

Correcting statistical models via empirical distribution functions

Alexander Munteanu¹ · Max Wornowizki²

Received: 3 October 2014 / Accepted: 2 July 2015 / Published online: 25 July 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract We consider the two-sample homogeneity problem where the information contained in two samples is used to test the equality of the underlying distributions. In cases where one sample is simulated by a procedure modelling the data generating process of another observed sample, a mere rejection of the null hypothesis is unsatisfactory. Instead, the data analyst would like to know how the simulation can be improved. Based on the popular Kolmogorov–Smirnov test and a general mixture model, we propose an algorithm that determines an appropriate correction distribution function. Complementing the simulation sample by a given proportion of observations sampled from this distribution reduces the Kolmogorov–Smirnov distance between the modified and the observed sample. Therefore, the correction distribution indicates possible improvements to the current simulation process. We prove our algorithm to run in linear time when applied to sorted samples. We further illustrate its intuitive results on simulated as well as on real data sets from astrophysics and bioinformatics.

Keywords Nonparametric mixture model · Kolmogorov–Smirnov test · Quantify discrepancy between data and model · Efficient algorithm

✉ Max Wornowizki
wornowizki@statistik.tu-dortmund.de
Alexander Munteanu
alexander.munteanu@tu-dortmund.de

¹ Department of Computer Science, Technische Universität Dortmund, 44221 Dortmund, Germany

² Department of Statistics, Technische Universität Dortmund, 44221 Dortmund, Germany

1 Introduction

Our work focuses on developing a new statistical method to assess the quality of simulations and to provide useful information that can help to improve simulation procedures. Since often arbitrary amounts of simulated data can be generated, the algorithm is mainly designed for applications in different domains of research dealing with medium to large data sets, but it is also applicable for small sample sizes. In order to give a clear motivation to the problem under study, we consider the gamma ray detectors MAGIC-I and MAGIC-II. These telescopes are located at the Roque de los Muchachos on the Canary Island La Palma. The interested reader is referred to [Cortina et al. \(2009\)](#) and [The MAGIC Collaboration \(2014\)](#) for detailed information. The telescopes make use of a mirror surface of over 200 square metres to measure atmospheric signals induced by the interaction of high energetic photons, called gamma rays, with the atmosphere. Since they do not have an electric charge, gamma rays do not interact with magnetic fields and thus provide useful information about their sources which the physicists want to explore. However, these are not the only particles generating such atmospheric signals. For each gamma ray in the measurements there are about 1000 observations of so called background events, which are not of interest in the given context. The background events mainly consist of protons, but also contain heavier hadrons and electrons. Classification algorithms based on characteristics of the measured signals could be applied in order to distinguish between the background and the gamma particles. Unfortunately, these methods cannot be trained on real data, because it is not labelled. Therefore, simulation procedures for gamma rays as well as for protons have been constructed based on models of particle propagation and improved in several steps. The main program that is used to generate such simulations is CORSIKA ([Heck et al. 1998](#)).

Clearly, it is of major importance to compare the simulated proton samples with the observed data, which is the aim of our algorithm. On the one hand, suitable artificial background data is crucial for the classification analysis. Hence, variables with low agreement of generated and real data must be identified. On the other hand, small deviations between simulations and real data can be caused by gamma ray signals. If one assumes to have a reasonable simulation, variables with comparably high discrepancies can be quite helpful in the upcoming classification task.

A typical statistical approach to check the similarity of two data sets is the application of a nonparametric two-sample test like for example the two-sample Kolmogorov–Smirnov test. However, a mere rejection of the null hypothesis is not satisfying in practice. If the simulations are seriously inadequate, the data analyst wants to quantify the issues by identifying the regions with too many or insufficiently many observations, respectively. Such information can then be used to update CORSIKA or related methods using more suitable hyperparameters or even give rise to the inclusion of additional simulation steps in the particle simulation. If the discrepancies between the samples potentially stem from the gamma ray signals, their quantification is necessary as well, allowing to assess and validate the gamma ray simulations.

In the present work, we develop a novel approach allowing to gain additional insight into the discrepancies between an observed and a simulated sample based on the two-sample Kolmogorov–Smirnov test. Note that while the application aims at *designing*

a reasonable simulation procedure, our contribution helps to *improve* an existing simulation procedure which is based on prior domain specific knowledge. We therefore assume that such a simulation procedure exists a priori. In order to compare the data sets, we make use of a mixture model linking the distributions of the observed and the simulated samples by a third distribution. The latter represents all discrepancies between the first two and will be called *correction distribution* hereafter. We propose an algorithm that determines an empirical distribution function of this correction distribution and a mixing proportion in the mixture model. Both are determined such that the resulting mixture with the simulated data does not lead to another rejection by the Kolmogorov–Smirnov test when compared to the observed data. Note that the algorithm in principle does not aim at statistical testing and thus the type I and type II error after modification of the simulated sample are not investigated. The method rather makes use of the quantiles of the Kolmogorov–Smirnov distribution to obtain intuitive bounds on the distance between empirical distribution functions. The algorithm does not construct a mixture fitting the observed data perfectly, but leads to a reasonably close approximation taking the sample variance into account. The amount of closeness can be regulated by the critical value K_α or equivalently by the significance level α and may be adjusted for a given application. For the sake of brevity, we formulate the problem focusing on the improvement of a simulation procedure in the following. However, the method can also be applied to approximate the distribution of subgroups in the data as argued for the gamma ray signals above.

There already exist various semi- and nonparametric suggestions on mixture models in the literature. They focus on the estimation of the densities and the number of components in a mixture model of a given single sample. In this situation the authors often have to work in a multidimensional setting that allows to impose more information about the structure. As shown by [Hall et al. \(2005\)](#), the quantities in a reasonable mixture model with two nonparametric components are not identifiable for one- and two-dimensional problems even under certain independence assumptions. The methods often rely on adjusted versions of the EM algorithm ([Pilla and Lindsay 2001](#)) or a Newton method ([Wang 2010](#); [Schellhase and Kauermann 2012](#)) and in addition can make use of appropriate data transformations ([Hettmansperger and Thomas 2000](#)). There also exist several nonparametric approaches to problems involving multiple samples and finite mixture models, as for example proposed by [Kolossiatis et al. \(2013\)](#). However, to the authors' knowledge, there is no literature addressing the two sample problem outlined above in the context of mixture models.

We close this gap by proposing a correction of one sample to resemble another sample based on the corresponding empirical distribution functions. This allows us to derive simple, monotone and asymptotically distribution free confidence bands. Working with nonparametric density estimators may seem more intuitive, especially since many of them (e.g. [Schellhase and Kauermann 2012](#)) attain the appealing form of convex combinations of suitable basis functions:

$$\hat{f}(x) = \sum_{i=1}^b a_i f_i(x).$$

The natural idea to solve our problem in this setting is forcing the density approximation of the simulated sample to be part of the basis for the estimation on the observed sample. One thus first estimates the density on the simulated sample. In a second step this estimator \hat{g} is fixed as one of the basis functions, say $f_1 = \hat{g}$, for the estimation of the density on the observed sample. The main problem with this approach is that in general one cannot guarantee that the corresponding coefficient a_1 properly reflects the importance of the simulated data. Often a_1 will be way too small if the remaining basis functions are not chosen appropriately, because these also contribute to the region of values modelled by f_1 . In our setting this corresponds to discarding the simulation almost completely, which is not desirable. Since choosing the remaining basis functions in an adequate way is nontrivial, straightforward application of density based approaches are not satisfactory in our application. In addition, the optimisation of the coefficients and potentially the basis functions themselves can be computationally costly, especially in the medium to large sample cases we aim for.

The remainder of this paper is structured as follows: In Sect. 2 we formalise the problem and propose a mixture model related to the nonparametric two-sample problem. We introduce several desirable properties of the model parameters and formulate two optimisation problems allowing to identify them. In Sect. 3 we present an algorithm to solve the problems introduced in the second section and provide intuitive explanations of the main ideas of each step of our algorithm. The proofs of correctness and linear running time are conducted in Sect. 4. In Sect. 5 our procedure is applied to real and simulated data and the results are illustrated. Section 6 concludes with a summary and an outlook on possible future work.

2 Problem definition

In this section we introduce our basic notations and consider a general mixture model based on nonparametric estimators. Within this model, all discrepancies between the distributions of the observed and the simulated data, respectively, are represented by a third distribution. Our task will be to assess this so called *correction distribution* in order to identify the aforementioned discrepancies. To this end, the model is transferred to an empirical equivalent. Thereafter, several constraints on parameters of the empirical model are motivated allowing to identify them properly.

We observe the sample $x_1, \dots, x_{n_1} \in \mathbb{R}$ stemming from an unknown continuous distribution F . The underlying data generating process is modelled by a simulation procedure represented by the distribution G . To evaluate the quality of the simulation, consider n_2 simulated observations y_1, \dots, y_{n_2} drawn independently from G . If the simulation procedure works well, G resembles F and thus the samples are similar.

To test the equality of F and G using the two-sample Kolmogorov–Smirnov test denote the empirical distribution functions of the samples by F_e and G_e , respectively, and let $N = \frac{n_1 \cdot n_2}{n_1 + n_2}$. Setting $M = \mathbb{R}$ the null hypothesis $H_0 : F = G$ is rejected if the test statistic

$$D_M(F_e, G_e) = \sqrt{N} \sup_{x \in M} |F_e(x) - G_e(x)|$$

exceeds an appropriately chosen critical value K_α .

In order to consider the procedure from a different perspective, we define an upper boundary function U setting $U(x) = \min(1, F_e(x) + \frac{K_\alpha}{\sqrt{N}})$ for all $x \in M$, and in analogy define a lower boundary function L by $L(x) = \max(0, F_e(x) - \frac{K_\alpha}{\sqrt{N}})$. With these definitions the Kolmogorov–Smirnov test does not reject H_0 if and only if G_e is an element of the set

$$B = \{f : \mathbb{R} \rightarrow [0, 1] \mid \forall x \in M : L(x) \leq f(x) \leq U(x)\}$$

called the confidence band. As argued above, we are interested in the regions of undersampling respectively oversampling, i.e., the regions where G_e violates L or U .

To model the problem described above, we work with the fairly general two-component mixture model

$$F = \tilde{s} \cdot G + (1 - \tilde{s}) \cdot H,$$

where the so-called mixture proportion or shrinkage factor \tilde{s} measures the degree of agreement of F and G while the distribution H represents all dissimilarities between F and G . Since F is fully described by G , \tilde{s} and H , we are interested in identifying \tilde{s} and H , because these quantities contain all relevant information for an appropriate modification of G .

It is clear that the choice $\tilde{s} = 0$ and $H = F$ solves the above equation. However, this solution is not of interest in our setting, because the data analyst wants to correct and not to discard the current simulation, which is often based on expert knowledge. This may give more insight into the data generating process itself and is thus preferable. In the other extreme case, $\tilde{s} = 1$, the simulation is correct and H is irrelevant. However, for any $\tilde{s} \in (0, 1)$ the corresponding H is unique and demixing F , that is estimating \tilde{s} and H , provides useful information for improving the simulation.

Since the distributions F and G are not available in practice, we replace the corresponding distribution functions by the standard empirical estimators F_e and G_e . Combining the Kolmogorov–Smirnov distance with the above mixture model, we propose to identify a shrinkage factor $s \in [0, 1]$ and a correction function \mathcal{H} such that the function

$$\mathcal{F} = s \cdot G_e + (1 - s) \cdot \mathcal{H} \tag{1}$$

lies in the confidence band B and thus the Kolmogorov–Smirnov test would not reject H_0 if the distribution functions G_e and \mathcal{F} were compared. Since \mathcal{H} is a substitute for H , it should be a distribution function and therefore lie in the set

$$\mathcal{M} = \left\{ f : \mathbb{R} \rightarrow [0, 1] \mid f \text{ monotone, } \lim_{x \rightarrow -\infty} f(x) = 0 \right\}$$

that is a superset of the set of all distribution functions on \mathbb{R} . Obviously, neither s nor \mathcal{H} are unique in this situation. Hence, in the following we set some additional constraints and describe the problem in more detail allowing us to determine reasonable solutions.

Since we work with empirical distribution functions, all derived quantities are characterized by their values on the joint sample $x_1, \dots, x_{n_1}, y_1, \dots, y_{n_2}$. Therefore, instead of considering all functions $\mathcal{H} \in \mathcal{M}$, we restrict ourselves to those which may be discontinuous only on $Z = \{z_1, \dots, z_{n_1+n_2}\}$ consisting of the ordered joint sample. We denote this set of functions by $\mathcal{M}^* \subset \mathcal{M}$. This restriction is not very strong regardless of the sample sizes, since it does not make a difference for the Kolmogorov–Smirnov distance whether we stick to the original data or add observations on intermediate values. In addition, the choice of the value for an added observation would be arbitrary between two of the given data points with respect to our distance. Thus, we focus on the original observations which has the additional advantage of lower computational cost. Also, the sample sizes in applications we aim for are comparably large so that the restricted set is often quite dense.

Motivated by the fact that the data analyst is interested in making as little changes as possible concerning the current simulation, we can make the mixture proportion s in the model identifiable by choosing s maximally such that the mixture \mathcal{F} fits the observed data. This directly implies a minimal weight $1 - s$ for the correction function \mathcal{H} . We thus formulate **Problem 1**:

$$\begin{aligned} \max_{s \in [0,1]} & : && s \\ \text{s.t.} & : && \exists \mathcal{H} \in \mathcal{M}^* : s \cdot G_e + (1 - s) \cdot \mathcal{H} \in B. \end{aligned}$$

Note that for $s^* = \frac{K_\alpha}{\sqrt{N}}$ and $\mathcal{H}^* = \frac{1}{1-s^*} \cdot L$ the property $s^* \cdot G_e + (1 - s^*) \cdot \mathcal{H}^* \in B$ holds. Thus, the optimal value of s , called s_{opt} in the following, is always greater than 0. Hence, the simulated data is always properly included in the mixture.

After Problem 1 is solved, the resulting mixture $\mathcal{F} = s_{opt} \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H}$ lies in B . Since this does not imply the property $\lim_{x \rightarrow \infty} \mathcal{F}(x) = 1$, the function \mathcal{H} could be an improper distribution function. Therefore, there might exist several choices of \mathcal{H} solving Problem 1 given s_{opt} . However, consider the pointwise minimal function \mathcal{H}_{min} defined by $\forall z \in Z : \mathcal{H}_{min}(z) = \min \mathcal{H}(z)$. Hereby, the minimization is taken over the non-empty compact set of all functions $\mathcal{H} \in \mathcal{M}^*$ satisfying $\mathcal{F}_{min} = s_{opt} \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H} \in B$. The function \mathcal{H}_{min} is clearly unique. To find a reasonable distribution function \mathcal{H} , we propose to first identify and then enlarge \mathcal{H}_{min} , that is to construct \mathcal{H} such that $\mathcal{H} \geq \mathcal{H}_{min}$, in an adequate way, so that the final mixture is a proper empirical distribution function lying in B .

Before we formulate this enlargement as an optimisation problem, we want to point out that, quite intuitively, \mathcal{H}_{min} should not be enlarged for small $z \in Z$. In particular, if \mathcal{F}_{min} intersects the upper boundary U , adding mass before the maximal value $z \in Z$ where $\mathcal{F}_{min}(z)$ is equal to $U(z)$, that is, $z_{meq} = \max_{z \in Z} \{z | \mathcal{F}_{min}(z) = U(z)\}$, leads to violations of U in z_{meq} . Note that in case of such an intersection, the global Kolmogorov–Smirnov distance on $M = \mathbb{R}$ between the final mixture and F_e will be the radius of the confidence band, regardless of the enlargement of \mathcal{H}_{min} . However, on subsets of \mathbb{R} the distance can be improved if \mathcal{H}_{min} is enlarged appropriately. Hence, we propose to identify z_{norm} , the smallest value after z_{meq} such that adding mass after z_{norm} minimises the Kolmogorov–Smirnov distance restricted to the set $M_{norm} = \{z \in Z | z \geq z_{norm}\}$. We then add the probability mass in such a way that the minimal

distance $D_{\{z \geq z_{norm}\}}$ is attained. If there is no intersection between $\mathcal{F}_{min}(z)$ and $U(z)$, we set $z_{meq} = \min \{Z\}$ and proceed in the same way. Using the notations introduced above, finding a suitable distribution function \mathcal{H} for a given value of s_{opt} can be formalised in **Problem 2**:

$$\begin{aligned} \min_{\mathcal{H} \in \mathcal{M}^*} & : D_{M_{norm}}(\mathcal{F}, F_e) \\ \text{s.t.} & : \mathcal{F} = s_{opt} \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H} \in B \\ & \mathcal{H} \geq \mathcal{H}_{min} \\ & \lim_{x \rightarrow \infty} \mathcal{H}(x) = 1. \end{aligned}$$

An optimal solution to Problem 2 is called \mathcal{H}_{opt} . The corresponding final mixture is denoted by

$$\mathcal{F}_{opt} = s_{opt} \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H}_{opt}. \tag{2}$$

Note that, even with these constraints, the solution to the problem of identifying $\mathcal{F} \in B$ may not be unique. Although the shrinkage factor s_{opt} is unique by its maximality property, there may be several optimal enlargements of \mathcal{H}_{min} equally appropriate in the sense of the restricted Kolmogorov–Smirnov distance.

3 The algorithm

In this section we propose an algorithm for solving Problems 1 and 2 introduced in Sect. 2. At first, the main procedure is described. All subsequent subroutines called within the main algorithm are explained in more detail hereafter. Pseudocode is provided in order to illustrate the algorithm and its subroutines.

3.1 The main algorithm

Algorithm 1 is our main procedure to solve Problems 1 and 2. It requires two sorted sample vectors $x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$ and a significance level α . At first, it calculates the empirical distribution functions F_e and G_e of the samples and determines the critical value K_α at level α . In fact, K_α is the α -quantile of the distribution of $K = \sup_{t \in [0,1]} B(t)$, where $B(t)$ is a Brownian bridge (Durbin 1973). For the typical significance levels $\alpha_1 = 0.05$ and $\alpha_2 = 0.01$ the critical values are $K_{\alpha_1} = 1.358$ and $K_{\alpha_2} = 1.628$, respectively. The values s and \mathcal{F} , candidates for the shrinkage factor s_{opt} and the final mixture \mathcal{F}_{opt} , are initialised and the lower bound for performing a binary search is set to s^* (cf. the description of Problem 1). The upper and lower boundary functions of the confidence band around F_e, U respectively L , are computed next. These steps can be considered as preprocessing and are carried out in the lines 1 and 2. The two-sample Kolmogorov–Smirnov test does not reject the null hypothesis of equal distributions if the relation $L \leq G_e \leq U$ holds. In this case the empirical distribution functions resemble each other well enough and the algorithm stops in line 4. If the test rejects the null hypothesis, the algorithm carries out certain steps to

determine an optimal mixture within the confidence band. To solve Problem 1, the following operations are applied iteratively in the main loop in lines 5–11: a candidate \mathcal{F} lying above the upper boundary somewhere has to be shrunk, that is, multiplied by a factor from the interval $(0, 1)$, in order to correct the violation of U . This problem is addressed in line 7 in the so called *Shrink-Down* algorithm. On the contrary, a candidate falling below the lower boundary L must receive additional probability mass in appropriate regions. This is taken into account in line 9 by calling the *Push-Up* algorithm. The two operations are applied whenever necessary in the presented order. However, since they have opposite effects, some data situations require multiple executions of the Shrink-Down and the Push-Up step. Iteration of these steps generates a decreasing sequence of upper bounds to s_{opt} . The well-known binary search technique embedded in the demixing algorithm in line 10 takes another approach by bounding s_{opt} from below and above. It is connected with the Shrink-Down and Push-Up step by using the current shrinkage factor s learned from them as an upper bound to s_{opt} . In return, the binary search updates s and \mathcal{F} , which are then passed to the Shrink-Down and Push-Up steps. The lower bound for the optimal shrinkage factor, l_b , is updated by the binary search itself.

Algorithm 1: Demixing-Algorithm

Input : Sorted observations $x \in \mathbb{R}^{n_1}$, $y \in \mathbb{R}^{n_2}$, significance level α
Output: Optimal shrinkage factor s_{opt} ,
 optimal correcting function $\mathcal{H}_{opt} \in \mathcal{M}^*$

- 1 $Z \leftarrow (x, y)$; $K_\alpha \leftarrow K(\alpha)$; $N \leftarrow \frac{n_1 \cdot n_2}{n_1 + n_2}$; $l_b \leftarrow \frac{K_\alpha}{\sqrt{N}}$; $s \leftarrow 1$;
- 2 $F_e \leftarrow \text{EmpDistrFun}(x)$; $G_e \leftarrow \text{EmpDistrFun}(y)$; $\mathcal{F} \leftarrow G_e$;
 $L \leftarrow \max \left\{ 0, G_e - \frac{K_\alpha}{\sqrt{N}} \right\}$; $U \leftarrow \min \left\{ 1, G_e + \frac{K_\alpha}{\sqrt{N}} \right\}$;
- 3 **if** $\forall z \in Z : L(z) \leq \mathcal{F}(z) \leq U(z)$ **then**
- 4 **return** $(s, 0)$
- 5 **repeat**
- 6 **if** $\exists z \in Z : \mathcal{F}(z) > U(z)$ **then**
- 7 $(s, \mathcal{F}) \leftarrow \text{Shrink-Down}(s, \mathcal{F})$;
- 8 **if** $\exists z \in Z : \mathcal{F}(z) < L(z)$ **then**
- 9 $(s, \mathcal{F}) \leftarrow \text{Push-Up}(s, \mathcal{F})$;
- 10 $(l_b, s, \mathcal{F}) \leftarrow \text{BinSearch}(l_b, s, \mathcal{F})$;
- 11 **until** $\forall z \in Z : L(z) \leq \mathcal{F}(z) \leq U(z)$;
- 12 $\mathcal{F} \leftarrow \text{Normalise}(\mathcal{F})$;
- 13 $\mathcal{H} \leftarrow (\mathcal{F} - s \cdot G_e) / (1 - s)$;
- 14 **return** (s, \mathcal{H}) ;

Once the main loop is terminated, the optimal shrinkage factor s_{opt} and the corresponding minimal correction function \mathcal{H}_{min} introduced on page 6 are determined and thus Problem 1 is solved. The normalisation step in line 12 takes care of Problem 2 returning an optimal mixture \mathcal{F}_{opt} . This allows to identify a reasonable correction function \mathcal{H}_{opt} in line 13 by rearranging Eq. (2), which is returned afterwards together with the optimal shrinkage factor s_{opt} .

In the remainder of this section the subroutines of the main algorithm are described in detail.

3.2 The Shrink-Down algorithm

This procedure is applied whenever a candidate \mathcal{F} exceeds the upper boundary U at some point. Following the mixture model (1), it is intuitive to solve this problem by computing the maximal shrinkage value $s_d \in (0, 1)$ such that $s_d \cdot \mathcal{F}$ does not violate U any more. In other words, \mathcal{F} is shrunk down. The maximal shrinkage factor to achieve this is $s_d = \min_{z \in Z} \{ \frac{U(z)}{\mathcal{F}(z)} \}$, where we set $\frac{a}{0} = \infty$ for every $a > 0$. The Shrink-Down subroutine presented in Algorithm 1 calculates this factor in line 1. Then, the total shrinkage and the candidate function \mathcal{F} are updated accordingly and are eventually returned.

Algorithm 2: Shrink-Down

Input : Current values of \mathcal{F} and s
Output: Updated values \mathcal{F} and s

- 1 $s_d \leftarrow \min_{z \in Z} \left\{ \frac{U(z)}{\mathcal{F}(z)} \right\};$
- 2 $s \leftarrow s_d \cdot s;$
- 3 $\mathcal{F} \leftarrow s_d \cdot \mathcal{F};$
- 4 **return** $(s, \mathcal{F});$

3.3 The Push-Up algorithm

The Push-Up step presented in Algorithm 3 is carried out whenever the current candidate \mathcal{F} violates the lower boundary of the confidence band, L . In order to increase the values of the mixture in the problematic regions, probability mass must be added there. Note that \mathcal{F} may lie below L before the smallest value $z \in Z$ where $\mathcal{F}(z)$ equals $U(z)$, called $z_{eq} = \min_{z \in Z} \{z | U(z) = \mathcal{F}(z)\}$, as well as after that point. However, these two cases have a crucial difference. Adding probability mass before z_{eq} leads to a new violation of the upper boundary U in z_{eq} , while adding mass after z_{eq} does not imply this problem. In order to distinguish between these cases, the algorithm first identifies z_{eq} in line 1. This value z_{eq} equals $\max(Z)$ after initialisation with $\mathcal{F} = G_e$ because $\mathcal{F}(\max(Z)) = G_e(\max(Z)) = 1 = U(\max(Z))$ holds. As we argue later in Lemma 4, z_{eq} is also well defined after modifications of \mathcal{F} .

If there are violations of L before z_{eq} , a shrinkage is necessary. Thus, keeping in mind Problem 1, the maximal shrinkage factor s_u must be identified, so that the residuals to L before z_{eq} do not exceed the residual to U in z_{eq} after shrinking. Otherwise, adding appropriate probability mass will cause a violation of U in z_{eq} . More formally, the shrinkage factor

$$s_u = \max_{s \in [0,1]} \{s \mid \forall z < z_{eq} : L(z) - s \cdot \mathcal{F}(z) \leq U(z_{eq}) - s \cdot \mathcal{F}(z_{eq})\}$$

must be determined. We have $s_u = \min_{z < z_{eq}} \left\{ \frac{\mathcal{F}(z_{eq}) - L(z)}{\mathcal{F}(z_{eq}) - \mathcal{F}(z)} \right\}$ using basic arithmetic transformations of the constraint and $\mathcal{F}(z_{eq}) = U(z_{eq})$. After s_u is determined in line 3, the shrinkage factor s as well as \mathcal{F} are updated.

In order to shift the current candidate \mathcal{F} appropriately, the positive residuals $d(z) = \max\{0, L(z) - \mathcal{F}(z)\}$ to L are computed for all $z \in Z$. These are the minimal amounts which must be added to \mathcal{F} so that the lower boundary L is no longer violated. The residuals d are added to the current correction term $\mathcal{F} - s \cdot G_e$ and the sum is minimally monotonised, cf. line 7. The result, denoted by \mathcal{H} , is added to $s \cdot G_e$ yielding the new candidate mixture \mathcal{F} .

Algorithm 3: Push-Up

Input : Current values of \mathcal{F} and s
Output: Updated values \mathcal{F} and s

```

1  $z_{eq} \leftarrow \min_{z \in Z} \{z \mid U(z) = \mathcal{F}(z)\};$ 
2 if  $\exists z < z_{eq} : \mathcal{F}(z) < L(z)$  then
3    $s_u \leftarrow \min_{z < z_{eq}} \left\{ \frac{\mathcal{F}(z_{eq}) - L(z)}{\mathcal{F}(z_{eq}) - \mathcal{F}(z)} \right\};$ 
4    $s \leftarrow s_u \cdot s;$ 
5    $\mathcal{F} \leftarrow s_u \cdot \mathcal{F};$ 
6  $\forall z \in Z : d(z) \leftarrow \max\{0, L(z) - \mathcal{F}(z)\};$ 
7  $\forall z \in Z : \mathcal{H}(z) \leftarrow \max_{z' \leq z} \{\mathcal{F}(z') - s \cdot G_e(z') + d(z')\};$ 
8  $\mathcal{F} \leftarrow s \cdot G_e + \mathcal{H};$ 
9 return  $(s, \mathcal{F});$ 

```

3.4 The Binary search algorithm

The binary search step presented in Algorithm 4 is called at the end of every iteration in the main loop. Its input consists of l_b and u_b , the current lower respectively upper bound for s_{opt} . While l_b is derived from previous binary search steps, u_b is set to the current value of s . The algorithm computes the average of the given bounds in line 1. Using this candidate, the minimum monotone step function \mathcal{H}_b is computed such that $\mathcal{F}_b = s_b \cdot G_e + \mathcal{H}_b \geq L$ holds, cf. lines 2 and 3. This is done in analogy to the corresponding lines in the Push-Up step.

If \mathcal{F}_b violates the upper boundary U , then, by minimality of \mathcal{H}_b , no monotone step function for the shrinkage factor s_b can exist such that the corresponding mixture lies within the confidence band B . Therefore, as implied by the monotonicity property proved in Lemma 1 below, it holds that $s > s_b > s_{opt}$. Thus, in this case the algorithm updates s to s_b as a new upper bound for s_{opt} and sets the current mixture candidate to \mathcal{F}_b in lines 6 and 7. Otherwise, again by Lemma 1, the relation $s_{opt} \geq s_b > l_b$ must hold, since there exists a monotone step function for the shrinkage factor s_b leading to a mixture in B . Thus, s_b is a better lower bound to s_{opt} so that l_b is updated to s_b , while all other quantities are kept.

Algorithm 4: BinSearch

Input : l_b and u_b , current lower and upper bounds on s_{opt}
Output: Updated values \mathcal{F} , s and l_b

```

1  $s_b \leftarrow (l_b + u_b)/2$ ;
2  $\forall z \in Z : d(z) \leftarrow \max\{0, L(z) - s_b \cdot G_e(z)\}$ ;
3  $\forall z \in Z : \mathcal{H}_b(z) \leftarrow \max_{z' \leq z} \{d(z')\}$ ;
4  $\mathcal{F}_b \leftarrow s_b \cdot G_e(z) + \mathcal{H}_b$ ;
5 if  $\exists z \in Z : \mathcal{F}_b(z) > U(z)$  then
6    $s \leftarrow s_b$ ;
7    $\mathcal{F} \leftarrow \mathcal{F}_b$ ;
8 else
9    $l_b \leftarrow s_b$ ;
10 return  $(l_b, s, \mathcal{F})$ ;
```

3.5 The normalisation step

As we show in Theorem 1 below, Problem 1 is solved when the loop of Algorithm 1 (lines 5–11) stops. At this point, the current value of s is the optimal shrinkage factor s_{opt} , while the current mixture is $\mathcal{F} = s_{opt} \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H}_{min}$ and lies within the confidence band. However, as pointed out in the description of Problem 2, \mathcal{F} may not be a proper distribution function since $\lim_{x \rightarrow \infty} \mathcal{F}(x) < 1$ may hold. This deficiency is corrected by the normalisation step presented in Algorithm 5.

Algorithm 5: Normalise

Input : Current value of \mathcal{F}
Output: Final mixture \mathcal{F}

```

1 if  $\forall z \in Z : \mathcal{F}(z) < U(z)$  then
2    $z_{meq} \leftarrow \min(Z)$ ;
3 else
4    $z_{meq} \leftarrow \max_{z \in Z} \{z \mid U(z) = \mathcal{F}(z)\}$ ;
5 if  $z_{meq} \neq \max(Z)$  then
6    $z_{norm} \leftarrow \min_{z > z_{meq}} \{z \mid \mathcal{F}(z) < F_e(z)\}$ ;
7    $\forall z \geq z_{norm} : d(z) \leftarrow \min\{F_e(z) - \mathcal{F}(z), 1 - \mathcal{F}(\max(Z))\}$ ;
8   if  $\max_{z \geq z_{norm}} \{-d(z)\} \geq 1 - \mathcal{F}(\max(Z))$  then
9      $\bar{z} \leftarrow \max_{z \geq z_{norm}} \{z \mid -d(z) \geq 1 - \mathcal{F}(\max(Z))\}$ ;
10     $z_{norm} \leftarrow \min_{z > \bar{z}} \{z \mid d(z) > 0\}$ ;
11     $\forall z \geq z_{norm} : \mathcal{H}_{norm}(z) \leftarrow \max \left\{ 0, \left( \max_{z_{norm} \leq z' \leq z} \{d(z')\} + \min_{z'' \geq z} \{d(z'')\} \right) / 2 \right\}$ ;
12     $\forall z < z_{norm} : \mathcal{H}_{norm}(z) \leftarrow 0$ ;
13     $\mathcal{F} \leftarrow \mathcal{F} + \mathcal{H}_{norm}$ ;
14 return  $(\mathcal{F})$ ;
```

To check whether \mathcal{F} must be enlarged, the algorithm computes z_{meq} , the maximal value $z \in Z$ where $\mathcal{F}(z)$ equals $U(z)$. When there is no intersection of \mathcal{F} and U , which can happen if the last candidate mixture was proposed by the binary search, the algorithm sets $z_{meq} = \min(Z)$. If $z_{meq} = \max(Z)$ is satisfied, the property $\mathcal{F}(\max(Z)) = \mathcal{F}(z_{meq}) = U(z_{meq}) = U(\max(Z)) = 1$ holds, so no further corrections are necessary and \mathcal{F} is returned.

Otherwise, as stated in the motivation to Problem 2, adding any probability mass before z_{meq} would lead to a violation of U in z_{meq} . Since s_{opt} is already determined, such a violation cannot be repaired by further shrinking as in the Push-Up step. Thus, probability mass has to be added after z_{meq} . In fact, the region where mass should be added can be restricted even further. Therefore, we denote by z_{norm} the smallest value $z \in Z$ such that $z > z_{meq}$ and $\mathcal{F}(z) < F_e(z)$ holds. Since adding mass between z_{meq} and z_{norm} cannot reduce the Kolmogorov–Smirnov distance between F_e and \mathcal{F} , we focus on all $z \geq z_{norm}$ in the following.

Hence, the deviations $d(z) = F_e(z) - \mathcal{F}(z)$ are computed for all $z \geq z_{norm}$ in line 7. Deviations above the remaining mass $1 - \mathcal{F}(\max(Z))$ are decreased to this value. Hereafter, the algorithm checks whether the maximal increase of \mathcal{F} above F_e , the maximum of all negative deviations $-d(z)$, is greater or equal to the imposed maximal decrease of \mathcal{F} below F_e , namely $1 - \mathcal{F}(\max(Z))$. As long as this is the case, adding probability mass will not decrease the Kolmogorov–Smirnov distance. Hence, in line 9 the algorithm determines the last position where the above property holds and updates z_{norm} to be greater than this position. This yields the set $M_{norm} = \{z \in Z | z \geq z_{norm}\}$, where a reduction of the Kolmogorov–Smirnov distance is possible. At the latest, M_{norm} is the last region where \mathcal{F} lies below F_e .

In order to compute a distribution function \mathcal{H}_{norm} , which has to be added to the remaining region M_{norm} , the residuals d are considered on this set. Determining \mathcal{H}_{norm} there may be seen as an L_∞ isotonic regression problem. Since we work in the setting of distribution functions, a monotone function should be constructed, which fits the residuals $d(z)$ best in the sense of the L_∞ -norm. Unweighted isotonic regression problems under the L_∞ norm can be efficiently solved in linear time for sorted samples. This can be achieved by a simple approach, which is referred to as *Basic* by Stout (2012). This method is applied in line 11 of Algorithm 5. For each residual it computes the maximum of all previous values and the minimum of all subsequent values and determines the regression value as the average of these two quantities.

Note that a solution to the isotonic regression may be negative, but the distribution function \mathcal{H}_{opt} must be nonnegative. However, as we will prove in Lemma 5, setting all of its negative values to 0 results in an optimal solution to the isotonic regression problem constraint to nonnegativity. Since no correction is applied before z_{norm} , \mathcal{H}_{norm} is set to 0 before z_{norm} in line 12. Finally, \mathcal{F} is updated and returned.

4 Analysis of the algorithms

In this section theoretical results for the algorithms of Sect. 3 are provided. Among other things, we prove a monotonicity property allowing to apply the binary search technique to Problem 1 and demonstrate that the Shrink-Down and Push-Up step

always lead to upper bounds on s_{opt} proving their correctness. While the first part of this section deals with the correctness of our algorithm, the second one presents its runtime analysis.

We introduce the essential notations used repeatedly in our proofs. The shrinkage factor of G_e , the correction function and the mixture candidate after the k th iteration of the main loop of Algorithm 1 (lines 5–11) are denoted by s_k, \mathcal{H}_k and $\mathcal{F}_k = s_k \cdot G_e + \mathcal{H}_k$, respectively. In order to initialise them, we set $s_0 = 1, \mathcal{H}_0 = 0$ and $\mathcal{F}_0 = G_e$. Let $s_{d,k}$ denote the update of the shrinkage factor determined in the Shrink-Down step in the k th iteration. Whenever this update is not computed, we set $s_{d,k} = 1$. The update of the shrinkage factor determined in the Push-Up step of the k th iteration is called $s_{u,k}$ and treated in the same way.

4.1 Correctness of the algorithm

Our first result, mainly proving the correctness of the binary search step, shows that the property of lying within the confidence band is monotone in s . In other words, for any $s > s_{opt}$ a corresponding mixture violates a boundary of B , while for $s \leq s_{opt}$ it is always possible to find a mixture lying in B .

Lemma 1 $(\exists \mathcal{H} \in \mathcal{M}^* : s \cdot G_e + (1 - s) \cdot \mathcal{H} \in B) \Leftrightarrow s \in [0, s_{opt}]$.

Proof First we recall the definition of Problem 1 from page 6:

$$\begin{aligned} \max_{s \in [0,1]} : & \quad s \\ \text{s.t.} : & \quad \exists \mathcal{H} \in \mathcal{M}^* : \forall z \in Z : L(z) \leq s \cdot G_e(z) + (1 - s) \cdot \mathcal{H}(z) \leq U(z), \end{aligned} \tag{3}$$

where \mathcal{M}^* denotes the set of all nondecreasing, nonnegative step functions varying on Z only and converging to 0 for $x \rightarrow -\infty$.

Now we introduce an alternative characterization of s_{opt} by Problem A:

$$\begin{aligned} \max_{s \in [0,1]} : & \quad s \\ \text{s.t.} : & \quad \forall z \in Z : s \cdot G_e(z) \leq U(z) \tag{4a} \\ & \quad \forall z' < z'' \in Z : L(z') - s \cdot G_e(z') \leq U(z'') - s \cdot G_e(z'') \tag{4b} \end{aligned}$$

Before we proceed with proving the proposition, we show the equivalence of Problem 1 and Problem A. For this sake, choose an arbitrary $s \in [0, 1]$ such that (3) holds. Then for all $z \in Z$ it follows that $s \cdot G_e(z) \leq U(z) - (1 - s) \cdot \mathcal{H}(z) \leq U(z)$ by nonnegativity of $(1 - s) \cdot \mathcal{H}$, which proves that the inequality (4a) holds. Furthermore, choose $z' < z''$ from Z arbitrarily. Then $L(z') - s \cdot G_e(z') \leq (1 - s) \cdot \mathcal{H}(z') \leq (1 - s) \cdot \mathcal{H}(z'') \leq U(z'') - s \cdot G_e(z'')$ follows by monotonicity of \mathcal{H} . Thus (4b) is also respected.

For the other direction let $s \in [0, 1]$ respect constraints (4a) and (4b). From (4a) it is clear that $s \cdot G_e(z)$ never exceeds the upper boundary. From (4b) we know that

correcting any deficiency to the lower boundary L is possible without violating the upper boundary U on subsequent positions. Thus choosing

$$(1 - s) \cdot \mathcal{H}(z) = \max \left\{ 0, \max_{z^* \leq z} \{L(z^*) - s \cdot G_e(z^*)\} \right\}$$

will result in a mixture within the confidence band. This means that (3) holds.

We now make use of the above equivalence of Problem 1 and Problem A to prove the proposition:

$$(\exists \mathcal{H} \in \mathcal{M}^* : s \cdot G_e + (1 - s) \cdot \mathcal{H} \in B) \Leftrightarrow s \in [0, s_{opt}].$$

For $s \in (s_{opt}, 1]$ the property $s \cdot G_e + (1 - s) \cdot \mathcal{H} \notin B$ immediately follows by definition of s_{opt} . So let $s \in [0, s_{opt}]$ be arbitrarily chosen and note that constraints (4a) and (4b) are respected for s_{opt} . From this we deduce that both conditions must also hold for s since $\forall z \in Z : s \cdot G_e(z) \leq s_{opt} \cdot G_e(z) \leq U(z)$ and furthermore for all $z', z'' \in Z$ with $z' < z''$ it follows

$$\begin{aligned} L(z') - s \cdot G_e(z') &= L(z') - s_{opt} \cdot G_e(z') - (s - s_{opt}) \cdot G_e(z') \\ &\leq U(z'') - s_{opt} \cdot G_e(z'') - \underbrace{(s - s_{opt}) \cdot G_e(z')}_{\geq 0} \\ &\leq U(z'') - s_{opt} \cdot G_e(z'') - (s - s_{opt}) \cdot G_e(z'') \\ &= U(z'') - s \cdot G_e(z''). \end{aligned}$$

Hence, $L(z') - s \cdot G_e(z') \leq U(z'') - s \cdot G_e(z'')$ holds.

Since, as argued before, constraints (4a) and (4b) are equivalent to constraint (3), there exists an \mathcal{H} for which the mixture $s \cdot G_e + (1 - s) \cdot \mathcal{H}$ lies in B , which completes the proof. □

In the next lemma, the correction function \mathcal{H}_k computed in the k th iteration of the main loop for the shrinkage factor s_k is considered. We prove that \mathcal{H}_k is indeed the minimal function in \mathcal{M}^* resolving violations of the lower boundary L . This result contributes to the correctness of our construction of \mathcal{H}_{min} and is used in the subsequent proofs.

Lemma 2 \mathcal{H}_k is the pointwise minimal function among all $\mathcal{H} \in \mathcal{M}^*$ satisfying $s_k \cdot G_e + \mathcal{H} \geq L$.

Proof Let $\mathcal{H}_{k,min} \in \mathcal{M}^*$ be the minimal function with the property $s_k \cdot G_e + \mathcal{H}_{k,min} \geq L$. To prove the result we must thus show $\mathcal{H}_k = \mathcal{H}_{k,min}$. Now, the correction function \mathcal{H}_k is either computed in the binary search step or in the Push-Up step. In the first case, the residuals between $s_k \cdot G_e$ and the lower boundary L are determined and then minimally monotonised, cf. lines 2 and 3 of Algorithm 4. This monotonisation is performed by considering the maximum of preceding values and is therefore minimal. Hence, this procedure must yield $\mathcal{H}_{k,min}$. In the remainder of this proof we thus treat the second case, namely the computation of \mathcal{H}_k in the Push-Up step.

Following the lines 6–7 in Algorithm 3, we denote by d_k the positive deficiencies to L after the shrinking in the Push-Up step of iteration k , i.e., $d_k = \max(0, L - s_{d,k} \cdot s_{u,k} \cdot \mathcal{F}_{k-1})$. Setting $\tilde{\mathcal{F}}_k = s_{d,k} \cdot s_{u,k} \cdot \mathcal{F}_{k-1} + d_k$, the correction function \mathcal{H}_k can be expressed as $\mathcal{H}_k = \text{mon}(\tilde{\mathcal{F}}_k - s_k \cdot G_e)$. Thereby, $\text{mon}(f)$ denotes the minimal monotone function such that $\text{mon}(f) \geq f$. This monotonisation is performed analogically to the one in the binary search step by considering the maximum of preceding values. Note that the monotonising operator is itself monotone, that is, $\text{mon}(f_1) \leq \text{mon}(f_2)$ holds if $f_1 \leq f_2$. We show the proposition by induction:

Base case $k = 1$: By assumption, \mathcal{H}_1 is computed in the Push-Up step, so $s_1 = s_{d,1} \cdot s_{u,1}$ holds. In addition, \mathcal{F}_0 is defined by $\mathcal{F}_0 = G_e$. Hence, $d_1 = \max(0, L - s_{d,1} \cdot s_{u,1} \cdot \mathcal{F}_0) = \max(0, L - s_1 \cdot G_e) \leq \mathcal{H}_{1,min}$ must hold, since the last inequality holds by definition of $\mathcal{H}_{1,min}$. Because of $\tilde{\mathcal{F}}_1 = s_1 \cdot G_e + d_1$ we obtain $\mathcal{H}_1 = \text{mon}(\tilde{\mathcal{F}}_1 - s_1 \cdot G_e) = \text{mon}(d_1) \leq \text{mon}(\mathcal{H}_{1,min}) = \mathcal{H}_{1,min}$, where the inequality follows by the monotonicity of the monotonising operator. Thus $\mathcal{H}_1 \leq \mathcal{H}_{1,min}$ is established. To prove the other inequality, note that $\mathcal{H}_1 \in \mathcal{M}^*$ and $\mathcal{H}_1 = \text{mon}(d_1) \geq d_1$. Hence, $\mathcal{H}_{1,min} \leq \mathcal{H}_1$ follows by the definition of $\mathcal{H}_{1,min}$. Altogether, we get $\mathcal{H}_{1,min} = \mathcal{H}_1$.

Inductive step $k - 1 \Rightarrow k$: The shrink updates $s_{d,k}$ and $s_{u,k}$ are bounded by 1 by construction and thus the inequality $s_k \leq s_{d,k} \cdot s_{u,k} \cdot s_{k-1} \leq s_{k-1}$ holds. Hence, the shrinkage factor s_k does not increase in k and therefore the corresponding minimal correction function $\mathcal{H}_{k,min}$ does not decrease in k . Consequently, we get $\mathcal{H}_{k,min} \geq \mathcal{H}_{k-1,min} \geq s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1,min}$. The correctness of the $(k - 1)$ -th step assumed by the induction principle yields $\mathcal{H}_{k-1,min} = \mathcal{H}_{k-1}$ resulting in $\mathcal{H}_{k,min} \geq s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1}$. Now, rewriting $s_{d,k} \cdot s_{u,k} \cdot \mathcal{F}_{k-1}$ to $s_k \cdot G_e + s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1}$ allows to interpret d_k as the minimal function which must be added to $s_k \cdot G_e + s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1}$ so that the lower boundary L of the confidence band is not violated any more. Together with $\mathcal{H}_{k,min} \geq s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1}$ established above this implies $s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1} + d_k \leq \mathcal{H}_{k,min}$. Since in addition d_k is by construction minimally chosen such that $\tilde{\mathcal{F}}_k = s_{d,k} \cdot s_{u,k} \cdot \mathcal{F}_{k-1} + d_k \geq L$ holds, we deduce

$$L - s_k \cdot G_e \leq \tilde{\mathcal{F}}_k - s_k \cdot G_e = s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1} + d_k \leq \mathcal{H}_{k,min}.$$

Applying the monotonising operator and exploiting its monotonicity this implies

$$\begin{aligned} L - s_k \cdot G_e &\leq \text{mon}(L - s_k \cdot G_e) \\ &\leq \underbrace{\text{mon}(s_{d,k} \cdot s_{u,k} \cdot \mathcal{H}_{k-1} + d_k)}_{=\mathcal{H}_k} \\ &\leq \text{mon}(\mathcal{H}_{k,min}) \\ &= \mathcal{H}_{k,min}, \end{aligned} \tag{5}$$

and therefore $\mathcal{H}_k \leq \mathcal{H}_{k,min}$. To prove $\mathcal{H}_k \geq \mathcal{H}_{k,min}$, note that \mathcal{H}_k is a function in \mathcal{M}^* . The inequalities (5) imply $L \leq s_k \cdot G_e + \mathcal{H}_k$. So, by definition of $\mathcal{H}_{k,min}$, $\mathcal{H}_k \geq \mathcal{H}_{k,min}$ follows and thus overall $\mathcal{H}_k = \mathcal{H}_{k,min}$ holds. \square

The next result shows that the Shrink-Down step always leads to overall shrinkage factors not lower than s_{opt} and therefore may be used as an improved upper bound for s_{opt} in the binary search procedure.

Lemma 3 *If $s_k > s_{opt}$ then $s_{d,k+1} \cdot s_k \geq s_{opt}$.*

Proof The proposition is trivial for $s_{d,k+1} = 1$ so in the following $s_{d,k+1} < 1$ is assumed. This means that the $(k + 1)$ th Shrink-Down step is not skipped but executed. So \mathcal{F}_k must lie above the upper boundary U for some values.

Together with the definition of $s_{d,k+1}$, this ensures the existence of a $z_{eq} \in Z$ such that $s_{d,k+1} \cdot \mathcal{F}_k(z_{eq}) = U(z_{eq})$ holds. In the following we consider the two possible cases for the correction function $\mathcal{H}_k = \mathcal{F}_k - s_k \cdot G_e$:

Case $\mathcal{H}_k(z_{eq}) = 0$: Using the definition of z_{eq} and \mathcal{F}_k , we deduce

$$\begin{aligned} U(z_{eq}) &= s_{d,k+1} \cdot \mathcal{F}_k(z_{eq}) \\ &= s_{d,k+1} \cdot (s_k \cdot G_e(z_{eq}) + \mathcal{H}_k(z_{eq})) \\ &= s_{d,k+1} \cdot s_k \cdot G_e(z_{eq}) \\ &< G_e(z_{eq}), \end{aligned}$$

where the last inequality follows since $0 < s_{d,k+1} < 1$, $0 < s_k \leq 1$ and $0 < G_e(z_{eq})$. The latter is satisfied, because otherwise $0 = G_e(z_{eq})$ and $\mathcal{H}_k(z_{eq}) = 0$ immediately imply $0 = U(z_{eq})$, which is a contradiction to the positivity of U .

The calculations above show that the function G_e lies above the upper boundary U in z_{eq} before any shrinking. However, the first Shrink-Down step would solve this problem and because of $\mathcal{H}_k(z_{eq}) = 0$ there cannot be a new violation of U in z_{eq} in subsequent steps. Hence, $k = 0$ and consequently $s_k = 1$ must hold. We have that $s_k \cdot s_{d,k+1} = s_{d,1} \geq s_{opt}$ holds in this case, since $s_{d,1}$ is by construction the maximal shrinkage factor for avoiding violations of U before adding any correction function.

Case $\mathcal{H}_k(z_{eq}) > 0$: Let $\tilde{\mathcal{H}} \in \mathcal{M}^*$ be the minimal function one must add to $s_{d,k+1} \cdot s_k \cdot G_e$ in order to correct violations of the lower boundary L . Due to $s_{d,k+1} \leq 1$ we get $s_{d,k+1} \cdot s_k \cdot G_e \leq s_k \cdot G_e$ and thus $\tilde{\mathcal{H}} \geq \mathcal{H}_k$ holds by minimality of \mathcal{H}_k shown in Lemma 2. This allows to prove

$$\begin{aligned} U(z_{eq}) &= s_{d,k+1} \cdot \mathcal{F}_k(z_{eq}) \\ &= s_{d,k+1} \cdot (s_k \cdot G_e(z_{eq}) + \mathcal{H}_k(z_{eq})) \\ &< s_{d,k+1} \cdot s_k \cdot G_e(z_{eq}) + \mathcal{H}_k(z_{eq}) \\ &\leq s_{d,k+1} \cdot s_k \cdot G_e(z_{eq}) + \tilde{\mathcal{H}}(z_{eq}). \end{aligned}$$

Thus, $s_{d,k+1} \cdot s_k \cdot G_e + \tilde{\mathcal{H}}$ violates the upper boundary of the confidence band and thus does not lie in B . By minimality of $\tilde{\mathcal{H}}$ Lemma 1 yields $s_{d,k+1} \cdot s_k > s_{opt}$, which completes the proof. □

The following proposition concerns the additional shrinkage performed in the Push-Up step. Similarly to Lemma 3, it states that a Push-Up step cannot lead to factors below s_{opt} and therefore yields the correctness of using the overall shrinkage factor to improve the upper bound on s_{opt} .

Lemma 4 *If $s_{d,k+1} \cdot s_k > s_{opt}$ then $s_{u,k+1} \cdot s_{d,k+1} \cdot s_k \geq s_{opt}$.*

Proof The statement is immediately given for $s_{u,k+1} = 1$. It is also clear in case of $k = 0$ by construction of the shrink update $s_{u,1}$. So in the following let $s_{u,k+1} < 1$ and $k \geq 1$ hold. We prove the proposition by contradiction so assume

$$s_{d,k+1} \cdot s_k > s_{opt} > s_{d,k+1} \cdot s_{u,k+1} \cdot s_k. \tag{6}$$

Consider the preceding candidate \mathcal{F}_k . $\mathcal{F}_k \notin B$ must hold, because otherwise the algorithm would have stopped after k steps. Furthermore $\mathcal{F}_k \geq L$ is guaranteed by construction of the Push-Up and binary search steps. Therefore, \mathcal{F}_k must violate the upper boundary U in the assumed case $k \geq 1$. Thus, a Shrink-Down step was executed before the current Push-Up step. Hence, the point

$$z_{eq} = \min_{z \in Z} \{z \mid s_{d,k+1} \cdot \mathcal{F}_k(z) = U(z)\}$$

is well defined, as pointed out in the description of the Push-Up step. The assumption $s_{u,k+1} < 1$ implies that a Push-Up step is carried out and $\exists z \in Z : z < z_{eq}$. By definition of z_{eq} , each $z < z_{eq}$ satisfies $s_{d,k+1} \cdot \mathcal{F}_k(z) < U(z) \leq U(z_{eq})$ and hence we deduce that

$$\forall z < z_{eq} : s_{d,k+1} \cdot \mathcal{F}_k(z) - U(z_{eq}) < 0. \tag{7}$$

Now consider the point

$$z' = \max \left\{ \operatorname{argmax}_{z < z_{eq}} (L(z) - s_{d,k+1} \cdot s_{u,k+1} \cdot \mathcal{F}_k(z)) \right\}.$$

By the definition of

$$s_{u,k+1} = \max_{s \in [0,1]} \{s \mid \forall z < z_{eq} : L(z) - s \cdot s_{d,k+1} \cdot \mathcal{F}_k(z) \leq U(z_{eq}) \cdot (1 - s)\}$$

it follows that

$$L(z') - s_{u,k+1} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z') = U(z_{eq}) \cdot (1 - s_{u,k+1}). \tag{8}$$

Also consider $z'' = \min\{\operatorname{argmax}_{z \leq z_{eq}} \mathcal{H}_k(z)\}$. Using the minimal property of \mathcal{H}_k proved in Lemma 2, for $k \geq 1$ one can deduce $\mathcal{F}_k(z'') = L(z'')$, which implies $z'' < z_{eq}$. For all $z \leq z''$ we obtain

$$\begin{aligned} L(z) - s_{u,k+1} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z) &\leq \mathcal{F}_k(z) - s_{u,k+1} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z) \\ &= (1 - s_{u,k+1} \cdot s_{d,k+1}) \cdot \mathcal{F}_k(z) \\ &\leq (1 - s_{u,k+1} \cdot s_{d,k+1}) \cdot \mathcal{F}_k(z'') \\ &= \mathcal{F}_k(z'') - s_{u,k+1} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z'') \\ &= L(z'') - s_{u,k+1} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z''), \end{aligned}$$

where the first inequality holds because of $\mathcal{F}_k \geq L$ by construction of \mathcal{F}_k . Combining this result with the already mentioned fact that $z'' < z_{eq}$ holds, we get $z' \geq z''$. Together with the monotonicity of \mathcal{H}_k and the definition of z'' we deduce

$$\mathcal{H}_k(z') = \mathcal{H}_k(z'') = \mathcal{H}_k(z_{eq}). \tag{9}$$

We now combine (6), (7), (8) and (9) to prove the proposition. By Lemma 3 and $s_{opt} \geq s^* > 0$ shown on page 6 the inequality $s_{d,k+1} \cdot s_k > 0$ holds. Thus, we can define $s_{u2} = \frac{s_{opt}}{s_{d,k+1} \cdot s_k}$ and inequality (6) implies

$$1 \geq s_{u2} > s_{u,k+1}, \tag{10}$$

which allows us to show

$$\begin{aligned} &L(z') - s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z') \\ &= L(z') + (-s_{u,k+1} \cdot s_{d,k+1} + s_{u,k+1} \cdot s_{d,k+1} - s_{u2} \cdot s_{d,k+1}) \cdot \mathcal{F}_k(z') \\ &\stackrel{(8)}{=} U(z_{eq}) \cdot (1 - s_{u,k+1}) + s_{u,k+1} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z') - s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z') \\ &= U(z_{eq}) - s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z') + s_{u,k+1} \cdot \underbrace{(s_{d,k+1} \cdot \mathcal{F}_k(z') - U(z_{eq}))}_{< 0 \text{ by (7)}} \\ &\stackrel{(10)}{>} U(z_{eq}) - s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z') + s_{u2} \cdot (s_{d,k+1} \cdot \mathcal{F}_k(z') - U(z_{eq})) \\ &= U(z_{eq}) \cdot (1 - s_{u2}). \end{aligned}$$

So we have

$$U(z_{eq}) \cdot (1 - s_{u2}) < L(z') - s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z'). \tag{11}$$

Hence we get

$$\begin{aligned} U(z_{eq}) &= U(z_{eq}) + s_{u2} \cdot \underbrace{(s_{d,k+1} \cdot \mathcal{F}_k(z_{eq}) - U(z_{eq}))}_{= 0 \text{ by definition of } z_{eq}} \\ &= U(z_{eq}) \cdot (1 - s_{u2}) + s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z_{eq}) \\ &\stackrel{(11)}{<} L(z') - s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z') + s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z_{eq}) \\ &= L(z') - s_{u2} \cdot s_{d,k+1} \cdot (s_k \cdot G_e(z') + \mathcal{H}_k(z')) + s_{u2} \cdot s_{d,k+1} \cdot \mathcal{F}_k(z_{eq}) \\ &= L(z') - \underbrace{s_{u2} \cdot s_{d,k+1} \cdot s_k}_{= s_{opt}} \cdot G_e(z') + s_{u2} \cdot s_{d,k+1} \cdot (\mathcal{F}_k(z_{eq}) - \mathcal{H}_k(z')) \\ &\leq \mathcal{H}_{opt}(z') + s_{u2} \cdot s_{d,k+1} \cdot (s_k \cdot G_e(z_{eq}) + \underbrace{\mathcal{H}_k(z_{eq}) - \mathcal{H}_k(z')}_{= 0 \text{ by (9)}}) \\ &\leq \mathcal{H}_{opt}(z_{eq}) + s_{opt} \cdot G_e(z_{eq}) \end{aligned}$$

where we also used $s_{opt} \cdot G_e + \mathcal{H}_{opt} \geq L$, which holds by definition of \mathcal{H}_{opt} . Thus, the upper boundary U is violated for s_{opt} , which contradicts its definition, so that the proposition follows. \square

The next result justifies the way we correct a solution to the unconstrained isotonic regression problem in line 11 of Algorithm 5. To be more precise, we show that setting its negative values to zero leads to the same L_∞ -error as in the constrained problem and therefore yields an optimal solution to the latter. Keep in mind that the unconstrained isotonic regression problem is solved by the *Basic* approach (Stout 2012), which computes the maximum of all previous values and the minimum of all subsequent values for each observation and determines the regression value as the average of these two quantities.

Lemma 5 *Let $x \in \mathbb{R}^d$ be arbitrary. Denote by x_L the optimal solution of the L_∞ isotonic regression of x computed by the Basic approach (Stout 2012) and define the new vector $x_{L0} = \max(x_L, 0)$ by component wise comparison to 0. Let x_{Lc} be an optimal solution of the L_∞ isotonic regression of x with the constraint of nonnegativity. Then x_{L0} is also an optimal solution to the constraint problem, i.e. $L_\infty(x, x_{Lc}) = L_\infty(x, x_{L0})$ holds.*

Proof We show the statement considering the two distinct cases $\min(x) \geq 0$ and $\min(x) < 0$ consecutively. At first, assume that $\min(x) \geq 0$ holds. Then, by construction of x_L , we can deduce $x_L \geq 0$. Thus, x_{L0} is equal to x_L and, as a nonnegative vector, satisfies $L_\infty(x, x_{Lc}) \leq L_\infty(x, x_{L0})$. Since introducing constraints to a problem cannot lead to a better value of the objective function in the optimum, it must hold that $L_\infty(x, x_{Lc}) \geq L_\infty(x, x_L) = L_\infty(x, x_{L0})$. Together this yields the result restricted to the case $\min(x) \geq 0$.

We now consider the case $\min(x) < 0$. Since the negative values of x_L set to zero in x_{L0} result in a maximal deviation of $-\min(x)$ to x , we get $L_\infty(x, x_{L0}) = \max(L_\infty(x, x_L), -\min(x))$. Also, $\min(x) < 0$ and $x_{Lc} \geq 0$ imply $L_\infty(x, x_{Lc}) \geq -\min(x)$, so that we deduce

$$\begin{aligned} L_\infty(x, x_{L0}) &= \max(L_\infty(x, x_L), -\min(x)) \\ &\leq \max(L_\infty(x, x_L), L_\infty(x, x_{Lc})) \\ &= L_\infty(x, x_{Lc}), \end{aligned}$$

where the last step follows, because a constraint problem cannot lead to a solution with a better value of the objective function compared to the corresponding unconstrained problem. Thus, $L_\infty(x, x_{L0}) \leq L_\infty(x, x_{Lc})$ holds. The converse inequality $L_\infty(x, x_{L0}) \geq L_\infty(x, x_{Lc})$ follows from the definition of x_{Lc} , since $x_{L0} \geq 0$. Both together yield the result restricted to the case $\min(x) < 0$, which completes the proof. \square

Using the above results we prove the correctness of our algorithm in the following theorem.

Theorem 1 *Algorithm 1 returns s_{opt} and a corresponding solution \mathcal{H}_{opt} both optimal in the sense of Problems 1 and 2, respectively.*

Proof Lemma 1 shows that for $s > s_{opt}$ no mixture can lie within the confidence band B while for $s \leq s_{opt}$ there always exists a mixture lying in B . By the monotonicity of this property the binary search step converges to s_{opt} . Lemmas 3 and 4 allow to update the upper bound of the binary search by the values of the shrinkage factor after each Shrink-Down and Push-Up step. Hence, these steps further reduce the range of possible candidates for s_{opt} , while never excluding s_{opt} and therefore the correct s_{opt} is still determined. Lemma 2 implies, that the correcting function \mathcal{H}_k after termination of the main loop of Algorithm 1 is the function \mathcal{H}_{min} introduced on page 6, which is required for solving Problem 2. Having found the set M_{norm} in the lines 5–10 of Algorithm 5, we use Lemma 5 to see that the corrected solution to the unconstrained L_∞ isotonic regression problem is an optimal solution to the constrained problem. Thus, it is a valid solution \mathcal{H}_{opt} , which completes the proof. \square

4.2 Runtime analysis

For the runtime analysis we introduce a precision parameter ε . Note that ε never appears in our pseudo code or actual implementation. Instead, think of it as the *machine precision*, which might depend on the physical architecture, operating system or programming environment. Note that the main loop of Algorithm 1 in lines 5 to 11 runs until the mixture \mathcal{F} is in the confidence band up to an additive deviation of ε . In other words, the loop stops when for all $z \in Z$ the property $L(z) - \varepsilon \leq \mathcal{F}(z) \leq U(z) + \varepsilon$ holds. In the following theorem we prove that this condition is met after a constant number of iterations yielding an overall running time linear in the input size and logarithmic in $\frac{1}{\varepsilon}$. Note that we exclude the $\Omega(n \log n)$ time needed for computing the cumulative distribution functions by assuming sorted input data, but rather focus on the linear running time of the actual analysis.

Theorem 2 *Let $\varepsilon \in (0, 1)$ be a fixed machine precision parameter. On an input of $n = n_1 + n_2$ observations, Algorithm 1 runs for at most $O(\log(\frac{1}{\varepsilon}))$ iterations. Each iteration can be implemented to run in time $O(n)$. The total running time is therefore of order $O(n \log(\frac{1}{\varepsilon}))$.*

Proof First note that the Shrink-Down, the Push-Up, the binary search step and the normalisation step can be implemented in linear, i.e. $O(n)$ time. Particularly, the solution to the isotonic regression subproblem (line 11 in Algorithm 5) can be computed in linear time as noted by Stout (2012). Therefore, it remains to bound the number of iterations of the loop in lines 5–11 of the main algorithm. The search interval for s is initialized to $[s^*, 1] \subset [0, 1]$ and halved at the end of every iteration where the binary search step is performed. The Shrink-Down and Push-Up steps can only further decrease the upper bound and consequently the size of the search interval. Therefore, after $\lceil \log_2(\frac{2}{\varepsilon}) \rceil$ iterations the size of the interval decreases to at most $2^{-\lceil \log_2(\frac{2}{\varepsilon}) \rceil} < \frac{\varepsilon}{2}$. So, after $\lceil \log_2(\frac{2}{\varepsilon}) \rceil$ iterations every value between the upper and lower boundary lies within additive precision $\frac{\varepsilon}{2}$ to s_{opt} . Consider an $s \in [s_{opt} - \frac{\varepsilon}{2}, s_{opt} + \frac{\varepsilon}{2}]$ and let $\mathcal{H}_s \in \mathcal{M}^*$ be the minimal function such that $s \cdot G_e + (1 - s) \cdot \mathcal{H}_s \geq L$ holds. Using $s \geq s_{opt} - \frac{\varepsilon}{2}$ we see that $s \cdot G_e \geq (s_{opt} - \frac{\varepsilon}{2}) \cdot G_e = s_{opt} \cdot G_e - \frac{\varepsilon}{2} \cdot G_e \geq s_{opt} \cdot G_e - \frac{\varepsilon}{2}$

holds. The property $s \cdot G_e \geq s_{opt} \cdot G_e - \frac{\varepsilon}{2}$ implies $(1 - s) \cdot \mathcal{H}_s \leq (1 - s_{opt}) \cdot \mathcal{H}_{opt} + \frac{\varepsilon}{2}$ and we deduce

$$\begin{aligned} s \cdot G_e + (1 - s) \cdot \mathcal{H}_s &\leq \left(s_{opt} + \frac{\varepsilon}{2}\right) \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H}_{opt} + \frac{\varepsilon}{2} \\ &\leq s_{opt} \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H}_{opt} + \varepsilon \\ &\leq U + \varepsilon, \end{aligned}$$

because $s_{opt} \cdot G_e + (1 - s_{opt}) \cdot \mathcal{H}_{opt} \leq U$ holds by definition of s_{opt} and \mathcal{H}_{opt} . An analogous argument shows $s \cdot G_e + (1 - s) \cdot \mathcal{H}_s \geq L - \varepsilon$. Thus, the stopping criterion $L - \varepsilon \leq \mathcal{F} \leq U + \varepsilon$ is met after $\lceil \log_2 \left(\frac{2}{\varepsilon}\right) \rceil$ iterations and the result follows. \square

5 Application

In this section we evaluate the algorithm by applying it to simulated and real data sets. We compare the running time of our algorithm to alternative procedures on artificial data, investigate its capability to estimate the disagreement of the distributions for finite Gaussian mixtures and examine its performance in case of false rejections of the null hypothesis. Furthermore, the algorithm is illustrated on astrophysical data and on spectrometry data from a biological domain. All methods conducted in this section are carried out using the significance level $\alpha = 0.05$.

For our empirical evaluation we implemented all algorithms using the statistical software R (R Core Team 2013), version 2.15.1-gcc4.3.5. To run the experiments in a batch and to distribute the computations to the cores of our computer the R-package BatchExperiments by Bischl et al. (2013) was used. The computations were conducted on a 3.00GHz Intel Xeon E5450 machine with 15GB of available RAM running a SuSE EL 11 SP0 Linux distribution.

Note that our algorithm represents the determined correction distribution by its cumulative distribution function. However, since densities and the first two moments allow to capture the main features of a distribution more intuitively, we present our results via estimated densities and empirical moments rather than using the determined correction distribution functions itself. To attain those, we first determine the empirical density function corresponding to the determined correction distribution function by considering consecutive differences of \mathcal{H}_{opt} . We then generate 10,000 artificial observations from this density using weighted sampling. Finally, the standard kernel density estimator as well as the empirical mean and the empirical variance are computed on this artificial data. Note that in applications this approach is not mandatory, since the determined distribution function contains all relevant information available. Therefore, improving simulations based on this distribution function directly is perfectly fine in practice and there is not absolutely necessary to introduce the density and moment estimation, which we conduct for the purpose of presentation only. Thus, the kernel density estimation and the moment estimation are *not* regarded as part of our method. To asses their effect the interested reader is referred to Serfling (1980) and Devroye and Gyrfi (1985).

5.1 Performance and runtime evaluation

In order to evaluate the algorithm output we first consider the popular setting of finite Gaussian mixtures. We generate equally sized dataset pairs for each of the sample sizes $n_1 = n_2 = 100, 500, 1000, 5000, 10,000, 50,000, 100,000$. In every data set pair one sample is drawn from a standard Gaussian distribution. The other also consists of observations from the standard Gaussian distribution to a fraction of $s = 0.3$. The remaining fraction of 0.7 of the observations stem from a second Gaussian distribution with mean 3 and standard deviation 1. Our demixing algorithm is therefore supposed to notice the different distributions of the samples, estimate a mixing proportion of about 0.3 and recommend a correction distribution with a mean near 3 and a standard deviation near 1.

We also investigate another data case for the same sample sizes and $n_1 = n_2$, in the following referred to as the zero mixing data scenario. It is more specific and resembles some of the situations encountered in our real data application. Instead of mixing two Gaussian distributions, we set the constant value 0 for 70% of the observations in each mixed data set. The remaining 30% are sampled from the Gaussian distribution with mean 3 and standard deviation 1. The corresponding second sample representing the simulated data consist of observations of the same Gaussian distribution entirely. In this setting the method should again determine a shrinkage value s_{opt} around 0.3 and propose a correction distribution putting most of its probability mass at 0. Both scenarios are replicated 1000 times for each of the sample sizes, respectively.

We study the running time of Algorithm 1 by comparing it to a simpler demixing approach. The algorithm, called binary search procedure in the following, determines the optimal shrinkage factor s_{opt} relying only on the binary search. In contrast to Algorithm 1, the Shrink-Down and Push-Up steps are not conducted. Keep in mind that both steps are in principle not necessary to obtain the correct solutions to Problem 1 and 2 but are supposed to accelerate the computation. Thus, the determined s_{opt} and \mathcal{H}_{opt} are identical for both methods, but the running times differ.

Table 1 shows the results for both data cases described above. In the Gaussian mixture setting we list the determined shrinkage factors s_{opt} as well as the mean and standard deviation of samples of size 10,000 drawn from the determined correction distribution \mathcal{H}_{opt} for each sample pair. The second half of the table corresponds to the zero mixing data scenario. In addition to the determined shrinkage factors s_{opt} we give the probability which is assigned to the value 0 by the determined distribution functions \mathcal{H}_{opt} . The corresponding running times for both algorithms are presented in Fig. 1 for the Gaussian mixture case in seconds. Thereby, the time needed for precomputing the empirical distribution functions is not included. For the sake of presentation we do not show the running times for the two largest sample sizes $n_1 = n_2 = 50,000$ and $n_1 = n_2 = 100,000$. These were 1.18 and 2.47 s, respectively, for Algorithm 1 and 6.2 and 12.33 s, respectively, for the binary search procedure. We also omit the running times for the zero mixture case, which are essentially the same as in the Gaussian mixture case. All results are averages over 1000 repetitions.

The results for both simulations suggest that demixing leads to an overestimation of the expected mixing proportion 0.3, which decreases as the sample size increases. This is not surprising, since by definition s_{opt} is the maximal shrinkage factor such that

Table 1 Determined shrinkage factors s_{opt} , estimations of the mean and the standard deviation of the correction distribution \mathcal{H}_{opt} in the Gaussian mixture case (upper part of table) and estimated probability mass assigned to 0 by \mathcal{H}_{opt} in zero mixture case (lower part of table) for different sample sizes

	100	500	1000	5000	10,000	50,000	100,000
s_{opt}	0.516	0.409	0.380	0.341	0.331	0.317	0.313
Mean	3.504	3.282	3.217	3.121	3.096	3.055	3.043
SD	0.693	0.795	0.832	0.892	0.910	0.944	0.954
s_{opt}	0.481	0.381	0.358	0.326	0.318	0.308	0.306
$P_{\mathcal{H}_{opt}}(0)$	0.979	0.993	0.995	0.998	0.999	0.999	1.000

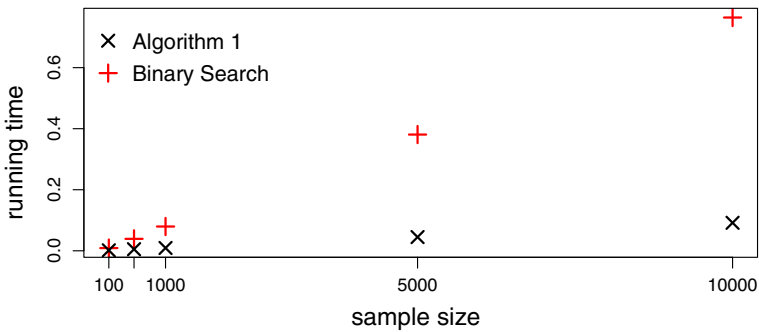


Fig. 1 Average running times of Algorithm 1 and the binary search procedure computed on sorted samples for different sample sizes in seconds

the corresponding mixture lies in the confidence band. Therefore, as the sample size grows, the radius of the confidence band becomes smaller and hence s_{opt} converges towards s . The estimated mean and standard deviation in the Gaussian mixture case behave similarly approaching 3 and 1, respectively. In the zero mixing data scenario even for small sample sizes an overwhelming majority of the probability mass in \mathcal{H}_{opt} is assigned to the value 0. This is correct, since by construction the differences between the sample pairs are caused by the zero values only. Thus, the correction distributions proposed by the methods reflect the discrepancies between the sample pairs quite well.

The running time for both algorithms grows linearly in the sample size, given sorted input data, and is by a factor of approximately 6 smaller for Algorithm 1 than for the binary search procedure. This holds for both data cases and shows that the Shrink-Down and Push-Up steps lead to large savings in computation time and are therefore very valuable for large data sets.

In Fig. 2 we illustrate the output in the Gaussian mixture case for $n_1 = n_2 = 1000$. In the upper row, kernel density estimations of the provided samples are presented. Demixing the samples using Algorithm 1 leads to the shrinkage factor $s_{opt} = 0.39$, which is a reasonable approximation of the true mixture proportion $s = 0.3$. Using the approach explained above we generate a third sample with 10,000 observations from the correction distribution characterised by \mathcal{H}_{opt} . Its mean 3.3 and standard deviation 0.81 are also close to the desired values 3 and 1, respectively. The corresponding kernel density estimation shown on the right in the lower row is almost symmetrical

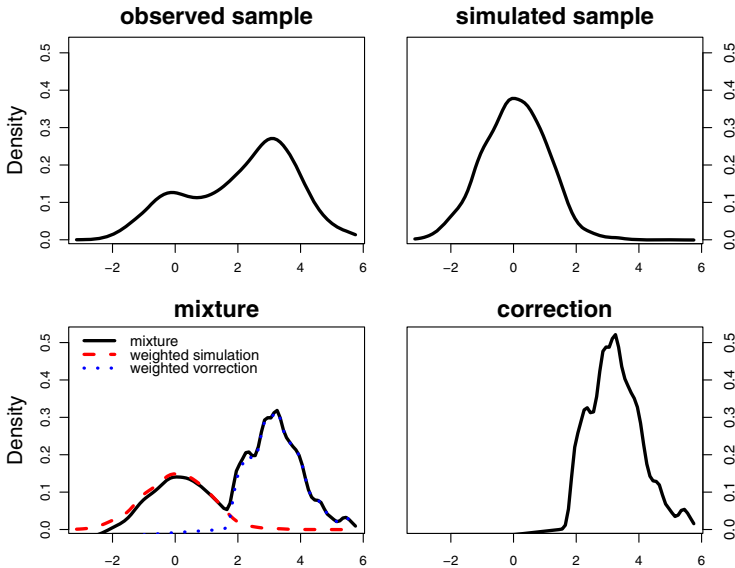


Fig. 2 Kernel density estimations for two samples, the computed mixture and the correction distribution in the Gaussian mixture setup

and unimodal. Hence, the correction distribution represents the deviation between the underlying distributions of the first and the second sample quite well. The final mixture distribution proposed by Algorithm 1, which is the sum of the weighted distribution of the second sample and the weighted correction, is given by the corresponding estimated density in the lower left corner. The curve resembles the one of the first sample as desired.

5.2 Estimated shrinkage factors under the null hypothesis

Under the null hypothesis H_0 , the analysed samples stem from the same distribution. In this situation, the Kolmogorov–Smirnov test rejects by mistake in about an α -fraction of the cases, where α is the predefined significance level. In these cases, a reasonable procedure comparing the samples in the mixture framework should recognise their similarity. Thus, a shrinkage factor near 1 is desirable after a false rejection of the null hypothesis.

To check the performance of our method under H_0 , dataset pairs are generated for the sample sizes $n_1 = n_2 = 100, 500, 1000, 5000, 10,000$. All samples stem from the standard Gaussian distribution. Other distributions like the exponential and the t-distribution were also considered and led to comparable results. For each sample size, dataset pairs are simulated until the Kolmogorov–Smirnov test rejects in 1000 cases. These 1000 dataset pairs are passed to Algorithm 1, which determines corresponding shrinkage factors. These are presented via boxplots in Fig. 3. All of them are less than 1 by construction.

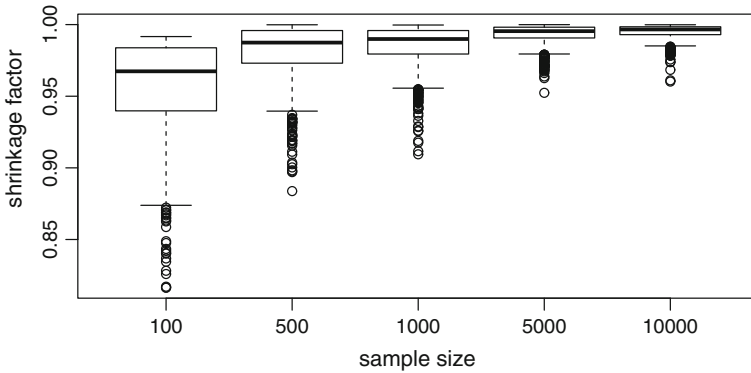


Fig. 3 Shrinkage factors determined by Algorithm 1 for varying sample sizes after a false rejection of the null hypothesis $H_0 : P = Q$, where P is the standard Gaussian distribution

As the results show even for small sample sizes the majority of shrinkage values are greater than 0.9. Increasing the sample size further reduces the amount of small shrinkage values. Thus, our method performs as desired: If no modifications are actually necessary, the algorithm proposes to perform none or only small modifications to the current samples.

5.3 Application to real data from astrophysics

We now illustrate our Algorithm 1 applying it to the astrophysical problem mentioned in the introduction. More precisely, we consider simulated proton data and compare it to observations recorded by the gamma ray detectors MAGIC-I and MAGIC-II. The latter are almost completely induced by protons. Both datasets consist of 5000 observations and contain 54 continuous attributes that we work with. Among other features, these variables mainly include characteristics of the recorded atmospheric signals and their reconstructed trajectory and are identical for both data sets. The results of our method allow to determine attributes which differ the most for simulated protons and observed data and to quantify their discrepancies. This information can subsequently be used to improve the background simulation.

The Kolmogorov–Smirnov test comparing the real data and the simulation rejects the null hypothesis of identical distributions for all but two attributes. However, 37 of the 54 attributes have shrinkage factors above 0.85, which indicates a suitable proton simulation overall. The upper row of Fig. 4 provides kernel density estimations for the observed and simulated data for the attribute Length1, which describes the length of an ellipse fitted to an atmospheric signal measured by the MAGIC-I detector. The Kolmogorov–Smirnov test for Length1 rejects the null hypothesis of equal distributions of observed and simulated data leading to a comparably low shrinkage factor of 0.75. Therefore, the simulation of this variable might be inadequate and the corresponding simulation steps seem to be worth inspecting in more detail. In the lower right corner a kernel density estimation is presented for the correction distribution characterised by \mathcal{H}_{opt} . It is based on 10000 observations generated by the

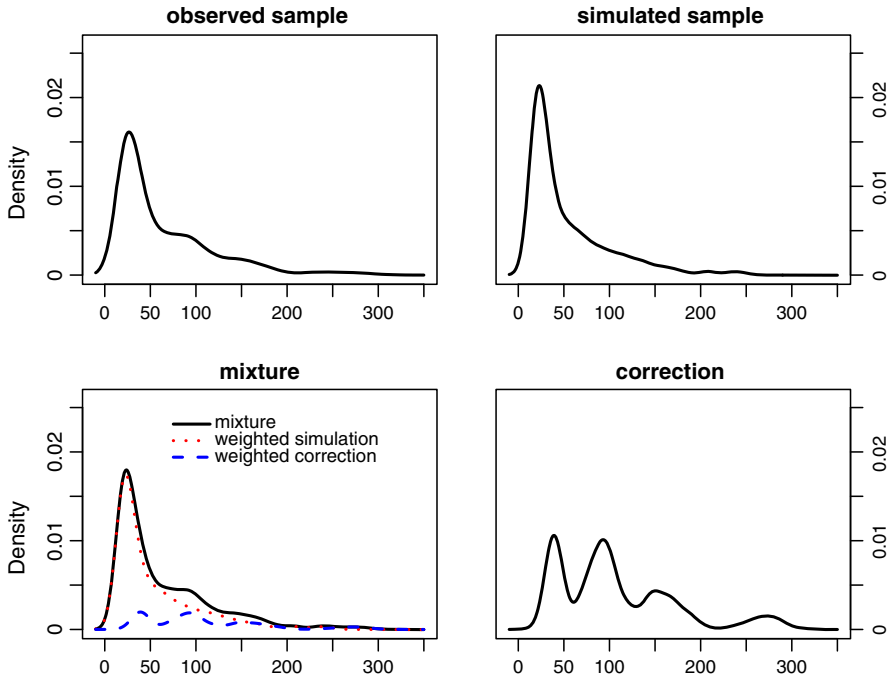


Fig. 4 Kernel density estimations for the attribute Length1 based on the observed data, the simulated data, the determined mixture of the simulated and corrected distributions and the correction distribution

sampling technique explained on page 22. The plot in the lower left corner shows the density estimations of the simulated and the correction distribution weighted by 0.75 and 0.25, respectively, as well as the density estimation for the final mixture, which is the sum of the weighted estimations. All plots are presented on the same scale.

The coarse form of the density estimations for the observed data and the simulation in the upper row is quite similar showing one major peak around 25. However, there are some slight discrepancies. Compared to the real data curve, the main peak of the simulation is considerably higher. While the curve for the real data has a plateau around 90, we have a steadily falling curve for the simulation. Although these differences are not very large, it is quite unlikely that they are induced by the sample variance due to the large sample sizes. In order to verify this hypothesis, we have conducted several simulations considering kernel density estimators for a broad class of distributions using 5000 observations in each sample. The dissimilarities in these simulations were much smaller than for the Length1 attribute supporting the conjecture that the Kolmogorov–Smirnov test correctly rejected the null hypothesis.

In order to correct the simulated sample, one should obviously generate less observations around 25 and more around 90. This is exactly what is proposed by the corrected distribution presented in the lower right corner of Fig. 4, which reflects the Length1

distribution of non-proton events. The corresponding density curve based on the estimated \mathcal{H}_{opt} has a peak near 25, but also another one of comparable height and greater width near 90. Therefore, it gives the region around 90 about as much weight as the one around 25, in contrast to the simulated sample.

Combining the simulated and the correction distributions in the proportions determined by the algorithm, we get the density curve of the final mixture presented in black in the lower left plot of Fig. 4. It resembles the density estimation for the observed data above it quite well. On the one hand, the height of the main peak is corrected, which is achieved by the shrinking. On the other hand, the required plateau is introduced to the mixture by the correction density.

5.4 Application to real data from bioinformatics

Algorithm 1 is applied to evaluate so called ion mobility spectrometry (IMS) measurements that help to detect volatile organic compounds in the air or in exhaled breath. Motivated by the need to process such measurements in real-time as they arrive one-by-one, it is a usual approach to find and annotate major peaks in the data. In this way the original information is summarised in a compressed representation. In an effort to automate and speed-up the computations, D'Addario et al. (2014) propose to approximate the measurements by finite mixtures of probability density functions, whose parameters are estimated using a variant of the EM algorithm. The computations are performed on a sequence of measurements leading to a two dimensional problem, where both dimensions are modelled independently by mixtures of inverse Gaussian densities.

Focusing on one of the dimensions and conditioning on the other, we are given 6000 spectrograms consisting of 12,500 data points each, stemming from 10 minutes of IMS measurement (cf. Koczynski et al. 2012). In this data, we identify 187 groups of spectrograms belonging to the same peak models, respectively. We take samples of size 1000 generated from each spectrogram and the corresponding mixture model given from the bioinformatics algorithms. Both of these are regarded as probability density functions up to some normalising constants. In order to evaluate their models, we apply our algorithm at a significance level of five percent. Hereby we inspect the discrepancies to the samples taken from their corresponding spectrograms.

In general our algorithm suggests that the models fitted by the bioinformaticians approximate their spectrograms reasonably well, since in 152 of the 187 groups the mean shrinkage factor for the spectrograms is above 0.8. In addition, we identify some interesting groups of spectrograms. The shrinkage factors of two of these are shown in Fig. 5. Keep in mind that the spectrogram index represents the second dimension of the data we condition on. In both groups the model in the second dimension consists of a single inverse Gaussian density.

Our algorithms' results for group A suggest that the first half of the measurements are modelled quite well by the bioinformaticians' EM algorithm, but for increasing spectrogram indices the approximation is getting worse. This shows that the bioinformatics model in the second dimension is not appropriate. Instead of a single inverse

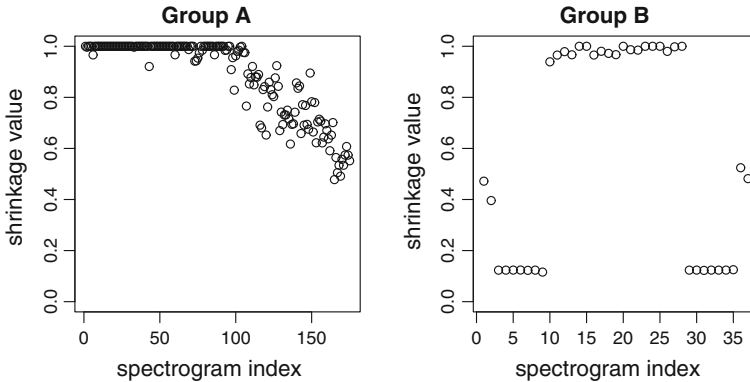


Fig. 5 Shrinkage factors of two groups of spectrograms determined by Algorithm 1

Gaussian density, two components would probably lead to better approximations, since they allow to model both halves of the spectrograms with a density function, respectively. In contrast to that, the shrinkage factors for group B indicate a sufficient number of components used in the second dimension. For the spectrograms in the middle we have shrinkage factors close to one. This means that their corresponding models are close to the spectrograms. However, going to the left and right borders, the spectrograms seem to be fitted quite badly since the shrinkage factors are lower than 0.2. The two leftmost and two rightmost models are a little closer to their spectrograms with shrinkage values between 0.4 and 0.6. Taking the models of [Kopczynski et al. \(2012\)](#) into account this indicates that their fitted density mixture might be too wide or too narrow in the second dimension. Thus, the approximation could be substantially improved by excluding the two spectrograms on both margins, respectively, from this group and treating them by further models.

We also illustrate the method using a single spectrogram from the data set. The upper row of [Fig. 6](#) provides a kernel density estimation for the measurement 1157 and its model. Since all four plots are given on the same scale, the two peaks in the model are more narrow and differ much more in height than the ones in the original data. In addition, the peak on the left is missing. Although it looks small in this scale, it appears noteworthy when compared to the other two. In the second row on the right a kernel estimation for the correction distribution characterised by \mathcal{H}_{opt} is presented. It is based on 10000 observations generated by the sampling approach described on page 22. As expected, the correction distribution puts mass on the very right peak in order to fix the height proportions between the peaks on the right. In addition it generates the left peak missing in the model. The plot in the lower left corner shows the estimations of the modelled and the correction distribution weighted by the determined shrinkage value 0.76 and the remaining mass 0.24, respectively, as well as the kernel estimation for the final mixture, which is the sum of the weighted estimations. The proposed mixture is still somewhat narrow, but the proportions of the peak heights as well as the small peak are better represented compared to the original model.

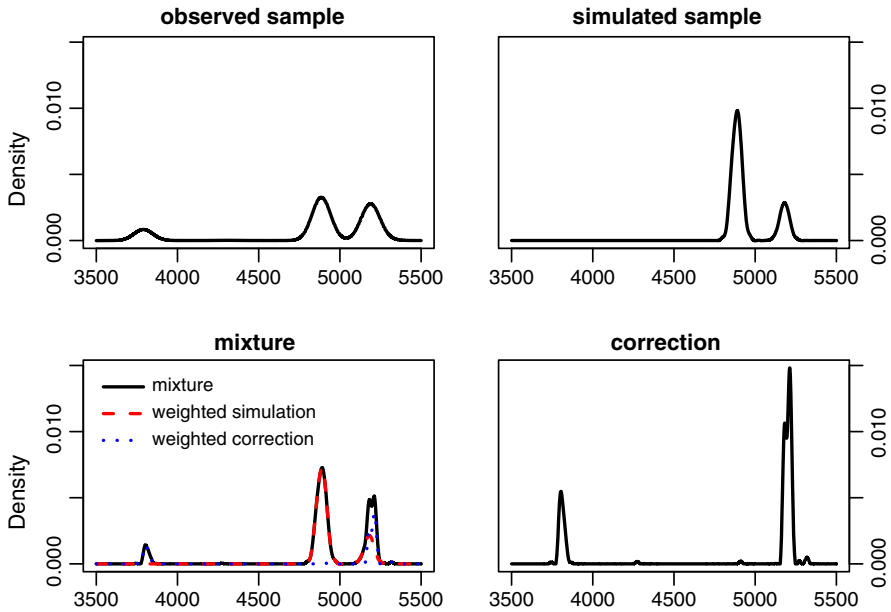


Fig. 6 Kernel density estimations for a spectrogram based on the measurements, the corresponding inverse Gaussian model, the determined mixture of the model and the correction distribution

6 Conclusion

This article deals with the nonparametric two sample homogeneity problem. A widely-used tool to test the equality of the distributions corresponding to the two samples is the Kolmogorov–Smirnov test. We develop an algorithm which, in case of a rejection by this test, determines how the first sample should be modified to resemble the second one in the Kolmogorov–Smirnov sense. This modification is quantified by an empirical correction distribution and the corresponding proportion determined by our method. Combining the information of the first sample with this correction in the proportion identified leads to an appropriate mixture. Its Kolmogorov–Smirnov distance to the second sample is so small that the test would not reject the hypothesis of equal distributions any more. Our method is especially of interest as an assisting tool in the task of designing an adequate simulation modelling an observed data generating process. Comparing a sample from an existing simulation based on domain specific prior knowledge and a sample of observed data, our algorithm determines a correction distribution. The information provided therein may subsequently be used to improve the current simulation. Since the procedure is completely nonparametric, it is widely applicable and in particular not only useful in the settings considered in our simulations.

The algorithm proceeds in an iterative manner applying several correction steps linked with a modified binary search technique. The constructed distribution function is shown to be optimal in a reasonable sense and the running time of the algorithm is proved to be of linear order on sorted data. The Shrink-Down and Push-Up steps applied in addition to the standard binary search algorithm lead to large savings in

computation time. In our experience it converges in three iterations in the majority of all cases independent of the sample sizes. The algorithm proposes none or only slight corrections in cases where both datasets stem from the same distributions. The correction distributions proposed in simulations as well as for a real data example are intuitive and adequate.

There are several possibilities to extend the presented ideas in future work. On the one hand, instead of focusing on distribution functions, a density based approach to the demixing problem could also be of interest. Working with densities is often even more intuitive than using distribution functions and there exists a broad literature on mixture models dealing with density estimation (e.g. [Schellhase and Kauermann 2012](#)). However, as pointed out in the introduction, transferring the two sample problem to the density framework is not straightforwardly achieved by applying the existing techniques and may be computationally more demanding, so that much work has to be done here. On the other hand, one could use alternative test procedures for distribution functions besides the Kolmogorov–Smirnov test to construct the confidence bands. Although the Kolmogorov–Smirnov test is quite popular, competitors like the Anderson–Darling and the Cramér von Mises test detect differences between two distributions more often in certain settings (cf. [Razali and Wah 2011](#)) and could thus lead to better demixing results. In this work we focused on the Kolmogorov–Smirnov test since the simple shape of the corresponding confidence band allows for finding an efficient algorithm solving the demixing problem. The extension to analytically more sophisticated distance measures where our proofs do not carry over in a straightforward manner is a challenging and promising open problem for future research. Another direction is to generalize our method to distributions over multi-dimensional domains based on appropriate extensions of the Kolmogorov–Smirnov test. Several multivariate versions of the test are surveyed by [Lopes et al. \(2007\)](#) and could serve as a starting point towards extending our method to the multivariate setting.

Acknowledgments We thank the anonymous referees for their valuable remarks which helped us to improve this work. We would like to thank Prof. Dr. Roland Fried for many helpful suggestions and the MAGIC collaboration for providing us with astrophysical data. We thank Marianna D’Addario and Dominik Kopczynski, both members of the Bioinformatics group of Prof. Dr. Sven Rahmann, for their data from another interesting domain in cooperation with Project SFB876-TB1. Part of the work on this article has been supported by Deutsche Forschungsgemeinschaft (DFG) within the Collaborative Research Center SFB 876 “Providing Information by Resource-Constrained Analysis”, projects C3 and C4.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Bischl B, Lang M, Mersmann O (2013) BatchExperiments: statistical experiments on batch computing clusters. R package version 1.0-968. <http://CRAN.R-project.org/package=BatchExperiments>
- Cortina J, Goebel F, Schweizer T (2009) for the MAGIC collaboration: technical performance of the MAGIC telescopes. [arXiv:0907.1211](https://arxiv.org/abs/0907.1211)
- D’Addario M, Kopczynski D, Baumbach JI, Rahmann S (2014) A modular computational framework for automated peak extraction from ion mobility spectra. *BMC Bioinform* 15(1):25

- Devroye L, Györfi L (1985) Nonparametric density estimation: the L_1 view. Wiley, New York
- Durbin J (1973) Distribution theory for tests based on the sample distribution function. In: Regional conference series in applied mathematics, vol 9. SIAM, Philadelphia
- Hall P, Neeman A, Pakyari R, Elmore R (2005) Nonparametric inference in multivariate mixtures. *Biometrika* 92(3):667–678
- Heck D, Knapp J, Capdevielle JN, Schatz G, Thouw T (1998) CORSIKA: a Monte Carlo code to simulate extensive air showers. Technical report FZKA 6019, Forschungszentrum Karlsruhe
- Hettmansperger TP, Thomas H (2000) Almost nonparametric inference for repeated measures in mixture models. *J R Stat Soc Ser B* 62(4):811–825
- Kolossiatos M, Griffin JE, Steel MFJ (2013) On Bayesian nonparametric modelling of two correlated distributions. *Stat Comput* 23(1):1–15
- Kopczynski D, Baumbach JI, Rahmann S (2012) Peak modeling for ion mobility spectrometry measurements. In: Proceedings of the 20th European signal processing conference (EUSIPCO 2012), pp 1801–1805
- Lopes RHC, Reid I, Hobson PR (2007) The two-dimensional Kolmogorov-Smirnov test. In: Proc. XI Int. Workshop Adv. Comput. Anal. Tech. Phys. Res
- Pilla RS, Lindsay BG (2001) Alternative EM methods for nonparametric finite mixture models. *Biometrika* 88(2):535–550
- R Core Team (2013) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org>
- Razali N, Wah YB (2011) Power comparisons of Shapiro–Wilk, Kolmogorov–Smirnov, Lilliefors and Anderson–Darling tests. *J Stat Model Anal* 2(1):21–33
- Schellhase C, Kauermann G (2012) Density estimation and comparison with a penalized mixture approach. *Comput Stat* 27:757–777
- Serfling RJ (1980) Approximation theorems of mathematical statistics. Wiley, New York
- Stout QF (2012) Strict L_∞ isotonic regression. *J Optim Theory Appl* 152(1):121–135
- The MAGIC Collaboration: The MAGIC telescopes. <https://magic.mpp.mpg.de/> (2014)
- Wang Y (2010) Maximum likelihood computation for fitting semiparametric mixture models. *Stat Comput* 20(1):75–86