Quantitative Finance, 2014
Vol. 14, No. 5, 913–930, http://dx.doi.org/10.1080/14697688.2013.861075

A mixture of Gaussians approach to mathematical portfolio oversight: the EF3M algorithm

MARCOS LÓPEZ DE PRADO*†‡ and MATTHEW D. FOREMAN§

†Quantitative Trading, Hess Energy Trading Company, New York, NY, USA
‡Lawrence Berkeley National Laboratory, Computational Research, 1 Cyclotron Road, Berkeley, CA, USA
§Mathematics, University of California, Irvine, CA, USA

(*Received 12 April 2012; accepted 22 October 2013*)

An analogue can be made between: (a) the slow pace at which species adapt to an environment, which often results in the emergence of a new distinct species out of a once homogeneous genetic pool and (b) the slow changes that take place over time within a fund, mutating its investment style. A fund's track record provides a sort of genetic marker, which we can use to identify mutations. This has motivated our use of a biometric procedure to detect the emergence of a new investment style within a fund's track record. In doing so, we answer the question: *What is the probability that a particular PM's performance is departing from the reference distribution used to allocate her capital?* The EF3M algorithm, inspired by evolutionary biology, may help detect early stages of an evolutionary divergence in an investment style and trigger a decision to review a fund's capital allocation.

Keywords: Skewness; Kurtosis; Mixture of Gaussians; Moment matching; Maximum likelihood; EM algorithm

JEL Classification: C13, C15, C16, C44

1. Introduction

Shortly after the publication of Darwin (1859), several statistical methods were devised to find empirical evidence supporting the Theory of Evolution. To that purpose, Francis Galton set the foundations of 'regression analysis'. With the help of other students of Darwin's theory, Galton established the journal *Biometrika*, where Karl Pearson, Ronald Fisher, William (Student) Gosset, Francis Edgeworth, David Cox and other 'founding fathers' of modern statistical analysis published their work. 'Mixture distributions' were originally devised as a tool to demonstrate *Evolutionary divergence*. Pearson (1894) noted that the breadth of Naples crabs' forehead could be accurately modelled by mixing two Gaussian distributions, which would indicate that a new species of crab was emerging from, and becoming distinctly different to, a once homogeneous species. Many statistical methods were inspired by 'Evolutionary' ideas, and remembering that connection can help us see apparently unrelated matters in a new light.

Ever since Pearson's work, mixtures have been applied to problems as varied as modelling complex financial risks (Alexander 2001, 2004, Tashman and Frey 2008), fitting the implied volatility surface (Rebonato and Cardoso 2004), stochastic processes (Brigo *et al*. 2002), handwriting recognition (Bishop 2006), housing prices, topics in a document, speech recognition and many examples of clustering or unsupervised learning procedures in the fields of Biology, Medicine, Psychology, Geology, etc. (Makov *et al*. 1985). In this paper, we will apply mixtures to the problem of 'portfolio oversight'. In the financial application we present in this paper, the connection between mixtures and 'Evolution' is more evident than in other instances cited above.

Mixture distributions are derived as convex combinations of other distribution functions. They are non-Normal, because their observations are not drawn simultaneously from all distributions, but from one distribution at a time. For example, in the case of a mixture of two Gaussians, each observation has a probability p of being drawn from the first distribution and a probability $1-p$ of coming from the second distribution (the observation cannot be drawn from both). Mixtures of Gaussians are extremely flexible non-Normal distributions and even the mixture of two Gaussians covers an impressive subspace of moments' combinations (Bailey and López de Prado 2012).

*Corresponding author. Email: lopezdeprado@lbl.gov

Pearson (1894) was the first to propose the method of moment matching (MM), which consists in finding the parameters of a mixture of two Gaussians to match the first five moments. A gigantic work of algebraic manipulation allowed him to represent the solution in terms of a 9th degree (nonic) polynomial in five variables. This results in a variety of roots and the issue of making the correct choice arises. Undoubtedly, he must have thought that this problem was worth the weeks and months that deriving this complex system must have consumed. Computational limitations of those days made this approach intractable. More recently, Craigmile and Titterington (1997), Wang (2001) and McWilliam and Loh (2008) have revived interest in MM algorithms.

While working at NASA, Cohen (1967) devised a rather convoluted way to circumvent Pearson's nonic equation by initially assuming equality of the variances. A cubic equation could then be solved for a unique negative root, which could then be fed into an iterative process. With the help of a conditional maximum likelihood (ML) procedure, he attempted to eliminate the effect of sampling errors resulting from the direct use of the fifth moment. Similarly, Day (1969) published in 'Biometrika' a procedure for estimating the components of a mixture of two Normal distributions through ML.

Since Dempster *et al*. (1977) introduced the expectation-maximization (EM) algorithm, this has become the preferred general approach to fitting a mixture distribution (see Hamilton 1994 for an excellent reference). The EM algorithm searches for the parameter estimates that maximize the posterior conditional distribution function over the entire sample. Higher moments for which the researcher may have no theoretical interpretation or confidence, are impacting the parameter estimates. For example, financial markets theories typically have no interpretation for moments beyond the fourth order. It seems reasonable to focus, as the algorithm we present does, primarily on the first four moments for which one has higher confidence and a theoretical interpretation.

Although the MM, ML and EM approaches are extremely valuable, a number of reasons have motivated our proposal for a new answer to this century-old question. First, researchers usually have greater *confidence* in the first three or four moments of the distribution than on higher moments or the overall sample (Bailey and López de Prado 2012). An exact match of the fourth and particularly fifth moment is not always desirable due to their significant sampling errors, which are a function of those moments' magnitude. Basing our estimates on even higher (and therefore noisier) moments, as the ML and EM-based algorithms do, is far from ideal. We would rather have a distribution of parameter estimates we can trust, than a unique solution that is derived from unreliable higher moments. Second, in the Quantitative Finance literature, it is the first four moments that play a key role in the *theoretical modelling* of risk and portfolio optimization (see Hwang and Satchell 1999, Favre and Galeano 2002, Jurcenzko and Maillet 2002, to mention only a few), not the fifth and beyond. Third, in the context of *risk simulation*, often we face the problem of modelling a distribution that exactly matches the empirically observed first three or four moments (e.g. Brooks and Kat 2002, López de Prado and Peijan 2004, López de Prado and Rodrigo 2004). Fourth, EM algorithms are *computationally intensive* as a function of the sample size and tend to get trapped on local minima (Xu and Jordan 1996). Speed, and therefore simplicity, is a critical concern, considering that datasets nowadays often exceed hundreds of millions of observations. Fifth, a mixture of two Gaussians offers sufficient flexibility for modelling a wide range of skewness and kurtosis scenarios. Risk and portfolio managers (PM) would greatly benefit from an intuitive algorithm that liberates them from the ubiquitous assumption of Normality.

In this paper, we present a new, practical approach to exactly matching the first three moments of a mixture of two Gaussians. The fourth and fifth moments are used to guide the convergence of the mixing probability, but they are not exactly matched. We call this algorithm *EF3M*, as it delivers an **E**xact **F**it of the first **3** **M**oments. We believe this framework is more representative of the standard problem faced by many researchers. Our examples are inspired by financial applications, however the algorithm is valid for a general mixture of two Gaussians in general. Our algorithm is purely algebraic—given the first few moments of the mixture, we algebraically estimate the means and variances of the two Gaussians and the parameter p for mixing them. The only interaction with the data is in extracting the moments of the mixture. Thus, unlike ML and EM, the EF3M algorithm does not require numerically intensive tasks, and its performance is independent of the sample size, making it more efficient in 'big data' settings, like high-frequency trading.

The moments used to fit the mixture may be derived directly from the data or be the result of an annualization or any other type of time projection, such as proposed by Meucci (2010). For example, we could estimate the moments based on a sample of *monthly* observations, project them over a horizon of one year (i.e. the projected moments for the implied distribution of annual returns) and then fit a mixture on the projected moments, which can then be used to draw random *annual* (projected) returns.

Standard structural break tests (see Maddala and Kim 1999 for a treatise on the subject) attempt to identify a 'break' or permanent shift from one regime to another within a time series. In contrast, the methodology we present here signals the emergence of a new regime as it happens, while it co-exists with the old regime (thus the mixture). This is a critical advantage of EF3M, in terms of providing an early signal. For example, in the particular application discussed in Section 4, the 'portfolio oversight' department will be able to assess the representativeness of a track record very early in their post-track observations. The assumptions and data demands are minimal.

The rest of the paper is organized as follows: Section 2 presents a brief recitation on mixtures. Section 3 introduces the EF3M algorithm. A first variant uses the fourth moment to lead the convergence of the mixing probability, p. A numerical example and the results of a Monte Carlo experiment are presented. In this variant of EF3M, the fifth moment is merely used to select one solution for each run of EF3M. Some researchers may have enough confidence and understanding of the fifth moment to use it for guiding the convergence of p, in which case we propose a second variant of EF3M. Section 4 introduces the concept of *probability of divergence*. Section 5 discusses possible extensions to this methodology. Section 6 outlines our conclusions. Three mathematical appendices prove the equations used by EF3M and a fourth appendix offers the Python code that implements both variants. Sometimes a researcher is only concerned with modelling the variance and tails of the distribution, and not with its mean and skewness. For that particular case, Appendix 5 provides an exact analytical solution.

2. Mixtures of distributions

Here, we provide the highlights of the theory behind mixtures of distributions. Readers familiar with the topic may prefer to skip it and move to Section 3.

Consider a set of distributions $D_0, D_1, \ldots, D_{n-1}$ and positive real coefficients $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ that sum to one. A new distribution D can be defined as a convex combination $D_0, D_1, \ldots, D_{n-1}$ with coefficients $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$, where $D = \sum_{i=0}^{n-1} \alpha_i D_i$. If $d_0, d_1, \ldots, d_{n-1}$ are the densities associated with $D_0, D_1, \ldots, D_{n-1}$ then it is immediate that $d = \sum_{i=0}^{n-1} \alpha_i d_i$ is the density associated with D.

We now give a more conceptual description of D. We put a standard probability measure on $\mathbb{R}$ in the usual way by setting $\mu_i([a,b]) = \int_a^b d_i dx$ and similarly $\mu([a,b]) = \int_a^b d\, dx$. Let (X_i, v_i) be measure spaces isomorphic to $(\mathbb{R}, \mu_i)$ where the X_i's are pairwise disjoint. Define a new measure space (Y, v) by putting $Y = \cup_i X_i$ and $v(A) = \sum_i \alpha_i v_i(A \cap X_i)$. If $f_i : X_i \rightarrow \mathbb{R}$ is the isomorphism between X_i and $\mathbb{R}$, then we can define a random variable $f : X \rightarrow \mathbb{R}$ by setting $f(y) = f_i(y)$ if $y \in X_i$.

Heuristically, f is defined as follows: we first choose i with probability α_i. Then, conditioned on this choice, we use the measure v_i to choose a $y \in X_i$. The value of f is $f_i(y)$. Direct computation then shows that D is the distribution of f. Indeed the probability that $f(y) \in [a,b]$ is given as $\sum_i \alpha_i E_i[f(y) \in [a,b]]$, where E_i is the probability conditioned on $y \in X_i$. This in turn is $\sum_i \alpha_i \int_a^b d_i dx$.

Summarizing in the case of two distributions D_0 and D_1, we can build a third by first choosing with some probabilities $\{p, 1-p\}$ either the first or second distribution and then using the corresponding density to choose a value for our random variable. The new distribution D is given formally by $pD_0 + (1-p)D_1$. We will denote a D distribution built this way a *mixture of* D_0 *and* D_1.

In general, an arbitrary distribution D can be decomposed into mixtures $D = \sum_i \alpha_i D_i$ in infinitely many ways. However, there is a unique canonical distribution of a mixture of Gaussians. Namely, if $D = \sum_i \alpha_i D_i = \sum_j \beta_j E_j$ where each D_i and each E_j is Gaussian, then for all i, $\alpha_i = \beta_i$ and $D_i = E_i$. This follows immediately from the linear independence of the family of Gaussian density functions: if we let $d(a,b,c) = exp(ax^2 + bx + c)$, then the collection of functions $\mathcal{F} = \{1\} \cup \{d(a,b,c) :$ at least *one of* a or b is not zero$\}$ forms a linearly independent family over $\mathbb{R}$. Hence, a density d can be expressed in at most one way as a mixture of functions from $\mathcal{F}$. Since $\mathcal{F}$ contains all of the density functions of Gaussians, a representation of an arbitrary density function as a mixture of Gaussians is unique.

3. The EF3M algorithm

Suppose that we are given the first four or five moments of a distribution D that we assume to be a mixture of two Gaussian distributions D_0 and D_1. We address the question of *How we can estimate the parameters determining* D_0*,* D_1 *and the probability p giving the mixture?* This requires estimating five parameters in total: the mean and variance of the first Gaussian the mean and variance of the second Gaussian, and the probability with which observations are drawn from the first distribution. Knowing that probability, p, implies the probability for the second distribution, $1-p$, because by definition the sum of both probabilities must add up to one.

If D is a mixture of Gaussians then the moments of D can be computed directly from the five parameters determining it. In Appendix 1, we derive D's moments from D's parameters. Unfortunately, in general, knowledge of the first five moments of a mixture of Gaussians is not sufficient to recover the unique parameters of the mixture, so we cannot reverse this computation. On the other hand, using higher moments to recover a unique set of parameters is problematic, as they have substantial measuring errors. Our approach to finding D starts with the first five observed moments about the origin $\tilde{E}[r^i]$ determined by data sampled from D (which we assume to be a mixture of Gaussians). The algorithm starts with some random data and generates mixture parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ that give implied $E[r^i]$ well approximating $\tilde{E}[r^i]$. As we will see later, all we need is an estimate of the first four or five moments, $\tilde{E}[r^i]$, computed on a sample or population. This is the only stage of our analysis in which we deal with actual observations. Should the moments have been computed about the mean instead of the origin, Appendix 1 also shows how to derive the latter from the former.

Notation: Let μ_1, μ_2 be the means of the first two distributions, σ_1, σ_2 the standard deviations and p be the probability determining the mixing. We use the notation $E[r^i]$ to denote the ith moment of our mixture, as implied by μ_1, μ_2, σ_1, σ_2 and p. Appendix 1 shows how the mixture's moments are implied from the mixture's parameters. Later in the paper we will be concerned about fitting some data we have observed that we assume is sampled from D, with observed raw moments $\tilde{E}[r^i]$. We will assume that D is a mixture with actual parameters $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{p})$.

The equations for the moments $E[r^i]$ in terms of μ_1, μ_2, σ_1, σ_2 and p have some useful properties that allow our algorithm to work:

(a) The moments of D about the mean $E[r]$ can be computed as polynomial combinations of the moments about the origin and vice versa. Thus, it suffices to do our computations using moments about the origin.

(b) From the expressions for the first four moments about the origin we get some rational functions that successively express μ_1 in terms of μ_2 and p, σ_1 in terms of μ_1, μ_2 and p, and σ_2 in terms of the 4-tuple $(p, \mu_2, \mu_1, \sigma_1)$.

(c) From the expressions for the first five moments about the origin we get another, independent, rational function that expresses p in terms of $(\mu_2, \mu_1, \sigma_1, \sigma_2)$.

This leads to a very general form of algorithm: To compute some parameters $c_0, c_1, \ldots, c_n$, we express c_1 in terms of c_0, c_2 in terms of $(c_0, c_1), c_3$ in terms of (c_0, c_1, c_2) etc. Finally we express c_0 independently in terms of $(c_1, \ldots, c_n)$. Thus, the algorithm then runs as follows:

(i) An initial guess is made for c_0.
(ii) The relations defining c_i in terms of $(c_0, c_1, \ldots, c_{i-1})$ are used to compute a guess for a candidate for c_i.
(iii) Having defined $(c_1, \ldots, c_n)$ in terms of c_0, we get a new estimate for c_0 using its independent expression in terms of $(c_0, \ldots, c_n)$.
(iv) One loops back the previous two steps to get new guesses for $(c_0, \ldots, c_n)$.
(v) The algorithm runs until some termination criterion is achieved.

In the following sections we describe two algorithms of this form and consider their convergence behaviour.

3.1. *Convergence of the fourth moment*

In this section we describe in more detail our algorithm for recovering the actual parameters $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{p})$, by fitting the raw moments $(E[r], E[r^2], E[r^3], E[r^4])$ implied by $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ to the observed $(\tilde{E}[r], \tilde{E}[r^2], \tilde{E}[r^3], \tilde{E}[r^4])$. We refer the reader to Appendix 2 for the expressions needed by the algorithm. The algorithm starts by taking an initial guess for μ_2 and p. We then successively estimate μ_1, σ_2 and σ_1 using equations (21), (23) and (22) from Appendix 2. Finally, equation (24) allows us to get a new guess for p. We iterate this procedure until the results are stable within some tolerance ε. By trying alternative μ_2 seeds we obtain potential solutions whose first three moments $(E[r], E[r^2], E[r^3])$ exactly match $(\tilde{E}[r], \tilde{E}[r^2], \tilde{E}[r^3])$. Among these solutions we then choose the one that minimizes the error $\omega(\tilde{E}[r^4] - E[r^4])^2 + (1-\omega)(\tilde{E}[r^5] - E[r^5])^2$, where $\omega \in [\frac{1}{2}, 1]$ represents the greater confidence the researcher has on $\tilde{E}[r^4]$ relative to $\tilde{E}[r^5]$.[†] This procedure can be repeated as many times as needed to generate a distribution of mixture parameters consistent with the observed moments.

More precisely, let λ define the range of the search, i.e. we will scan for solutions in $\mu_2 \in [\tilde{E}[r], \tilde{E}[r] + \lambda\sqrt{\tilde{E}[r^2] - (\tilde{E}[r])^2}]$. For a ε tolerance threshold, this defines a step size $\delta = \frac{\lambda\sqrt{\tilde{E}[r^2] - (\tilde{E}[r])^2}}{\varepsilon}$. Scanning equidistant μ_2 within that range, with a sufficiently small step size δ, is approximately equivalent to uniform sampling of μ_2 values. Therefore, the algorithm is bootstrapping the distribution of solutions from the subspace of mixture parameters that fit the first three moments.

Given the moments $(\tilde{E}[r], \tilde{E}[r^2], \tilde{E}[r^3], \tilde{E}[r^4], \tilde{E}[r^5])$, EF3M algorithm requires the following steps:

(1) $\mu_2 = \tilde{E}[r]$.
(2) A random seed for p is drawn from a $U(0, 1)$ distribution, $0 < p < 1$.
(3) Sequentially estimate:
 (a) μ_1: equation (22).
 (b) σ_2^2: equation (24). If the estimate of $\sigma_2^2 < 0$, go to Step 7.
 (c) σ_1^2: equation (23). If the estimate of $\sigma_1^2 < 0$, go to Step 7.
(4) Adjust the guess for p: equation (25). If invalid probability, go to Step 7.
(5) Loop to Step 3 until p converges within a tolerance level ε.
(6) Store $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ and the corresponding $E[r^i]$, $i = 1, \ldots, 5$.
(7) Add δ to μ_2 and loop to Step 2 until $\mu_2 = \tilde{E}[r] + \lambda\sqrt{\tilde{E}[r^2] - (\tilde{E}[r])^2}$.
(8) *Optional tiebreak:* Among all stored results, we can select the $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ for which $\omega(\tilde{E}[r^4] - E[r^4])^2 + (1-\omega)(\tilde{E}[r^5] - E[r^5])^2$ is minimal.

Steps 2 to 5 are represented in figure 1. Our solution requires a small number of operations thanks to the special sequence we have followed when nesting one equation into the next (see Appendix 2). A different sequence would have led to the polynomial equations that made Cohen (1967) somewhat convoluted.

Because not all guessed μ_2 can match all sets of first three moments, we must include the possibility of imaginary roots considered by the algorithm (steps 3.b and 3.c), and invalid probability (step 4). For a feasible μ_2, each iteration of EF3M delivers values that *exactly* matches $\tilde{E}[r]$, $\tilde{E}[r^2]$, $\tilde{E}[r^3]$ and using $\tilde{E}[r^4]$ our simulations showed that the output values p very quickly settle into a neighbourhood of radius ε. Finally, the fifth moment is used for evaluation purposes only, but it is neither exactly fit (like the first three moments) nor it drives the convergence (as the fourth moment does). Appendix 4 shows an implementation of the EF3M algorithm in Python language.

The solution to the problem that is the subject of this paper is rarely unique when the only reliable moments are the first five. This bothered Pearson (1894), who advised using the sixth moment to choose among results. But elying on noisy high moments for selecting one possible solution

[†] This tiebreak step is not essential to the algorithm. Its purpose is to deliver one and only one solution for each run, based on the researcher's confidence on the fourth and fifth moments. In absence of a view on this regard, the researcher may ignore the tiebreak and use every solution to which the algorithm converges (one or more per run).

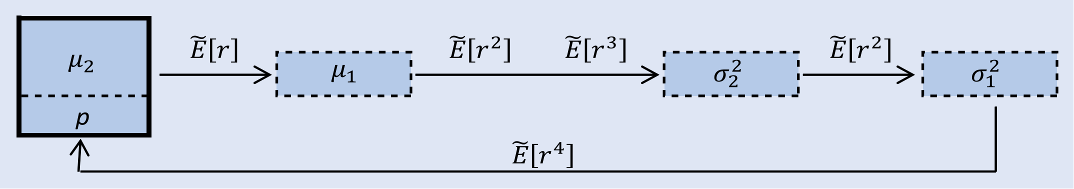
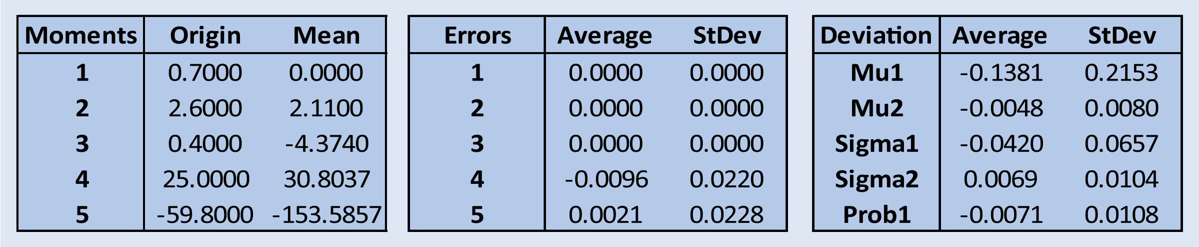


Figure 1. Algorithm's flow diagram (with four moments).



Moments	Origin	Mean
1	0.7000	0.0000
2	2.6000	2.1100
3	0.4000	-4.3740
4	25.0000	30.8037
5	-59.8000	-153.5857

Errors	Average	StDev
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	-0.0096	0.0220
5	0.0021	0.0228

Deviation	Average	StDev
Mu1	-0.1381	0.2153
Mu2	-0.0048	0.0080
Sigma1	-0.0420	0.0657
Sigma2	0.0069	0.0104
Prob1	-0.0071	0.0108

Figure 2. Moments (left), their estimation errors (centre) and departure of the recovered parameters from the original parameters (right).

out of many valid ones seems quite arbitrary. Indeed, from a Bayesian perspective we would prefer working with a distribution of parameter estimates rather than with a unique value for each. Such approach is made possible by the simplicity (translated into speed) of our EF3M's algorithm. This allows the researcher to bootstrap the distribution of alternative values for the mixture's parameters, which can then be used to simulate competing scenarios.

3.2. *A numerical example*

In this section, we consider an example distribution D formed by mixing Gaussians with an arbitrarily chosen set of parameters $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{p}) = (-2, 1, 2, 1, \frac{1}{10})$. The mixture's moments about the origin $(\tilde{E}[r^i])$ and about the mean are given in the left box of figure 2.

We apply EF3M for $\varepsilon = 10^{-4}$, $\lambda = 5$, $\omega = \frac{1}{2}$. This implies searching within the range $\mu_2 \in [0.7, 7.9629]$ by trying 10 000 uniform (equidistant) partitions. We can then repeat this exercise another 10 000 times† and study the distribution of the parameter estimates.

For a given output $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ of the algorithm, we can compare:

(1) The difference between the first five moments $E[r^i]$ implied from $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ and the moments $\tilde{E}[r^i]$ of D,
(2) The differences between $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ and $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{p})$, i.e. between $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ and $(-2, 1, 2, 1, \frac{1}{10})$.

The right two boxes in figure 2 show the average errors over the simulation. The middle middle box gives the errors in the first five moments and the right most box shows the errors in estimating the mixture parameters. The results show that recovered parameters are generally very close to the mixture parameters from D. Figure 3 is a histogram showing with what frequency various estimates of μ_1 occur as outputs of EF3M, in this particular example. Most of the 'errors' in figure 2 are due to the existence of an alternate solution for μ_1 around $\mu_1 \approx -1.56$.

Figure 3 illustrates the fact that, as discussed earlier, there is not a unique mixture that matches the first (and only reliable) moments. However, faced with the prospect of having to use unreliable moments in order to be able to pick one solution, we prefer bootstrapping the distribution of possible mixture's parameters that are consistent with the reliable moments. Our approach is therefore representative of the indetermination faced by the researcher. Section 4 will illustrate how that indetermination can be injected into the experiments, thus enriching simulations with a multiplicity of scenarios.

3.3. *Monte Carlo simulations*

In the previous section, we illustrated the performance of 10 000 runs of EF3M over a particular choice of $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{p})$. The results raise the issue of whether the promising performance is related to the properties of the vector $(-2, 1, 2, 1, \frac{1}{10})$ or whether the algorithm behaves well in general. To test this we randomly change $(\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{p})$ by drawing them from the uniform distributions with boundaries $-1 \leq \mu_1 \leq 0 \leq \mu_2 \leq 1$, $0 < \sigma_1 \leq 1$, $0 < \sigma_2 \leq 1$, $0 < p < 1$.

For $\varepsilon = 10^{-4}$, $\lambda = 5$, $\omega = \frac{1}{2}$, figure 4 shows the statistics of the estimation errors $(\tilde{E}[r^i] - E[r^i])$ and deviations of the fit $(\tilde{\mu}_1 - \mu_1,\ \tilde{\mu}_2 - \mu_2,\ \tilde{\sigma}_1 - \sigma_1,\ \tilde{\sigma}_2 - \sigma_2,\ \tilde{p} - p)$. There is a small departure between the original and recovered parameters, but as the numerical example illustrated, this is explained by the multiplicity of decompositions of a given mixture into its component Gaussians.

3.4. *Convergence of the fifth moment*

We have argued that EF3M's approach is representative of the typical problem faced by most researchers fitting a

†In each repetition we use the same μ_2, however the values of p differ between runs, as they are drawn from a uniform $U(0, 1)$ distribution.

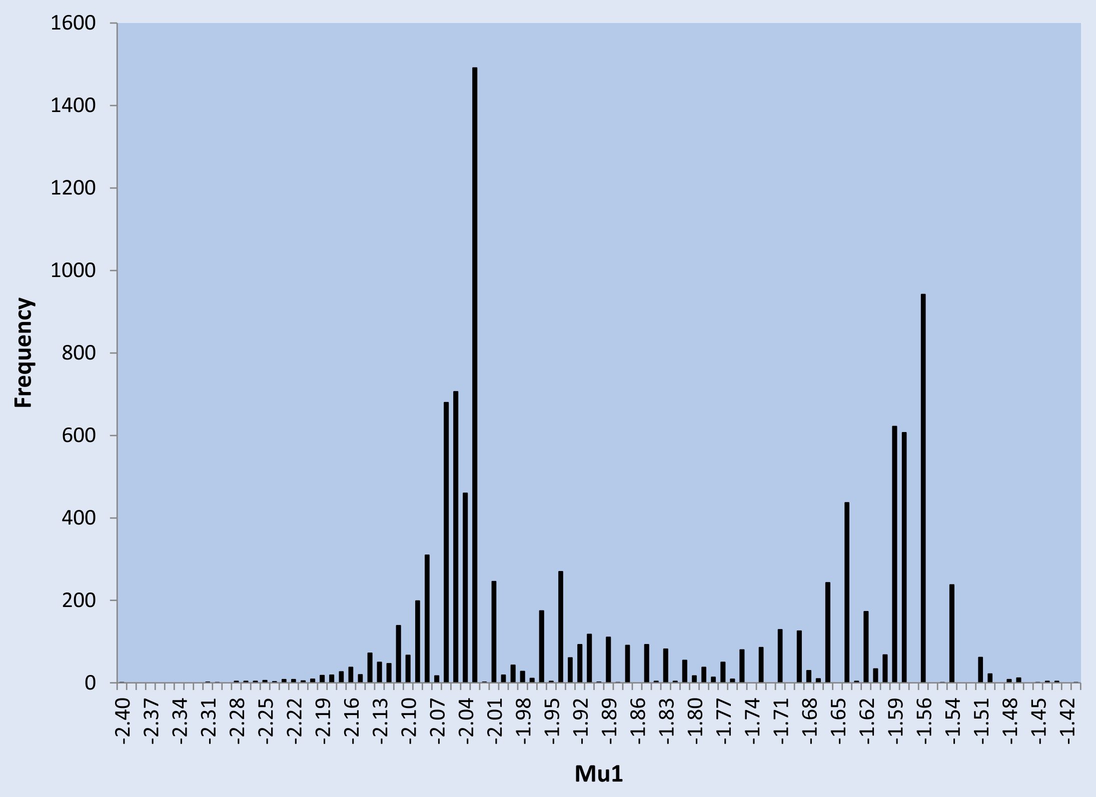
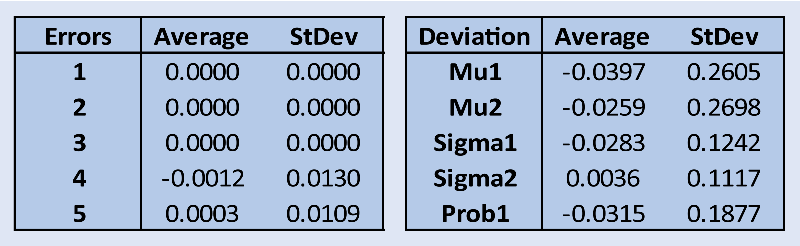


Figure 3. Marginal distribution of probability of the μ_1 parameter.



Errors	Average	StDev
1	0.0000	0.0000
2	0.0000	0.0000
3	0.0000	0.0000
4	-0.0012	0.0130
5	0.0003	0.0109

Deviation	Average	StDev
Mu1	-0.0397	0.2605
Mu2	-0.0259	0.2698
Sigma1	-0.0283	0.1242
Sigma2	0.0036	0.1117
Prob1	-0.0315	0.1877

Figure 4. Estimation error statistics (left) and departure of the recovered parameters from the original parameters (right).

mixture: using the first four moments, for which we have some degree of confidence and theoretical interpretation, to find the distribution of the mixture's parameters. As a consequence of using only four moments, we must start with a guess μ_2 and a random seed p. Although the fourth moment is used for the convergence of p, μ_2 was not re-estimated in each iteration.

However, in those cases where the researcher has some confidence in the fifth moment's estimate, we could use the fourth moment to re-estimate μ_2 and the fifth moment to re-estimate p. In this way, no parameter remains constant across the iterations, which has the further advantage of accelerating the speed of convergence. This second variant of EF3M is very similar to the first one, as it can be seen next:

(1) $\mu_2 = \tilde{E}[r]$.
(2) A random seed for p is drawn from a $U(0,1)$ distribution, $0 < p < 1$.
(3) Sequentially estimate:
    (a) μ_1: equation (26).
    (b) σ_2^2: equation (26). If the estimate of $\sigma_2^2 < 0$, go to Step 8.
    (c) σ_1^2: equation (26). If the estimate of $\sigma_1^2 < 0$, go to Step 8.
(4) Adjust the guess for μ_2: equation (27). If imaginary root, go to Step 8.
(5) Adjust the guess for p: equation (28). If invalid probability, go to Step 8.
(6) Loop to Step 3 until p converges within a tolerance level ε.
(7) Store $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ and the corresponding $E[r^i]$, $i = 1, \ldots, 5$.
(8) Add δ to μ_2 and loop to Step 2 until $\mu_2 = \tilde{E}[r] + \lambda\sqrt{\tilde{E}[r^2] - (\tilde{E}[r])^2}$.
(9) *Optional tiebreak:* Among all stored results, we can select the $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$ for which $\omega(\tilde{E}[r^4] - E[r^4])^2 + (1-\omega)(\tilde{E}[r^5] - E[r^5])^2$ is minimal.

Appendix 3 details the relations used in this second variant of the algorithm. Steps 2 to 6 are represented in figure 5. Note that, although we are re-estimating the value of our guesses of both p and μ_2 during the algorithm, our

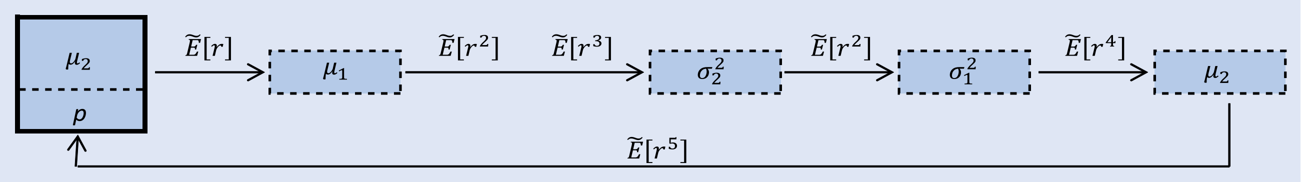


Figure 5. Algorithm's flow diagram (with 5th moment).

initial guesses for μ_2 are still uniformly spaced in our search interval. Thus, this second variant of the EF3M algorithm only requires one additional step 4 and a modification of the equation used in step 5. As it can be seen in Appendix 4, both variants of the EF3M algorithm can be implemented in the same code, with a single line setting the difference.

4. A practical application: portfolio oversight

Investment styles are not immutable, but rather evolve, as prompted by technological and computational advances, among other factors (López de Prado 2011). We began our paper recalling the evolutionary motivation behind many statistical methods and in the case of mixtures in particular. A parallel can be drawn between the slow pace at which species adapt to an environment, creating new distinct species out of a once homogeneous genetic pool and the slow changes that take place over time within a fund. Darwinian arguments can be applied with regards to a fund's or even a PM's struggle for survival in a competitive financial environment. Although a fund's or PM's investment style may evolve so slowly that those changes will be undetectable in the short-run, the track record will accumulate evidence of the 'evolutionary divergence' taking place. Sometimes this divergence will occur as a PM attempts to *adapt* her style to prevail in a certain environment or as an environmental change affects a style's performance. An example of the former is a new technology giving an edge to some market participants and an example for the latter is when the rest of the market adopts that successful technology over time, thus suppressing its competitive advantage.

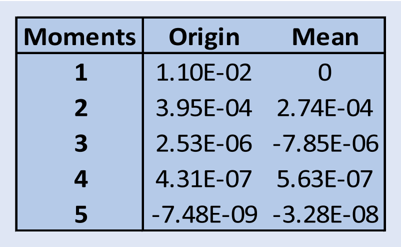
A hedge fund's 'portfolio oversight' department assesses the operational risk associated with individual PMs, identifies desirable traits and monitors the emergence of undesirable ones. The decision to fund a PM is typically informed by her track record. If her recent returns deviate from the track record used to inform the funding decision, the portfolio oversight department must detect it. This is distinct from the role of risk manager, which is dedicated to assessing the possible losses under a variety of scenarios. For example, even if a PM is running risks below her authorized limits, she may not be taking the bets she was expected to, thus delivering a performance inconsistent with her track record (and funding). The risk department may not notice anything unusual regarding that PM, however the portfolio oversight department is charged with policing and detecting such situation. A track record can be expressed in terms of its moments, thus the task of overseeing a PM can be understood as detecting an inconsistency between the PM's recent returns and her 'approved' track record.

More specifically, suppose that we invest in a PM with a track record characterized by IID returns with moments listed in figure 6. Because we have little to no knowledge regarding her investment process, we cannot be certain about how a number of evolving factors (replacement of PMs, variations to the investment process, market conditions, technological changes, financial environment, etc.) may be altering the distribution that governs those returns. We need to determine a probability that a sequence of returns is consistent with a pre-existing track record, which will inform our decision to re-allocate or possibly redeem our investment. Generally stated, *at what point do we have information sufficient to assess whether a sequence of observations significantly departs from the original distribution?*

This question can be reformulated in the following manner:

1. *Assumption*: Suppose that the returns of a PM are drawn from a time-invariant process, such as a process that is independent and identically distributed (IID).
2. *Data*: We are given
   - (a) A sequence of returns, $\{r_t\}$, for $t = 1, \ldots, T$ (testing set).
   - (b) A reference distribution, based on a sample of returns available prior to $t = 1$ (training set) or some prior knowledge
3. *Goal*: We would like to determine the probability at t that the cumulative return up to t is consistent with that reference distribution.

A first possible solution could entail carrying out a generic Kolmogorov–Smirnov test in order to determine the distance between the reference (or track) and post-track distributions. Being a non-parametric test, this approach has the drawback that it might require impractically large data sets for both distributions.



Moments	Origin	Mean
1	1.10E-02	0
2	3.95E-04	2.74E-04
3	2.53E-06	-7.85E-06
4	4.31E-07	5.63E-07
5	-7.48E-09	-3.28E-08

Figure 6. Moments from the ex-ante distribution.

A second possible solution would be to run a structural break test, in order to determine at what observation t the observations are no longer being drawn from the reference distribution and are coming from a different process instead. Standard structural break tests include CUSUM, Chow, Hartley, etc. However, a divergence from the reference distribution is not necessarily the result of a structural break or breaks. In our experience, a PM's style evolves slowly over time, by gradually transitioning from one set of strategies to another, in an attempt to adapt better to the investment environment—just as a species adapts to a new environment in order to maximize its chances of survival. As the new set of strategies emerge and become more prominent, the old set of strategies does not cease to exist. Therefore, there may not be a clean structural break that these tests could identify.

We propose a faster, more robust and less computationally intensive approach. The method consists of: (i) applying the EF3M for matching the track record's moments, (ii) simulating path scenarios consistent with the matched moments, (iii) deriving a distribution of scenarios based on that match and (iv) evaluate what percentile of the distribution corresponds with the PM's recent performance. Note that there is nothing in the EF3M algorithm that takes any time structure into account, which might be present in the reference and/or target data sets. The reason is, the PM's returns are assumed to be drawn from a time-invariant process, such as a process that is IID, which is the standard assumption used by capital allocation methodologies. If the process is not time-invariant and as a result the post-track process significantly diverges from the track process, it is the goal of this approach to bring that situation to the attention of the portfolio oversight officer.

An important feature of EF3M is its ability to estimate a *distribution* of the possible mixture parameters of our data using information on the reliable moments. Step 2 simulates a path scenario for each output and step 3 uses this distribution on mixture parameters to get a cumulative distribution of returns at a given horizon t. Thus, at time t we can ask what percentile a given cumulative return corresponds to, relative to a collection of simulations corresponding to all of the outputs of the EF3M algorithm (step 4). The results allow us to determine difference percentiles associated with each drawdown and each time under the water.

4.1. *Estimating the distribution of mixture's parameters*

Our procedure starts using the EF3M algorithm to search for parameters that give mixtures whose moments closely match those of the track record's non-Normal distribution. For this particular exercise, the first variant of the EF3M algorithm will be applied, though the second variant would work equally well-given reliable information about the fifth moment. Using the first four moments given in figure 6, we have run the EF3M algorithm 100 000 times and obtained a distribution of parameter estimates for a mixture of two Gaussians. Figure 7 displays the moments' estimation errors $(\tilde{E}[r^i] - E[r^i])$ and average fitted parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$.

4.2. *Simulating performance paths on the parameters' distribution*

Let's define the cumulative return from $t-h$ to t, denoted $R_{t,h}$, as

$$R_{t,h} = \prod_{i=t-h+1}^{t} (1 + r_i), \quad h = 1, \ldots, t \tag{1}$$

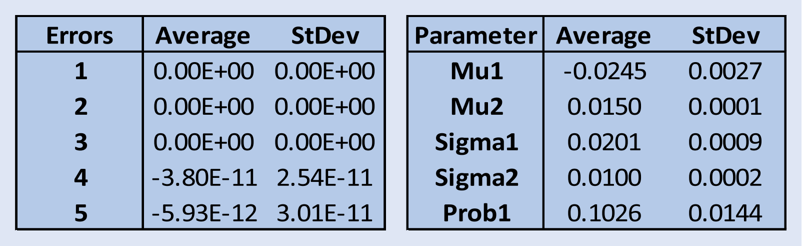
By making $h = t$, we are computing each cumulative return looking back the full available post-track sample (an increasing window). We would like to simulate paths of cumulative returns $(R_{t,t})$ consistent with the observed moments of simple (non-cumulative) returns (r_i). A Monte Carlo simulation of $R_{t,t}$ can be computed by making random draws of r_i. But which of the mixture solutions should we use? One option is to pick one of the $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$, e.g. the mode of the five-dimensional distribution of parameter estimates computed earlier. The problem with that option is that there are several valid combinations of parameters, some more likely than others. Figure 8 plots the *pdf* for a mixture that delivers the same moments as stated in figure 6. We cannot however postulate any particular parameter values to characterize the true ex-ante distribution, as there are multiple combinations able to deliver the observed moments.

A better approach consists in *running one Monte Carlo path $R_{t,t}$ for each of the 100 000 solutions estimated earlier*. Note that by associating an entire Monte Carlo path to the output from each run of our EF3M algorithm, we are implicitly giving higher weight to some outputs than others. This is due to the fact the outputs occur with different multiplicities. Outputs with high multiplicity are more heavily weighted than low multiplicity outputs. For example, the output corresponding to $\mu_1 = -2.03$ in figure 3 would occur over 1400 times and thus would be weighted heavily in the aggregated data about the Monte Carlo simulations corresponding to the moments given in Section 3.2.

4.3. *The distrubution of cumulative returns*

For observed return $\{r_t : 1 \leq t \leq T\}$ we can compare each $R_{t,h}$ with the expected returns predicted by our simulations. More precisely, the simulation we described gives us an approximation to the cumulative distribution $CDF_t{:}\mathbb{R} \to [0, 1]$, where $CDF_t(x)$ is the probability that the return on the portfolio from our original distribution D is less than or equal to x. By collecting our 100 000 simulated $R_{t,t}$ for a given t, we can derive an approximation to its CDF_t.† The cumulative distribution functions are consistent with the observed moments on simple returns and incorporate information about a variety of likely mixtures. The next step is to determine the different percentiles associated with each drawdown level and time under the water. Figure 9 plots various percentiles for each CDF_t. For example, with a 99% confidence, drawdowns of more than 5% from any

†This (non-Normal) CDF_t is on the cumulative returns, not the simple returns.



Errors	Average	StDev
1	0.00E+00	0.00E+00
2	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00
4	-3.80E-11	2.54E-11
5	-5.93E-12	3.01E-11

Parameter	Average	StDev
Mu1	-0.0245	0.0027
Mu2	0.0150	0.0001
Sigma1	0.0201	0.0009
Sigma2	0.0100	0.0002
Prob1	0.1026	0.0144

Figure 7. Moments estimation errors (left) and estimated parameters (right) after 100 000 runs of EF3M.

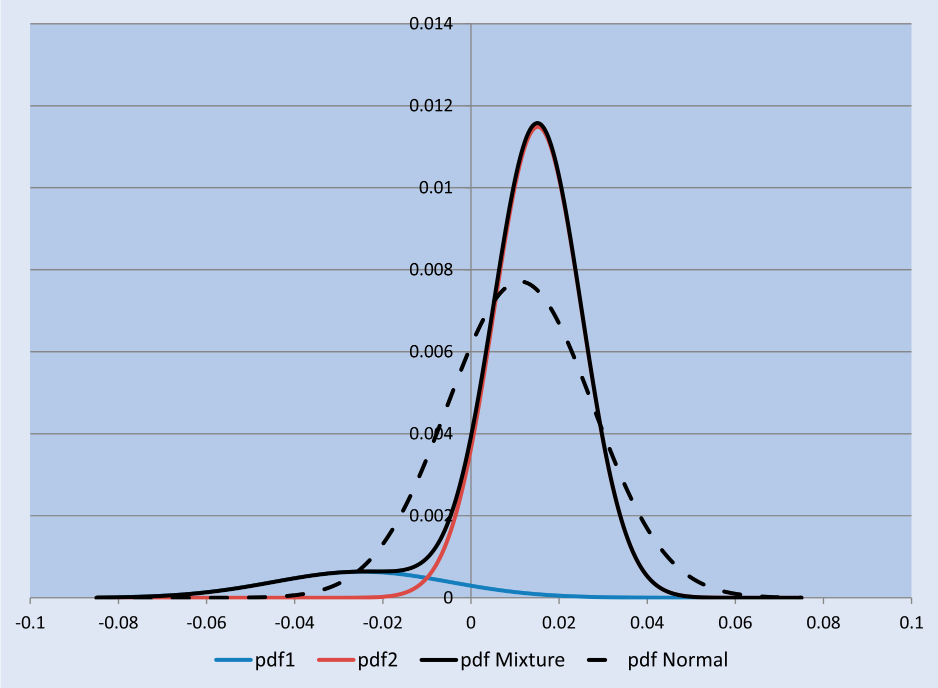


Figure 8. Example of mixture of two Gaussians consistent with the above moments $(\mu_1, \mu_2, \sigma_1, \sigma_2, p) = (-0.025, 0.015, 0.02, 0.01, 0.1)$.

given point after six observations would not be consistent with the ex-ante distribution of track record returns. Furthermore, even if the loss does not reach 5%, a time under the water beyond one year is highly unlikely (2.5% probability), thus it should alert the investor regarding the possibility that the track record's moments (and its Sharpe ratio in particular) are inconsistent with the current performance.

4.4. Probability of divergence

Finally, we are in a position to define the *probability of divergence*, PD_t, updated with every new observation, as

$$PD_t(R_{t,t}) = 2\left|CDF_t(R_{t,t}) - \frac{1}{2}\right| \tag{2}$$

We interpret this number as follows: At time t, $R_{t,t}$ is the total cumulative rate of return from observation 1 to t. Applying CDF_t to the number $R_{t,t}$ give us (our best approximation to) the percentile rank of $R_{t,t}$. In particular if $R_{t,t}$ is exactly the median predicted return, $PD_t(R_{t,t}) = 0$. More generally, if $PD_t(R_{t,t}) = \alpha$, then $R_{t,t}$ is either in the bottom or top $1 - \alpha$ proportion of the predicted returns, depending on the sign of $CDF_t(R_{t,t}) - \frac{1}{2}$. Viewed in this light, $PD_t(R_{t,t})$ measures the proportional departure from the median of our simulated returns.

Figure 10 plots 1000 returns generated from a mixture of Gaussians with moments matching those in figure 6, namely $(\mu_1, \mu_2, \sigma_1, \sigma_2, p) = (-0.025, 0.015, 0.02, 0.01, 0.1)$. PD may sporadically reach high levels, without becoming extreme permanently. What would happen if draws from the first Gaussian become more likely? For example, if $p = 0.2$ instead of $p = 0.1$, the mixture's distribution would become more negatively skewed and fat-tailed. As figure 11 evidences, that situation is distinct from the approved track record and PD slowly but surely converges to 1.

Figure 12 presents an example computed on a sequence of 1000 returns distributed IID Normal that match the mixture's mean and variance, i.e. $N(\mu, \sigma^2) = N(1.10E-02, 2.74E-04)$. PD approaches 1, although the model cannot completely discard the possibility that these returns in fact were drawn from the reference mixture.

Figure 13 presents another example computed on a sequence of 1000 returns distributed IID Normal with a mean half the mixture's and the same variance as the mixture, i.e. $N(\mu, \sigma^2) = N(5.5E-03, 2.74E-04)$. PD

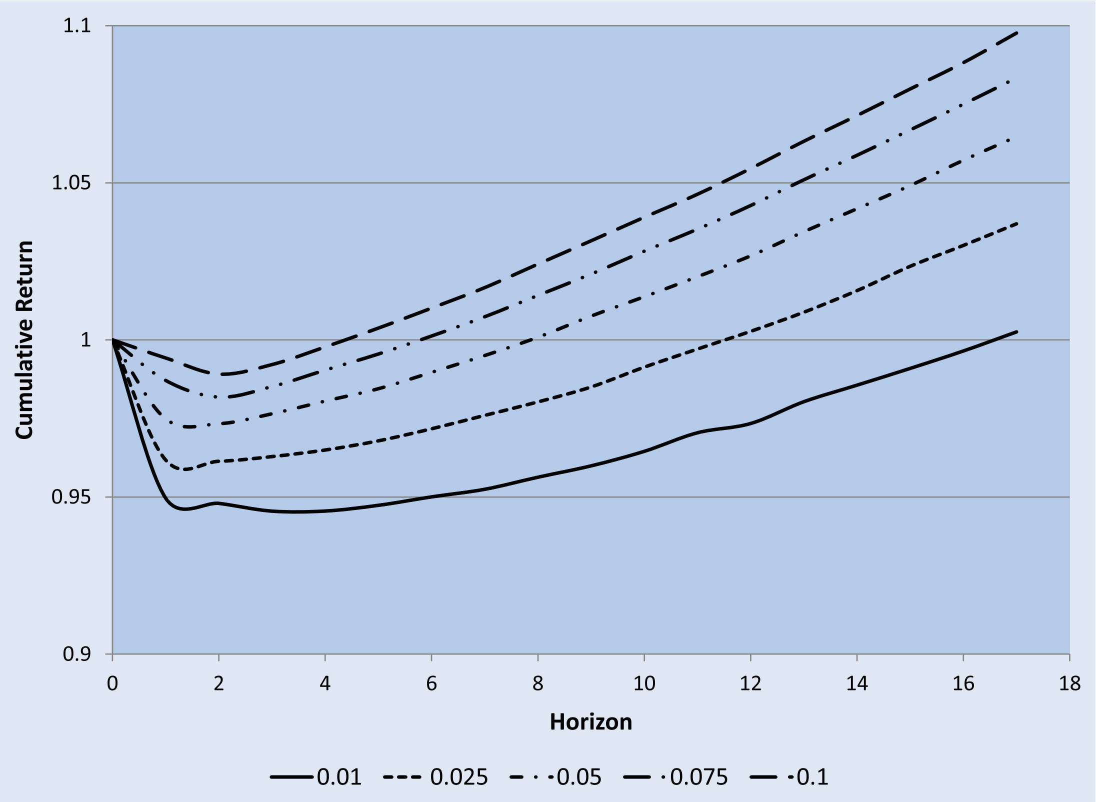


Figure 9. Performance for various confidence bands.

quickly converges to 1, as the model recognizes that those normally distributed draws do not resemble the mixture's simulated paths.

As measured above, an increase in the probability of divergence may not always be triggered by a change in the style, but in the way the style fits to changing market

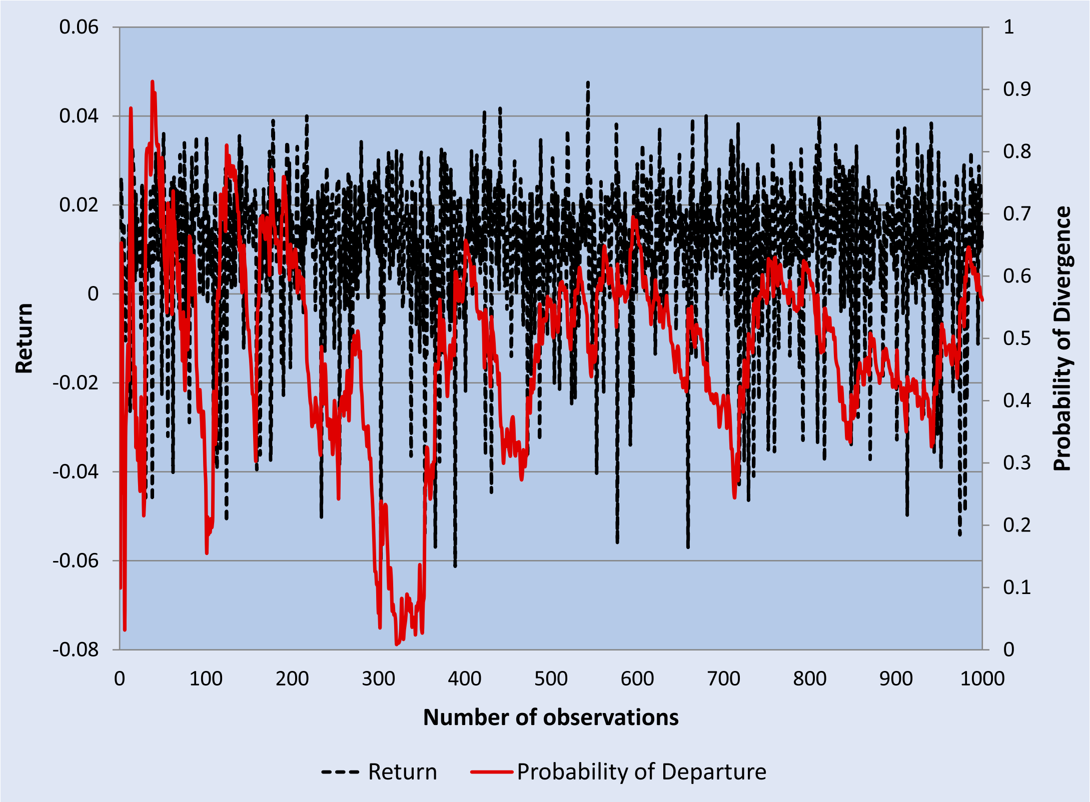


Figure 10. Returns and probability of divergence for draws from $(\mu_1, \mu_2, \sigma_1, \sigma_2, p) = (-0.025, 0.015, 0.02, 0.01, 0.1)$.

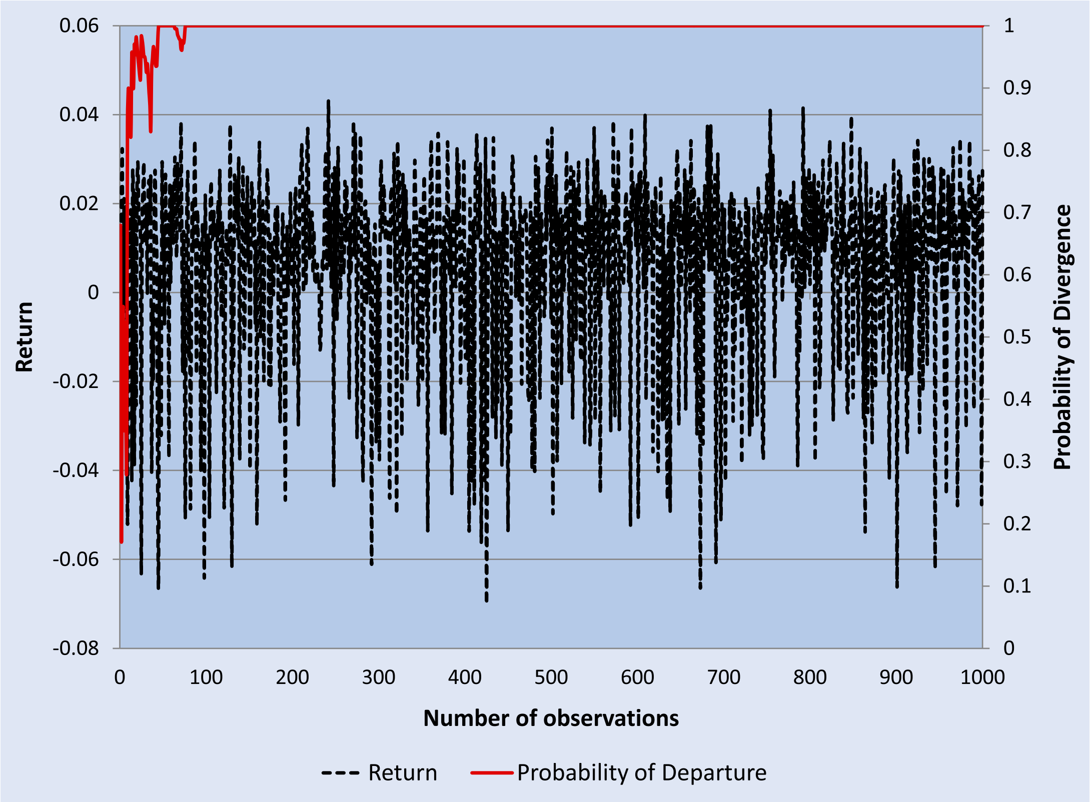


Figure 11. Returns and probability of divergence for draws from $(\mu_1, \mu_2, \sigma_1, \sigma_2, p) = (-0.025, 0.015, 0.02, 0.01, 0.2)$.

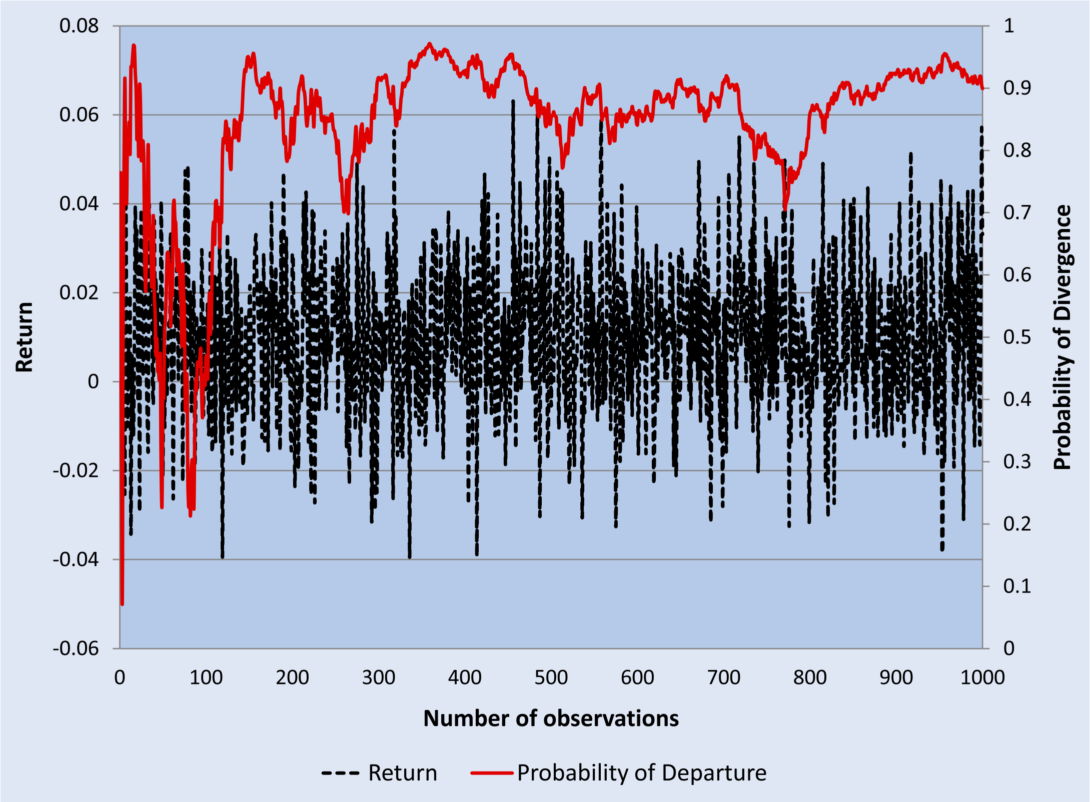


Figure 12. Returns and probability of divergence for $N(\mu, \sigma^2) = N(1.10E-02, 2.74E-04)$.

conditions. That distinction may be more of a philosophical disquisition, because either cause of an increase in the probability of departure (change of style or change of environment) should be brought up to the attention of the portfolio oversight officer and invite a review of the capital allocated to that PM or strategy.

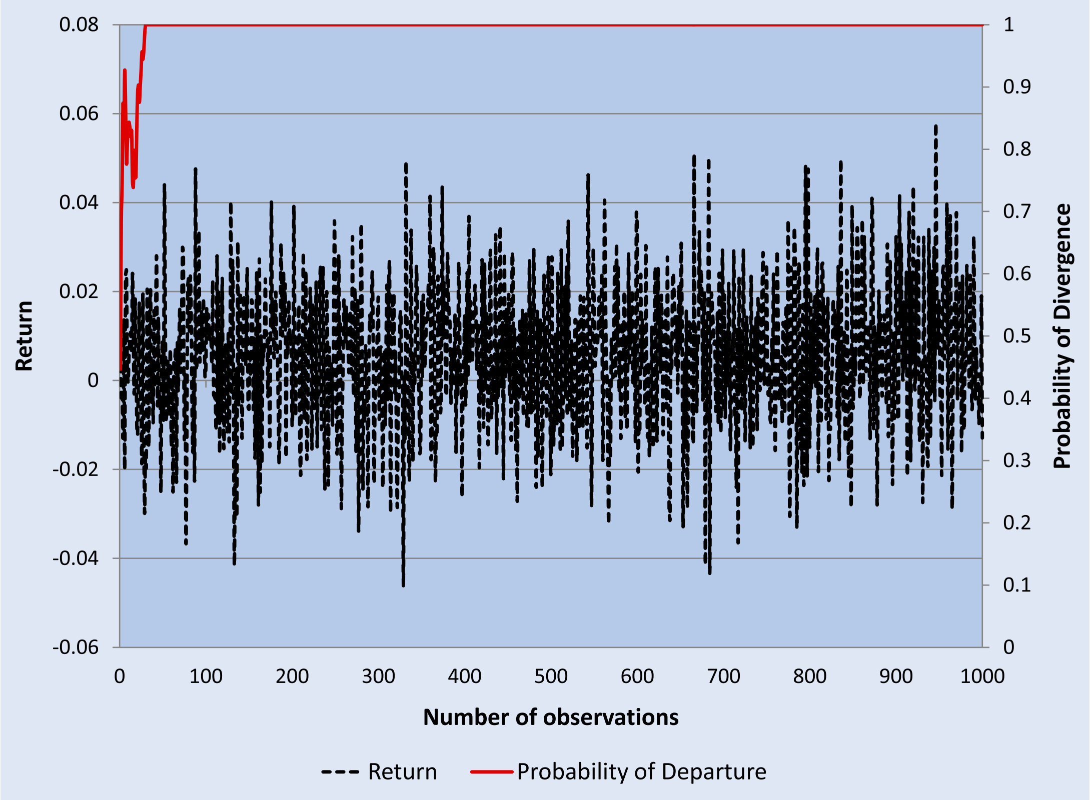


Figure 13. Returns and probability of divergence for $N(\mu, \sigma^2) = N(5.5E-03, 2.74E-04)$.

4.5. Cross-validation[†]

Suppose that the portfolio oversight officer sets a threshold PD^*, above which the probability of departure is deemed to be unacceptably high. Further, suppose that T observations are available out-of-sample (OOS) (i.e. not used in the EF3M estimation of the mixture's parameters) and that $PD_T(R_{T,T}) > PD^*$. Should T be large enough for estimating the five moments with reasonable confidence, it is possible to cross-validate the result that divergence has occurred, following these steps:

(1) We divide the sample of observations into two samples: in-sample (IS) and OOS. We assume that both samples are long enough for providing accurate estimates of five moments.

(2) IS: The training set, used to estimate the set of mixture parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$.

  (a) OOS: The testing set, used to calculate $PD_t(R_{t,t})$, using the fitted parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$.

  (b) Apply the EF3M algorithm OOS, to compute the mixture's parameter estimates on the testing set. We denote these parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)^{\mathrm{OOS}}$, to distinguish them from the set of parameters IS, $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)$.

(3) Using $(\mu_1, \mu_2, \sigma_1, \sigma_2, p)^{\mathrm{OOS}}$, compute $PD_t^{IS}(R_{t,t})$ on the IS data.

(4) Assess whether $PD_t^{IS}(R_{t,t}) > PD^*$. If that is the case, the divergence has been cross-validated. If not, additional evidence may be required, in the form of a longer T.

5. Extensions

A first possible extension of this approach would consist in allowing for any number of constituting distributions, not only two. However, that would require fitting a larger number of higher moments, which we have advised against on theoretical and empirical grounds. Also, if the divergence is caused by two or more new distributions, our PD statistic is expected to detect that situation as well, since it is able to detect the more challenging case of only one emerging style.

A second possible extension would mix multivariate Gaussian distributions. An advantage of doing so would be that we could directly track down which PMs are the source of a fund's divergence, however that would come at the cost of again having to use higher moments to fit the additional parameters. The source of the divergence can still be investigated by running this univariate procedure on subsets of PMs.

A third possible extension would involve modelling mixtures of other parametric distributions beyond the Gaussian case. That is a relatively simple change for the most common functional forms, following the same algebraic strategy presented in the Appendix.

6. Conclusions

In this paper, we have described a method of evaluating the probability that a PM's returns correspond to a reference distribution (denoted *probability of divergence*), which answers a critical concern of portfolio oversight. Our

[†] We are thankful to the referee for suggesting this Section.

method gives investors the ability to assess the representativeness of a PM's track record very early in her post-track observations. It is based on an algorithm for finding the parameters determining a mixture of Gaussians just from the first four or five moments of a given mixture *D*. Determining these moments is the only interaction the algorithm has with the underlying data.

Accordingly, we have devised the EF3M algorithm, which exactly matches the first three moments (on which the researcher usually has greatest confidence), reserving the fourth moment for guiding the convergence of the mixture probability. That the algorithm converges to a solution based on the first four moments is consistent with a theoretical understanding of their meaning. The fourth moment is closely approximated but not exactly matched, because of its sampling error. In a second variant of the EF3M algorithm, we also allow the fifth moment to lead the convergence of the algorithm, should the researcher be confident in that moment's estimate.

The decomposition of a mixture of Gaussians into its component distributions is rarely unique when the only reliable inputs are the first four or five moments. Rather than searching for a unique solution, we advocate computing a distribution of probability for the fitted parameters. This is a Bayesian-like approach, by which we would simulate a large variety of scenarios consistent with the probable values of the mixture's parameters. This approach is made possible, thanks to the relative simplicity (translated into speed) of our EF3M's algorithm. Monte Carlo experiments confirm the validity of our method.

Originally inspired by Galton and Pearson's 'Mathematical Theory of Evolution', mixtures of Gaussians are nowadays widely used in a number of scientific applications. The problem of fitting the characteristic parameters for a mixture of two Gaussians has received a number of solutions over the last 120 years. We have identified several scenarios under which MM, ML and EM algorithms may not fully address the problems faced by many researchers, particularly in the field of Quantitative Finance. MM algorithms present the disadvantage that the solution is impacted by the sample error of the fourth moment. Besides, requiring a fifth moment introduces the problem of basing our solution on a moment for which the researcher typically has no theoretical interpretation. ML and EM algorithms also suffer the criticism of sampling error and theoretical interpretation of the moments used, besides getting trapped on local minima and increased computational intensity as a function of sample size.

An analogue can be made between: (a) the slow pace at which species adapt to an environment, which often results in the emergence of a new distinct species out of a once homogeneous genetic pool and (b) the slow changes that take place over time within a fund, mutating its investment style. A fund's track record provides a sort of genetic marker, which we can use to identify mutations. This has motivated our use of a biometric procedure to detect the emergence of a new investment style within a fund's track record. In doing so, we answer the question: 'What is the probability that a particular PM's performance is departing from the reference distribution used to allocate her capital?' Overall, we believe that EF3M is well suited to answer this critical question.

Acknowledgments

We wish to thank Robert Almgren (Quantitative Brokers, NYU), Tony Anagnostakis (Moore Capital), Marco Avellaneda (NYU), David H. Bailey (Lawrence Berkeley National Laboratory), Sid Browne (Guggenheim Partners), John Campbell (Harvard University), Peter Carr (Morgan Stanley, NYU), David Easley (Cornell University), Ross Garon (SAC Capital), Robert Jarrow (Cornell University), Andrew Karolyi (Cornell University), David Leinweber (Lawrence Berkeley National Laboratory), Attilio Meucci (Kepos Capital, NYU), Maureen O'Hara (Cornell University), Riccardo Rebonato (PIMCO, University of Oxford), Luis Viceira (HBS), and participants at Morgan Stanley's Monthly Quant Seminar for their helpful comments.

References

Alexander, C., Option pricing with normal mixture returns. ISMA Centre Research paper, 2001.

Alexander, C., Normal mixture diffusion with uncertain volatility: Modeling short- and long-term smile effects. *J. Bank. Finance*, 2004, **28**(12), 2957–2980.

Bailey, D. and López de Prado, M., The Sharpe ratio efficient frontier. *J. Risk*, 2012, **15**(2), 3–44.

Bishop, C., *Pattern Recognition and Machine Learning*, 2006 (Springer: New York).

Brigo, D., Mercurio, F. and Sartorelli, G., Lognormal-mixture dynamics under different means. Working paper, UBM, 2002.

Brooks, C. and Kat, H., The statistical properties of hedge fund index returns and their implications for investors. *J. Altern. Invest.*, 2002, **5**(2), 26–44.

Cohen, C., Estimation in mixtures of two normal distributions. *Technometrics*, 1967, **9**(1), 15–28.

Craigmile, P. and Titterington, D., Parameter estimation for finite mixtures of uniform distributions. *Commun. Stat.—Theory Methods*, 1997, **26**(8), 1981–1995.

Darwin, C., *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, 1859 (John Murray: London).

Day, N., Estimating the components of a mixture of two normal distributions. *Biometrika*, 1969, **56**, 463–474.

Dempster, A., Laird, N. and Rubin, D., Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B (Method.)*, 1977, **39**(1), 1–38.

Favre, L. and Galeano, J., Mean-modified value-at-risk optimization with hedge funds. *J. Altern. Invest.*, 2002, **5**(2), 21–25.

Hamilton, J., *Time Series Analysis*, 1994 (Princeton University Press: Princeton, NJ).

Hwang, S. and Satchell, S., Modeling emerging markets risk premia using higher moments. *Int. J. Finance Econ.*, 1999, **4**, 271–296.

Jurcenzko, E. and Maillet, B., The four-moment capital asset pricing model: Some basic results. Working Paper, EDHEC-Risk Institute, 2002.

López de Prado, M., Advances in high frequency strategies. Doctoral dissertation, Complutense University, 2011. Available online at: http://ssrn.com/abstract=2106117

López de Prado, M. and Peijan, A., Measuring loss potential of hedge fund strategies. *J. Altern. Invest.*, 2004, **7**(1), 7–31. Available online at: http://ssrn.com/abstract=641702

López de Prado, M. and Rodrigo, C., *Invertiren Hedge Funds*, 2004 (Díaz de Santos: Madrid).

Maddala, G. and Kim, I., *Unit Roots, Cointegration, and Structural Change*, 1999 (Cambridge University Press: Cambridge).

Makov, U., Smith, A. and Titterington, D., *The Statistical Analysis of Finite Mixture Models*, 1985 (Wiley: Chichester).

McWilliam, N. and Loh, K., Incorporating multidimensional tail-dependencies in the valuation of credit derivatives. Working Paper, Misys Risk Financial Engineering, 2008.

Meucci, A., Annualization and general projection of skewness, kurtosis and all summary statistics, *GARP Risk Professional*, August, 2010, pp. 59–63.

Pearson, K., Contributions to the mathematical theory of evolution. *Philos. Trans. R. Soc.*, 1894, **185**, 71–110.

Rebonato, R. and Cardoso, M., Unconstrained fitting of implied volatility surfaces using a mixture of normal. *J. Risk*, 2004, **7**(1), 55–74.

Tashman, A. and Frey, R., Modelling risk in arbitrage strategies using finite mixtures. *Quant. Finance*, 2008, **9**(5), 495–503.

Wang, J., Generating daily changes in market variables using a multivariate mixture of normal distributions. Proceedings of the 33rd winter conference on simulation, IEEE Computer Society, 2001, pp. 283–289.

Xu, L. and Jordan, M., On convergence properties of the EM algorithm for Gaussian mixtures. *Neural Comput.*, 1996, **8**, 129–151.

Appendices

A.1. Higher moments of a mixture of *m* normal distributions

Let z be a random variable distributed as a standard normal, $z \sim N(0,1)$. Then, $\eta = \mu + \sigma z \sim N(\mu, \sigma^2)$, with *characteristic function*:

$$\vartheta_\eta(s) = E[e^{is\eta}] = E[e^{is\mu}]E[e^{is\sigma z}] = e^{is\eta}\underbrace{\vartheta_z(s\sigma)}_{e^{-\frac{1}{2}s^2\sigma^2}} = e^{is\mu-\frac{1}{2}s^2\sigma^2} \tag{3}$$

Let r be a random variable distributed as a mixture of m normal distributions, $r \sim D(\mu_1, \ldots, \mu_m, \sigma_1, \ldots, \sigma_m, p_1, \ldots, p_m)$, with $\sum_{j=1}^m p_j = 1$. Then:

$$\vartheta_r(s) = E[e^{isr}] = e^{\sum_{j=1}^m p_j\left(is\mu_j - \frac{1}{2}s^2\sigma_j^2\right)} \tag{4}$$

The k-th moment centred about zero of any random variable x can be computed as:

$$E[x^k] = \frac{\left|\frac{\partial^k \vartheta_x(s)}{\partial s^k}\right|_{s=0}}{i^k} \tag{5}$$

We can use the characteristic function to compute the first five moments about the origin (or centred about zero) in the case of a mixture of m Gaussians as:

$$E[r] = \sum_{j=1}^m p_j\mu_j \tag{6}$$

$$E[r^2] = \sum_{j=1}^m p_j(\sigma_j^2 + \mu_j^2) \tag{7}$$

$$E[r^3] = \sum_{j=1}^m p_j(3\sigma_j^2\mu_j + \mu_j^3) \tag{8}$$

$$E[r^4] = \sum_{j=1}^m p_j(3\sigma_j^4 + 6\sigma_j^2\mu_j^2 + \mu_j^4) \tag{9}$$

$$E[r^5] = \sum_{j=1}^m p_j(15\sigma_j^4\mu_j + 10\sigma_j^2\mu_j^3 + \mu_j^5) \tag{10}$$

A.1.1. *From moments about zero to moments about the mean*

We can use the first moments about the origin (equations (6)–(10)) together with Newton's binomium (equation 11) to derive the moments about the mean (equations (12)–(16)):

$$E[(r - E[r])^k] = \sum_{j=0}^k (-1)^j \binom{k}{j} (E[r])^j E[r^{k-j}] \tag{11}$$

$$E[r - E[r]] = 0 \tag{12}$$

$$E[(r - E[r])^2] = E[r^2] - (E[r])^2 \tag{13}$$

$$E[(r - E[r])^3] = E[r^3] - 3E[r^2]E[r] + 2(E[r])^3 \tag{14}$$

$$E[(r - E[r])^4] = E[r^4] - 4E[r^3]E[r] + 6E[r^2](E[r])^2 - 3(E[r])^4 \tag{15}$$

$$\begin{aligned} E[(r - E[r])^5] = {} & E[r^5] - 5E[r^4]E[r] + 10E[r^3](E[r])^2 \\ & - 10E[r^2](E[r])^3 + 4(E[r])^5 \end{aligned} \tag{16}$$

A.1.2. *From moments about the mean to moments about zero*

We have computed the moments about the mean from the moments about the origin. Using equations (12)–(16), the reverse transformation can be carried out easily:

$$\begin{aligned} E[r^k] = {} & (-1)^{k+1}(E[r])^k + \sum_{j=1}^{k-1}(-1)^{k-j+1}\binom{k}{j}(E[r])^{k-j}E[r^j] \\ & + E[(r - E[r])^k] \end{aligned} \tag{17}$$

$$E[r^2] = E[(r - E[r])^2] + (E[r])^2 \tag{18}$$

$$E[r^3] = E[(r - E[r])^3] + 3E[r^2]E[r] - 2(E[r])^3 \tag{19}$$

$$E[r^4] = E[(r - E[r])^4] + 4E[r^3]E[r] - 6E[r^2](E[r])^2 + 3(E[r])^4 \tag{20}$$

$$E[r^5] = E[(r - E[r])^5] + 5E[r^4]E[r] - 10E[r^3](E[r])^2 + 10E[r^2](E[r])^3 - 4(E[r])^5 \tag{21}$$

A.2. EF3M convergence using the 4th moment

For a given μ_2, we would like to find the $(\mu_1, \sigma_1, \sigma_2)$ that match the observed $(\tilde{E}[r], \tilde{E}[r^2], \tilde{E}[r^3])$, with p_1 approximating $\tilde{E}[r^4]$.

Let $p = p_1 = 1 - p_2$. With knowledge of the first four non-centred moments of the mixture, $\tilde{E}[r]$, $\tilde{E}[r^2]$, $\tilde{E}[r^3]$, $\tilde{E}[r^4]$, we can define the following relations among the mixture's parameters. If the known moments are centred, the non-centred moments can be readily computed from equations (10)–(14).

From equation (6), we insert our observation to derive

$$\mu_1 = \frac{\tilde{E}[r] - (1 - p)\mu_2}{p} \tag{22}$$

Likewise, from equation (7) we obtain

$$\sigma_1^2 = \frac{\tilde{E}[r^2] - \sigma_2^2 - \mu_2^2}{p} + \sigma_2^2 + \mu_2^2 - \mu_1^2 \tag{23}$$

Inserting equations (23) in (8) leads to

$$\sigma_2^2 = \frac{\tilde{E}[r^3] + 2p\mu_1^3 + (p-1)\mu_2^3 - 3\mu_1(\tilde{E}[r^2] + \mu_2^2(p-1))}{3(1-p)(\mu_2 - \mu_1)} \tag{24}$$

For a seed (μ_2, p), these relations give us the $\mu_1, \sigma_1, \sigma_2$ that match the first three moments. An algorithm can then be created to approximate (without exactly matching) the fourth moment by re-estimating p. For that, we need a new relationship, which can be derived from equation (9)

$$p = \frac{\tilde{E}[r^4] - 3\sigma_2^4 - 6\sigma_2^2\mu_2^2 - \mu_2^4}{3(\sigma_1^4 - \sigma_2^4) + 6(\sigma_1^2\mu_1^2 - \sigma_2^2\mu_2^2) + \mu_1^4 - \mu_2^4} \tag{25}$$

Because we do not have any relationship to re-estimate μ_2, that parameter remains fixed through every iteration of the algorithm. A fifth moment would be needed to allow for μ_2's convergence, as described in the next section.

A.3. EF3M convergence using the 5th moment

We will start by using some of the relationships identified earlier. In particular, for an initial (μ_2, p)

$$\mu_1 = \frac{\tilde{E}[r] - (1 - p)\mu_2}{p}$$

$$\sigma_1^2 = \frac{\tilde{E}[r^2] - \sigma_2^2 - \mu_2^2}{p} + \sigma_2^2 + \mu_2^2 - \mu_1^2 \tag{26}$$

$$\sigma_2^2 = \frac{\tilde{E}[r^3] + 2p\mu_1^3 + (p-1)\mu_2^3 - 3\mu_1\left(\tilde{E}[r^2] + \mu_2^2(p-1)\right)}{3(1-p)(\mu_2 - \mu_1)}$$

These are the same relations that exactly match the first three moments. Now we can use the fourth moment to re-estimate μ_2, thus allowing it to converge. From equation (9),

$$\mu_2 = \left[-3\sigma_2^2 \pm \left(6\sigma_2^4 + \frac{\tilde{E}[r^4] - p(3\sigma_1^4 + 6\sigma_1^2\mu_1^2 + \mu_1^4)}{1 - p}\right)^{1/2}\right]^{1/2} \tag{27}$$

but we only need to evaluate the '+' from '±', because $\sigma_2^2 > 0$.

Equation (10) allows us to use the fifth moment to lead p's convergence,

$$p = \frac{\tilde{E}[r^5] - b}{a - b} \tag{28}$$

with

$$a = 15\sigma_1^4\mu_1 + 10\sigma_1^2\mu_1^3 + \mu_1^5$$

$$b = 15\sigma_2^4\mu_2 + 10\sigma_2^2\mu_2^3 + \mu_2^5 \tag{29}$$

Unlike in the previous case, this solution incorporates a relationship to re-estimate μ_2 in each iteration. We are still matching the first three moments, with the difference that now moments fourth and fifth drive the convergence of our initial seeds, (μ_2, p).

A.4. EF3M implementation in python

Both variants of the EF3M algorithm are implemented in the following code. For the first variant, comment the line *parameters = iter5(mu2,p1,self.moments)* and leave uncommented the line *parameters = iter4(mu2,p1,self.moments)*. Do the reverse for the second variant.

```
#!/usr/bin/env python
# EF3M class for the exact fit of a mixture of two Gaussians
# On 20120217 by MLdP <lopezdeprado@lbl.gov>

import random

#-------------------------------------------
# Define the problem here
moments=[0.7,2.6,0.4,25,-59.8] # about the origin
epsilon=10**-5
factor=5 # this is the ‘lambda’ referred in the paper

#-------------------------------------------
# This is the mixture’s class
class M2N:
    def __init__(self,moments):
        self.moments=moments
        self.parameters=[0 for i in range(5)]
        self.error=sum([moments[i]**2 for i in range(len(moments))])

    def fit(self,mu2,epsilon):
        p1=random.random()
        numIter=0
        while True:
            numIter+=1
            try:
                #parameters=iter4(mu2,p1,self.moments) # for the first variant
                parameters=iter5(mu2,p1,self.moments) # for the second variant
            except:
                return
            moments=get_moments(parameters)
            error=sum([(self.moments[i]-moments[i])**2 for i in range(len(moments))])
            if error<self.error:
                self.parameters=parameters
                self.error=error
            if abs(p1-parameters[4])<epsilon:return
            if numIter>1/epsilon:return
            p1=parameters[4]
            mu2=parameters[1] # for the 5th moment’s convergence

#-------------------------------------------
# Derive the mixture’s moments from its parameters
def get_moments(parameters):
    m1=parameters[4]*parameters[0]+(1-parameters[4])*parameters[1]
    m2=parameters[4]*(parameters[2]**2+parameters[0]**2)+(1-parameters[4])* \
        (parameters[3]**2+parameters[1]**2)
    m3=parameters[4]*(3*parameters[2]**2*parameters[0]+parameters[0]**3)+ \
        (1-parameters[4])*(3*parameters[3]**2*parameters[1]+parameters[1]**3)
    m4=parameters[4]*(3*parameters[2]**4+6*parameters[2]**2*parameters[0]**2+ \
        parameters[0]**4)+(1-parameters[4])*(3*parameters[3]**4+6*parameters[3]**2* \
        parameters[1]**2+parameters[1]**4)
    m5=parameters[4]*(15*parameters[2]**4*parameters[0]+10*parameters[2]**2* \
        parameters[0]**3+parameters[0]**5)+(1-parameters[4])*(15*parameters[3]**4* \
        parameters[1]+10*parameters[3]**2*parameters[1]**3+parameters[1]**5)
    return [m1,m2,m3,m4,m5]
```

```
#--------------------------------------------
# Equations for variant 1
def iter4(mu2,p1,moments):
    mu1=(moments[0]-(1-p1)*mu2)/p1
    sigma2=((moments[2]+2*p1*mu1**3+(p1-1)*mu2**3-3*mu1*(moments[1]+mu2**2* \
        (p1-1)))/(3*(1-p1)*(mu2-mu1)))**(.5)
    sigma1=((moments[1]-sigma2**2-mu2**2)/p1+sigma2**2+mu2**2-mu1**2)**(.5)
    p1=(moments[3]-3*sigma2**4-6*sigma2**2*mu2**2-mu2**4)/(3*(sigma1**4-sigma2**4)+ \
        6*(sigma1**2*mu1**2-sigma2**2*mu2**2)+mu1**4-mu2**4)
    return [mu1,mu2,sigma1,sigma2,p1]

#--------------------------------------------
# Equations for variant 2
def iter5(mu2,p1,moments):
    mu1=(moments[0]-(1-p1)*mu2)/p1
    sigma2=((moments[2]+2*p1*mu1**3+(p1-1)*mu2**3-3*mu1*(moments[1]+mu2**2* \
        (p1-1)))/(3*(1-p1)*(mu2-mu1)))**(.5)
    sigma1=((moments[1]-sigma2**2-mu2**2)/p1+sigma2**2+mu2**2-mu1**2)**(.5)
    a=(6*sigma2**4+(moments[3]-p1*(3*sigma1**4+6*sigma1**2*mu1**2+mu1**4))/ \
        (1-p1))**.5
    mu2=(a-3*sigma2**2)**.5
    a=15*sigma1**4*mu1+10*sigma1**2*mu1**3+mu1**5
    b=15*sigma2**4*mu2+10*sigma2**2*mu2**3+mu2**5
    p1=(moments[4]-b)/(a-b)
    return [mu1,mu2,sigma1,sigma2,p1]

#--------------------------------------------
# Number of combinations of n over k
def binomialCoeff(n, k):
    if k<0 or k>n:return 0
    if k>n-k:k=n-k
    c = 1
    for i in range(k):
        c=c*(n-(k-(i+1)))
        c=c//(i+1)
    return c

#--------------------------------------------
# Compute moments about the mean (or centered) from moments about the origin
def centeredMoment(moments,order):
    moment_c=0
    for j in range(order+1):
        comb=binomialCoeff(order,j)
        if j==order:
            a=1
        else:
            a=moments[order-j-1]
        moment_c+=(-1)**j*comb*moments[0]**j*a
    return moment_c

#--------------------------------------------
# Main function
def main():
    stDev=centeredMoment(moments,2)**.5
    mu2=[float(i)*epsilon*factor*stDev+moments[0] for i in range(1,int(1/epsilon))]
    m2n=M2N(moments)
    err_min=m2n.error
    for i in mu2:
        m2n.fit(i,epsilon)
        if m2n.error<err_min:
            print m2n.parameters, m2n.error
            err_min=m2n.error

#--------------------------------------------
# Boilerplate
if __name__=='__main__': main()
```

A.5. Fitting a mixture to an observed variance and kurtosis

Let r be a random variable distributed as a mixture of 2 Normal distributions, $r \sim D(\mu_1, \mu_2, \sigma_1, \sigma_2, p_1, p_2)$, with $p_1 + p_2 = 1$ and ith population moment about the origin $E[r^i]$. Given some observed moments $\tilde{E}[r^2], \tilde{E}[r^4]$, we would like to estimate the symmetric mixture ($E[r^3] = 0$) centred about zero ($E[r] = 0$) such that its population moments match $E[r^2] = \tilde{E}[r^2]$ and $E[r^4] = \tilde{E}[r^4]$.

Rewrite $p_2 = 1 - p_1$. We have five free parameters $(\mu_1, \mu_2, \sigma_1, \sigma_2, p_1)$ to match only four moments $(E[r] = 0,\ E[r^2] = \tilde{E}[r^2],\ E[r^3] = 0,\ E[r^4] = \tilde{E}[r^4])$. A mixture of two Gaussians has mean $E[r] = p_1\mu_1 + (1 - p_1)\mu_2$ and a third moment about the origin $E[r^3] = p_1(3\sigma_1^2\mu_1 + \mu_1^3) + (1 - p_1)(3\sigma_2^2\mu_2 + \mu_2^3)$. Thus, $\mu_1 = \mu_2 = 0$ meets our requirement that $E[r] = E[r^3] = 0$. We still have three free parameters $(\sigma_1, \sigma_2, p_1)$ to match the two remaining moments $(E[r^2] = \tilde{E}[r^2],\ E[r^4] = \tilde{E}[r^4])$.

From equations (7) and (9), this particular problem reduces to the system

$$\tilde{E}[r^2] = p_1\sigma_1^2 + (1 - p_1)\sigma_2^2$$

$$\frac{1}{3}\tilde{E}[r^4] = p_1\sigma_1^4 + (1 - p_1)\sigma_2^4 \tag{30}$$

We find solutions in

$$\sigma_2^2 = \tilde{E}[r^2] \pm \left(\frac{p_1}{(1 - p_1)}\left(\frac{1}{3}\tilde{E}[r^4] - (\tilde{E}[r^2])^2\right)\right)^{1/2} \tag{31}$$

$$\sigma_1^2 = \frac{1}{p_1}(\tilde{E}[r^2] - (1 - p_1)\sigma_2^2)$$

but because the system is symmetric in σ_1 and σ_2, it suffices to evaluate the '+' in '±'. In that way, $\sigma_2 \geq \sigma_1$. A pending question is, what is the appropriate value for p_1? From the above equations we find that, in order to find roots in the real domain, an additional condition is

$$\tilde{E}[r^2] > (1 - p_1)\sigma_2^2 \tag{32}$$

which, after replacing σ_2^2 with its solution, leads to

$$p_1 > 1 - \frac{3(\tilde{E}[r^2])^2}{\tilde{E}[r^4]} \tag{33}$$

and obviously $\frac{\tilde{E}[r^4]}{(\tilde{E}[r^2])^2} \geq 3$, so that $0 \leq p_1 \leq 1$. Putting all pieces together, for any $0 < \delta < 1$, a feasible solution is given by

$$p_1 = 1 + (\delta - 1)\frac{3(\tilde{E}[r^2])^2}{\tilde{E}[r^4]}$$

$$\sigma_2^2 = \tilde{E}[r^2] \pm \left(\frac{p_1}{(1 - p_1)}\left(\frac{1}{3}\tilde{E}[r^4] - (\tilde{E}[r^2])^2\right)\right)^{1/2} \tag{34}$$

$$\sigma_1^2 = \frac{1}{p_1}(\tilde{E}[r^2] - (1 - p_1)\sigma_2^2)$$