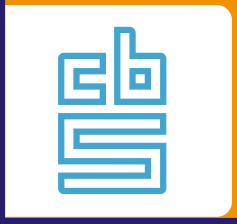


Discussion Paper

Multivariate Density Estimation by Neural Networks

Dewi Peerlings, Jan van den Brakel, Nalan Baştürk and Marco Puts

May 12, 2021

In this paper, we propose a new method to obtain the probability density function (PDF) to assess the properties of the underlying Data Generating Process (DGP) without imposing any assumptions, using neural networks. Knowledge about these distributional characteristics has a wide range of applications in statistics. In this paper it is proposed to filter a signal from a high volatility data set that also contains missing and erroneous observations. This is relevant, for example for national statistical institutes, where there is an increasing interest in the use of data which are generated as a by-product of processes not related to statistical production purposes as an alternative for survey sampling.

The proposed artificial neural networks have additional advantages compared to well-known parametric and non-parametric density estimators. Our approach builds on literature on cumulative distribution function (CDF) estimation using neural networks. We extend this literature by providing analytical derivatives of this obtained CDF. Our approach hence removes the approximation error in the second step of obtaining the PDF from the CDF output, leading to more accurate PDF estimates. We show that the proposed solution to obtain the PDF holds for correlated variables in a multivariate setting and for neural networks with several hidden layers. We illustrate the accuracy gains from our proposed method using several simulation examples, where the real DGP is known, hence the improvements in accuracy compared to existing methods can be assessed. We follow the illustrations in continuous data cases of related literature, a discrete data application of mixed correlated Poisson distributions and a multivariate data application concerning correlated standard normal distributions.

Key words:Feed Forward Neural Networks, Non-Parametric Density Estimation, Filtering, Count data, Simulation Studies

1 Introduction

In this paper a method to estimate probability densities using neural networks is proposed. This finds a wide range of applications in statistics and data analysis. The production of official statistics is one potential area of real life applications for this methodology. National statistical institutes, responsible for the production of official statistical information about modern societies, traditionally collect the required data through probability sampling, see Cochran (1977) or Särndal et al. (1992). In order to reduce administration costs and improve timeliness as well as the level of detail of this statistical information, there is a growing interest among national statistical institutes to make more use of data that are generated as a by-product of processes not directly related to statistical production purposes (so called big data or organically measured data). Examples are messages from social media platforms, search behaviour on internet, cell-phone data, and road sensor data. Such data, in contrast to 'designed data' collected from surveys, are often generated and collected at a relatively high frequency. Hence resulting in a large amount of data that is accessible. However, the amount of information in such organically measured data is often smaller than in designed data.[1)]

The quality of the information content in the former can be lower due to several factors, such as the existence of missing observations, incorrect measurements and outliers due to the differences between the population characteristics and those of the sample that the data are collected from or device malfunctioning. In addition, when the primary objective of data collection differs from the goal of the analysis, which is often the case in organically measured data, the information content of the data for the specific analysis can be low, see Puts et al. (2018).

The above-mentioned data characteristics are referred to as a low signal-to-noise property with typically high volatility, missing observations and sample selectivity, where the signal indicates the information content. However, such data could improve the timeliness and the level of detail of official statistics based on repeated probability samples, since these new data sources come at a higher frequency in considerably larger volumes van den Brakel et al. (2017). Official statistics can be updated more frequently by real time estimates which use information that is released on a higher frequency. These real time forecasts are also called nowcasts. Hence by the use of nowcasting, official statistics can be improved. The careful use of such data, brings the necessity to assess the signal and noise. A typical way to proceed in separating the signal and noise is filtering, where the adequate information content with lower volatility, the signal, is extracted, the effect of outliers are reduced and missing observations are imputed using a filter, see Durbin and Koopman (2012). Following this filtering process, the data can be used for real time predictions, as we elaborate in the next section.

In this paper, we focus on obtaining a probability density function (PDF) or probability mass function (PMF) to assess the properties of the underlying data generating process (DGP) using artificial neural networks. If we talk about both continuous and discrete distributions in general then we refer to this shortly as PDF. An appropriately obtained PDF can be used for several filters such as the recursive Bayesian filter or a particle filter, which require an underlying assumption about the probability density. The proposed

1) https://www.census.gov/newsroom/blogs/director/2011/05/designed-data-and-organic-data.html

artificial neural networks have additional advantages compared to well-known parametric and non-parametric density estimators. The parametric approaches in the literature are restrictive since an underlying DGP has to be assumed. Such assumptions, e.g. on the shape of the distribution can result in false extraction of the signal. The non-parametric approach which is Kernel Density Estimation (KDE), on the other hand, is highly dependent on selecting the bandwidth which affects the level of smoothing and suffers from an increase of parameters to estimate especially in high dimensions. As the number of observations grow, a higher number of parameters are added. The use of neural networks overcomes the above-mentioned disadvantages, see Bishop et al. (1995). The proposed method is semi-parametric, since it does not assume a parametric distribution on the underlying DGP a priori. Instead, it relies on a neural network that can be interpreted as a model to describe the distribution of the observed data, see also Magdon-Ismail and Atiya (2002).

Our approach builds on the literature on cumulative distribution function (CDF) estimation using neural networks, see Magdon-Ismail and Atiya (2002). In this literature, the output of a neural network is the CDF, the PDF is then obtained in a second step by numerically differentiating the obtained CDF values at grid points[2)]. We extend this literature by providing the analytical derivatives of the obtained CDF from the artificial neural network. Our approach hence removes the approximation error in the second step of obtaining the PDF from the CDF output, leading to more accurate PDF estimates. These analytical derivatives apply for any model. In other words these hold irrespective of the number of hidden layers and hidden neurons. In addition, model selection is executed using a novel approach. Several potential models, that is a specific number of hidden layers and hidden neurons for each model, are combined in only one neural network in order to save computational time. We show that the proposed solution to obtain the PDF from the CDF output of an artificial neural network holds for correlated variables in a multivariate setting and for a neural network with several hidden layers. We illustrate the accuracy gains from our proposed method using several simulation examples, where the real data generating process is known, hence the improvements in accuracy compared to existing methods can be assessed. More specifically, we follow the illustrations in continuous data cases of Magdon-Ismail and Atiya (2002) and Trentin et al. (2018), a discrete data application of mixed correlated Poisson distributions and a multivariate data application concerning correlated standard normal distributions.

To embody the importance of this research, section 2 gives a motivating example. Section 3 provides a brief literature review of the existing approaches. The proposed method and analytical derivation is described in section 4. Section 5 shows the results from the applications. Finally, a discussion is given.

2 Motivating Example

An example of organically measured data includes traffic information obtained from road sensors on highways. Such data are collected on a frequent basis from several locations in The Netherlands. These road sensors are used e.g. to detect accidents or to

[2)] We note that it is also possible to estimate the PDF by neural networks straight away, see Trentin et al. (2018). However, target values for a PDF are less intuitive to construct than target values for a CDF. In addition, the method has a built-in regularizer that smoothens the PDF potentially disproportionately.

adjust traffic flow. However, for government agencies, there are several secondary uses of these data. The statistics of interest in this case include, as an example, the level of usage in different highway sections such that a maintenance plan or capacity increases in highways can be made adequately or official statistics about traffic intensity can be produced.

Road sensors measure the speed and amount of several vehicle types passing at every minute, but the collected data are prone to occasional malfunctioning of the sensors, leading to an additional reason why the signal-to-noise ratio in these data is low. Preprocessing is required to impute noisy observations using filtering methods. When preprocessing is accomplished, these data are also useful for official statistics. See Figure 2.1 as an example. This Figure illustrates the amount of vehicle counts which pass a certain sensor for every minute in one day. Malfunctioning of the sensor is detected by an assigned value of -1 or by counting too many or too few vehicles. In the former case we are sure about malfunctioning, though in the latter case we are not so sure about this. In addition, there is a lot of fluctuation in counts during several subsequent minutes which is quite implausible to occur. However, some valuable information can be retrieved from the raw data, e.g. the expectation of having a rush hour in the morning and late afternoon. Around the 400th minute and 1000th minute of the day which is around 6:30 and 16:30 o'clock there are more vehicles tracked than at other time slots in the day.

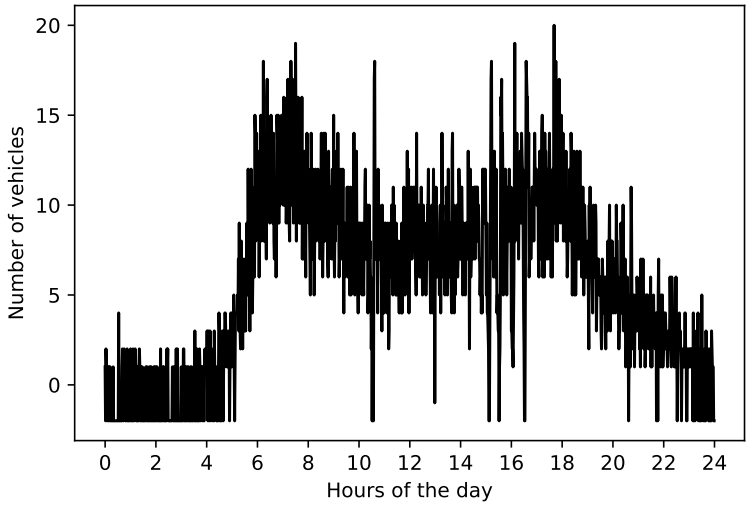


Figure 2.1 The number of vehicle counts passing road sensor `GEO02_R_RWSTI320`, lane 2 on the 26th of April 2016

A further consideration in analyzing the organically measured data in the above examples is the time-dimension. Traffic intensity has several properties according to the date and time of the day. For example, if the aim is to detect events of traffic jams, it is important to focus on rush hours or public holidays. For these cases, the filtering methods should include time series properties in order to obtain the conditional PDF of the amount of vehicles at a certain time horizon. Similar to the unconditional PDF, the conditional PDF can also be used as an input to other filtering methods.

The construction of the conditional PDF above is not straightforward due to the noise in the data as well as the secondary use of data. Summary statistics and properties of data

can be used as a basis to construct these conditional PDF values. A recent method for this purpose is proposed in Puts et al. (2018), where traffic intensity is characterised by a local level model for discrete road sensor vehicle counts using a Poisson distribution assumption. In addition, they introduce process noise which is normally distributed. Subsequently, the data is cleaned by applying the Kalman filter. The conditional distribution of the vehicle counts, however, is potentially complex[3)], hence the assumptions of a fully parametric model for the time series properties as well as the count distribution may be misleading. Moreover, the assumption that the process noise is normally distributed could be violated as well. As can be derived from Figure 2.1, minute to minute roadsensor data can behave quite erratic. Taking this fact into account, assuming that this process noise is normally distributed is counter intuitive. It is therefore useful to filter and estimate the conditional PDF using methods that do not rely on strict assumptions, such as the neural network methodology we present in this paper.

3 Related Literature

In this section we summarize the relevant literature in relation to the neural network approach we propose. We first focus on the PDF and CDF estimation methods in general. We next comment on the related filtering literature that will potentially benefit from the constructed PDF using our method.

Empirical PDF estimation: Probability Density Function (PDF) estimation on low dimensions has been analyzed for a long time, both parametrically and non-parametrically, see Cramer (1946). Parametric density estimation assumes a specific distribution for the data, and the parameters of this distribution are then estimated, using e.g. the maximum likelihood approach. A conventional distribution used for this purpose is the normal distribution, while the underlying DGPs are potentially differently distributed, see Magnus (2007), Eliason (1993) and Cramer (1989).

To tackle the above difficulty, non-parametric methods have also been developed and applied for PDF estimation. The non-parametric methods do not impose a distributional assumption, but instead determine the distribution properties based on the data itself. There are different ways to summarize distribution properties, in other words a PDF. A widely used example is the histogram. The more advanced successor of the histogram is the kernel density estimator. The link between the data and the estimated PDF based on KDE, relies on the choice of a kernel to construct a PDF. The main advantage of KDE over the histogram is the unnecessity of choosing an endpoint of the bin. Disadvantages of the KDE are the poor performance on high dimensions and the high dependence on the specification of bandwidth. Moreover, on high dimensions it is even more difficult to determine the bandwidth [4)]. The multivariate kernel density estimation evaluated at

[3)] The behavior of vehicle counts is different at given points in time.

[4)] Rules of thumb are created for this inconvenience, such as the Silverman's rule, see Silverman (2018). However, these rules are usually only implied for univariate cases.

$\mathbf{z} \in \mathbb{R}^N$ at a certain time t, is formulated as:

$$\hat{f}_{\mathbf{H}}(\mathbf{z}) = \frac{1}{T}\sum_{t=1}^{T} K_{\mathbf{H}}(\mathbf{z} - \mathbf{x}_{\cdot t}) \tag{1}$$

where $K_{\mathbf{H}}(\mathbf{z}) = |\mathbf{H}|^{-\frac{1}{2}} K(\mathbf{H}^{-\frac{1}{2}}\mathbf{z})$

Let x_{nt} denote N variables $n = 1, \dots, N$ for which T replicates are observed $t = 1, \dots, T$. In our example and in the case of signal extraction, t is typically a time dimension, but it might also denote just replicates of a variable. These observations are collected in an $N \times T$ matrix $\mathbf{X}$, where the rows $\mathbf{x}_{n\cdot} = (x_{n1}, x_{n2}, \dots, x_{nT})$ for $n = 1, \dots, N$ consist of the n^{th} variable and columns $\mathbf{x}_{\cdot t} = (x_{1t}, x_{2t}, \dots, x_{Nt})'$ for $t = 1, \dots, T$ consist of the t^{th} observation of the N variables. In (1), $K_{\mathbf{H}}$ is a multivariate density kernel. In addition, a positive definite $N \times N$ matrix $\mathbf{H}$ determines the amount of smoothing parameters within the kernel $K_{\mathbf{H}}$. The number of these smoothing parameters increases with N, hence the calculation is computationally expensive for large N depending on the specified kernel.

Next to these two approaches, there exists a recently developed third approach, the Artificial Neural Network (ANN) approach[5]. It attempts to obtain the best of both worlds: no assumptions are made a priori and the extension to multiple dimensions is straightforward without contemplating with issues as memory and time. Specifically, non-parametric and semi-parametric density techniques that use ANNs to estimate the PDF of an underlying DGP are investigated in the literature. Specifically, Multilayer Perceptrons (MLPs) are used, feeding forward from the input layer to the output layer[6].

There are two distinct methods within this approach. Trentin et al. (2018) directly estimate the PDF by neural networks. The second method is to estimate the CDF by neural networks and from this to derive the PDF as Magdon-Ismail and Atiya (2002) and Zhang (2018) do. This paper follows the non-parametric estimation method of Magdon-Ismail and Atiya (2002). Zhang (2018) uses a similar technique. He however, argues that also the PDF of non-smooth distributions, next to continuous distributions, can be estimated by imposing a different activation function. He also shows the analytical derivative of a simple case. The proposed method extends this literature by showing the analytical derivatives of any model. That is, the MLP can consist of as many hidden layers as desired, as opposed to Magdon-Ismail and Atiya (2002), Zhang (2018) and Trentin et al. (2018), who use only one hidden layer. Next to the number of hidden layers, the number of hidden neurons complete the specification of the MLP or in other words the model. Model selection can be executed in several ways such as random search or a certain MLP can be chosen at random. Magdon-Ismail and Atiya (2002), Trentin et al. (2018) and Zhang (2018) fix the number of hidden layers to 1, such that only the number of hidden neurons need to be specified. Note however, when applying a nonlinear activation function, using more hidden layers increases the ability to estimate a highly nonlinear distribution of the underlying DGP. Magdon-Ismail and Atiya (2002) and Zhang (2018) do not explain why they use a certain structure. Trentin et al. (2018) train a network several times with a different number of neurons, then they pick one amongst these according to a log-likelihood criterion. This slows down the process of model selection heavily though. In this paper a novel model selection procedure, which may be computationally more efficient is proposed as an alternative.

5) Also referred to as neural networks.
6) When referring to the neural networks model itself, either the term model or MLP is used.

By using neural networks, the input variables need to be labeled such that it can be used for training. Trentin et al. (2018) construct these labels like the KDE does with some alterations on bandwidth selection and bias prevention. They show that their method, called Parzen Neural Networks (PNN), is less sensitive to bandwidth specification than KDE is. In addition, they overcome other shortcomings of KDE such as memory issues. However, it is more natural to construct labels for the CDF values instead for PDF values. More importantly, this is less sensitive to statistical error, since the integral to construct a PDF from a histogram can be regarded as a regularizer, as Magdon-Ismail and Atiya (2002) argue amongst other reasons. Moreover, they show density estimation in the multivariate setting as well. Note that as more variables N are used, the more hyper parameters need to be specified for the bandwidth for Trentin et al. (2018)'s approach, the $N \times N$ matrix **H** and the initial bandwidth. However, as this bandwidth needs to be selected empirically, choosing this value in a multivariate setting is even more difficult. More importantly, Trentin et al. (2018) assume that the underlying N variables need to be independent. The method proposed for density estimation in this paper, does not imply any assumptions on these relationships.

This paper extends the literature of CDF estimation by neural networks. Instead of numerically deriving the PDF from the obtained CDF, analytical derivations are shown for any specified model. That is, the MLP can consist of as many hidden layers as desired, contrary to Magdon-Ismail and Atiya (2002), Zhang (2018) and Trentin et al. (2018). A novel approach for model selection is shown which is computational time efficient opposed to Trentin et al. (2018). Moreover, the PDF estimation works for any correlated variables in the multivariate setting. Trentin et al. (2018) also performs multivariate density estimation, conditioned that the variables are independent though.

Relation to filtering: The method we propose does not specifically model time-varying properties, but it is potentially an input for existing filtering methods that take the time dimension of the data into account. A conventional way to model time-varying properties of the data is to use State Space Models and the Kalman Filter. The applicability of the Kalman filter, however, relies on distributional assumptions on the state variable. In most cases, the disturbance terms of state variables is assumed to be normally distributed, hence the standard Kalman filtering and recursion steps can be applied. Puts et al. (2018) follow a similar approach for analyzing road sensor data and assumes that the state variable, which is the parameter of the Poisson distribution follows a random walk with a normal distribution process error. When applying the Kalman filter to certain cases, this Gaussianity assumption can be quite restrictive though. The Kalman filter can only be used when the estimated PDFs are normally distributed. If the latent state variable is differently distributed, the Kalman filter would incorrectly assess its data generating process (DGP). Think about the mobility of the Dutch population tracked by mobile phone data or other trackers like GPS systems or the number of kilometers driven by the Dutch population on the highways in the Netherlands. It would be obvious to assume that these events are not normally distributed. An alternative filter that does not assume Gaussianity is the particle filter. This filter can use any non-Gaussian PDF function as input for the prediction and update recursions. This filter is also less restrictive in the sense of tracking the nonlinear relation between the observed time series and the true underlying hidden state. Therefore using PDFs constructed by the proposed method as inputs for this filter seems to be more realistic when applying it to certain applications.

4 Methodology

We propose a methodology to obtain the cumulative distribution function (CDF) and probability density function (PDF) of univariate or multivariate random variables using neural networks. The output of the proposed neural networks approach is the estimated empirical CDF. The PDF is obtained as the analytical derivative of the estimated CDF by neural networks with respect to the input variables. To the best of our knowledge, such analytical derivatives for a general neural networks approach for univariate and multivariate PDF estimation has not been considered in the literature. The main advantage of our approach compared to the related literature is reducing the approximation error in PDF estimation. Specifically, the approximation error in the second step of obtaining the PDF from the CDF output is smaller when using analytical derivatives instead of numerical derivatives as in Magdon-Ismail and Atiya (2002). In this section we present the proposed neural networks approach together with a novel method for model selection and the analytical derivatives for obtaining the PDF.

The constructed PDFs using our methodology can be used, e.g., as input to a particular Bayesian filter. This filter would impute any missing observations or smooth out outliers of a time series data set. Note that the considered time series needs to be stationary. More specifically, conditional PDFs on previous time t need to be constructed. This follows the idea of a Markov Chain where the true state depends on the state attained at previous time t. However, the procedure of constructing the underlying PDFs is always treated in the two-step procedure described below.

4.1 CDF Estimation using Neural Networks

In our application, the input layer, $x_{.t}$, consists of $N-$variate random variables and the output layer, $\hat{y}_t$ is the approximate empirical CDF of these variables at time t, $t = 1, \dots, T$. We have N input neurons, M hidden neurons for each hidden layer and H hidden layers. Let $W^{[q]}$ be the matrix of weights, where q represents the hidden layer of the neural network[7]. This implies that $W^{[0]}$ is the $N \times M$ matrix of weights from N input neurons in the input layer to M hidden neurons in hidden layer 1. $W^{[q]}$ for $q = 1, \dots, H-1$ is the $M \times M$ matrix of weights from layer q to layer $q+1$. $W^{[H]}$ is the $M \times 1$ matrix of weights from hidden layer H to the output layer. Note that when only an entry of the weight matrix is referred to, this is denoted as $w_{ij}^{[q]}$, for some $i = 1, \dots, I$ and $j = 1, \dots, J$. $\mathbf{b}^{[q]}$ for $q = 0, 1, \dots, H-1$ is an $M \times 1$ vector of possible biases introduced in layer $q+1$ and $\mathbf{b}^{[H]}$ is the scalar bias introduced in the output layer. Such a generalization of a neural network structure for input $x_{.t}$ and output $\hat{y}_t$, with H hidden layers is called a Multilayer Perceptron (MLP). This is illustrated in Figure 4.1 [8].

[7] H hidden layers imply $H+1$ weight matrices and biases.

[8] We follow both signal processing and time series notation as in Section 3 since it is also in line with the motivating example in Section 2.

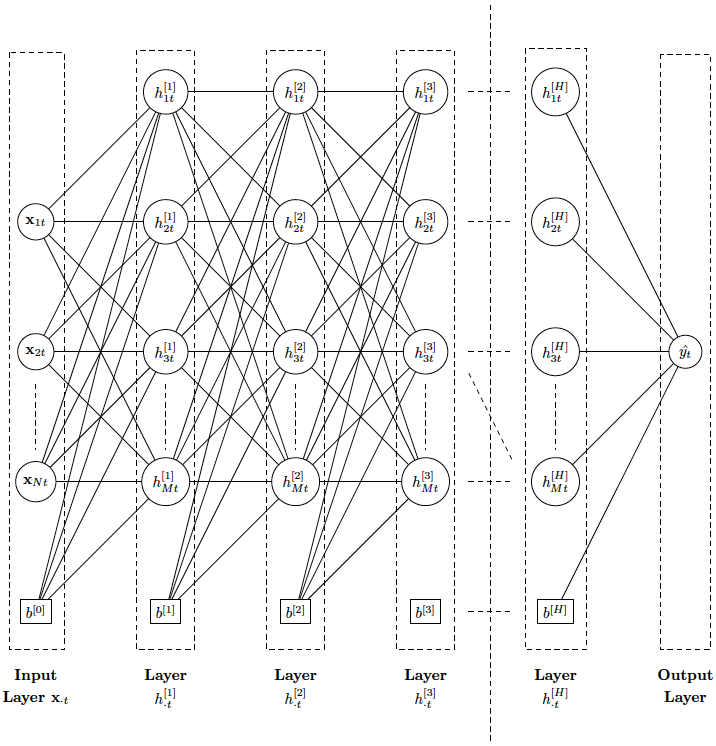


Figure 4.1 Multilayer Perceptron Visualization

The feed-forward algorithms for the layers of the neural network are defined as

$$\mathbf{h}_{\cdot t}^{[1]} = f\left(\mathbf{W}^{[0]'}\mathbf{x}_{\cdot t} + \mathbf{b}_t^{[0]}\right) \quad (2)$$

$$\mathbf{h}_{\cdot t}^{[q]} = f\left(\mathbf{W}^{[q-1]'}\mathbf{h}_{\cdot t}^{[q-1]} + \mathbf{b}_t^{[q-1]}\right), \quad \text{for } q = 2, \dots, H \quad (3)$$

$$\tilde{y}_t = \mathbf{W}^{[H]'}\mathbf{h}_{\cdot t}^{[H]} + \mathbf{b}_t^{[H]} \quad (4)$$

where $f(\cdot)$ is a function that takes as input an $M \times 1$ vector, which implies $f(\mathbf{h}_{\cdot t}^{[q]}) : \mathbb{R}^M \to \mathbb{R}^M$, with $q = 1, \dots, H$.

The neural networks with these input variables are trained to approximate target variable $\hat{y}_t$ for $t = 1, \dots, T$, which corresponds to the empirical CDF of input variables $x_{1t}, \dots, x_{Nt}$. We use the notation $\hat{y}_t$ to indicate such approximated CDF values at time t. This empirical CDF target variable can be approximated using grid-based methods over the multivariate input space, such as evenly spaced grid values γ_{ng} for $g = 1, \dots, G$ and $n = 1, \dots, N$, defined over the range of input variables. This leads to a grid with dimension $N \times G$. In general, the empirical CDF estimation is more accurate with an increase in sample size T and the obtained empirical CDF is smoother with an increase in the number of grids G. In the remainder of this paper, we set the number of grids $G > T$, with a step size of 0.001 to obtain smooth CDF estimates. The smallest grid value is equal to the smallest input variable observation. The largest grid value is the step size plus the largest input variable observation.

In case of a single input variable $N = 1$, we define the following grid values: $\gamma = (\gamma_1, \dots, \gamma_G)$ with $\gamma_g < \gamma_{g+1}$ for $g = 1, \dots, G-1$ where $\gamma_1, \dots, \gamma_G$ are evenly spaced, capturing the range of input variables. The target values $\hat{y} = (\hat{y}_1, \dots, \hat{y}_T)'$ are calculated

as follows:

$$\gamma_t = \underset{\gamma_g \in \{\gamma_1,\dots,\gamma_G\}}{\arg,\min} \left(x_{1t} - \gamma_g \mid x_{1t} \geq \gamma_g\right) \tag{5}$$

$$\hat{y}_t = \frac{\sum_{t'=1}^{T} I(x_{1t'} \leq \gamma_t)}{T} \tag{6}$$

for $t = 1, \dots, T$ and $I()$ is the indicator function that takes the value 1 if the argument is true and 0 otherwise. I.e. for each observation we choose the highest grid value that is smaller than the observation in (5). The empirical CDF of each output point is then calculated using the number of observations below the grid point, as in (6).

For the multivariate density problem the specified grid Γ is of dimension $\mathbb{R}^{N \times G}$ where a vector of grid values are constructed for each combination of input variables. For $\Gamma = (\gamma_{\cdot 1}, \dots, \gamma_{\cdot G})$ with $\gamma_{\cdot g} = (\gamma_{1g}, \dots, \gamma_{Ng})'$ for $g = 1, \dots, G$. The target values $\hat{y} = (\hat{y}_1, \dots, \hat{y}_T)'$ are calculated as follows:

$$\gamma_{\cdot t} = \underset{\gamma_{ng} \in \{\gamma_{n1},\dots,\gamma_{nG}\}}{\arg,\min} \left(x_{nt} - \gamma_{ng} \mid x_{nt} \geq \gamma_{ng}\right) \quad \forall n \tag{7}$$

$$\hat{y}_t = \frac{\sum_{t'=1}^{T} I(x_{nt'} \leq \gamma_{nt}, \forall n)}{T} \tag{8}$$

where the grids are defined for each input variable in (7). The empirical CDF of each output point, is calculated using the number of observations that are included in the N-dimensional grid box given in (8). Note that in (8), the indicator function takes the value of 1 for a certain t', only if $\forall n, \quad x_{nt'} \leq \gamma_{nt}$.

4.2 MLP Selection

The selection of an appropriate neural network is achieved by minimizing a loss function. The loss function represents the discrepancy between the estimated CDF $\tilde{y}_t$ by neural networks and the target CDF $\hat{y}_t$ represented by the empirical CDF calculated as in (6) or (8). For the simulation experiments in section 5, the least squares (L2) loss function $L2_{\widehat{\text{CDF}}}$ is used:

$$L2_{\widehat{\text{CDF}}} = \sum_{t=1}^{T} \left[\tilde{y}_t(x_{\cdot t}) - \hat{y}_t(x_{\cdot t})\right]^2 \tag{9}$$

The loss function calculates the squared difference between the target CDF $\hat{y}_t$ and the estimated CDF $\tilde{y}_t$. The loss function can be adjusted using penalties, e.g. to force monotonicity like Magdon-Ismail and Atiya (2002) do. In this way, the function that is estimated results in a valid non-decreasing CDF. For this penalty, tuning is required though. A sensitivity analysis is included in appendix E.

The target CDF $\hat{y}_t$ is an approximation of the true CDF y_t. To measure the discrepancy between the true CDF y_t and the estimated CDF $\tilde{y}_t$, the $L2_{\text{CDF}}$ loss function is used:

$$L2_{\text{CDF}} = \sum_{t=1}^{T} \left[y_t(x_{\cdot t}) - \tilde{y}_t(x_{\cdot t})\right]^2 \tag{10}$$

In a real data application, the true CDF is not known. Hence the feasible loss function is $L2_{\widehat{\text{CDF}}}$. The approximation of the true CDF y_t by the target CDF $\hat{y}_t$ is therefore important in order to obtain an adequate CDF estimate $\tilde{y}_t$. This relative loss obtained by the

empirical CDF $\hat{y}_t$ which approximates the true CDF y_t, is represented by the $L2_{\widehat{CDF}}$ loss function:

$$L2_{\widehat{CDF}} = \sum_{t=1}^{T} [y_t(x_{.t}) - \hat{y}_t(x_{.t})]^2 \tag{11}$$

Hence there are 3 distinct loss measurements. In case of simulated data, the loss can be obtained using the difference between the true CDF y_t and the estimated CDF $\tilde{y}_t$, represented by $L2_{CDF}$ in equation (10), while in real data applications, the true CDF is not known, thus the calculated loss is based on the difference between the target CDF $\hat{y}_t$ and the estimated CDF $\tilde{y}_t$ defined by $L2_{\widetilde{CDF}}$ in equation (9). The third loss measures the discrepancy of the approximation which determines the maximum performance of the neural networks. Note that the discrepancy between the estimated CDF $\tilde{y}_t$ and the true CDF y_t is not equal to adding the L2 losses of equations (9) and (11). When the trained neural network is differentiated resulting in the estimated PDF $\tilde{y}_t'$, the difference between the estimated PDF $\tilde{y}_t'$ and the true PDF y_t', is summarized in the $L2_{PDF}$ loss function:

$$L2_{PDF} = \sum_{t=1}^{T} [y_t'(x_{.t}) - \tilde{y}_t'(x_{.t})]^2 \tag{12}$$

For discrete distributions this is similarly defined and denoted as $L2_{PMF}$ loss. How to obtain the estimated PDF or PMF $\tilde{y}_t'$ is explained in the next subsection 4.3. Since we are only working with simulated data in section 5, all above losses are used to asses the simulation experiments. Specifically, means and corresponding standard errors of these loss functions over 100 simulation replications are shown.

Based on the feasible loss function $L2_{\widetilde{CDF}}$, model selection is performed by a novel approach. Model selection applies to the number of hidden layers and hidden neurons specified for the neural network, hence the MLP. Although some intuition behind this selection is available, which is discussed in appendix A, the exact number of hidden layers and neurons cannot be set in advance. Accordingly several potentially useful MLPs should be trained and compared by the $L2_{\widetilde{CDF}}$loss defined in equation (9). Training a neural network several times results in a computational burden though. A novel method is proposed in this paper to improve the computational efficiency during the model selection process.

The novel method is used to overcome this computational burden by training only one MLP instead of several potential MLPs. To determine the parameters of the MLP, several potential subMLPs, are combined into one large MLP. Note that all other (hyper)parameters are already specified, the only difference among these subMLPS is the number of hidden layers and the number of hidden neurons per layer. The computational burden mainly arises from the number of training iterations, neither from the numerous number of hidden layers nor hidden neurons. Therefore training a larger MLP is equally computational intensive as training a smaller MLP. Hence it is more time efficient to train only one larger MLP (more hidden layers and hidden neurons) for k iterations than to train 4 smaller MLP's separately for k iterations, just because $k < 4k$.

Training MLPs is more time efficient by enlarging the size (more hidden layers and hidden neurons) compared to enlarging the number of iterations

4.3 PDF Estimation using Analytical Derivatives

As the empirical CDF $\tilde{y}_t$ is estimated by neural networks, the corresponding PDF can be derived from this functional form. Using equations (2)–(4), the output of the neural network can be written as:

$$\tilde{y}_t = \mathbf{W}^{[H]'} f\left(\mathbf{W}^{[H-1]'} \cdots f\left(\mathbf{W}^{[1]'} f\left(\mathbf{W}^{[0]'} \mathbf{x}_{\cdot t} + \mathbf{b}_t^{[0]}\right) + \mathbf{b}_t^{[1]}\right) \cdots + \mathbf{b}_t^{[H-1]}\right) + \mathbf{b}_t^{[H]} \quad (13)$$

$$\tilde{y}_t = f\left(\mathbf{W}^{[H]'} \cdots f\left(\mathbf{W}^{[1]'} f\left(\mathbf{W}^{[0]'} \mathbf{x}_{\cdot t} + \mathbf{b}_t^{[0]}\right) + \mathbf{b}_t^{[1]}\right) \cdots + \mathbf{b}_t^{[H]}\right) \quad (14)$$

which is a composition of several nested activation functions in each hidden layer, either with an activation function on the output layer (14) or not (13). The joint PDF is obtained by differentiating (13) or (14) w.r.t. N input variables. Therefore an activation function that is N differentiable needs to be chosen. Potential activation functions $f(\cdot)$ that are considered are sigmoid functions [9]. Examples of possible activation functions $f(\cdot)$ are the logistic function $\sigma(\cdot)$ and the hyperbolic tangent function denoted as $\tau(\cdot)$, see appendix B for details of the derivations of these functions. Derivatives are conducted using $\sigma(\cdot)$, but $\tau(\cdot)$ is incorporated similarly. The input of these activation functions is any linear combination of $x_{\cdot t}$ together with weights (and biases) nested with previous layers, which implies $\tau(\mathbf{h}_{\cdot t}^{[q]}) : \mathbb{R}^M \to \mathbb{R}^M$ and $\sigma(\mathbf{h}_{\cdot t}^{[q]}) : \mathbb{R}^M \to \mathbb{R}^M$ defined as

$$\sigma(\mathbf{h}_{\cdot t}^{[q]}) = \frac{1}{1 + \exp\left(-\mathbf{h}_{\cdot t}^{[q]}\right)}. \quad (15)$$

Since equations (13) and (14) are a combination of nested functions, $\tilde{y}_t$ can also be represented by:

$$\tilde{y}_t = g^{[H]} \circ f \circ g^{[H-1]} \circ f \circ \cdots \circ f \circ g^{[1]} \circ f \circ g^{[0]} \quad (16)$$

$$\tilde{y}_t = f \circ g^{[H]} \circ f \circ g^{[H-1]} \circ f \circ \cdots \circ f \circ g^{[1]} \circ f \circ g^{[0]} \quad (17)$$

where $f(\cdot)$ represents the nonlinear activation function and $g^{[q]}(\cdot)$ represents the linear function at hidden layer q, combining weights and variables $\mathbf{x}_{\cdot t}$ or $\mathbf{h}_{\cdot t}^{[q-1]}$. Equation (16) refers to a neural network without an activation function that is imposed on the output layer and equation (17) refers to a neural network with an activation function imposed on the output layer.

To acquire the PDF function, the estimated CDF function $\tilde{y}$ is differentiated w.r.t. all N input variables denoted as:

$$\tilde{y}_t' = \frac{\partial^N}{\partial x_{1t} \ldots \partial x_{Nt}} \tilde{y}_t\left(\mathbf{x}_{\cdot t}\right) \quad (18)$$

Specifically, differentiation of the neural network starts from the right side of equation (16) and (17) and ends at the left side of the equation, where each step treats one composite function at a time. The subsequent step takes this resulting composite function as one function which forms a new composite function together with the next function. In this way, differentiation is performed by tackling composite functions subsequently. Each step which tackles one composite function, calculates all partial derivatives and the joint derivative. For example, for 3 input neurons, there are 6 partial derivatives and 1 joint derivative. To obtain the N^{th} derivative of these composite functions, denoted as in equation (18), the chain rule is used. Faà di Bruno (1855) gives a generalization of the chain rule for this higher derivative, which was originally proposed

[9] The popular rectified linear unit (ReLU) activation function is not used, since higher derivatives are zero. This implies that a lot of information would be lost in order to obtain the joint PDF.

by Arbogast (1800) 50 years earlier. In this way, differentiating nested composite functions w.r.t. N variables can be tracked in a structured way. Hence to differentiate this composition, Faà di Bruno's differentiation method is used, see Hardy (2006) for a more modern approach. When computing the N^{th} derivative, N input variables are partitioned in all possible combinations of subsets. For N input variables there are P different partitions π_p for $p = 1, \dots, P$. In each partition π_p the N variables are divided in a unique way over subsets B_{pk}. The number of subsets in partition π_p is represented by the cardinality of π_p, denoted as $|\pi_p|$. So B_{pk} denotes the k-th subset of partition π_p, where $k = 1, \dots, |\pi_p|$. Accordingly, each subset B_{pk} has cardinality $|B_{pk}|$, which is the number of variables in a subset or the size of a subset. In each partition, each variable occurs only one time in one of the $|\pi_p|$ subsets of this partition. Thus for each π_p, all subsets B_{pk} contain all input variables N combined in a different way than other partitions $\pi_{p'}$, where $p \neq p'$. The intersection of all subsets of a particular partition are empty. Thus $\cup_{k=1}^{|\pi_p|} B_{kp} = N$ and $\cap_{k=1}^{|\pi_p|} B_{kp} = \emptyset$. See chapter 9 of Wilson and Watkins (2013) for more on partitions.

The nested composite function of equations (13)–(17) consists of two type of functions which are non-linear and linear functions. These are alternately, outer and inner functions, $f_1(\cdot)$ and $f_2(\cdot)$, respectively. When considering only one composite function, the following representation of Faà di Bruno (FDB) is used:

$$\frac{\partial^N}{\partial x_{1t} \cdots \partial x_{Nt}} f_1(f_2(\mathbf{x}_{\cdot t})) = \sum_{p=1}^{P} \frac{\partial^{|\pi_p|} f_1(f_2(\mathbf{x}_{\cdot t}))}{\partial f_2(\mathbf{x}_{\cdot t})^{|\pi_p|}} \prod_{k=1}^{|\pi_p|} \frac{\partial^{|B_{pk}|} f_2(\mathbf{x}_{\cdot t})}{\prod_{n \in B_{pk}} \partial x_{nt}} \tag{19}$$

$$\equiv \sum_{p=1}^{P} D_{1p} \prod_{k=1}^{|\pi_p|} D_{2k}$$

$\partial^{|\pi_p|}$ refers to the $|\pi_p|^{th}$ derivative. Similarly, $\partial^{|B_{pk}|}$ refers to the $|B_{pk}|^{th}$ partial derivative. For notational reasons, D_{1p} represents $\frac{\partial^{|\pi_p|} f_1(f_2(\mathbf{x}_{\cdot t}))}{\partial f_2(\mathbf{x}_{\cdot t})^{|\pi_p|}}$ for partition π_p and D_{2k} represents $\frac{\partial^{|B_{pk}|} f_2(\mathbf{x}_{\cdot t})}{\prod_{n \in B_{pk}} \partial x_{nt}}$ for subset B_{pk} of partition π_p. Hence for each partition π_p, there is one D_1 and $|\pi_p|$ number of D_2.

As mentioned before, differentiation is performed by differentiating from right to left of equation (16) or (17), treating one composite function in each step. Depending on which composite function in the neural network is treated, either a non-linear function is the outer function and a linear function is the inner function $f(g^{[q]}(\cdot))$ or vice versa $g^{[q]}(f(\cdot))$. There are 3 different steps that are applied. The first step is called the ***input step***, which deals with the composite function of the input layer to the first hidden layer. This implies that $f_1(\cdot) : \mathbb{R}^N \to \mathbb{R}^M$ and $f_2(\cdot) : \mathbb{R}^N \to \mathbb{R}^M$, since this step operates between the input layer consisting of N input neurons and the first hidden layer consisting of M hidden neurons. The second step is called the ***hidden step***. This step treats differentiation of the composite functions within the hidden layers of the neural network. This entails that $f_1(\cdot) : \mathbb{R}^M \to \mathbb{R}^M$ and $f_2(\cdot) : \mathbb{R}^M \to \mathbb{R}^M$, since this step operates in between hidden layers that consist of M hidden neurons. The hidden step tackles two different types, treated subsequently. Type 1 accommodates differentiation when the outer function is linear, that is $g^{[q]} \circ f$. Type 2 corresponds to differentiation when the outer function is nonlinear, that is $f \circ g^{[q]}$. Note that when the outer function is linear, all higher derivatives D_1 are 0 and when the inner function is linear, all higher derivatives of D_2 are 0. This implies that many parts of the derivatives are eliminated.

The third step is called the ***output step***, performing differentiation from the last hidden layer to the output layer. This means that $f_1(\cdot) : \mathbb{R}^M \to \mathbb{R}^1$ and $f_2(\cdot) : \mathbb{R}^M \to \mathbb{R}^1$. This step is also divided into 2 types, which depends on imposing an activation function on the output layer or not. That is, type 1 and type 2, which correspond to $g \circ f$ and $f \circ g$, respectively. If there is an activation function imposed on the output layer, then type 1 and 2 are treated subsequently.

All steps are summarized in algorithm 1. Derivations are denoted on neuron-level. That is, derivations of every hidden layer are computed for each hidden neuron. Therefore the multivariate functions of equations (13)–(17) turn into univariate functions of hidden neuron m, $f(\cdot)_m : \mathbb{R}^1 \to \mathbb{R}^1$ and $g^{[q]}(\cdot)_m : \mathbb{R}^1 \to \mathbb{R}^1$ which are non-linear and linear functions, respectively. In each step, the FDB derivative is computed for all possible combinations of N input variables. These Faà di Bruno derivatives in each step, depicted by the grey boxes in Figure 4.2, are used as D_{2k} for each $k \in \pi_p$ for all p in the next step. Hence in each step, equation (19) is repeated for all combinations of input variables N.

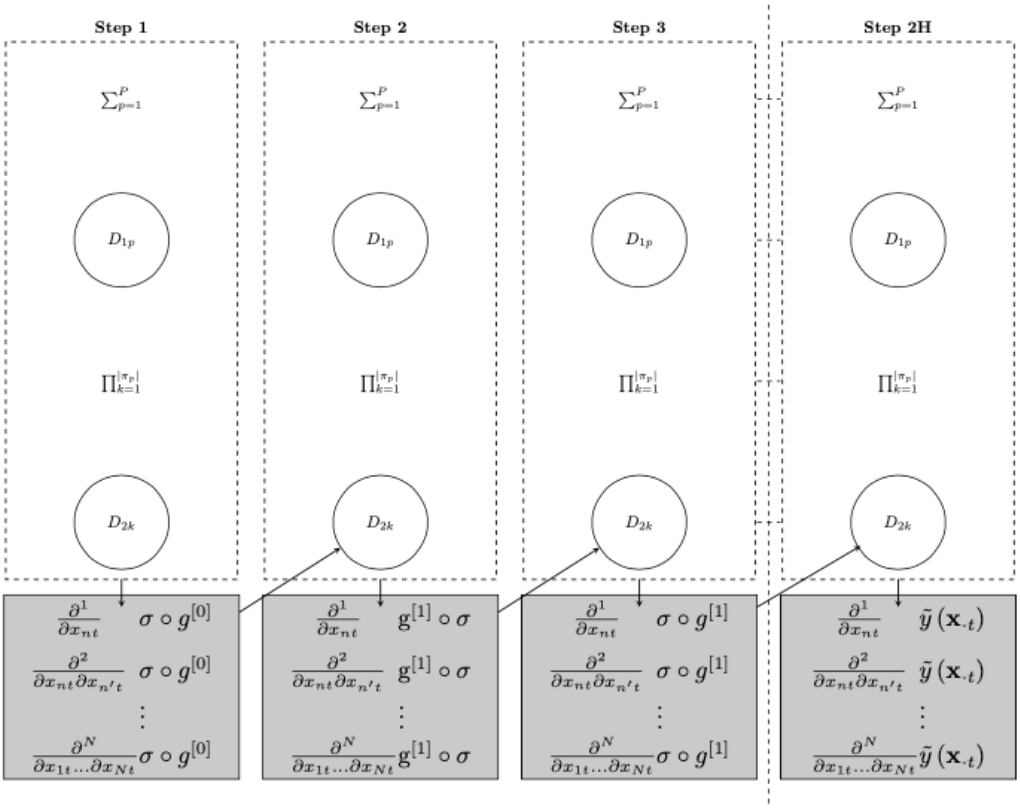


Figure 4.2 Function of nested composite functions combined with Faà di Bruno's method

The FDB derivatives w.r.t. all combinations of input variables of each step, that is the FDB derivatives depicted in the grey box of Figure 4.2, are referred to as the complete (partial) derivatives of each step in the example illustrated later on. These are different from the final derivatives which are the derivatives resulting from the output step. The final derivatives are the aimed PDF estimates acquired from the CDF $\tilde{y}_t$. The difference between two subsequent steps is that the composite function $f_1(f_2(\cdot))$ in the first step changes to $f_2(f_1(\cdot))$ in the subsequent step. That is, the outer function of the previous step turns into the inner function of the next step. In other words, the FDB derivatives for all combinations of input variables N of the previous step turn into D_{2k} for all

partitions π_p in the subsequent step. In total there are $2 + 2(H-1) = 2H$ steps, the input step, output step and $2(H-1)$ hidden steps.

Algorithm 1 Differentiation of the functional form of the empirical CDF $\tilde{y}_t$

1: Let $\tilde{y}_t$ be the nested function composed of nonlinear activation functions $f(\cdot)$ and linear functions $g^{[q]}(\cdot)$ where q represents the hidden layer of the neural network.

$$\tilde{y}_t = g^{[H]} \circ f \circ g^{[H-1]} \circ f \circ \cdots \circ g^{[1]} \circ f \circ g^{[0]}$$

or

$$\tilde{y}_t = f \circ g^{[H]} \circ f \circ g^{[H-1]} \circ f \circ \cdots \circ g^{[1]} \circ f \circ g^{[0]}$$

Note that each linear function consists of different parameter weights $W^{[q]}$ and biases $b^{[q]}$.

2: Input step: Differentiate $f \circ g^{[0]} = f\left(W^{[0]'} x_{\cdot t} + b_t^{[0]}\right)$ for each neuron m, where the outer function is a nonlinear activation function $f(\cdot) : \mathbb{R}^M \to \mathbb{R}^M$ and the inner linear function $g^{[q]}(\cdot) : \mathbb{R}^M \to \mathbb{R}^M$ which is the linear combination of input variables and weights, resulting in:

$$\frac{\partial (f \circ g^{[0]})_m}{\partial x_{nt}} \quad \frac{\partial^2 (f \circ g^{[0]})_m}{\partial x_{nt} \partial x_{n't}} \quad \cdots \quad \frac{\partial^N (f \circ g^{[0]})_m}{\partial x_{1t} \cdots \partial x_{Nt}} \quad \text{for all n and for all n} \neq \text{n'}$$

where the derivations are treated univariately for each hidden neuron m in hidden layer 1, that is $(f \circ g^{[0]})_m : \mathbb{R}^1 \to \mathbb{R}^1$. Similarly for each hidden neuron m in hidden layer $q+1$, $(f \circ g^{[q]})_m : \mathbb{R}^1 \to \mathbb{R}^1$.

3: **while** q < H **do**

4: Hidden step, type 1: Differentiate the composite function that connects two subsequent hidden layers where the outer function is linear: $g^{[q]} \circ f$ for each neuron m of hidden layer $q+1$. Note that each neuron in layer $q+1$ is connected to all neurons in hidden layer q or the input layer which results in a long sum of derivatives.

$$\frac{\partial (g^{[q]} \circ f)_m}{\partial x_{nt}} \quad \frac{\partial^2 (g^{[q]} \circ f)_m}{\partial x_{nt} \partial x_{n't}} \quad \cdots \quad \frac{\partial^N (g^{[q]} \circ f)_m}{\partial x_{1t} \cdots \partial x_{Nt}} \quad \text{for all n and for all n} \neq \text{n'}$$

5: Hidden step, type 2: Finish the hidden step by differentiating two subsequent hidden layers where the outer function is nonlinear $f \circ g^{[q]}$ for each neuron m of hidden layer $q+1$.

$$\frac{\partial (f \circ g^{[q]})_m}{\partial x_{nt}} \quad \frac{\partial^2 (f \circ g^{[q]})_m}{\partial x_{nt} \partial x_{n't}} \quad \cdots \quad \frac{\partial^N (f \circ g^{[q]})_m}{\partial x_{1t} \cdots \partial x_{Nt}} \quad \text{for all n and for all n} \neq \text{n'}$$

6: **return to step 3**

7: Output step: Now the final layer needs to be differentiated for each neuron m of layer H: $g^{[H]} \circ f$ likewise:

$$\frac{\partial (g^{[H]} \circ f)_m}{\partial x_{nt}} \quad \frac{\partial^2 (g^{[H]} \circ f)_m}{\partial x_{nt} \partial x_{n't}} \quad \cdots \quad \frac{\partial^N (g^{[H]} \circ f)_m}{\partial x_{1t} \cdots \partial x_{Nt}} \quad \text{for all n and for all n} \neq \text{n'}$$

If the output activation function is used, subsequently the following type has to be performed, for each neuron m of layer H:

$$\frac{\partial (f \circ g^{[H]})_m}{\partial x_{nt}} \quad \frac{\partial^2 (f \circ g^{[H]})_m}{\partial x_{nt} \partial x_{n't}} \quad \cdots \quad \frac{\partial^N (f \circ g^{[H]})_m}{\partial x_{1t} \cdots \partial x_{Nt}} \quad \text{for all n and for all n} \neq \text{n'}$$

Depending on the imposed activation output function, we obtain, next to the partial derivatives, the following joint derivative for each neuron m of the last hidden layer H:

$$\sum_{m=1}^{M} \frac{\partial^N (g^{[H]} \circ f)_m}{\partial x_{1t} \cdots x_{Nt}} \qquad \text{or} \qquad \sum_{m=1}^{M} \frac{\partial^N (f \circ g^{[H]})_m \circ f}{\partial x_{1t} \cdots x_{Nt}}$$

Note that also partial derivatives are obtained by summing over all M.

We illustrate this PDF estimation procedure with two examples. Firstly, suppose the neural network has two input neurons and one hidden layer with one hidden neuron. Then the following is the representation of the output:

$$\tilde{y}_t = w_{11}^{[1]} \cdot f(w_{11}^{[0]} \cdot x_{1t} + w_{21}^{[0]} \cdot x_{2t} + b_t^{[0]}) + b_t^{[1]}$$

This representation consists of 3 nested functions: $\tilde{y}_t = g^{[1]} \circ f \circ g^{[0]}$. The last two functions $f \circ g^{[0]}$ are considered first. Next these two functions are seen as one function, represented as f. Then the last composite function is treated, $g^{[1]} \circ f$, in order to obtain the joint PDF, or in other words the 2^{nd} derivative of this output. Since there are 2 input variables, we consider the following set of partitions$\{\{\{1,2\}\},\{\{1\},\{2\}\}\}$. Every π_p gives:

π_1: $\{\{1,2\}\}$ has $|\pi_1| = 1$ and $|B_{11}| = 2$
π_2: $\{\{1\},\{2\}\}$ has $|\pi_2| = 2$ and $|B_{2k}| = 1 \quad$ for $\quad k = 1,2$

Considering the first composite function $f \circ g^{[0]}$ and following Faà di Bruno, this results in the following differentiation to obtain the 2^{nd} derivative of $f \circ g^{[0]}$:

$$\begin{aligned}\frac{\partial^2}{\partial x_{1t}\partial x_{2t}} f(g^{[0]}(\mathbf{x}_{\cdot t})) = &\frac{\partial f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})} \cdot \frac{\partial^2 g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{1t}\partial x_{2t}} \\ &+ \frac{\partial^2 f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})^2} \cdot \frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{1t}} \cdot \frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{2t}}\end{aligned} \tag{20}$$

As illustrated in Figure 4.2, also the partial derivatives of all combinations of input variables N are derived. In this case these are the first derivatives w.r.t. x_{nt} with $n = 1,2$:

$$\frac{\partial}{\partial x_{nt}} f(g^{[0]}(\mathbf{x}_{\cdot t})) = \frac{\partial f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})} \cdot \frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{nt}} \tag{21}$$

As there are many nested functions, partial derivatives are needed to complete FDB's differentiation in subsequent steps of the derivation. Equations (20) and (21) are repeated in every step in order to obtain the joint PDF. Every step treats a composite function, which is a part of the nested expression as result of the neural network. This implies that 3 derivatives are calculated for each layer in order to acquire the joint PDF of 2 input variables, $\frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{1t}}$, $\frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{2t}}$ and $\frac{\partial^2 g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{1t}\partial x_{2t}}$. As is evident, 1^{st} derivatives are needed to obtain higher derivatives, in this case the 2^{nd} derivative of $g^{[1]} \circ f$ is then:

$$\begin{aligned}\frac{\partial^2}{\partial x_{1t}\partial x_{2t}} g^{[1]}(f(\mathbf{x}_{\cdot t})) = &\frac{\partial g^{[1]}(f(\mathbf{x}_{\cdot t}))}{\partial f(\mathbf{x}_{\cdot t})} \cdot \frac{\partial^2 f(\mathbf{x}_{\cdot t})}{\partial x_{1t}\partial x_{2t}} \\ &+ \frac{\partial^2 g^{[1]}(f(\mathbf{x}_{\cdot t}))}{\partial f(\mathbf{x}_{\cdot t})^2} \cdot \frac{\partial f(\mathbf{x}_{\cdot t})}{\partial x_{1t}} \cdot \frac{\partial f(\mathbf{x}_{\cdot t})}{\partial x_{2t}} \\ &= \frac{\partial g^{[1]}(f(\mathbf{x}_{\cdot t}))}{\partial f(\mathbf{x}_{\cdot t})} \cdot \left[\frac{\partial f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})} \cdot \right. \\ &\left. \frac{\partial^2 g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{1t}\partial x_{2t}} + \frac{\partial^2 f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})^2} \cdot \frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{1t}} \cdot \frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{2t}}\right] \\ &+ \frac{\partial^2 g^{[1]}(f(\mathbf{x}_{\cdot t}))}{\partial f(\mathbf{x}_{\cdot t})^2} \cdot \frac{\partial f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})} \cdot \\ &\frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{1t}} \cdot \frac{\partial f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})} \cdot \frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{2t}}\end{aligned} \tag{22}$$

where $\frac{\partial^2 f(\mathbf{x}_{\cdot t})}{\partial x_{1t}\partial x_{2t}}$ is substituted with equation (20) and $\frac{\partial f(\mathbf{x}_{\cdot t})}{\partial x_{1t}}$, $\frac{\partial f(\mathbf{x}_{\cdot t})}{\partial x_{2t}}$ with equation (21). Now the final joint PDF is obtained. To obtain the final marginal PDF, Faà di Bruno is

executed w.r.t. input variable n:

$$\frac{\partial}{\partial x_{nt}} g^{[1]}(f(\mathbf{x}_{\cdot t})) = \frac{\partial g^{[1]}(f(\mathbf{x}_{\cdot t}))}{\partial f(\mathbf{x}_{\cdot t})} \cdot \frac{\partial f(\mathbf{x}_{\cdot t})}{\partial x_{nt}} \\ = \frac{\partial g^{[1]}(f(\mathbf{x}_{\cdot t}))}{\partial f(\mathbf{x}_{\cdot t})} \cdot \frac{\partial f(g^{[0]}(\mathbf{x}_{\cdot t}))}{\partial g^{[0]}(\mathbf{x}_{\cdot t})} \cdot \frac{\partial g^{[0]}(\mathbf{x}_{\cdot t})}{\partial x_{nt}} \tag{23}$$

where $\frac{\partial f(\mathbf{x}_{\cdot t})}{\partial x_{nt}}$ is again replaced with equation (21). Each hidden neuron has a FDB derivative. As a consequence the final derivative is a sum over all these hidden neurons, as algorithm 1 states. The treated example has only one hidden neuron though. Therefore there is no sum in equations (22) and (23). Let's treat a more general case thoroughly. This example uses notation introduced in table 4.1.

$s^{(1)}$	$\sigma(1-\sigma)$	
$s^{(2)}$	$\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)$	
$s^{(3)}$	$\sigma(1-\sigma)^3 - 4\sigma^4(1-\sigma)^2 + \sigma^3(1-\sigma)$	
$\Delta^{[q]}_{mn}$	$\sum_{m'=1}^{M} w^{[q]}_{m'm} \frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{nt}}$	$n = 1,2,3$
$\Delta^{[q]}_{mnn'}$	$\sum_{m'=1}^{M} w^{[q]}_{m'm} \frac{\partial^2(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{nt}\partial x_{n't}}$	$n \neq n'$
$\Delta^{[q]}_{m123}$	$\sum_{m'=1}^{M} w^{[q]}_{m'm} \frac{\partial^3(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}$	

Table 4.1 Notation introduced in example with $N = 3$

The second example uses more than two neurons. Suppose the following neural network with three input neurons, M hidden neurons per layer and H hidden layers. In this example, the logistic function defined in equation (15) is used as activation function. This implies that the derivatives of the activation function $f(\cdot) = \sigma(\cdot)$ are derived by equation (B.1) in appendix B[10]. Since $N = 3$, the following set of partitions of all input variables is considered:

$$\{\{\{1,2,3\}\}, \{\{1\},\{2\},\{3\}\}, \{\{1,2\},\{3\}\}, \{\{1,3\},\{2\}\}, \{\{2,3\},\{1\}\}\}$$

Every π_p gives:

π_1: $\{\{1,2,3\}\}$ has $|\pi_1| = 1$ and $|B_{11}| = 3$
π_2: $\{\{1\},\{2\},\{3\}\}$ has $|\pi_2| = 3$ and $|B_{2k}| = 1 \quad \text{for} \quad k = 1,2,3$
π_3: $\{\{1,2\},\{3\}\}$ has $|\pi_3| = 2$ and $|B_{31}| = 2, |B_{32}| = 1$
π_4: $\{\{1,3\},\{2\}\}$ has $|\pi_4| = 2$ and $|B_{41}| = 2, |B_{42}| = 1$
π_5: $\{\{2,3\},\{1\}\}$ has $|\pi_5| = 2$ and $|B_{51}| = 2, |B_{52}| = 1$

See the following subsections for the input step, hidden step and output step for which

[10] For derivations with $f(\cdot) = \tau(\cdot)$, see equation (B.2) in appendix B.

in each step the following partial derivatives are estimated:

$$\frac{\partial}{\partial x_{nt}} f_1(f_2(\mathbf{x}_{\cdot t})) \quad \text{for all} \quad n = 1,2,3 \tag{24}$$
$$\frac{\partial^2}{\partial x_{nt}\partial x_{n't}} f_1(f_2(\mathbf{x}_{\cdot t})) \quad \text{for all} \quad n \neq n'$$
$$\frac{\partial^3}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} f_1(f_2(\mathbf{x}_{\cdot t}))$$

where $f(\cdot)$ and $g^{[q]}(\cdot)$ are the respective functions $f_1(\cdot)$ and $f_2(\cdot)$ which alternately form the inner and outer functions in every step. Note that if both n and n' are used, 3 different pairs are treated, $(n, n') = (1,2), (2,3), (1,3)$. Hence in this example, 7 derivatives are derived in each step. More importantly, FDB is performed w.r.t. input variables $N = 1$, $N = 2$ and $N = 3$ for the 3 equations in (24), respectively. Thus the first equation is derived 3 times ($N = 1$), the second equation is derived 3 times ($N = 2$) and the third equation is derived once ($N = 3$). Note that the input step and output step are performed only once while the hidden step $2(H - 1)$ times.

Input step: $\sigma \circ g^{[0]}$

This step is only performed when moving from the input layer to the first hidden layer. In particular, the first, second and third derivatives of the outer function w.r.t. the inner linear function are derived. This inner linear function is the combination of the input variables with $\mathbf{W}^{[0]}$ and $b^{[0]}$ to each hidden neuron m in the first hidden layer. Hence this is the m^{th} row of $g^{[0]} = \left[\mathbf{W}^{[0]'}\mathbf{x}_{\cdot t} + b^{[0]}\right]$. This means that the outer function $\sigma(\cdot)$ is a function of $g^{[0]}$, which will be denoted as σ for notational reasons. Following Faà di Bruno, the outer function needs to be differentiated for each π_p. Hence D_{1p} for each π_p gives, for each m in the first hidden layer:

$$\frac{\partial \sigma}{\partial g_m^{[0]}} = \sigma(1-\sigma) \equiv s^{(1)} \quad \text{for} \quad \pi_1$$
$$\frac{\partial^2 \sigma}{\partial g_m^{[0]^2}} = \sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \equiv s^{(2)} \quad \text{for} \quad \pi_3, \quad \pi_4 \quad \text{and} \quad \pi_5$$
$$\frac{\partial^3 \sigma}{\partial g_m^{[0]^3}} = \sigma(1-\sigma)^3 - 4\sigma^4(1-\sigma)^2 + \sigma^3(1-\sigma) \equiv s^{(3)} \quad \text{for} \quad \pi_2$$

where the logistic function $\sigma(\cdot)$ is defined in equation (15). The nonzero partial derivatives of the inner function w.r.t. the input variables represented by D_{2k} for all B_{pk}, for each hidden neuron m in the first hidden layer, are:

$$\frac{\partial g_m^{[0]}}{\partial x_{1t}} = w_{1m}^{[0]} \quad \frac{\partial g_m^{[0]}}{\partial x_{2t}} = w_{2m}^{[0]} \quad \frac{\partial g_m^{[0]}}{\partial x_{3t}} = w_{3m}^{[0]}$$

Note that every partial derivative of the linear function $g^{[0]}$ w.r.t. x_{nt}, with a derivative order higher than one is zero, thus the partial derivations of the input step, especially the joint derivative, are brief. Following Faà di Bruno defined in equation (19) and

equation (24), combining D_{1p} and D_{2k} gives for each neuron m:

$$\frac{\partial(\sigma \circ g^{[0]})_m}{\partial x_{nt}} = w_{nm}^{[0]} \frac{\partial \sigma}{\partial g_m^{[0]}} \quad \text{for all} \quad n = 1,2,3 \tag{25}$$
$$= w_{nm}^{[0]} \sigma(1-\sigma) \equiv w_{nm}^{[0]} s^{(1)}$$

$$\frac{\partial^2(\sigma \circ g^{[0]})_m}{\partial x_{nt} \partial x_{n't}} = w_{nm}^{[0]} w_{n'm}^{[0]} \frac{\partial^2 \sigma}{\partial {g_m^{[0]}}^2} \quad \text{for all} \quad n \neq n' \tag{26}$$
$$= w_{nm}^{[0]} w_{n'm}^{[0]} \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right]$$
$$\equiv w_{nm}^{[0]} w_{n'm}^{[0]} s^{(2)}$$

$$\frac{\partial^3(\sigma \circ g^{[0]})_m}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} = w_{1m}^{[0]} w_{2m}^{[0]} w_{3m}^{[0]} \frac{\partial^3 \sigma}{\partial {g_m^{[0]}}^3} \tag{27}$$
$$= w_{1m}^{[0]} w_{2m}^{[0]} w_{3m}^{[0]} \left[\sigma(1-\sigma)^3 - 4\sigma^4(1-\sigma)^2 + \sigma^3(1-\sigma)\right]$$
$$\equiv w_{1m}^{[0]} w_{2m}^{[0]} w_{3m}^{[0]} s^{(3)}$$

where m refers to the hidden node at which the derivation is executed, in other words the m^{th} row of $\sigma \circ g^{[0]} = \sigma\left[{W^{[0]}}' \mathbf{x}_t + b^{[0]}\right]$. These complete partial derivatives turn into D_{2k} in the next step since the next step estimates derivatives of the composite function $g^{[1]} \circ \sigma \circ g^{[0]}$ which is referred to as $g^{[1]} \circ \sigma$, as pointed out by Figure 4.2. In other words, these complete partial derivatives are the derivatives of the inner function in the next step.

Hidden step: $g^{[q]} \circ \sigma$ and $\sigma \circ g^{[q]}$

This step is performed when moving from hidden layer $q-1$ to the next hidden layer q, for $q = 1, \dots, H$ and it consists of two types of derivations to be performed adjacently for all inner hidden layers. The first type corresponds to the following composite function:

$$g^{[q]} \circ \sigma = {W^{[q]}}' \sigma(\cdots \sigma({W^{[0]}}' \mathbf{x}_t + b_t^{[0]}) \cdots) + b_t^{[q]} \tag{28}$$

The 3 derivatives of the outer linear function w.r.t. the inner nonlinear activation function, represented by D_{1p} for all π_p, in hidden layer q for each node m are:

$$\frac{\partial g_m^{[q]}}{\partial \sigma(g_{m'm}^{[q-1]})} = w_{m'm}^{[q]} \quad \text{for} \quad \pi_1$$

$$\frac{\partial^2 g_m^{[q]}}{\partial(\sigma(g_{m'm}^{[q-1]}))^2} = 0 \quad \text{for} \quad \pi_3, \quad \pi_4 \quad \text{and} \quad \pi_5$$

$$\frac{\partial^3 g_m^{[q]}}{\partial(\sigma(g_{m'm}^{[q-1]}))^3} = 0 \quad \text{for} \quad \pi_2$$

where m refers to the hidden node in layer q and m' refers to the hidden node in layer $q-1$. Hence these are the outer derivatives of the m^{th} row of equation (28). The nonzero partial derivatives of the inner nonlinear functions w.r.t. the input variables, that is the D_{2k} for all $k \in \pi_p$ for each p, are the findings from the previous step, as mentioned before and further illustrated in Figure 4.2. The inner function is denoted as

$(\sigma \circ g^{[0]})_m$ or $(\sigma \circ g^{[q]})_m$. Then D_{2k} for each $k \in \pi_p$ for all P gives:

$$\frac{\partial(\sigma \circ g^{[q]})_m}{\partial x_{nt}} = \frac{\partial \sigma}{\partial g_m^{[q]}} \frac{\partial g_m^{[q]}}{\partial x_{nt}} \qquad \text{for all} \quad n = 1, 2, 3$$

$$\frac{\partial^2(\sigma \circ g^{[q]})_m}{\partial x_{nt} \partial x_{n't}} = \frac{\partial \sigma}{\partial g_m^{[q]}} \frac{\partial^2 g_m^{[q]}}{\partial x_{nt} \partial x_{n't}} + \frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{nt}} \frac{\partial g_m^{[q]}}{\partial x_{n't}} \qquad \text{for all} \quad n \neq n'$$

$$\begin{aligned} \frac{\partial^3(\sigma \circ g^{[q]})_m}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} = & \frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{1t}} \frac{\partial^2 g_m^{[q]}}{\partial x_{2t} \partial x_{3t}} + \frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{2t}} \frac{\partial^2 g_m^{[q]}}{\partial x_{1t} \partial x_{3t}} \\ & + \frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{3t}} \frac{\partial^2 g_m^{[q]}}{\partial x_{1t} \partial x_{2t}} + \frac{\partial \sigma}{\partial g_m^{[q]}} \frac{\partial^3 g_m^{[q]}}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} \\ & + \frac{\partial^3 \sigma}{\partial {g_m^{[q]}}^3} \frac{\partial g_m^{[q]}}{\partial x_{1t}} \frac{\partial g_m^{[q]}}{\partial x_{2t}} \frac{\partial g_m^{[q]}}{\partial x_{3t}} \end{aligned}$$

These derivatives are conditional on the previous step. If the previous step is the input step, then equations (25), (26) and (27) are used. In other words, the derivatives for the first hidden layer are substituted with the values acquired in the input step:

$$\frac{\partial(\sigma \circ g^{[0]})_m}{\partial x_{nt}} = w_{nm}^{[0]} \sigma(1 - \sigma) \equiv w_{nm}^{[0]} s^{(1)} \qquad \text{for all} \quad n = 1, 2, 3$$

$$\begin{aligned} \frac{\partial^2(\sigma \circ g^{[0]})_m}{\partial x_{nt} \partial x_{n't}} &= w_{nm}^{[0]} w_{n'm}^{[0]} \left[\sigma(1 - \sigma)^2 - \sigma^2(1 - \sigma)\right] \qquad \text{for all} \quad n \neq n' \\ &\equiv w_{nm}^{[0]} w_{n'm}^{[0]} s^{(2)} \end{aligned}$$

$$\begin{aligned} \frac{\partial^3(\sigma \circ g^{[0]})_m}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} &= w_{1m}^{[0]} w_{2m}^{[0]} w_{3m}^{[0]} \left[\sigma(1 - \sigma)^3 - 4\sigma^4(1 - \sigma)^2 + \sigma^3(1 - \sigma)\right] \\ &\equiv w_{1m}^{[0]} w_{2m}^{[0]} w_{3m}^{[0]} s^{(3)} \end{aligned}$$

Note that in these equations $\frac{\partial \sigma}{\partial g_m^{[q]}} \frac{\partial^2 g_m^{[q]}}{\partial x_{nt} \partial x_{n't}}$, $\frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{1t}} \frac{\partial^2 g_m^{[q]}}{\partial x_{2t} \partial x_{3t}}$, $\frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{2t}} \frac{\partial^2 g_m^{[q]}}{\partial x_{1t} \partial x_{3t}}$, $\frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{3t}} \frac{\partial^2 g_m^{[q]}}{\partial x_{1t} \partial x_{2t}}$, $\frac{\partial \sigma}{\partial g_m^{[q]}} \frac{\partial^3 g_m^{[q]}}{\partial x_{1t} \partial x_{2t} \partial x_{3t}}$ are equal to 0. This is due to the input step of which the derivatives of the linear function $g^{[0]}$ contain zero higher derivatives. Any linear functions $g^{[q]}$, with $q > 0$, do not have any zero higher derivatives since these are implicit composite functions $g^{[q]} \circ \sigma \circ g^{[q-1]} \circ \cdots \circ \sigma \circ g^{[0]}$. This different case would be treated if the previous step is also a hidden step. Then the complete derivative of type 2 of the hidden step is used. That is, the derivatives of the $(q+1)^{th}$ hidden layer are substituted with the values acquired in the previous hidden step. These derivatives do not contain any zero partial derivatives as in the latter case. Then the complete derivation of this step consists again of 3 first derivatives, 3 second derivatives and a joint derivative following equation (24). Note that many parts of these derivatives are eliminated since the outer function is linear. Moreover, note that the derivative consists of a sum over all hidden neurons in the previous layer $q - 1$. Since the derivation is performed among two hidden layers, the derivative of neuron m in layer q depends on all hidden neurons M' in the previous layer $q - 1$. Each derivative of neuron m' consists of every input variable x_{nt}. Therefore these derivatives of all neurons are summed up in

order to obtain the derivative w.r.t. any input variable or combination of input variables, of neuron m in hidden layer q. Crucial here is to multiply all parts of the sum with the correct weight. Hence putting all parts D_{1p} and D_{2k} together like in equation (19), results in the complete partial derivatives of type 1 of the hidden step following (24):

$$\frac{\partial(g^{[q]} \circ \sigma)_m}{\partial x_{nt}} = \sum_{m'=1}^{M} w_{m'm}^{[q]} \frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{nt}} \equiv \Delta_{mn}^{[q]} \quad \text{for all} \quad n = 1, 2, 3 \tag{29}$$

$$\frac{\partial^2(g^{[q]} \circ \sigma)_m}{\partial x_{nt}\partial x_{n't}} = \sum_{m'=1}^{M} w_{m'm}^{[q]} \frac{\partial^2(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{nt}\partial x_{n't}} \equiv \Delta_{mnn'}^{[q]}, \quad \text{for all} \quad n \neq n' \tag{30}$$

$$\frac{\partial^3(g^{[q]} \circ \sigma)_m}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} = \sum_{m'=1}^{M} w_{m'm}^{[q]} \frac{\partial^3(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} \equiv \Delta_{m123}^{[q]} \tag{31}$$

where m refers to the node at which the derivation is executed, in other words the derivative of the m^{th} row of equation (28), and n refers to the input neuron. Moreover, the derivatives of this step correspond to $q > 1$. If the previous layer is the input layer then the equations (25), (26) and (27) can be substituted in the complete derivatives of type 1 of the hidden step (29)–(31) without taking any sum.

The second type, that needs to be performed subsequently, corresponds to the following composite function:

$$\sigma \circ g^{[q]} = \sigma(W^{[q]'}\sigma(W^{[q-1]'} \cdots \sigma(W^{[0]'}\mathbf{x}_t + b_t^{[0]}) \cdots + b_t^{[q-1]}) + b_t^{[q]}) \tag{32}$$

That is, the outer nonlinear function is a function of $W^{[q]'}\sigma(\cdots\sigma(W^{[0]'}\mathbf{x}_t + b_t^{[0]}) \cdots + b_t^{[q]}$. Hence now the outer function is nonlinear instead of linear and the inner function is linear instead of nonlinear. The derivatives of the outer nonlinear function w.r.t. the inner linear function, in other words D_{1p} for all $p = 1, \ldots, P$, in hidden layer q of each node m are:

$$\frac{\partial\sigma}{\partial g_m^{[q]}} = \sigma(1-\sigma) \equiv s^{(1)} \quad \text{for} \quad \pi_1$$

$$\frac{\partial^2\sigma}{\partial g_m^{[q]^2}} = \sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \equiv s^{(2)} \quad \text{for} \quad \pi_3, \quad \pi_4 \quad \text{and} \quad \pi_5$$

$$\frac{\partial^3\sigma}{\partial g_m^{[q]^3}} = \sigma(1-\sigma)^3 - 4\sigma^4(1-\sigma)^2 + \sigma^3(1-\sigma) \equiv s^{(3)} \quad \text{for} \quad \pi_2$$

The nonzero partial derivatives of the inner functions are the findings from the previous step, for notational reasons $W^{[q]'}\sigma(\cdots\sigma(W^{[0]'}\mathbf{x}_t + b_t^{[0]}) \cdots) + b_t^{[q]}$ is denoted as $g_m^{[q]}$, see (32). These derivatives are the D_{2k} parts of equation (19) for each $k = 1, \ldots, |\pi_p|$ for all P. In this case the D_{2k} parts come from equations (29), (30) and (31). In other words, putting together all constructed D_{1p} and D_{2k} of equation (19), gives the complete partial

derivatives of type 2 of the hidden step following (24):

$$\frac{\partial(\sigma \circ g^{[q]})_m}{\partial x_{nt}} = \frac{\partial \sigma}{\partial g_m^{[q]}} \frac{\partial g_m^{[q]}}{\partial x_{nt}} \qquad \text{for all} \quad n = 1, 2, 3 \qquad (33)$$

$$= \sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[q]} \frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{nt}}$$

$$\equiv s^{(1)} \Delta_{mn}^{[q]}$$

(34)

$$\frac{\partial^2(\sigma \circ g^{[q]})_m}{\partial x_{nt} \partial x_{n't}} = \frac{\partial \sigma}{\partial g_m^{[q]}} \frac{\partial^2 g_m^{[q]}}{\partial x_{nt} \partial x_{n't}} + \frac{\partial^2 \sigma}{\partial {g_m^{[q]}}^2} \frac{\partial g_m^{[q]}}{\partial x_{nt}} \frac{\partial g_m^{[q]}}{\partial x_{n't}} \qquad \text{for all} \quad n \neq n'$$

$$= \sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[q]} \frac{\partial^2(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{nt} \partial x_{n't}}$$

$$+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right]$$

$$\sum_{m'=1}^{M} w_{m'm}^{[q]} \frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{nt}}$$

$$\sum_{m'=1}^{M} w_{m'm}^{[q]} \frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{n't}}$$

$$\equiv s^{(1)} \Delta_{mnn'}^{[q]} + s^{(2)} \Delta_{mn}^{[q]} \Delta_{mn'}^{[q]}$$

$$
\begin{aligned}
\frac{\partial^3(\sigma \circ g^{[q]})_m}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} &= \frac{\partial^2\sigma}{\partial {g_m^{[q]}}^2}\frac{\partial g_m^{[q]}}{\partial x_{1t}}\frac{\partial^2 g_m^{[q]}}{\partial x_{2t}\partial x_{3t}} + \frac{\partial^2\sigma}{\partial {g_m^{[q]}}^2}\frac{\partial g_m^{[q]}}{\partial x_{2t}}\frac{\partial^2 g_m^{[q]}}{\partial x_{1t}\partial x_{3t}} \\
&+ \frac{\partial^2\sigma}{\partial {g_m^{[q]}}^2}\frac{\partial g_m^{[q]}}{\partial x_{3t}}\frac{\partial^2 g_m^{[q]}}{\partial x_{1t}\partial x_{2t}} + \frac{\partial\sigma}{\partial g_m^{[q]}}\frac{\partial^3 g_m^{[q]}}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} \\
&+ \frac{\partial^3\sigma}{\partial {g_m^{[q]}}^3}\frac{\partial g_m^{[q]}}{\partial x_{1t}}\frac{\partial g_m^{[q]}}{\partial x_{2t}}\frac{\partial g_m^{[q]}}{\partial x_{3t}} \\
&= \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{1t}} \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial^2(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{2t}\partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{2t}} \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial^2(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{1t}\partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{3t}} \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial^2(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{1t}\partial x_{2t}} \\
&+ \sigma(1-\sigma)\sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial^3(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^3 - 4\sigma^4(1-\sigma)^2 + \sigma^3(1-\sigma)\right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{1t}} \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{2t}} \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[q]}\frac{\partial(\sigma \circ g^{[q-1]})_{m'}}{\partial x_{3t}} \\
&\equiv s^{(2)}\Delta_{m1}^{[q]}\Delta_{m23}^{[q]} + s^{(2)}\Delta_{m2}^{[q]}\Delta_{m13}^{[q]} + s^{(2)}\Delta_{m3}^{[q]}\Delta_{m12}^{[q]} \\
&+ s^{(1)}\Delta_{m123}^{[q]} + s^{(3)}\Delta_{m1}^{[q]}\Delta_{m2}^{[q]}\Delta_{m3}^{[q]}
\end{aligned}
\tag{35}
$$

where m refers to the node at which the derivation is executed, or the derivative of the m^{th} row of equation (32), and n refers to the input neuron. These complete derivatives of type 2 of the hidden step are used for D_{2k} for each k for all P in the next hidden step or the output step, as mentioned before and depicted in Figure 4.2. Hence depending on the level of differentiation, Faà di Bruno is used again. This is actually happening in all steps. However, several ingredients of FDB derivatives are occasionally equal to zero, due to linearity, which result in more simple derivatives.

This hidden step is performed $H-1$ times until the last layer H is reached. Then the output step is performed, see the next subsection.

Output step: $g^{[H]} \circ \sigma$ or $\sigma \circ g^{[H]}$

This step is performed when moving from the last hidden layer to the output layer. In this step, either only type 1 is executed or both type 1 and type 2 are executed

depending on the presence of an activation function on the output layer. Type 1 corresponds to the following composite function:

$$g^{[H]} \circ \sigma = W^{[H]'}\sigma(W^{[H-1]'}\sigma(\cdots\sigma(W^{[0]'}\mathbf{x}_t + b_t^{[0]})\cdots + b_t^{[H-1]}) + b_t^{[H]} \tag{36}$$

Hence this is the case where no activation function is applied on the output layer[11]. The derivatives of the outer linear function , D_{1p} for all $p = 1, \dots, P$, is a function of $\sigma(W^{[H-1]'}\sigma(\cdots\sigma(W^{[0]'}\mathbf{x}_t + b_t^{[0]}) + b_t^{[q]}\cdots + b_t^{[H-1]})$ and needs to be derived only for one output neuron. However, the derivative still depends on m of previous layer H:

$$\frac{\partial g_m^{[H]}}{\partial\sigma(g_m^{[H-1]})} = w_{m1}^{[H]} \qquad \text{for} \quad \pi_1$$

$$\frac{\partial^2 g_m^{[H]}}{\partial\sigma(g_m^{[H-1]})^2} = 0 \qquad \text{for} \quad \pi_3, \quad \pi_4 \quad \text{and} \quad \pi_5$$

$$\frac{\partial^3 g_m^{[H]}}{\partial\sigma(g_m^{[H-1]})^3} = 0 \qquad \text{for} \quad \pi_2$$

The nonzero partial derivatives of the inner functions, D_{2k} for all $k = 1, \dots, |\pi_p|$ in each π_p for each hidden neuron m, are the findings from the previous step. That is, the derivative of the m^{th} row of $\sigma(W^{[H-1]'}\sigma(\cdots W^{[q]'}\sigma(\cdots\sigma(W^{[0]'}\mathbf{x}_t + b_t^{[0]})\cdots + b_t^{[q]}\cdots) + b^{[H-1]})$, where notation of the last hidden step (type 2) is used, represents D_{2k}, see equations (33), (34) and (35) for $q = H - 1$. Then the complete derivation of this step consists of 3 first derivatives, that is, $\frac{\partial g^{[H]}\circ\sigma}{\partial x_{1t}}$, $\frac{\partial g^{[H]}\circ\sigma}{\partial x_{2t}}$ and $\frac{\partial g^{[H]}\circ\sigma}{\partial x_{3t}}$, 3 second derivatives, $\frac{\partial^2 g^{[H]}\circ\sigma}{\partial x_{1t}\partial x_{2t}}$, $\frac{\partial^2 g^{[H]}\circ\sigma}{\partial x_{2t}\partial x_{3t}}$, $\frac{\partial^2 g^{[H]}\circ\sigma}{\partial x_{1t}\partial x_{3t}}$ and a joint derivative $\frac{\partial^3 g^{[H]}\circ\sigma}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}$. In other words putting together D_{1p} and D_{2k} following equation (19), gives the complete partial derivatives of type 1 of the output step following (24):

$$\begin{aligned}\frac{\partial(g^{[H]}\circ\sigma)_m}{\partial x_n} &= w_{m1}^{[H]}\left(\frac{\partial g_m^{[H-1]}}{\partial x_{nt}}\frac{\partial\sigma}{\partial g_m^{[H-1]}}\right) \qquad \text{for all} \quad n = 1,2,3 \\ &= w_{m1}^{[H]}\left(\sigma(1-\sigma)\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{nt}}\right) \\ &\equiv w_{m1}^{[H]}\left(s^{(1)}\Delta_{mn}^{[H-1]}\right)\end{aligned} \tag{37}$$

[11] This type is also performed in the case when there is an activation function applied on the output layer.

$$\frac{\partial^2 (g^{[H]} \circ \sigma)_m}{\partial x_{nt} \partial x_{n't}} = w_{m1}^{[H]} \Bigg(\frac{\partial^2 g_m^{[H-1]}}{\partial x_{nt} \partial x_{n't}} \frac{\partial \sigma}{\partial g_m^{[H-1]}} \qquad \text{for all} \quad n \neq n' \qquad (38)$$

$$+ \frac{\partial g_m^{[H-1]}}{\partial x_{nt}} \frac{\partial g_m^{[H-1]}}{\partial x_{n't}} \frac{\partial^2 \sigma}{\partial g_m^{[H-1]^2}} \Bigg)$$

$$= w_{m1}^{[H]} \Bigg(\sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2 (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{nt} \partial x_{n't}}$$

$$+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \right]$$

$$\sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{nt}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{n't}} \Bigg)$$

$$\equiv w_{m1}^{[H]} \left(s^{(1)} \Delta_{mnn'}^{[H-1]} + s^{(2)} \Delta_{mn}^{[H-1]} \Delta_{mn'}^{[H-1]} \right)$$

$$
\begin{aligned}
\frac{\partial^3 (g^{[H]} \circ \sigma)_m}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} &= w_{m1}^{[H]} \Bigg(\frac{\partial^2 \sigma}{\partial g_m^{[H-1]^2}} \frac{\partial g_m^{[H-1]}}{\partial x_{1t}} \frac{\partial^2 g_m^{[H-1]}}{\partial x_{2t} \partial x_{3t}} \\
&+ \frac{\partial^2 \sigma}{\partial g_m^{[H-1]^2}} \frac{\partial g_m^{[H-1]}}{\partial x_{2t}} \frac{\partial^2 g_m^{[H-1]}}{\partial x_{1t} \partial x_{3t}} + \frac{\partial^2 \sigma}{\partial g_m^{[H-1]^2}} \frac{\partial g_m^{[H-1]}}{\partial x_{3t}} \frac{\partial^2 g_m^{[H-1]}}{\partial x_{1t} \partial x_{2t}} \Bigg) \\
&+ w_{m1}^{[H]} \Bigg(\frac{\partial \sigma}{\partial g_m^{[H-1]}} \frac{\partial^3 g_m^{[H-1]}}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} \\
&+ \frac{\partial^3 \sigma}{\partial g_m^{[H-1]^3}} \frac{\partial g_m^{[H-1]}}{\partial x_{1t}} \frac{\partial g_m^{[H-1]}}{\partial x_{2t}} \frac{\partial g_m^{[H-1]}}{\partial x_{3t}} \Bigg) \\
&= w_{m1}^{[H]} \Bigg(\left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2 (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t} \partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2 (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t} \partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{3t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2 (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t} \partial x_{2t}} \Bigg) \\
&+ w_{m1}^{[H]} \Bigg(\sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^3 (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^3 - 4\sigma^4(1-\sigma)^2 + \sigma^3(1-\sigma) \right] \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}} \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial (\sigma \circ g^{[H-2]})_{m'}}{\partial x_{3t}} \Bigg)
\end{aligned}
\tag{39}
$$

$$
\begin{aligned}
\frac{\partial^3 (g^{[H]} \circ \sigma)_m}{\partial x_{1t} \partial x_{2t} \partial x_{3t}} &\equiv w_{m1}^{[H]} \Bigg(s^{(2)} \Delta_{m1}^{[H-1]} \Delta_{m23}^{[H-1]} + s^2 \Delta_{m2}^{[H-1]} \Delta_{m13}^{[H-1]} \\
&+ s^{(2)} \Delta_{m3}^{[H-1]} \Delta_{m12}^{[H-1]} + s^{(1)} \Delta_{m123}^{[H-1]} + s^{(3)} \Delta_{m1}^{[H-1]} \Delta_{m2}^{[H-1]} \Delta_{m3}^{[H-1]} \Bigg)
\end{aligned}
$$

where m refers to the node at the last hidden layer and n refers to the input neuron. When there is no activation function imposed on the output layer, the final joint PDF, that is the derivative of equation (36), is equal to summing over $\frac{\partial^3 (g^{[H]} \circ \sigma)_m}{\partial x_{1t} \partial x_{2t} \partial x_{3t}}$ for all m. Similarly, the final marginal PDF of input variable n is obtained by summing over $\frac{\partial g^{[H]} \circ \sigma}{\partial x_{nt}}$

for all m. Thus the final derivative is formed by:

$$\frac{\partial}{\partial x_{1t}}\tilde{y}_t = \sum_{m=1}^{M} \frac{\partial (g^{[H]} \circ \sigma)_m}{\partial x_{nt}}$$

$$\frac{\partial^2}{\partial x_{nt}\partial x_{n't}}\tilde{y}_t = \sum_{m=1}^{M} \frac{\partial^2 (g^{[H]} \circ \sigma)_m}{\partial x_{nt}\partial x_{n't}}$$

$$\frac{\partial^3}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}\tilde{y}_t = \sum_{m=1}^{M} \frac{\partial^3 (g^{[H]} \circ \sigma)_m}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}$$

Thus in each step, FDB is performed w.r.t. all input variables N and combinations of these. For $N = 3$, there are 7 complete FDB derivatives in each step. Each complete derivative is substituted in the corresponding D_{2k} of the complete FDB derivatives in the next step. This implies that the complete FDB derivative of the next step is formed by using complete FDB derivatives of the previous step. By repeatedly inserting complete derivatives of the previous step in the current step, the final derivative w.r.t. the input variables, or combinations of these, of the output step is attained. This is done $2(H-1)$ times, that is for all hidden steps, until the output step function is reached.

Type 2 is performed as a follow-up of type 1 when an activation function is applied on the output layer. It might be preferable to impose the logistic function on the output function since its range is between 0 and 1. This corresponds to the output values that represent probabilities. Other activation functions, like Trentin et al. (2018) use for example, are unnecessary. Therefore if an activation function is imposed on the output layer, only the logistic function is considered. In this way, the output values are forced to the admissable range $(0, 1)$. This type corresponds to the following composite function:

$$\sigma \circ g^{[H]} = \sigma(W^{[H]'}\sigma(W^{[H-1]'}\sigma(\cdots\sigma(W^{[0]'}\mathbf{x}_t + b_t^{[0]})\cdots) + b_t^{[H-1]}) + b_t^{[H]}) \qquad (40)$$

The derivatives of the outer function, D_{1p} for all π_p, w.r.t. the inner linear function of the output node, of which input $g^{[H]}$ is still related to the m^{th} node of the last hidden layer, are:

$$\frac{\partial \sigma}{\partial g_m^{[H]}} = \sigma(1-\sigma) \equiv s^{(1)} \qquad \text{for} \quad \pi_1$$

$$\frac{\partial^2 \sigma}{\partial {g_m^{[H]}}^2} = \sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \equiv s^{(2)} \qquad \text{for} \quad \pi_3, \quad ,\pi_4 \quad \text{and} \quad \pi_5$$

$$\frac{\partial^3 \sigma}{\partial {g_m^{[H]}}^3} = \sigma(1-\sigma)^3 - 4\sigma^4(1-\sigma)^2 + \sigma^3(1-\sigma) \equiv s^{(3)} \qquad \text{for} \quad \pi_2$$

The nonzero partial derivatives of the inner functions are the findings from type 1, referred to as D_{2k} for each $k = 1, \ldots, |\pi_p|$ of all π_p. These complete derivatives are represented by equations (37), (38) and (39). Then the complete derivation of this step

following equation (24), combining D_{1p} and D_{2k} as in equation (19), is:

$$\frac{\partial(\sigma \circ g^{[H]})_m}{\partial x_{nt}} = \frac{\partial \sigma}{\partial g_m^{[H]}} \left(w_{m1}^{[H]} \frac{\partial g_m^{[H-1]}}{\partial x_{nt}} \frac{\partial \sigma}{\partial g_m^{[H-1]}} \right) \qquad \text{for all} \quad n = 1,2,3 \qquad (41)$$

$$= \sigma(1-\sigma) \left(w_{m1}^{[H]} \sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{nt}} \right)$$

$$\equiv s^{(1)} \left(w_{m1}^{[H]} s^{(1)} \Delta_{mn}^{[H-1]} \right)$$

$$\frac{\partial^2(\sigma \circ g^{[H]})_m}{\partial x_{nt} \partial x_{n't}} = \frac{\partial \sigma}{\partial g_m^{[H]}} \Big(w_{m1}^{[H]} \Big[\frac{\partial^2 g_m^{[H-1]}}{\partial x_{nt} \partial x_{n't}} \frac{\partial \sigma}{\partial g_m^{[H-1]}} \qquad \text{for all} \quad n \neq n' \qquad (42)$$

$$+ \frac{\partial g_m^{[H-1]}}{\partial x_{nt}} \frac{\partial g_m^{[H-1]}}{\partial x_{n't}} \frac{\partial^2 \sigma}{\partial {g_m^{[H-1]}}^2} \Big] \Big)$$

$$+ \frac{\partial^2 \sigma}{\partial {g_m^{[H]}}^2} \left(w_{m1}^{[H]} \frac{\partial g_m^{[H-1]}}{\partial x_{nt}} \frac{\partial \sigma}{\partial g_m^{[H-1]}} \right) \left(w_{m1}^{[H]} \frac{\partial g_m^{[H-1]}}{\partial x_{n't}} \frac{\partial \sigma}{\partial g_m^{[H-1]}} \right)$$

$$= \sigma(1-\sigma) \Big(w_{m1}^{[H]} \Big[\sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{nt} \partial x_{n't}}$$

$$+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \right]$$

$$\sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{nt}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{n't}} \Big] \Big)$$

$$+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma) \right]$$

$$\left(w_{m1}^{[H]} \sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{nt}} w_{m1}^{[H]} \right)$$

$$\left(w_{m1}^{[H]} \sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{n't}} \right)$$

$$\equiv s^{(1)} \left(w_{m1}^{[H]} \left[s^{(1)} \Delta_{mnn'}^{[H-1]} + s^{(2)} \Delta_{mn}^{[H-1]} \Delta_{mn'}^{[H-1]} \right] \right)$$

$$+ s^{(2)} \left(w_{m1}^{[H]} s^{(1)} \Delta_{mn}^{[H-1]} \right) \left(w_{m1}^{[H]} s^{(1)} \Delta_{mn'}^{[H-1]} \right)$$

$$
\begin{aligned}
&\frac{\partial^3(\sigma \circ g^{[H]})_m}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} = \\
&\frac{\partial^2\sigma}{\partial {g_m^{[H]}}^2}\left(w_{m1}^{[H]}\left[\frac{\partial^2 g_m^{[H-1]}}{\partial x_{1t}\partial x_{2t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}} + \frac{\partial g_m^{[H-1]}}{\partial x_{1t}}\frac{\partial g_m^{[H-1]}}{\partial x_{2t}}\frac{\partial^2\sigma}{\partial {g_m^{[H-1]}}^2}\right]\right) \\
&\left(w_{m1}^{[H]}\frac{\partial g_m^{[H-1]}}{\partial x_{3t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}}\right) \\
&+ \frac{\partial^2\sigma}{\partial {g_m^{[H]}}^2}\left(w_{m1}^{[H]}\left[\frac{\partial^2 g_m^{[H-1]}}{\partial x_{2t}\partial x_{3t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}} + \frac{\partial g_m^{[H-1]}}{\partial x_{2t}}\frac{\partial g_m^{[H-1]}}{\partial x_{3t}}\frac{\partial^2\sigma}{\partial {g_m^{[H-1]}}^2}\right]\right) \\
&\left(w_{m1}^{[H]}\frac{\partial g_m^{[H-1]}}{\partial x_{1t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}}\right) \\
&+ \frac{\partial^2\sigma}{\partial {g_m^{[H]}}^2}\left(w_{m1}^{[H]}\left[\frac{\partial^2 g_m^{[H-1]}}{\partial x_{1t}\partial x_{3t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}} + \frac{\partial g_m^{[H-1]}}{\partial x_{1t}}\frac{\partial g_m^{[H-1]}}{\partial x_{3t}}\frac{\partial^2\sigma}{\partial {g_m^{[H-1]}}^2}\right]\right) \\
&\left(w_{m1}^{[H]}\frac{\partial g_m^{[H-1]}}{\partial x_{2t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}}\right) \\
&+ \frac{\partial\sigma}{\partial g_m^{[H]}}\left(w_{m1}^{[H]}\left[\frac{\partial^2\sigma}{\partial {g_m^{[H-1]}}^2}\frac{\partial g_m^{[H-1]}}{\partial x_{1t}}\frac{\partial^2 g_m^{[H-1]}}{\partial x_{2t}\partial x_{3t}}\right.\right. \\
&\left.\left.+ \frac{\partial^2\sigma}{\partial {g_m^{[H-1]}}^2}\frac{\partial g_m^{[H-1]}}{\partial x_{2t}}\frac{\partial^2 g_m^{[H-1]}}{\partial x_{1t}\partial x_{3t}} + \frac{\partial^2\sigma}{\partial {g_m^{[H-1]}}^2}\frac{\partial g_m^{[H-1]}}{\partial x_{3t}}\frac{\partial^2 g_m^{[H-1]}}{\partial x_{1t}\partial x_{2t}}\right]\right) \\
&+ \frac{\partial\sigma}{\partial g_m^{[H]}}\left(w_{m1}^{[H]}\left[\frac{\partial\sigma}{\partial g_m^{[H-1]}}\frac{\partial^3 g_m^{[H-1]}}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}\right.\right. \\
&\left.\left.+ \frac{\partial^3\sigma}{\partial {g_m^{[H-1]}}^3}\frac{\partial g_m^{[H-1]}}{\partial x_{1t}}\frac{\partial g_m^{[H-1]}}{\partial x_{2t}}\frac{\partial g_m^{[H-1]}}{\partial x_{3t}}\right]\right) \\
&+ \frac{\partial^3\sigma}{\partial {g_m^{[H]}}^3}\left(w_{m1}^{[H]}\frac{\partial g_m^{[H-1]}}{\partial x_{1t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}}\right)\left(w_{m1}^{[H-1]}\frac{\partial g_m^{[H-1]}}{\partial x_{2t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}}\right) \\
&\left(w_{m1}^{[H-1]}\frac{\partial g_m^{[H-1]}}{\partial x_{3t}}\frac{\partial\sigma}{\partial g_m^{[H-1]}}\right)
\end{aligned}
\tag{43}
$$

$$
\begin{aligned}
&= \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right]\left(w_{m1}^{[H]}\left[\sigma(1-\sigma)\right.\right. \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}\partial x_{2t}} + \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right] \\
&\quad \left.\left. \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}} \right]\right) \\
&\quad \left(w_{m1}^{[H]}\sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{3t}} \right) \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right]\left(w_{m1}^{[H]}\left[\sigma(1-\sigma)\right.\right. \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}\partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right] \\
&\quad \left.\left. \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{3t}} \right]\right) \\
&\quad \left(w_{m1}^{[H]}\sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}} \right) \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right]\left(w_{m1}^{[H]}\left[\sigma(1-\sigma)\right.\right. \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}\partial x_{3t}} + \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right] \\
&\quad \left.\left. \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{3t}} \right]\right) \\
&\quad \left(w_{m1}^{[H]}\sigma(1-\sigma) \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}} \right) \\
&+ \sigma(1-\sigma)\left(w_{m1}^{[H]}\left[\left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right]\right.\right. \\
&\quad \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}} \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial^2(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}\partial x_{3t}} \\
&+ \left[\sigma(1-\sigma)^2 - \sigma^2(1-\sigma)\right] \\
&\quad \left.\left. \sum_{m'=1}^{M} w_{m'm}^{[H-1]} \frac{\partial(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{2t}} \sum_{m'=1}^{M} w_{m'm}^{[H]} \frac{\partial^2(\sigma \circ g^{[H-2]})_{m'}}{\partial x_{1t}\partial x_{3t}} \right]\right)
\end{aligned}
$$

$$
\begin{aligned}
&+\left[\sigma(1-\sigma)^2-\sigma^2(1-\sigma)\right]\\
&\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{3t}}\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial^2(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{1t}\partial x_{2t}}\Bigg]\Bigg)\\
&+\sigma(1-\sigma)\Bigg(w_{m1}^{[H]}\Bigg[\sigma(1-\sigma)\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial^3(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}\\
&+\left[\sigma(1-\sigma)^3-4\sigma^4(1-\sigma)^2+\sigma^3(1-\sigma)\right]\\
&\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{1t}}\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{2t}}\\
&\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{3t}}\Bigg]\Bigg)\\
&+\left[\sigma(1-\sigma)^3-4\sigma^4(1-\sigma)^2+\sigma^3(1-\sigma)\right]\\
&\left(w_{m1}^{[H]}\sigma(1-\sigma)\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{1t}}\right)\\
&\left(w_{m1}^{[H]}\sigma(1-\sigma)\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{2t}}\right)\\
&\left(w_{m1}^{[H]}\sigma(1-\sigma)\sum_{m'=1}^{M} w_{m'm}^{[H-1]}\frac{\partial(\sigma\circ g^{[H-2]})_{m'}}{\partial x_{3t}}\right)
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^3(\sigma\circ g^{[H]})_m}{\partial x_{1t}\partial x_{2t}\partial x_{3t}} \equiv\ & s^{(2)}\left(w_{m1}^{[H]}\left[s^{(1)}\Delta_{m12}^{[H-1]}+s^{(2)}\Delta_{m1}^{[H-1]}\Delta_{m2}^{[H-1]}\right]w_{m1}^{[H]}s^{(1)}\Delta_{m3}^{[H-1]}\right)\\
&+s^{(2)}\left(w_{m1}^{[H]}\left[s^{(1)}\Delta_{m3}^{[H-1]}+s^{(2)}\Delta_{m2}^{[H-1]}\Delta_{m3}^{[H-1]}\right]w_{m1}^{[H]}s^{(1)}\Delta_{m1}^{[H-1]}\right)\\
&+s^{(2)}\left(w_{m1}^{[H]}\left[s^{(1)}\Delta_{m13}^{[H-1]}+s^{(2)}\Delta_{m1}^{[H-1]}\Delta_{m3}^{[H-1]}\right]w_{m1}^{[H]}s^{(1)}\Delta_{m2}^{[H-1]}\right)\\
&+s^{(1)}\\
&\left(w_{m1}^{[H]}\left[s^{(2)}\Delta_{m1}^{[H-1]}\Delta_{m23}^{[H-1]}+s^{(2)}\Delta_{m2}^{[H-1]}\Delta_{m13}^{[H-1]}+s^{(2)}\Delta_{m3}^{[H-1]}\Delta_{m12}^{[H-1]}\right]\right)\\
&+s^{(1)}\left(w_{m1}^{[H]}\left[s^{(1)}\Delta_{m123}^{[H-1]}+s^{(3)}\Delta_{m1}^{[H-1]}\Delta_{m2}^{[H-1]}\Delta_{m3}^{[H-1]}\right]\right)\\
&+s^{(3)}\left(w_{m1}^{[H]}s^{(1)}\Delta_{m1}^{[H-1]}\right)\left(w_{m1}^{[H]}s^{(1)}\Delta_{m2}^{[H-1]}\right)\left(w_{m1}^{[H]}s^{(1)}\Delta_{m3}^{[H-1]}\right)
\end{aligned}
$$

where m refers to the node at the last hidden layer and n refers to the input neuron. Finally, the joint derivative of equation (40), is given by summing $\frac{\partial^3(\sigma\circ g^{[H]})_m}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}$ over all m. Moreover, the marginal and partial derivatives are also given by summing over

$\frac{\partial(\sigma\circ g^{[H]})_m}{\partial x_{nt}}$ or $\frac{\partial^2(\sigma\circ g^{[H]})_m}{\partial x_{nt}\partial x_{n't}}$ over all m. Thus the final derivatives are given by:

$$\frac{\partial}{\partial x_{1t}}\tilde{y}_t = \sum_{m=1}^{M}\frac{\partial(\sigma\circ g^{[H]})_m}{\partial x_{nt}} \qquad \text{for all} \quad n = 1,2,3$$

$$\frac{\partial^2}{\partial x_{nt}\partial x_{n't}}\tilde{y}_t = \sum_{m=1}^{M}\frac{\partial^2(\sigma\circ g^{[H]})_m}{\partial x_{nt}\partial x_{n't}} \qquad \text{for all} \quad n \neq n'$$

$$\frac{\partial^3}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}\tilde{y}_t = \sum_{m=1}^{M}\frac{\partial^3(\sigma\circ g^{[H]})_m}{\partial x_{1t}\partial x_{2t}\partial x_{3t}}$$

Thus in each step, FDB is performed w.r.t. all input variables N and combinations of these. For $N = 3$, there are 7 complete FDB derivatives in each step. Each complete derivative is substituted in the corresponding D_{2k} of the complete FDB derivatives in the next step. This implies that the complete FDB derivative of the next step is formed by using complete FDB derivatives of the previous step. By repeatedly inserting complete derivatives of the previous step in the current step, the final derivative w.r.t. the input variables, or combinations of these, of the output step is attained. This is done $2(H-1)+1$ times, that is for all hidden steps and type 1 of the output step, until the output step function is reached.

Note that in these derivations, the logistic activation function $\sigma(\cdot)$ is used. However, other N-differentiable derivatives can be used, such as the hyperbolic tangent function $\tau(\cdot)$. See appendix B for the N^{th} derivatives of the hyperbolic tangent function[12)].

5 Results

In this section we illustrate the proposed PDF application method with several simulation examples generated from four DGPs and compare our findings to the closest methodology in the literature, Trentin et al. (2018), and the KDE, when applicable. Specifically, we consider simulated data sets generated from four distinct DGPs: PDF estimation where the underlying DGP is a mixture of normal distributions, a mixture of Generalized Extreme Value (GEV) distributions, a mixture of Poisson distributions and a mixture of correlated normal distributions. In the first and the second DGP simulations, a mixture of normal distributions (5.1) and a mixture of GEV distributions (5.2) are studied to illustrate the performance of our method in case of multiple nonlinearities and extremes in the target distribution. These simulation studies follow the simulation settings as in Magdon-Ismail and Atiya (2002) and Trentin et al. (2018), respectively. For these simulations, we compare our results to KDEs and Trentin et al. (2018)'s method. The KDE uses Scott's and/or Silverman's bandwidth. Trentin et al. (2018)'s method uses several different initial bandwidths, specified empirically. The third and fourth simulations, a mixture of Poisson distributions (5.3) and mixture of correlated normal distributions(5.4), aim to illustrate the performance of our method in case of discrete data and correlated multivariate data. The third simulation aims to present the performance of our approach for discrete data applications, such as the road sensor data

12) Also, the N^{th} derivatives of the logistic activation function are explained in appendix B.

explained in Section 2. A univariate and a correlated bivariate case are shown. The last simulation aims to show the verification of expanding input dimensions, irrespective whether these are correlated or not. Applications to a bivariate and a trivariate correlated standard normal distribution are shown. For these simulations, we only compare our results to KDEs, as the other baseline method Trentin et al. (2018) is not applicable to correlated multivariate data. Hence performances in terms of L2 losses of only our method and the KDE are given in the fourth simulation studies.

The proposed method uses the L2 loss functions $L2_{\widehat{CDF}}$, $L2_{CDF}$, $L2_{\widehat{CDF}}$ and $L2_{PDF}$ or $L2_{PMF}$, defined in section 4.2. Note that Trentin et al. (2018) does not train a neural network to estimate the CDF, but directly estimates the PDF of a specific DGP. Therefore, when we compare our results with those of Trentin et al. (2018), we additionally report this methods' performance using the following L2 loss functions:

(i) $L2_{\widehat{PDF}}$ based on the difference between the target PDF $\hat{y}_t'$ and the estimated PDF $\tilde{y}_t'$
(ii) $L2_{PDF}$ based on the difference between the true PDF y_t' and the estimated PDF $\tilde{y}_t'$
(iii) $L2_{\widehat{PDF}}$ based on the difference between the true PDF y_t' and the target PDF $\hat{y}_t'$.

The KDE does not use a target PDF and therefore the corresponding results of this method are only assessed by the $L2_{PDF}$ loss function. Hence all methods obtain the $L2_{PDF}$ loss which can be used to compare performances.

The best, average and worst CDF and PDF estimates out of 100 simulation replications in terms of $L2_{\widehat{CDF}}$ loss are illustrated for the proposed method. Similarly, the best, average and worst PDF estimates are shown for Trentin et al. (2018)'s method based on the $L2_{\widehat{PDF}}$ loss. $L2_{\widehat{CDF}}$ and $L2_{\widehat{PDF}}$ losses are used as these are the feasible loss functions for a real data application. For the multivariate simulation studies these estimates are shown in terms of differences. Specifically, differences of each data point between the target CDF $\hat{y}_t$ and estimated CDF $\tilde{y}_t$ together with differences of each data point between the true PDF y_t' and the estimated PDF $\tilde{y}_t'$ are shown for the best, average and worst CDF and PDF estimates. The CDF and PDF estimates are shown in appendices F, G, H, I, and J. True, target and estimated CDFs and true and estimated PDFs shown in these appendices are very similar, therefore differences are shown with the consequence that these differences look very much magnified. Furthermore corresponding squared errors of each data point over 100 simulation replications are shown. These summarize specific size and domain errors for each distribution over all simulation replications, instead of only the best average and worst estimates. In this way, results that are not shown by the latter figures are provided. For the proposed method the following squared errors of each data point are illustrated:

(i) $S2_{\widehat{CDF}}$ based on the difference between the target CDF $\hat{y}_t$ and the estimated CDF $\tilde{y}_t$
(ii) $S2_{CDF}$ based on the difference between the true CDF y_t and the estimated CDF $\tilde{y}_t$
(iiia) $S2_{PDF}$ based on the difference between the estimated PDF $\tilde{y}_t'$ and true PDF y_t'
(iiib) $S2_{PMF}$ based on the difference between the estimated PMF $\tilde{y}_t'$ and true PMF y_t'.

Trentin et al. (2018)'s approach uses the following squared errors of each data point:

(i) $S2_{\widehat{PDF}}$ based on the difference between the target PDF $\hat{y}_t'$ and the estimated PDF $\tilde{y}_t'$
(ii) $S2_{PDF}$ based on the difference between the true PDF y_t' and estimated PDF $\tilde{y}_t'$.

Also, the $S2_{PDF}$ errors of the KDE are shown. Hence different methods are compared by analyzing the $S2_{PDF}$ squared errors.

For each DGP, we consider different simulation settings to illustrate the method's performance in different sample sizes. Training is done using K-fold cross validation. Complex distributions need many training iterations to estimate the weight parameters

of the neural network. Using a small batch size is a cheap way of training. However, there is a trade-off between training complex distributions and the representativeness of each batch. Therefore using a small batch size for small sample sizes can be a potential pitfall. If this is the case a batch size as large as the training sample is used. Then the number of training iterations is equal to the number of epochs. Otherwise the number of training iterations is equal to the number of epochs times (K-1), where K is the number of folds in K-fold cross validation. In this application, $K = 5$. After training, the performance of the methods are then evaluated in a test sample that is $1/5^{th}$ of the training sample. In terms of the sample size for training the neural network, we consider a small sample of $T = 600$ observations with a training size of $T = 500$, batch size of $T = 100$ and test size of $T = 100$, a medium sample of $T = 6000$ observations with a training size of $T = 5000$, batch size of $T = 1000$ and test size of $T = 1000$ and a large sample of $T = 24,000$ observations with a training size of $T = 20,000$, batch size of $T = 4000$ and test size of $T = 4000$. Note that the test sample is not used for training. For the bivariate simulation cases, only the medium and large sample sizes are used since small sample sizes are already difficult to estimate for univariate simulation cases. Additionally for these simulation cases, negative values are truncated to zero as negative probabilities are not feasible. Simulation settings of the GEV distribution in section 5.2 follow Trentin et al. (2018)'s set-up of 10-fold cross validation. There are two distinct sample sizes for training, a small size of $T = 11,000$ with a training size of $T = 10,000$, batch size of $T = 1000$ and test size of $T = 1000$ and a medium size of $T = 110,000$ observations with a training size of $T = 100,000$, batch size of $T = 10,000$ and test size of $T = 10,000$. Each simulation study is based on 100 Monte Carlo replications.

During K-fold cross validation training, the best performing model is chosen in terms of $L2_{\widetilde{CDF}}$ loss as this is the feasible loss function for a real data application. Optimal hyperparameters are derived first. Specifically, the learning rate, number of epochs, batch size and activation function[13] are determined by grid search. Hence before training, we specify the above mentioned hyperparameters by grid search using $L2_{\widetilde{CDF}}$ with a large MLP consisting of a set of MLPs with pre-specified hidden layers and neurons. Then using these optimal hyperparameters together with the large MLP, training is done using all folds in order to show predictions on the test set. Corresponding $L2_{\widetilde{CDF}}$, $L2_{CDF}$, $L2_{\widehat{CDF}}$ and $L2_{PDF}$ or $L2_{PMF}$ losses on K-fold cross validation samples, hereafter called validation sample, and test sample are summarized in tables for the proposed method, $L2_{\widetilde{PDF}}$, $L2_{PDF}$ and $L2_{\widehat{PDF}}$ for Trentin et al. (2018)'s method. Note that the L2 losses on the validation sample are means taken over K folds for each model for each simulation replication. Similar L2 losses are given for the test sample using all 5 folds as training sample. Moreover, results are given for each model but also for the selected optimal model in each simulation replication. This selection can differ over simulation replications. Due to the advantage of using a large MLP instead of several subMLPs, performance of each model can be compared not only over the validation sample but also over the test sample using the same amount of computational power. For univariate simulation cases, two hidden layers are used to show the additional gain from using more than 1 hidden layer opposed to Magdon-Ismail and Atiya (2002) and Trentin et al. (2018) with the exception of the univariate GEV distribution where three hidden layers are used due to the mix of three distributions instead of two. For the

[13] Xavier initialization and scaled variance is used to match the properties of the sigmoid activation functions, see Glorot and Bengio (2010).

bivariate and trivariate simulation cases, three and four hidden layers together with more hidden neurons are used due to the additional dimensions. For all simulation cases, the logistic activation function is used except for the univariate Poisson distribution. For this simulation case, the hyperbolic tangent function is used since this function is more steep than the logistic function. In this way learning is faster which is required due to the discrete nature that restricts the number of events. This is unnecessary for the bivariate Poisson distribution because the loss function is more complex using two dimensions and there are a larger numbers of events. See Table K.1 in Appendix K for the exact neural network settings.

5.1 Mixed Normal Distribution

In this section we consider simulated data from the following mixed normal distributions:

$$p(x) = 0.3\mathcal{N}(-30, 9) + 0.7\mathcal{N}(9, 81),$$

where $\mathcal{N}(\mu, \sigma^2)$ stands for the normal distribution with mean μ and variance σ^2. This setting is similar to the simulation setting of Magdon-Ismail and Atiya (2002) with different sample sizes. We present the true CDF $p(x)$ denoted as y_t using the CDF of the mixed normal distributions, target CDF $\hat{y}_t$ corresponding to the empirical CDF, and estimated CDF $\tilde{y}_t$ of the proposed method together with true PDF y_t' and estimated PDF $\tilde{y}_t'$ of the best, average and worst estimates from 100 simulation replications, in terms of $L2_{\widetilde{CDF}}$ loss. The PDFs estimated by the KDE and Trentin et al. (2018) are also presented. These results are shown for the small, medium and large sample sizes, using 10,000 training epochs and a learning rate of 0.01. A batch size as large as the training sample $T = 400$ is used for the small sample size. The logistic output function is used as activation function in all layers except for the output layer. No activation function is used on the output layer for the proposed method. In Table 5.1 the different models that are applied to the mixed normal distribution are summarized.

model	1	2	3	4	5	6
# hidden layers	1	1	1	2	2	2
# hidden neurons	3	4	5	3	4	5

Table 5.1 Overview of models considered for the mixed normal distribution

Small sample results:

Table 5.2 presents the small sample results for the validation sample and test sample. The mean and standard error of L2 losses for the training set are calculated over the mean of 5 folds of each simulation replication. Out of 100 simulation replications, models 1, 4, and 6 are the three best performing models obtaining the smallest $L2_{\widetilde{CDF}}$ loss in the validation sample, see row 1 of Table 5.2. On average both the feasible loss, $L2_{\widetilde{CDF}}$ in row 1, and the true loss, $L2_{CDF}$ in row 2, are smallest for model 4, closely followed by model 1. In this small sample experiment, the $L2_{CDF}$ loss is strongly influenced by the difference between the true and target CDF, e.g. the $L2_{CDF}$ loss for model 1 and 2 are similar to a digit even though model 2 performs worse than model 1 in terms of $L2_{\widetilde{CDF}}$. Model 4 has a substantial smaller $L2_{PDF}$ loss than model 1. Model 4 is chosen as optimal MLP since it obtains the smallest $L2_{\widetilde{CDF}}$ feasible loss with corresponding smallest $L2_{CDF}$ and $L2_{PDF}$ loss values in the validation sample. Similar performances of models 1, 4 and 6 are observed for the test sample, denoted in rows 4, 5, and 6.

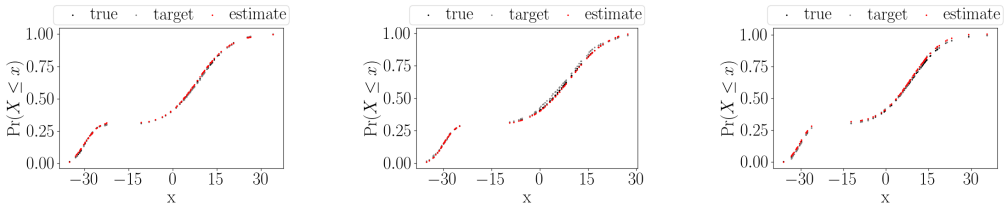
model	optimal	1	2	3	4	5	6
Validation sample with $L2_{\widehat{CDF}}$= 20.288 (1.579)							
$L2_{\widehat{CDF}}$	4.127	5.082	6.196	6.854	4.936	8.573	5.498
	(0.123)	(0.165)	(0.185)	(0.212)	(0.155)	(0.220)	(0.232)
$L2_{CDF}$	18.147	19.260	19.260	20.043	17.767	22.356	20.217
	(1.569)	(1.546)	(1.541)	(1.653)	(1.527)	(1.583)	(1.623)
$L2_{PDF}$	0.721	1.177	1.177	1.219	0.641	2.027	1.044
	(0.037)	(0.027)	(0.038)	(0.034)	(0.033)	(0.046)	(0.050)
Test sample with $L2_{\widehat{CDF}}$= 15.498 (1.078)							
$L2_{\widehat{CDF}}$	8.659	9.027	10.915	10.245	8.495	12.280	8.655
	(0.447)	(0.481)	(0.679)	(0.460)	(0.399)	(0.555)	(0.452)
$L2_{CDF}$	17.783	17.155	18.865	18.757	17.131	21.465	18.344
	(1.346)	(1.293)	(1.348)	(1.336)	(1.397)	(1.684)	(1.411)
$L2_{PDF}$	0.684	0.611	1.081	1.126	0.571	1.913	0.939
	(0.040)	(0.024)	(0.048)	(0.041)	(0.033)	(0.073)	(0.056)
Values in the table scaled by 10^{-3}							

Table 5.2 small sample results of mixed normal data
L2 losses for mixed normal data, small sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

Figures 5.1 and 5.2 present the best, average, and worst CDF and PDF estimates for the small test sample in terms of $L2_{\widehat{CDF}}$ loss for model 4. The best CDF estimate in Figure 5.1(a) closely fits the tails of the distribution. In general, the CDF losses are similar across models with smallest, average, and largest $L2_{\widehat{CDF}}$ loss, with small differences in the middle of the distribution. From these models, the average model in Figure 5.1(b) is closest to the mean value of the $L2_{\widehat{CDF}}$ loss for model 4 in Table 5.2. There is a clear distinction between the approximated target CDF and the true CDF, potentially due to the small sample size. Figure 5.1(c) also shows a clear distinction between the target and true CDF, around the tails of the distribution.

The PDF estimates of the models in Figure 5.2 show the differences between models more clearly. The average model in figure 5.2(b) has relatively larger losses around the right mode of the distribution, the worst model in Figure 5.2(c) has relatively larger losses around the right mode and the tails of the distribution, and the best model in 5.2(a) has relatively larger losses around the left mode.

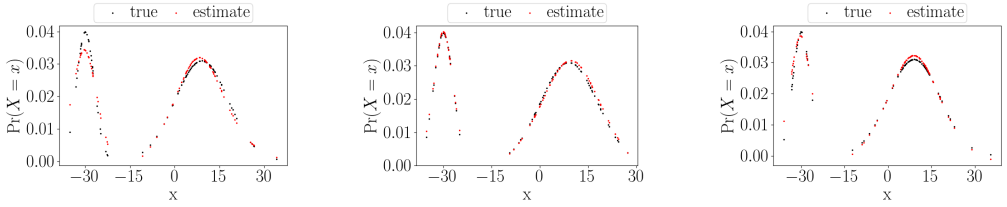
The squared errors $S2_{\widehat{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 4 are provided in Figure 5.3. Figure 5.3(a) illustrates that most points are closely fitted to the target CDF, with a $S2_{\widehat{CDF}}$ smaller than $1.25 \cdot 10^{-3}$ in 95% of the cases. Relatively large squared errors in Figures 5.3(a) and 5.3(b) occur around the two modes of the distribution with some exceptions for the $S2_{CDF}$ errors which show relatively large errors around the tails of the distribution as well. We expect that this difference is due to the inaccurate approximation of the target CDF compared to the true CDF, which is in line with the differences between the target and true CDF values in Figure 5.1(c). The corresponding PDF estimate discrepancies are illustrated in Figure 5.3(c). The PDF estimates show relatively smaller errors around the right mode compared to the left mode similar to the simulation example in Figure 5.2(a).



(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.1 small sample CDF estimates

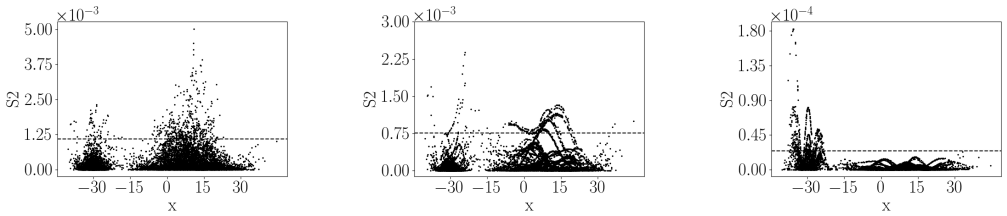
True, target, and estimated CDF using the proposed method for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 3 neurons are given.



(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.2 small sample PDF estimates

True and estimated PDF using the proposed method for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 3 neurons are given.



(a) $S2_{\widehat{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure 5.3 small sample squared errors

Squared errors calculated using the $S2_{\widehat{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the small sample size, using 2 hidden layers and 3 neurons of the proposed method.

Medium sample results:

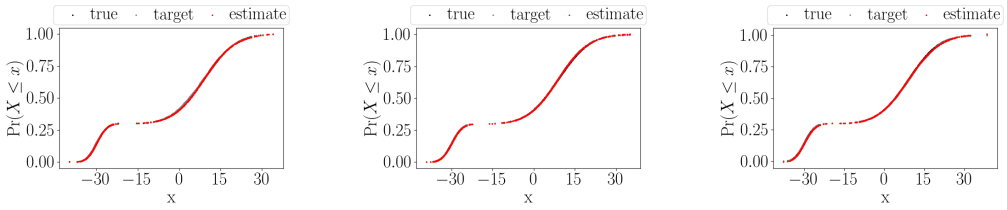
Table 5.3 presents the medium sample results for the validation and test samples. The obtained loss values are larger in the medium sample compared to the small sample since the loss functions in Section 4.2 are the total losses from all observations. Models 1, 4, and 6 are again the three best performing models in the validation and test samples, with the smallest $L2_{\widehat{CDF}}$ loss value obtained by model 4 in row 4. For the medium sample results, we mainly focus on model 1 that obtains the smaller $L2_{\widehat{CDF}}$ feasible loss in the validation sample.

model	optimal	1	2	3	4	5	6
Validation sample with $L2_{\widehat{CDF}}$= 23.321 (1.921)							
$L2_{\widehat{CDF}}$	5.022	6.719	10.006	9.076	7.997	17.928	7.873
	(0.168)	(0.293)	(0.491)	(0.423)	(0.704)	(0.572)	(0.473)
$L2_{CDF}$	22.532	21.828	25.441	24.192	24.779	34.165	26.741
	(1.937)	(1.974)	(2.090)	(1.921)	(2.387)	(2.379)	(2.084)
$L2_{PDF}$	0.870	0.618	1.625	1.351	0.688	3.835	1.227
	(0.051)	(0.032)	(0.054)	(0.045)	(0.040)	(0.106)	(0.056)
Test sample with $L2_{\widehat{CDF}}$= 19.001 (1.665)							
$L2_{\widehat{CDF}}$	13.200	13.200	15.640	16.997	9.935	15.361	13.289
	(1.060)	(0.811)	(1.611)	(1.342)	(0.638)	(0.801)	(1.263)
$L2_{CDF}$	20.663	18.696	22.028	25.037	17.044	22.762	22.047
	(1.405)	(1.337)	(2.012)	(1.897)	(1.090)	(1.309)	(1.573)
$L2_{PDF}$	0.806	0.566	1.064	1.019	0.659	2.860	1.186
	(0.042)	(0.022)	(0.036)	(0.041)	(0.026)	(0.104)	(0.060)
Values in the table scaled by 10^{-3}							

Table 5.3 medium sample results of mixed normal data
L2 losses for mixed normal data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

The best, average, and worst CDF and PDF estimates in terms of $L2_{\widehat{CDF}}$ loss estimated by model 1 are shown in Figures 5.4 and 5.5, respectively. The CDF losses are similar across the models with smallest, average, and largest $L2_{\widehat{CDF}}$ loss, with small differences in the tails of the distribution. This is more easily retrieved from Figure 5.5. All models in Figure 5.5 have relatively larger CDF losses for the right tail of the distribution, while the best and average PDF estimates, depicted in Figures 5.5(a) and 5.5(b) have additionally larger losses at the left tail of the distribution.

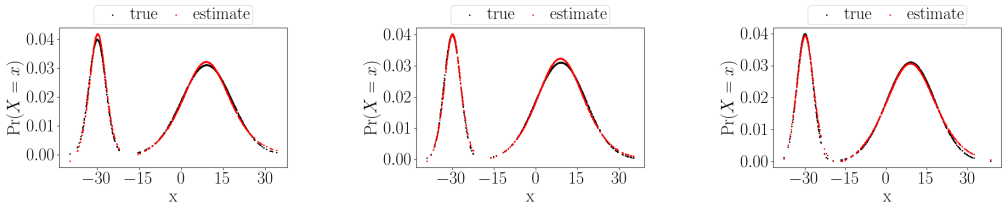
The squared errors $S2_{\widehat{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 1 are shown in Figure 5.6. The squared errors $S2_{\widehat{CDF}}$ and $S2_{CDF}$ are large at the right tail of the distribution in Figures 5.6(a) and 5.6(b) in line with Figure 5.4, while 5.6(a) shows additional large errors around the second mode of the distribution. The difference between these figures are the result of the approximation between the true and target CDF. In addition, squared errors in PDF estimates in Figure 5.6(c) are larger in the left tail of the distribution in line with Figures 5.5(a) and 5.5(b).



(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.4 medium sample CDF estimates

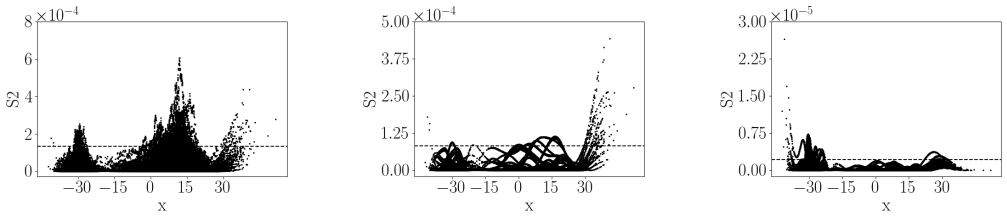
True, target, and estimated CDF using the proposed method for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 1 hidden layer and 3 neurons are given.



(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.5 medium sample PDF estimates

True and estimated PDF using the proposed method for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 1 hidden layer and 3 neurons are given.



(a) $S2_{\widehat{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure 5.6 medium sample squared errors

Squared errors calculated using the $S2_{\widehat{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the medium sample size, using 2 hidden layers and 3 neurons of the proposed method.

Large sample results:

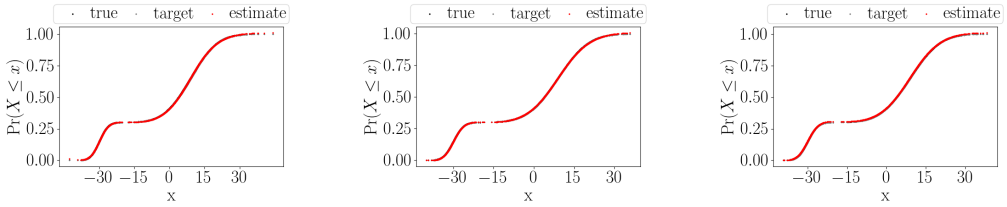
Table 5.4 presents the large sample results for all models for the validation sample and test sample. The large sample size obtains more wide-spread $L2_{\widehat{CDF}}$ loss values across models for the validation sample than for the smaller sample sizes. Only model 1 performs best due to the small $L2_{\widehat{CDF}}$, $L2_{CDF}$, and $L2_{PDF}$ loss values opposed to the small and medium sample, where models 1, 4, and 6 performed best. We therefore focus on model 1 in the following large sample results.

model	optimal	1	2	3	4	5	6
Validation sample with $L2_{\widehat{CDF}}$= 22.110 (1.881)							
$L2_{\widehat{CDF}}$	9.600	13.144	25.246	25.973	17.411	61.885	19.396
	(0.374)	(0.565)	(1.364)	(1.873)	(0.996)	(3.218)	(1.399)
$L2_{CDF}$	24.989	25.875	40.200	41.458	31.442	77.857	36.352
	(1.938)	(1.862)	(2.886)	(3.053)	(2.449)	(5.202)	(2.668)
$L2_{PDF}$	1.423	0.900	4.276	3.398	0.959	14.833	1.720
	(0.092)	(0.036)	(0.127)	(0.067)	(0.041)	(0.181)	(0.090)
Test sample with $L2_{\widehat{CDF}}$= 17.210 (1.424)							
$L2_{\widehat{CDF}}$	19.190	17.961	24.470	17.679	23.585	55.906	22.928
	(1.855)	(1.010)	(1.822)	(1.036)	(2.977)	(2.103)	(2.883)
$L2_{CDF}$	27.302	24.717	30.018	25.391	30.462	62.274	34.563
	(2.540)	(1.861)	(2.360)	(1.731)	(3.639)	(3.077)	(4.129)
$L2_{PDF}$	1.202	0.740	3.380	2.275	0.749	11.831	1.618
	(0.076)	(0.023)	(0.122)	(0.049)	(0.024)	(0.188)	(0.053)
Values in the table scaled by 10^{-3}							

Table 5.4 large sample results of mixed normal data
L2 losses for mixed normal data, large sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

The best, average, and worst CDF and PDF estimates in terms of $L2_{\widehat{CDF}}$ loss of model 1 are shown in Figures 5.7 and 5.8. All estimates perform relatively worse in the left and right tails of the distribution both in terms of the CDF loss and the PDF loss. In addition, the CDF estimates have relatively worse fit in the right tail of the distribution. These misfits are reflected in the corresponding PDF estimates depicted in Figure 5.8. The CDF estimates shown in Figure 5.7 have a slight decrease in the left tail of the distribution. In other words, the CDF estimate violates the monotone increasing function property and the corresponding PDF estimates, illustrated in Figures 5.8, have negative probabilities. This inconsistency can be corrected for by changing the decreasing CDF points with the next non-decreasing CDF point or by implementing a penalty as mentioned in Section 4.2.

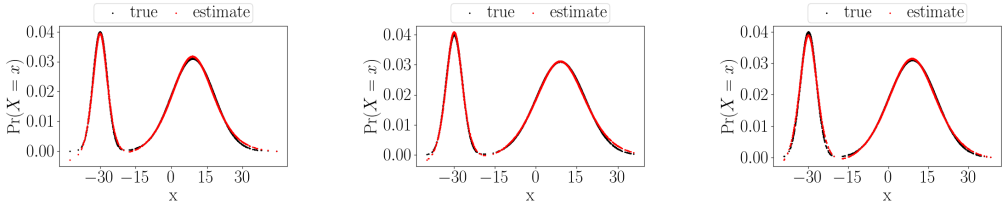
The squared errors $L2_{\widehat{CDF}}$, $L2_{CDF}$, and $L2_{PDF}$ for each data point over 100 simulation replications using model 1 are shown in Figure 5.9. From Figures 5.9(a) and 5.9(b) the largest errors are obtained by the right tail, similarly as for the medium sample. The large error values for the second mode of Figure 5.9(a) is due to the approximation of the target CDF, again in line with the medium sample. However, these approximation error values decrease as the sample size grows, see Figure 5.3 for the small sample results with squared errors not larger than $5.0 \cdot 10^{-3}$, see Figure 5.6 for the medium sample results with values not larger than $6.0 \cdot 10^{-4}$ and 5.9 for the large sample results



(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.7 large sample CDF estimates

True, target, and estimated CDF using the proposed method for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the large sample size using 1 hidden layer and 3 neurons are given.

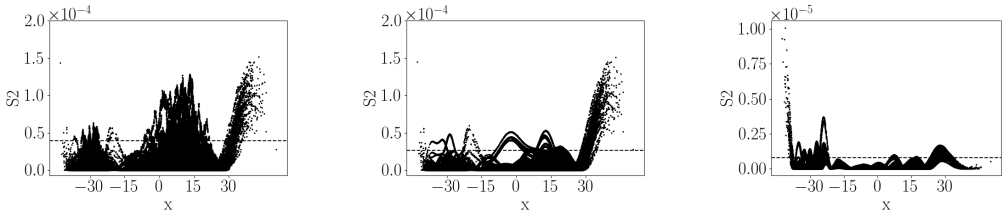


(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.8 large sample PDF estimates

True and estimated PDF using the proposed method for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the large sample size using 1 hidden layer and 3 neurons are given.

with values not larger than $1.5 \cdot 10^{-4}$. The corresponding $S2_{PDF}$ errors illustrated in Figure 5.9(c) show relatively large errors for the left tail of the distribution in line with the small and medium sample results. Furthermore, the misfit in the right tail of the distribution of the CDF estimates due the neural network is also translated to the PDF estimates. This is in line with Figure 5.8. These $L2_{PDF}$ squared error values are smaller than the $L2_{\widetilde{CDF}}$ and $L2_{CDF}$ error values again in line with the medium sample.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure 5.9 large sample squared errors
Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the large sample size, using 1 hidden layer and 3 neurons of the proposed method.

Hence the CDF and PDF estimates improve as sample sizes increase. Furthermore, model 1, using 1 hidden layer and 3 neurons is used in the medium sample size and large sample size. For the small sample size this model closely follows optimal model 4, using 2 hidden layers and 3 neurons. Due to the small number of available observations in the small sample, more non-linearity is needed by using 1 more hidden layer. However as more observations are added, model 1 outperforms model 4. Hence both models 1 and 4 fit this distribution with similar $L2_{PDF}$ loss values.

Now the results are benchmarked against two baseline methods, Trentin et al. (2018)'s method and the kernel density estimator. Trentin et al. (2018)'s results are shown for the small, medium and large sample sizes, using 10,000 training epochs and a learning rate of 0.01. The batch size should be as large as the test size in order to estimate consistent target PDF values due to the bandwidth size. This initial bandwidth size is determined empirically, as Trentin et al. (2018) does. Three different initial bandwidth sizes are used, $5.0/\sqrt{T-1}$, $7.5/\sqrt{T-1}$ and $10.0/\sqrt{T-1}$. The logistic output function is used as activation function in all layers except for the output layer.

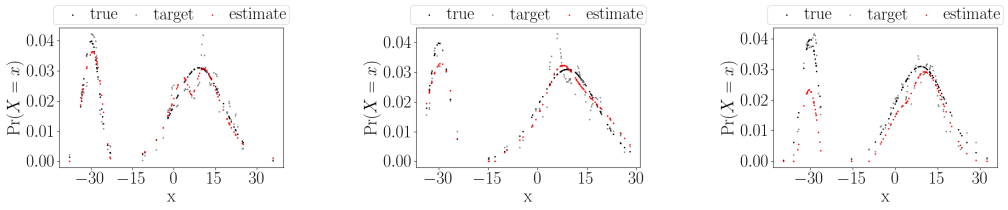
Small sample results for Trentin et al. (2018):

Table 5.5 presents the small sample results for the validation sample and test sample using Trentin et al. (2018). Out of 100 simulation replications model 5 is the best performing model obtaining the smallest $L2_{\widetilde{PDF}}$ loss of the validation sample irrespective which initial bandwidth is used, see rows 1, 3, and 5. This also holds for the $L2_{\widetilde{PDF}}$ losses in the test sample. Initial bandwidth $10.0/\sqrt{T-1}$ obtains the smallest $L2_{\widetilde{PDF}}$ loss for the validation sample and test sample. Using initial bandwidth $7.5/\sqrt{T-1}$ obtains the smallest $L2_{PDF}$ loss for the test sample. Estimates using initial bandwidth $5.0/\sqrt{T-1}$ perform worst with largest $L2_{\widetilde{PDF}}$ and $L2_{PDF}$ losses. Model 5 using initial bandwidth $10.0/\sqrt{T-1}$ is chosen as optimal model due to the smallest $L2_{\widetilde{PDF}}$ feasible loss with an $L2_{PDF}$ loss of $1.492 \cdot 10^{-3}$ for the validation sample and $1.055 \cdot 10^{-3}$ for the test sample. The proposed method outperforms Trentin et al. (2018) in this application with an $L2_{PDF}$ loss of $0.641 \cdot 10^{-3}$ for the validation sample and $0.571 \cdot 10^{-3}$ for the test sample, see Table 5.2.

model	optimal	1	2	3	4	5	6
Validation sample							
bandwidth $5.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 2.328 (0.067)							
$L2_{\widetilde{PDF}}$	2.394	5.548	4.139	4.319	5.670	2.516	5.299
	(0.093)	(0.161)	(0.107)	(0.137)	(0.138)	(0.109)	(0.139)
$L2_{PDF}$	1.689	4.124	2.885	3.026	4.584	1.918	4.572
	(0.099)	(0.117)	(0.099)	(0.106)	(0.062)	(0.122)	(0.078)
bandwidth $7.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 1.486 (0.051)							
$L2_{\widetilde{PDF}}$	1.491	4.571	2.975	3.233	4.659	1.626	4.398
	(0.071)	(0.141)	(0.100)	(0.115)	(0.121)	(0.094)	(0.116)
$L2_{PDF}$	1.487	4.134	2.620	2.824	4.567	1.672	4.533
	(0.089)	(0.112)	(0.104)	(0.107)	(0.073)	(0.113)	(0.074)
bandwidth $10.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 1.080 (0.041)							
$L2_{\widetilde{PDF}}$	0.994	3.890	2.421	2.583	4.029	1.125	3.836
	(0.055)	(0.123)	(0.109)	(0.099)	(0.098)	(0.081)	(0.095)
$L2_{PDF}$	1.266	4.057	2.540	2.691	4.493	1.492	4.469
	(0.069)	(0.109)	(0.104)	(0.098)	(0.068)	(0.107)	(0.069)
Test sample							
bandwidth $5.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 1.918 (0.064)							
$L2_{\widetilde{PDF}}$	2.212	4.674	3.711	3.115	5.662	2.026	5.341
	(0.147)	(0.163)	(0.169)	(0.150)	(0.127)	(0.155)	(0.114)
$L2_{PDF}$	1.505	3.473	2.688	1.990	4.742	1.362	4.703
	(0.154)	(0.147)	(0.192)	(0.174)	(0.128)	(0.155)	(0.114)
bandwidth $7.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 1.226 (0.046)							
$L2_{\widetilde{PDF}}$	1.344	4.132	2.719	2.404	4.545	1.090	3.993
	(0.096)	(0.181)	(0.138)	(0.133)	(0.101)	(0.053)	(0.126)
$L2_{PDF}$	1.175	3.355	2.244	1.857	4.426	1.020	4.102
	(0.097)	(0.168)	(0.160)	(0.130)	(0.115)	(0.081)	(0.144)
bandwidth $10.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 0.899 (0.035)							
$L2_{\widetilde{PDF}}$	0.874	3.228	1.927	1.701	3.901	0.747	3.627
	(0.069)	(0.142)	(0.114)	(0.102)	(0.092)	(0.046)	(0.124)
$L2_{PDF}$	1.131	3.584	2.094	1.744	4.494	1.055	4.293
	(0.085)	(0.171)	(0.147)	(0.106)	(0.097)	(0.081)	(0.139)
Values in the table scaled by 10^{-3}							

Table 5.5 small sample results of mixed normal data by Trentin et al. (2018)
L2 losses for mixed normal data, small sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated PDF obtained by Trentin et al. (2018) for the validation and test samples.

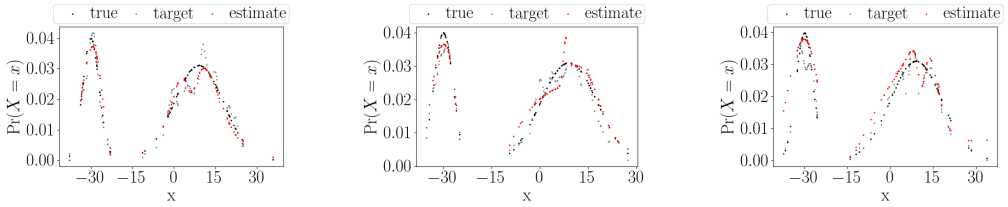
The best, average, and worst CDF and PDF estimates in terms of $L2_{\widetilde{PDF}}$ loss of model 5 using different bandwidths are shown in Figures 5.10, 5.11, and 5.12. Across bandwidths target PDFs are different with more volatility for the smaller bandwidths and all estimates are quite volatile or show large discrepancies. For all estimates, the tails of the first distribution are estimated well with only differences in the peak of the mode except for the worst estimate using bandwidth $10.0/\sqrt{T-1}$, see Figure 5.12(c). This estimate misfits the first distribution completely.



(a) smallest $L2_{\widetilde{PDF}}$ loss (b) average $L2_{\widetilde{PDF}}$ loss (c) largest $L2_{\widetilde{PDF}}$ loss

Figure 5.10 small sample PDF estimates by Trentin et al. (2018) with initial bandwidth $5.0/\sqrt{T-1}$

True, target, and estimated PDF using Trentin et al. (2018)'s method with an initial bandwidth of $5.0/\sqrt{T-1}$ for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widetilde{PDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 4 neurons are given.

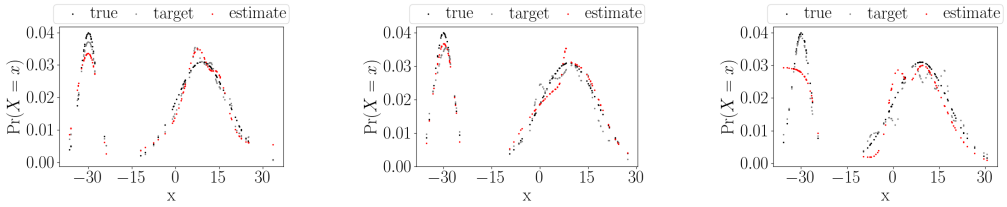


(a) smallest $L2_{\widetilde{PDF}}$ loss (b) average $L2_{\widetilde{PDF}}$ loss (c) largest $L2_{\widetilde{PDF}}$ loss

Figure 5.11 small sample PDF estimates by Trentin et al. (2018) with initial bandwidth $7.5/\sqrt{T-1}$

True, target, and estimated PDF using Trentin et al. (2018)'s method with an initial bandwidth of $7.5/\sqrt{T-1}$ for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widetilde{PDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 4 neurons are given.

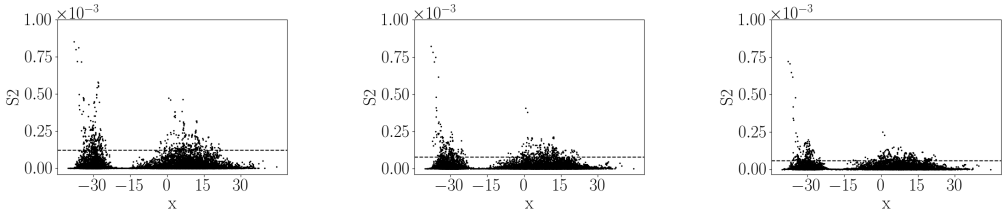
The squared errors $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ of model 5 using different bandwidths for each data point over 100 simulation replications are shown in Figures 5.13 and 5.14. Similar patterns and error values are shown across bandwidths. This implies that Trentin et al. (2018)'s method is quite robust against different initial bandwidth values for this distribution. Overall, $S2_{\widetilde{PDF}}$ errors are larger than $S2_{PDF}$ errors on the same domains, especially in the left tail of the first distribution and the mode of the second distribution. The proposed method has relatively similar large errors for the left tail of the first distribution though the error values are 10 times smaller.



(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.12 small sample PDF estimates by Trentin et al. (2018) with initial bandwidth $10.0/\sqrt{T-1}$.

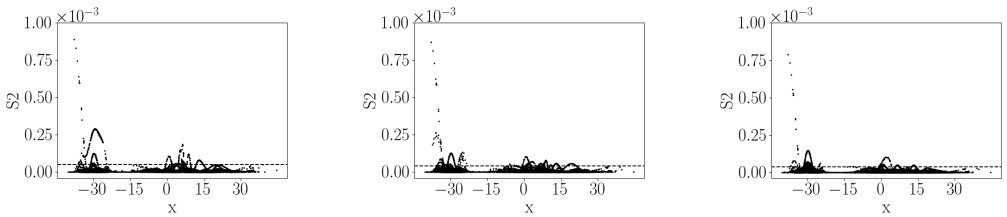
True, target, and estimated PDF using Trentin et al. (2018)'s method with an initial bandwidth of $10.0/\sqrt{T-1}$ for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widehat{PDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 4 neurons are given.



(a) $5.0/\sqrt{T-1}$ (b) $7.5/\sqrt{T-1}$ (c) $10.0/\sqrt{T-1}$

Figure 5.13 small sample squared errors based on $S2_{\widehat{PDF}}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{\widehat{PDF}}$ differences together with the 95^{th} percentile for the small sample size, using 2 hidden layers and 4 neurons of Trentin et al. (2018)'s method.



(a) $5.0/\sqrt{T-1}$ (b) $7.5/\sqrt{T-1}$ (c) $10.0/\sqrt{T-1}$

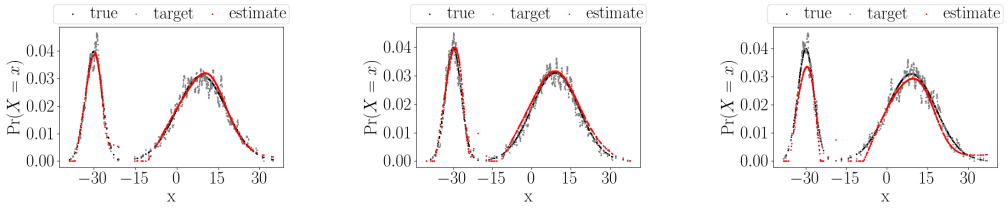
Figure 5.14 small sample squared errors based on $S2_{PDF}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{PDF}$ differences together with the 95^{th} percentile for the small sample size, using 2 hidden layers and 4 neurons of Trentin et al. (2018)'s method.

Medium sample results for Trentin et al. (2018):

Table 5.6 presents the medium sample results for the validation sample and test sample using Trentin et al. (2018). Model 5 obtains the smallest $L2_{\widetilde{PDF}}$ and $L2_{PDF}$ values for the medium sample as well as for the small sample, irrespective of initial bandwidth. The largest bandwidth $10.0/\sqrt{T-1}$ outperforms the smaller ones. Hence similar conclusions hold for the small and medium sample sizes. The smallest $L2_{PDF}$ loss is $5.304 \cdot 10^{-3}$ for the validation sample and $3.525 \cdot 10^{-3}$ for the test sample. The proposed method outperforms Trentin et al. (2018) in the medium sample as in the small sample with an $L2_{PDF}$ loss of $0.618 \cdot 10^{-3}$ for the validation sample and $0.566 \cdot 10^{-3}$ for the test sample, see Table 5.3. The $L2_{PDF}$ loss for the small and medium sample size estimated by the proposed method, shown in Tables 5.2 and 5.3, do not differ substantially. Hence, increasing the number of observations improves the proposed method by using 1 hidden layer less and sustaining a similar $L2_{PDF}$ loss value. Trentin et al. (2018)'s approach does not benefit from adding more observations as they still need 2 hidden layers to estimate the distribution and have slightly larger loss values for the medium sample compared to the small sample.

The best, average, and worst CDF and PDF estimates in terms of $L2_{\widetilde{PDF}}$ loss of model 5 using different bandwidths are shown in Figures 5.15, 5.16, and 5.17. Across bandwidths target PDFs are different. The target estimated by initial bandwidth $7.5/\sqrt{T-1}$ closely approximates the peak of the first mode. For the other two bandwidths, the peak estimated by the target is slightly higher than the true PDF. The estimates of the worst models with bandwidths $5.0/\sqrt{T-1}$ and $10.0/\sqrt{T-1}$ misfit the peak of the first mode. However, all other estimates fit nicely around the true peak as well as the rest of the distribution components except for the tails. The right tail of the first distribution and the left tail of the second distribution is misfitted irrespective of which bandwidth is used. Most of the models fit the outer left and outer right tail closely though. All tails except for the right tail of the second distribution are negative estimates. These are truncated at zero which result in the horizontal lines in the PDF curves. The target PDFs are quite volatile as in the small sample size, especially for the smaller bandwidths. Unlike the small sample results, the modes of the medium sample estimates correspond to the modes of the true density. Hence, increasing the number of observations increase the accuracy of the PDF estimates.



(a) smallest $L2_{\widetilde{PDF}}$ loss (b) average $L2_{\widetilde{PDF}}$ loss (c) largest $L2_{\widetilde{PDF}}$ loss

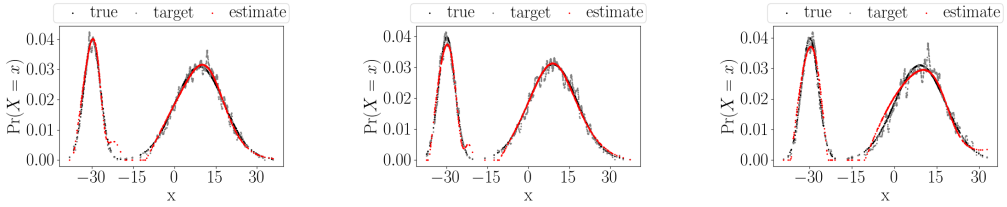
Figure 5.15 medium sample PDF estimates by Trentin et al. (2018) with initial bandwidth $5.0/\sqrt{T-1}$
True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $5.0/\sqrt{T-1}$ for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widetilde{PDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 4 neurons are given.

The squared errors $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ for model 5 using different bandwidths for each data point over 100 simulation replications are shown in Figures 5.18 and 5.19. The patterns

model	optimal	1	2	3	4	5	6
Validation sample							
bandwidth $5.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 8.360 (0.145)							
$L2_{\widetilde{PDF}}$	11.774	37.371	22.097	17.560	45.613	11.896	39.933
	(0.268)	(0.554)	(0.400)	(0.411)	(0.411)	(0.320)	(0.916)
$L2_{PDF}$	4.736	29.842	14.736	10.024	38.600	4.865	33.135
	(0.200)	(0.472)	(0.332)	(0.324)	(0.206)	(0.272)	(0.910)
bandwidth $7.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 5.577 (0.121)							
$L2_{\widetilde{PDF}}$	9.393	34.463	20.004	14.923	42.892	9.572	37.463
	(0.328)	(0.604)	(0.512)	(0.518)	(0.453)	(0.376)	(0.982)
$L2_{PDF}$	5.200	29.797	15.531	10.461	38.693	5.379	33.392
	(0.295)	(0.506)	(0.460)	(0.458)	(0.253)	(0.350)	(0.957)
bandwidth $10.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 4.170 (0.106)							
$L2_{\widetilde{PDF}}$	7.839	32.459	18.974	13.905	41.108	8.031	35.470
	(0.264)	(0.549)	(0.523)	(0.497)	(0.404)	(0.309)	(1.001)
$L2_{PDF}$	5.185	29.527	16.148	10.940	38.639	5.304	33.114
	(0.258)	(0.457)	(0.482)	(0.443)	(0.228)	(0.291)	(1.006)
Test sample							
bandwidth $5.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 7.018 (0.123)							
$L2_{\widetilde{PDF}}$	10.028	29.498	18.336	13.378	43.034	9.991	23.118
	(0.248)	(0.530)	(0.324)	(0.435)	(0.495)	(0.249)	(1.526)
$L2_{PDF}$	3.027	21.880	9.513	5.396	37.121	3.000	16.833
	(0.194)	(0.362)	(0.273)	(0.255)	(0.339)	(0.194)	(1.591)
bandwidth $7.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 4.669 (0.100)							
$L2_{\widetilde{PDF}}$	7.615	31.218	20.534	10.231	40.520	6.865	19.678
	(0.454)	(0.297)	(0.424)	(0.517)	(0.451)	(0.169)	(1.526)
$L2_{PDF}$	3.327	26.114	15.099	5.237	37.051	2.584	15.972
	(0.444)	(0.396)	(0.447)	(0.421)	(0.335)	(0.133)	(1.611)
bandwidth $10.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 3.477 (0.085)							
$L2_{\widetilde{PDF}}$	6.631	32.637	13.647	9.639	39.591	6.402	19.416
	(0.395)	(0.956)	(0.238)	(0.431)	(0.479)	(0.249)	(1.427)
$L2_{PDF}$	3.797	29.050	9.361	5.979	37.519	3.525	17.179
	(0.428)	(0.826)	(0.214)	(0.328)	(0.356)	(0.258)	(1.502)
Values in the table scaled by 10^{-3}							

Table 5.6 medium sample results of mixed normal data by Trentin et al. (2018)

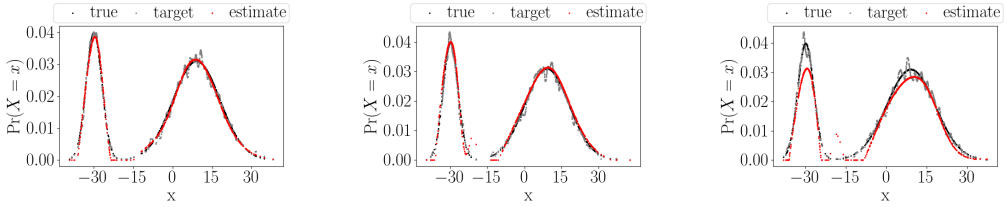
L2 losses for mixed normal data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated PDF obtained by Trentin et al. (2018) for the validation and test samples.



(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.16 medium sample PDF estimates by Trentin et al. (2018) with initial bandwidth $7.5/\sqrt{T-1}$

True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $7.5/\sqrt{T-1}$ ***for mixed normal data. The best, average, and worst estimates in terms of*** $L2_{\widehat{PDF}}$ ***loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 4 neurons are given.***



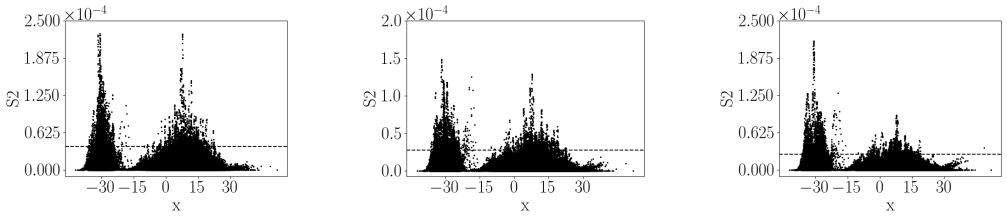
(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.17 medium sample PDF estimates by Trentin et al. (2018) with initial bandwidth $10.0/\sqrt{T-1}$**.**

True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $10.0/\sqrt{T-1}$ ***for mixed normal data. The best, average, and worst estimates in terms of*** $L2_{\widehat{PDF}}$ ***loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 4 neurons are given.***

and error values across bandwidths are similar to the small sample. The values of the scales are comparable for bandwidths $7.5/\sqrt{T-1}$ and $10.0/\sqrt{T-1}$, the values for $5.0/\sqrt{T-1}$ are slightly larger for the $S2_{\widehat{PDF}}$ errors. Furthermore, the $S2_{\widehat{PDF}}$ and $S2_{PDF}$ errors differ from each other. The estimated PDF compared to the target PDF especially misfits the first and second modes due to the erratic nature of the target PDF.

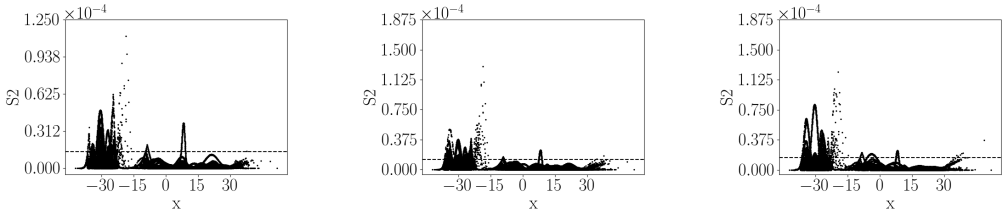
Similar to small sample results, Trentin et al. (2018)'s method is quite robust against different bandwidths for this distribution. However, the method is not robust against different sample sizes. For the small sample, the method struggled mainly with estimating the left tail of the first distribution and the mode of the second distribution. For the medium sample, large error values are found for the right tails of the first and second distributions just as for the proposed method. Though the proposed method has 10 times smaller error values. Moreover, the tails in the middle are truncated to zero to avoid negative probabilities.



(a) $5.0/\sqrt{T-1}$ (b) $7.5/\sqrt{T-1}$ (c) $10.0/\sqrt{T-1}$

Figure 5.18 medium sample squared errors based on $S2_{\widehat{PDF}}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{\widehat{PDF}}$ differences together with the 95^{th} percentile for the medium sample size, using 2 hidden layers and 4 neurons of Trentin et al. (2018)'s method.



(a) $5.0/\sqrt{T-1}$ (b) $7.5/\sqrt{T-1}$ (c) $10.0/\sqrt{T-1}$

Figure 5.19 medium sample squared errors based on $S2_{PDF}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{PDF}$ differences together with the 95^{th} percentile for the medium sample size, using 2 hidden layers and 4 neurons of Trentin et al. (2018)'s method.

Large sample results for Trentin et al. (2018):

Table 5.7 presents the large sample results for the validation sample and test sample using Trentin et al. (2018)'s method. Model 5 obtains the smallest $L2_{\widehat{PDF}}$ and $L2_{PDF}$ values for the large sample using bandwidths $7.5/\sqrt{T-1}$ and $10.0/\sqrt{T-1}$, model 6 is optimal for bandwidth $5.0/\sqrt{T-1}$ for the validation sample. Bandwidth $7.5/\sqrt{T-1}$ obtains the smallest $L2_{\widehat{PDF}}$ loss in the validation sample. The smallest $L2_{PDF}$ loss is $15.179 \cdot 10^{-3}$ for the validation sample and $12.342 \cdot 10^{-3}$ for the test sample. The proposed method

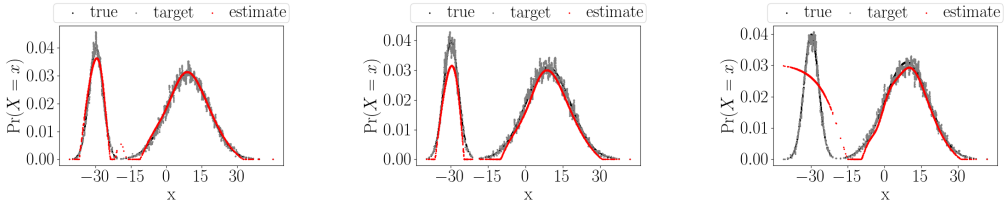
again outperforms Trentin et al. (2018) with an $L2_{PDF}$ loss of $0.900 \cdot 10^{-3}$ for the validation sample and $0.740 \cdot 10^{-3}$ for the test sample, see Table 5.4. Similarly as for the medium sample, by adding more observations the proposed method benefits by using 1 hidden layer less and sustaining a similar $L2_{PDF}$ loss value. Trentin et al. (2018)'s approach does not benefit from adding more observations as it still requires 2 hidden layers to estimate the distribution with a slightly larger $L2_{PDF}$ loss.

model	optimal	1	2	3	4	5	6
Validation sample							
bandwidth $5.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 16.993 (0.242)							
$L2_{\widetilde{PDF}}$	29.147	132.937	165.038	165.059	162.525	32.014	31.858
	(0.489)	(0.998)	(0.856)	(0.904)	(0.813)	(0.557)	(0.741)
$L2_{PDF}$	13.113	116.024	148.627	148.687	145.737	15.742	15.951
	(0.400)	(0.860)	(0.365)	(0.380)	(0.314)	(0.475)	(0.685)
bandwidth $7.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 11.289 (0.189)							
$L2_{\widetilde{PDF}}$	22.680	127.635	159.078	158.915	156.780	25.722	25.827
	(0.380)	(1.142)	(0.866)	(0.872)	(0.822)	(0.463)	(0.822)
$L2_{PDF}$	12.390	116.711	148.654	148.396	145.969	15.179	15.636
	(0.288)	(0.953)	(0.385)	(0.388)	(0.310)	(0.415)	(0.797)
bandwidth $10.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 8.402 (0.157)							
$L2_{\widetilde{PDF}}$	30.629	107.867	92.975	55.238	157.265	30.681	129.720
	(0.946)	(1.577)	(1.541)	(1.872)	(0.903)	(0.954)	(3.974)
$L2_{PDF}$	22.952	100.207	85.386	47.575	149.795	22.983	122.383
	(1.008)	(1.537)	(1.482)	(1.911)	(0.597)	(1.013)	(4.049)
Test sample							
bandwidth $5.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 14.189 (0.198)							
$L2_{\widetilde{PDF}}$	52.543	93.845	67.454	37.884	164.076	32.877	75.684
	(4.557)	(1.629)	(0.666)	(0.788)	(0.998)	(1.061)	(5.560)
$L2_{PDF}$	37.122	76.038	48.073	18.703	152.429	15.991	62.474
	(4.794)	(1.370)	(0.578)	(0.799)	(0.810)	(0.819)	(6.041)
bandwidth $7.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 9.419 (0.161)							
$L2_{\widetilde{PDF}}$	46.300	78.104	61.718	28.856	154.008	23.375	61.427
	(4.529)	(0.622)	(0.709)	(0.318)	(0.728)	(0.377)	(5.363)
$L2_{PDF}$	37.329	67.715	47.246	15.183	147.451	12.342	53.704
	(4.943)	(0.389)	(0.637)	(0.380)	(0.603)	(0.351)	(5.820)
bandwidth $10.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 7.009 (0.137)							
$L2_{\widetilde{PDF}}$	21.059	121.193	75.084	27.147	159.783	20.986	63.980
	(0.701)	(2.982)	(1.740)	(0.636)	(0.839)	(0.700)	(5.748)
$L2_{PDF}$	12.636	115.740	65.613	15.634	155.865	12.616	58.683
	(0.557)	(3.055)	(1.796)	(0.513)	(0.882)	(0.557)	(6.236)
Values in the table scaled by 10^{-3}							

Table 5.7 large sample results of mixed normal data by Trentin et al. (2018)
L2 losses for mixed normal data, large sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated PDF obtained by Trentin et al. (2018) for the validation and test samples.

The best, average and worst CDF and PDF estimates in terms of $L2_{\widetilde{PDF}}$ loss of models 5

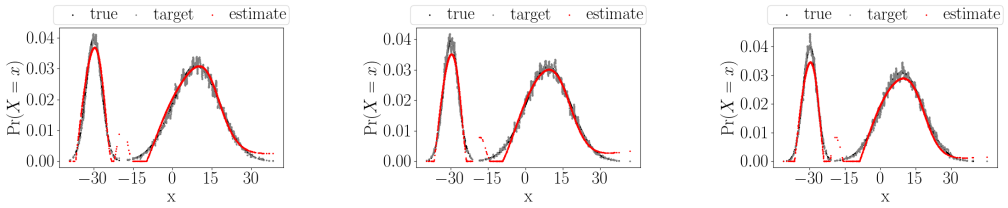
and 6 using different bandwidths are shown in Figures 5.20, 5.21, and 5.22. Differences between target PDFs across bandwidths are again minor. The average and worst estimates with bandwidth $5.0/\sqrt{T-1}$ misfit the first distribution, by either not reaching the same peak value or misfitting the shape. The other two bandwidths struggle with the tails of the second distribution. Furthermore, these Figures show non-symmetrical modes for the second distribution. Similarly to the medium sample, all bandwidths show negative values between the two distributions as well as an additional mode around the value of -20.



(a) smallest $L2_{\widetilde{PDF}}$ loss (b) average $L2_{\widetilde{PDF}}$ loss (c) largest $L2_{\widetilde{PDF}}$ loss

Figure 5.20 large sample PDF estimates by Trentin et al. (2018) with initial bandwidth $5.0/\sqrt{T-1}$

True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $5.0/\sqrt{T-1}$ for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widetilde{PDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 4 neurons are given.

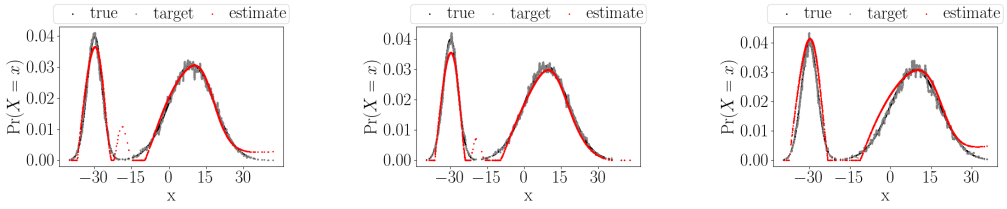


(a) smallest $L2_{\widetilde{PDF}}$ loss (b) average $L2_{\widetilde{PDF}}$ loss (c) largest $L2_{\widetilde{PDF}}$ loss

Figure 5.21 large sample PDF estimates by Trentin et al. (2018) with initial bandwidth $7.5/\sqrt{T-1}$

True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $7.5/\sqrt{T-1}$ for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widetilde{PDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 4 neurons are given.

The squared errors $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ of models 5 and 6 using different bandwidths for each data point over 100 simulation replications are shown in Figures 5.23 and 5.24. The patterns and error values for bandwidths $7.5/\sqrt{T-1}$ and $10.0/\sqrt{T-1}$ are similar to the medium sample size. The values of the scales are comparable for bandwidths $7.5/\sqrt{T-1}$ and $10.0/\sqrt{T-1}$, the values for $5.0/\sqrt{T-1}$ are larger for the both $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ errors. This is also due to the use of a different model 6 instead of model 5. Furthermore, the $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ errors differ from each other. For the larger bandwidths, the estimated PDF compared to the target PDF especially misfits the first and second modes due to the erratic nature of the target PDF. The largest error peak corresponds to the third non-existent mode that is estimated in between the two modes. This peak is equally large for the $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ error values. The estimates are

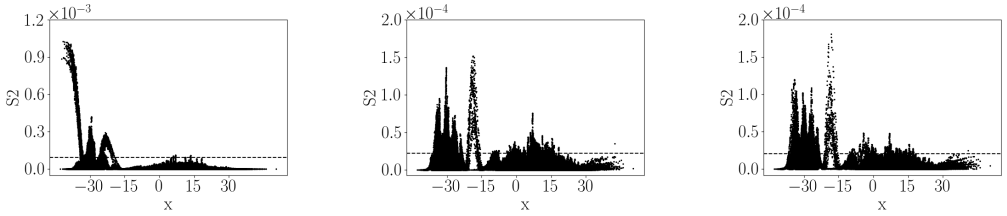


(a) smallest $L2_{\widetilde{PDF}}$ loss (b) average $L2_{\widetilde{PDF}}$ loss (c) largest $L2_{\widetilde{PDF}}$ loss

Figure 5.22 large sample PDF estimates by Trentin et al. (2018) with initial bandwidth $10.0/\sqrt{T-1}$.

True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $10.0/\sqrt{T-1}$ for mixed normal data. The best, average, and worst estimates in terms of $L2_{\widetilde{PDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 4 neurons are given.

especially misfitting the first distribution and the dip between the two distributions. These conclusions are different for the smallest bandwidth. These estimates show large errors only for the left tail of the distribution. The difference between this bandwidth and the other larger bandwidths could be a consequence of using a different model. Especially large error values for the PDF estimates are found at the dip in between the two distributions, as well as the first distribution and the right tail of the second distribution. Although, different conclusions are drawn between the small and medium sample, the medium and large sample draw similar conclusions. Trentin et al. (2018)'s approach is robust against different initial bandwidths, though the smallest bandwidth chooses a different model than the other bandwidths. The results from model 6 are different than for model 5.

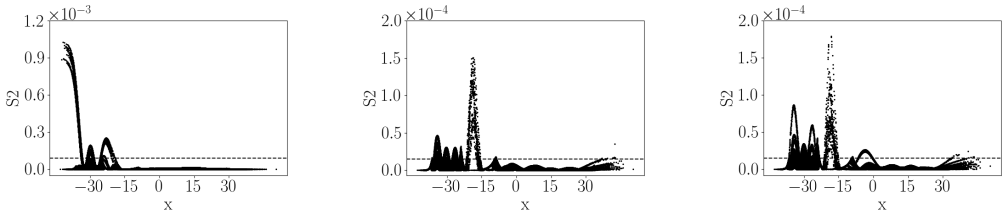


(a) $5.0/\sqrt{T-1}$ (b) $7.5/\sqrt{T-1}$ (c) $10.0/\sqrt{T-1}$

Figure 5.23 large sample squared errors based on $S2_{\widetilde{PDF}}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{\widetilde{PDF}}$ differences together with the 95^{th} percentile for the large sample size, using 2 hidden layers and 4 neurons of Trentin et al. (2018)'s method.

We next compare the results of the proposed method with KDE for all sample sizes.



(a) $5.0/\sqrt{T-1}$ (b) $7.5/\sqrt{T-1}$ (c) $10.0/\sqrt{T-1}$

Figure 5.24 large sample squared errors based on $S2_{PDF}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{PDF}$ differences together with the 95^{th} percentile for the large sample size, using 2 hidden layers and 4 neurons of Trentin et al. (2018)'s method.

Results for KDE:

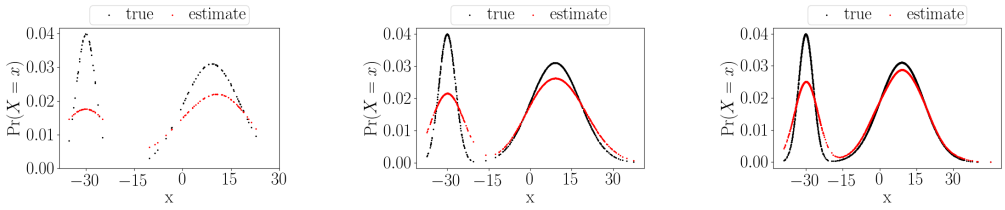
For the ease of comparison, we present the test sample results of the proposed method, Trentin et al. (2018)'s method with different bandwidths and the KDE for all sample sizes in Table 5.8. The KDE has a substantially larger $L2_{PDF}$ loss value compared to the proposed and Trentin et al. (2018) methods. This discrepancy enlarges as sample size grows since the loss functions in Section 4.2 are the total losses from all observations.Trentin et al. (2018)'s method shows this effect as well, though to a lesser extent. However, for the proposed method this effect is not observed. Similar $L2_{PDF}$ loss values are found across sample sizes. Trentin et al. (2018)'s method is quite robust against different bandwidths, as the $L2_{PDF}$ loss values are similar though larger than the $L2_{PDF}$ loss values of the proposed method. Model 5 is chosen as optimal for all sample sizes and bandwidths except for the smallest bandwidth $5.0/\sqrt{T-1}$ in the large sample size. Model 6 is chosen as optimal even though model 5 obtains a similar though slightly larger $L2_{\widetilde{PDF}}$ loss. Model 5 obtains a similar $L2_{PDF}$ loss as in the table with value 15.991, opposed to model 6 with a value of 62.474.

The average PDF estimates in terms of $L2_{PDF}$ loss estimated by the KDE are shown in Figure 5.25. For all sample sizes, the modes and tails are misfit. All simulation replications result in similar estimates, therefore it suffices to only show the average estimates, see Figures of the squared errors $S2_{PDF}$in Figure 5.26 for the small, medium and large samples, respectively. For all sample sizes the mode and tails of the first distribution are misfit in line with the PDF estimates shown in Figure 5.25. As the sample size grows, these misfits decrease in value but are still ten times larger than for the proposed method for the medium and large sample sizes. The proposed method mainly obtains large error values for the left tail of the first distribution opposed to KDE that mainly obtains large error values for the mode of the first distribution. Trentin et al. (2018) obtains large errors for the left tail as well as errors for the dip in between the distributions for the medium and large samples.

Method	Bandwidth	Sample Size		
		small	medium	large
proposed		0.571	0.566	0.740
		(0.033)	(0.022)	(0.023)
Trentin et al. (2018)	$5.0/\sqrt{T-1}$	1.362	3.000	62.474
		(0.155)	(0.194)	(6.041)
Trentin et al. (2018)	$7.5/\sqrt{T-1}$	1.020	2.584	12.342
		(0.081)	(0.133)	(0.351)
Trentin et al. (2018)	$10.0/\sqrt{T-1}$	1.055	3.525	12.616
		(0.081)	(0.258)	(0.557)
KDE	Scott	11.226	53.915	121.330
		(0.135)	(0.266)	(0.603)
		Values in the table scaled by 10^{-3}		

Table 5.8 test sample results of mixed normal data for the proposed, Trentin et al. (2018) and KDE

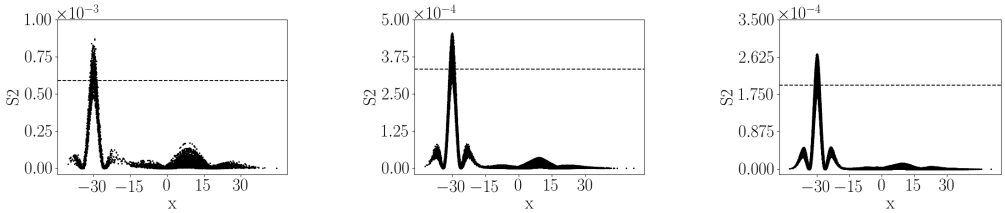
The mean (standard error) of the calculated $L2_{PDF}$ loss obtained by model 1 and 4 for the proposed method, model 5 for Trentin et al. (2018)'s method and the KDE of the test sample of the small, medium, and large sample sizes. Only the optimal bandwidth for KDE is shown.



(a) small sample size (b) medium sample size (c) large sample size

Figure 5.25 PDF estimates by KDE

True and estimated PDF using KDE for mixed normal data. The average estimates in terms of $L2_{PDF}$ loss values out of 100 simulation replications for all sample sizes are given.



(a) small sample size (b) medium sample size (c) large sample size

Figure 5.26 Squared errors obtained by the KDE

Squared errors calculated using the $S2_{PDF}$ differences together with the 95^{th} percentile for the small, medium and large sample size, using KDE method with Scott's rule.

5.2 Generalized Extreme Value Distribution

In this section we consider simulated data from the following three-component mixture of generalized extreme value (GEV) distributions:

$$p(x) = 0.35GEV(2.0, 0.6, 0) + 0.5GEV(5.0, 0.7, 0) + 0.15GEV(7.0, 0.5, 0)$$

where $GEV(\mu, \sigma, \xi)$ stands for the GEV distribution with mean μ, variance σ^2, and shape parameter ξ, as in Trentin et al. (2018) for different sample sizes [14]. The results are shown for the small sample size and the medium sample size with batch sizes as large as the test samples, using 10,000 training epochs and a learning rate of 0.01. The logistic function is used as activation function in all layers except for the output layer. In Table 5.9 the different models that are applied to the mixed GEV distribution are summarized.

model	1	2	3	4	5	6	7	8	9
# hidden layers	1	1	1	2	2	2	3	3	3
# hidden neurons	5	9	15	5	9	15	5	9	15

Table 5.9 Overview of models considered for the mixed GEV distribution

Small sample results:

Table 5.10 presents the small sample results for the validation sample and test sample. Models 4, 5, 6, 7, 8, and 9 are the best performing models obtaining the smallest $L2_{\widetilde{CDF}}$ losses for the validation sample and the test sample. These models also attain the smallest $L2_{PDF}$ losses for both samples. Because model 9 has the smallest $L2_{\widetilde{CDF}}$ feasible loss for the validation sample, this model is further analyzed.

Figures 5.27 and 5.28 present the best, average, and worst CDF and PDF estimates for the small test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 9. There are three regions, which are the three modes of the distribution, where the CDF estimates deviate from the target CDF and the true CDF. The best estimate performs relatively well compared to the average and worst estimates, which have particular difficulty around the second and third mode of the distribution. The corresponding PDF estimates show that these differences do not result in tail estimation error. However, too much probability mass is allocated around the first mode or third mode, while the second mode shows local bi-modal behavior, see Figure 5.28. This bi-modal behavior in the modes is not expected when analyzing the CDF estimates in Figure 5.27. The target CDFs that approximate the true CDFs depicted in this Figure are inaccurate as this distribution case is highly non-linear. This is probably due to the small number of observations that are used to construct this target CDF. An alternative explanation would be that model 9 is too large to fit this distribution case. Using smaller models will prevent this bi-modal behavior around the modes due to the extensive use of hidden layers or neurons as corresponding CDF estimates do not show this volatile behavior. See Figure 5.29 for the PDF estimates of smaller model 7, using 5 neurons instead of 15 neurons. These estimates fit the true PDF closely with slight differences in the modes. Model 7 closely follows model 9 in terms of $L2_{\widetilde{CDF}}$ loss. Moreover, corresponding $L2_{PDF}$ losses for both samples are substantially smaller for model 7 than for model 9. We choose the optimal model based on the smallest $L2_{\widetilde{CDF}}$ loss as this is the only feasible loss in real data applications. However, any model that uses more than 1 hidden layer would be appropriate, e.g.

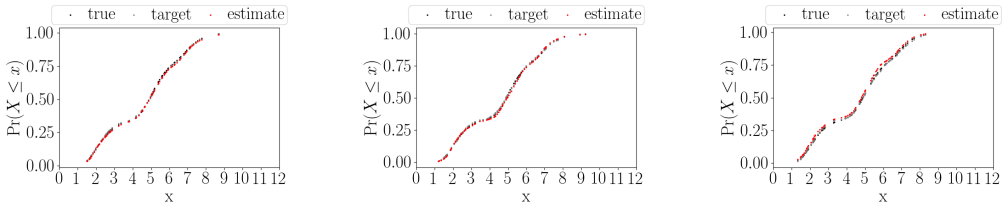
[14] The largest sample size is not performed due to the lack of available computational power.

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample with $L2_{\widehat{CDF}}$= 7.400 (0.474)										
$L2_{\widetilde{CDF}}$	1.029	8.381	8.897	8.616	2.078	2.171	1.662	1.301	1.414	1.249
	(0.020)	(0.295)	(0.295)	(0.326)	(0.080)	(0.137)	(0.077)	(0.047)	(0.051)	(0.044)
$L2_{CDF}$	7.164	12.289	12.075	11.918	7.602	7.812	7.542	7.068	7.466	7.591
	(0.503)	(0.502)	(0.449)	(0.469)	(0.528)	(0.512)	(0.535)	(0.487)	(0.549)	(0.543)
$L2_{PDF}$	37.852	62.533	69.024	73.662	28.612	30.674	29.530	28.034	33.834	44.394
	(1.970)	(1.569)	(1.071)	(5.830)	(1.227)	(1.452)	(1.417)	(1.364)	(1.716)	(1.930)
Test sample with $L2_{\widehat{CDF}}$= 5.873 (0.283)										
$L2_{\widetilde{CDF}}$	2.284	10.121	10.148	9.680	4.264	4.841	3.995	3.532	3.503	3.812
	(0.315)	(0.449)	(0.395)	(0.517)	(0.268)	(0.347)	(0.263)	(0.314)	(0.299)	(0.381)
$L2_{CDF}$	7.460	14.503	14.199	15.078	8.379	10.291	9.294	8.701	9.233	10.109
	(0.819)	(0.922)	(0.776)	(1.103)	(0.650)	(0.871)	(0.686)	(0.789)	(0.781)	(0.925)
$L2_{PDF}$	40.492	104.742	67.069	123.180	25.411	28.137	28.511	29.913	36.979	50.917
	(2.414)	(42.482)	(4.363)	(62.996)	(1.093)	(1.266)	(1.469)	(1.661)	(2.201)	(2.690)
Values in the table scaled by 10^{-3}										

Table 5.10 small sample results of mixed GEV data

L2 losses for mixed GEV data, small sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

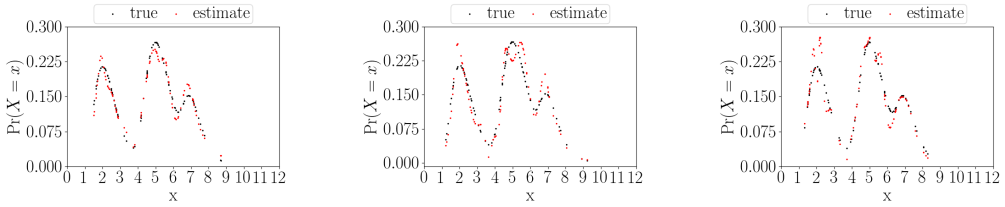
model 4 could also be chosen as optimal based on parsimony additionally to a relatively small $L2_{\widetilde{CDF}}$ loss. See appendix C for results using the hyperbolic tangent function instead of the logistic function and appendix D for corresponding results using model 4 with 2 hidden layers instead of model 7 with 3 hidden layers. These results show that model 4 closely estimates the CDF and PDF estimates of this simulation distribution for both samples. Similar results, in terms of estimates and S2 errors, using model 7 are given irrespective of which activation function is used.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.27 small sample CDF estimates

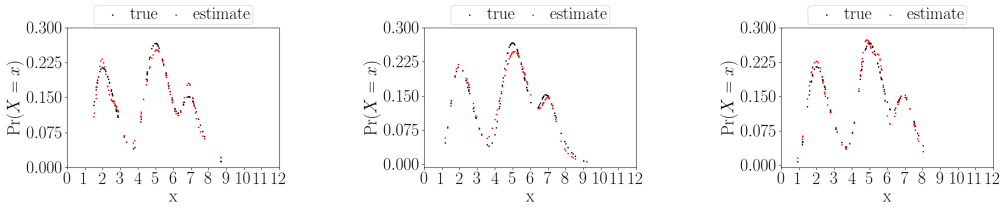
True, target, and estimated CDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 15 neurons are given.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.28 small sample PDF estimates

True and estimated PDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 15 neurons are given.

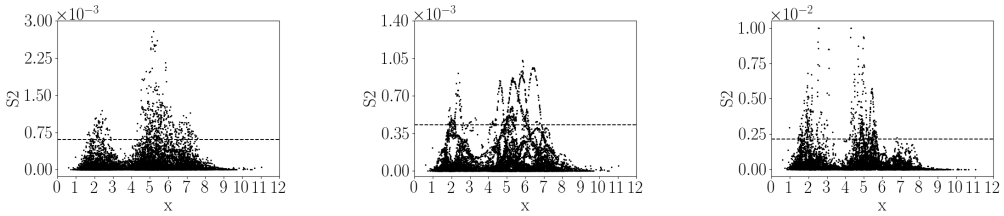


(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.29 small sample PDF estimates using model 7

True and estimated PDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 5 neurons are given.

The squared errors $S2_{\widetilde{CDF}}$ $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 9 are provided in Figure 5.30. The $S2_{CDF}$ and $S2_{\widetilde{CDF}}$ errors show misfits of all three modes. Tails of the distribution seem to be estimated well in all cases. The corresponding $S2_{PDF}$ squared errors show misfits of only the first and second modes. This is also in line with the PDF estimates in Figure 5.28.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

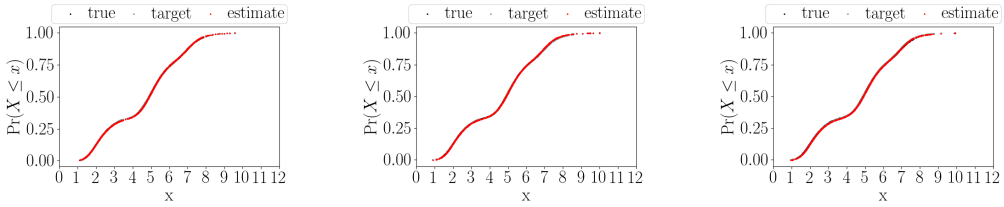
Figure 5.30 small sample squared errors

Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the small sample size, using 3 hidden layers and 15 neurons of the proposed method.

Medium sample results:

Table 5.11 presents the medium sample results for the validation sample and test sample. Models 4, 5, 6, 7, 8, and 9 obtain the smallest $L2_{\widetilde{CDF}}$ loss in the validation sample. In the corresponding $L2_{\widetilde{CDF}}$ losses for the test sample, model 4 and 7 perform best. Model 7 is further analyzed since this model has the smallest $L2_{\widetilde{CDF}}$ feasible loss for the validation sample. Models 1, 2, and 3 have substantial larger $L2_{\widetilde{CDF}}$ and $L2_{PDF}$ losses for both the validation and test samples. This is in line with the small sample results. In this simulation application, models with only 1 hidden layer perform substantially worse.

Figures 5.31 and 5.32 present the best, average, and worst CDF and PDF estimates for the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 7. The best, average and worst CDF estimates follow the shape of the true CDF closely. The corresponding best, average and worst PDF estimates reflect this by following the tails and modes of the distribution closely.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.31 medium sample CDF estimates

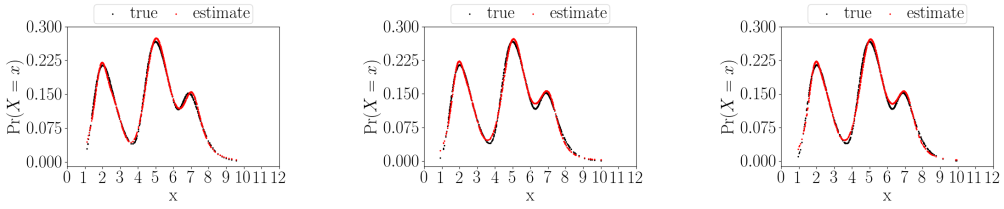
True, target, and estimated CDF using the proposed method for mixed GEV data. The best, median, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 5 neurons are given.

The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications for model 7 are provided in Figure 5.33. The $S2_{\widetilde{CDF}}$ and $S2_{CDF}$ depicted in

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample with $L2_{\widehat{CDF}}$= 7.996 (0.513)										
$L2_{\widetilde{CDF}}$	1.770	52.780	34.822	29.891	2.742	3.538	2.881	2.422	3.082	2.725
	(0.058)	(0.738)	(0.740)	(0.478)	(0.110)	(0.196)	(0.154)	(0.105)	(0.174)	(0.121)
$L2_{CDF}$	7.934	58.145	40.988	36.024	8.732	9.887	9.165	8.348	9.667	9.261
	(0.497)	(0.990)	(0.969)	(0.759)	(0.528)	(0.561)	(0.616)	(0.477)	(0.596)	(0.538)
$L2_{PDF}$	32.292	418.193	262.517	278.351	30.277	29.466	28.911	33.836	30.529	37.158
	(1.459)	(4.482)	(6.755)	(3.968)	(1.245)	(1.341)	(1.347)	(1.308)	(1.513)	(1.631)
Test sample with $L2_{\widehat{CDF}}$= 6.484 (0.434)										
$L2_{\widetilde{CDF}}$	2.172	47.409	33.082	55.427	5.424	9.508	22.976	5.517	12.319	23.582
	(2.311)	(0.904)	(0.831)	(0.841)	(0.421)	(1.248)	(3.479)	(0.449)	(1.873)	(4.997)
$L2_{CDF}$	6.927	50.166	37.752	61.964	8.662	12.872	27.766	9.371	15.284	27.832
	(2.668)	(0.959)	(1.201)	(0.905)	(0.524)	(1.250)	(3.728)	(0.556)	(1.791)	(4.827)
$L2_{PDF}$	41.457	364.086	302.218	488.457	34.365	32.486	44.044	41.793	32.216	48.643
	(2.120)	(3.024)	(9.323)	(4.413)	(1.353)	(1.527)	(2.171)	(1.594)	(1.835)	(2.115)
Values in the table scaled by 10^{-3}										

Table 5.11 medium sample results of mixed GEV data

L2 losses for mixed GEV data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

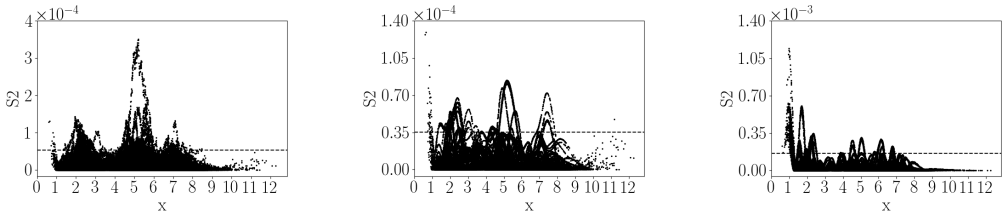


(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.32 medium sample PDF estimates

True and estimated PDF using the proposed method for mixed GEV data. The best, median, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 5 neurons are given.

Figures 5.33(a) and 5.33(b) show that estimating the modes is more difficult than estimating the tails with large values for the second mode. These results are also in line with the small sample results. The $S2_{PDF}$ errors illustrated by Figure 5.33(c) reflect this with larger error values at the left tail. This is also in line with the small sample $S2_{PDF}$ errors. Note that the squared errors for the small sample size are ten times larger than the squared errors for the medium sample indicating that the increase in sample size improves approximation properties.



(a) $S2_{\widehat{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure 5.33 medium sample squared errors

Squared errors calculated using the $S2_{\widehat{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the medium sample size, using 3 hidden layers and 5 neurons of the proposed method.

Hence the CDF and PDF estimates differ across sample sizes partly due to model selection. Different models are used across sample sizes. For the small sample size, 3 hidden layers and 15 neurons are used while for the medium sample size, 3 hidden layers and 5 neurons are used. Both models use 3 hidden layers with less number of nodes for the medium sample size. Models with only 1 hidden layer perform substantially worse.

Now these results are compared with Trentin et al. (2018)'s method and KDE. Trentin et al. (2018)'s results are applied to the small and medium sample sizes, using 10,000 training epochs and a learning rate of 0.01. The batch size in these applications is set as large as the test size in order to estimate consistent target PDF values due to the bandwidth size. This initial bandwidth size is determined empirically, as in Trentin et al. (2018). Three different initial bandwidth sizes are used, $0.75/\sqrt{T-1}$, $1.0/\sqrt{T-1}$, and $2.5/\sqrt{T-1}$. The logistic output function is used as activation function in all layers except for the output layer. Note that Trentin et al. (2018) uses the initial bandwidth $1.0/\sqrt{T-1}$. This result is improved by using a different bandwidth and model.

Small sample results for Trentin et al. (2018):

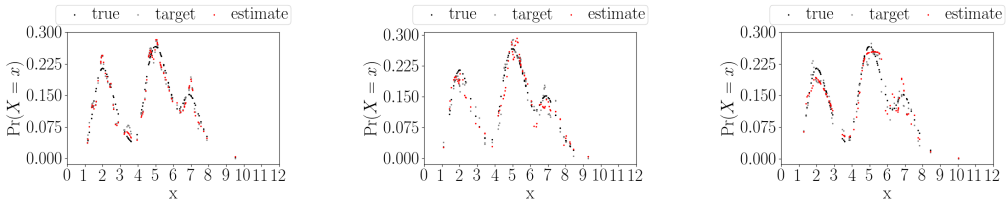
Table 5.12 presents the small sample results for the validation sample and test sample using Trentin et al. (2018). Model 9 is the best performing model obtaining the smallest $L2_{\widetilde{PDF}}$ loss irrespective of which initial bandwidth is used for both samples. Initial bandwidth $2.5/\sqrt{T-1}$ obtains the smallest $L2_{\widetilde{PDF}}$ and $L2_{PDF}$ loss and initial bandwidth $0.75/\sqrt{T-1}$ obtains the largest $L2_{\widetilde{PDF}}$ and $L2_{PDF}$ loss for both samples. Note that Trentin et al. (2018) only reports results of model 2, with 1 hidden layer and 5 neurons, with an initial bandwidth of $1.0/\sqrt{T-1}$. The method imposed by Trentin et al. (2018) using model 2 is improved by using models with 2 hidden layers or more with a similar or larger bandwidth for both samples. For the validation sample using models with 2 hidden layers or more, except for model 9, with the smallest bandwidth also improves this result. The results which use bandwidths $1.0/\sqrt{T-1}$ and $2.5/\sqrt{T-1}$ outperform the proposed method using model 9 shown in Table 5.10. The proposed method using model 9 outperforms Trentin et al. (2018) using bandwidth $0.75/\sqrt{T-1}$ and using model 7 outperforms Trentin et al. (2018) using bandwidths $0.75/\sqrt{T-1}$ and $1.0/\sqrt{T-1}$, shown in Table 5.10. Hence Trentin et al. (2018) outperforms the proposed method depending on which bandwidth is used.

Figures 5.34, 5.35, and 5.36 present the best, average and worst PDF estimates in terms of $L2_{\widetilde{PDF}}$ loss using initial bandwidths $0.75/\sqrt{T-1}$, $1.0/\sqrt{T-1}$, and $2.5/\sqrt{T-1}$ for the test sample using model 9. The PDF estimates show no tail estimation error. However, the modes are misfit just as for the proposed method. This extent of volatility is due to the inaccurate approximation of the target PDFs using only a small number of observations. The PDF estimates, especially the best estimates in all Figures, follow the target PDFs very closely. Hence both methods show volatile estimates due to the small sample size for which the target CDFs or PDFs are approximated inaccurately. Target PDFs using a bandwidth of $2.5/\sqrt{T-1}$ contain less variability than smaller bandwidths. Therefore this bandwidth outperforms the others. However, these target PDFs contain a lower, non-symmetrical second mode which is reflected in the best, average and worst PDF estimates in Figure 5.36. Furthermore, for the average PDF estimate in Figure 5.36(b) the third mode together with the dip before is not well represented by the target PDF. Hence there is a trade-off between the specification of bandwidths. A too small bandwidth results in more variability of the target PDF though these values are comparable to the true PDF. A too large bandwidth results in a smooth non-symmetrical target PDF with values that are too low compared to the true PDF. This is also shown in the large sample results of the univariate mixed normal distribution in section 5.1.

The squared errors $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ for each data point over 100 simulation replications using model 9 are provided in Figures 5.37 and 5.38, respectively. The $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ errors show misfits of all modes. Tails of the distribution seem to be estimated well in all cases. These results are similar as for the proposed method. The differences in the $S2_{PDF}$ errors show that the modes are correctly estimated with error values of at most $2.1 \cdot 10^{-3}$ in 95% of the cases for the proposed method and for Trentin et al. (2018)'s method using an initial bandwidth of $0.75/\sqrt{T-1}$. For larger bandwidths these values are smaller, $1.5 \cdot 10^{-3}$ and $0.9 \cdot 10^{-3}$ in 95% of the cases for Trentin et al. (2018).

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample										
bandwidth $0.75/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 52.672 (1.463)										
$L2_{\widetilde{PDF}}$	4.042	64.501	53.488	66.943	27.309	23.364	18.558	20.647	11.728	4.042
	(0.137)	(1.642)	(1.488)	(1.780)	(0.972)	(0.738)	(0.583)	(0.631)	(0.352)	(0.137)
$L2_{PDF}$	49.848	46.368	37.643	47.151	37.003	34.715	38.765	38.328	43.569	49.848
	(1.489)	(1.170)	(1.121)	(1.377)	(1.536)	(1.436)	(1.474)	(1.458)	(1.500)	(1.489)
bandwidth $1.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 38.016 (1.213)										
$L2_{\widetilde{PDF}}$	1.574	50.671	50.470	52.242	15.234	14.289	11.849	9.464	3.249	1.585
	(0.067)	(1.444)	(1.709)	(1.575)	(0.494)	(0.894)	(0.469)	(0.377)	(0.134)	(0.080)
$L2_{PDF}$	36.937	45.682	45.801	46.574	29.018	31.665	30.998	32.606	35.624	36.922
	(1.218)	(1.148)	(1.403)	(1.017)	(1.244)	(1.419)	(1.210)	(1.185)	(1.217)	(1.218)
bandwidth $2.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 21.574 (0.867)										
$L2_{\widetilde{PDF}}$	0.199	18.166	13.746	18.242	1.165	1.708	0.773	1.658	0.761	0.206
	(0.000)	(0.765)	(0.653)	(0.965)	(0.131)	(0.169)	(0.048)	(0.131)	(0.052)	(0.011)
$L2_{PDF}$	21.764	42.288	37.379	40.985	22.360	22.958	22.010	23.057	22.005	21.779
	(0.869)	(0.989)	(1.222)	(1.132)	(0.894)	(0.960)	(0.872)	(0.872)	(0.851)	(0.867)
Test sample										
bandwidth $0.75/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 49.219 (1.748)										
$L2_{\widetilde{PDF}}$	10.374	65.433	57.809	65.579	28.614	23.663	22.626	20.654	15.243	10.374
	(0.369)	(1.345)	(1.496)	(1.746)	(0.834)	(0.754)	(0.746)	(0.656)	(0.531)	(0.369)
$L2_{PDF}$	56.871	43.940	35.901	39.868	38.688	37.410	42.597	43.872	51.032	56.871
	(2.140)	(1.245)	(1.148)	(1.269)	(1.673)	(1.556)	(1.942)	(1.895)	(2.030)	(2.140)
bandwidth $1.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 35.474 (1.373)										
$L2_{\widetilde{PDF}}$	5.780	49.423	39.591	70.577	14.679	14.663	10.663	12.016	7.853	5.763
	(0.289)	(1.599)	(1.310)	(13.653)	(0.539)	(0.668)	(0.430)	(0.501)	(0.337)	(0.290)
$L2_{PDF}$	37.740	46.226	37.909	51.823	30.096	29.142	31.216	31.357	35.096	37.746
	(1.206)	(1.290)	(3.210)	(9.292)	(1.140)	(1.203)	(1.173)	(1.152)	(1.190)	(1.202)
bandwidth $2.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 20.299 (0.834)										
$L2_{\widetilde{PDF}}$	1.642	16.795	13.898	18.077	2.532	3.258	1.991	3.107	1.911	1.561
	(0.087)	(0.633)	(0.734)	(0.669)	(0.135)	(0.195)	(0.086)	(0.227)	(0.096)	(0.078)
$L2_{PDF}$	25.609	43.904	39.164	42.802	25.991	26.986	25.787	26.790	25.565	25.573
	(1.122)	(1.366)	(1.378)	(1.265)	(1.119)	(1.180)	(1.095)	(1.160)	(1.112)	(1.121)
Values in the table scaled by 10^{-3}										

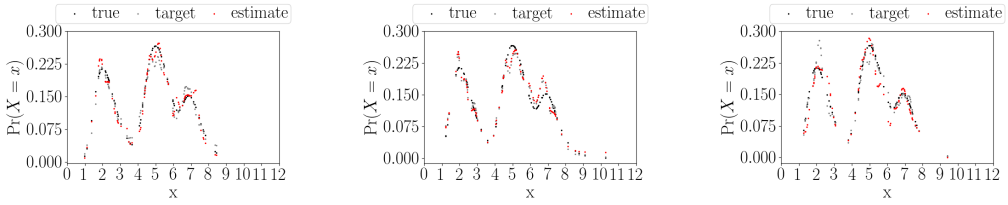
Table 5.12 small sample results of mixed GEV data by Trentin et al. (2018)
L2 losses for mixed GEV data, small sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated PDF obtained by Trentin et al. (2018) for the validation and test samples.



(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.34 small sample PDF estimates by Trentin et al. (2018) with initial bandwidth $0.75/\sqrt{T-1}$

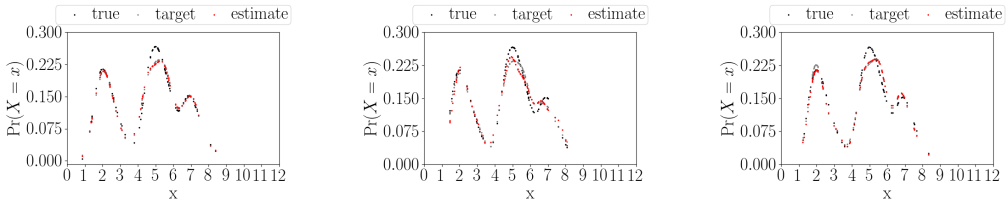
True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $0.75/\sqrt{T-1}$ for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widehat{PDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 15 neurons are given.



(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.35 small sample PDF estimates by Trentin et al. (2018) with initial bandwidth $1.0/\sqrt{T-1}$

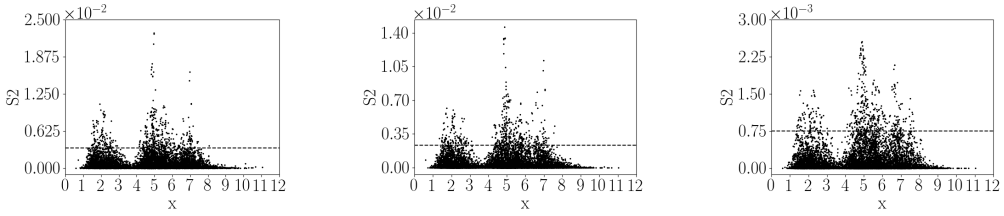
True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $1.0/\sqrt{T-1}$ for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widehat{PDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 15 neurons are given.



(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.36 small sample PDF estimates by Trentin et al. (2018)'s method with initial bandwidth $2.5/\sqrt{T-1}$

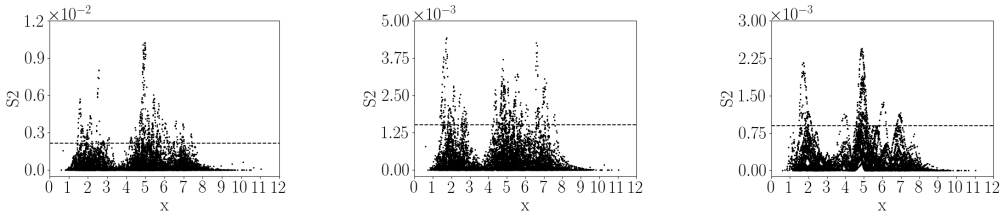
True, target, and estimated PDF by Trentin et al. (2018) using an initial bandwidth of $2.5/\sqrt{T-1}$ for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widehat{PDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 15 neurons are given.



(a) $0.75/\sqrt{T-1}$ (b) $1.0/\sqrt{T-1}$ (c) $2.5/\sqrt{T-1}$

Figure 5.37 small sample squared errors based on $S2_{\widehat{PDF}}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{\widehat{PDF}}$ differences together with the 95^{th} percentile for the small sample size, using 3 hidden layers and 15 neurons of Trentin et al. (2018)'s method.



(a) $0.75/\sqrt{T-1}$ (b) $1.0/\sqrt{T-1}$ (c) $2.5/\sqrt{T-1}$

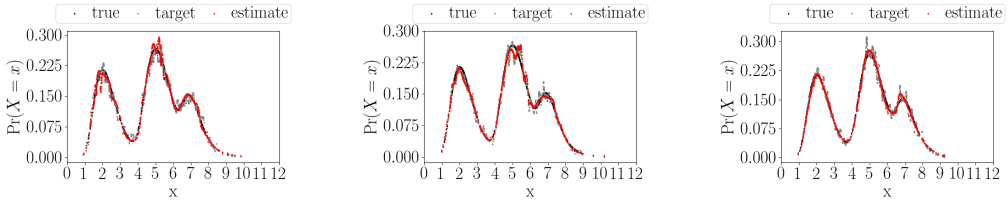
Figure 5.38 small sample squared errors based on $S2_{PDF}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{PDF}$ differences together with the 95^{th} percentile for the small sample size, using 3 hidden layers and 15 neurons of Trentin et al. (2018)'s method.

Medium sample results for Trentin et al. (2018):

Table 5.13 presents the medium sample results for the validation sample and test sample using Trentin et al. (2018)'s method. Model 9 is the best performing model obtaining the smallest $L2_{\widetilde{PDF}}$ loss irrespective of which initial bandwidth is used for both samples. Initial bandwidth $2.5/\sqrt{T-1}$ obtains the smallest $L2_{\widetilde{PDF}}$ and $L2_{PDF}$ loss and initial bandwidth $0.75/\sqrt{T-1}$ obtains the largest $L2_{\widetilde{PDF}}$ and $L2_{PDF}$ loss for both samples. Trentin et al. (2018) reports results using model 2 and using initial bandwidth $1.0/\sqrt{T-1}$. The method imposed by Trentin et al. (2018) using model 2 is improved by using models with 2 hidden layers or more irrespective of which bandwidth is used for both samples. The proposed method with model 7 outperform results estimated by optimal model 9 of Trentin et al. (2018). Morever, the results obtained by Trentin et al. (2018) are again very dependent of the specification of the initial bandwidth. Especially, the difference between $1.0/\sqrt{T-1}$ and $2.5/\sqrt{T-1}$ is quite substantial. The specification of this bandwidth presents a trade-off between variability and bias. A relatively larger bandwidth improves the $L2_{\widetilde{PDF}}$ loss but introduces a bias in the target PDF e.g. by introducing skewed modes. A relatively smaller bandwidth introduces variability in the target PDF which increases the $L2_{\widetilde{PDF}}$ loss but does not include any bias.

Figures 5.39, 5.40, and 5.41 present the best, average, and worst PDF estimates in terms of $L2_{\widetilde{PDF}}$ loss using initial bandwidths $0.75/\sqrt{T-1}$, $1.0/\sqrt{T-1}$, and $2.5/\sqrt{T-1}$ for the test sample using model 9. The PDF estimates show no tail estimation error similarly for the small sample. However, the modes are misfit in some cases. The PDF estimates with bandwidth $0.75/\sqrt{T-1}$ have again bi-modal peaks. The corresponding target PDF is more erratic than for the larger bandwidths. The PDF estimates with $1.0/\sqrt{T-1}$ and $2.5/\sqrt{T-1}$ bandwidths do not have smooth lines as the true PDF has, though not as erratic as for the smallest bandwidth.



(a) smallest $L2_{\widetilde{PDF}}$ loss (b) average $L2_{\widetilde{PDF}}$ loss (c) largest $L2_{\widetilde{PDF}}$ loss

Figure 5.39 medium sample PDF estimates by Trentin et al. (2018) with initial bandwidth $0.75/\sqrt{T-1}$
True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $0.75/\sqrt{T-1}$ for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{PDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 15 neurons are given.

The squared errors $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ for each data point over 100 simulation replications using model 9 are provided in Figures 5.42 and 5.43, respectively. The $S2_{\widetilde{PDF}}$ and $S2_{PDF}$ errors show misfits of the first and second modes and to a lower extent for the third mode opposed to all modes for the small sample size. Tails of the distribution seem to be estimated well in all cases similarly as for the small sample size. The error values with bandwidth $2.5/\sqrt{T-1}$ are substantially smaller. Overall, Trentin et al. (2018)'s method has large errors for the first and second modes with largest values around $3.0 \cdot 10^{-3}$, $2.5 \cdot 10^{-3}$ and $0.9 \cdot 10^{-3}$ for each bandwidth compared to $0.6 \cdot 10^{-3}$ for corresponding modes of the proposed method. The proposed method has large errors for the left tail

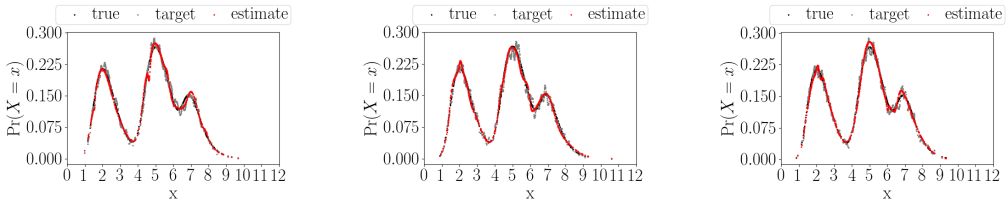
model	optimal	1	2	3	4	5	6	7	8	9
Validation sample										
bandwidth $0.75/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 189.108 (3.048)										
$L2_{\widetilde{PDF}}$	113.826	372.035	281.691	388.030	167.332	176.102	173.270	163.712	162.356	113.826
	(2.019)	(11.244)	(6.633)	(5.914)	(3.237)	(2.623)	(2.914)	(2.754)	(2.628)	(2.019)
$L2_{PDF}$	85.053	226.603	138.365	240.732	51.721	46.816	57.625	57.464	48.318	85.053
	(2.882)	(10.048)	(5.454)	(3.246)	(2.822)	(2.003)	(2.181)	(2.319)	(1.810)	(2.882)
bandwidth $1.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 139.480 (2.594)										
$L2_{\widetilde{PDF}}$	64.123	327.362	185.481	264.191	112.121	125.210	100.170	108.391	105.665	64.123
	(1.456)	(10.948)	(3.339)	(5.534)	(2.284)	(2.026)	(1.900)	(2.022)	(2.142)	(1.456)
$L2_{PDF}$	79.355	231.455	89.826	168.372	46.263	46.273	48.366	51.681	44.511	79.355
	(2.647)	(9.608)	(2.880)	(3.892)	(2.463)	(1.690)	(2.042)	(2.133)	(1.821)	(2.647)
bandwidth $2.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 51.419 (1.629)										
$L2_{\widetilde{PDF}}$	8.450	205.957	91.144	165.064	25.072	37.625	19.441	27.045	25.504	8.450
	(0.291)	(10.926)	(2.479)	(4.356)	(0.880)	(1.078)	(0.571)	(0.846)	(0.930)	(0.291)
$L2_{PDF}$	45.012	207.079	87.619	164.444	36.858	41.578	37.240	40.423	36.360	45.012
	(1.677)	(9.995)	(2.703)	(3.890)	(1.668)	(1.680)	(1.511)	(1.626)	(1.550)	(1.677)
Values in the table scaled by 10^{-3}										

Table 5.13 medium sample results of mixed GEV data by Trentin et al. (2018)

L2 losses for mixed GEV data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated PDF obtained by Trentin et al. (2018) for the validation and test samples. (Continued on next page)

model	optimal	1	2	3	4	5	6	7	8	9
					Test sample					
					bandwidth $0.75/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 170.214 (2.955)					
$L2_{\widetilde{PDF}}$	105.075	369.855	242.848	353.764	153.200	165.457	158.315	151.763	146.334	105.075
	(2.421)	(12.031)	(4.701)	(7.385)	(3.009)	(2.738)	(2.546)	(2.683)	(2.517)	(2.421)
$L2_{PDF}$	93.790	235.954	118.498	217.898	48.615	47.424	50.895	59.123	54.682	93.790
	(3.369)	(10.821)	(6.003)	(6.697)	(2.655)	(2.966)	(2.316)	(3.429)	(2.407)	(3.369)
					bandwidth $1.0/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 125.366 (2.465)					
$L2_{\widetilde{PDF}}$	66.942	322.333	185.613	304.141	106.551	114.273	119.268	106.521	106.742	66.942
	(1.795)	(11.891)	(3.763)	(5.767)	(2.356)	(1.719)	(3.259)	(2.239)	(2.092)	(1.795)
$L2_{PDF}$	83.174	232.296	105.245	214.839	49.318	43.025	57.933	58.246	51.969	83.174
	(2.822)	(10.604)	(4.580)	(5.280)	(2.625)	(2.405)	(4.230)	(3.272)	(1.923)	(2.822)
					bandwidth $2.5/\sqrt{T-1}$ with $L2_{\widehat{PDF}}$= 45.588 (1.416)					
$L2_{\widetilde{PDF}}$	10.831	206.880	75.921	154.089	22.201	34.811	19.849	26.190	21.673	10.831
	(0.314)	(11.793)	(1.780)	(4.686)	(0.669)	(1.172)	(0.550)	(1.080)	(0.760)	(0.314)
$L2_{PDF}$	52.134	205.886	76.141	147.555	40.476	41.289	45.745	41.904	40.225	52.134
	(1.792)	(10.497)	(3.112)	(5.817)	(1.867)	(1.917)	(1.992)	(1.995)	(1.668)	(1.792)
										Values in the table scaled by 10^{-3}

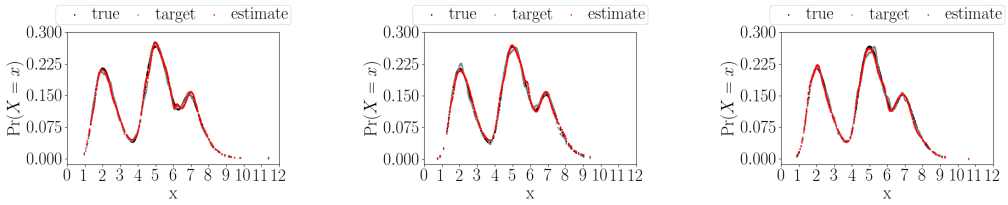
Table 5.13 medium sample results of mixed GEV data by Trentin et al. (2018) (Cont.)
L2 losses for mixed GEV data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated PDF obtained by Trentin et al. (2018) for the validation and test samples.



(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.40 medium sample PDF estimates by Trentin et al. (2018) with initial bandwidth $1.0/\sqrt{T-1}$

True, target, and estimated PDF using Trentin et al. (2018)'s method using an initial bandwidth of $1.0/\sqrt{T-1}$ for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widehat{PDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 15 neurons are given.

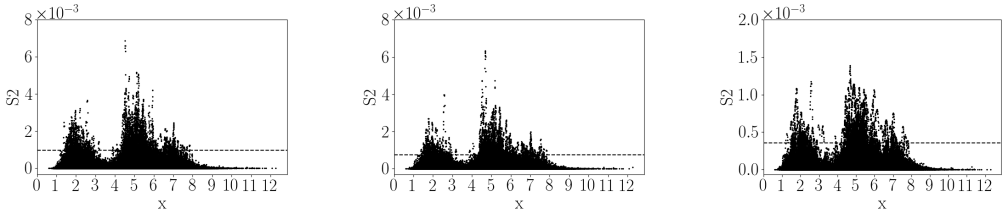


(a) smallest $L2_{\widehat{PDF}}$ loss (b) average $L2_{\widehat{PDF}}$ loss (c) largest $L2_{\widehat{PDF}}$ loss

Figure 5.41 medium sample PDF estimates by Trentin et al. (2018)'s method with initial bandwidth $2.5/\sqrt{T-1}$

True, target, and estimated PDF by Trentin et al. (2018) using an initial bandwidth of $2.5/\sqrt{T-1}$ for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widehat{PDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 15 neurons are given.

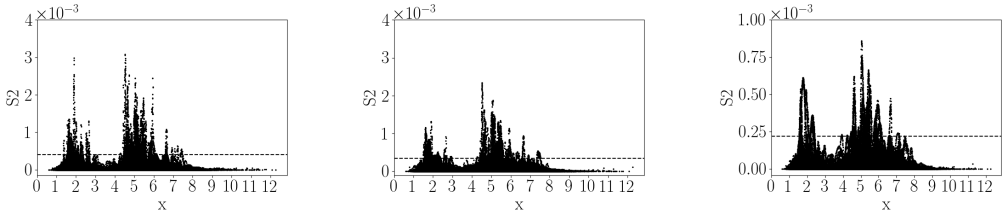
of the distribution around $1.2 \cdot 10^{-3}$ though.



(a) $0.75/\sqrt{T-1}$ (b) $1.0/\sqrt{T-1}$ (c) $2.5/\sqrt{T-1}$

Figure 5.42 medium sample squared errors based on $S2_{\widehat{PDF}}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{\widehat{PDF}}$ differences together with the 95^{th} percentile for the medium sample size, using 3 hidden layers and 15 neurons of Trentin et al. (2018)'s method.



(a) $0.75/\sqrt{T-1}$ (b) $1.0/\sqrt{T-1}$ (c) $2.5/\sqrt{T-1}$

Figure 5.43 medium sample squared errors based on $S2_{PDF}$ obtained by Trentin et al. (2018)

Squared errors calculated using the $S2_{PDF}$ differences together with the 95^{th} percentile for the medium sample size, using 3 hidden layers and 15 neurons of Trentin et al. (2018)'s method.

We next compare the results of the proposed method with KDE for all sample sizes.

Results for KDE:

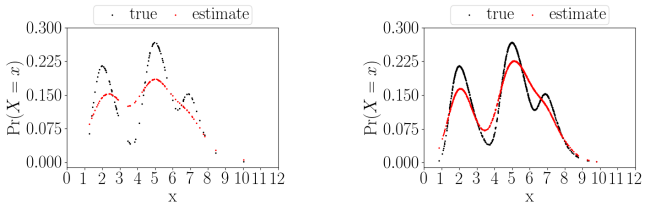
For the ease of comparison, we present the test sample results of the proposed method, Trentin et al. (2018)'s method with different bandwidths and the KDE for all sample sizes in Table 5.14. The KDE has substantial larger $L2_{PDF}$ loss values compared to the proposed and Trentin et al. (2018) methods. This discrepancy enlarges as sample size grows since the loss functions in Section 4.2 are the total losses from all observations. Trentin et al. (2018)'s method shows this enlarging effect as well, though to a lesser extent. However, for the proposed method this effect is not found. Similar $L2_{PDF}$ loss values are found across sample sizes. Actually, the $L2_{PDF}$ loss of the medium sample size is even smaller than for the small sample size. Trentin et al. (2018)'s method is not robust against different bandwidths as the $L2_{PDF}$ loss values differ across these. The $L2_{PDF}$ loss of the small sample using bandwidth $0.75/\sqrt{T-1}$ is larger than for the proposed method, though using bandwidths $1.0/\sqrt{T-1}$ and $2.5/\sqrt{T-1}$ result in smaller $L2_{PDF}$ loss values than for the proposed method. The $L2_{PDF}$ losses of the medium sample differ among each other as well. Moreover, these $L2_{PDF}$ loss values are larger than for the proposed method. Hence for the small sample size, Trentin et al. (2018) obtains the smallest $L2_{PDF}$ loss, though very dependent on the specification on the bandwidth and for the medium sample size, the proposed method outperforms the other baseline methods.

Method	Bandwidth	Sample Size	
		small	medium
proposed		50.917	41.793
		(2.690)	(1.594)
Trentin et al. (2018)	$0.75/\sqrt{T-1}$	56.871	93.790
		(2.140)	(3.369)
Trentin et al. (2018)	$1.0/\sqrt{T-1}$	37.746	83.174
		(1.202)	(2.822)
Trentin et al. (2018)	$2.5/\sqrt{T-1}$	25.573	52.134
		(1.121)	(1.792)
KDE	Scott	251.600	936.603
		(4.526)	(12.848)
		Values in the table scaled by 10^{-3}	

Table 5.14 test sample results of mixed GEV data for proposed, Trentin et al. (2018) and KDE

The mean (standard error) of the calculated $L2_{PDF}$ loss obtained by model 7 and 9 for the proposed method, model 9 for Trentin et al. (2018)'s method and the KDE of the test sample of the small and medium sample sizes. Only the optimal bandwidth for KDE is shown.

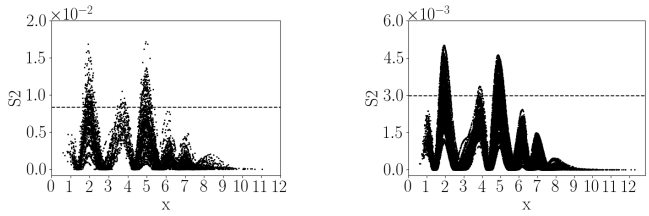
The average PDF estimates in terms of $L2_{PDF}$ loss estimated by the KDE are shown in Figure 5.44. The first and second modes are misfit as well as the dips in between. Moreover, the third mode is not captured at all. All simulation replications result in similar estimates, therefore it suffices to only show the average estimates, see Figure 5.45 for the squared errors $S2_{PDF}$ for the small, medium and large samples, respectively. Only the first and second modes are misfit as well as the dip between these modes. The third mode has smaller error values compared to the first and second modes opposed to the proposed method which has large errors for all modes. However, the proposed method and Trentin et al. (2018)'s method do not obtain large error values for the dip between the two first modes. Moreover, the $S2_{PDF}$ errors of the KDE are larger than for the other methods. The $95th$ percentile values of the KDE are approximately ten times larger than for the proposed method for both sample sizes.



(a) small sample size (b) medium sample size

Figure 5.44 PDF estimates by KDE

True and estimated PDF using KDE for mixed GEV data. The average estimates in terms of $L2_{PDF}$ loss values out of 100 simulation replications for all sample sizes are given.



(a) small sample size (b) medium sample size

Figure 5.45 Squared errors obtained by the KDE
Squared errors calculated using the $S2_{PDF}$ differences together with the 95^{th} percentile for the small and medium sample size, using KDE method using Scott's rule for specifying the bandwidth.

5.3 Mixed Poisson Distribution

5.3.1 Univariate case

In this section we consider simulated data from the following mixed Poisson distribution:

$$p(x) = 0.4Pois(2) + 0.6Pois(9),$$

where $Pois(\lambda)$ stands for the Poisson distribution with mean and variance λ. The simulated data has similar characteristics as the application of the road sensor data considered in Section 2. Figure 5.46 shows the histogram of several sensors measured on the 26^{th} of April as in Figure 2.1. The former mentioned mixed Poisson distribution represents a close fit to this application. This is the reason why this specific mixed probability mass function (PMF) is chosen.

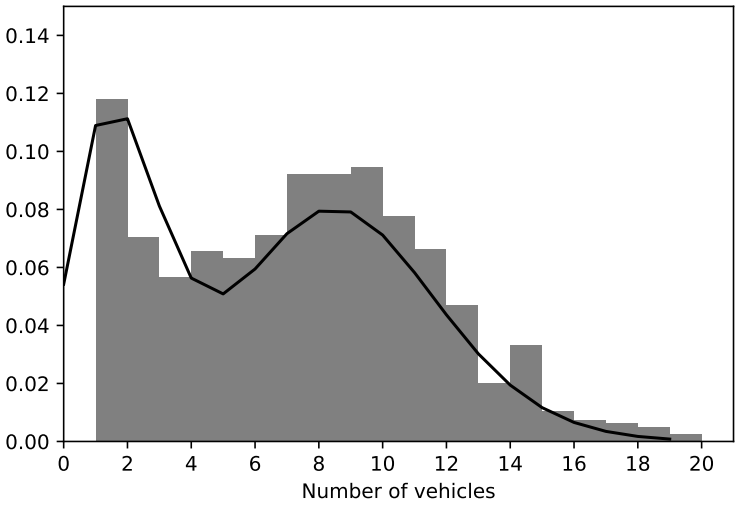


Figure 5.46 Histogram of the number of vehicle counts passing road sensor `GEO02_R_RWSTI320`, lane 2 on the 26th of April 2016 together with a mixed Poisson distribution fitted to the shape of the road sensors

The results are shown for the small sample size, the medium sample size and the large sample size, using 10,000 training epochs and a learning rate of 0.001. The hyperbolic tangent function is used as activation function in all layers except for the output layer. In Table 5.15 the different models that are applied to the mixed Poisson distribution are summarized.

model	1	2	3	4	5	6
# hidden layers	1	1	1	2	2	2
# hidden neurons	3	6	9	3	6	9

Table 5.15 Overview of models considered for the mixed Poisson distribution

Small sample results:

Table 5.16 presents the small sample results for the validation sample and test sample. Models 3, 4, and 6 are the best performing models obtaining the smallest $L2_{\widetilde{CDF}}$ loss in the validation sample. Model 6 has the smallest $L2_{\widetilde{CDF}}$ loss but does not have the smallest corresponding $L2_{PMF}$ loss. Model 4 has the smallest $L2_{PMF}$ loss. There is a large difference between these $L2_{PMF}$ losses. Models 3 and 6 both perform well based on the $L2_{\widetilde{CDF}}$ loss, with a relative small standard error. Both have a large $L2_{PMF}$ loss with a large standard error compared to model 4. This can be due to the model structure using 9 neurons or the inaccurate approximation of the target CDF. Similar conclusions can be drawn from the test sample. Model 6 is chosen as optimal based on the feasible $L2_{\widetilde{CDF}}$ loss value in the validation sample.

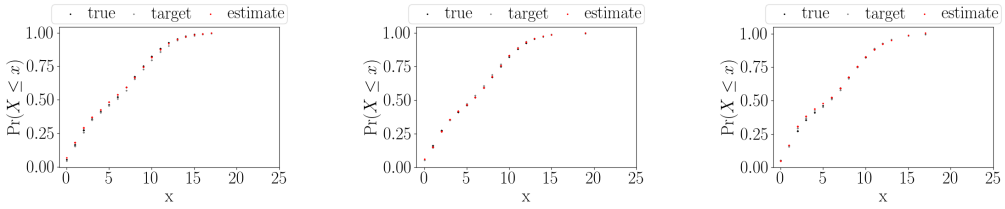
model	optimal	1	2	3	4	5	6
Validation sample with $L2_{\widehat{CDF}}$= 20.647 (1.406)							
$L2_{\widetilde{CDF}}$	1.746	12.068	3.363	2.647	2.622	2.917	1.894
	(0.101)	(1.487)	(0.191)	(0.149)	(0.142)	(0.159)	(0.112)
$L2_{CDF}$	19.422	25.142	19.032	18.914	19.344	19.575	19.608
	(1.445)	(1.818)	(1.431)	(1.485)	(1.482)	(1.487)	(1.455)
$L2_{PMF}$	142.203	41.867	34.100	158.170	29.008	53.248	153.938
	(25.738)	(1.909)	(1.796)	(39.683)	(3.109)	(4.287)	(25.589)
Test sample with $L2_{\widehat{CDF}}$= 15.366 (1.075)							
$L2_{\widetilde{CDF}}$	5.965	9.172	7.600	6.543	6.364	6.937	5.704
	(0.303)	(0.503)	0.367)	(0.327)	(0.307)	(0.329)	(0.273)
$L2_{CDF}$	20.330	20.070	19.908	19.900	19.475	20.464	20.355
	(1.572)	(1.483)	(1.455)	(1.591)	(1.560)	(1.589)	(1.571)
$L2_{PMF}$	226.253	40.255	32.196	293.473	32.159	43.477	170.451
	(42.309)	(3.539)	(1.889)	(53.984)	(4.904)	(2.593)	(29.143)
Values in the table scaled by 10^{-3}							

Table 5.16 small sample results of univariate mixed Poisson data
L2 losses for mixed Poisson data, small sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PMF obtained by the proposed method for the validation and test samples.

Figures 5.47 and 5.48 present the best, average, and worst CDF and PMF estimates for the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 6. Figure 5.47 shows that the target CDF obtains approximation errors of the true CDF. The difficulty of this distribution is the discrete nature. All CDF estimates are offset with a small value on different domains. This affects the ability of the neural network to follow the shape of the distribution. The best CDF estimate in Figure 5.47(a) is offset by a small value, especially on the domain on the left. The average CDF estimate in Figure 5.47(b) is offset by small values on a larger domain between 5 and 13. The worst estimate in Figure 5.47(c) is mainly offset on the domain between 0 and 5 with larger differences. The PMF estimates correspond to

these offsets and show some discrepancies between the estimated and true PMFs in the corresponding domains.

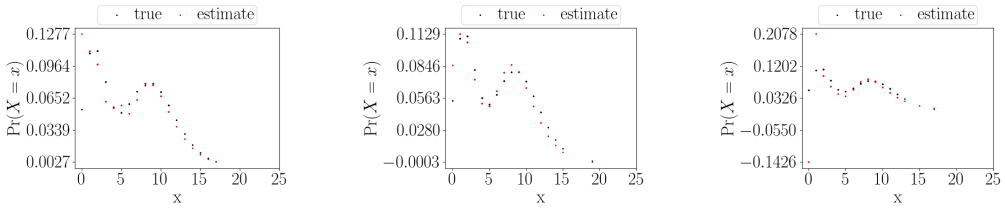
The best PMF estimate depicted in Figure 5.48(a) shows large discrepancies in the first distribution, the average PMF estimate in Figure 5.48(b) shows large discrepancies in the second distribution. The worst PMF estimate in Figure 5.48(c) shows large discrepancies in the left tail of the distribution which consists of 2 points.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.47 small sample CDF estimates

True, target, and estimated CDF using the proposed method for mixed Poisson data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 9 neurons are given.

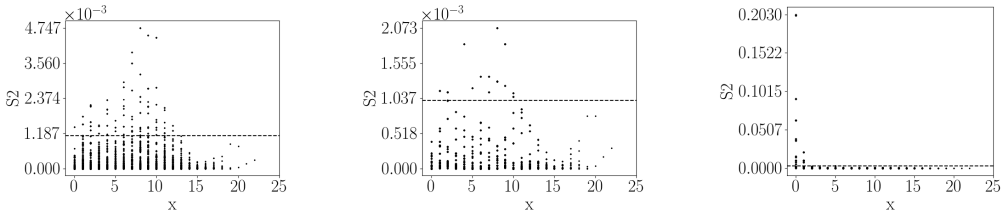


(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.48 small sample PMF estimates

True and estimated PMF using the proposed method for mixed Poisson data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 9 neurons are given.

The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PMF}$ for each data point over 100 simulation replications using model 6 are provided in Figure 5.49. The differences between Figure 5.49(a) and 5.49(b) are mainly due to the approximation errors of the target CDF. The CDF estimates misfit the first and second distributions with largest errors between domain values 5 and 10. The right tail of the distribution has very small errors below the 95^{th} percentile. Figure 5.49(c) shows large errors for the first two points of the distribution, which represents the left tail of the distribution in line with the best and worst Figures 5.48(a) and 5.48(c).



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PMF}$

Figure 5.49 small sample squared errors

Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PMF}$ differences together with the 95^{th} percentile for the small sample size, using 2 hidden layers and 9 neurons of the proposed method.

Medium sample results:

Table 5.17 presents the medium sample results for the validation sample and test sample. Models 4 and 6 are best performing models with smallest $L2_{\widetilde{CDF}}$ loss for the validation sample. Model 6 has the smallest $L2_{\widetilde{CDF}}$ loss but the largest $L2_{PMF}$ loss. Model 4 obtains the smallest $L2_{PMF}$ loss. Again, there is a large difference between these $L2_{PMF}$ losses of 428.310 and 138.875. The results for the test sample are slightly different. Model 4 is the best performing model based on both $L2_{\widetilde{CDF}}$ and $L2_{PMF}$ losses.

model	optimal	1	2	3	4	5	6
Validation sample with $L2_{\widetilde{CDF}}$= 22.090 (1.748)							
$L2_{\widetilde{CDF}}$	1.948	3.290	5.259	3.563	2.690	6.203	2.326
	(0.120)	(0.205)	(0.272)	(0.176)	(0.181)	(0.265)	(0.135)
$L2_{CDF}$	21.153	20.358	23.194	21.104	21.083	24.992	21.548
	(1.776)	(1.770)	(1.776)	(1.779)	(1.744)	(1.825)	(1.750)
$L2_{PMF}$	361.689	161.866	207.134	913.613	138.875	281.279	428.310
	(44.359)	(4.327)	(7.164)	(169.598)	(3.956)	(7.505)	(43.582)
Test sample with $L2_{\widetilde{CDF}}$= 15.359 (0.994)							
$L2_{\widetilde{CDF}}$	8.650	8.173	8.207	8.726	7.890	10.597	9.069
	(0.652)	(0.554)	(0.535)	(0.535)	(0.531)	(0.538)	(0.750)
$L2_{CDF}$	27.866	24.439	25.543	25.910	26.445	26.917	29.407
	(2.611)	(2.173)	(2.087)	(2.330)	(2.279)	(2.180)	(2.828)
$L2_{PMF}$	573.284	151.424	180.764	3627.946	139.163	278.286	306.958
	(164.946)	(3.520)	(4.947)	(502.006)	(3.354)	(6.435)	(23.519)

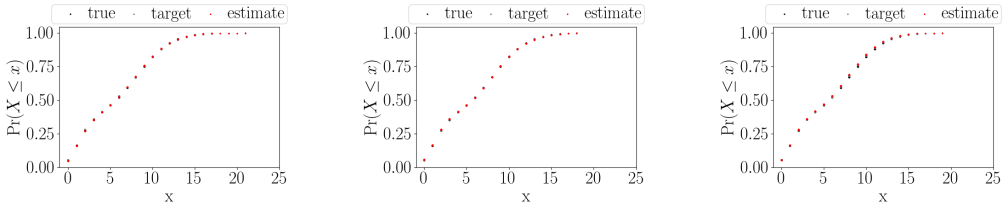
Values in the table scaled by 10^{-3}

Table 5.17 medium sample results of univariate mixed Poisson data

L2 losses for mixed Poisson data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PMF obtained by the proposed method for the validation and test samples.

Figures 5.50 and 5.51 present the best, average, and worst CDF and PMF estimates for the medium test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 6. The target CDF closely approximates the true CDF for all models. There are hardly any differences detected in the CDF estimates except for the worst estimate in Figure 5.50(c) which has a slight decrease in the right tail of the distribution. This is reflected in the corresponding PMF estimate depicted in Figure 5.51(c) which shows slight negative values in the right tail.

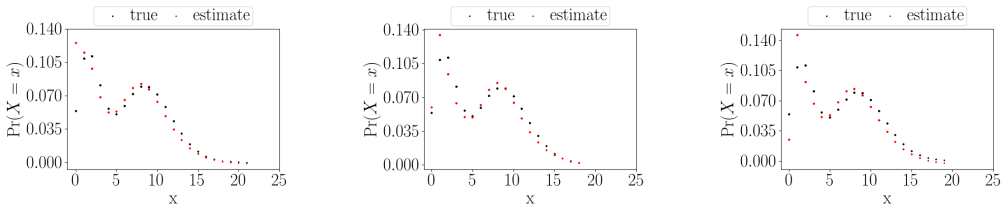
Furthermore, all PMF estimates show some differences in the left tail of the distribution.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.50 medium sample CDF estimates

True, target, and estimated CDF using the proposed method for mixed Poisson data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 9 neurons are given.

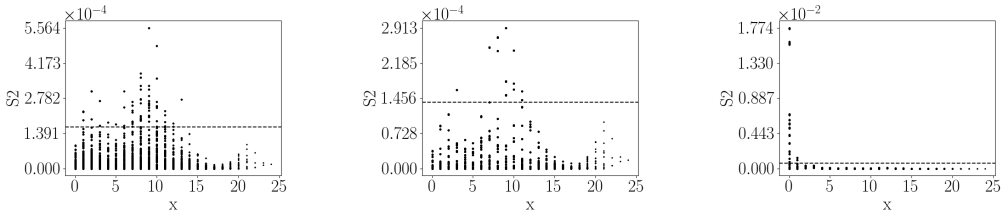


(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.51 medium sample PMF estimates

True and estimated PMF using the proposed method for mixed Poisson data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 9 neurons are given.

The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PMF}$ for each data point over 100 simulation replications using model 6 are provided in Figure 5.52. Figures 5.52(a) and 5.52(b) show similar patterns due to the close approximation of the target CDF. Again the CDF estimates struggle with the domain between 5 and 10. Furthermore, the right tail obtains relatively larger errors compared to the small sample size. In addition the left tail of the PMF is misfit many times, see Figure 5.52(c) for the $S2_{PMF}$ errors. This is reflected in the PMF estimates shown in Figure 5.51. The first data point is misfit many times. The corresponding errors are ten times smaller than for the small sample size. Hence the results are substantially improved by using more observations.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PMF}$

Figure 5.52 medium sample squared errors

Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PMF}$ differences together with the 95^{th} percentile for the medium sample size, using 2 hidden layers and 9 neurons of the proposed method.

Large sample results:

Table 5.18 presents the large sample results for the validation sample and test sample. Models 1, 2, 4, and 6 are the best performing models with smallest $L2_{\widetilde{CDF}}$ loss for the validation sample. Model 4 has the lowest $L2_{\widetilde{CDF}}$ loss and the smallest $L2_{PMF}$ loss. Similar conclusions can be drawn for the test sample. Hence a smaller model is needed for this larger sample size opposed to the small and medium sample size.

model	optimal	1	2	3	4	5	6
			Validation sample with $L2_{\widehat{CDF}}$= 20.893 (1.729)				
$L2_{\widetilde{CDF}}$	2.477	3.182	3.387	6.890	2.986	15.943	4.243
	(0.097)	(0.144)	(0.145)	(0.267)	(0.120)	(0.539)	(0.257)
$L2_{CDF}$	20.186	20.039	21.280	22.762	20.116	32.919	20.686
	(1.716)	(1.687)	(1.661)	(1.723)	(1.732)	(1.688)	(1.640)
$L2_{PMF}$	880.013	587.136	708.686	1335.271	526.858	1085.461	1626.181
	(66.604)	(7.698)	(7.527)	(150.754)	(6.545)	(12.897)	(68.018)
			Test sample with $L2_{\widehat{CDF}}$= 14.293 (0.888)				
$L2_{\widetilde{CDF}}$	6.406	5.948	6.512	8.145	5.991	17.652	7.479
	(0.393)	(0.277	(0.394)	(0.334)	(0.275)	(0.535)	(0.549)
$L2_{CDF}$	19.793	20.617	21.065	22.262	19.026	34.682	20.194
	(1.114)	(1.069)	(1.143)	(0.954)	(0.988)	(1.230)	(1.211)
$L2_{PMF}$	1087.165	561.749	787.191	4678.201	514.790	1115.912	1960.357
	(141.323)	(8.750)	(12.017)	(608.003)	(7.370)	(13.911)	(67.739)

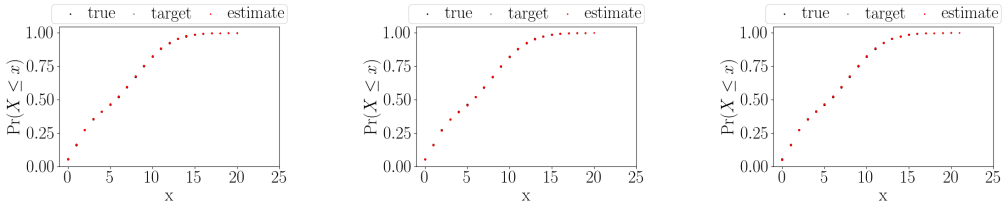
Values in the table scaled by 10^{-3}

Table 5.18 large sample results of univariate mixed Poisson data

L2 losses for mixed Poisson data, large sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PMF obtained by the proposed method for the validation and test samples.

Figures 5.53 and 5.54 present the best, average and worst CDF and PMF estimates for the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 4. The target CDF again closely approximates the true CDF for all models. Both CDF and PMF estimates look similar to each other. The corresponding PMF estimates again misfit the left tail of the distribution. Hence these conclusions are similar to the medium sample size.

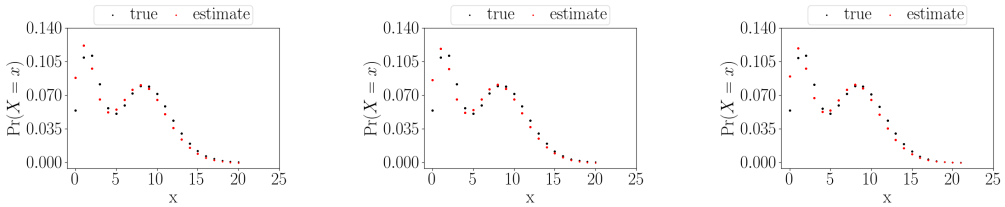
The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PMF}$ for each data point over 100 simulation replications using model 4 are provided in Figure 5.55. The approximation errors are



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.53 large sample CDF estimates

True, target, and estimated CDF using the proposed method for mixed Poisson data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 3 neurons are given.

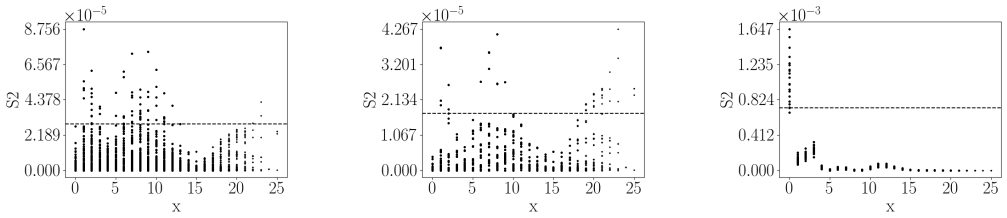


(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.54 large sample PMF estimates

True and estimated PMF using the proposed method for mixed Poisson data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 3 neurons are given.

very small as Figures 5.55(a) and 5.55(b) are very similar. Large errors are encountered in the left tail, the domain between 5 and 10 and the right tail. The right tail has relatively larger errors comparing these errors with the medium sample errors. However, these errors are ten times smaller than for the medium sample. The first data point estimate of the PMF remains an issue, see Figure 5.55(c). This error is as large as for the medium sample size.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PMF}$

Figure 5.55 large sample squared errors
Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PMF}$ differences together with the 95^{th} percentile for the large sample size, using 2 hidden layers and 3 neurons of the proposed method.

Model 4 outperforms all other models for all sample sizes in terms of $L2_{PMF}$ loss. However, model 6 outperforms model 4 for the small and medium sample size based on the feasible $L2_{\widetilde{CDF}}$ loss. As the sample size grows, the target CDF approximates the true CDF closer. All sample sizes show large $S2_{\widetilde{CDF}}$ and $S2_{CDF}$ errors for the domain values between 5 and 10. The medium and large sample sizes show large errors for these domain values as well as the right tail of the distribution. All sample sizes show large $S2_{PMF}$ errors only for the first two domain values, which corresponds to the left tail of the distribution. This might be caused by a combination of factors. Continuous functions are used to approximate a discrete distribution. This might be complicated, particularly if the distribution is also highly skewed as in the case of Poisson distributions with a small mean. Large errors indeed occur in the left area of the Poisson distribution.

5.3.2 Bivariate case

In this section we consider simulated data from a bivariate correlated mixed Poisson distribution. This is achieved by using three Poisson variables, Y_1, Y_2 and Y_{12} with corresponding parameters λ_i, with $i = 1, 2, 12$ in order to obtain two dependent Poisson variables X_1 and X_2, see Shin and Pasupathy (2007):

$$X_1 = Y_1 + Y_{12}$$
$$X_2 = Y_2 + Y_{12}.$$

The corresponding PMF is:

$$p(x_1, x_2) = exp(-(\lambda_1 + \lambda_2 + \lambda_{12})) \sum_{i=0}^{min(x_1,x_2)} \frac{\lambda_1^{x_1-i} \lambda_2^{x_2-i} \lambda_{12}^{i}}{(x_1 - i)!(x_2 - i)i!}$$

where $\lambda_1 = 4, \lambda_2 = 7$ and $\lambda_{12} = 10$. These parameter values are chosen such that the resulting correlation is substantial, which is calculated as follows:

$$\rho = \frac{\lambda_{12}}{\sqrt{(\lambda_1 + \lambda_{12})(\lambda_2 + \lambda_{12})}}$$

Using above parameter values the corresponding correlation amounts 0.65. For other details about this distribution, see Shin and Pasupathy (2007).

Results are shown for the medium and large sample size using 10,000 training epochs and a learning rate of 0.001. The logistic function is used as activation function in all layers except for the output layer. In Table 5.19 the different models that that are applied to the bivariate mixed Poisson distribution are summarized.

model	1	2	3	4	5	6	7	8	9
# hidden layers	1	1	1	2	2	2	3	3	3
# hidden neurons	10	25	50	10	25	50	10	25	50

Table 5.19 Overview of models considered for the bivariate mixed Poisson distribution

Medium sample results:

Table 5.20 presents the medium sample results for the validation sample and test sample. Model 8 is the best performing model obtaining the smallest $L2_{\widetilde{CDF}}$ for the validation sample and test sample. Smallest $L2_{\widetilde{CDF}}$ and $L2_{PMF}$ losses are obtained by models 6, 7, 8, and 9 in both the validation sample and test sample.

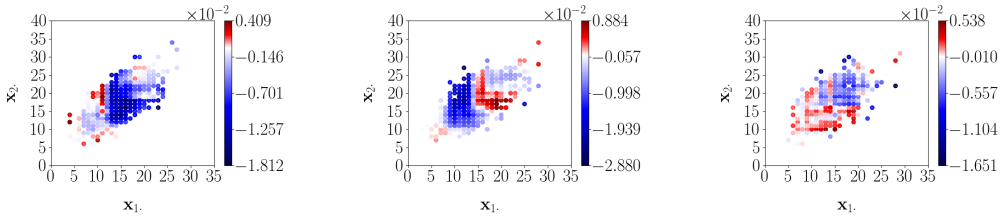
model	optimal	1	2	3	4	5	6	7	8	9
					Validation sample					
$L2_{\widetilde{CDF}}$	7.403	35.432	48.235	51.493	51.445	65.869	16.088	8.864	8.543	10.284
	(0.177)	(0.636)	(0.855)	(6.197)	(0.705)	(1.040)	(0.484)	(0.208)	(0.246)	(0.368)
$L2_{PMF}$	0.865	2.555	3.386	2.746	2.522	2.789	0.875	0.979	0.867	0.770
	(0.020)	(0.034)	(0.046)	(0.030)	(0.027)	(0.035)	(0.017)	(0.020)	(0.018)	(0.017)
					Test sample					
$L2_{\widetilde{CDF}}$	13.299	31.891	42.888	48.027	43.664	66.003	19.222	11.826	11.729	14.032
	(0.891)	(1.019)	(1.002)	(2.944)	(0.819)	(2.838)	(1.595)	(0.398)	(0.864)	(1.113)
$L2_{PMF}$	0.871	2.235	2.878	2.389	2.341	2.728	0.869	0.957	0.864	0.779
	(0.021)	(0.053)	(0.056)	(0.035)	(0.030)	(0.051)	(0.021)	(0.020)	(0.021)	(0.020)

Values in the table scaled by 10^{-3}

Table 5.20 medium sample results of bivariate mixed Poisson data
L2 losses for bivariate mixed Poisson data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the target and estimated CDF and between the true and estimated PMF obtained by the proposed method for the validation and test samples.

Figures 5.56 and 5.57 present differences between the target and estimated CDF and between the true and estimated PMF for the best, average, and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 8. See appendix F for the CDF and PMF estimates instead of differences shown in this section. All CDF estimates show more negative values than positive values.

The differences between the true and estimated PMF are illustrated in Figure 5.57. The differences between the loss values of the PMFs are a factor 10 smaller than the differences between the CDFs. The PMF estimates show similar patterns of positive/negative areas that flow into negative/positive values. More specifically, in the lower left area mostly positive loss values are found and the corresponding tails in this

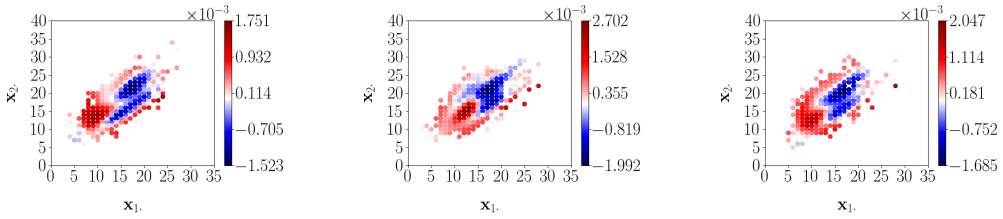


(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.56 medium sample CDF errors

Target and estimated CDF using the proposed method for mixed Poisson data, medium sample size. Differences between these are shown for the best, average and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications using 3 hidden layers and 25 neurons.

area are negative. In the upper right area mostly negative loss values are found and the corresponding outer tails in this area are positive.

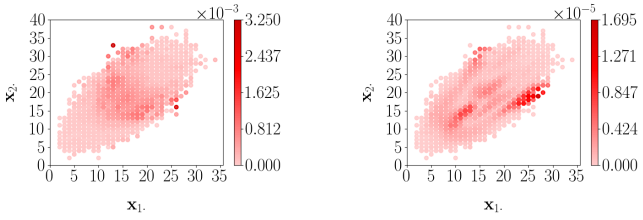


(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.57 medium sample PMF errors

True and estimated PMF using the proposed method for mixed Poisson data, medium sample size. Differences between these are shown for the best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications using 3 hidden layers and 25 neurons.

The squared errors $S2_{\widehat{CDF}}$ and $S2_{PMF}$ for each data point over 100 simulation replications using model 8 are provided in Figure 5.58. Tails of the distribution seem to be estimated well for both the CDF and PMF with some larger values for the mode. There are a few outliers in the tails for both cases. Note that the $S2_{PMF}$ errors are 100 times smaller than the $S2_{\widehat{CDF}}$ errors.



(a) $S2_{\widehat{CDF}}$ (b) $S2_{PMF}$

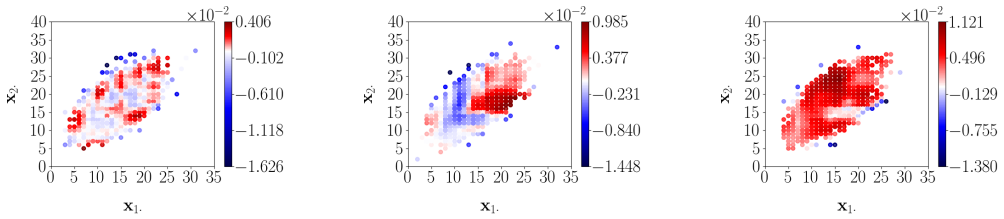
Figure 5.58 medium sample squared errors

Squared errors calculated using the $S2_{\widehat{CDF}}$ and $S2_{PMF}$ differences for the medium sample size, using 3 hidden layers and 25 neurons of the proposed method.

Large sample results:

Table 5.21 presents the large sample results for the validation sample and test sample. Models 6, 7, 8, and 9 have again a substantial smaller $L2_{\widetilde{CDF}}$ loss than smaller models. Model 8 is again the best performing model obtaining the smallest $L2_{\widetilde{CDF}}$ for the validation sample and test sample.

Figures 5.59 and 5.60 present differences between the target and estimated CDF and between the true and estimated PMF for the best, average, and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 8. See appendix F for the CDF and PMF estimates instead of differences shown in this section. Both positive and negative values are found, with largest deviations in the tails.

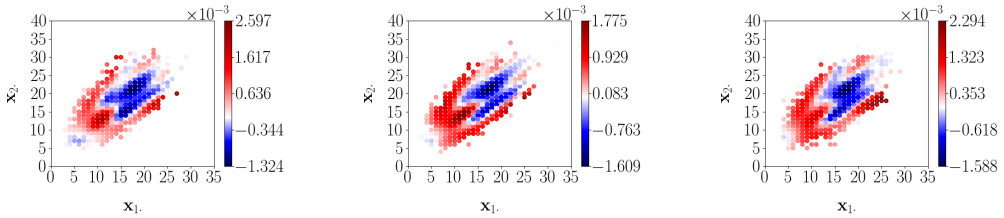


(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.59 large sample CDF errors

Target and estimated CDF using the proposed method for mixed Poisson data, large sample size. Differences between these are shown for the best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications using 3 hidden layers and 10 neurons.

The differences between the true and estimated PMF are illustrated in Figure 5.60. The loss values for these estimates are translated to a smaller scale compared with the CDF estimates. The loss values for these estimates are are a factor 10 smaller than the CDF estimates. Similar structures are found among these in line with the medium sample size.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.60 large sample PMF errors

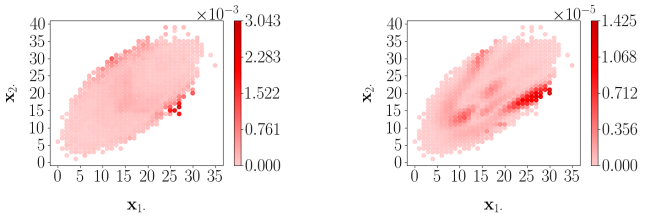
True and estimated PMF using the proposed method for mixed Poisson data, large sample size. Differences between these are shown for the best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications using 3 hidden layers and 10 neurons.

The squared errors $S2_{\widetilde{CDF}}$ and $S2_{PMF}$ for each data point over 100 simulation replications using model 8 are provided in Figure 5.61. Tails of the distribution seem to be estimated well for both the CDF and PMF with some larger values for the mode. There are a few outliers in the tails for both cases. Also the error values are as large for the medium sample size.

model	optimal	1	2	3	4	5	6	7	8	9
					Validation sample					
$L2_{\widetilde{CDF}}$	14.579	79.149	93.930	128.600	140.960	205.379	44.646	22.048	19.035	27.261
	(0.500)	(1.114)	(0.939)	(1.631)	(1.143)	(1.869)	(2.707)	(0.412)	(1.059)	(1.717)
$L2_{PMF}$	2.970	7.919	9.155	8.847	9.119	10.210	2.873	3.822	2.892	2.535
	(0.061)	(0.051)	(0.077)	(0.051)	(0.042)	(0.058)	(0.029)	(0.045)	(0.032)	(0.031)
					Test sample					
$L2_{\widetilde{CDF}}$	18.643	67.885	64.991	91.144	103.718	170.013	33.731	20.417	16.561	25.185
	(1.885)	(1.518)	(1.654)	(3.868)	(1.515)	(3.362)	(3.385)	(0.794)	(1.550)	(2.747)
$L2_{PMF}$	2.878	7.472	6.885	7.502	8.384	10.021	2.756	3.592	2.789	2.518
	(0.051)	(0.076)	(0.074)	(0.042)	(0.058)	(0.069)	(0.025)	(0.055)	(0.026)	(0.025)
										Values in the table scaled by 10^{-3}

Table 5.21 large sample results of bivariate mixed Poisson data

L2 losses for bivariate mixed Poisson data, large sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the target and estimated CDF and between the true and estimated PMF obtained by the proposed method for the validation and test samples.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{PMF}$

Figure 5.61 large sample squared errors
Squared errors calculated using the $S2_{\widetilde{CDF}}$ and $S2_{PMF}$ differences for the large sample size, using 3 hidden layers and 10 neurons of the proposed method.

So the difference between the medium and large sample sizes are negligible. Both sample sizes use model 8 as optimal model. The resulting PMFs both show a certain pattern flowing from negative/positive to positive/negative values with in between values close to zero. Both show similar squared errors in the exact same areas. For the CDF and the PMF the mode has larger values with large errors for a 2 areas of the tail. The squared errors for the PMF are 100 times smaller though.

5.4 Correlated Standard Normal Distribution

5.4.1 Bivariate case

In this section we consider simulated data from two standard normal distributions, with correlation $\rho = 0.5$:

$$p(x) = (2\pi|\Sigma|)^{-0.5} exp(-0.5(x-\mu)^T \Sigma^{-1}(x-\mu))$$

where the mean vector is $\mu = (0,0)^T$ and the covariance matrix is $\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$.
Results are shown for the medium sample size and the large sample size using 5,000 training epochs and a learning rate of 0.001. The logistic function is used as activation function in all layers except for the output layer. In Table 5.22 the different models that are applied to the bivariate mixed normal distribution are summarized.

model	1	2	3	4	5	6	7	8	9
# hidden layers	1	1	1	2	2	2	3	3	3
# hidden neurons	10	25	50	10	25	50	10	25	50

Table 5.22 Overview of models considered for the bivariate mixed normal distribution

Medium sample results:

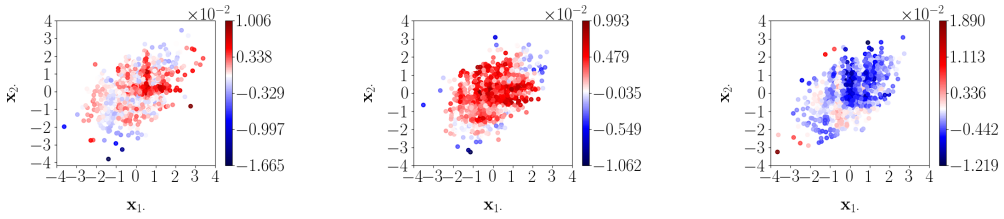
Table 5.23 presents the medium sample results for the validation sample and test sample. Model 4 attains the smallest $L2_{\widetilde{CDF}}$ loss for the validation sample. The smallest $L2_{PDF}$ loss is obtained by model 6, but closely followed by model 4. Additionally, model 4 obtains the smallest $L2_{\widetilde{CDF}}$, $L2_{CDF}$ and $L2_{PDF}$ losses for the test sample.

Figures 5.62 and 5.63 present differences between the target and estimated CDF and between the true and estimated PDF for the best, average and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 4. See Appendix G for the CDF and PDF estimates instead of differences shown in this section. The best CDF estimate in Figure

model	optimal	1	2	3	4	5	6	7	8	9
	Validation sample with $L2_{\widehat{CDF}}$= 36.094 (3.026)									
$L2_{\widehat{CDF}}$	6.451	19.489	7.661	10.721	6.773	7.276	10.257	8.028	8.470	13.751
	(0.104)	(0.325)	(0.135)	(0.160)	(0.111)	(0.135)	(0.287)	(0.165)	(0.145)	(0.341)
$L2_{CDF}$	29.478	41.471	30.681	33.534	29.652	29.824	32.678	31.280	31.779	36.767
	(2.444)	(2.527)	(2.497)	(2.643)	(2.510)	(2.440)	(2.650)	(2.522)	(2.473)	(2.600)
$L2_{PDF}$	35.091	113.98	138.390	99.219	31.586	39.680	28.928	57.269	57.786	67.489
	(1.227)	(1.903)	(0.972)	(2.230)	(1.027)	(0.947)	(1.066)	(1.580)	(1.100)	(12.525)
	Test sample with $L2_{\widehat{CDF}}$= 27.958 (2.362)									
$L2_{\widehat{CDF}}$	9.753	20.211	10.558	12.902	9.621	10.862	13.734	10.857	11.803	17.795
	(0.362)	(0.531)	(0.358)	(0.506)	(0.346)	(0.510)	(0.929)	(0.409)	(0.560)	(1.109)
$L2_{CDF}$	33.241	41.917	33.847	33.713	32.460	34.083	36.243	34.009	34.854	40.249
	(3.241)	(3.080)	(3.250)	(2.992)	(3.183)	(3.303)	(3.433)	(3.113)	(3.241)	(4.050)
$L2_{PDF}$	29.056	85.391	31.484	68.669	25.247	34.284	25.841	47.434	50.428	56.497
	(1.141)	(1.698)	(0.892)	(1.894)	(0.991)	(0.913)	(1.068)	(1.677)	(1.195)	(1.359)
										Values in the table scaled by 10^{-3}

Table 5.23 medium sample results of bivariate mixed normal data
L2 losses for bivariate mixed normal data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

5.62(a), attains many point loss values close to 0. Several positive point losses are found close to the mode, in the upper left area of the distribution. The average CDF estimate in Figure 5.62(b) has many losses scattered over the surface, not related to a specific area. The most outward data points have negative loss values opposed to the more centered data points that have positive loss values. The worst CDF estimate in Figure 5.62(c) has many negative loss values around the upper left area of the distribution.

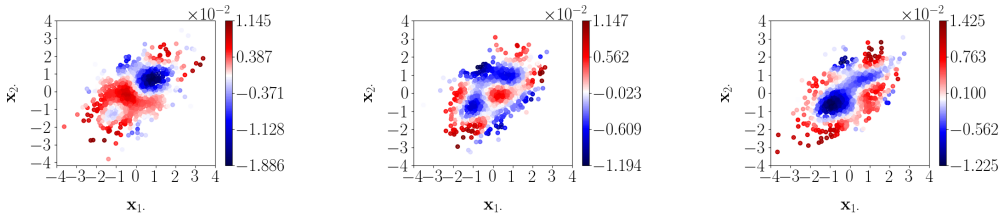


(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure 5.62 medium sample CDF errors
The difference between target and estimated CDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the best, average, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications using 2 hidden layers and 10 neurons.

The PDF estimates in Figure 5.63 attain similar patterns found for the bivariate Poisson distribution. Positive loss values are found in the tails in the lower left and upper right area of the PDF estimates. Note that the the transition areas from negative to positive or vice versa are very close to 0. Largest losses are found in the lower left and upper right tails.

The squared errors $S2_{\widehat{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation

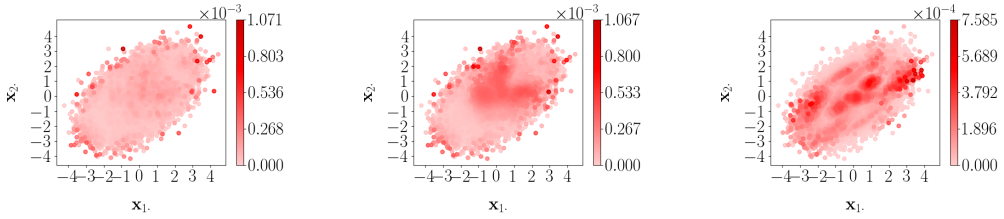


(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.63 medium sample PDF errors

The difference between true and estimated PDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications using 2 hidden layers and 10 neurons.

replications using model 4 are illustrated by Figure 5.64. The $S2_{\widetilde{CDF}}$ errors depicted by Figure 5.64(a) are mainly located in the tails. The $S2_{CDF}$ errors in Figure 5.64(b) confirm this by showing similar errors in these tails. The $S2_{CDF}$ errors around the upper left area of the mode are approximation errors of the target CDF as Figure 5.64(a) does not show these large error values around the mode. The corresponding $S2_{PDF}$ errors in Figure 5.64(c) are smaller than for the $S2_{\widetilde{CDF}}$ and $S2_{CDF}$ errors. Largest errors are found in the mode and lower left and upper right tails.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure 5.64 medium sample squared errors

Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences for the medium sample size, using 2 hidden layers and 10 neurons of the proposed method.

Large sample results:

Table 5.24 presents the large sample results for the validation sample and test sample. Model 4 attains the smallest $L2_{\widetilde{CDF}}$ loss for the validation sample. The smallest $L2_{PDF}$ loss is obtained by model 4 as well opposed to the medium sample results. Additionally, model 4 obtains the smallest $L2_{\widetilde{CDF}}$, $L2_{CDF}$ and $L2_{PDF}$ losses for the test sample in line with the medium sample results.

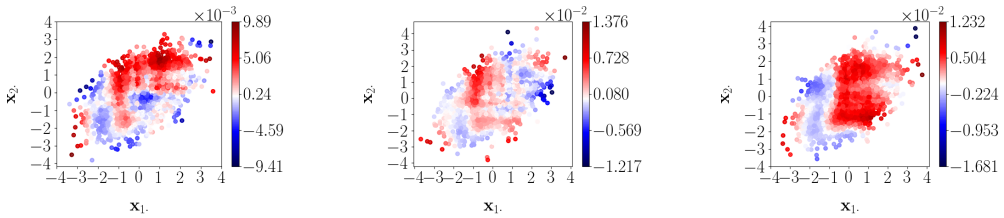
Figures 5.65 and 5.66 present differences between the target and estimated CDF and between the true and estimated PDF for the best, average, and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 4. See Appendix G for the CDF and PDF estimates instead of differences shown in this section. The loss differences of Figure 5.65 are smallest for the best CDF estimate and largest for the average and worst CDF estimates. The best CDF estimate in Figure 5.65(a) attains largest point loss values in the tails. The average CDF estimate in Figure 5.65(b) has many loss values scattered over the surface close to 0. The data points in the tails have negative and positive loss values

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample with $L2_{\widehat{CDF}}$= 31.097 (2.479)										
$L2_{\widetilde{CDF}}$	8.752	46.279	11.779	20.496	9.074	12.975	25.349	11.057	18.656	38.975
	(0.126)	(0.548)	(0.166)	(0.290)	(0.146)	(0.218)	(1.057)	(0.226)	(0.366)	(2.125)
$L2_{CDF}$	30.148	65.612	32.865	40.584	30.025	34.360	47.031	32.622	40.294	60.885
	(2.372)	(2.492)	(2.463)	(2.408)	(2.391)	(2.534)	(3.369)	(2.514)	(2.586)	(3.521)
$L2_{PDF}$	61.326	277.556	76.427	250.847	55.420	104.885	82.049	90.674	167.090	169.358
	(1.414)	(2.895)	(1.373)	(4.068)	(1.082)	(1.296)	(2.399)	(2.859)	(2.495)	(2.155)
Test sample with $L2_{\widehat{CDF}}$= 23.321 (1.689)										
$L2_{\widetilde{CDF}}$	11.372	43.159	15.490	20.168	14.101	20.105	33.199	14.696	23.145	41.006
	(0.421)	(0.886)	(0.777)	(0.632)	(0.709)	(1.211)	(3.171)	(0.613)	(0.135)	(3.262)
$L2_{CDF}$	30.051	62.165	36.033	39.701	34.587	40.778	52.008	36.156	45.075	62.871
	(2.264)	(3.024)	(2.720)	(2.433)	(2.769)	(3.675)	(5.185)	(3.247)	(3.678)	(5.622)
$L2_{PDF}$	63.700	204.460	59.234	175.374	44.324	77.108	63.620	69.144	122.960	124.751
	(2.988)	(1.937)	(1.167)	(3.184)	(0.961)	(1.193)	(2.197)	(6.914)	(3.309)	(2.446)
Values in the table scaled by 10^{-3}										

Table 5.24 large sample results of bivariate mixed normal data

L2 losses for bivariate mixed normal data, large sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

which are larger than in the mode. The worst CDF estimate in Figure 5.65(c) has many positive loss values around the upper right centered area of the distribution. All CDF estimates have positive loss values in the lower left tail of the distribution and negative loss values in the upper right area of the distribution. This implies that all estimated data points are higher than the target data points in this lower area and lower than the target data points in the upper area.

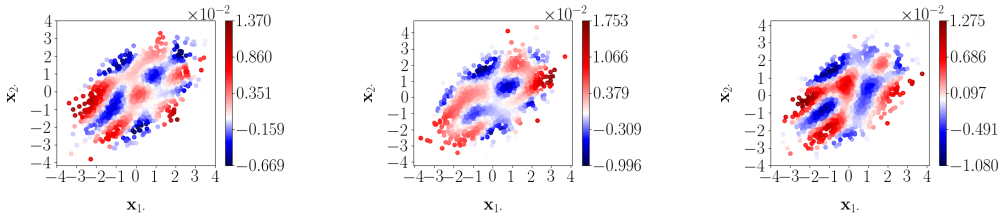


(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.65 large sample CDF errors

The difference between target and estimated CDF using the proposed method for mixed normal data, large sample size. Differences are shown for the best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications using 2 hidden layers and 10 neurons.

The PDF estimates in Figure 5.66 all have in common that positive and negative losses are dispersed over the distribution area which merge together with loss values very close to 0. They all look very similar. The lower left area contains mainly positive loss values. The most upper right area has negative loss values. Largest losses are found in the lower left and upper right tails in line with the medium sample results. Note that these areas are smaller and flow over more often than for the medium sample though.



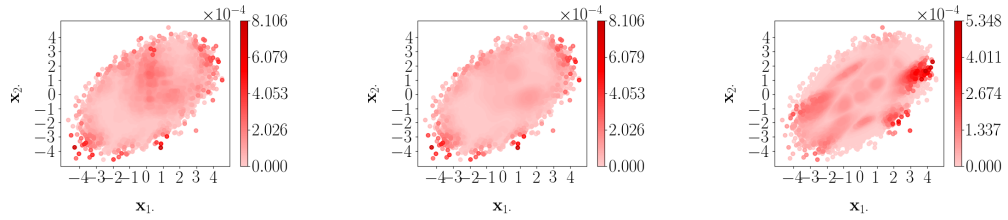
(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure 5.66 large sample PDF errors

The difference between true and estimated PDF using the proposed method for mixed normal data, large sample size. Differences are shown for the best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications using 2 hidden layers and 10 neurons.

The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 4 are illustrated by Figure 5.67. The largest $S2_{\widetilde{CDF}}$ errors depicted by Figure 5.67(a) are located both around the mode and in the upper right and lower left tails. Also the $S2_{CDF}$ errors in Figure 5.67(b) confirm this by showing similar errors in these tails. Approximation errors between the true and target CDF are smaller than for the medium sample size as Figures 5.67(a) and 5.67(b) look very similar. The main difference between these figures, in other words the approximation error of the target CDF, is located around the mode. These conclusions are similar for the medium sample errors. The corresponding $S2_{PDF}$ errors in Figure 5.64(c) are again slightly smaller

than for the $S2_{\widetilde{CDF}}$ and $S2_{CDF}$ errors. Largest errors are found in the lower left and right sided tails similarly as for the medium sample size.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure 5.67 large sample squared errors
Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences for the large sample size, using 2 hidden layers and 10 neurons of the proposed method.

Estimates of the large sample size improve compared to the medium sample size, using 2 hidden layers and 10 neurons for both sample sizes. This can be seen by the smaller squared error values depicted in Figures 5.64 and 5.67. We next compare the results of the proposed method with KDE for all sample sizes.

Results for KDE:

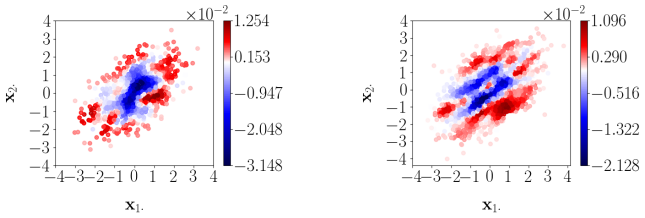
For the ease of comparison, we present the test sample results of the proposed method and the KDE for all sample sizes in Table 5.25. For both sample sizes, KDE has a substantial larger $L2_{PDF}$ loss value. Opposed to the univariate simulation cases, the $L2_{PDF}$ loss values obtained by the proposed method increase when the sample sizes increase though the standard errors decrease. The standard errors of the KDE increase as sample sizes increase though. However, the increase in the $L2_{PDF}$ loss values is relatively slower than for the proposed method.

Method	Bandwidth	Sample Size	
		medium	large
proposed		25.247	44.324
		(9.910)	(9.607)
KDE	Scott	114.095	198.405
		(44.907)	(65.469)
Values in the table scaled by 10^{-3}			

Table 5.25 test sample results of bivariate normal data
The mean (standard error) of the calculated $L2_{PDF}$ loss obtained by model 4 for the proposed method and the KDE of the test sample of the medium and large sample sizes. Only the optimal bandwidth for KDE is shown.

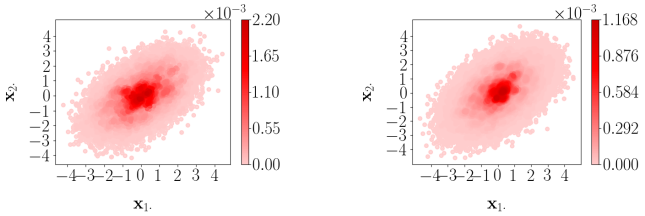
Figure 5.68 presents differences between the true and estimated PDF for the average estimate of the test sample in terms of $L2_{PDF}$ loss. See Appendix H for the PDF estimates instead of differences shown in this section. Largest loss differences are found around the mode. These are mostly negative though for the large sample size the peak has positive loss differences. The squared errors of $S2_{PDF}$ for each data point over 100 simulation replications are illustrated by Figure 5.69(a) for the medium sample size and Figure 5.69(b) for the large sample size. Both illustrate that KDE have large errors around the mode in line with Figure 5.68. The proposed method shows errors over the entire distribution flowing from positive/negative errors to negative/positive errors. The

proposed method has 3 times smaller error values for the medium sample, see Figure 5.63, and 2.5 times smaller error values for the large sample compared to KDE, see Figure 5.66.



(a) medium sample size (b) large sample size

Figure 5.68 PDF errors by KDE
The difference between true and estimated PDF using KDE for mixed normal data. Differences are shown for the average estimate in terms of $L2_{PDF}$ loss values out of 100 simulation replications.



(a) medium sample size (b) large sample size

Figure 5.69 squared errors obtained by the KDE
Squared errors calculated using the $S2_{PDF}$ differences for the medium and large sample size, using KDE together with Scott's bandwidth.

5.4.2 Trivariate case

In this section we consider simulated data from three standard normal distributions, with correlation $\rho = 0.5$:

$$p(x) = (2\pi|\Sigma|)^{-0.5} exp(-0.5(x-\mu)^T \Sigma^{-1}(x-\mu))$$

with mean vector $\mu = (0,0,0)^T$ and covariance matrix $\Sigma = \begin{bmatrix} 1.0 & 0.5 & 0.5 \\ 0.5 & 1.0 & 0.5 \\ 0.5 & 0.5 & 1.0 \end{bmatrix}$. Results are shown for the medium and large sample size using 20,000 training epochs and a learning rate of 0.001. The logistic function is used as activation function in all layers except for the output layer. Negative values are truncated to zero as negative probabilities are not feasible. In Table 5.26 the different models that are applied to the bivariate mixed normal distribution are summarized.

model	1	2	3	4	5	6	7	8	9
# hidden layers	2	2	2	3	3	3	4	4	4
# hidden neurons	10	25	50	10	25	50	10	25	50

Table 5.26 Overview of models considered for the trivariate mixed normal distribution

Medium sample results:

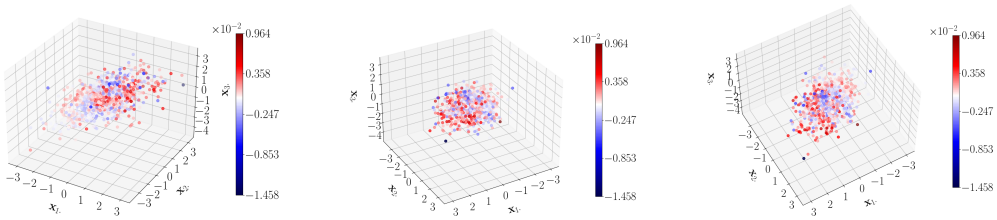
Table 5.27 presents the medium sample results for the validation sample and test sample. Model 5 obtains the smallest $L2_{\widetilde{CDF}}$ loss for the validation sample and test sample. For both samples, models that use more than 10 neurons obtain consistently relatively small $L2_{PDF}$ losses with smallest values for the least number of hidden layers used.

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample with $L2_{\widehat{CDF}}$= 30.535 (2.054)										
$L2_{\widetilde{CDF}}$	6.294	8.597	6.998	7.871	7.512	7.025	7.943	12.107	7.160	8.494
	(0.083)	(0.113)	(0.102)	(0.172)	(0.103)	(1.130)	(0.163)	(0.199)	(0.125)	(0.181)
$L2_{CDF}$	28.128	29.004	28.535	29.972	28.153	28.930	30.115	32.702	28.766	30.245
	(1.977)	(1.986)	(2.042)	(2.186)	(2.025)	(2.037)	(2.146)	(2.037)	(1.945)	(2.132)
$L2_{PDF}$	34.221	43.157	30.858	27.245	43.382	34.341	36.148	94.932	44.284	31.021
	(0.878)	(0.792)	(0.568)	(0.450)	(0.679)	(0.542)	(0.648)	(2.564)	(0.873)	(0.512)
Test sample with $L2_{\widehat{CDF}}$= 25.264 (1.716)										
$L2_{\widetilde{CDF}}$	9.197	10.184	9.105	9.614	9.273	8.650	9.713	12.956	9.377	10.276
	(0.450)	(0.307)	(0.393)	(0.534)	(0.338)	(0.365)	(0.477)	(0.385)	(0.423)	(0.562)
$L2_{CDF}$	19.861	19.819	19.771	20.360	18.726	20.228	20.756	22.812	20.886	21.399
	(1.196)	(1.026)	(1.152)	(1.301)	(1.027)	(1.345)	(1.265)	(1.116)	(1.352)	(1.395)
$L2_{PDF}$	31.396	38.905	27.894	23.485	44.664	31.316	31.511	89.191	40.409	26.040
	(0.893)	(0.886)	(0.626)	(0.497)	(0.965)	(0.677)	(0.646)	(3.196)	(0.728)	(0.576)
Values in the table scaled by 10^{-3}										

Table 5.27 medium sample results of trivariate mixed normal data
L2 losses for trivariate mixed normal data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.

Figures 5.70, 5.71, and 5.72 present differences between target and estimated CDF for the best, average, and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 5. Figures 5.73, 5.74, and 5.75 present differences between the true and estimated PDF for the best, average, and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ using model 5. Differences are now in two-dimensional space and are plotted from three different angles. See Appendix I for the CDF and PDF estimates instead of differences shown in this section. The average and worst CDF estimates obtain more negative point error values than positive point error values. The best CDF estimate obtains the smallest errors. All PDF estimates obtain a similar pattern as for the bivariate mixed Poisson and mixed normal distribution cases before where positive/negative error areas flow into negative/positive error areas with in between error values that are very close to 0.

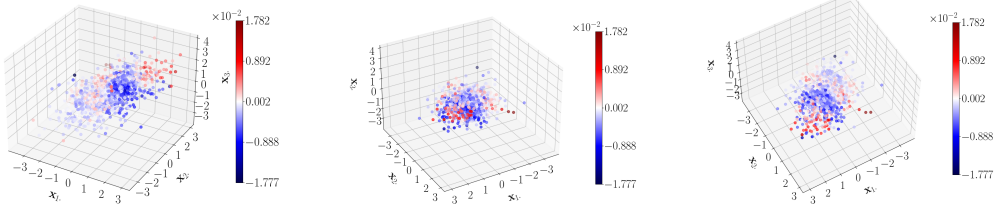
The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 5 are provided in Figures 5.76, 5.77, and 5.78. Largest $S2_{\widetilde{CDF}}$ error values are obtained for the upper left area. To a lesser extent $S2_{CDF}$ error values are obtained for the same area. The difference between these two error values are the approximation errors obtained by the target CDF. The $S2_{PDF}$ error values are also largest for the upper left area of the distribution.



(a) front angle (b) left angle (c) top angle

Figure 5.70 medium sample best CDF errors

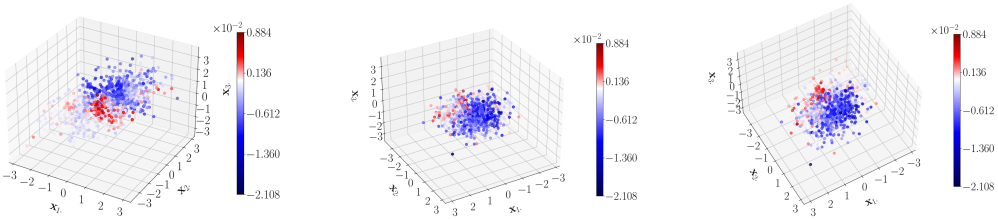
The difference between target and estimated CDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the best estimate in terms of L2$_{\widetilde{CDF}}$ loss using 3 hidden layers and 25 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.71 medium sample average CDF errors

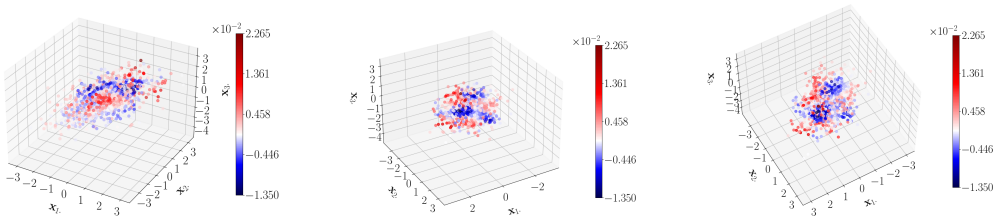
The difference between target and estimated CDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the average estimate in terms of L2$_{\widetilde{CDF}}$ loss using 3 hidden layers and 25 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.72 medium sample worst CDF errors

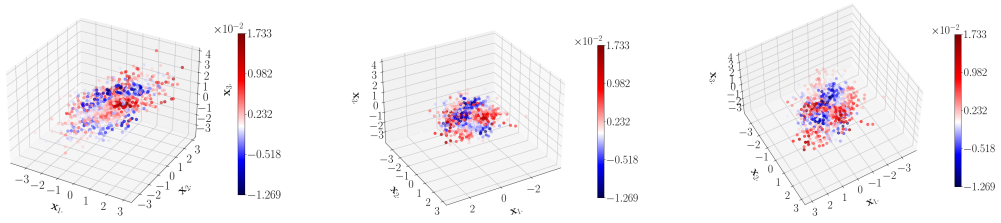
The difference between target and estimated CDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the worst estimate in terms of L2$_{\widetilde{CDF}}$ loss using 3 hidden layers and 25 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.73 medium sample best PDF errors

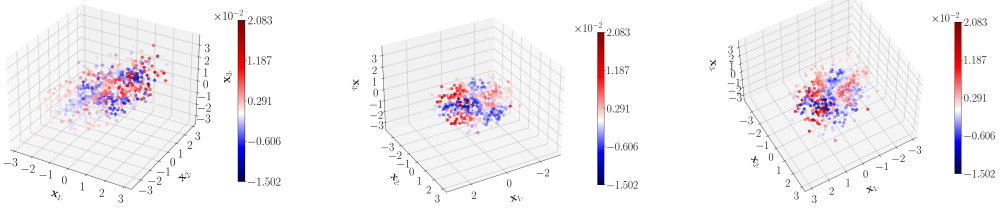
The difference between true and estimated PDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the best estimate in terms of L2$_{\widetilde{CDF}}$loss using 3 hidden layers and 25 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.74 medium sample average PDF errors

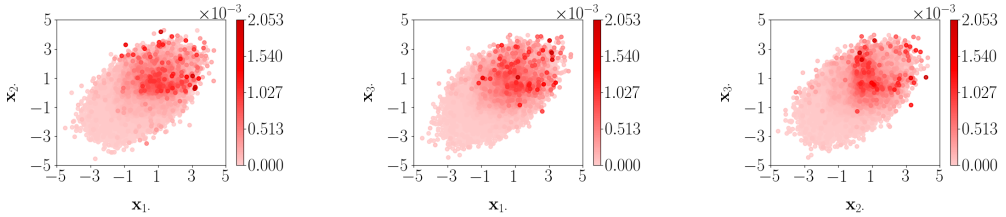
The difference between true and estimated PDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the average estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 25 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.75 medium sample worst PDF errors

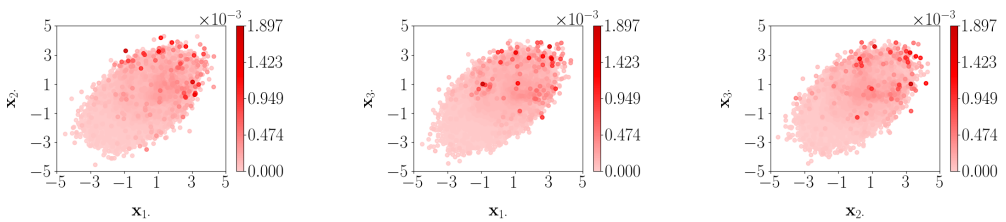
The difference between true and estimated PDF using the proposed method for mixed normal data, medium sample size. Differences are shown for the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 25 neurons.



(a) V1 V2 (b) V1 V3 (c) V2 V3

Figure 5.76 medium sample $S2_{\widetilde{CDF}}$ squared errors

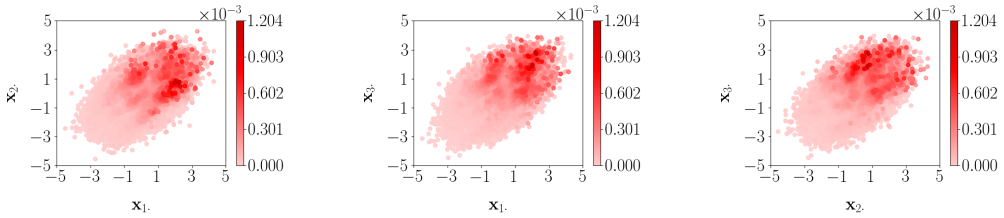
Squared errors calculated using $S2_{\widetilde{CDF}}$ differences for the medium sample size, using 3 hidden layers and 25 neurons of the proposed method.



(a) V1 V2 (b) V1 V3 (c) V2 V3

Figure 5.77 medium sample $S2_{CDF}$ squared errors

Squared errors calculated using $S2_{CDF}$ differences for the medium sample size, using 3 hidden layers and 25 neurons of the proposed method.



(a) V1 V2 (b) V2 V3 (c) V1 V3

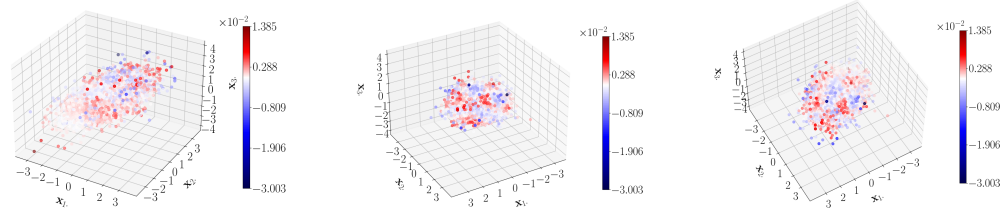
Figure 5.78 medium sample $S2_{PDF}$ squared errors

Squared errors calculated using $S2_{PDF}$ differences for the medium sample size, using 3 hidden layers and 25 neurons of the proposed method.

Large sample results:

Table 5.28 presents the large sample results for the validation sample and test sample. Model 4 obtains the smallest $L2_{\widetilde{CDF}}$ loss value for the validation sample. This model obtains one of the the largest $L2_{PDF}$ loss for the test sample and $L2_{PDF}$ loss for the validation sample out of all models. For both samples, models that use more than 10 neurons obtain consistently relatively small $L2_{PDF}$ losses with smallest values for the least number of hidden layers used. This is in line with the medium sample results. By using models with sufficient number of neurons, the tails of the distribution are closely estimated. Hence model 5 better estimates the tails than model 4. Therefore the results of model 5 are also shown for the large sample, see appendix I.

Figures 5.79, 5.80, and 5.81 present differences between the target and estimated CDF for the best, average, and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 4. Figures 5.82, 5.83, and 5.84 present differences between true and estimated PDF for the best, average, and worst estimates of the test sample in terms of $L2_{\widetilde{CDF}}$ using model 4. Differences are now in two-dimensional space and are plotted from three different angles. See Appendix I for the CDF and PDF estimates instead of differences shown in this section. The worst CDF estimate obtains more negative point error values than positive point error values. The best CDF estimate obtains the smallest scales of errors. All PDF estimates again obtain this particular pattern where positive/negative error areas flow into negative/positive error areas with in between error values that are very close to 0 in line with the medium sample results.



(a) front angle (b) left angle (c) top angle

Figure 5.79 large sample best CDF errors

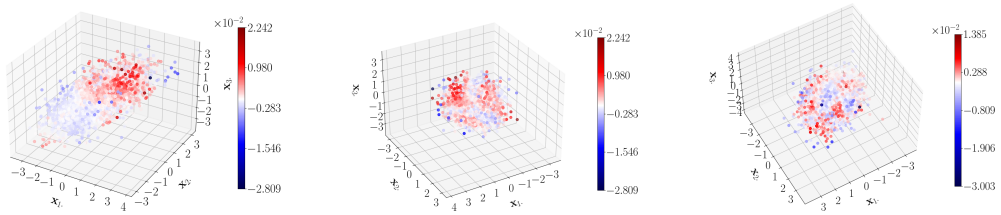
The difference between target and estimated CDF using the proposed method for mixed normal data, large sample size. Differences are shown for the best estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 10 neurons.

The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 4 are provided in Figures 5.85, 5.86, and 5.87. Largest $S2_{\widetilde{CDF}}$ and $S2_{CDF}$ error values are obtained for the tails in the upper left area. The $S2_{\widetilde{CDF}}$ errors are

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample with $L2_{\widehat{CDF}}$= 41.324 (3.399)										
$L2_{\widetilde{CDF}}$	11.537	15.802	12.963	18.755	12.945	14.156	16.839	28.351	14.995	19.440
	(0.125)	(0.166)	(0.209)	(0.638)	(0.125)	(0.347)	(0.561)	(0.399)	(0.267)	(0.704)
$L2_{CDF}$	40.586	44.502	42.256	49.011	41.498	44.610	47.329	56.932	45.316	50.662
	(3.237)	(3.389)	(3.382)	(3.767)	(3.471)	(3.439)	(3.438)	(3.542)	(3.460)	(3.629)
$L2_{PDF}$	104.205	116.052	96.318	80.702	143.394	95.323	99.484	358.237	121.646	93.738
	(2.481)	(1.830)	(1.068)	(0.898)	(1.498)	(1.400)	(1.220)	(6.740)	(1.430)	(1.213)
Test sample with $L2_{\widehat{CDF}}$= 36.642 (3.068)										
$L2_{\widetilde{CDF}}$	18.682	16.138	14.765	21.923	19.351	15.014	16.402	21.797	17.567	21.553
	(1.871)	(0.331)	(0.865)	(2.154)	(0.407)	(0.649)	(1.152)	(0.558)	(1.109)	(1.742)
$L2_{CDF}$	47.907	43.887	40.368	56.635	45.876	46.795	50.269	48.695	46.171	49.174
	(4.313)	(3.284)	(3.683)	(4.844)	(3.569)	(4.876)	(4.174)	(4.038)	(4.604)	(4.814)
$L2_{PDF}$	99.382	99.281	68.871	62.104	210.858	77.544	77.105	241.323	89.557	78.394
	(5.767)	(1.433)	(1.004)	(0.726)	(1.913)	(0.997)	(1.097)	(3.973)	(1.769)	(1.461)
Values in the table scaled by 10^{-3}										

Table 5.28 large sample results of trivariate mixed normal data

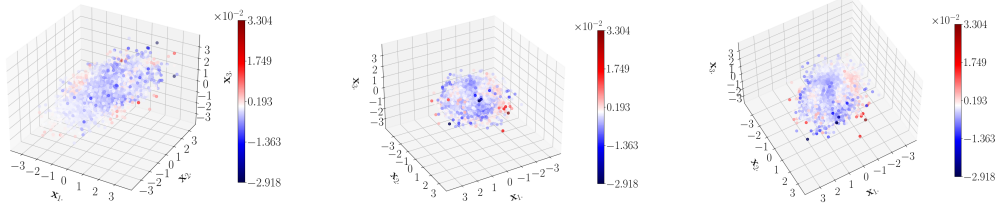
L2 losses for trivariate mixed normal data, large sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.



(a) front angle (b) left angle (c) top angle

Figure 5.80 large sample average CDF errors

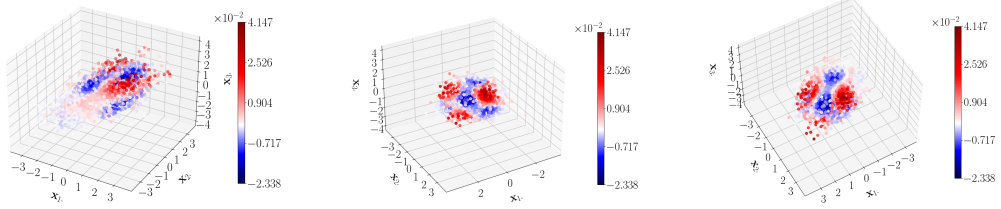
The difference between target and estimated CDF using the proposed method for mixed normal data, large sample size. Differences are shown for the average estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 10 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.81 large sample worst CDF errors

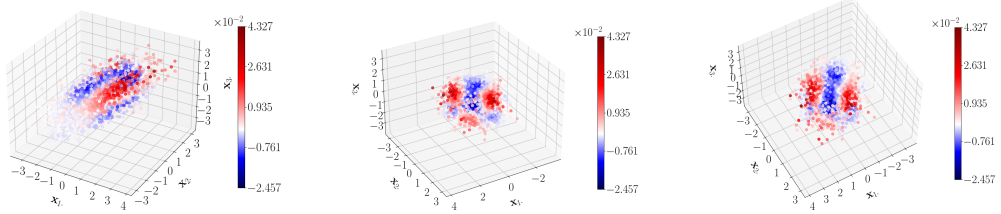
The difference between target and estimated CDF using the proposed method for mixed normal data, large sample size. Differences are shown for the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 10 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.82 large sample best PDF errors

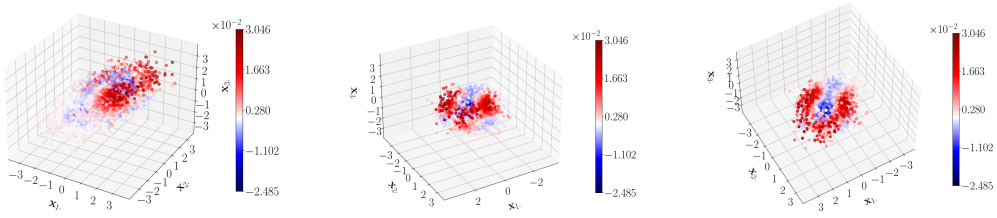
The difference between true and estimated PDF using the proposed method for mixed normal data, large sample size. Differences are shown for the best estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 10 neurons.



(a) front angle (b) left angle (c) top angle

Figure 5.83 large sample average PDF errors

The difference between true and estimated PDF using the proposed method for mixed normal data, large sample size. Differences are shown for the average estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 10 neurons.

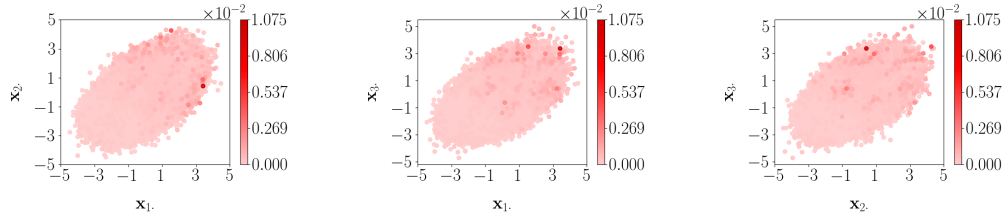


(a) front angle (b) left angle (c) top angle

Figure 5.84 large sample worst PDF errors

The difference between true and estimated PDF using the proposed method for mixed normal data, large sample size. Differences are shown for the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss using 3 hidden layers and 10 neurons.

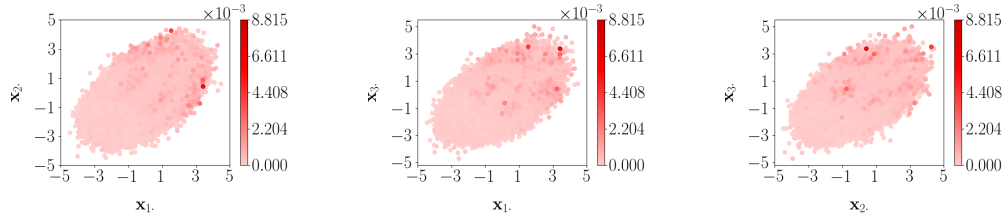
larger than the $S2_{CDF}$ errors but located at the exact same area. The $S2_{PDF}$ error values are largest for the upper left area of the distribution in line with the medium sample results.



(a) V1 V2 (b) V1 V3 (c) V2 V3

Figure 5.85 large sample $S2_{\widetilde{CDF}}$ squared errors

Squared errors calculated using $S2_{\widetilde{CDF}}$ differences for the large sample size, using 3 hidden layers and 10 neurons of the proposed method.

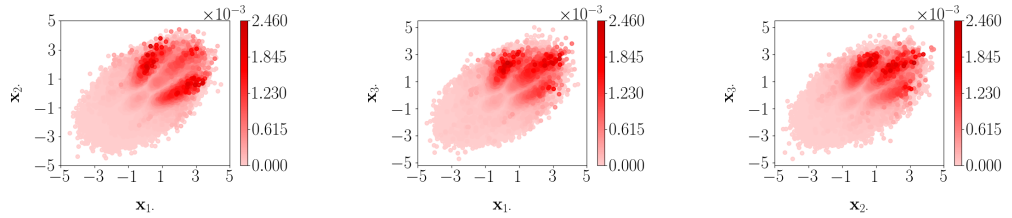


(a) V1 V2 (b) V1 V3 (c) V2 V3

Figure 5.86 large sample $S2_{CDF}$ squared errors

Squared errors calculated using $S2_{CDF}$ differences for the large sample size, using 3 hidden layers and 10 neurons of the proposed method.

We next compare the results of the proposed method with KDE for all sample sizes.



(a) V1 V2 (b) V1 V3 (c) V2 V3

Figure 5.87 large sample $S2_{PDF}$ squared errors
Squared errors calculated using $S2_{PDF}$ differences for the large sample size, using 3 hidden layers and 10 neurons of the proposed method.

Results for KDE:

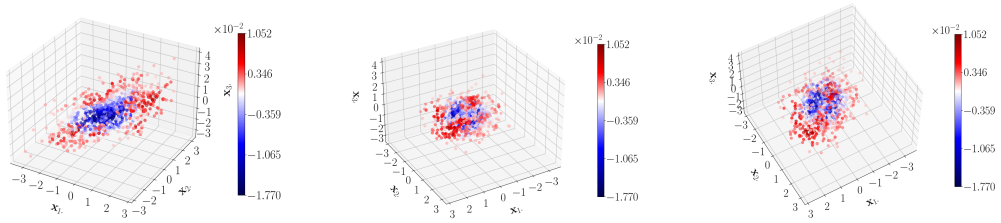
For the ease of comparison, we present the test sample results of the proposed method and the KDE for all sample sizes in Table 5.29. For both samples the proposed method using model 5 outperforms KDE irrespective which bandwidth is used. However, KDE, irrespective which bandwidth, outperforms the proposed method using model 4. More specifically, models 2, 3, 5, and 6 outperform KDE for the medium sample size and models 2 and 3 outperform KDE for the large sample size. These models use 25 or 50 neurons and 2 or 3 hidden layers. Hence models that use 10 neurons or 4 hidden layers are outperformed by KDE. This emphasizes the importance of selecting the optimal model for each distribution case. In this case, using too few hidden neurons or too many hidden layers results in larger $S2_{PDF}$ error values.

Method	Model	Bandwidth	Sample Size	
			medium	large
proposed	model 4		44.664	210.858
			(0.965)	(1.913)
proposed	model 5		31.316	77.544
			(0.677)	(0.997)
KDE		Scott	37.659	78.596
			(1.012)	(2.031)
KDE		Silverman	36.792	76.083
			(0.951)	(1.915)
			Values in the table scaled by 10^{-3}	

Table 5.29 test sample results of bivariate normal data
The mean (standard error) of the calculated $L2_{PDF}$ loss obtained by model 4 for the proposed method and the KDE of the test sample of the medium and large sample sizes. Only the optimal bandwidth for KDE is shown.

Figures 5.88 and 5.89 present differences between the true and estimated PDF for the average estimate of the test sample in terms of $L2_{PDF}$ loss. See Appendix J for the PDF estimates instead of differences shown in this section. For both samples, the mode of the distribution is higher than estimated by the KDE and the tails of the distribution are lower than estimated by KDE.

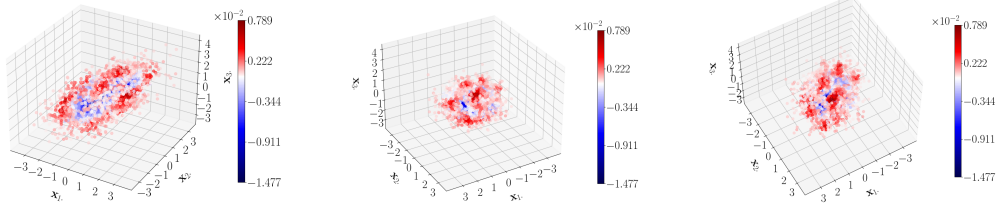
The squared errors of $S2_{PDF}$ for each data point over 100 simulation replications are illustrated by Figure 5.90 for the medium sample size and Figure 5.91 for the large sample size. Both illustrate that KDE have large errors around the mode in line with Figures 5.88 and 5.89.



(a) front angle (b) left angle (c) top angle

Figure 5.88 medium sample average PDF errors by KDE

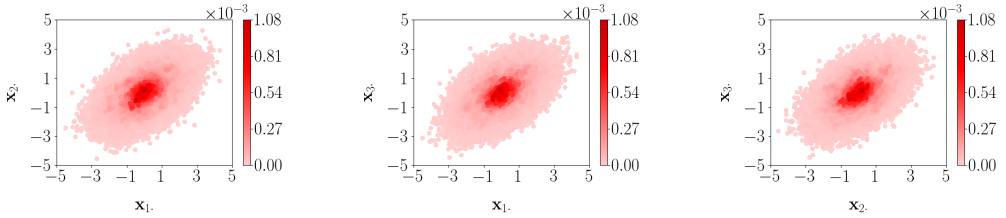
The difference between true and estimated PDF using KDE for mixed normal data. Differences are shown for the average estimate in terms of $L2_{PDF}$ loss values out of 100 simulation replications.



(a) front angle (b) left angle (c) top angle

Figure 5.89 large sample average PDF errors by KDE

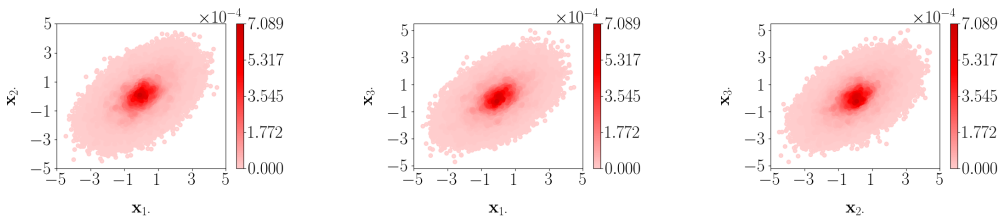
The difference between true and estimated PDF using KDE for mixed normal data. Differences are shown for the average estimate in terms of $L2_{PDF}$ loss values out of 100 simulation replications.



(a) V1 V2 (b) V1 V3 (c) V2 V3

Figure 5.90 medium sample $S2_{PDF}$ squared errors by KDE

Squared errors calculated using $S2_{PDF}$ differences for the medium sample size, using KDE with Scott's rule.



(a) V1 V2 (b) V1 V3 (c) V2 V3

Figure 5.91 large sample $S2_{PDF}$ squared errors by KDE

Squared errors calculated using $S2_{PDF}$ differences for the large sample size, using KDE with Scott's rule.

Hence the pattern of both methods are very different from each other. The proposed method has a pattern flowing from positive to negative areas with values close to 0 in between. The KDE obtains large $S2_{PDF}$ errors in the mode of the distribution. For the medium sample, model 5 with 3 hidden layers and 25 neurons is chosen to be the optimal model. For the large sample, model 4 with 3 hidden layers and 10 neurons is chosen as optimal though. As is seen from Tables 5.27 and 5.28, models with more than 10 neurons should be chosen as optimal. Hence model 5 also closely estimates the distribution for the large sample. These estimates are shown in Appendix I.

For this simulation case, the proposed method outperforms the KDE for both samples concerning the bivariate results. Regarding the trivariate results, the proposed method outperforms KDE only for specific models. The $L2_{\widetilde{CDF}}$ and $L2_{CDF}$ losses are similar across models though $L2_{PDF}$ losses are very different across models as follows from Table 5.28. This complicates the optimal model choice for the trivariate case.

6 Discussion

Non-parametric density estimation in a multivariate setting is challenging. Especially when dimensions are correlated. This paper proposes a new method to obtain the PDF to assess the properties of the underlying DGP without assuming any parametric distribution at fore hand, using neural networks. The procedure consists of two steps, CDF estimation by neural networks and PDF estimation by analytical derivatives. The first step consists of crucial components such as selecting an appropriate model for the neural network and constructing the empirical CDF as an approximation of the real but unknown CDF. Accuracy of empirical CDF approximation depends on several factors. The empirical CDF converges to the true CDF as more observations are used. Moreover, the empirical CDF becomes smoother as more grid values are used (for a fixed number of observations). Furthermore, the second step of the procedure, differentiating neural networks, is a very generic derivation. The algorithm can be applied to any model, irrespective of the number of input variables, hidden layers or hidden neurons. The only important condition in order to differentiate a certain model structure is the N-differentiability of the activation functions, where N is the number of input neurons, i.e. the dimension of the distribution. Therefore two sigmoid functions are taken into consideration, the hyperbolic tangent function and the logistic output function.

Our approach builds on the literature on CDF estimation using neural networks, see Magdon-Ismail and Atiya (2002). Magdon-Ismail and Atiya (2002) use a numerical differentiation scheme to obtain the PDF from the CDF output of the neural network for larger dimensions. We extend this literature by providing the analytical derivatives of the obtained CDF from any neural network. Our approach hence removes the approximation error in the second step of obtaining the PDF from the CDF output, leading to more accurate PDF estimates. We show that the proposed solution to obtain the PDF from the CDF output of a neural network holds in a multivariate setting and for an MLP with several hidden layers. Hence this solution holds for any neural network. Moreover, in line with not imposing any assumptions about the underlying DGP, correlation in the multivariate setting is dealt with. Next to continuous distributions also discrete distributions are treated.

Training neural networks is affected by the specification of many hyperparameters such as the model structure and the chosen activation function. Most hyperparameters can

be chosen intuitively according to the data properties, such as the activation function. Other parameters are selected by comparing the performance of different network structures, i.e. the number of hidden layers and neurons. Training neural networks is a computationally intensive task. To compare different neural network structures in a computational efficient way, it is proposed in this paper to combine them in one large neural network that is trained. In this way several neural network models, consisting of different hidden layers and neurons, are trained and compared simultaneously. The computation time to train this combined neural network is just as fast as training one network separately and thus reduces the computational time required for model selection considerably.

The performance of the proposed method is tested in a large simulation study. Both the univariate mixed normal (section 5.1) and univariate mixed GEV (section 5.2) simulation cases are compared with Trentin et al. (2018) and KDE. For the univariate simulation cases the proposed method estimates improve by using larger sample sizes. The approximation errors of the target CDF decrease and a smaller, more parsimonious, model is selected. Moreover, model selection is easier due to the more wide-spread CDF losses across models for larger samples. Under small sample sizes the CDF and PDF estimates are somewhat erratic. The reason for the relative volatile behaviour of the small sample PDF estimates are the large approximation errors of the target CDF in combination with the selection of too complex neural network models. For the mixed normal simulation case it appears that for the small sample, the approximation errors of the target CDF explain the erratic behaviour of the PDF estimates. In the case of the mixed GEV distribution, a too large neural network model is selected, which introduces more volatility in the small sample PDF estimates. For the proposed method, it is observed that model complexity decreases with the sample size. This is probably the result of a better approximation of the real CDF by the target CDF in case of large sample sizes. PDF estimates obtained with Trentin et al. (2018) also improve as the sample size improves. Erratic estimates for the small sample size are also observed with this method. Trentin et al. (2018) use larger models than the proposed method though. Moreover, their estimates depend on the initial bandwidth selection. For this method, there is a bias/variance trade-off introduced by specifying this initial bandwidth. By specifying a relatively larger bandwidth, the loss is smaller though the estimates are biased e.g. by introducing a non-symmetrical distribution around the mode in the case of the mixed normal distribution. When specifying a relatively smaller bandwidth, the loss is larger and the estimates are not biased but more volatile and not as smooth as the proposed method. Compared to Trentin et al. (2018) and KDE, the proposed method benefits more from using more observations, by selecting smaller models and obtaining smaller losses. These methods do not benefit as much from using larger sample sizes. KDE performs substantially worse in terms of loss values with obtaining largest errors around the modes of the distributions. Although estimates are more stable for Trentin et al. (2018), a bias-variance trade-off by specifying the initial bandwidth remains. Moreover, Trentin et al. (2018) results are often negative and have to be truncated to zero as negative probabilities do not exist. For the mixed normal simulation case, a third non-existent mode is estimated in between these distributions. The loss values obtained with Trentin et al. (2018) are 10 times larger compared to the proposed method. For the GEV simulation case, Trentin et al. (2018) outperform the proposed method for the small sample size though the proposed method outperforms Trentin et al. (2018) for the medium sample size. For all sample sizes, the losses obtained by KDE are 10 times larger than for the proposed method.

Both the bivariate mixed normal (section 5.4.1) and trivariate mixed normal (section 5.4.2) simulation cases are compared with KDE. Results obtained by Trentin et al. (2018) cannot be compared since variables are assumed to be uncorrelated under this method. As dimensions of the distribution increase, largest losses are found in the tails. For the bivariate simulation the proposed method outperforms KDE, since KDE obtains substantially larger losses for both samples. For the trivariate simulation case, KDE obtains larger losses for both samples when the proposed method uses sufficient number of neurons. In the large sample size, a model with only 10 neurons is selected which is outperformed by KDE. If a model with 25 neurons is used, the proposed method outperforms KDE in terms of PDF loss. PDF differences obtained by KDE are very different from the proposed method with largest errors again around the mode in line with the univariate estimates. The results obtained by the proposed method show stable estimates with a pattern for the PDF differences flowing from positive/negative to negative/positive loss areas with in between loss values close to 0.

Model selection is relative straightforward for univariate simulation cases, since differences between CDF losses across models increases with the sample size. For the multivariate simulation cases, however, model selection is less trivial, since differences between CDF losses across models remain small, even under large sample sizes. It is important for these cases to have sufficient number of neurons to cope with the larger probability space. For the univariate case not more than 15 neurons are needed though for the trivariate case at least 25 neurons are needed. Both for univariate and multivariate simulation cases, more hidden layers are needed when the distribution is more non-linear and contains more extremes.

The proposed method is also applied to two discrete distributions: the univariate mixed Poisson (section 5.3.1) and bivariate mixed Poisson (section 5.3.2). From this point of view, the proposed method is an extension of KDE and the method of Trentin et al. (2018), since the latter two cannot treat discrete distributions. For the univariate mixed Poisson the hyperbolic tangent function is used instead of the logistic function which is used for the bivariate mixed Poisson. The two sigmoid functions that are considered are different in terms of their gradients. The hyperbolic tangent function is more flexible since this function is centered around 0. Hence this function explores the loss function area faster. This simulation case has only a few discrete events which complicates the CDF estimation and therefore more training is required. This can be achieved by using more training epochs, a larger step size or by using the hyperbolic tangent. This case used the latter option. The bivariate simulation case uses the logistic activation function as more dimensions sensitives the loss function and therefore smaller steps for learning are required. For the univariate simulation case, we face similar approximation errors as we face for the univariate mixed normal and GEV simulation cases. The target CDF obtains approximation errors for the small sample which results in erratic estimates. For the larger sample sizes the approximation errors are substantially smaller. For all sample sizes, PMF estimates show large discrepancies in the left tail. This might be caused by a combination of factors. Continuous functions are used to approximate a discrete distribution. This might be complicated, particularly if the distribution is also highly skewed as in the case of Poisson distributions with a small mean, which is the case in this simulation setting. Large errors indeed occur in the left area of the Poisson distribution. This could be solved by using a different activation function as Zhang (2018) does or using models with an additional hidden layer in order to capture this highly non-linear behavior. Another option would be to use the logistic function with more training epochs instead of the hyperbolic tangent function but we leave this for further research.

For the bivariate simulation case similar losses are obtained for both the medium and large sample sizes. For the corresponding PMF differences a pattern flowing from positive/negative to negative/positive loss areas with in between loss values close to 0 are again found in line with the multivariate cases mentioned above.

The proposed approach for density estimation has many potential applications in statistics. Currently we explore the possibilities to use it as candidate distributions in particle or Bayesian filters to nowcast traffic intensity using volatile and noisy road sensor data with discrete correlated data points in order to predict traffic intensity. Next to this, there are many aspects that can extend the method developed in this paper. An unsolved issue is to incorporate a penalty in the loss function to enforce monotone increasing function properties for the CDF. A sensitivity analysis of the penalty used by Magdon-Ismail and Atiya (2002) revealed that this easily deteriorates the CDF estimates. Different model structures, activation functions, number of epochs and learning rate are closely related. Adjusting these would be interesting to use for density estimation as well.

List of variables

Symbol	Description
X	N × T matrix of input variables n=1, ..., N over time dimension t=1, ..., T
$x_{\cdot t}$	vector of input variables N at time t
$x_{n\cdot}$	vector of input variable n over time
γ	multidimensional evenly spaced grid to construct target variables
y	true CDF
$\hat{y}$	target variable which represents the empirical CDF
$\tilde{y}$	estimate of the empirical CDF
$h_t^{[q]}$	hidden neuron of layer $q = 1, \ldots, H$ containing M hidden neurons at time t
$b^{[q]}$	bias of layer q containing connections to M hidden neurons
$W^{[q]}$	weight matrix of layer $q = 1, \ldots, H$
$w_{ij}^{[q]}$	entry of weight matrix of layer $q = 1, \ldots, H$ from neuron $i = 1, \ldots, I$ from the former layer to neuron $j = 1, \ldots, J$ of the subsequent layer

List of functions

Symbol	Description
$f(\cdot)$	activation function
$\tau(\cdot)$	hyperbolic tangent activation function
$\sigma(\cdot)$	logistic activation function
$g^{[q]}(\cdot)$	linear function, connecting the weights and input variables or hidden neurons together with an optional bias

Appendix A Model selection

The capacity of a neural network refers to the ability to fit the target variable correctly. This can be increased by several parameters. Especially, the number of epochs and the model of the neural network play an important role. To select the best neural network structure (or model), several models are trained and compared. To improve computational efficiency, the different models that are compared in the model selection stage are combined in one large MLP. Thus one MLP, consisting of several subMLPs, is trained. Each subMLP is a potential model to be considered in the model selection process. For each subMLP a value for the L2 loss is obtained. Note that other hyperparameters such as activation function, learning rate, weight initialization, etc. are specified similarly for each subMLP. The more layers and neurons the MLP contains, the larger the complexity of obtaining the estimate can be.

There is a substantial difference between setting the number of hidden layers or the number of hidden neurons. Increasing the number of hidden layers is mainly done when the distribution of the target variable is non-linear. The more hidden layers, the more activation functions are nested within each other, the more non-linear and complex the neural network. There are also several reasons for increasing the number of hidden neurons. Hidden neurons specify the 'flat' capacity of the neural network opposed to the 'deep' capacity that hidden layers create. Note that without enough neurons in each layer, a more complex distribution, e.g. a multinomial distribution, is impossible to fit. Hence without enough neurons, increasing the number of hidden layers does not create enough capacity. However, if the capacity is larger than needed, estimates are more volatile than desired. Especially, including too many hidden layers results in a very erratic empirical CDF, as has been shown in section 5. By taking into account this intuition, the potential subMLPs to include in the 'large' MLP can be decided upon.

Figure A.1 shows an example of an MLP with k subMLPs. The k distinct models result in k different error values $L2_{\widehat{CDF}}$ defined in (9). The MLP with the lowest error value is chosen to be used for derivation of the corresponding PDF.

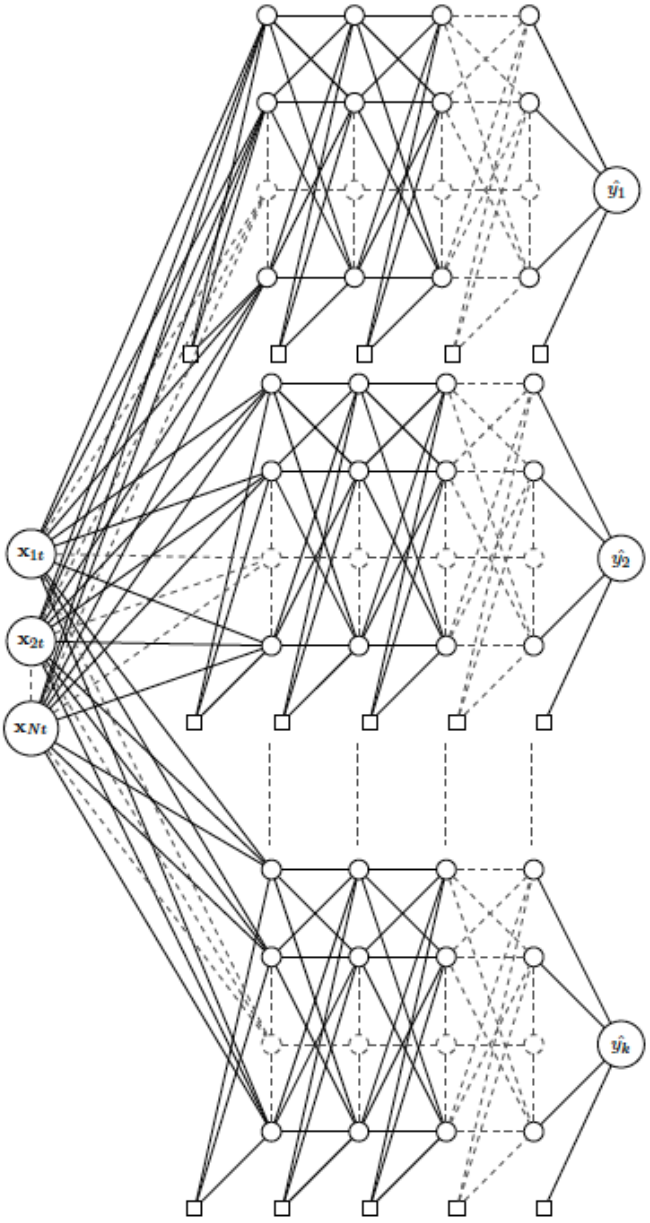


Figure A.1 Architecture of one MLP consisting of several subMLPs

B Differentiation of sigmoid functions

Two sigmoid functions are considered, the logistic output function denoted by $\sigma(\cdot)$ and the hyperbolic tangent function $\tau(\cdot)$. These functions receive as inputs either a linear combination of input variables $x_{\cdot t}$ together with weights (and biases) or a linear combination of previous layers $h_{\cdot t}^{[q]}$ together with weights (and biases), see below:

$$\sigma(\mathbf{h}_{\cdot t}^{[q]}) = \frac{1}{1 + e^{-[\mathbf{W}^{[q+1]'}\mathbf{h}_{\cdot t}^{[q]} + \mathbf{b}_t^{[q+1]}]}}$$

$$\tau(\mathbf{h}_{\cdot t}^{[q]}) = tanh(\mathbf{W}^{[q+1]'}\mathbf{h}_{\cdot t}^{[q]} + \mathbf{b}_t^{[q+1]})$$

Note that $\sigma(\mathbf{h}_{.t}^{[q]}) : \mathbb{R}^M \to \mathbb{R}^M$ and $\tau(\mathbf{h}_{.t}^{[q]}) : \mathbb{R}^M \to \mathbb{R}^M$. The first layer would take input variables as input. Then $\sigma(\mathbf{x}_{.t}) : \mathbb{R}^N \to \mathbb{R}^M$ and $\tau(\mathbf{x}_{.t}) : \mathbb{R}^N \to \mathbb{R}^M$ are operations that map $\mathbf{x}_{.t}$ by a linear combination to an M-vector where M is the number of the neurons of the first hidden layer. It is also possible to impose an activation function on the last input layer. Then $\sigma(\mathbf{h}_{.t}^{[H]}) : \mathbb{R}^M \to \mathbb{R}^1$ and $\tau(\mathbf{h}_{.t}^{[H]}) : \mathbb{R}^M \to \mathbb{R}^1$ are operations that map $\mathbf{h}_{.t}^{[H]}$ by a linear combination to the output layer that consists of only 1 neuron. In the simulation study no activation functions are imposed on the last layer under the proposed method.

The advantage of the logistic and hyperbolic tangent function is the N-differentiability. See Minai and Williams (1993) for the derivations of the N^{th} derivative of the logistic function:

$$\sigma^{(n)} = \sum_{k=1}^{n} (-1)^{k-1} A_{n,k-1} \sigma^k (1-\sigma)^{n+1-k} \tag{B.1}$$

$$A_{n,k-1} = \sum_{l=0}^{k} (-1)^l \binom{n+1}{l} (k-l)^n$$

$A_{n,k-1}$ gives the formula for the Eulerian number. See Boyadzhiev (2009) for the derivations of the N^{th} derivative of the hyperbolic tangent function:

$$\tau^{(n)} = (-2)^n (\tau+1) \sum_{k=0}^{n} \frac{k!}{2^k} \left\{ {n \atop k} \right\} (\tau-1)^k \tag{B.2}$$

$$\left\{ {n \atop k} \right\} = \frac{1}{k!} \sum_{l=0}^{k} (-1)^l \binom{k}{l} (k-l)^n$$

$\left\{ {n \atop k} \right\}$ are the Stirling numbers of the second kind. This is incorporated similarly as for $\sigma(\cdot)$ explained in the methodology section 4.3.

There is a crucial difference between the logistic function and the hyperbolic tangent function though. The hyperbolic tangent function is centered around 0 whereas the logistic function is centered around $\frac{1}{2}$. This results in differences in the value of the gradient of the activation function which decides upon the speed of training. Activation functions that are centered around 0, result in faster convergence of the algorithm. The output of each hidden layer will be centered around 0, hence some positive values and some negative values. When the gradient needs to change direction, learning is easier than when all output values are either positive or negative. Hence the hyperbolic tangent function explores more areas of the loss function than the logistic function does. This can result in obtaining the global minimum which a logistic function does not reach due to the starting value. However this could potentially also result in skipping this global minimum because of greater gradients.

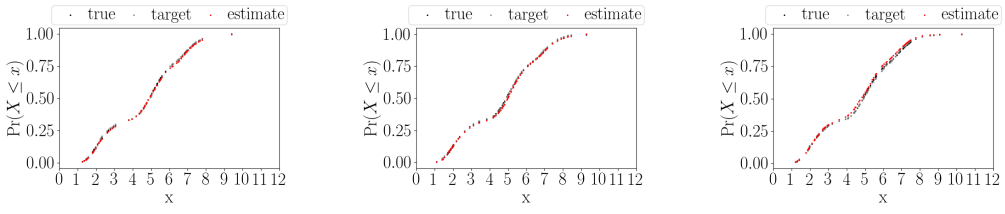
C GEV small and medium sample results using the hyperbolic tangent function

Results are shown for the small sample size and the medium sample size, using similar settings as in section 5.2 but using the hyperbolic tangent function instead of the logistic function as activation function.

Small sample results:

Table C.1 presents the small sample results for the validation sample and test sample. Models 4, 5, and 7 are the best performing models obtaining the smallest $L2_{\widetilde{CDF}}$ loss for the validation sample. Model 4 performs substantially better than the other models based on $L2_{PDF}$ loss. Moreover, even models 1 and 2 perform better than models 5 and 7 in terms of $L2_{PDF}$ loss. The $L2_{PDF}$ loss values are particularly small for models with the least number of layers (1 and 2) and neurons (5 and 9). Hence a model using 3 layers or 15 neurons shows probably bi-modal behavior around modes in the simulation distribution. Model 7 attains the smallest $L2_{\widetilde{CDF}}$ loss for the validation sample and is further analyzed.

Figures C.1 and C.2 present the best, average, and worst CDF and PDF estimates for the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 7. The CDF estimates follow the shape of the true CDF proportionally. The corresponding PDF estimates show that these minor differences do not result in tail estimation error. However, large $L2_{PDF}$ losses are found around the modes. Either the first or second modes are bi-modal. This extent of non-linearity in the modes is not expected when analyzing the CDF estimates. This can either be due to the small sample size or a too large model. Using a smaller model, like model 4, will prevent this potential volatile behavior. Estimates using model 4 are shown in Appendix D. Hence these conclusions are in line with the neural network models that use the logistic function as activation function.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

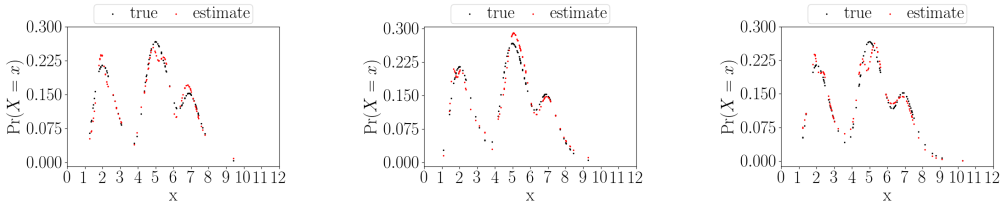
Figure C.1 small sample CDF estimates using model 7
True, target, and estimated CDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 5 neurons are given.

The squared errors, $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 7 are provided in Figure C.3. The $S2_{CDF}$ and $S2_{\widetilde{CDF}}$ errors show that the largest errors occur around the modes. The $S2_{PDF}$ errors show small errors for the third mode compared to the first and second modes.

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample with $L2_{\widehat{CDF}}$= 7.400 (0.474)										
$L2_{\widetilde{CDF}}$	1.122	3.662	3.971	6.131	1.479	1.699	5.758	1.351	3.071	2.867
	(0.021)	(0.119)	(0.218)	(0.518)	(0.048)	(0.103)	(2.510)	(0.038)	(1.162)	(0.176)
$L2_{CDF}$	7.315	8.490	8.647	10.648	7.140	7.912	11.654	7.537	9.507	9.607
	(0.490)	(0.487)	(0.541)	(0.656)	(0.510)	(0.514)	(2.261)	(0.545)	(1.259)	(0.528)
$L2_{PDF}$	56.749	34.251	34.712	50.304	29.690	47.780	58.315	48.555	139.691	125.166
	(4.081)	(1.261)	(1.632)	(1.381)	(1.584)	(2.225)	(2.568)	(2.347)	(57.472)	(3.779)
Test sample with $L2_{\widehat{CDF}}$= 5.873 (0.283)										
$L2_{\widetilde{CDF}}$	1.708	6.691	6.122	5.524	3.622	3.573	4.442	3.278	3.229	3.279
	(0.264)	(0.539)	(0.390)	(0.280)	(0.216)	(0.282)	(0.390)	(0.287)	(0.278)	(0.306)
$L2_{CDF}$	6.719	11.552	10.192	9.680	8.854	9.023	9.705	8.862	9.047	9.060
	(0.723)	(0.973)	(0.711)	(0.645)	(0.676)	(0.682)	(0.766)	(0.737)	(0.668)	(0.669)
$L2_{PDF}$	82.248	33.284	31.299	42.081	30.684	48.736	67.856	54.769	91.370	125.516
	(5.097)	(1.474)	(1.594)	(1.969)	(1.503)	(3.340)	(9.276)	(3.510)	(4.944)	(5.408)
Values in the table scaled by 10^{-3}										

Table C.1 small sample results of mixed GEV data

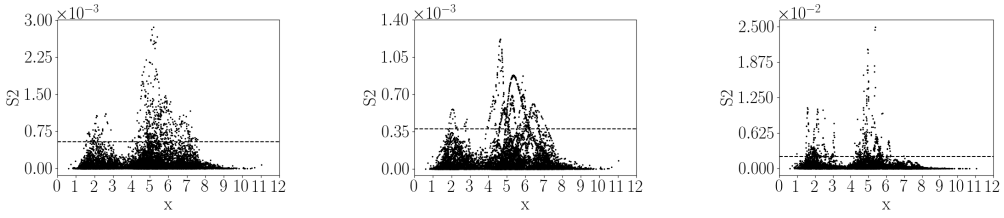
L2 losses for mixed GEV data, small sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.



(a) smallest $L2_{\widetilde{CDF}}$loss (b) average $L2_{\widetilde{CDF}}$loss (c) largest $L2_{\widetilde{CDF}}$loss

Figure C.2 small sample PDF estimates using model 7

True and estimated PDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 3 hidden layers and 5 neurons are given.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

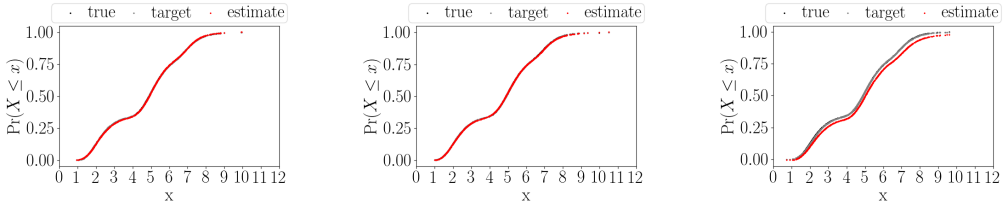
Figure C.3 small sample squared errors using model 7

Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the small sample size, using 3 hidden layers and 5 neurons of the proposed method.

Medium sample results:

Table C.2 presents the medium sample results for the validation sample and test sample. Models 4, 5 and 7 obtain again the smallest $L2_{\widetilde{CDF}}$ loss for the validation sample. In this larger sample the corresponding $L2_{CDF}$ and $L2_{PDF}$ losses are smallest for model 7. In the corresponding losses for the test sample, the standard errors of $L2_{\widetilde{CDF}}$ and $L2_{CDF}$ losses are very large for model 7 opposed to all smaller models. This is further analyzed using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ errors. Note that the $L2_{PDF}$ loss for the small models using only 1 hidden layer are substantially larger than other models. It seems that opposed to the smaller sample, smaller models using only 1 hidden layer have a larger estimation loss than larger models which use 2 or 3 hidden layers.

Figures C.4 and C.5 present the best, median, and worst CDF and PDF estimates for the test sample in terms of $L2_{\widetilde{CDF}}$ loss using model 7. As indicated by Table C.2, the $L2_{\widetilde{CDF}}$ loss of model 7 is very large compared to other models and the model itself of the validation sample. The CDF estimates by model 7 vary considerably with a standard error of $10.794 \cdot 10^{-3}$. The mean value of $L2_{\widetilde{CDF}}$ loss is $16.669 \cdot 10^{-3}$ opposed to the corresponding median value of $3.392 \cdot 10^{-3}$. Hence the CDF estimates are quite skewed. Therefore instead of the mean, the median CDF and PDF estimates are shown in Figures C.4(b) and C.5(b). The best and median CDF estimates, depicted in Figures C.4(a) and C.4(b), follow the shape of the true CDF closely. However, the worst CDF estimate illustrated by Figure C.4(c) is offset by a small value. Nevertheless, the estimate follows the shape of the CDF closely. The corresponding PDF estimates for the best and median estimates, shown in Figures C.5(a) and C.5(b), follow the tails and modes of the distribution closely. The worst PDF estimate shown in Figure C.5(c) still follows the shape of the distribution proportionally, as the CDF estimate indicated as well. However this is again offset by a small value.



(a) smallest $L2_{\widetilde{CDF}}$loss (b) average $L2_{\widetilde{CDF}}$loss (c) largest $L2_{\widetilde{CDF}}$loss

Figure C.4 medium sample CDF estimates using model 7
True, target, and estimated CDF using the proposed method for mixed GEV data. The best, median, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 5 neurons are given.

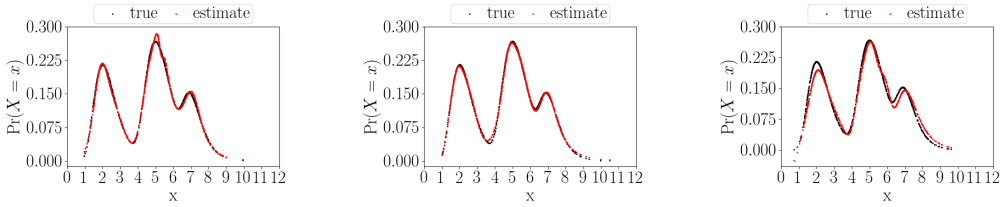
The squared errors, $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using model 7 are provided in Figure C.6. The $S2_{\widetilde{CDF}}$ and $S2_{CDF}$ errors of model 7 depicted in Figures C.6(a) and C.6(b) show that one simulation replication considerably misfits the target CDF compared to all other simulation replications. This outlier is the worst CDF estimate depicted in Figure C.4(c). This also explains the skewed $L2_{\widetilde{CDF}}$ loss value and large standard error in the test sample. The $S2_{PDF}$ errors of model 7 illustrated by Figure C.6(c) show largest errors around the first and second mode in line with the small sample results.

For the small sample size, model 7 is used with bi-modal behavior around modes.

model	optimal	1	2	3	4	5	6	7	8	9
Validation sample with $L2_{\widehat{CDF}}$= 7.966 (0.513)										
$L2_{\widetilde{CDF}}$	2.220	16.794	27.892	33.136	4.511	5.522	17.216	3.410	7.777	13.682
	(0.084)	(0.413)	(0.914)	(0.897)	(0.245)	(1.250)	(0.693)	(0.284)	(1.978)	(3.727)
$L2_{CDF}$	7.966	22.745	32.576	37.630	10.200	11.665	23.257	9.786	14.664	21.036
	(0.499)	(0.797)	(1.073)	(1.112)	(0.510)	(1.248)	(6.664)	(0.591)	(2.260)	(3.945)
$L2_{PDF}$	42.784	142.646	217.006	323.809	41.133	44.463	48.291	39.327	86.625	99.771
	(1.782)	(1.386)	(3.112)	(3.240)	(1.496)	(1.765)	(3.348)	(1.729)	(31.942)	(8.308)
Test sample with $L2_{\widehat{CDF}}$= 6.484 (0.434)										
$L2_{\widetilde{CDF}}$	1.982	29.942	15.093	33.109	4.804	6.341	9.914	16.669	7.804	21.649
	(0.657)	(0.797)	(0.640)	(1.474)	(0.453)	(0.936)	(1.413)	(10.794)	(1.711)	(14.103)
$L2_{CDF}$	7.297	37.244	15.093	38.865	9.967	10.604	13.955	22.587	12.555	27.405
	(0.771)	(1.090)	(0.756)	(1.730)	(0.809)	(0.889)	(1.415)	(11.478)	(1.948)	(15.377)
$L2_{PDF}$	56.955	190.118	137.908	343.466	30.776	44.725	51.330	48.924	48.846	97.976
	(2.198)	(1.733)	(1.424)	(3.732)	(1.404)	(2.119)	(2.752)	(2.767)	(3.466)	(7.525)
Values in the table scaled by 10^{-3}										

Table C.2 medium sample results of mixed GEV data

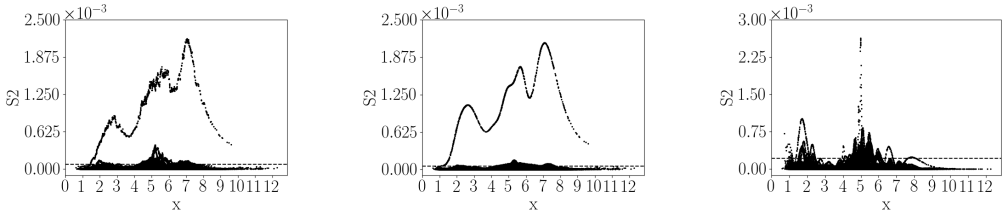
L2 losses for mixed GEV data, medium sample. The Table reports the mean (standard error in parentheses) of the L2 losses using the differences between the true, target, and estimated CDF and between the true and estimated PDF obtained by the proposed method for the validation and test samples.



(a) smallest $L2_{\widehat{CDF}}$ loss (b) average $L2_{\widehat{CDF}}$ loss (c) largest $L2_{\widehat{CDF}}$ loss

Figure C.5 medium sample PDF estimates using model 7

True and estimated PDF using the proposed method for mixed GEV data. The best, median, and worst estimates in terms of $L2_{\widehat{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 5 neurons are given.



(a) $S2_{\widehat{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure C.6 medium sample squared errors using model 7

Squared errors calculated using the $S2_{\widehat{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the medium sample size, using 3 hidden layers and 5 neurons of the proposed method.

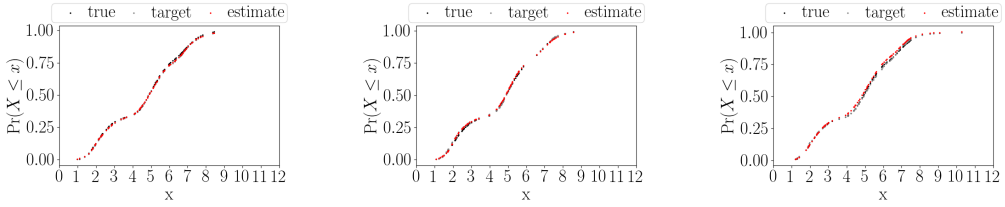
However, model 7 estimates the simulation distribution closely for the medium sample. Model 4 with 2 hidden layers and 5 neurons follows the distribution closely for both the small and medium sample size, as can be seen from Appendix D.

D GEV small and medium sample results for model 4 using the hyperbolic tanget function

Instead of using model 7, based on the $L2_{\widetilde{CDF}}$ losses of Tables C.1 and C.2, results of model 4 for the small and medium samples are shown in this section.

Model 4 with 2 hidden layers and 5 neurons follows model 7 with 3 hidden layers and 5 neurons with $0.13 \cdot 10^{-03}$ $L2_{\widetilde{CDF}}$ loss difference for the small sample size, see Table C.1. Moreover, this model has the smallest $L2_{CDF}$ and $L2_{PDF}$ loss values. Most importantly, this model has only 2 hidden layers opposed to 3 hidden layers. Therefore this model introduces less non-linearity in the PDF estimates.

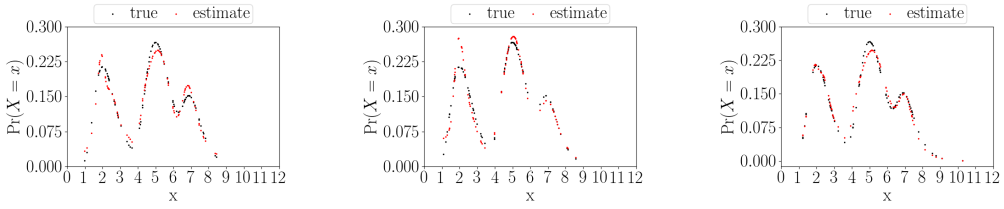
Figures D.1 and D.2 show the best, average, and worst CDF and PDF estimates for the test sample in terms of $L2_{\widetilde{CDF}}$ using model 4 instead of model 7. The CDF estimates are not very different from the estimates by model 7 depicted in Figure C.1. However, the PDF estimates by model 4 do not show any bi-modal behavior around modes opposed to model 7. This shows the importance of MLP selection for small sample sizes. Using too many hidden layers, results in PDF estimates that are bi-modal around modes. In this Figure the tails are again closely estimated and the modes are not bi-modal though still slightly volatile.



(a) smallest $L2_{\widetilde{CDF}}$ loss (b) average $L2_{\widetilde{CDF}}$ loss (c) largest $L2_{\widetilde{CDF}}$ loss

Figure D.1 small sample CDF estimates using model 4

True, target and estimated CDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 5 neurons are given.

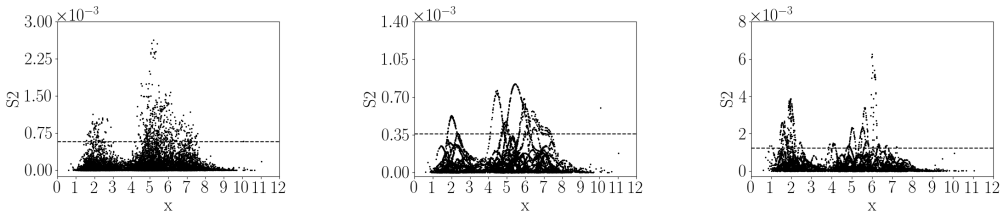


(a) smallest $L2_{\widetilde{CDF}}$loss (b) average $L2_{\widetilde{CDF}}$loss (c) largest $L2_{\widetilde{CDF}}$loss

Figure D.2 small sample PDF estimates using model 4

True and estimated PDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the small sample size using 2 hidden layers and 5 neurons are given.

The squared errors $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ for each data point over 100 simulation replications using 3 hidden layers and 2 hidden layers are provided Figures C.3 and D.3. These Figures show similar $S2_{CDF}$ and $S2_{\widetilde{CDF}}$ errors. Under both models the largest errors occur around the modes of the distribution. The size of the errors are equal for both models. This again confirms that even though both models are capable of training the mixed multimodal distribution, too large models do not provide any additional gain. The corresponding differences in the $L2_{PDF}$ loss depicted in Figures C.3(c) and D.3(c) also show similar dynamics, with small error values for the third mode where the $S2_{PDF}$ errors using model 7 are 3 times larger than the $S2_{PDF}$ errors using model 4. This implies that the $L2_{PDF}$ loss values using 2 hidden layers opposed to using 3 hidden layers is substantially smaller in value. Hence, even though training the CDF is equally well done for models 4 and 7, the third hidden layer adds too much non-linearity to the functional form of the CDF such that the corresponding PDF estimates are bi-modal around modes using model 7.

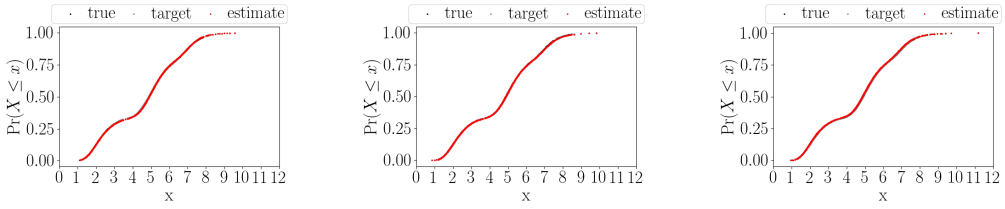


(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure D.3 small sample squared errors using model 4

Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the small sample size, using 2 hidden layers and 5 neurons using the proposed method.

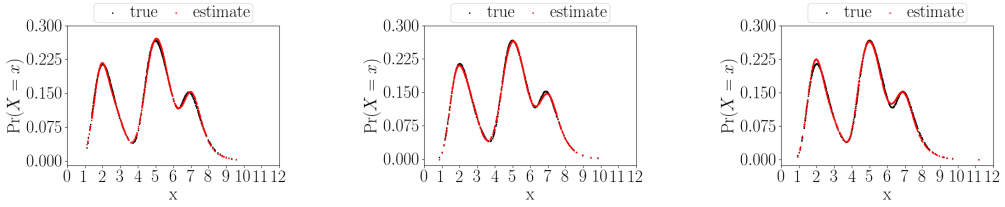
See Table C.2 for the medium sample results, model 4 performs better and more consistently in the test sample than model 7, just like for the small sample size. As can be retrieved from the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ errors illustrated by Figure C.6 using model 7, there is one outlier that affects the means and standard errors of the $L2_{\widetilde{CDF}}$, $L2_{CDF}$ and $L2_{PDF}$ losses. The best, average and worst CDF and PDF estimates using model 4 are shown in Figures D.4 and D.5. The CDF and PDF estimates follow the shape of the distribution very closely and consistently from the best to the worst estimates.



(a) smallest $L2_{\widetilde{CDF}}$loss (b) average $L2_{\widetilde{CDF}}$loss (c) largest $L2_{\widetilde{CDF}}$loss

Figure D.4 medium sample CDF estimates using model 4

True, target, and estimated CDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 5 neurons are given.

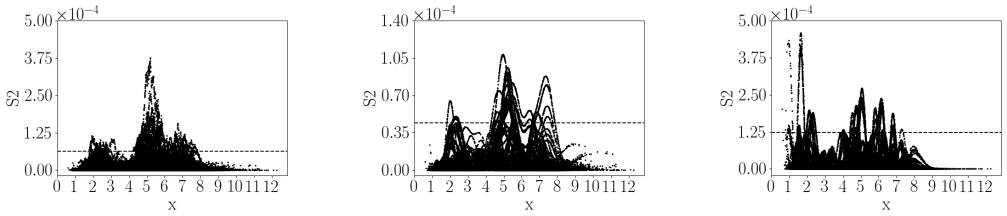


(a) smallest $L2_{\widetilde{CDF}}$loss (b) average $L2_{\widetilde{CDF}}$loss (c) largest $L2_{\widetilde{CDF}}$loss

Figure D.5 medium sample PDF estimates using model 4

True and estimated PDF using the proposed method for mixed GEV data. The best, average, and worst estimates in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 5 neurons are given.

The squared errors $S2_{\widetilde{CDF}}$ and $S2_{CDF}$ errors using model 4, illustrated by Figures D.6(a) and D.6(b), are as large as the errors using model 7, except for its outlier depicted in Figure C.6. This is also in line with the CDF estimates shown in Figures C.4 and D.4 that are similar. The corresponding $S2_{PDF}$ errors depicted in Figure D.6(c) show large error values around the first, second and third mode as well as the dip between the second and third mode. Model 7 does not have large errors for this dip nor the third mode. Hence even though model 7 shows bi-modal behavior around modes in the distribution for the small sample size, this does not occur anymore for the medium sample size. Hence both models 4 and 7 estimate this distribution for the medium sample equally well, with similar $S2_{PDF}$ errors for the medium sample.



(a) $S2_{\widetilde{CDF}}$ (b) $S2_{CDF}$ (c) $S2_{PDF}$

Figure D.6 medium sample squared errors using model 4
Squared errors calculated using the $S2_{\widetilde{CDF}}$, $S2_{CDF}$, and $S2_{PDF}$ differences together with the 95^{th} percentile for the medium sample size, using 2 hidden layers and 5 neurons of the proposed method.

All in all, the small sample results have the smallest errors using model 4 and the medium sample has smallest errors for both models 4 and 7. Hence for small sample sizes, model selection is crucial due to the small number of observations.

E Sensitivity Analysis of the loss function including penalty term

In this appendix, a sensitivity analysis of adding a monotonicity penalty to the L2 loss function defined in equation (9) is given. The L2 loss function (9) can be extended with a monotonicity penalty as Magdon-Ismail and Atiya (2002) do:

$$\lambda \sum_{t=1}^{T} [\tilde{y}_t(z_{.t}) - \tilde{y}_t(z_{.t} + \Delta)]^2 \, I(\tilde{y}_t(z_{.t}) - \tilde{y}_t(z_{.t} + \Delta) \geq 0)$$

where λ is the penalty parameter, $z_{.t}$ are points of the CDF where monotonicity is enforced and Δ is a small constant value. $I(\cdot)$ is the indicator function which is 1 when the condition inside the brackets is satisfied and 0 otherwise. The loss function consists of two parts, the L2 loss and the penalty loss.

$$L = \sum_{t=1}^{T} [\tilde{y}_t(x_{.t}) - \hat{y}_t(x_{.t})]^2 \tag{E.1}$$

$$+ \lambda \sum_{t=1}^{T} [\tilde{y}_t(z_{.t}) - \tilde{y}_t(z_{.t} + \Delta)]^2 \, I(\tilde{y}_t(z_{.t}) - \tilde{y}_t(z_{.t} + \Delta) \geq 0)$$

$$\equiv \text{L2} + \text{penalty}$$

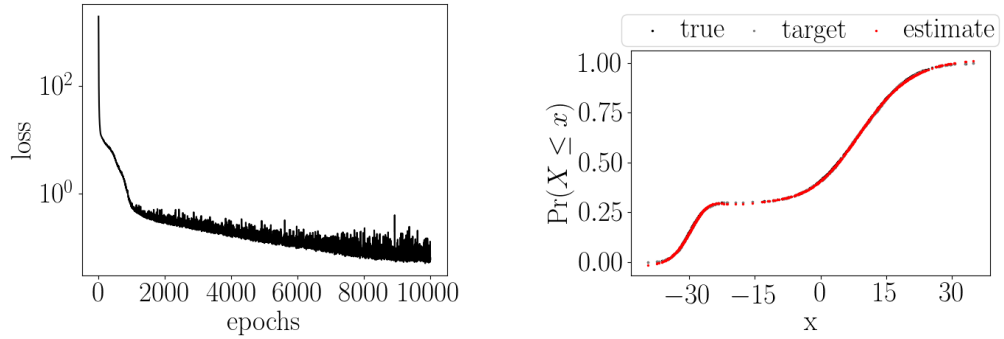
Magdon-Ismail and Atiya (2002) use loss function (E.1) where they enforce penalty λ only at some points $z_{.t}$. In this way the monotone increasing function property of a CDF remains preserved. In this section a sensitivity analysis about incorporating the above monotonicity penalty to the L2 loss function defined in equation (9) is conducted. Opposed to Magdon-Ismail and Atiya (2002), monotonicity is forced at *each* input

variable $x_{\cdot t}$ at time t:

$$L = \sum_{t=1}^{T} [\tilde{y}_t(x_{\cdot t}) - \hat{y}_t(x_{\cdot t})]^2 \tag{E.2}$$
$$+ \lambda \sum_{t=1}^{T} [\tilde{y}_t(x_{\cdot t}) - \tilde{y}_t(x_{\cdot t} + \Delta)]^2 I(\tilde{y}_t(x_{\cdot t}) - \tilde{y}_t(x_{\cdot t} + \Delta) \geq 0)$$
$$\equiv \text{L2} + \text{penalty}$$

This should not have a large impact since the penalty is zero at points where monotonicity does not need to be enforced. Furthermore, instead of using very small distances Δ between input variables, the differences of subsequent input variables are used. This alleviates the computational burden that the penalty brings to the loss function. The method is applied to the simulations of the mixed normal distribution.

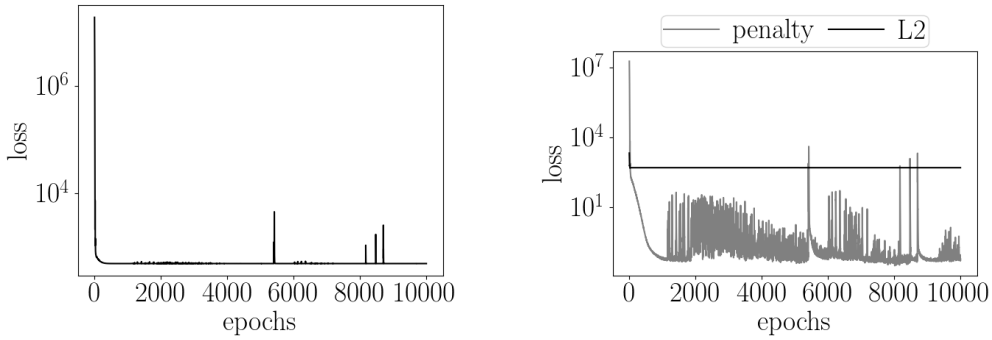
The aim of this sensitivity analysis is to find the optimal value for the penalty parameter λ in loss function (E.2). First, the course of the loss function without a penalty is plotted in Figure E.1(a). The loss function keeps decreasing during 10,000 epochs. The corresponding estimate is given in Figure E.1(b). The difference between estimate, target and true CDF is negligible. In the remainder of this appendix, this model fit is further referred to as the benchmark model.



(a) loss (b) CDF

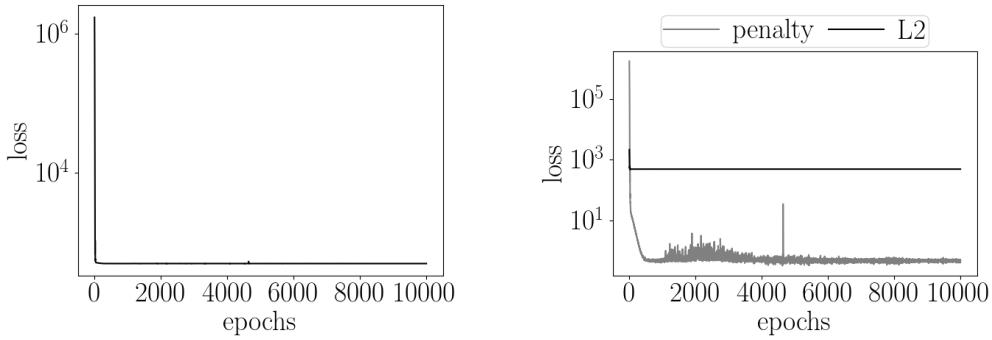
Figure E.1 The course of the L2 loss function without penalty together with the corresponding estimate of the CDF.

A range of penalty parameters is considered, with the largest parameter value of 10,000 in line with Magdon-Ismail and Atiya (2002) and the smallest parameter value 0.1. The course of the loss function and the respective parts, the L2 loss and the penalty loss, for penalty parameters 10,000, 1000, 100 and 10 are shown in Figures E.2, E.3, E.4 and E.5. The loss functions converge before the 1000^{th} epoch opposed to the benchmark loss function that continues decreasing for all 10,000 epochs. The value of the loss function after convergence is also substantially larger compared to the value of the benchmark loss function. The L2 loss converges almost immediately and the penalty loss converges around the 2000^{th} epoch. For the smaller penalty parameters of 100 and 10 the penalty loss is less erratic compared to the larger penalties 10,000 and 1000.



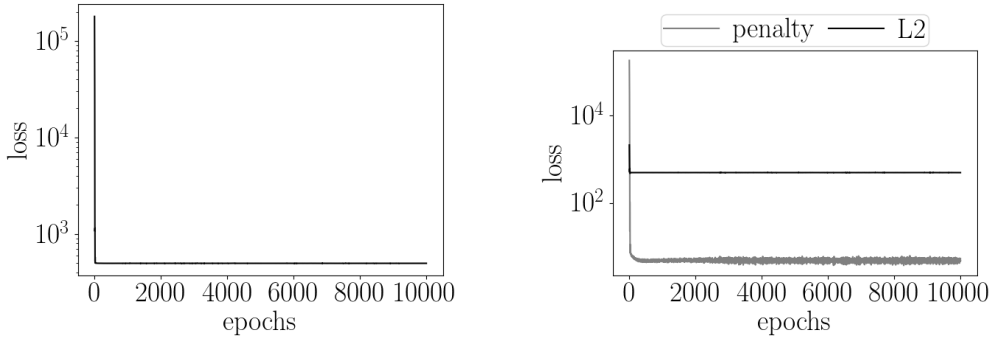
(a) loss (b) loss components

Figure E.2 The loss function with its components, the L2 loss and the penalty loss with penalty parameter value 10,000.



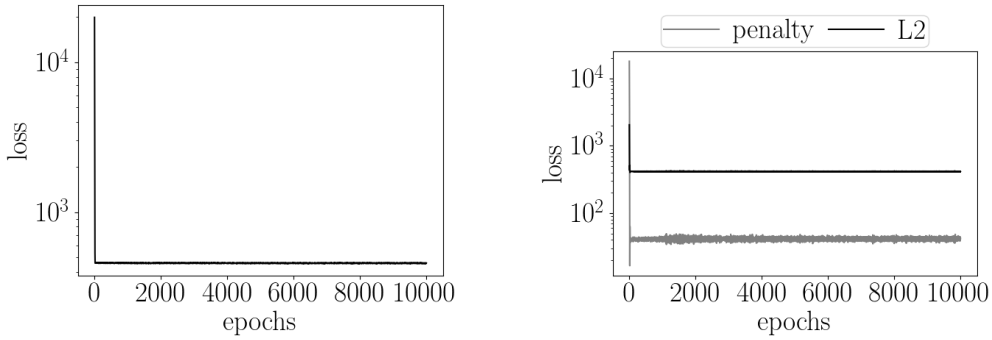
(a) loss (b) loss components

Figure E.3 The loss function with its components, the L2 loss and the penalty loss with penalty parameter value 1000.



(a) loss (b) loss components

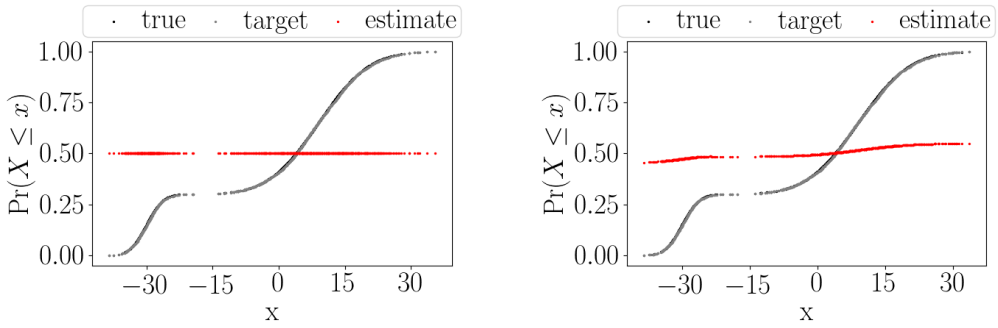
Figure E.4 The loss function with its components, the L2 loss and the penalty loss with penalty parameter value 100.



(a) loss (b) loss components

Figure E.5 The loss function with its components, the L2 loss and the penalty loss with penalty parameter value 10.

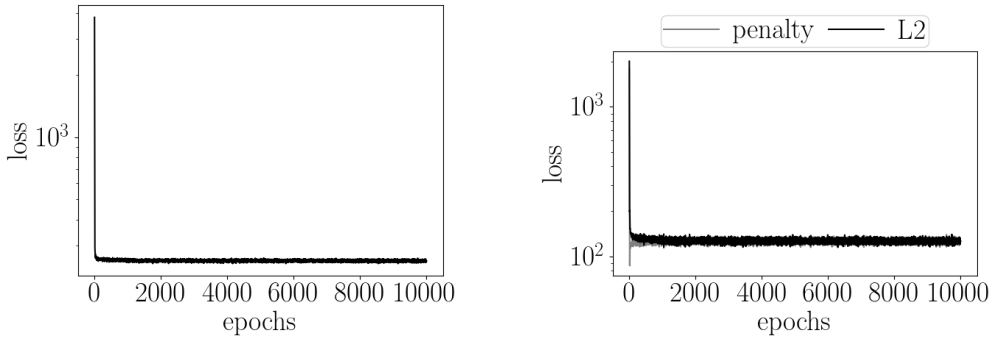
The corresponding CDF estimates are similar for all penalty parameters larger than 10, see Figure E.6(a). The CDF difference between the estimate and the target CDF are substantial. The loss function converges quite early, which implies that learning stops immediately which explains the straight horizontal estimate. As a result the estimate completely misfits the target and true CDF. For penalty parameter value 10, the difference between the L2 loss and penalty loss is smaller compared to the other penalty parameters, see Figure E.5(b). Figure E.6(b) shows that the corresponding CDF estimate slightly improves.



(a) penalty > 10 (b) penalty 10

Figure E.6 True, target and fitted CDF using the proposed method for data with a mixed normal distribution, 1 simulation replication, with a sample size of 5 000.

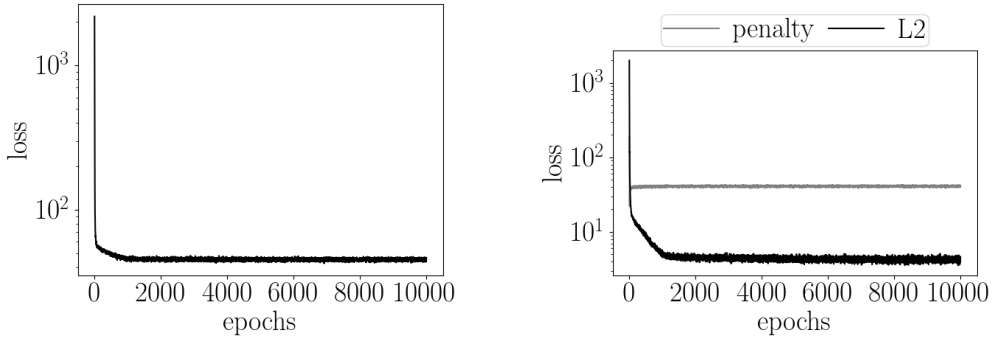
Figure E.7 shows the loss function using penalty parameter value 1 with respective parts L2 loss and penalty loss that are equal. The corresponding estimate is improved which is illustrated in Figure E.9(a).



(a) loss (b) loss components

Figure E.7 The loss function with its components, the L2 loss and the penalty loss with penalty parameter value 1.

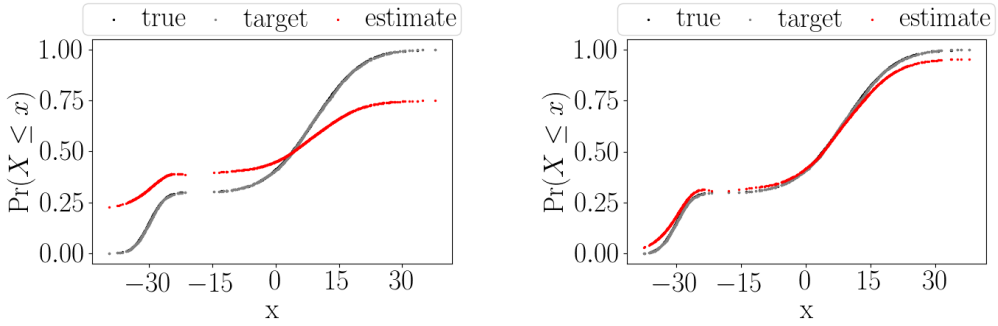
Figure E.8 shows the loss function using penalty parameter value 0.1. The L2 loss is smaller than the penalty loss. Hence the difference between the respective parts is larger as for the larger penalty parameters but now the L2 loss is smaller than the penalty loss. This estimate improves compared to penalty parameter value 1 where the L2 loss and the penalty loss are equal.



(a) loss (b) loss components

Figure E.8 The loss function with its components, the L2 loss and the penalty loss with penalty parameter value 0.1.

Careful inspection of figures E.9(a) and E.9(b) learns that the monotonicity property is still violated around -25. This would point to the fact that the penalty parameter value is too low. However as explained, increasing the penalty parameter value results in underfitting. The loss function without a penalty, shown in Figure E.1, keeps decreasing over 10,000 epochs opposed to all other loss functions using a penalty. It seems that the penalty is activated in early stage of the training which affects the loss function by decreasing into the incorrect direction. In this way, the loss function converges quite early and does not decrease over the course of 10,000 epochs.



(a) penalty 1 (b) penalty 0.1

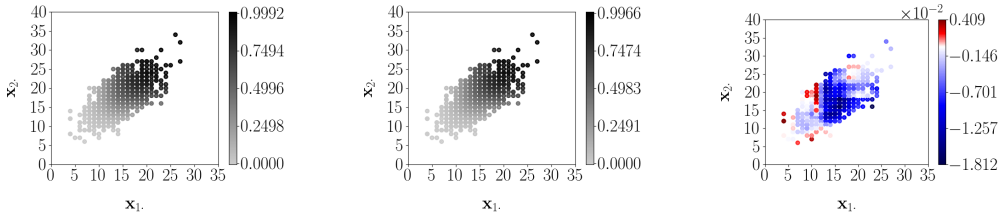
Figure E.9 True, target and fitted CDF using the proposed method for data with a mixed normal distribution, 1 simulation replication, with a sample size of 5 000.

For all penalties, learning stops quite early over the course of epochs on the contrary to a loss function without a penalty defined in equation (9). This potentially refers to incorporating too much regularization which prevents the neural network from learning. The penalty parameter λ determines how much regularization is imposed, see Goodfellow et al. (2016), Chapter 7. A too large value results in too much regularization, though a too small value results in insufficient amount of regularization. For the larger parameters 10,000, 1000, 100 and 10, the CDF estimates are flat. This refers to too much regularization imposed by a too large penalty parameter. For penalty parameter value 10, the CDF estimate improves slightly which implies that the parameter value is still quite large. For penalty parameters 1 and 0.1, the CDF estimates improve substantially though the monotone sequence properties are violated.

Obviously, tuning the penalty parameter is quite complex. Therefore the L2 loss function without penalty, defined in (9), is used for the simulation study. This topic belongs to future research. Potential improvements could be tuning the parameter Δ or checking whether the first derivative $\frac{\tilde{y}_{t+s}-\tilde{y}_t}{s}$ over a certain window s is negative. Then the windows s that violate the monotone increasing function property can be penalized. In the case treated above, the windows around -25 should be penalized. Other options would be to impose the penalty only after a certain number of epochs or rewarding increasing parts instead of only penalizing decreasing parts.

F Bivariate mixed Poisson Distribution

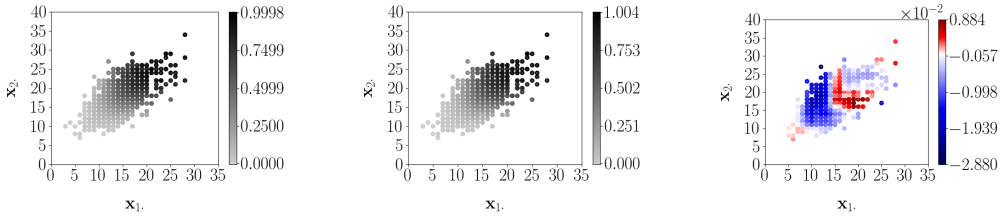
This section presents differences between the target CDF and the estimated CDF as well as differences between the true PMF and the estimated PMF, together with the estimates of the CDF and PMF itself of the Poisson simulation case in section 5.3.2. Figure F.1 presents the best CDF estimate in terms of lowest $L2_{\widetilde{CDF}}$ loss. This implies that Figure 5.56(a) corresponds to the difference of F.1(a) and F.1(b), also represented in Figure F.1(c). This is also done for the average CDF estimate in Figure F.2 and the worst CDF estimate in Figure F.3.



(a) Target CDF (b) Estimated CDF (c) Estimated - Target

Figure F.1 medium sample best CDF estimates

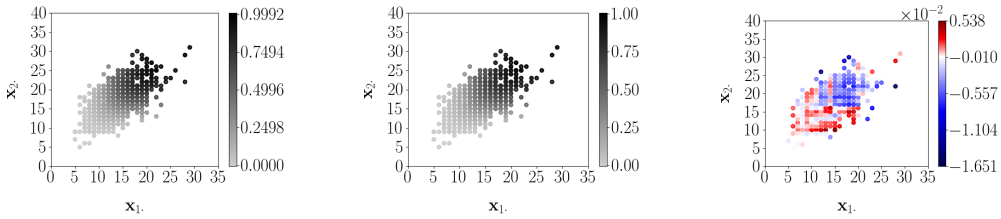
Target and estimated CDF using the proposed method for mixed Poisson data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) Target CDF (b) Estimated CDF (c) Estimated - Target

Figure F.2 medium sample average CDF estimates

Target and estimated CDF using the proposed method for mixed Poisson data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.

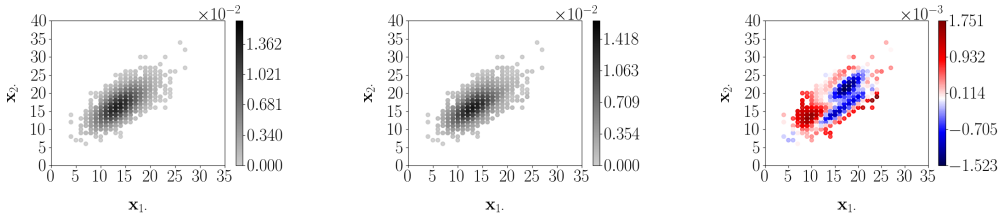


(a) Target CDF (b) Estimated CDF (c) Estimated - Target

Figure F.3 medium sample worst CDF estimates

Target and estimated CDF using the proposed method for mixed Poisson data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.

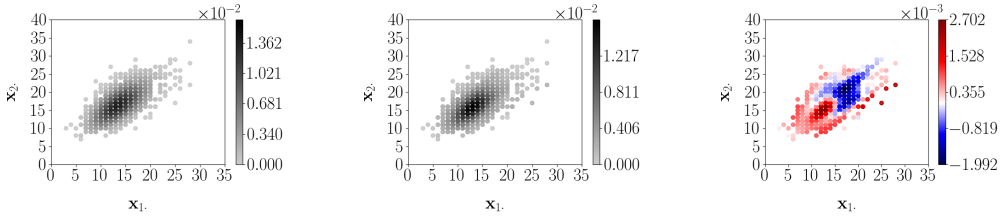
Figure F.4 presents the best PMF estimate in terms of lowest $L2_{\widetilde{CDF}}$ loss. This implies that Figure 5.57(a) corresponds to the difference of F.4(a) and F.4(b), also represented in Figure F.4(c). This is also done for the average PMF estimate in Figure F.5 and the worst PMF estimate in Figure F.6.



(a) True PMF (b) Estimated PMF (c) Estimated - True

Figure F.4 medium sample best PMF estimates

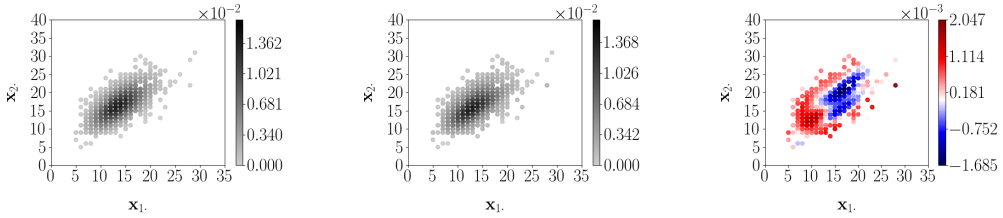
True and estimated PMF using the proposed method for mixed Poisson data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) True PMF (b) Estimated PMF (c) Estimated - True

Figure F.5 medium sample average PMF estimates

True and estimated PMF using the proposed method for mixed Poisson data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.

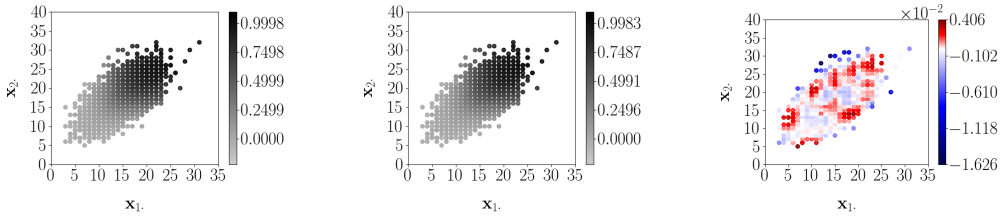


(a) True PMF (b) Estimated PMF (c) Estimated - True

Figure F.6 medium sample worst PMF estimates

True and estimated PMF using the proposed method for mixed Poisson data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.

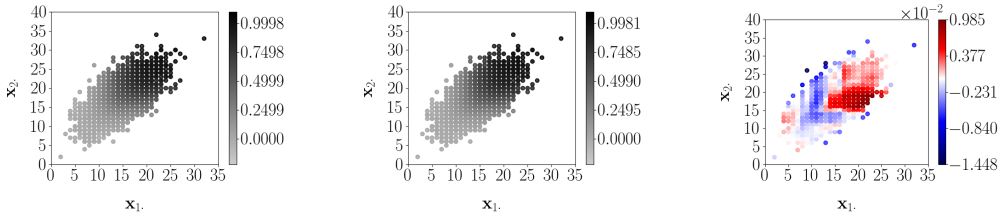
Similar figures are shown for the large sample size. Figure F.7 presents the best CDF estimate in terms of lowest $L2_{\widetilde{CDF}}$ loss. This implies that Figure 5.59(a) corresponds to the difference of F.7(a) and F.7(b), also represented in Figure F.7(c). This is also done for the average CDF estimate in Figure F.8 and the worst CDF estimate in Figure F.9.



(a) Target CDF (b) Estimated CDF (c) Estimated - Target

Figure F.7 large sample best CDF estimates

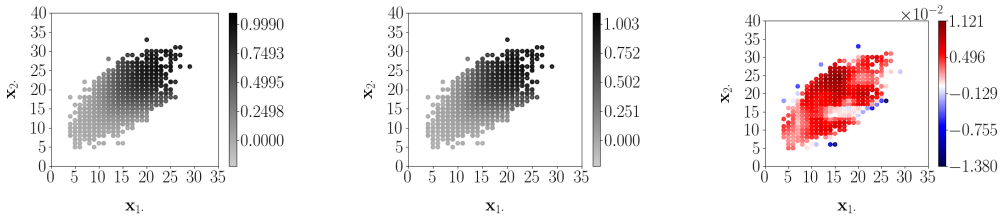
Target and estimated CDF using the proposed method for mixed Poisson data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) Target CDF (b) Estimated CDF (c) Estimated - Target

Figure F.8 large sample average CDF estimates

Target and estimated CDF using the proposed method for mixed Poisson data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.

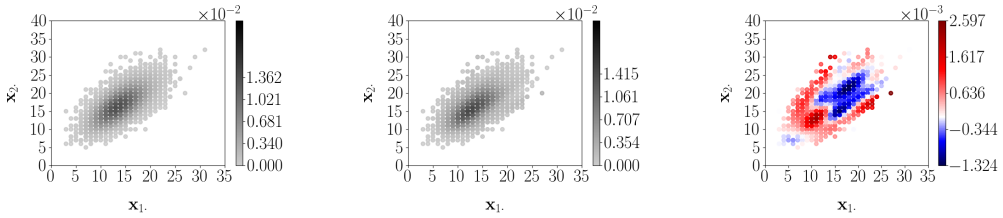


(a) Target CDF (b) Estimated CDF (c) Estimated - Target

Figure F.9 large sample worst CDF estimates

Target and estimated CDF using the proposed method for mixed Poisson data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.

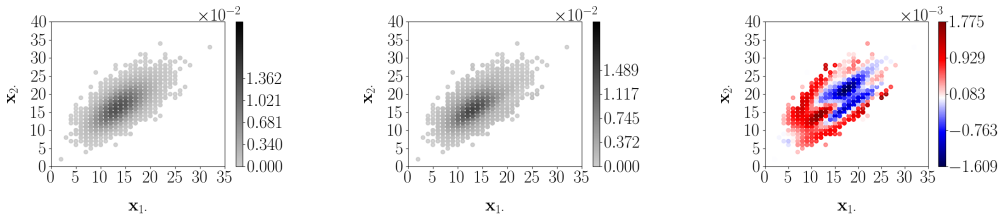
Figure F.10 presents the best PMF estimate in terms of lowest $L2_{\widetilde{CDF}}$ loss. This implies that Figure 5.60(a) corresponds to the difference of F.10(a) and F.10(b), also represented in Figure F.10(c). This is also done for the average PMF estimate in Figure F.11 and the worst PMF estimate in Figure F.12.



(a) True PMF (b) Estimated PMF (c) Estimated - True

Figure F.10 large sample best PMF estimates

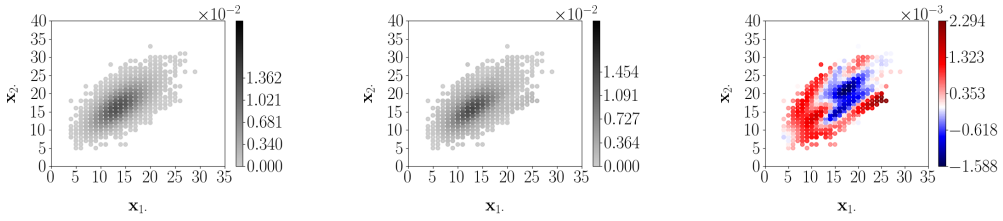
True and estimated PMF using the proposed method for mixed Poisson data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) True PMF (b) Estimated PMF (c) Estimated - True

Figure F.11 large sample average PMF estimates

True and estimated PMF using the proposed method for mixed Poisson data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) True PMF (b) Estimated PMF (c) Estimated - True

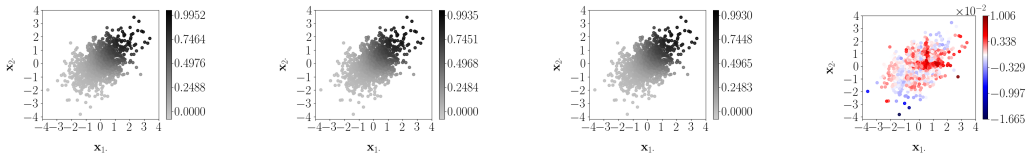
Figure F.12 large sample worst PMF estimates

True and estimated PMF using the proposed method for mixed Poisson data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.

G Bivariate mixed normal Distribution

This section presents differences between the target CDF and the estimated CDF as well as the differences between the true PDF and the estimated PDF, together with the estimates of the CDF and PDF of the bivariate simulation cases in section 5.4. Figure G.1 presents the best CDF estimate in terms of lowest $L2_{\widetilde{CDF}}$ loss. This implies that Figure

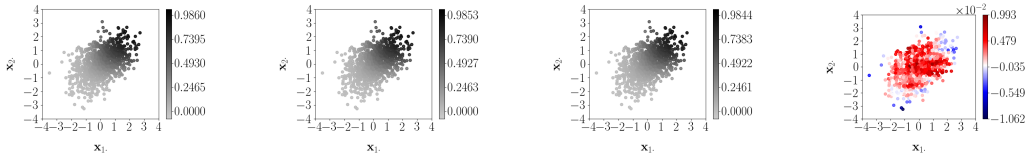
5.62(a) corresponds to the difference of G.1(b) and G.1(c), also represented in Figure G.1(d). This is also done for the average CDF estimate in Figure G.2 and the worst CDF estimate in Figure G.3.



(a) True CDF (b) Target CDF (c) Estimated CDF (d) Estimated - Target

Figure G.1 medium sample best CDF estimates

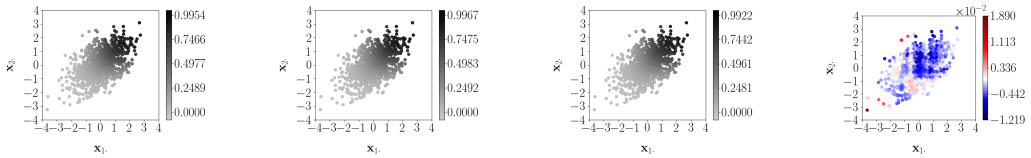
True, target, and estimated CDF using the proposed method for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 10 neurons are given.



(a) True CDF (b) Target CDF (c) Estimated CDF (d) Estimated - Target

Figure G.2 medium sample average CDF estimates

True, target, and estimated CDF using the proposed method for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 10 neurons are given.

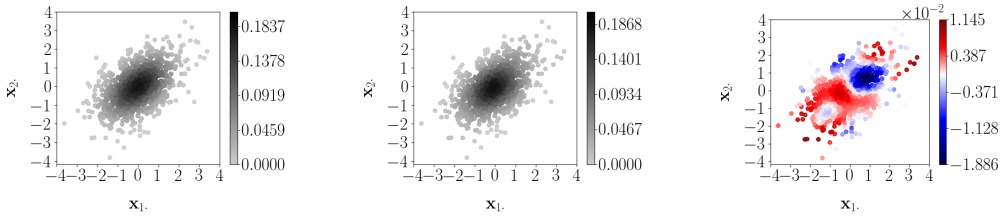


(a) True CDF (b) Target CDF (c) Estimated CDF (d) Estimated - Target

Figure G.3 medium sample worst CDF estimates

True, target, and estimated CDF using the proposed method for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 10 neurons are given.

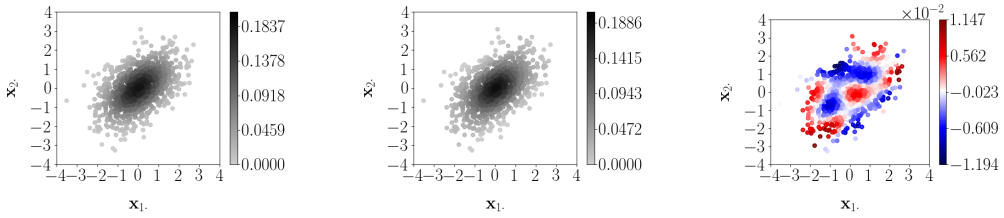
Figure G.4 presents the best PDF estimate in terms of lowest $L2_{\widetilde{CDF}}$ loss. This implies that Figure 5.63(a) corresponds to the difference of G.4(a) and G.4(b), also represented in Figure G.4(c). This is also done for the average PDF estimate in Figure G.5 and the worst PDF estimate in Figure G.6.



(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure G.4 medium sample best PDF estimates

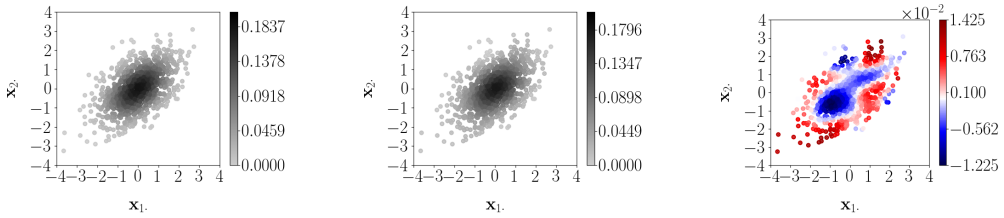
True and estimated PDF using the proposed method for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 10 neurons are given.



(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure G.5 medium sample average PDF estimates

True and estimated PDF using the proposed method for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 10 neurons are given.

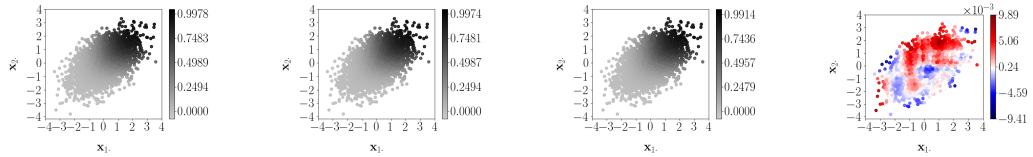


(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure G.6 medium sample worst PDF estimates

True and estimated PDF using the proposed method for mixed Poisson data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 2 hidden layers and 10 neurons are given.

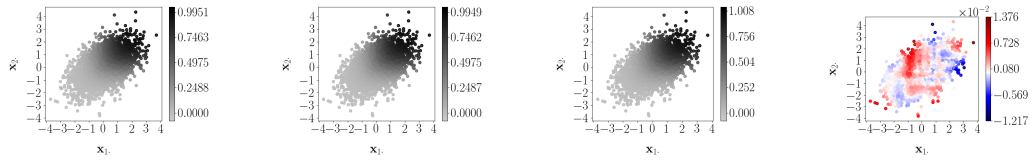
Similar figures are shown for the large sample size. Figure G.10 presents the best PDF estimate in terms of lowest $L2_{\widetilde{CDF}}$ loss. This implies that Figure 5.66(a) corresponds to the difference of G.10(a) and G.10(b), also represented in Figure G.10(c). This is also done for the average PDF estimate in Figure G.11 and the worst PDF estimate in Figure G.12.



(a) True CDF (b) Target CDF (c) Estimated CDF (d) Estimated - Target

Figure G.7 large sample best CDF estimates

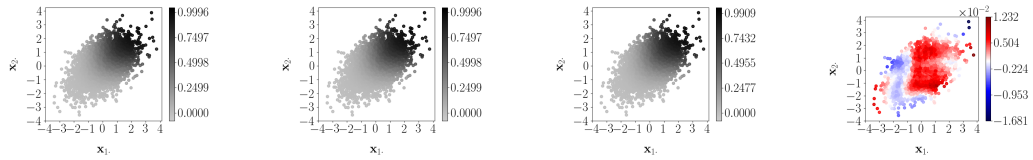
True, target, and estimated CDF using the proposed method for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 10 neurons are given.



(a) True CDF (b) Target CDF (c) Estimated CDF (d) Estimated - Target

Figure G.8 large sample average CDF estimates

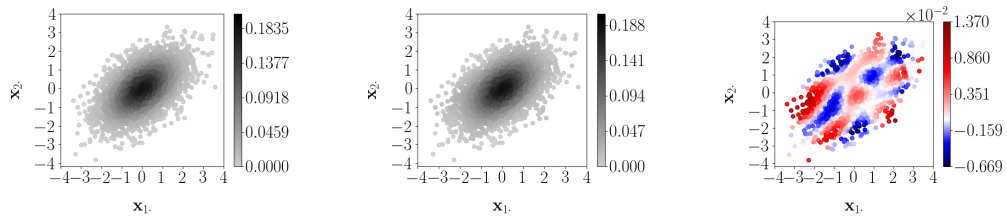
True, target, and estimated CDF using the proposed method for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 10 neurons are given.



(a) True CDF (b) Target CDF (c) Estimated CDF (d) Estimated - Target

Figure G.9 large sample worst CDF estimates

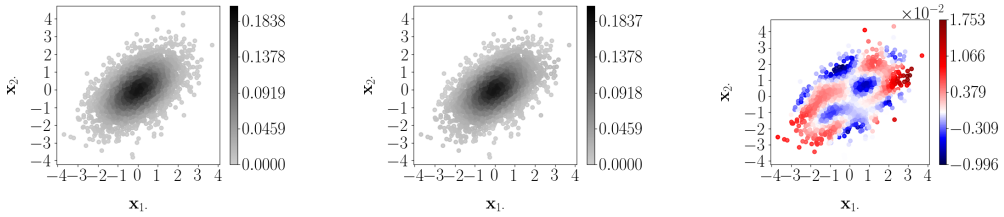
True, target, and estimated CDF using the proposed method for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 10 neurons are given.



(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure G.10 large sample best PDF estimates

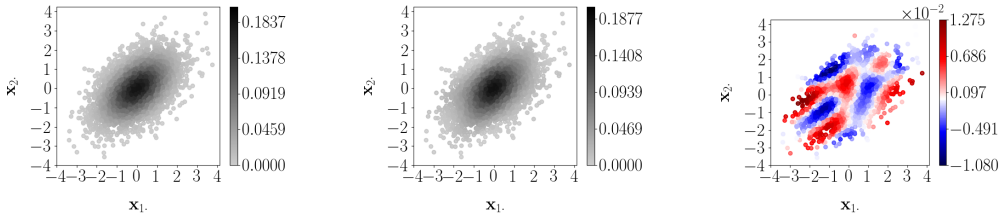
True and estimated PDF using the proposed method for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 10 neurons are given.



(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure G.11 large sample average PDF estimates

True and estimated PDF using the proposed method for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 10 neurons are given.



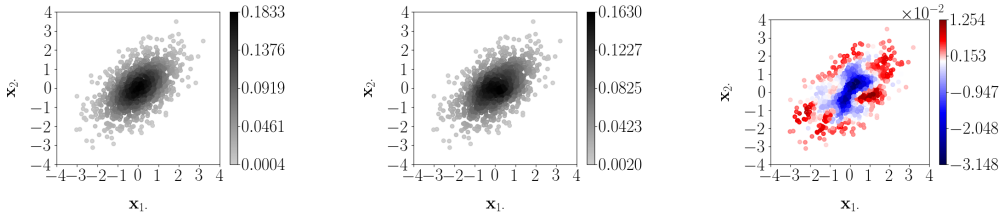
(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure G.12 large sample worst PDF estimates

True and estimated PDF using the proposed method for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 2 hidden layers and 10 neurons are given.

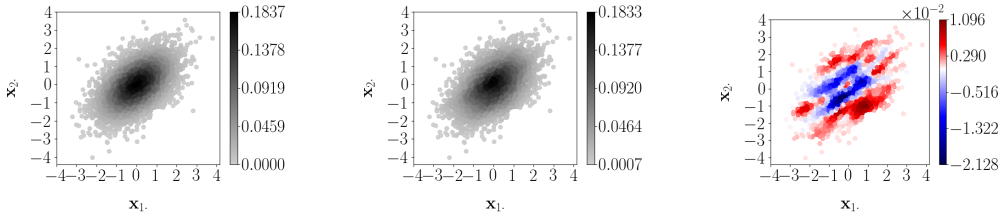
H Bivariate mixed normal Distribution by KDE

This section presents differences between the true PDF and the estimated PDF, together with the estimates of the PDF of the bivariate simulation cases estimated by KDE in section 5.4. Figure H.1 presents the average PDF estimate in terms of lowest $L2_{PDF}$ loss. This implies that Figure 5.68(a) corresponds to the difference of H.1(a) and H.1(b), also represented in Figure H.1(c). This is also done for the large sample PDF estimate by KDE in Figure H.2.



(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure H.1 medium sample average PDF estimates by KDE
True and estimated PDF using KDE for mixed normal data. The average estimate in terms of $L2_{PDF}$ loss values out of 100 simulation replications for the medium sample size using Scott's bandwidthare given.

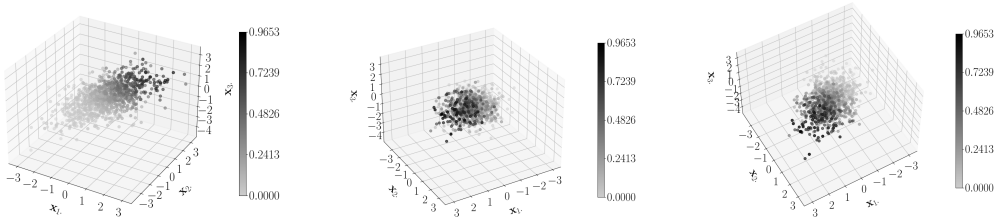


(a) True PDF (b) Estimated PDF (c) Estimated - True

Figure H.2 large sample average PDF estimates by KDE
True and estimated PDF using KDE for mixed normal data. The average estimate in terms of $L2_{PDF}$ loss values out of 100 simulation replications for the medium sample size using Scott's bandwidth are given.

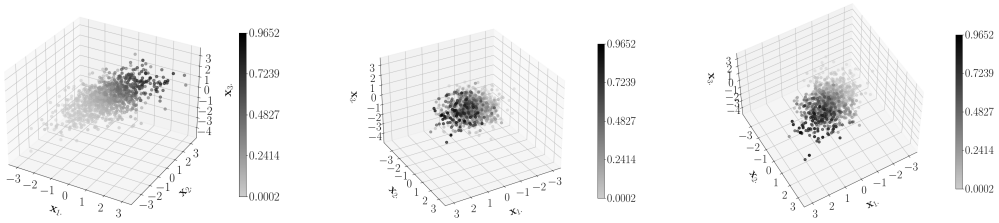
I Trivariate mixed normal Distribution

This section presents the estimates of the true, target, and estimated CDF and true and estimated PDF of the trivariate simulation cases shown in section 5.4. Figures I.1, I.2, and I.3 present the true, target, and estimated CDF corresponding to the simulation with the smallest $L2_{\widetilde{CDF}}$ loss. Three different angles are shown of each Figure. This is also done for the average true, target, and estimated CDF shown in Figures I.4, I.5, and I.6 and the worst true, target, and estimated CDF shown in Figures I.7, I.8, and I.9.



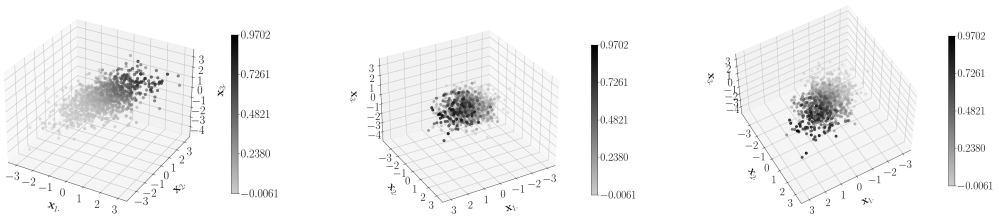
(a) front angle (b) left angle (c) top angle

Figure I.1 medium sample true CDF corresponding to best CDF estimate
True PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



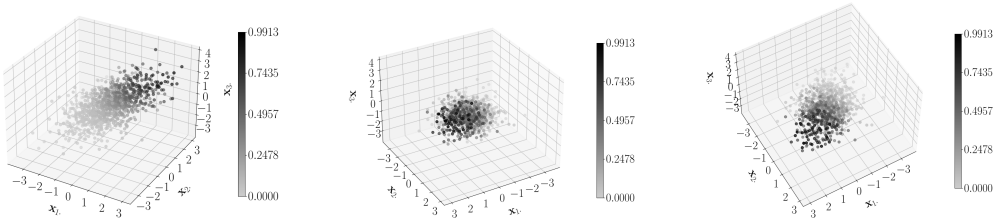
(a) front angle (b) left angle (c) top angle

Figure I.2 medium sample target CDF corresponding to best CDF estimate
Target PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



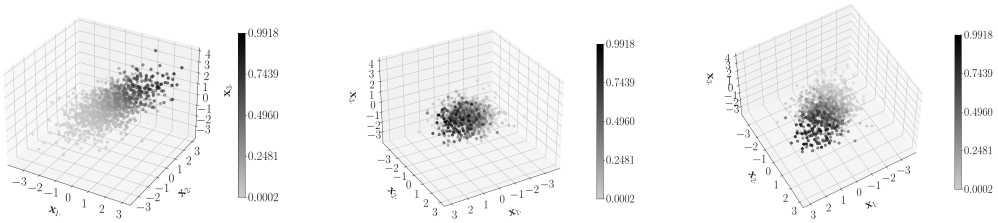
(a) front angle (b) left angle (c) top angle

Figure I.3 medium sample best CDF estimate
Estimated PDF for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



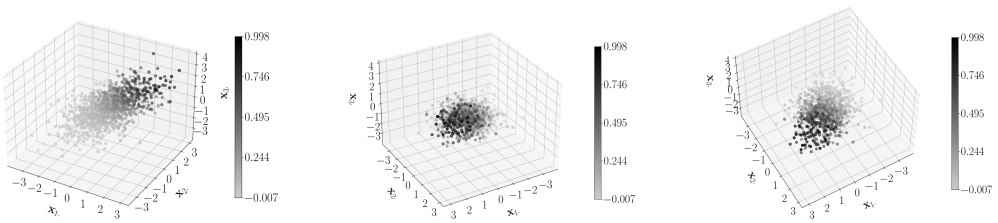
(a) front angle (b) left angle (c) top angle

Figure I.4 medium sample true CDF corresponding to average CDF estimate
True PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



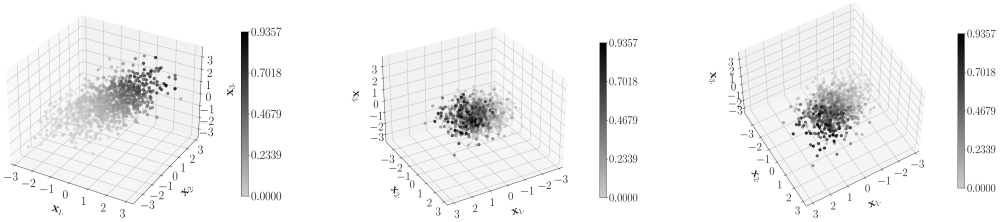
(a) front angle (b) left angle (c) top angle

Figure I.5 medium sample target CDF corresponding to average CDF estimate
Target PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



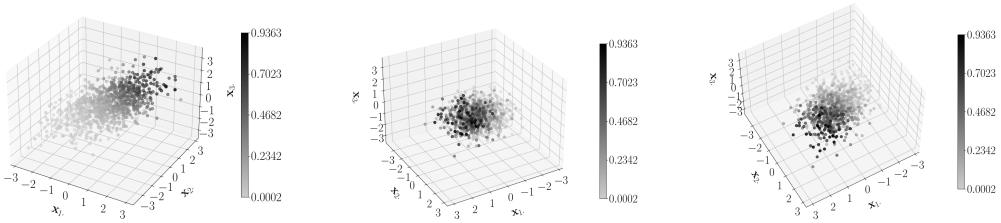
(a) front angle (b) left angle (c) top angle

Figure I.6 medium sample average CDF estimate
Estimated PDF for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



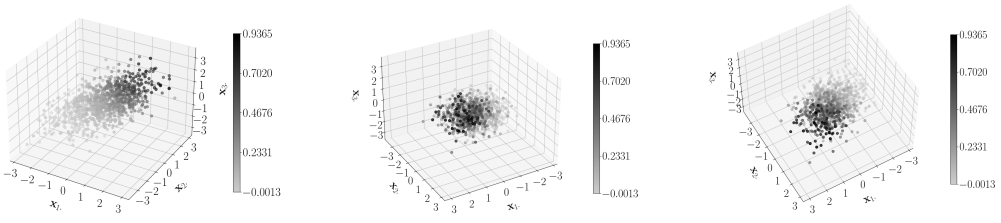
(a) front angle (b) left angle (c) top angle

Figure I.7 medium sample true CDF corresponding to worst CDF estimate
True PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

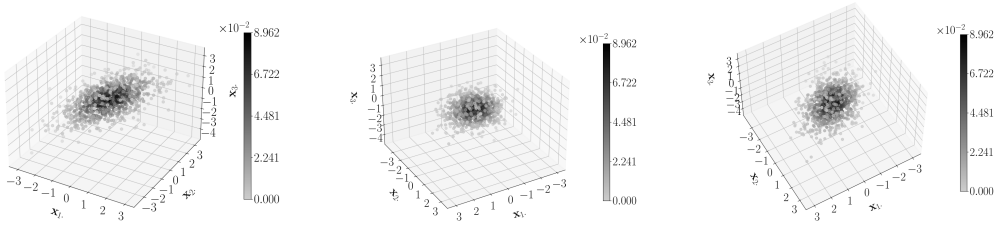
Figure I.8 medium sample target CDF corresponding to worst CDF estimate
Target PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.9 medium sample worst CDF estimate
Estimated PDF for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.

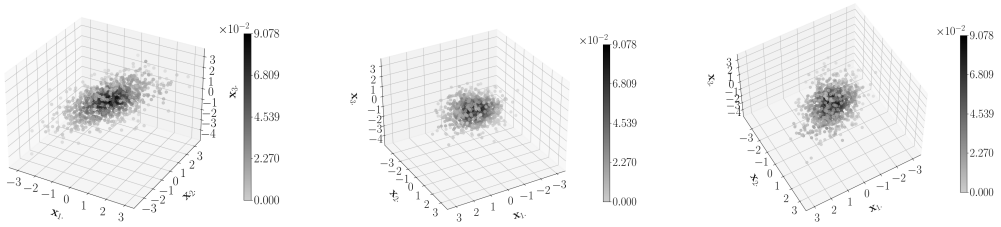
Figures I.10 and I.11 present the true and estimated PDF corresponding to the simulation with the smallest $L2_{\widetilde{CDF}}$ loss. Three different angles are shown of each Figure. This is also done for the average true and estimated PDF shown in Figures I.12 and I.13 and the worst true and estimated PDF shown in Figures I.14, I.15.



(a) front angle (b) left angle (c) top angle

Figure I.10 medium sample true PDF corresponding to medium sample best PDF estimate

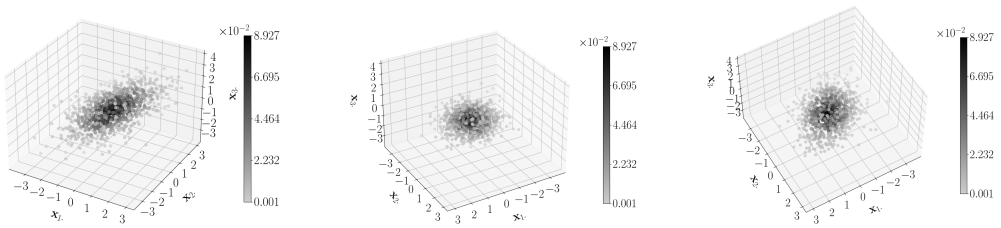
True PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.11 medium sample best PDF estimate

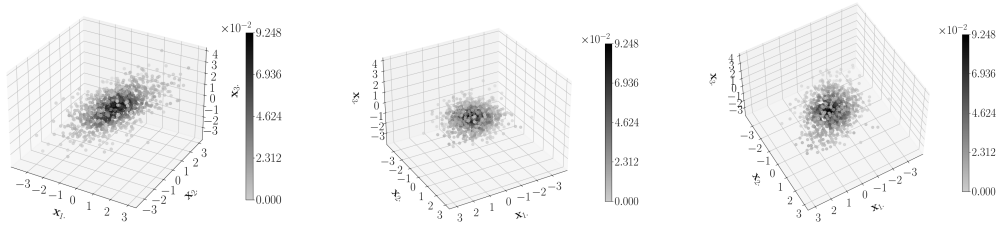
Estimated PDF for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.12 medium sample true PDF corresponding to medium sample average PDF estimate

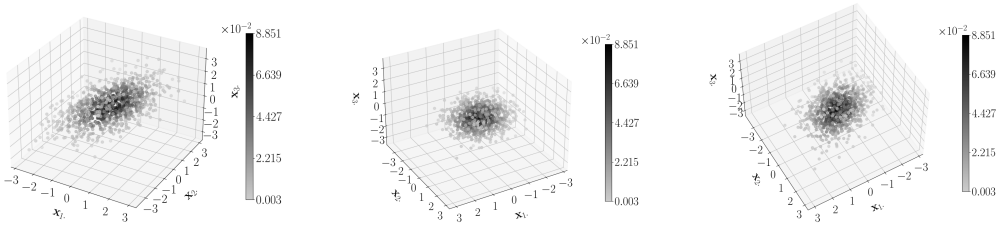
True PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.13 medium sample average PDF estimate

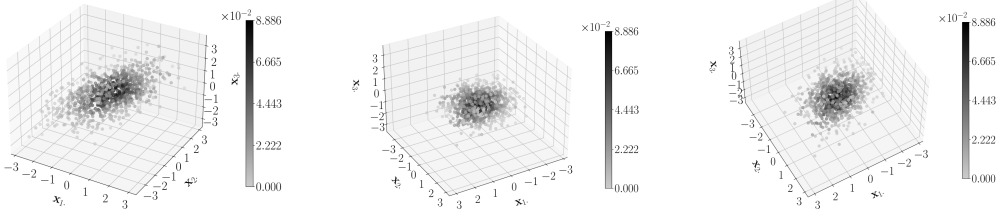
Estimated PDF for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.14 medium sample true PDF corresponding to medium sample worst PDF estimate

True PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.

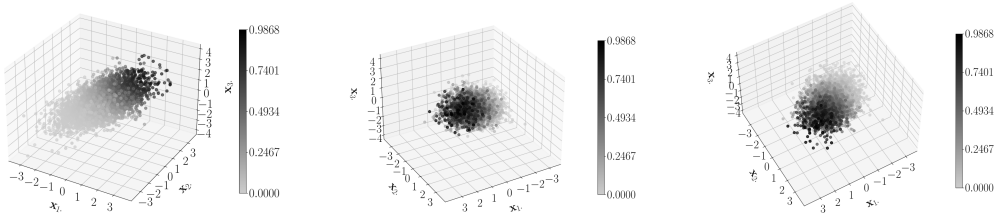


(a) front angle (b) left angle (c) top angle

Figure I.15 medium sample worst PDF estimate

Estimated PDF for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the medium sample size using 3 hidden layers and 25 neurons are given.

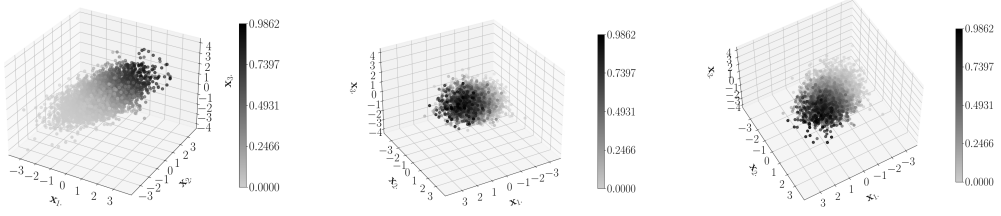
Similar Figures are shown for the large sample size. Figures I.16, I.17, and I.18 present the true, target, and estimate corresponding to the simulation with the smallest $L2_{\widetilde{CDF}}$ loss. Three different angles are shown of each Figure. This is also done for the average true, target, and estimated CDF shown in Figures I.19, I.20, and I.21 and the worst true, target, and estimated CDF shown in Figures I.22, I.23, and I.24.



(a) front angle (b) left angle (c) top angle

Figure I.16 large sample true CDF corresponding to best CDF estimate

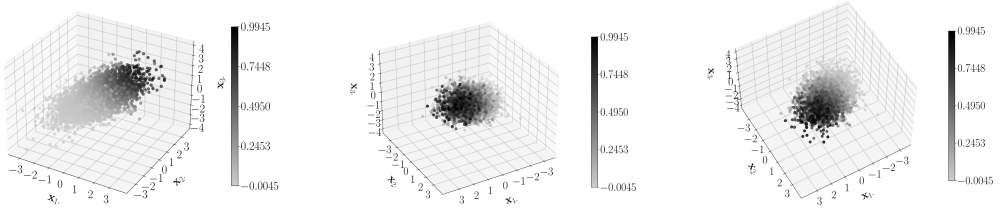
True PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.17 large sample target CDF corresponding to best CDF estimate

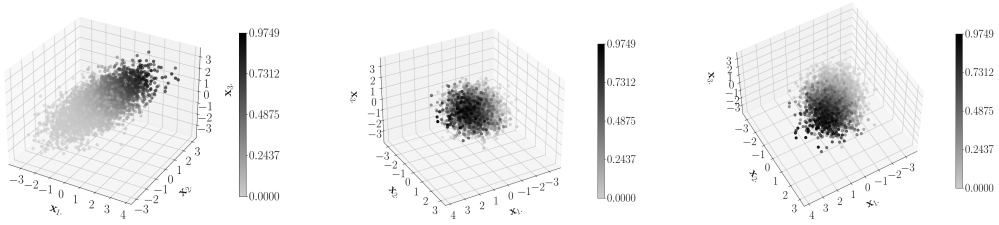
Target PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



(a) front angle (b) left angle (c) top angle

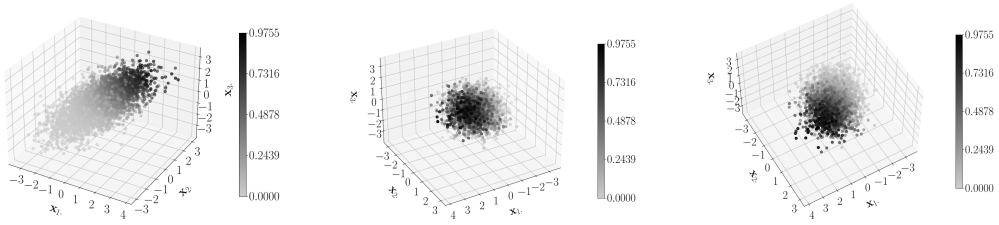
Figure I.18 large sample best CDF estimate

Estimated PDF for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



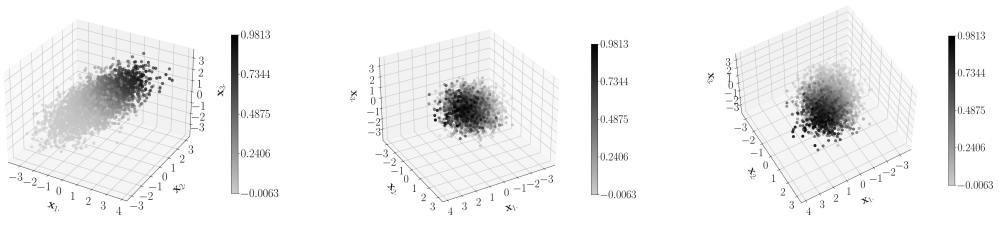
(a) front angle (b) left angle (c) top angle

Figure I.19 large sample true CDF corresponding to average CDF estimate
True PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



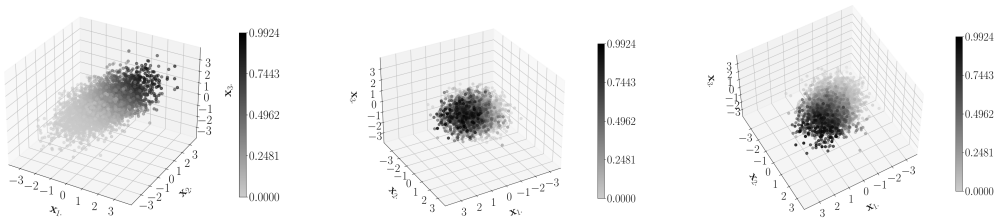
(a) front angle (b) left angle (c) top angle

Figure I.20 large sample target CDF corresponding to average CDF estimate
Target PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



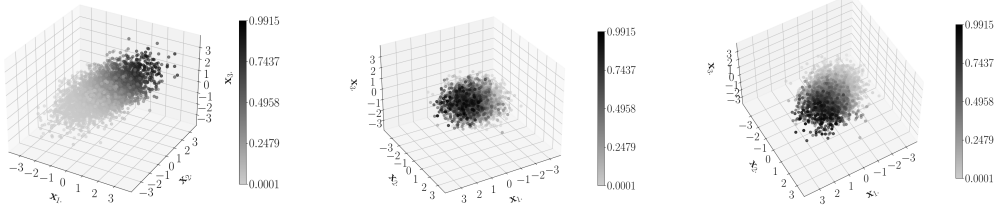
(a) front angle (b) left angle (c) top angle

Figure I.21 large sample average CDF estimate
Estimated PDF for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



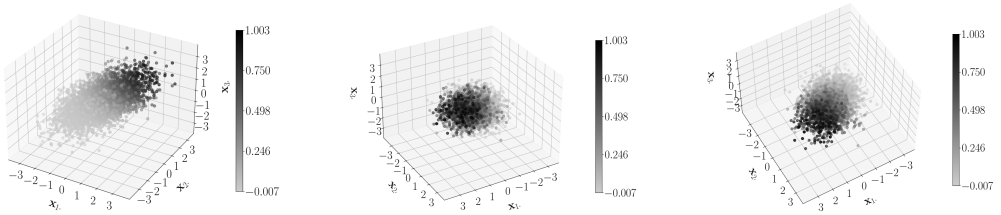
(a) front angle (b) left angle (c) top angle

Figure I.22 large sample true CDF corresponding to worst CDF estimate
True PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



(a) front angle (b) left angle (c) top angle

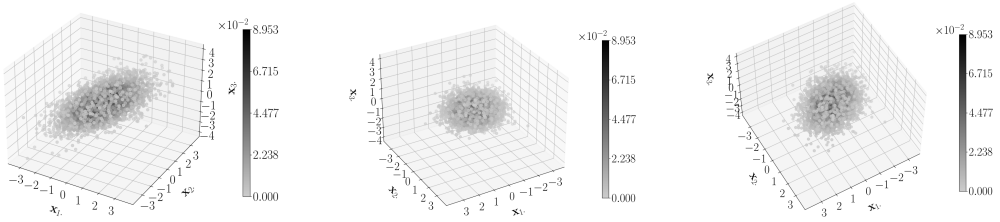
Figure I.23 large sample target CDF corresponding to worst CDF estimate
Target PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



(a) front angle (b) left angle (c) top angle

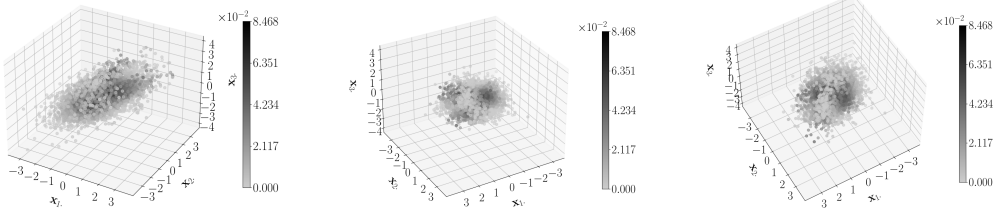
Figure I.24 large sample worst CDF estimate
Estimated PDF for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.

Figures I.25 and I.26 present the true and estimated PDF corresponding to the simulation with the smallest $L2_{\widetilde{CDF}}$ loss. Three different angles are shown of each Figure. This is also done for the average true and estimated PDF shown in Figures I.27 and I.28 and the worst true and estimated PDF shown in Figures I.29, I.30.



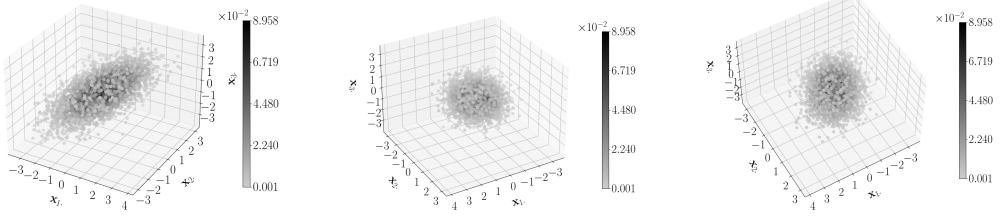
(a) front angle (b) left angle (c) top angle

Figure I.25 large sample true PDF corresponding to best PDF estimate
True PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



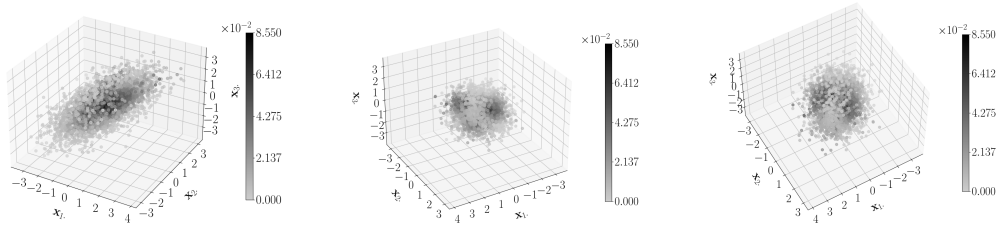
(a) front angle (b) left angle (c) top angle

Figure I.26 large sample best PDF estimate
Estimated PDF for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



(a) front angle (b) left angle (c) top angle

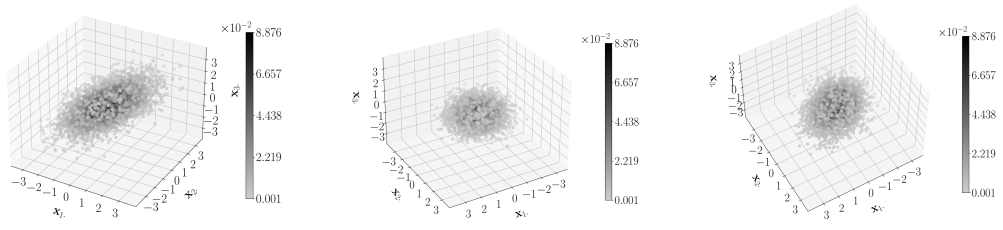
Figure I.27 large sample true PDF corresponding to average PDF estimate
True PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.28 large sample average PDF estimate

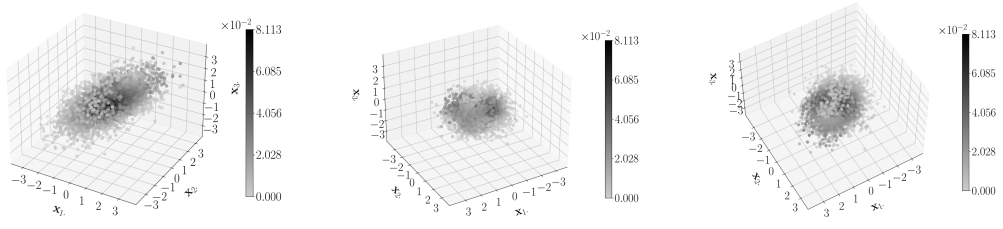
Estimated PDF for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.29 large sample true PDF corresponding to worst PDF estimate

True PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.

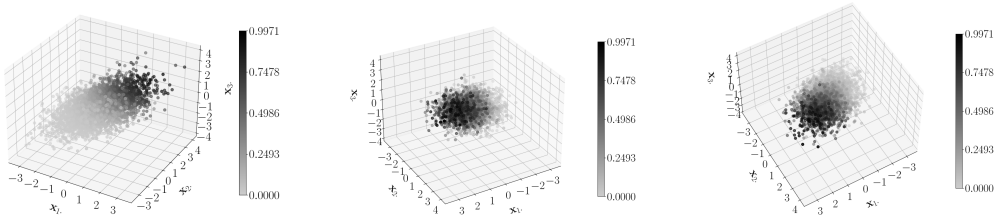


(a) front angle (b) left angle (c) top angle

Figure I.30 large sample worst PDF estimate

Estimated PDF for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 10 neurons are given.

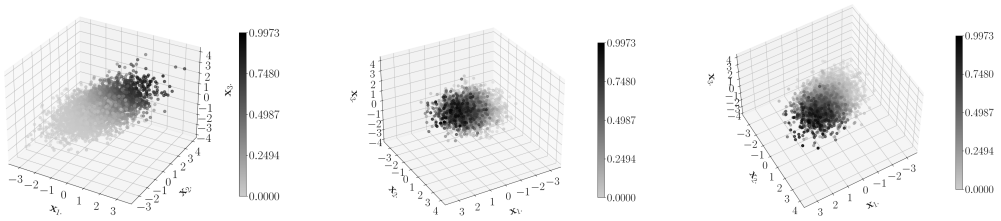
Similar Figures are shown for the large sample size using model 5 instead of model 4 due to the small $L2_{PDF}$ loss in Table 5.28. Figures I.31, I.32, and I.33 present the true, target, and estimate corresponding to the simulation with the smallest $L2_{\widetilde{CDF}}$ loss. Three different angles are shown of each Figure. This is also done for the average true, target, and estimated CDF shown in Figures I.34, I.35, and I.36 and the worst true, target, and estimated CDF shown in Figures I.37, I.38, and I.39.



(a) front angle (b) left angle (c) top angle

Figure I.31 large sample true CDF corresponding to best CDF estimate using model 5

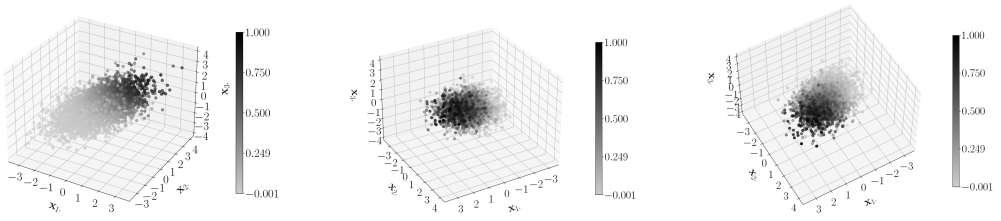
True PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.32 large sample target CDF corresponding to best CDF estimate using model 5

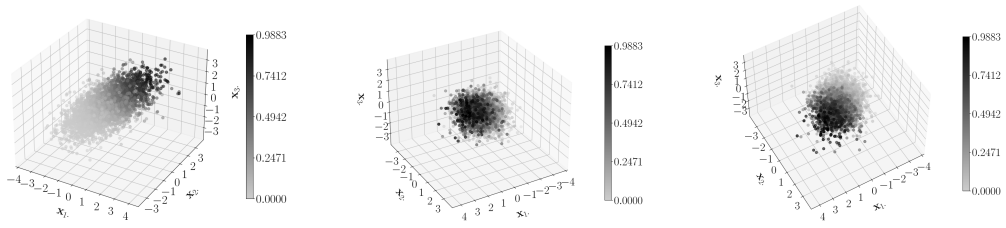
Target PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.33 large sample best CDF estimate using model 5

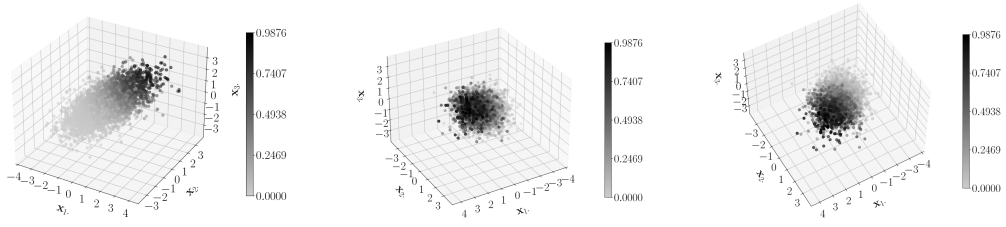
Estimated PDF for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.34 large sample true CDF corresponding to average CDF estimate using model 5

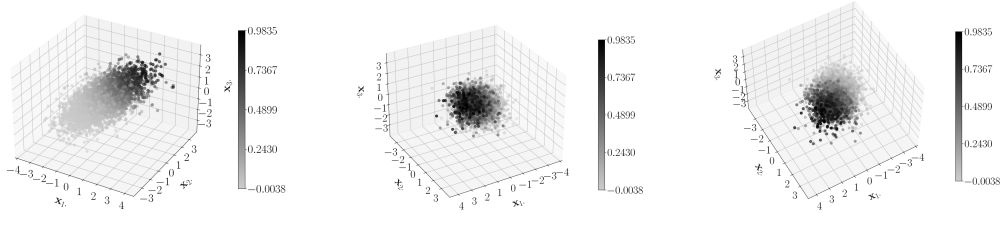
True PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.35 large sample target CDF corresponding to average CDF estimate using model 5

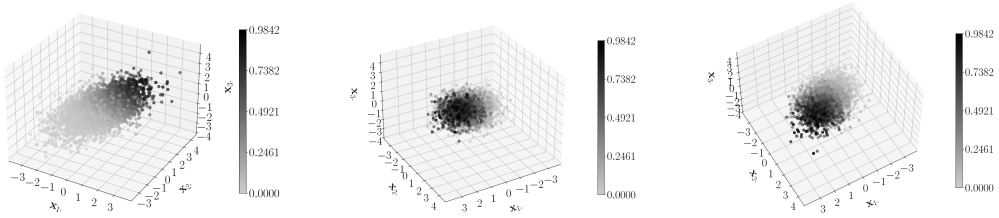
Target PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.36 large sample average CDF estimate using model 5

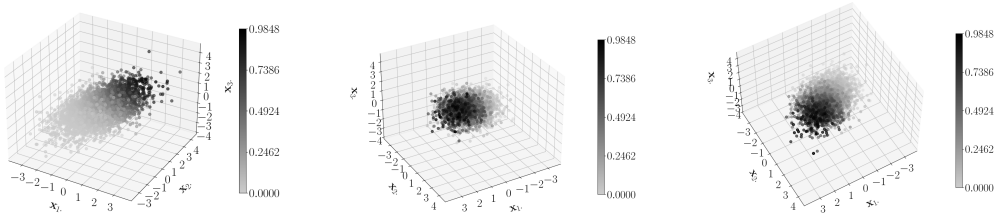
Estimated PDF for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.37 large sample true CDF corresponding to worst CDF estimate using model 5

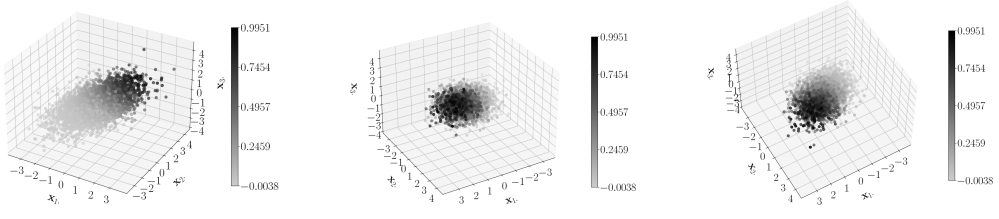
True PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.38 large sample target CDF corresponding to worst CDF estimate using model 5

Target PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.

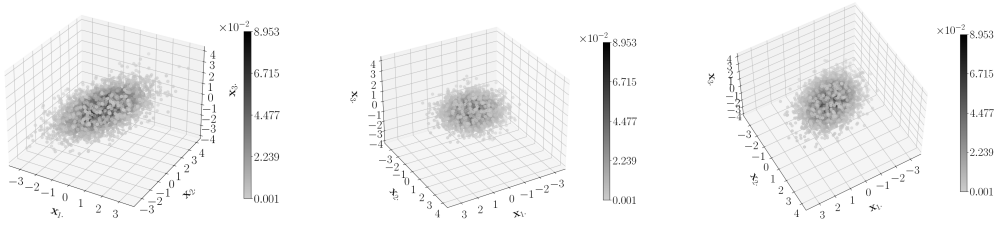


(a) front angle (b) left angle (c) top angle

Figure I.39 large sample worst CDF estimate using model 5

Estimated PDF for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.

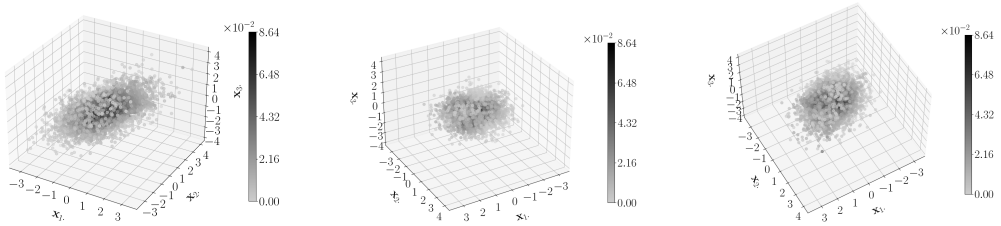
Figures I.40 and I.41 present the true and estimated PDF corresponding to the simulation with the smallest $L2_{\widetilde{CDF}}$ loss. Three different angles are shown of each Figure. This is also done for the average true and estimated PDF shown in Figures I.42 and I.43 and the worst true and estimated PDF shown in Figures I.44, I.45.



(a) front angle (b) left angle (c) top angle

Figure I.40 large sample true PDF corresponding to best PDF estimate using model 5

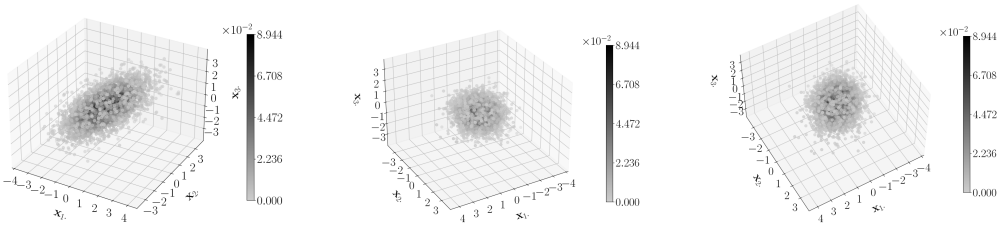
True PDF for mixed normal data corresponding to the best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.41 large sample best PDF estimate using model 5

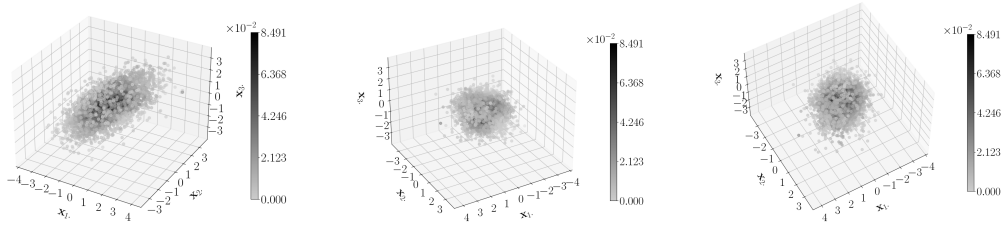
Estimated PDF for mixed normal data. The best estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.42 large sample true PDF corresponding to average PDF estimate using model 5

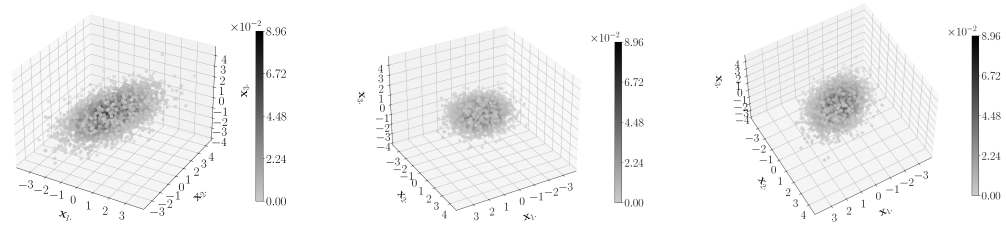
True PDF for mixed normal data corresponding to the average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.43 large sample average PDF estimate using model 5

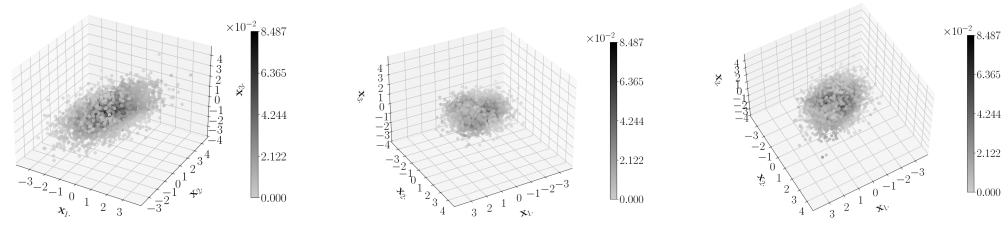
Estimated PDF for mixed normal data. The average estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

Figure I.44 large sample true PDF corresponding to worst PDF estimate using model 5

True PDF for mixed normal data corresponding to the worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.



(a) front angle (b) left angle (c) top angle

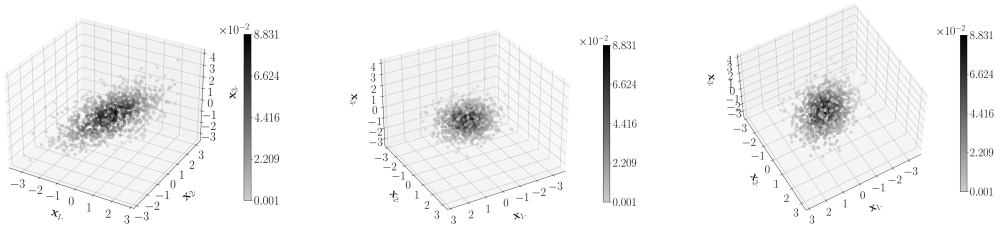
Figure I.45 large sample worst PDF estimate using model 5

Estimated PDF for mixed normal data. The worst estimate in terms of $L2_{\widetilde{CDF}}$ loss values out of 100 simulation replications for the large sample size using 3 hidden layers and 25 neurons are given.

J Trivariate mixed normal Distribution by KDE

This section presents the estimates of the true and estimated PDF of the trivariate simulation cases by KDE shown in section 5.4. Figures J.1 and J.2 present the true and estimated PDF corresponding to the smallest $L2_{PDF}$ loss. Three different angles are

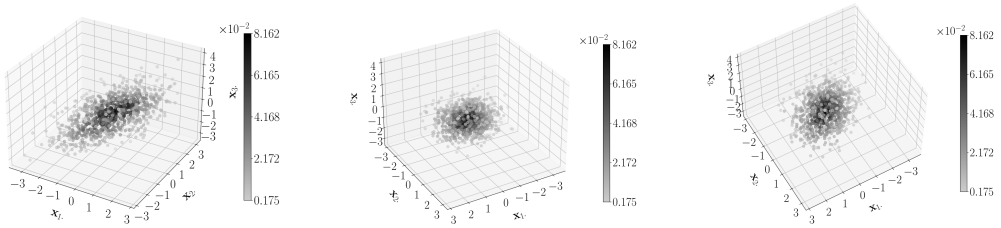
shown of each Figure. This is also done for the average true and estimated PDF for the large sample shown in Figures J.3 and J.4.



(a) front angle (b) left angle (c) top angle

Figure J.1 medium sample true PDF corresponding to average PDF estimate by KDE

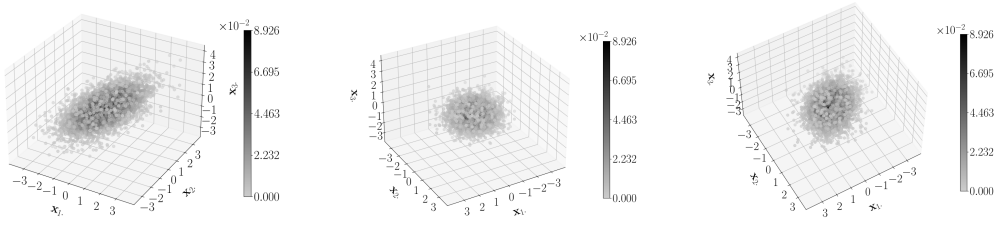
True PDF for mixed normal data corresponding to the average estimate by KDE in terms of $L2_{PDF}$ loss values out of 100 simulation replications for the medium sample size using Scott's rule are given.



(a) front angle (b) left angle (c) top angle

Figure J.2 medium sample average PDF estimate by KDE

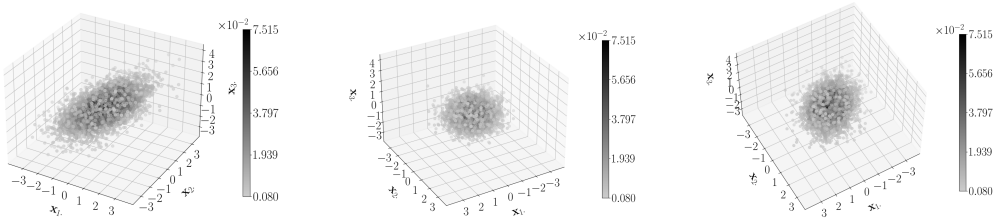
Estimated PDF by KDE for mixed normal data. The average estimate in terms of $L2_{PDF}$ loss values out of 100 simulation replications for the large sample size using Scott's rule are given.



(a) front angle (b) left angle (c) top angle

Figure J.3 large sample true PDF corresponding to average PDF estimate by KDE

True PDF for mixed normal data corresponding to the average estimate by KDE in terms of $L2_{PDF}$ loss values out of 100 simulation replications for the large sample size using Scott's rule are given.



(a) front angle (b) left angle (c) top angle

Figure J.4 large sample average PDF estimate by KDE

Estimated PDF by KDE for mixed normal data. The average estimate in terms of $L2_{PDF}$ loss values out of 100 simulation replications for the large sample size using Scott's rule are given.

K Simulation settings

simulation study	method	sample size	epochs	LR	activation	output	HL	HN
mixed normal distribution	Proposed	S, M, L	10 000	0.01	logistic	-	1, 2	3, 4, 5
	TEA	S, M, L	10 000	0.01	logistic	linear	1, 2	3, 4, 5
	MIA's settings	S, 2S	10 000	?	tanh	erf[a)]	1	3
mixed GEV distribution	Proposed	S	10 000	0.01	logistic & tanh	-	1, 2, 3	5, 9, 15
	TEA	S, M	10 000	0.01	logistic	linear	1, 2, 3	5, 9, 15
	TEA's settings	S, M, 10M	1000?	0.01	logistic	linear	1	9
poisson distribution (uni)	Proposed	S, M, L	10 000	0.001	tanh	-	1, 2	4, 6, 8
poisson distribution (bi)	Proposed	M, L	10 000	0.001	logistic	linear	1, 2, 3	10, 25, 50
std normal distribution (bi)	Proposed	M, L	5 000	0.001	logistic	linear	1, 2, 3	10, 25, 50
std normal distribution (tri)	Proposed	M, L	20 000	0.001	logistic	linear	2, 3, 4	10, 25, 50

Table K.1 An overview of simulation settings used by the Proposed method, TEA (Trentin et al. (2018)) and MIA (Magdon-Ismail and Atiya (2002))

a) $erf(x) = 1/\sqrt{(2\pi)} \int_{-\infty}^{x} exp(-t^2/2)dt$

Disclaimer

The views expressed in this paper are those of the authors and do not necessarily reflect the policy of Statistics Netherlands.

Reviewed by Rob Willems.

References

Arbogast, L. F. A. (1800). *Du calcul des dérivations*. Levrault, frères.

Bishop, C. M. et al. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Boyadzhiev, K. N. (2009). Derivative polynomials for tanh, tan, sech and sec in explicit form. *arXiv preprint arXiv:0903.0117*.

Cochran, W. (1977). *Sampling Techniques*. Wiley and Sons.

Cramer, H. (1946). Mathematical methods of statistics, princeton, 1946. *Mathematical Reviews (Math-SciNet): MR16588 Zentralblatt MATH 63*, 300.

Cramer, J. S. (1989). *Econometric applications of maximum likelihood methods*. CUP Archive.

Durbin, J. and S. J. Koopman (2012). *Time series analysis by state space methods*. Oxford University Press.

Eliason, S. R. (1993). *Maximum likelihood estimation: Logic and practice*. Number 96 in 07. Sage.

Faà di Bruno, F. (1855). Sullo sviluppo delle funzioni. *Annali di scienze matematiche e fisiche 6*, 479–480.

Glorot, X. and Y. Bengio (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT press.

Hardy, M. (2006). Combinatorics of partial derivatives. *The Electronic Journal of Combinatorics 13*(1), 1.

Magdon-Ismail, M. and A. Atiya (2002). Density estimation and random variate generation using multilayer networks. *IEEE Transactions on Neural Networks 13*(3), 497–520.

Magnus, J. R. (2007). The asymptotic variance of the pseudo maximum likelihood estimator. *Econometric Theory 23*(5), 1022–1032.

Minai, A. A. and R. D. Williams (1993). On the derivatives of the sigmoid. *Neural Networks 6*(6), 845–853.

Puts, M. J., P. J. Daas, M. Tennekes, and C. d. Blois (2018). Using huge amounts of road sensor data for official statistics. *Repository Radboud Universiteit*.

Särndal, C.-E., B. Swensson, and J. Wretman (1992). *Model Assisted Survey Sampling*. Springer.

Shin, K. and R. Pasupathy (2007). A method for fast generation of bivariate poisson random vectors. In *Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come*, pp. 472–479. IEEE Press.

Silverman, B. W. (2018). *Density estimation for statistics and data analysis*. Routledge.

Trentin, E., L. Lusnig, and F. Cavalli (2018). Parzen neural networks: fundamentals, properties, and an application to forensic anthropology. *Neural Networks 97*, 137–151.

van den Brakel, J., E. Söhler, P. Daas, and B. Buelens (2017). Social media as a data source for official statistics; the dutch consumer confidence index. *Survey Methodology 43*(2).

Wilson, R. and J. J. Watkins (2013). *Combinatorics: ancient & modern*. OUP Oxford.

Zhang, S. (2018). From cdf to pdf—a density estimation method for high dimensional data. *arXiv preprint arXiv:1804.05316*.

Colophon

Publisher
Statistics Netherlands
Henri Faasdreef 312, 2492 JP The Hague
www.cbs.nl

Prepress
Statistics Netherlands, Grafimedia

Design
Edenspiekermann

Information
Telephone +31 88 570 70 70, fax +31 70 337 59 94
Via contact form: www.cbs.nl/information