Hierarchical Denoising of Ordinal Time Series of Clinical Scores

Jonathan Koss, Sule Tinaz, and Hemant D. Tagare, Senior Member, IEEE

Abstract—Clinical scores (disease rating scales) are ordinal in nature. Longitudinal studies which use clinical scores produce ordinal time series. These time series tend to be noisy and often have a short-duration. This paper proposes a denoising method for such time series. The method uses a hierarchical approach to draw statistical power from the entire population of a study’s patients to give reliable, subject-specific results. The denoising method is applied to MDS-UPDRS motor scores for Parkinson’s disease.

Index Terms—Time series analysis, Parkinson’s disease, Machine learning, Clinical neuroscience

I. INTRODUCTION

CLINICAL symptoms of neuropsychiatric and neurodegenerative disorders are often evaluated using ordinal ratings, or scores. For example, in Parkinson’s Disease (PD), patients exhibit several motor disabilities, and each disability is rated using the ordinal scores: Normal, Slight, Mild, Moderate, Severe.

In longitudinal clinical studies, ordinal scores are collected at periodic intervals, producing one ordinal time series per symptom per patient. These time series are a rich source of information about how the disease is progressing, but their analysis is quite challenging. There are several reasons for this:

First, the time series are usually very noisy. Noise is partly due to the stochastic nature of the symptoms, and partly because it is difficult to differentiate between neighboring categories such as Slight, Mild, and Moderate [1]. Noise can be so severe that it masks the underlying progression of the disease. Real-world examples of this are given in Section V.

Second, the data in the time series are ordinal. Thus, commonly used vector space techniques, such as additive noise models, cannot be used for creating denoising algorithms [2].

Third, the time series are short because most longitudinal studies have a limited time duration. A typical time series might have 10 samples. Fitting a statistical model to such a short time series becomes problematic, especially as the number of parameters in the model approaches the number of samples.

Finally, the time series are unevenly spaced and often have missing data.

A fine-grained and nuanced understanding of disease progression is possible if a method can be developed to denoise the ordinal time series of each symptom, taking all of the above-mentioned concerns into account. The goal of this paper is to propose such a denoising method.

The statistical model behind our method accounts for all of the above-mentioned characteristics of clinical scores. It explicitly takes the ordinal nature of the time series into account by using ordinal time regression. Further, the time regression is situated in a hierarchical framework, so that patients whose time series proceed similarly draw statistical power from each other, overcoming the limitation of the short duration of each series. And the model allows for uneven time spacing and missing data. In Section V, we show that using this model to denoise clinical time series helps substantially with understanding the progression of symptoms.

There is another important aspect of disease progression: Disease progression is often heterogeneous. That is, different patients progress along different trajectories at different rates. Heterogeneity is usually understood in terms of progression subtypes, which are prototypes of different trajectories of progression. The hierarchical design of our model automatically provides progression subtypes.

We evaluate the performance of our method by using synthetic data and also by applying the method to real-world PD data. The PD data come from the Parkinson’s Progression Marker’s Initiative (PPMI). Details of the PPMI data are given in Section II.

This paper is meant to be methodological. Our goal is to report a new denoising method and to demonstrate its utility by applying it to some aspects of PD. In forthcoming publications, we hope to report more comprehensive insights into PD and other diseases using our method.

The rest of the paper is organized as follows: Section II contains a literature review and background information on PD; Sections III and IV describe our model; Section V contains results of applying our method to synthetic and real-world data; Section VI contains a discussion and a conclusion.

II. BACKGROUND AND LITERATURE REVIEW

A. Ordinal Regression Models

Ordinal regression models have a long history, originating with the Cox model [3] from over 50 years ago. A complete review of ordinal regression models can be found in [4].
Textbooks such as [5] and [6] contain detailed explanations of ordinal data analysis.

Modern ordinal models can be divided into three categories [4]. In the first category are naïve approaches which do not account for the ordinality of the data. They either treat clinical scores as cardinal numbers or as classes but without any order. These approaches then apply a traditional regression technique such as support vector machines (SVM) [7], [8]. Methods like [9] use a categorical classification technique but then alter the cost function to assume label distances corresponding to the cardinal values of the categories.

The second category of models decomposes the ordinal-regression problem into several binary regression problems which can be solved separately with multiple models, typically using SVMs [10] or using a single multi-output model such as a neural network [11], [12]. Common choices for decomposing the problem are comparing all pairs of neighboring categories, e.g., for ordinal classes \( c_1, \cdots, c_4 \) comparing \( \{c_1 \text{ VS } c_2\}, \{c_2 \text{ VS } c_3\}, \{c_3 \text{ VS } c_4\} \), or comparing a single category with all the following (or previous) categories \( \{c_2 \text{ VS } c_1\}, \{c_3 \text{ VS } c_1, c_2\}, \{c_4 \text{ VS } c_1, c_2, c_3\} \).

The last category of ordinal-regression models treat the ordinal variable as originating from a continuous underlying latent variable, which is divided into categories by a set of thresholds. One of the early and prominent threshold techniques is the proportional odds model [13], which is also the basis of our model. The proportional-odds model and other related models are discussed immediately below. Other threshold-based models like [14], [15] utilize SVMs while [16], [17] offer neural-network-based approaches.

**Proportional Odds Model:** The proportional odds model extends the linear logistic model [3] to handle more than two categories. For an ordinal response variable \( Y \in \{0, 1, \cdots, C - 1\} \) and an explanatory variable \( X \), the proportional odds model assumes that the log-odds ratio varies linearly with \( X \):

\[
\log \left( \frac{P(Y \leq c)}{P(Y > c)} \right) = \alpha_c + \beta X,
\]

which in turn implies

\[
P(Y \geq c) = \frac{1}{1 + \exp(-\alpha_c - \beta X)}
\]

where \( \alpha_c \) and \( \beta \) are scalar parameters learned from the data.

In the proportional odds model, \( \beta \) is independent of the category, \( c \) [18]. If the data do not meet this assumption, then the partial proportional odds model is likely to be more useful [19], [20].

**Partial Proportional Odds Model:** The partial proportional odds model takes the form

\[
P(Y \geq c | X) = \frac{1}{1 + \exp(-\alpha_c - \beta X - \gamma_c X)}.
\]

In this model, the \( \beta \) term of the proportional odds model is effectively set to \( \beta + \gamma_c \), where \( \gamma_c \) is category dependent. If all \( \gamma_c \approx 0 \), then equation (3) simplifies to the proportional odds model.

The generalized ordered logit model [21] offers a model with similar properties.

### B. Parkinson’s Disease

Parkinson’s disease (PD) is a neurodegenerative disorder that manifests with motor and non-motor symptoms. In the early stages of the disease, the motor symptoms tend to be prominent [22]. The current standard for clinically assessing disease severity in PD is the Movement Disorder Society’s Unified Parkinson’s Disease Rating Scale (MDS-UPDRS) [23]. The MDS-UPDRS has four parts, each consisting of a set of items scored on an ordinal scale of 0, 1, 2, 3, 4 which correspond to Normal, Slight, Mild, Moderate, and Severe ratings. In the first two parts of the MDS-UPDRS, patients and/or caregivers rate symptoms associated with non-motor and motor aspects of daily living, respectively. Part 3 is the motor examination portion of the MDS-UPDRS, where a physician observes a patient performing a number of motor tasks and scores each on an ordinal scale. Part 3 contains 33 ordinal scores, which mainly focus on the cardinal motor features of PD including tremor, rigidity, bradykinesia (slowness of movement), and axial symptoms (e.g., gait impairment, postural instability). Part 3 scores are usually summed to create a total movement score. Higher total movement scores indicate greater overall motor severity. Part 4 assesses motor complications including dyskinesia and motor fluctuations in response to medication.

In this paper we focus on Part 3 scores. Rather than summing up the scores into a total movement score, we analyze scores of the cardinal PD symptoms independently.

PD is a heterogeneous disease with distinct clinical subtypes and progression trajectories [24]. These subtypes are traditionally determined by clinical observations (e.g., tremor-dominant or postural instability/gait difficulty subtypes) [25]. More recent data-driven clustering methods identify the following subtypes: 1) young-age onset and old-age onset subtypes with the latter progressing relatively faster than the former, but both progressing slowly 3) tremor-dominant, and 4) dominance of bradykinesia, rigidity, and postural instability/gait difficulty [26].

### C. Parkinson’s Progression Markers Initiative (PPMI)

The MDS-UPDRS data that we use comes from the de novo PD cohort of the Parkinson’s Progression Markers Initiative (PPMI), a longitudinal study with the goal of finding biomarkers for PD (www.ppmi-info.org) [27].

PPMI contains MDS-UPDRS data for 423 patients who were observed for 8 years after the initial PD diagnosis. MDS-UPDRS scores are available for these patients for up to 17 visits. At the start of the study, visits are scheduled every three months, decreasing in frequency to every six months, and then to once a year. During the visits, MDS-UPDRS scores are gathered in the off-medication state, which is defined as the condition where the patient has not taken medication for \( \geq 6 \) hours, so that their symptoms are not masked by medication.

### III. Denoising An Ordinal Time Series

When a time series takes values in a vector space, noise is usually assumed to be additive and Gaussian. Given a prior for the underlying signal, noisy measurements of the time series...
are used to calculate the posterior distribution of the signal, and the mean of the posterior distribution is taken as the denoised estimate of the signal.

This vector space idea does not carry over to ordinal time series in a straightforward manner because ordinal time series do not belong to a vector space and there is no simple additive noise model for such series. Nevertheless, there is a sense in which generalization is possible. The idea is illustrated in Fig. 1. Assuming 4 ordinal categories for the moment, Fig. 1a shows how the probabilities of observing an ordinal category change with time as the disease progresses. The probability of observing ordinal category \( y = 0, 1, 2, 3 \) at time \( t \) is \( p(y|t, \theta) \), where \( \theta \) is a set of parameters. Thus, \( p(0|t, \theta) \) is initially large, followed by \( p(1|t, \theta) \) being large, etc. The progress of the disease can be summarized by the mean \( \bar{m}(t) = \sum_{y=0}^{3} yp(y|t, \theta) \), which we may take as the underlying “signal” (See Fig. 1a). Any realization of the time series is a sequence of ordinal values at times \( t_1, t_2, \ldots \), drawn from the probabilities \( p(y|t, \theta) \).

Assuming a prior on \( \theta \), we may estimate its value \( \hat{\theta} \) from the observed time series, and use the estimate to obtain a denoised estimate \( \hat{m}(t) = \sum_{y=0}^{3} \hat{p}(y|t, \hat{\theta}) \) of the mean. This is the sense in which we can denoise an ordinal time series of clinical symptoms.

As mentioned in Section I, if the time series is short, we may not get a reliable estimate of \( \theta \). To ameliorate this problem, we use a hierarchical model in which \( \theta_i \) for the time series of the \( i \)th subject is assumed to have a mixture of normals prior (see Fig. 1b). All subjects with similar underlying disease progression are assumed to come from a single component of the mixture. Simultaneously estimating \( \theta_i \)’s and the parameters of the mixture allows the \( \theta_i \)’s to draw statistical power from similarly progressing subjects.

It is not uncommon in mixture models to have some components with only a few data points. To estimate these components reliably, we draw statistical power from the remaining components by introducing a hyper-prior which relates all components to each other (Fig. 1b).

The \( \theta_i \)’s of individual subjects, and parameters of the mixture prior, and of the hyper mixture prior, form the total set of parameters to be estimated from the set of all time series of a symptom for all subjects. After estimating the parameters, the time series for subject \( i \) is denoised as \( \hat{m}_i(t) = \sum_{y=0}^{C-1} \hat{p}(y|t, \hat{\theta}_i) \).

IV. Model

We now explain mathematical details of the model. A glossary of variables used in this section can be found in Table II.

A. Ordinal Random Variables

Suppose \( y \) is an ordinal random variable taking values in ordinal categories \( \{0, \ldots, C-1\} \). A common model [19], [21] for the probability distribution of \( y \) is

\[
\begin{align*}
    p(y | \gamma) &= \begin{cases} 
    F(\gamma_1) & \text{for } y = 0 \\
    F(\gamma_2) - F(\gamma_1) & \text{for } y = 1 \\
    \vdots & \text{ } \\
    F(\gamma_k) - F(\gamma_{k-1}) & \text{for } y = k \\
    \vdots & \text{ } \\
    1 - F(\gamma_{C-1}) & \text{for } y = C - 1,
    \end{cases} \\
\end{align*}
\]

where \( F \) is the cumulative distribution function (cdf) of the standard normal density and \( \gamma = (\gamma_1, \ldots, \gamma_{C-1}) \in R^{C-1} \) is a parameter of the distribution and is subject to the constraint

\[
\gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_{C-1}.
\]

Fig. 2 illustrates the model. \( F(\gamma_1), \ldots, F(\gamma_{C-1}) \) are thresholds which divide the interval [0,1] into subintervals representing the probabilities of \( y \). The constraints of equation (5) guarantee that the probabilities satisfy \( p(y|\gamma) \geq 0 \) for all \( y \) and \( \sum_{y=0}^{C-1} p(y|\gamma) = 1 \).

The set of \( \gamma \) satisfying the constraints of equation (5) form a proper subset \( \Omega \subset R^{C-1} \). The boundary of \( \Omega \) corresponds to one or more of the inequalities in equation (5) being exact. The interior of \( \Omega \), denoted \( \text{int}(\Omega) \), is not empty. It corresponds to all \( \gamma \) that satisfy equation (5) with strict inequalities.
C. Multiple Time Series

Suppose there are $N$ subjects, with time series $Y_i = (y_{i,1}, \cdots, y_{i,T_i})$, $i = 1, \cdots, N$. Assuming that the $i$th subject has the subject’s own parameter $\theta_i = (u_i, v_i)$,

$$p(Y_i \mid \theta_i) = \prod_{t=1}^{T_i} p(y_{i,t} \mid \Gamma(u_i)(1-t) + \Gamma(v_i)t)$$  \hspace{1cm} (8)

D. Prior on $\theta_i$

As mentioned above, we set the prior for $\theta_i$ to a normal mixture with $K$ components. To limit the number of parameters in the model, all components of the mixture are assumed to have the same diagonal covariance matrix $\rho_{\mu} I$, with $\rho_{\mu} > 0$. Letting $z_i$ be the latent variable used to identify which component of the mixture $\theta_i$ comes from, the prior is given by

$$p(\theta_i \mid z_i = k, \{\mu_k\}, \rho_{\mu}) = N(\mu_k, \rho_{\mu} I),$$  \hspace{1cm} (9)

where $z_i = k$ indicates that $\theta_i$ belongs to the $k$th component. The set of means $\{\mu_k\}$ and $\rho_{\mu}$ are the parameters of the prior distribution. The dimension of $\mu_k$ is $D$.

E. Hyper-prior on $\{\mu_k\}$ and $\rho_{\mu}$

We set the hyper-prior on $\{\mu_k\}$ to a normal distribution, and the hyper-prior on $\rho_{\mu}$ to an inverse-gamma distribution:

$$p(\mu_k \mid \zeta, \rho_{\zeta}) = N(\zeta, \rho_{\zeta} I),$$  \hspace{1cm} (10)

$$p(\rho_{\mu} \mid a, b) = IG(a, b) = \frac{b^a}{\Gamma(a)(\rho_{\mu})^{a+1}} e^{-\frac{b}{\rho_{\mu}}}. \hspace{1cm} (11)$$

The hyper-prior parameters $a, b, \rho_{\zeta}$ are fixed and set by a procedure described later. This normal-inverse-gamma hyperprior is conjugate to the components of the normal mixture.

F. The full model

Grouping together all parameters to be estimated as $\phi = (\{\theta_i\}, \{\mu_k\}, \rho_{\mu}, \{\alpha_k\}, \zeta)$, we get:

$$p(\{Y_i\} \{z_i\} \mid \phi) = \prod_{i=1}^{N} p(Y_i \mid \theta_i)p(\theta_i \mid z_i, \{\mu_k\}, \rho_{\mu}) \hspace{1cm} (12)$$

$$\times \left( \prod_{k=1}^{K} p(\mu_k, \rho_{\zeta}) \right) \times p(\rho_{\mu} | \alpha, \beta) \times \prod_{i=1}^{N} p(z_i \mid \{\alpha_k\}).$$

Marginalizing out the latent variables $\{z_i\}$ gives

$$p(\{Y_i\} \mid \phi) = \sum_{z_{11} \cdots z_{2N}} \sum_{z_{11} \cdots z_{2N}} p(\{Y_i\} \{z_i\} \mid \phi),$$  \hspace{1cm} (13)

from which the maximum-likelihood estimates of $\phi$ are obtained as $\phi = \arg\max_{\phi} \log p(\{Y_i\} \mid \phi)$. The maximum-likelihood estimates are easily calculated by the EM-algorithm whose details are given below.
The maximum likelihood estimates of $\hat{\theta}_i = (\hat{u}_i, \hat{v}_i)$ give the estimated mean and variance:

$$m_i(t) = \sum_{y=0}^{C-1} y p(y_i | \Gamma(\hat{u}_i)(1 - t) + \Gamma(\hat{v}_i)t)$$

and

$$\hat{s}_i^2(t) = \sum_{y_i=0}^{C-1} (y_i - \hat{m}_i(t))^2 p(y_i | \Gamma(\hat{u}_i)(1 - t) + \Gamma(\hat{v}_i)t).$$

The estimated mean is the denoised signal.

### G. The EM Algorithm

The Q-function for the EM algorithm which maximizes the log-likelihood for the model equation (13) is:

$$Q(\phi | \phi^{[n]}) = \sum_{i=1}^{N} \log P(Y_i | \theta_i^{[n]})$$

$$+ \sum_{i=1}^{N} \sum_{z_i=1}^{K} \log p(\theta_i, z_i | \{\mu_k\}^{[n]}, \rho_\mu^{[n]} p(z_i | \theta_i^{[n]}, \{\mu_k\}^{[n]}, \rho_\mu^{[n]})$$

$$+ \sum_{k=1}^{K} \log p(\mu_k | \zeta^{[n]}, \rho_\zeta^{[n]}) + K \log p(\rho_\mu | a, b),$$

where

$$p(z_i | \theta_i^{[n]}, \{\mu_k\}^{[n]}, \rho_\mu^{[n]}) = \frac{p(\theta_i, z_i = k | \{\mu_k\}^{[n]}, \rho_\mu^{[n]})}{\sum_{k=1}^{K} p(\theta_i, z_i = k | \{\mu_k\}^{[n]}, \rho_\mu^{[n]})}$$

$$= w_i^{[n]}_{k},$$

and the superscript $[n]$ refers to values at the $n$th iteration.

The iterations of the EM algorithm are given in Table I. All updates have closed form solution except those of $\theta_i$. The $\theta_i$ updates are obtained by numerical gradient ascent using Adam [28] on those terms of the Q-function that depend on $\theta_i$. The EM updates are iterated until convergence.

### H. Initialization

To initialize the EM algorithm, the parameters $\theta_i = (u_i, v_i)$ are estimated for each subject by maximizing the log-likelihood $\log p(Y_i | \theta_i)$ of equation (8) using Adam. That is, the hierarchical structure is ignored. Then the $k$-means algorithm is used to group the estimates $\{\hat{\theta}_i\}$ into $K$ clusters whose means and isotropic variance are taken as initial estimates of $\mu_k$ and $\rho_\mu$. The hyperparameters $\zeta$ and $\rho_\zeta$ are initialized as $\hat{\zeta} = \frac{1}{N} \sum_{n=1}^{N} \hat{\theta}_i$, and $\hat{\rho}_\zeta = \frac{1}{N} \sum_{n=1}^{N} (\hat{\theta}_i - \hat{\zeta})^2$. (17)

### I. Cross Validation

We set the hyper parameter $a = 2.0$ (informal experimentation showed that the results are insensitive to changes in $a$). Then, the number of clusters $K$ and the parameter $b$ are determined by cross-validation. The number of clusters varies from 0 to 5. The 0 cluster corresponds to ignoring the prior and fitting equation (8) individually to each subject.

Given a data-set of ordinal time series of a symptom for multiple subjects, 90% of the time series are set aside as training data and 10% set aside as test data. For each series in the training data, 10% of the data points are set aside for cross-validation, and 10-fold cross validation is used to determine $K$ and $b$. The log-likelihood of the cross-validation data are assessed using equation (8) for the value of $t$ corresponding to the cross-validation time points. The values of $K$ and $b$ maximizing the validation log-likelihood are chosen.

### J. Denoising

After the EM iterations converge, the $\hat{u}_i, \hat{v}_i$ estimates (obtained from $\hat{\theta}_i$ estimates) are used to calculate the denoised time series and its variance from equation (14).

### K. Processing the Test Set

After all parameters are estimated from the training set, the estimated prior and hyperprior parameters are fixed and used with the test set. 10% of the data points in the test set are set aside to evaluate the performance of the model. The $\theta_i$’s for the test set, which are subject specific, are estimated via the update rule shown in Table I, using Adam.

### L. Evaluating the Model

Note that 10% of the data are left out in the training set and in the test set. After fitting the model to the training and the test set as described above, we evaluate the performance of the model by calculating the log likelihood of the left-out data (using equation (8) on the left-out data) for both sets. This gives a fair comparison of how well the model fits the data, irrespective of whether the prior is used ($K = 0$) or not ($K > 0$).

### V. Experimental Results

To demonstrate the effectiveness of our hierarchal model we apply it to two data sets. The first is a simulated data set, which allows us to test the model’s performance against ground truth. The second data set comes from PPMI MDS-UPDRS Part 3 scores.
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2022.3163126, IEEE Journal of Biomedical and Health Informatics

A. Simulated Data

The idea behind the simulation is to create ordinal time series with two progression subtypes: one in which the probability of observing higher ordinal classes increases with time, and one in which the probability decreases with time. To achieve this, we used Bernstein polynomials of degree three defined over \( I = [0, 1] \) [29]. The polynomials, denoted \( b_0, \ldots, b_3 \), have the property that they are non-negative over \([0, 1]\) and at every point in \( I \) they sum to 1.

The polynomials were converted into the two types of ordinal time series by mapping \( t \in I \) into the interior of \( I \) by one of two affine maps \( \phi \):

\[
\phi(t) = \begin{cases} 
  u(1-t) + vt & \text{(map 1)}, \\
  v(1-t) + ut & \text{(map 2)},
\end{cases}
\]

where \( u \) is randomly chosen from \([0.1, 0.25]\) and \( v \) is randomly chosen from \([0.75, 0.9]\). Thus map 1 maps \( I \) to the interior of \( I \) in monotonically increasing fashion, while map 2 maps in a monotonically decreasing fashion. Therefore \( p_i(t) = b_i(\phi(t)), \ i = 0, \ldots, 3 \) are ordinal class probabilities where the chances of observing higher ordinal classes increase with time for map 1, and decrease with time for map 2. Finally \( t \in [0, 1] \) was sampled at 11 uniform points and the probabilities \( p_i \) at these points are used to generate the ordinal time series.

A total of 300 ordinal time series were used to train our model with 150 time series coming from each map. Another 50 time series from each map were used as a test set.

Results from cross-validation are given in Table III. The \( l_{\text{train}} \) row shows the log-likelihood of the left-out data in the training set. The \( l_{\text{test}} \) row shows the the log-likelihood of the left-out data in the test set. Note that \( l_{\text{test}} \) is similar to \( l_{\text{train}} \) showing that the model does not overfit for any \( K \). The maximum value of \( l_{\text{train}} \) is obtained for \( K = 2 \), showing that the method correctly identifies the number of progression subtypes.

Using \( K = 2 \) and the corresponding \( b \) value, the EM-algorithm was used to estimate the rest of the parameters from the training set. Then, the prior and hyper-prior parameters \( \{\mu_k\}, \rho_k, \{\alpha_k\}, \zeta \) were fixed to their estimates from the training set, and \( \theta_i \) determined for the test set. Fig. 4 shows

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_t )</td>
<td>Ordinal Time Series</td>
</tr>
<tr>
<td>( \hat{m}_i )</td>
<td>Denoised time series</td>
</tr>
<tr>
<td>( \delta_i )</td>
<td>Standard deviation of denoised time series</td>
</tr>
<tr>
<td>( u )</td>
<td>Parameters in eq. (6) at ( t=0 )</td>
</tr>
<tr>
<td>( v )</td>
<td>Parameters in eq. (6) at ( t=1 )</td>
</tr>
<tr>
<td>( \theta_i )</td>
<td>( {u, v} )</td>
</tr>
<tr>
<td>( \mu_k )</td>
<td>Mixture prior means</td>
</tr>
<tr>
<td>( \rho_i )</td>
<td>Mixture prior variance</td>
</tr>
<tr>
<td>( K )</td>
<td>Number of clusters in the mixture prior</td>
</tr>
<tr>
<td>( z_j )</td>
<td>Latent variable indicating mixture component</td>
</tr>
<tr>
<td>( \alpha_k )</td>
<td>Mixture coefficients</td>
</tr>
<tr>
<td>( \zeta )</td>
<td>Hyperprior mean</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Hyperprior variance</td>
</tr>
<tr>
<td>( \alpha, \beta )</td>
<td>Inverse-gamma prior parameters</td>
</tr>
</tbody>
</table>

**TABLE II**: Descriptions of variables used in our model

<table>
<thead>
<tr>
<th>K</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>( b )</td>
<td>-</td>
<td>270</td>
<td>245</td>
<td>200</td>
<td>215</td>
<td>250</td>
<td>-</td>
</tr>
<tr>
<td>( l_{\text{train}} )</td>
<td>-2.06</td>
<td>-1.28</td>
<td>-1.24</td>
<td>-1.30</td>
<td>-1.40</td>
<td>-1.74</td>
<td>-</td>
</tr>
<tr>
<td>( l_{\text{test}} )</td>
<td>-2.00</td>
<td>-1.28</td>
<td>-1.20</td>
<td>-1.20</td>
<td>-1.24</td>
<td>-1.29</td>
<td>-</td>
</tr>
<tr>
<td>( f_{\text{MSE}} )</td>
<td>0.073</td>
<td>0.060</td>
<td>0.027</td>
<td>0.056</td>
<td>0.062</td>
<td>0.065</td>
<td>0.066</td>
</tr>
<tr>
<td>( f_{\text{MSE}} )</td>
<td>0.067</td>
<td>0.073</td>
<td>0.021</td>
<td>0.021</td>
<td>0.036</td>
<td>0.056</td>
<td>0.051</td>
</tr>
</tbody>
</table>

**TABLE III**: Cross validation results for the simulation and algorithm execution times. \( K \) is the number of clusters in the mixture prior (\( K = 0 \) represents no prior). \( b \) is a parameter of the inverse-Gamma hyperprior. \( l_{\text{train}}, l_{\text{test}} \) are the log-likelihood of the probability model of left-out data from equation (8). \( f_{\text{MSE}} \) are the normalized mean squared error values from the ground truth for the training and test sets. Performance of linear regression is also included as a benchmark. The column with \( k = 2 \) is in bold font because it has the highest training and test log-likelihood.

The shaded regions represent the standard deviation of the ground truth trajectory, the denoised trajectory, and a linear fit. The standard deviation of the estimated ordinal probabilities, \( \hat{m}_i(t) \), is shown in black, and the true probabilities in gray.

![Fig. 3: Bernstein polynomials of degree 3 over the interval [0, 1]. u and v are selected from the gray regions for map 1 and map 2.](image)

![Fig. 4: (a) Trajectories and probabilities of simulated time series. (b) Comparison of the synthetic time series with the expected ground truth trajectory, the denoised trajectory, and a linear fit. (c) Comparison of the learned probabilities (black) and the true probabilities (gray).](image)
two typical ordinal time series from the simulation, their true and estimated probabilities, true and estimated mean, and true and estimated standard deviation.

The algorithm results were evaluated further for the model with the best $K$ and $b$ from cross-validation. First, we evaluated in detail whether the model overfit the training data. For this, we calculated the $L_1$ norm between estimated and real probabilities for the left-out points in the simulated training and test sets. Fig. 5 shows the histogram of the $L_1$ norm square of the true and estimated probabilities, true and estimated mean, and true and estimated standard deviation.

Finally, we evaluated the model’s ability to capture the two subtypes of ordinal time series. For this, we took the estimated $\hat{\theta}_i$’s for each time series and embedded them in a 2-dimensional space using multi-dimensional scaling [30]. Fig. 6 shows the embedded $\hat{\theta}_i$’s shaded with the type of map (of equation (18)) that the series was generated with. Clearly, the estimated $\hat{\theta}_i$’s have captured the “bi-modality” of the simulated data generation. Fig. 6 also shows that the estimated means, $(\hat{\mu}_1, \hat{\mu}_2)$ of the mixture of normals prior lie within the two clusters indicating natural progression subtypes for the time series.

**B. Parkinson’s Disease**

Next, we analyzed data from PPMI. We used motor exam scores from MDS-UPDRS Part 3. After excluding patients with fewer than four visits, there were 377 PD patients with $10.4 \pm 3.1$ (mean \pm stdev) visits. Instead of summing all scores into a total movement score, we separately analyzed the time series of one symptom for each of the three cardinal symptoms of PD (rigidity, tremor, and bradykinesia). We analyzed scores for left-arm rigidity, right-leg rest tremor amplitude and right-leg agility (a measure of bradykinesia). Since PPMI data focuses on early-stage PD patients, scores of 4 were uncommon, and thus we combined categories 3 and 4 into a single category named 3.

Cross validation revealed optimal values of $K = 2$ and $b = 1167$ for left-arm rigidity, $K = 2$ and $b = 750$ for right-leg rest tremor amplitude, and $K = 1$ (single mixture component) and $b = 950$ for right-leg agility. Using the cross-validated parameters, the model was fit to the training set for each question. After training, we retained the trained prior and hyperprior parameters, and the model was again fit to the test set. The algorithm typically converged in 50-100 EM iterations, which took less than a minute for a MATLAB implementation. Fig. 7(a,c,e) show histograms of the log-likelihoods of the training left-out data and of the test data. The histograms indicate that there is no overfitting.

Fig. 7(b,d,f) show scatter plots of training and test $\hat{\theta}_i$’s along with the prior means $\mu_k$ all projected via multi-dimensional scaling onto a 2-dim plane. Note that the $\hat{\theta}_i$’s have a greater dispersion between the clusters than the simulated data. Because the cluster variance is estimated using the data by our method, such dispersion is handled automatically.

Fig. 8 shows typical time series along with their estimated means and standard deviations. It is clear from the figure that the denoised time series capture the progression of the symptom. Some patients appear to progress more rapidly compared to others.

The patterns revealed by denoising are displayed in Figs. 9-11. The (a) part of Fig. 9 shows raw left-arm rigidity scores
Fig. 7: (a,c,e) Comparisons of the distributions of log-likelihood of left-out data for training and test data. (b,d,f) $\theta_i$'s embedded in 2-dim space with multi-dimensional scaling. (a,b) Right-Leg Rest Tremor Amplitude. (c,d) Left-Arm Rigidity. (e,f) Right-Leg Agility.

of 20 randomly chosen subjects. Any disease progression pattern in this cohort is difficult to discern. The (b) part of the figure shows denoised scores for the same subjects along with trajectories corresponding to the means of the two components in the mixture prior. The latter are the progression subtypes for this score. The disease progression is quite clear in the denoised scores. Also, the subtypes clearly show that this symptom steadily progresses with one group of subjects starting at a higher value relative to the other.

The effect of denoising is even more striking when we analyze the changes in the score from baseline. For each subject, the change from baseline at time $t$ is the score at time $t$ minus the score at baseline. For a steadily progressing disease, such as PD, the change from baseline should be positive for all $t$. Fig. 9 (c) shows the change from baseline for raw left-arm rigidity scores for the subjects whose scores are plotted in Fig. 9 (a). Note that the noise in the scores overwhelms the perception of any pattern. Fig. 9 (d) shows the change from baseline for denoised scores. The progression of the disease is quite clear in Fig. 9 (d).

Fig. 9 (c) and (d) suggest that the denoised change-from-baseline scores are more useful to understand the underlying progression patterns than the raw change-from-baseline scores. This suggestion can be quantitatively evaluated as follows: Suppose we calculate the ratio of the mean to the standard deviation (across all subjects) of the change-from-baseline scores. If this ratio is higher, it is easier to quantitatively detect disease progression. The top two rows of Table IV show the change-from-baseline mean to standard deviation ratio for raw and denoised left-arm rigidity scores. Note the significantly larger ratios for denoised scores, clearly pointing to the utility of denoising.

The results for right-leg rest tremor amplitude and right-leg agility are similar. Fig. 10 (a) and (b) show the raw and denoised scores for right-leg rest tremor amplitude for 20 randomly selected subjects. Again, progression patterns are difficult to discern in the raw scores. Denoising reveals that most patients' trajectories are relatively flat over time, while there is a small group for whom tremor typically begins a little higher and rises more rapidly. This is also reinforced by the trajectories of the prior component means (subtypes), $\hat{\mu}_k$, $k = 1, 2$. The raw and denoised change from baseline (Fig. 10 (c) and (d)) behave similar to that of left-arm rigidity. The
Variation from this prior seems evenly distributed above and below the mean trajectory. Gradual progression with time is due to denoising.

Change from baseline, denoised change from baseline) are shown in Fig. 11 for 20 randomly selected patients. The last two rows of Table IV show the improved mean to standard deviation ratio due to denoising.

### VI. DISCUSSION AND CONCLUSION

The experimental results with simulated data clearly show that the hierarchical model is able to closely estimate the ground truth signal and does not overfit data. The number of clusters in the hierarchical model was successfully determined through cross-validation and matches the number of underlying clusters in the simulated data. The denoised mean provided by the model was similar to the ground truth. The mixture prior means, μ₁, μ₂, provide a good indication of a prototypical trajectory from each progression subtype (map).

The experimental results also show that the EM algorithm converges reliably in spite of the multi-level hierarchy for both the simulated data and the MDS-UPDRS data. For the MDS-UPDRS data, the hierarchical model also does not overfit, providing similar results on both training and test examples.

Analysis of the MDS-UPDRS data clearly shows the power of the method to elucidate underlying patterns in noisy time series of real-world clinical scores. The patterns exposed by the denoised series are more informative than those in the raw series. This is significant given the large number of studies which gather such data.

Identification of clinical subtypes and disease progression patterns in neuropsychiatric and neurodegenerative disorders is important for research and clinical trials. Different clinical subtypes may have different underlying pathological mechanisms and may respond to therapeutic interventions differently. Our methodology identifies disease progression subtypes from noisy real-world clinical data and is a promising tool for use in clinical studies.

Our model has some limitations which we discuss now: First, the model requires the entire ordinal time series before denoising is attempted. Thus, in its current form, the model cannot be used in an on-line fashion to update the estimate of the denoised series as the data arrive. However, it is not too difficult to modify the model to create such an on-line algorithm. Second, the mixture prior has some limitations. If the θᵢ parameters have a very wide spread, then some other form of a prior with a wider support may be necessary.

While processing the simulated and MDS-UPDRS data, we have not also evidenced in the denoised change-from-baseline in Fig. 11 (d).
noticed the need for wider-support priors. However, if the need arises, the Gaussian components may be replaced with heavier-tailed t-distributed or other wide-support components. Finally, we note that there may be reason to use methods other than cross-validation to determine $K$, the number of components in the prior. For example, in some applications, it may be useful to just use $K = 1$ if there is reason not to cluster the $\theta_i$'s.

In conclusion, we have reported a method for denoising noisy ordinal time series of clinical scores which are common not only in PD but also in studies of other neurological diseases such as multiple sclerosis [31], [32], and in psychiatric conditions such as depression [33], [34] and anxiety [35], [36]. The method explicitly takes the ordinal nature of time series into account and draws statistical power from similarly progressing time series to overcome the short duration of the series. When applied to real-world clinical data, the method clarifies the underlying progression patterns.

ACKNOWLEDGMENT

This research was supported by the grant R01NS107328 from the NIH (NINDS). Data used in the preparation of this article were obtained from the Parkinson’s Progression Markers Initiative (PPMI) database (www.ppmi-info.org/data). For up-to-date information on the study, visit www.ppmi-info.org. PPMI – a public-private partnership – is funded by the Michael J. Fox Foundation for Parkinson’s Research and funding partners. The full list of funding partners can be found at https://www.ppmi-info.org/about-ppmi/who-we-are/study-sponsors/.

REFERENCES


