DynaMORE

Dynamic MOdelling of REsilience

H2020 - 777084

D2.1– Open source code for approach for preprocessing longitudinal data

Dissemination level	Public
Contractual date of delivery	31 March 2020
Actual date of delivery	31 March 2020
Type	Other
Version	1
Filename	DynaMORE Deliverable Report D2.1
Workpackage	WP2
Workpackage leader	Harald Binder

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777084.

Author list

Organisation	Name	Contact information
UKLFR	**Harald Binder**	**binderh@imbi.uni-freiburg.de**
UKLFR	**Göran Köber**	**koeber@imbi.uni-freiburg.de**

Executive Summary

We contribute open source code for preprocessing longitudinal data by combining dynamic modeling and deep learning. More specifically, we develop an algorithm which maps longitudinal observations into the latent space using a deep generative model and estimate the trajectories using another neural network, which parameterizes a system of differential equations. In doing so, we overcome the limitations of classical statistical approaches, deep learning, and dynamic modeling when applied exclusively. This algorithm is also attractive to subject matter researchers because it allows estimating resilience free of distributional limitations. The deep generative model estimates distributions in the latent space where the variance of the distribution expresses uncertainty related to this particular observation (see Figure 1). Sampling from these distributions in the latent space allows generating an arbitrary number of imputations that can be smoothly integrated into the established toolkits for multiple imputation. We developed this algorithm interactively with our collaboration partners, discussing dynamic modeling with WP1 and receiving subject matter insights from WP3. MARP and LORA data were also provided by WP3, which we used for developing and extensive testing. The code is available to WP1 and WP3 on our shared cloud (`DynaCLOUD`), as well as our `GitLab` repository.

Abbreviations

VAE	Variational Autoencoder
ODE	Ordinary Differential Equation
ELBO	Evidence Lower Bound

Table of Contents

1 Deliverable report

1.1 Introduction

We started our journey by investigating more classical statistical approaches for longitudinal data—such as landmarking and joint modeling—but we turned our attention to a combination of differential equations and deep learning for several reasons. Compared to classical statistical approaches, differential equations offer a variety of advantages. Most importantly, they overcome the common assumption of discrete-time models that time spans between measurements need to be equal. This has substantive benefits also for dealing with missing values and real-world observations patterns where, e.g., data on mental health is available but missing for stressors within one observation. On the other hand, deep generative models like Variational Autoencoders (VAEs) learn the data generating processes in a low-dimensional latent space. VAEs allow drawing samples from the learned distributions to generate multiple, differing, and likely samples of missing observations.

This reflects the very recent trend in machine learning and scientific modeling to merge the advantages of both fields. Most notably, combing deep learning with dynamic modeling promises to significantly lower the sample size requirements, which usually accompany deep learning algorithms. This advantage suits resilience data particularly well since many thousands of longitudinal training examples will not be available to resilience research for the foreseeable future. At the same time, deep learning allows to learn arbitrarily complex functions; this is a significant advantage compared to differential equations and classical statistical approaches. An additional advantage of our approach—particularly relevant to subject matter researchers—is that the parameters of the differential equations are estimated freely for every respondent, i.e., without constraints regarding their distribution, direction, or strength. This reflects the notion of resilience where individual differences in stressor response are of primary interest but do not necessarily follow a predefined distribution.

To accomplish this, we combined advanced dynamic modeling techniques with Variational Autoencoders (VAEs) and connected the elements of our algorithm with differentiable programming. For the current version of the algorithm, we made particular use of its capabilities to find low-dimensional representations in the latent space where the data is mapped non-linearly into the latent space. Compared to standard Autoencoders, VAEs model the latent space representations as distributions with a certain mean and variance. The variance captures the uncertainty related to this particular observation. Differentiable programming frees training procedures by allowing us to experiment with all kinds of parameter updates. We found in our extensive test runs that pretraining the VAEs and the ODEnet followed by sequential training of the VAEs and ODEnet, i.e., training each for one epoch before iterating again, results in the lowest losses.

1.2 The algorithm

An overview of the algorithm is depicted in Figure 1. The arrows indicate how the data flows through the algorithm. The suggested algorithm has two essential parts, dimensionality reduction, and trajectory estimation; both tasks are done with neural networks (upper row).

We trained two VAEs—one for mental health and one for daily hassles—to estimate the one-dimensional distributions in the latent space for each observation. Subsequently, we harness the temporal and individual structure and extract a variety of trajectory indicators. This information is used as inputs to a feed-forward neural net; this ODEnet is trained to provide ODE parameters that minimize the squared distance of the ODE trajectory, and the observation mapped into the latent space.

More detailed, the VAEs reduce the number of dimensions to one for each VAE (a) with the standard Evidence Lower Bound (ELBO) loss function with a Poisson log-likelihood which reflects the count character of the data (see figure 1). Thereby, we map the observations into the latent space using a VAE whose encoder and decoder weights (θ, ϕ) were trained minimizing the ELBO $\mathcal{L}(\theta, \phi; x)$ to reduce the dimensions of mental health problems $x_{i,j,t}^{ghq}$, $i \in \{1, \dots, N\}, j \in \{1, \dots, J\}, t \in \{1, \dots, T_i\}$ and stressors $x_{i,j,t}^{dh}$, $i \in \{1, \dots, N\}, j \in \{1, \dots, J\}$, $t \in \{1, \dots, T_i\}$ with $J \in \{28{,}58\}$, measured by the General Health Questionnaire 28 (ghq; where higher values indicate more mental health problems) and a battery of 58 daily hassles (dh; which is a subset of the hassles scale) to one dimensional latent representations $z_{i,t}^{ghq} \sim N(\mu, \sigma)$ and $z_{i,t}^{dh} \sim N(\mu, \sigma)$ each.

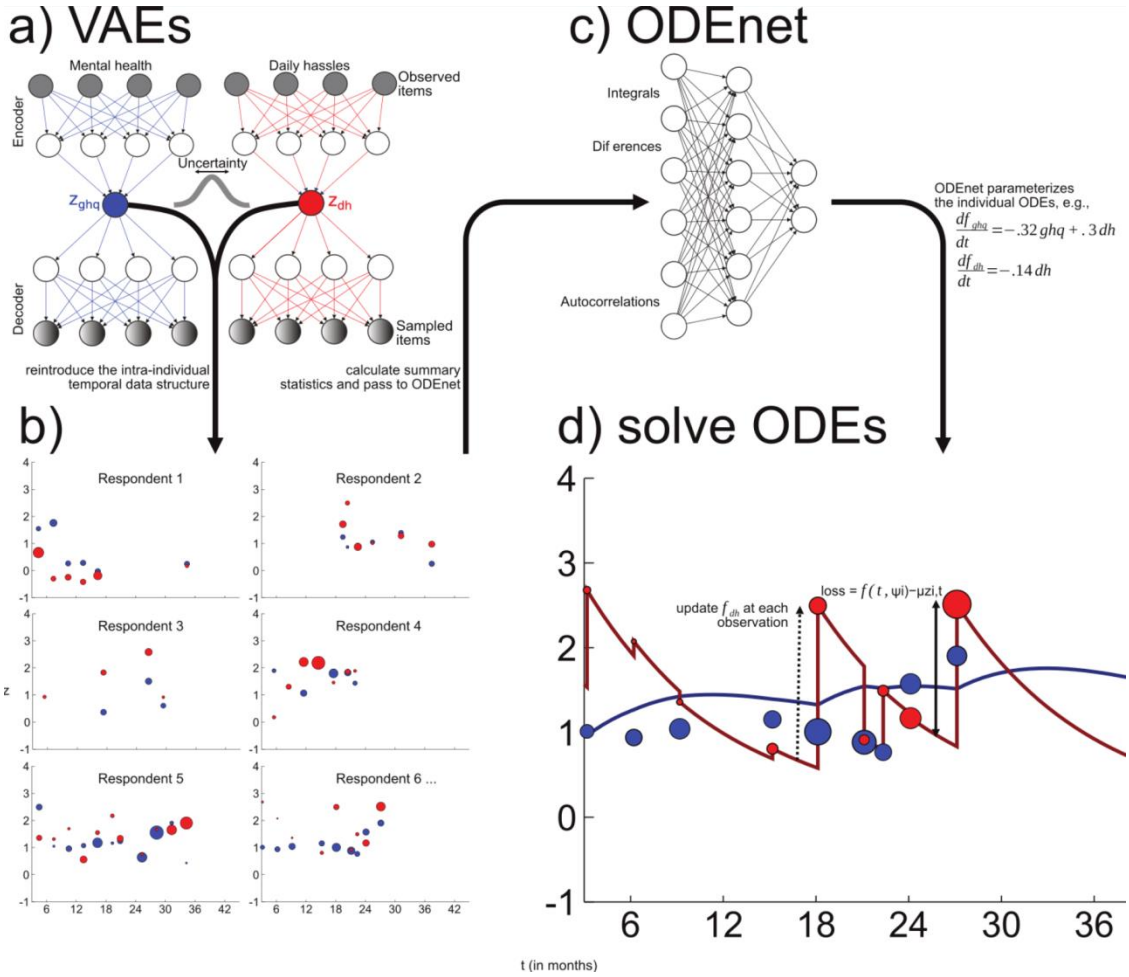

Figure 1: Overview of algorithm

After learning the location of the observed values in the latent space, we reintroduce the intra-individual temporal data structure applying the logic of a time plot (with z on the y-axis and continuous time on the x-axis) to each individual with at least two successful observations (b) in both groups of variables. This allows us to extract a variety of trajectory indicators, e.g., an estimate of the integral of both trajectories with a step function as well as the difference of the first and last z-value (see Table 1 for a full list). These trajectory indicators serve as inputs to the standard feed-forward ODEnet (c), which provides the ODE parameters. It was left to gradient descent and backpropagation to find the best combination of these inputs. The parameters of the ODEnet were trained to minimize the squared distance between the ODE trajectory and the observation evaluated precisely at the point in time when the respondents provided the data. The comparatively simple ODE system—which governs how the trajectories change and interact—is designed with the limited data size in mind.

Figure 1 d) shows a solved ODE which connects the means in the latent space. We use a system of ODEs to model the trajectories of mental health and stressors in the latent space. The exact design of such an ODE system is a crucial modeling decision since it governs how each component changes and, accordingly, requires domain expertise. Given our sample size and sampling frequency, we work with a rather simple ODE system, which reflects the basic notion of resilience to keep the number of estimated ODE parameters η_i small. Specifically, we used:

$$\frac{df_{z_i^{ghq}}}{dt} = \eta_{i,1} z_{i,t}^{ghq} + \eta_{i,2} z_{i,t}^{dh}$$

$$\frac{df_{z_i^{dh}}}{dt} = \eta_{i,3} z_{i,t}^{dh}$$

This structure is informed by the theoretical model of resilience, where $z_{i,t}^{ghq}$ and $z_{i,t}^{dh}$ change according to their own value, but also in response to the other. At each realized measurement, the value of the integrator of the latent daily hassles trajectory $f_{z_i^{dh}}$ is updated to the actually observed value $z_{i,t}^{dh}$. This reflects the theoretical notion that the experienced stressor levels are only partly endogenous. Rather, exogenous forces—which are not captured in this study—lead to abrupt changes in the stressor load.

The benefits of such an ODE system in comparison to discrete-time models like regression are manifold. For example, ODEs acknowledges the irregular sampling intervals of the data. We gained vital insight by our close cooperation with WP3 both regarding their domain expertise and data supply. Importantly, WP3 provided us with data from MARP and LORA study, which is very similar to what we expect in DynaM-OBS and DynaM-INT. In MARP and LORA, longitudinal observations stem from online assessments—where respondents were asked every three months to login and complete a questionnaire—which not all of them did in a timely manner. Furthermore, respondents did not respond at all or only to one of the two item batteries. We expect the temporal responding patterns to be much more irregular in DynaM-OBS and DynaM-INT, given the even more ambitious sampling scheme. In contrast to discrete-time models, differential equations can take all available information into account and overcome problems of classical statistical approaches elegantly by integrating

the system at the precise points in time. Furhermore, we had fruitful discussions with WP1 who are experts in the field of dynamical modeling.

An additional advantage of our approach is that the ODE parameters for every respondent are estimated freely, i.e., without constraints regarding their distribution, direction, or strength. This is accomplished by estimating the parameters of the ODE system $\eta_{i,1} - \eta_{i,3}$ with the ODEnet—a feed-forward neural network—with 16 summary statistics as inputs (see Figure 1 c and Table 1 for details). Importantly, the summary statistics are selected with data availability in mind. Since some individuals just started recently to provide data, all summary statistics need to be computable with only two observations of both item groups, which is the minimum requirement to be included in the longitudinal analysis.

The ODEnet provides the individual parameters of the ODE system $\eta_{i,1-3}$ as outputs. Accordingly, the overall structure of how mental health and daily stressors change is the same for all respondents and dictated by resilience research. Also, the ODEnet learns just one set of parameters. However, the summary statistics included in the ODEnet differ from person to person. The ODEnet is trained to minimize the squared difference between the trajectories $f_{z_i^{ghq}}$ and $f_{z_i^{dh}}$ and the observed means of $z_{i,t}^{ghq}$ and $z_{i,t}^{dh}$ with backpropagation. Accordingly, the individual parameters do not obey any predefined distribution and can be estimated in the absence of any distributional restriction coupled to the grand mean (which is what, e.g., random effects models would do) to minimize the loss function best.

We implemented this algorithm in the programming language `Julia` which is known for its performance and readability.

1.3 Methodological details

We choose VAEs to find the lower-dimensional representation because they provide a flexible framework with numerous potential extensions and applications. VAEs consist of a recognition model and a generative model. The purpose of training the recognition model (aka inference model or encoder) is to find the variational parameters ϕ of a neural net to approximate the posterior distribution of the latent variable z given the inputs x, i.e., $q_\phi(z|x)$. The parameters of the generative model (aka decoder) θ are trained to increase the log-likelihood of the inputs given random samples from the posterior distributions. To train the recognition and generative model simultaneously, we maximize the evidence lower bound (ELBO) of the marginal log-likelihood

$$\log p(x) = \sum_{n=1}^{N}\sum_{t=1}^{T}\log p(x) \geq \sum_{n=1}^{N}\sum_{t=1}^{T}\mathbb{E}_{q(z|x)}\left[\sum_{p=1}^{P}\log p(x|z)\right] - \sum_{n=1}^{N}\sum_{t=1}^{T}\mathrm{KL}\ (q(z|x)||p(z))$$

where the first term of the right hand is the expectation of the log-likelihood of x given z with respect to $q(z|x)$. The Kullback-Leibler divergence (D_{KL}) penalizes deviations of the posterior from the prior. For computation, we plug in the Poisson log-likelihood and the closed form of D_{KL} for a Gaussian prior and posterior. Thereby, our training objective becomes

$$\mathcal{L}(\theta, \phi; x) = \frac{1}{2}\sum_{i=1}^{N}(1 + \log((\sigma_i(x))^2) - (\mu_i(x))^2 - (\sigma_i(x))^2)\sum_{i=1}^{N}\lambda - x \times \log(\lambda)$$

where μ_i and σ_i are the mean and standard deviation of the observation, and λ is the expected value of the Poisson distribution. The Poisson distribution does not optimally suit the data since we found moderate levels of overdispersion for some of our items. However, the Poisson distribution has the computationally efficient assumption of a coupled expected value and dispersion. This assumption eases up computation and needs considerably fewer parameters to estimate; it suits most of our items reasonably well. We usd J nodes and $tanh$ activation functions in the mid-layers. In the final layer, we used a $ReLU$ activation function to strictly pass non-negative values to λ. All neural networks are trained with Flux.jl and ADAM.

1.4 Individual ODE parameter estimation with the ODEnet

The parameters of the ODEnet τ were trained to minimize the squared difference of the trajectory at the precise point in time t and the mean of the latent space distribution $\mu_{z_{i,t}}$.

$$\mathcal{L}_{ODE}(\tau, \eta_i) = \sum_{n=1}^{N}\sum_{t=1}^{T}(f_{z_i}(t, \eta_i) - \mu_{z_{i,t}})^2$$

The ODEnet is a separate feed-forward neural net with two layers and 16 inputs; the mid-layer and end-layer have three nodes (due to data size considerations). As inputs, we use a mixture of integrals, first and last observations, differences, and autocorrelations (see Table 1).

1.5 Details on training and data structure

When coding this algorithm, we made use of differentiable programming. This provides substantial flexibility when training the neural networks. In our empirical investigations with MARP data, we found that the best results were obtained with a combination of separate pretraining of the VAE and ODEnet. Subsequently, we compared joint training `fit_node(data; train_mode = "joint")` of all involved components vs. a sequential strategy `fit_node(data; train_mode = "sequential")` where we trained the VAE components separate from the ODEnet in one epoch. This leads to further decreases in the loss components also compared to the least ambitious training mode of separately training the VAE and the ODEnet `fit_node(data; train_mode = "ode_only")`. In this analysis, we pretrained both VAEs with a learning rate of $\alpha = 5^{-5}$ for 80 epochs. Then, we pretrained the ODEnet separately for 50 epochs with $\alpha = 1^{-6}$. Then, we sequentially trained both nets in the same epoch with $\alpha = 1^{-6}$ for 100 epochs.

To deal with unit non-response, $\mathcal{L}(\theta, \phi; x)$ and $\mathcal{L}_{ODE}(\tau, \eta_i)$ are only evaluated at actually occured measurements. To feed our networks, we impose a artifical grid strucutre on our data and impute the missing values with arbitrary values. We also generated two weight

vectors $\mathbf{w} \in \{0,1\}$ of lenght T with with $\mathbf{w} = 1$ for observed values and $\mathbf{w} = 0$ for the provisionally imputed observations. All loss terms above are multiplied with this weight vector before they are summed up.

2 Tables and other supporting documents

The ODEnet is a separate feed-forward neural net with two layers and 16 inputs; the mid-layer and end-layer have three nodes (due to data size considerations). As inputs, we use a mixture of integrals, first and last observations, differences, and autocorrelations (see Table 1).

Input	Category	Scaled by
First obs of ghq	O	1
First obs of dh	O	1
First obs ghq − first obs dh	D	1
First obs ghq − last obs ghq	D	1
First obs ghq − last ghq	D	1
First obs dh − last obs dh	D	1
First obs dh − last obs ghq	D	1
Integral of ghq	I	10
Integral of dh	I	10
Integral of ghq^2	I	10
Integral of dh^2	I	10
Integral of ghq (absolute value)	I	10
Integral of dh (absolute value)	I	10
Mean of Autocorrelation ghq	AC	100
Mean of Autocorrelation dh	AC	100

: Table 1: Details of ODEnet inputs

3 Conclusion

In this report, we described our open source code for preprocessing longitudinal data from psychological resilience studies, as we can expect in both DynaM-INT and DynaM-OBS. We presented a dynamic modeling approach using and extending state-of-the-art deep generative models. [1] More specifically, we adapted the standard VAE—which is usually used with Bernoulli variables—to a Poisson distribution that captures the empirical data

[1] Technically, nothing prevents us from replacing the VAE distributions with scores of more conventional dimensionality reduction techniques. We would lose, however, the quantification of uncertainty, which is characteristically for the VAE. Furthermore, our model is trained jointly, i.e., dimensionality reduction and trajectory estimation can inform each other. This joint training provides an optional advantage of this model compared to all other methods (to our best knowledge).

reasonably well. And we learned person-specific, distribution-free parameters to capture the trajectories in the latent space with the ODEnet. Both parts of the algorithm, the VAEs and the ODEnet can be trained separately or jointly harnessing the benefits of differentiable programming.

We argued that differential equations are capable of overcoming the known difficulties of most statistical and machine learning methods which usually accompany observational, longitudinal data of resilience studies. Differential equations take into account the irregularly spaced observations and are not affected by missing observations. We combined it with VAEs for several reasons. A widely-known advantage of such deep learning algorithms is their capability to provide non-linear transformations when mapping data into the latent space. Since these lower-dimensional representations are modeled as distributions, they already entail and quantify the uncertainty of these representations. We will extend this algorithm by integrating data at different time scales and more complicated ODE systems.