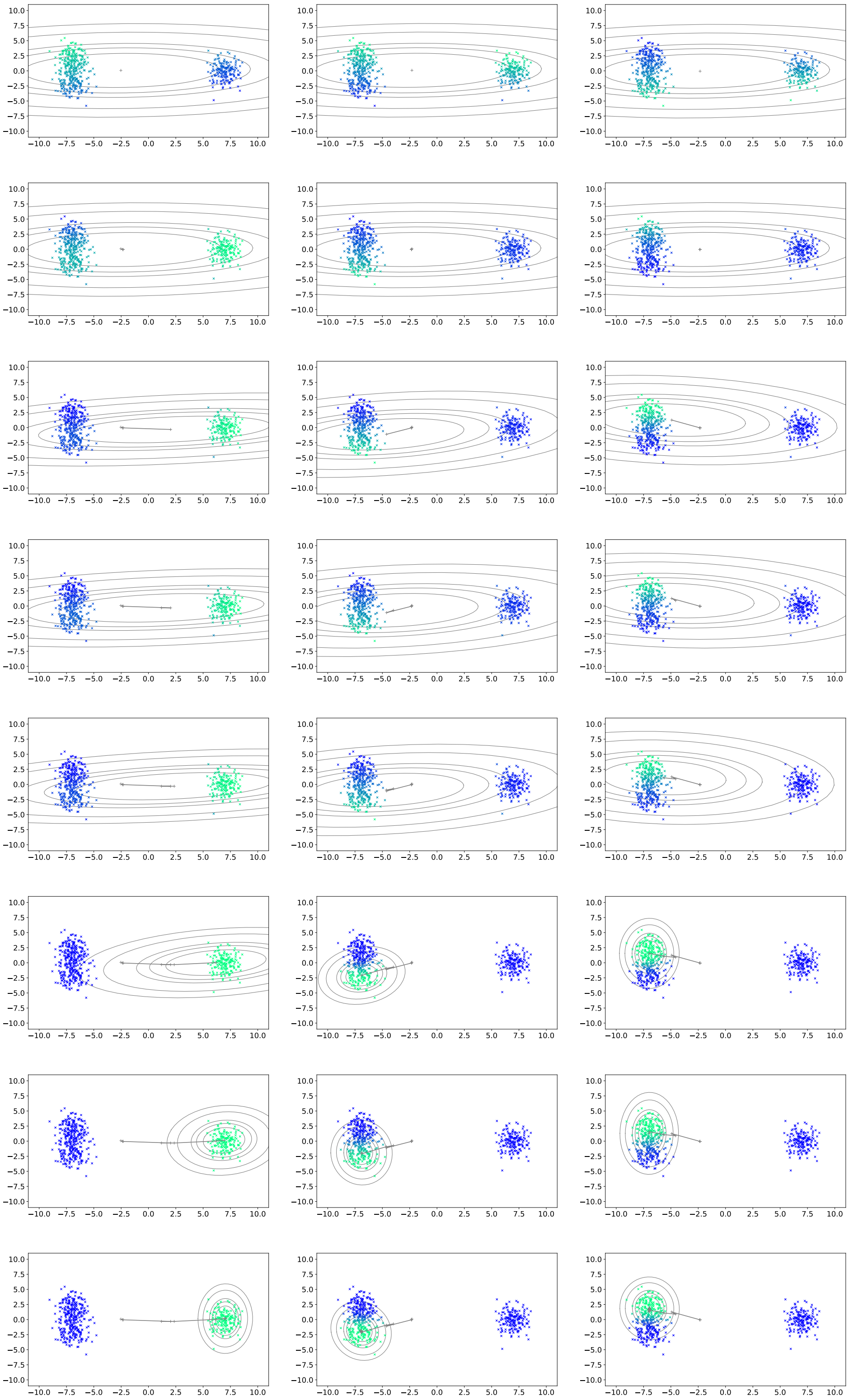


Figure S11. Paths of the centroids for tmp-EM with the "barycenter" initialisation. Parameter set 3 (most ambiguous).

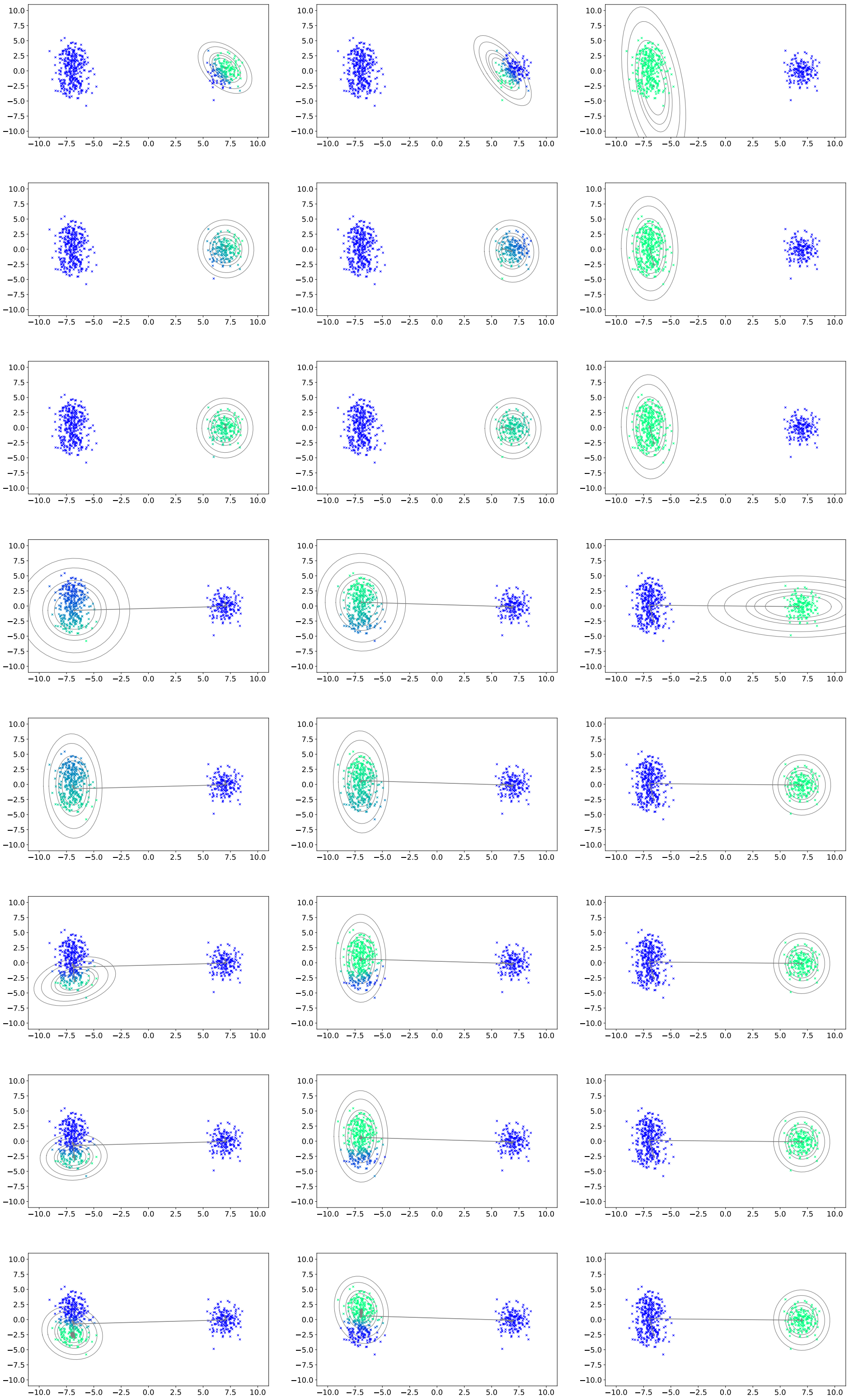


Figure S12. Paths of the centroids for tmp-EM with the "2v1" initialisation. Parameter set 3 (most ambiguous).

likelihood and the error on μ_k, the other relevant metrics, not presented here, show the same tendencies. We observe, as usual, that tmp-EM reaches in average a lower negative log-likelihood with lower variance. The class centres are also better estimated. As expected, the errors made by the EM are already fairly small, however tmp-EM manages to go further and lower the errors on each class by approximately 17%, 18% and 11% respectively.
The results demonstrate that tmp-EM can improve the EM result on real data. Since this is an easy dataset, the difference is not as drastic as in the hard synthetic cases we ran the EMs by. Still, there was room to improve the EM results, and tmp-EM found those better solutions.

Table S4. Average and (standard deviation) of the EM and tmp-EM results over 500 random initialisation on the Wine recognition dataset. The classes on this dataset are easily identifiable hence the errors are low. Yet tmp-EM still improved upon the solutions of EM

metric	cl.	EM	tmp-EM
$-\ln p_{\hat{\theta}}$		2923 (77)	**2905 (71)**
$\frac{\|\hat{\mu}_k-\mu_k\|^2}{\|\mu_k\|^2}$	1	0.017 (0.030)	**0.014 (0.028)**
	2	0.026 (0.034)	**0.021 (0.033)**
	3	0.089 (0.165)	**0.079 (0.156)**

3. Experiments on tmp-EM with Independent Factor Analysis

In this section, we present another application of the tmp-EM with Gaussian Mixture Models, but this time as part of a more complex model. The Independent Factor Analysis (IFA) model was introduced by [2] as an amalgam of Factor Analysis, Principal Component Analysis and Independent Component Analysis to identify and separate independent sources mixed into a single feature vector. From a practical standpoint, the mixing coefficient of each source is assumed to be drawn from a GMM, hence the EM. After estimation of the GMM parameters, the sources are recovered with an optimal non linear estimator. This is a complex model in which the EM plays a key part, works like [5] and [3] use it to assess new variants of the EM on a very practical application. The model is described as follows:

$$\forall i = 1, ..., L', \quad y_i = \sum_{j=1}^{L} H_{ij} x_j + u_i \, .$$

Where $y \in R^{L'}$ is one vector of observations, $H \in \mathbb{R}^{L'L}$ is the fixed matrix of the sources, $u \in R^{L'}$ the vector of noise, and $x \in R^{L}$ the random mixing coefficient. Each component x_j is assumed to be drawn from its own GMM.
An EM that converges too soon towards a local extremum has every chance to yield sub-optimal estimated sources. We demonstrate in this section that an IFA method with tmp-EM can recover sources closer to the original when they are known, and cleaner, more stable looking sources in general.

3.1. *Synthetic IFA*

We start with a toy example, where the true sources are two easily distinguishable images. As shown on Figure S14, one is a white square on a black background and the other is a white cross on a similar black background but positioned differently. However, once these two sources are mixed and noised, it becomes much harder to identify them with the naked eye - as illustrated by Figure S14 - and a quantitative method is required to properly separate them. To separate the sources, the identification model assumes that the coefficients used to mix the two sources are drawn from mixtures of gaussian. The outputs were voluntarily generated in a different way to show the generalisation capabilities of the

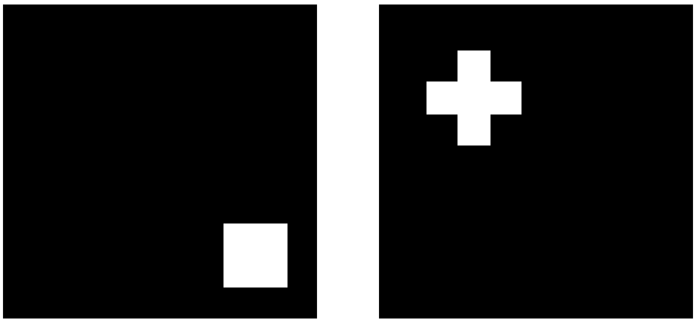


Figure S13. The two real sources of a synthetic source mixing model. They are images of size 20 × 20 made of a black background with a white symbol localised either on the bottom left or top right corner.

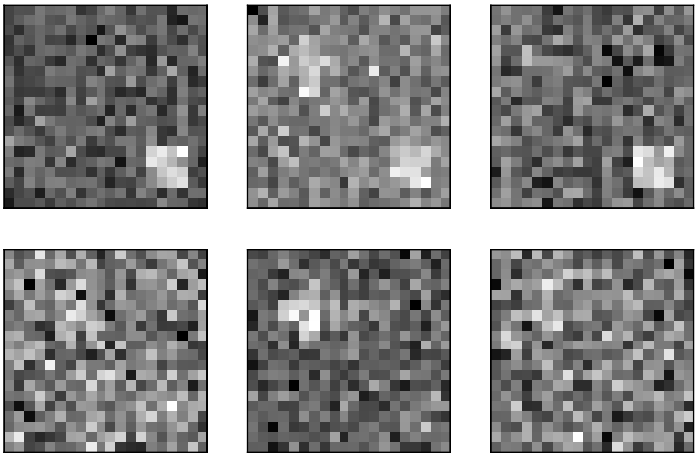


Figure S14. 6 typical observation obtained with the source mixing model. With the noise, the sources are harder to identify.

mixture of gaussian assumption. We run an EM and a tmp-EM algorithm to estimate the parameter of those mixtures, recovering in the process an estimation of the mixing matrix H. Figure S15 illustrates the sources typically estimated by each of the two procedure. Although there is noise, tmp-EM essentially identified and corrected the sources correctly. Whereas EM did not manage to completely turn off the square symbol in the estimated sources supposedly dedicated to the cross. Figure S16 displays the quantitative results of several runs over different simulated datasets. It represents the empirical distribution of l_2 errors made on the estimation of the source matrix H by the two EMs. As illustrated by the table in Figure S16, the solutions of tmp-EM have lower mean and median.

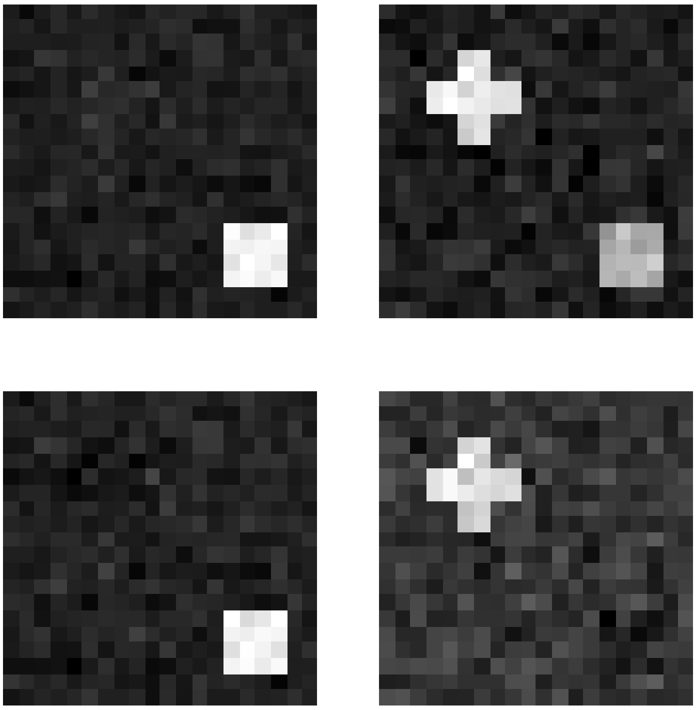
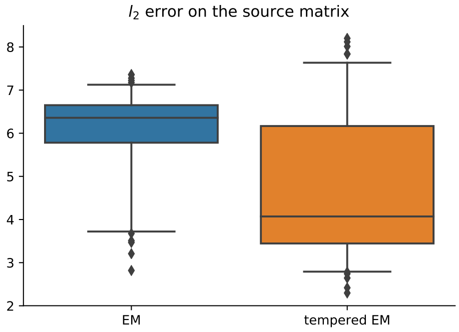


Figure S15. Estimated sources by EM (up) and tmp-EM (down). The two real sources were correctly identified by tmp-EM, but EM did not fully separate the cross and the square.



	EM	tmp-EM
median	6.36	**4.07**
mean	6.01	**4.75**
std	**1.05**	1.65
Q5	3.72	**2.79**
Q25	5.78	**3.44**
Q75	6.65	**6.17**
Q95	**7.13**	7.64

Figure S16. Empirical distribution of the l_2 error on the source matrix H made by EM and tmp-EM. With tmp-EM, we shift the distribution towards the lower errors, with smaller average and median. The numeric values of the quantiles and other statistics can be found in the table, the better ones being in **bold**.

3.2. ZIP code

We apply this IFA algorithm to the ZIP code dataset from Elements of Statistical learning. This dataset contains handwritten digits between 0 and 9. In this study, we keep only the digits 0,3, 8 (all three being ambiguously similar) and 7 (very different from the three others). We make all classes even by removing half of the 0 which are originally more numerous. When applying Independent Factor Analysis to such data, one hopes that the distinct digits will be identified as the separable sources making up the signal. We run the IFA model with a Mixture of Gaussians model with a regular and a tempered EM. In the mixing model used, each mixture is composed of two classes. The tempering was made with the oscillating profile, with hyper-parameters: $T_0 = 50$, $b = 20$, $r = 3$, $a = 0.02$. Figure S17 displays the estimated sources by the IFA procedure with either EM or tmp-EM at their core. EM did not really identify an "8" source. Instead, its "3" is a bit ambiguously close to and "8", and the rightmost source in Figure S17 seems like an amalgamation of the four digits. Moreover, the source "7" estimated by EM is actually a mix between a "7" and a "0". On the other hand, the sources estimated by tmp-EM each correspond clearly to a different digit. There is an "8", the "7" is not fused with a "0", the "3" is sharper and more distinct from an "8" then the corresponding EM source, and even the "0" is more symmetrical with tmp-EM than with EM. Tempering the EM within the IFA algorithm allowed for a cleaner separation of the sources. One can infer that tmp-EM was able to identify and reach a better local maximum of the loss function.

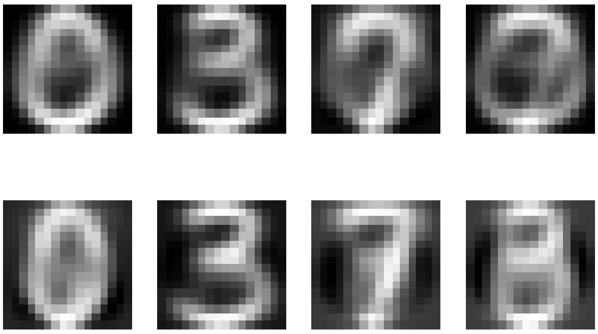


Figure S17. Estimated sources by EM (up) and tmp-EM (down). The "8" and the "7" in particular were much better identified by tmp-EM. Moreover, with tempering the "0" has a more symmetrical shape and the "3" is sharper and less ambiguous.

1. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
2. Attias, H. Independent factor analysis. *Neural computation* **1999**, *11*, 803–851.
3. Allassonnière, S.; Chevallier, J. A new class of stochastic EM algorithms. Escaping local maxima and handling intractable sampling. *Computational Statistics & Data Analysis* **2021**, *159*, 107159.
4. Dua, D.; Graff, C. UCI Machine Learning Repository, 2017.
5. Allassonniere, S.; Younes, L.; et al. A stochastic algorithm for probabilistic independent component analysis. *The Annals of Applied Statistics* **2012**, *6*, 125–160.