

Generative models for the trajectories of slow-progressing mobility diseases following medical interventions

Ferdinand Legros, SUNet: flegros

I. INTRODUCTION

CEREBRAL palsy, a medical condition that affects mobility, evolves over several years. Progression varies according to the health and characteristics of the patient. Clinicians attempt to alleviate the symptoms through surgery interventions. Since some surgical procedures are heavy, it is key to quantify the positive impact that they may bring about in the patient's state. Thus the availability of a robust predictor for disease trajectory will help clinicians to determine whether surgery intervention is appropriate for a patient at a given time.

The state of a patient is commonly summarized using their Gait Deviation Index (GDI) [1], which is assessed by clinical trials and examinations involving movement tracking. GDI ranges from 0 to 100, 100 indicating a completely normal state and 0 the worst possible state. In the patient data we study, GDI typically lies in [45, 100]. Our goal is to forecast GDI in the future given different surgery scenarios. The theoretical objective of this project is to understand how to best handle short irregular time series - the time series of patients' visits.

We implemented (i) several discriminative Machine Learning models (ii) two fully observable Bayesian Networks: one with continuous variables, one with discretized variables; (iii) three Recurrent Neural Networks of varying number of layers and nodes. The best performance is a Mean Squared Error of 40.5 and Mean Absolute Error of 4.98, it is provided by a 2-temporal-step Linear Gaussian Bayesian Network. The next step of the project is to incorporate the hidden information of the patient disease's severity.

II. DATA

A. Raw data

The data is provided by Stanford Mobilize Center <http://mobilize.stanford.edu/>. It includes information of 6000 clinical visits. For each patient, available data is

- General information: height, age, weight, etc.

- 2 to 10 clinical tests spread irregularly in time.

Each clinical examination includes data about the two sides of the patient, i.e. right and left.

- Numerical health indicators: measure of strength, indicator of reactivity for different muscles
- 11 time series corresponding to 11 angles describing the patients movements during a test assessing the patients ability to walk.
- GDI at the time of the examination

The data was transformed by researchers of Mobilize to a flat table. In this table, each row corresponds to the assessment of one side - left or right - of a patient during a given medical visit. Relevant features - min/max, length of cycle... - were extracted from time series. Below is the typical GDI trajectory of a patient over the years, from their first visit.

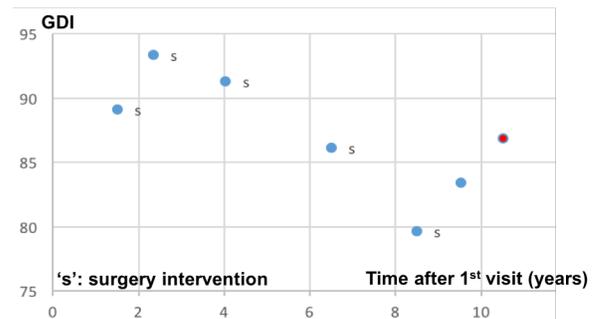


Fig. 1 Patient GDI trajectory

Our goal is to predict the future GDI of the patient at a given time - in red. We observe that the number of surgeries underwent by the patient is very high.

B. Preprocessing

Working with bayesian networks requires to select a small number of variables, so we aggregated some of the information available. We summarized the surgery information in a single binary variable (= 1 if any kind of surgery was performed between the two visits). We averaged the indicators of the two sides for each patient.

This is coherent with the choice of aggregating the surgery variable to a binary variable. We transformed the data table so that each row represents a pair of successive visits for a patient.

For discriminative ML models, we selected variables with a manually specified threshold on the covariance with the target variable of 0.2. This leads to a subset of 11 input variables. For the discrete BN, we manually isolated a subset of 5 variables with the help of a researcher from Mobilize. This subset balances interpretability and correlation with target variable. The subset comprises the following variables:

- Age
- motControlScore: score summarizing gait time series data
- strengthScore: score summarizing strength examination
- hadPrevSurgery
- GDI_t and GDI_{t+1} : GDI at visits t and $t + 1$

Let us note that we deliberately chose a simple variable selection technique since the goal of the project is to understand how to handle time. This allows us to explore rapidly many different options, incorporating more variables may come as a next step. Eventually, we added the temporal gap between current step t and predicted step $t + 1$ as a variable.

We standardized the data for discriminative ML models. For the discrete BN models, we discretized data. GDI was discretized into 12 5-points bins ([40,45], [45,50[,...]). Other variables were discretized in four quartiles.

III. LITERATURE REVIEW

Bayesian networks are widely used for healthcare applications, and they have been applied to treatment outcome prediction. Articles mostly use discretized variables in this setup, and the variables are usually derived from expert knowledge. For instance, continuous variables may be discretized by comparison to a threshold specified by a physician. We are interested in two types of models.

First come static models, which predict the outcome of a single treatment operation at a given time given patient information. In [3], Hoot et al. in specify a structure to predict 90-day survival chance of patients after liver transplant. Sesen et al. in [2] run structure learning to determine best possible treatment among 11 possibilities to cure lung cancer and predict the 1-year survival rate of patients. In [4] Jung et al. also use structure learning for lung cancer outcome prediction.

Second, we are interested in dynamic models which can take advantage of the sequence of visits for each patient. Cai et al. in [5] predict the number of remaining days in hospital in intense care unit with a Dynamic Bayesian Network (DBN). In [6], Watt et al. use a DBN to predict a binary variable "knee pain" relevant to the diagnosis of osteoarthritis. Wang et al. in [7] model lung disease state by a discrete variable and use Markov Jump Processes to learn a transition matrix that depends on the time spent between two medical trials.

The first challenge of our project compared to those existing models is that we want to predict a continuous variable. Second, even if our variables were first processed with the domain knowledge of Mobilize researchers, we do not have easy access to insights of a physician in order to tailor those variables to our problem. Finally, some of those approaches were possible because of a large dataset - [2], [7] count 100k+ patients - whereas our database contains 6000 medical examinations.

IV. DISCRIMINATIVE MODELS

We implemented linear regression, random forest and simple fully connected neural networks approaches to first build a static model that can predict GDI at time of next visit from information of the current visit and surgery information. Neural networks used the ReLu activation function. Neural network 1 (NN1) is composed of a single layer of 20 nodes. Neural Network 2 (NN2) is composed of an input layer of 12 nodes and a hidden layer of 6 nodes. The processing used is the 0.1 covariance threshold (12 variables) evoked above and standardization to 0 mean normalized variables. We performed 5 times 5-fold cross validation and we averaged the errors, we report them below. Since our training dataset has 2250 examples, a 20% test size seems to be a good way to evaluate robustly our methods while keeping enough training data.

Model	MSE	MAE
Lin. Reg.	53.32	5.78
RF	53.25	5.80
NN1	52.94	5.78
NN2	52.98	5.79

We observe that the differences between those models are not significant. This sets a first result on a static model. We do not dive in error analysis as we are principally interested in the results given by the Bayesian Networks and the Recurrent Neural Networks.

V. BAYESIAN NETWORK MODEL

A. Discrete BN

We built a discrete BN to predict (i) whether surgery should or should not be performed and (ii) GDI_{t+1} , given fully observable data at time step t . This is the standard method for a medical diagnosis tool. For implementation we used the Python library libpgm [8]. Architecture of the BN is given in Fig. 2. We split the data into 80% train data and 20% test data. Since this model will prove clearly not satisfactory, we do not perform further analysis.

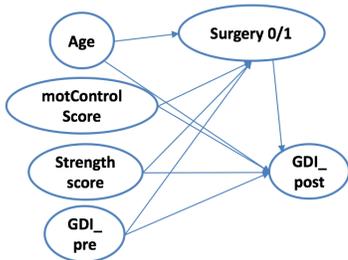


Fig. 2 Discrete BN structure

We learned parameters with Maximum Likelihood Estimation. In order to test the model, we performed exact inference given limited evidence. We computed $P(\text{Surgery} \mid \text{Age}, \text{motControlScore}, \text{strengthScore}, GDI_t)$ and $P(GDI_{t+1} \mid \text{Age}, \text{motControlScore}, \text{strengthScore}, GDI_t, \text{Surgery})$ for each instance of the dataset. The prediction for a variable is the most probable value for this variable. When there was a tie in the probabilities, we chose the most conservative choice: 1 for surgery, and the value of GDI_{t+1} closest to the one of GDI_t . We also implemented the results with tercile instead of quartile discretization, and obtained similar results.

Surgery prediction

In test data, 75% of observations of hadPrevSurgery were equal to 1, so we examine the confusion matrix to analyze predictor performance.

True / Pred	0	1
0	10	80
1	59	270

We see that the BN fails to identify cases where hadPrevSurg = 0. Given those limited results, we discussed about the feasibility of treatment selection for cerebral palsy with Mobilize researchers who had domain experience. Such task is particularly complex. In fact, treatment alleviates rather than cures the disease, so different surgeries could be relevant for a patient at

a given time. Moreover, using past surgeries as training would imply to assume that surgery was applied only when it was the best choice possible, which is one more approximation. Thus, we decided to focus on the prediction of GDI and not to pursue the classification task.

GDI prediction

GDI prediction outputted the right 5-point GDI_post bin in 24% of test cases, and had MAE = 8.05. This error is significantly higher than the previous discriminative models. An important feature of the result is that out of the 419 test examples, 173 predictions had more than one most probable value. This is due to the fact that GDI_{t+1} has too many parents, so the number of observations is not sufficient to estimate parameters well. In fact, when examining Conditional Probability Tables (CPTs), we noticed that many entries had equal probabilities.

We are confronted with the problem that CPTs have too many entries relatively to the available data. The reason for this is that GDI has 13 outcomes: [45-50[, [50,55[,..., [95,100[. In literature, a high number of target outcomes is compensated by a massive dataset - 120k patients in [2]. Another possibility is to reduce the number of target outcomes. In [4] Jung et al. measure surgery outcome by a simple binary variable "Success" or "Failure". Neither of those options is satisfactory to us. So we chose to switch to a model with much less parameters and which better fits regression problems: the Linear Gaussian Bayesian Network.

B. Linear Gaussian BN

LGBN Model

The most common continuous variables BN is the Linear Gaussian BN. It assumes that all variables have a gaussian distribution. A variable mean is assumed to be a linear combination of the values of its parents. It is also possible to incorporate discrete variables to this setup. Then, for every outcome of a discrete parent - surgery 0 or 1, the child node GDI_{t+1} has a different set of gaussian parameters.

So far, we have implemented a "1-step" model, which predicts GDI_{t+1} from information at time step t . Since the LGBN did not limit the number of input variables contrary to the discrete BN, we also built a "2-step" model which predicts GDI_{t+1} from steps t and $t - 1$. The resulting graph is the following. Dashed arrows are only included in the 2-step model.

Parameter learning was performed by MLE. Since libpgm does not comprise BN with discrete and

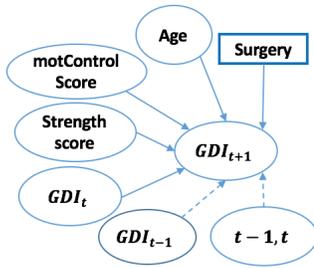


Fig. 3 LGBN structure

continuous node, we switched to the library Bayesian Network Toolbox in Matlab. [9]

LGBN Results

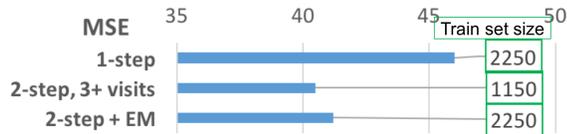


Fig. 4 LGBN results

We evaluated the model with 5 times 5 fold CV. The 1-step and 2-step models perform respectively ≈ 7 MSE points and ≈ 12 MSE points better than the baselines, which is a significant improvement. An important insight is that adding an additional time step improved the prediction. Moreover we gain interpretability compared to the discriminative models since we can compute confidence interval for GDI_{t+1} thanks to the estimated variances, respectively 44.2 and 40.7. This would be valuable to a physician.

Missing data and Expectation Maximization

Let us note that the training set for the 2-step model is smaller than the one of the 1-step model. Indeed, a training example for the 2-step model involves a triplet of visits $t-1, t, t+1$ whereas an example for the 1-step model involves only a pair of visits $t, t+1$. So the 2-step model must be trained on the dataset of patients with at least 3 visits - denoted by 3+ visits in Fig. 3.

In particular, we cannot use GDI_1 to train the 2-step model since it would require GDI_0 and GDI_{-1} . The latter denotes the GDI before the visit 0, and thus is not available. Usually, in time series prediction, this is not an issue since sequences are long enough so that the training set size is high enough after getting rid of the first points. Contrarily, sequences in our data are very short: <10 visits, and for most patients <5 visits. So

this makes a great difference in the training set size, as shown in Fig. 3: 2250 vs 1150 training examples.

We tried to address this problem by using Expectation Maximization. EM is a common way to deal with missing data. We used EM to guess GDI one year before the first visit. Even though the mean gap between two visits is higher - around 1.5 year, some patients are as young as 3 or 4 years old, so 1 year seemed more reasonable.

The EM algorithm converges in ≈ 10 iterations. We report the performance in Fig 4. The "2-step + EM experiment" consists in training the 2-step model with EM on the full dataset, and testing the performance of a test set comprising only patients with 3+ visits. Indeed, testing on patients with 2+ visits would have provided a biased result since patients with only a pair of visits have GDIs harder to predict - the gap between two visits is usually wider. We observe that using EM does not improve performance. Our hypothesis is that the guess of EM is too approximate to add information to the problem. In a complete error analysis, we could have compared the EM guess with the actual GDI on observable examples. Since EM did not seem promising, we chose to move forward in the project.

VI. RECURRENT NEURAL NETWORKS

RNN Models

Recurrent Neural Networks are a common method to predict time series. They have been used for diagnosis in a medical context to predict the onset of disease given time-irregular patient history, in [10] for instance. The typical length of input time series is above 100 points. We wanted to understand whether RNNs were relevant for our 5-10 points short sequences. RNNs were applied to 10 points sequences in [11]. Long Short Term Memory networks are a popular choice for time series prediction. However, since our sequences are very short, vanishing gradient should not be an issue. Moreover, with our scarce data, the number of parameters should be kept low. That is why, as in [11], we used Vanilla RNNs.

We used the 0.2 correlation threshold variables and temporal gaps between visits as input variables. In order to deal with the varying length of our patients' visits sequences, we used zero padding, as advised in [12]. We implemented the RNNs using Keras library [13]. We used the RMSProp optimization algorithm We built three models of various shapes that we report below.

Model	# Layers	# Nodes
Model 1	1	20
Model 2	2	20, 10
Model 3	2	50, 20

RNN Results

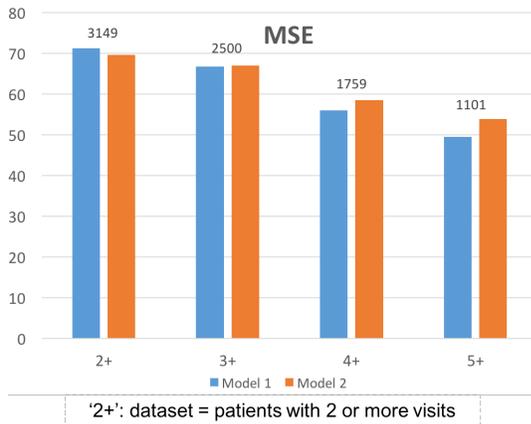


Fig. 5 RNN results

The advantage of RNNs compared to the previous methods is that they take into account the whole visit sequence instead of just one or two previous time steps. Therefore, they can be used to answer the following question: *Is there a long-term coherence in patients' medical histories?* This question is obvious in a usual diagnosis setup. However, for Cerebral Palsy data, we saw that patients undergo many surgeries. A surgery has such a strong impact on a patient that we could think that it breaks the coherence of a patient history.

To investigate this issue, we computed the 5-CV MSE for datasets comprising patients with at least 2, 3, 4 and 5 visits. The results are reported in Fig. 5. We report the corresponding number of visits in the training set on the top of the MSE columns.

First, we observe that overall, RNNs perform worse than our simpler LGBN model. The best performance is ≈ 50 MSE, obtained on the dataset of 5+ visits patients. This is 10 points higher than the 2-step LGBN model. Therefore it seems that even a simple RNN is a too cumbersome architecture for our problem. Building a more sophisticated model with more layers or more nodes is unlikely to help. Indeed, the results show signs of overfitting. We get the intuition of overfitting when we see that as we restrict the datasets to 3+, 4+, 5+ visits patients, the more complex RNN performs worse than the 1-layer RNN. Overfitting is clear when examining the MSE on train vs test sets. The typical MSE gap between train and test on 3+visits dataset is ≈ 5 for model 1, ≈ 15 for model 2, ≈ 20 for model 3. This is amplified when the dataset is even more restricted.

Second, we observe a temporal tendency in the results. We should compare it with control tests on the restricted datasets to make it sure. If it proves robust, then it

would mean that patient medical history is coherent, and that you can extract valuable information from the whole sequence. This is valuable information that would influence the design of hidden nodes in our bayesian network. Thus, even though RNNs do not provide the best performance, they provided an insight that will help the design of our prognosis tool.

VII. NEXT STEPS

The next steps of the projects aim at improving (i) performance and (ii) interpretability of the models. The next steps mainly deal with the Bayesian Network approach since the RNNs seemed not to be a great fit to the problem.

A. Possible performance improvements

To improve our model's performance, we could first test different problem frameworks. The framework we used allowed us to select the best performing model, that we can now fine-tune. In the beginning, we chose to aggregate the information of the two sides of patients. Instead we could try to predict the GDI of both sides and differentiate sides of surgery too. We could perform fine feature selection, with information gain for instance. An important next step when more variables are taken into account is structure learning. The structure that we manually defined out of common sense and Mobilize researchers' insights was satisfactory until now. However, with more variables, this structure may prove too simple and automatic learning may prove more appropriate.

Another challenge is the fine modeling of the irregular time steps. Until now, we consider the gap between two visits as an additional variable. Other methods take advantage of the structure of Bayesian Networks to discount values over time. This is what is achieved with Markov Jump Processes in [7].

B. Improving model interpretability

Model interpretability is a key aspect of any diagnosis tool. The main next step is the introduction of hidden nodes in the network. Such hidden nodes can (i) capture correlations between variables and (ii) provide meaningful hidden information to physicians. The hidden node we would like to introduce is a patient state node. Such a hidden variable would be discrete, as in [7], and would summarize the severity of the disease. This information could give an overview of the patient's disease progression to physicians. Moreover, as correlations are captured, performance may be improved.

REFERENCES

- [1] Schwartz et al., "The Gait Deviation Index: a new comprehensive index of gait pathology", *Gait Posture*, 2008 <https://www.ncbi.nlm.nih.gov/pubmed/18565753>
- [2] Sesen et al., "Bayesian Networks for Clinical Decision Support in Lung Cancer Care", *PLoS ONE* 8(12): e82349, [doi:10.1371/journal.pone.0082349](https://doi.org/10.1371/journal.pone.0082349), 2013
- [3] Hoot et al., "Using Bayesian Networks to Predict Survival of Liver Transplant Patients", *AMIA 2005 Symposium Proceedings*, 2005
- [4] Jung et al., "Bayesian network approach for modeling local failure in lung cancer", *Phys Med Biol*, 2011 March 21; 56(6): 16351651. [doi:10.1088/0031-9155/56/6/008](https://doi.org/10.1088/0031-9155/56/6/008), 2011
- [5] Cai et al., http://c.ymcdn.com/sites/www.hisa.org.au/resource/resmgr/bigdata2014/Xiongcai_Cai.pdf
- [6] Watt et al., "Evaluation of a Dynamic Bayesian Belief Network to Predict Osteoarthritic Knee Pain Using Data from the Osteoarthritis Initiative", *AMIA 2008 Symposium Proceedings*, 2008
- [7] Wang et al., "Unsupervised Learning of Disease Progression Models", *KDD14*, <http://dx.doi.org/10.1145/2623330.2623754>, 2014
- [8] <http://pythonhosted.org/libpgm/>
- [9] https://www.cs.utah.edu/~tch/notes/matlab/bnt/docs/bnt_pre_sf.html
- [10] Lipton et al., "Learning to diagnose with LSTM recurrent neural network", *ICLR 2016*
- [11] <http://simaaron.github.io/Estimating-rainfall-from-weather-radar-readings-using-recurrent-neural-networks/>
- [12] https://github.com/Vict0rSch/deep_learning/tree/master/keras/recurrent
- [13] <https://keras.io/>