Predicting States of Chronic Pain with Markov Models

Maia Posternack and Siddharth Tiwari

December 12th, 2022

1 Cover Letter

Dear Dusty,

We're so grateful Math 22 has a project rather than a final exam! We learned a lot about Markov chains through our research, and were able to apply them in a way that was in accord with our interests and possibly beneficial to others.

We split the paper so that, in general, Maia focused on section 4, proving different aspects about Markov models, and Sidd focused on sections 5 and 6, applying the model to real data. That said, we, of course, helped each other with these sections.

Most of the corrections we made for our final draft reflected comments made by our TF and reiterated by our commenters. Kathy Zhong, Hayden Graham, and our reader, Charles Wang, advised that our project draft focused mainly on the application and less on the proofs, and with our final draft, we sought to thoroughly highlight all relevant proofs and theorems for the basis of our proposed experimental metric. Andrew Li and Eyob Davidoff gave formatting advice for the proofs and the code which we implemented. Finally, Jenny Lu and Charles Wang suggested we have a simpler version of later complicated matrices as an example in section 4, which we found worked really well.

Upon full implementation of our code, we observed interesting results that led to a more thorough experiment. Although our TF told us it wasn't necessary to include full screenshots of our code within the project itself, we decided to include small chunks (formatted in latex this time) to emphasize the connection of the proofs/theorems we developed to the model and its subsequent computation.

Overall, we're grateful for the self-directed learning and inquiry that this project has brought for us; our paper is a little long, but we hope you'll share our enthusiasm as we detail results of, possibly, a new development in the realm of clinical pain research. Thank you so much for such a wonderful semester; we look forward to 22b in the spring.

-Sidd and Maia

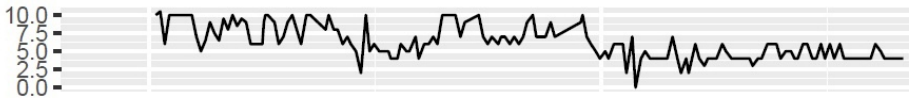
2 Abstract

Pain is the most ubiquitous and economically burdensome condition in the world. For this reason, pain relief—a decrease in pain over time—is the primary outcome for clinical trials across many fields, ranging from pharmaceuticals to physiotherapy. Often measured at few, discrete-time points (e.g., pre-and post-intervention), measures and statistical models of pain relief are poor. In our project, we propose a Discrete-Time Markov Model in order to obtain a steady-state approximation of the "end behavior" of a patient's pain trajectory. Using this model, corresponding analyses of pain trajectory, particularly those of analgesic clinical trials, will account for more variance, and clinicians and pain researchers can offer more predictive and critical insight into patient's subsequent pain conditions.

3 Introduction

Doctors often ask their patients to evaluate their pain levels from 0-10. They then compare current pain ranking to previous rankings to see if a patient's pain increased or decreased and then diagnose the patient. That said, a patient's pain tends to fluctuate. Studies show that pain shifts between normal trajectories and "flare states". So a doctor's evaluation of whether a patient's pain has decreased or increased overall may not be very useful if they currently are in a state of extreme or little pain. Thus, when evaluating a patient's pain, a model must take into account the probability that pain will dramatically increase or decrease (aka fluctuate) as well as a patients current pain state, in order to predict future pain states. Markov models allow us to do this efficiently.

For example, consider a patient's report on their pain over time in the figure below. At first glance, a doctor may simply look at the initial and final rankings of the patient and determine that their pain has decreased (here, approximately from a '10' to a '4'). They may predict, then, that the patient's pain would continue decreasing. But upon analyzing the graph, we can see that while the patient currently is in a state of relief, their pain may spike soon as it did in the past. A markov model would predict that.
In addition, rather than just decreasing, it appears that the pain levels approach a flat line (with some fluctuation). We can use steady state vectors based on the markov model to predict what this line may be.



4 Theoretical Foundations of the Model

Let us begin by exploring this problem on a simpler level. We will then apply what we learn to real patient data to predict the end behavior of their pain trajectory.

Consider a patient, Dusty. He has chronic back pain and tells his doctor about it. Every time he sees his doctor, his doctor asks him to rank the pain on a scale from 1-3 (1 being very little pain and 3 being excruciating pain). The doctor records this data.

These three levels of pain are examples of **states** that Dusty can be in. Thus, 1,2,3 is are in his **state space**

Definition 4.1 (State)**.** For any collection variables indexed by time s_n where $n = 0, 1, 2..$, if $s_n = i$ then i is a state at time n.

Definition 4.2 (State Space)**.** A state space is the set of values that each s_n can take.

For notation sake, we will say that if Dusty has a rating of 1 today, his state is:

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

If he has a rating of 2, his state is:

$$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

And if he has a rating of 3, his state is:

$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

These states are **probability vectors**.

Definition 4.3 (Probability Vector)**.** A probability vector $v \in \mathbb{R}^n$ with non-negative entries that add up to 1.

For example, if Dusty has a state of 1 today, the matrix

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

tells us that he currently has 100% chance of being in state 1, 0% chance of being in state 2, and 0% chance of being in state 3.

Now if Dusty has a pain rating of 1 today, he isn't equally likely to have a rating of 1, 2, or 3 tomorrow. He is more likely to remain at a pain of 1 than jump all the way to 3. We can represent the probability he transitions between states with a matrix.

$$\begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}$$

Each column represents a states Dusty is currently in (column one represents pain of 1 etc.) and each row represents a state that Dusty could be in tomorrow (row one represents pain of 1 etc.).
This matrix is called a **transition matrix**.

Definition 4.4 (Transition Matrix)**.** The transition matrix, P, is a square matrix describing the probabilities of moving between states.

Each entry, p_{ij}, represents the probability that a system will end up in state i given it is in state j

Thus, if Dusty currently has a pain rating of 1, tomorrow he will have a 70% chance of having a rating of 1 , 20% chance of having a rating of 2 and a 10% chance of having a rating of 3.

Thus, note that the transition matrix is **stochastic**.

Definition 4.5 (Stochastic)**.** A stochastic matrix is a square matrix whose columns are probability vectors

Theorem 1. *The product of a stochastic matrix and probability vector is a probability vector.*

Proof. Consider a stochastic matrix $P = \begin{bmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1} & \cdots & p_{nn} \end{bmatrix}$ and a probability vector $X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$.
We will prove that the product of P and X, PX, is a probability vector by proving (1) that it has non-negative real entries and (2) the sum of those entries is 1.

By matrix multiplication, $PX = x_1 \begin{bmatrix} p_{11} \\ \vdots \\ p_{n1} \end{bmatrix} + x_2 \begin{bmatrix} p_{12} \\ \vdots \\ p_{n2} \end{bmatrix} + ... + x_n \begin{bmatrix} p_{1n} \\ \vdots \\ p_{nn} \end{bmatrix}$

(1) Because each entry in PX is a linear combination of non-negative real numbers with non-negative real number weights, each entry must also be a non-negative real number.

(2) The sum of the entries in PX, then, is

$$x_1(p_{11} + ... + p_{n1}) + x_2(p_{12} + ... + p_{n2}) + ... + x_n(p_{1n} + ... + p_{nn})$$

By definition of stochastic matrices, each column is a probability vector, meaning the sum of the entries in the column is 1.

Thus,

$$x_1(p_{11} + ... + p_{n1}) + x_2(p_{12} + ... + p_{n2}) + ... + x_n(p_{1n} + ... + p_{nn}) =$$

$$x_1(1) + x_2(1) + ...x_n(1)$$

Since X is by definition a probability vector, its entries sum to 1 as well. Thus,

$$x_1(1) + x_2(1) + ...x_n(1) = 1$$

Thus, we have proven both (1) and (2), the axioms of probability vectors, and have therefore proven that PX is a probability vector.

□

Now when we multiply a stochastic matrix and a probability vector, we can get a **markov chain**.

So what is a markov chain?

Definition 4.6 (Markov Chain)**.** A Markov Chain, S, is a sequence of probability vectors

$$x_1, x_2, ..., x_n$$

together with a stochastic matrix, P, that when multiplied, have the **markov property**.

Definition 4.7 (Markov Property)**.**

$$P(x_k = i \mid x_{k-1} = j, x_{k-2}, ...x_1) = P(x_k = i \mid x_{k-1} = j)$$

In other words, a given state depends only on the previous state (not any of the states before that).

Now consider that Dusty comes to the doctor today with a pain rating of 3. How can you predict what his pain may be tomorrow? We see that we can multiply the transition matrix, T = $\begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}$ by his pain today, $s_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, to see that

$$\begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} .1 \\ .4 \\ .5 \end{bmatrix}$$

Thus, tomorrow, he has a 10% chance of having a pain of 1, a 40% chance of having a pain of 2 and a 50% chance of having a pain of 3.

Given that he is in that state tomorrow, we can multiply the T by $s_1 = \begin{bmatrix} .1 \\ .4 \\ .5 \end{bmatrix}$ to see that the day after tomorrow he will have a pain distribution, s_2, of

$$\begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix} \begin{bmatrix} .1 \\ .4 \\ .5 \end{bmatrix} = \begin{bmatrix} .24 \\ .46 \\ .3 \end{bmatrix}$$

Note if we keep multiplying our product by the transition matrix, to find some value, s_k, we are simply computing the equation $T^k s_0$

Lemma 1. *For every markov chain with state probability vectors r_0, r_1.. and transition matrix, P, any $k \in W$,$r_k = P^k r_0$*

Proof. Base Case (k=1):

By definition, $r_1 = Pr_0$. Thus, $P^1 r_0 = r_1$. Thus the base case holds.

Inductive Step (n $\longrightarrow$ n+1):

By definition, $r_{n+1} = Pr_n$. By the inductive hypothesis, we know that $r_n = P^n r_0$. Thus,

$$r_{n+1} = PP^n r_0 = P^{n+1} r_0$$

Thus we have proven the inductive step and base case, therefore proving the theorem. □

Now that we have proven this lemma, how might Dusty feel in 5 days?

$$s_5 = T^5 s_0 = \begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}^5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} .419 \\ .406 \\ .175 \end{bmatrix}$$

We can interpret this as him having a 41.9% chance of pain level 1, 40.6% chance of pain level 2, and 17.5% chance of pain level 3.

How will he feel in 2 weeks?

$$s_{14} = T^{14}s_0 = \begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}^{14} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} .444 \\ .389 \\ .167 \end{bmatrix}$$

How will he feel in a month?

$$s_{31} = T^{31}s_0 = \begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}^{31} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} .444 \\ .389 \\ .167 \end{bmatrix}$$

By observation, we see that this product, s_k seems to converge.
The vector it converges towards is called the **steady state vector**.

Definition 4.8 (Steady State Vector)**.** A steady state vector, q, is a probability vector such that

$$Pq = q$$

In other words, once the system is at state q, even it will remain there.

Theorem 2. *Every stochastic matrix, P, has an eigenvalue of 1. Further, 1 is the maximum eigenvalue for P.*

Proof. Consider the transpose of P, $P^T = \begin{bmatrix} p_{11} & \dots & p_{n1} \\ \vdots & \ddots & \vdots \\ p_{1n} & \dots & p_{nn} \end{bmatrix}$

Note that, if $P^T v = \lambda v$ for some vector v $\in \mathbb{R}^n$, $\lambda \in \mathbb{R}$ is an eigenvalue of P^T.

Now consider $\begin{bmatrix} p_{11} & \dots & p_{n1} \\ \vdots & \ddots & \vdots \\ p_{1n} & \dots & p_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} + \dots + p_{n1} \\ \vdots \\ p_{1n} + \dots + p_{nn} \end{bmatrix}$ by matrix multiplication.

Because P is stochastic, its columns sum to 1. Thus the rows of P^T sum to 1 and so

$$\begin{bmatrix} p_{11} + \dots + p_{n1} \\ \vdots \\ p_{1n} + \dots + p_{nn} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} = 1 \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Therefore, we see that $P^T v = \lambda v$ for $\lambda = 1$ and $v = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$.

Thus 1 is an eigenvalue for P^T.

Since P and P^T have the same eigenvalues by theorems proven in class, 1 is an eigenvalue for P.

Now we will prove that $\lambda = 1$ is the maximum eigenvalue for P.

Consider the equation $P^T v = \lambda v$

Consider the maximum entry in vector v, v'_m where $1 \leq m \leq n$

Via matrix multiplication, the mth row in $P^T v$ must equal the mth row in v, v'_m . So:

$$p_{1m}v_1 + p_{2m}v_2 + ... + p_{nm}v_n = \lambda v'_m$$

Since v'_m is the largest entry in vector v,

$$p_{1m}v'_m + p_{1m}v'_m + ... + p_{nm}v'_m \geq \lambda v_m$$

And via algebra we see that

$$(p_{1m} + p_{1nm} + .. + p_{nm})v'_m \geq \lambda v'_m$$

So

$$p_{1m} + p_{2m} + .. + p_{nm} \geq \lambda$$

Since P is stochastic, the rows of P^T sum to 1. Thus

$$p_{1m} + p_{2m} + .. + p_{nm} = 1$$

So

$$1 \geq \lambda$$

Since the eigenvalues for P and P^T are the same, we know that 1 is also the maximum eigenvalue for P.

□

Using this theorem, we can say even more about steady state vectors

Corollary 1. *Every transition matrix has a steady state vector.*

Proof. By definition, a transition matrix, P, is a stochastic matrix. Thus, by theorem 2, it has an eigenvalue of 1. Therefore, there must a non-zero vector, b, such that

$$Pb = 1b$$

or

$$Pb = b$$

.
Note that for any real number c,

$$c(Pb) = Pcb = P(cb) = (cb).$$

Because b is non-zero, then, it can be scaled by real number c such that its rows sum to 1.

Therefore, vector v = cb, is by definition a steady state vector. Thus, every transition matrix must have a steady state vector. □

This corollary is really helpful! It tells us how to easily compute a steady state vector. Simply find a vector that is in the eigenspace for $\lambda = 1$ and scale it so that the rows sum to 1. We will use this approach later in sections 5-7.

But now, what would happen if Dusty started off with a pain of 1 instead of 3.

In 2 days, his chances of pain would be:

$$s_2 = T^2 s_0 = \begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}^2 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} .56 \\ .3 \\ .14 \end{bmatrix}$$

In a week, his chances of pain would be:

$$s_7 = T^7 s_0 = \begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}^7 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} .447 \\ .387 \\ .166 \end{bmatrix}$$

In two, they will be:

$$s_{14} = T^{14} s_0 = \begin{bmatrix} .7 & .3 & .1 \\ .2 & .6 & .4 \\ .1 & .1 & .5 \end{bmatrix}^{14} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} .444 \\ .389 \\ .167 \end{bmatrix}$$

Once again, Dusty's pain state converges to the **exact same** vector, $\begin{bmatrix} .444 \\ .389 \\ .167 \end{bmatrix}$.
Note that it didn't matter at all whether he started off with a pain level of 1 or 3.

By observation, clearly, it appears that Dusty has a unique steady state vector. This, of course, is not the case for all stochastic matrices.

Consider the following stochastic matrix,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

We can see that both

$$\begin{bmatrix}0.6\\0.2\\0.2\end{bmatrix}, \begin{bmatrix}0.1\\0.45\\0.45\end{bmatrix}$$

are steady state vectors by calculating:

$$\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\\0 & 1 & 0\end{bmatrix}\begin{bmatrix}0.6\\0.2\\0.2\end{bmatrix} = \begin{bmatrix}0.6\\0.2\\0.2\end{bmatrix}$$

$$\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\\0 & 1 & 0\end{bmatrix}\begin{bmatrix}0.1\\0.45\\0.45\end{bmatrix} = \begin{bmatrix}0.1\\0.45\\0.45\end{bmatrix}$$

Thus, we have found a stochastic matrix that doesn't have a unique steady state vector.

In fact, only **regular** stochastic matrices can be guaranteed to have a unique steady state vector.

Definition 4.9 (Regular)**.** A stochastic matrix, S, is regular if S^k contains only strictly positive entries for some positive k $\in$W.

We see that, for example, Dusty's transition matrix is regular as $\begin{bmatrix}.7 & .3 & .1\\.2 & .6 & .4\\.1 & .1 & .5\end{bmatrix}^1$ is strictly positive.

In order to prove why a normal stochastic matrix has a unique steady state vector, we need the **Generalized Perron-Frobenius Theorem**.

Theorem 3 (Generalized Perron-Frobenius Theorem)**.** *If any n x n matrix, A, is a nonnegative regular matrix, then it has a unique maximum eigenvalue, λ' with algabriac and geometric multiplicty of 1, and an eigenvector, v', with strictly positive entries.*

Now proofs for this theorem are long and complicated so they are out of the scope of this project. For a complete proof, visit this page.

With the Generalized Perron-Frobenius Theorem, we can prove uniqueness of steady state vectors.

Theorem 4. *If S is a stochastic square regular matrix, it has a unique steady state vector, v.*

Proof. Because S is stochastic, by Theorem 2, it must then have an eigenvalue of 1 and this is its maximum eigenvalue.

Thus, by the generalized Perron-Frobenius theorem, because S is stochastic and regular, since the maximum eigenvalue, λ', is 1, 1 has an algebraic and geometric multiply of one.

Thus, the eigenspace of λ' is simply the span of one vector. We can scale that one vector such that all entries add up to 1 to see that there is a unique vector in the eigenspace of λ' that is a probability vector and therefore a unique steady state vector for S.

□

Now we know that all regular stochastic matrices have a unique steady state vector, but will they always reach that vector?

To answer this question we need to introduce one more theorem.

Theorem 5. *For any regular stochastic matrix, and initial state x_0, $S^k x_0$ converges as k approaches infinity.*

This theorem is also very long and out of the scope of this project. For a proof, visit this page.

From this theorem, though, our final corollary follows easily.

Corollary 2. *If x_0 any initial state, and S is a stochastic regular matrix, $S^n x_0$ converges to v, the unique steady state vector.*

Proof. By theorem 5, $S^n x_0$ must converge.

Because v is the unique steady state vector, we know that if $S^n x_0$ converges, it has to converge to v.

Thus, $S^n x_0$ must converge to v.

□

Now that we have proven the uniqueness of steady state vectors, we see that Dusty has a unique steady state vector of $\begin{bmatrix} .444 \\ .389 \\ .167 \end{bmatrix}$.

We could interpret this in a couple ways. 1) He has a 44% chance of having a pain level of 1. 2) He will have a pain level of 1 44% of the time. 3) He shouldn't be worried if he occasionally has level 3 pain. That is to be expected and it will eventually go back down to levels 1 and 2.

5 Methodology

Now that we know how to predict Dusty's steady state vector, and therefore the "end behavior" of his pain, we can do so for real people with real data. Note that the people in these trials rank their pain 0-10 rather than 1-3.

We will begin in section 5.1 by formally defining our data.

5.1 Data

Data from two, previously published randomized, double-blind, placebo-controlled trials conducted by the Pain and Passions Lab at Northwestern University in Chicago, IL was used to model our data. This data is also publicly available at openpain.org.

The code for processing raw pain units is included in Appendix A.3.

5.1.1 Placebo II (Chronic back pain)

Overview. The purpose of this study was to validate a prognostic model for classifying chronic pain patients based on their predicted improvement with placebo [**?**].

Pain data. Data were collected using a custom pain rating phone app through which 94 patients could rate their pain (0–10 NRS), as in Placebo I. Patients were asked to enter their pain 2 times/day over the course of the entire study. For the purposes of demonstration, here we averaged pain ratings within a single day.

5.1.2 Levodopa trial (Subacute back pain)

Overview. The purpose of this trial was to investigate whether levodopa (l-DOPA) can block patients' transition from subacute to chronic back pain. This 24-week double-blind parallel randomized controlled trial was conducted at Northwestern University (Chicago, IL, USA). [**?**].

Pain data. Data were collected using a custom pain rating phone app through which patients could rate their pain (0–10 NRS). 48 patients were asked to enter their pain 3 times/day over the course of the entire study (28 weeks). For the purposes of demonstration, here we averaged pain ratings within a single day.

5.2 Modeling

To demonstrate the prognostic ability of the Markov chain, we fit a Markov chain to **each individual subject** and obtained a corresponding steady-state vector. As proposed in Section 3, our Markov model will consist of 11 states, with each successive state mapping to a corresponding pain level, zero through ten (zero being no pain and ten being excruciating pain). We then find the

corresponding **steady-state vector**.

We use this vector to produce an estimate of the patient's overall change in pain by subtracting the patient's starting pain from a weighted average of the components of the steady vector ("ending pain" - start pain, similar to how the standard change-in-pain metric is calculated), with the corresponding ordinal values for the states as weights.

For example, if a patient had a steady state vector of

$$\begin{bmatrix} 0.05 \\ 0.40 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.05 \\ 0.15 \end{bmatrix}$$

Their end pain would be:

$$0.05\times0+0.4\times1+0.05\times2+0.05\times3+0.05\times4+0.05\times5+0.05\times6+0.05\times7+0.05\times8+0.05\times9+0.15\times10 = 4.1$$

We would then subtract their original pain from 4.1 in our experimental metric.

5.3 The Experiment

To test for whether our metric is **actually predictive** of future changes in pain or not, we implemented a two step process for 17 subsets of the data, comparing our experimental metric to the standard change in pain metric (last pain-rating minus first pain-rating). We used 17 subsets of the total pain sequence (10% to 90%, with 5% increments), where a percentage of the sequence, from the first point to that point in the sequence, to calculate our metrics. This is the process we used for the obtained sequences:

1. For n% of the points within a patient's trajectory, we calculate both our estimate of the patient's overall in change in pain by utilizing a Markov model and the standard change-in-pain metric (based simply on the difference between final and initial pain) from the starting point to the n% mark.

2. Then, we calculate the standard change-in-pain metric for the TOTAL sequence.

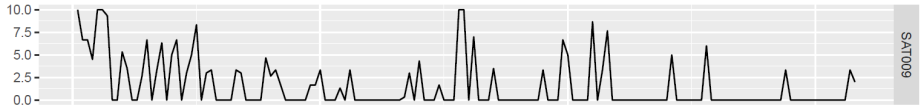
With these predictions, we calculated and plot the root mean-squared errors (RMSE) of the predictions for evaluation of the metrics' prognostic ability by each of the $n\%$ values; the purpose of this is to compare how predictive our experimental metric is vs. the standard metric with $n\%$ of the data to use for calculations. As the RMSE denotes the difference between the observed and expected value, ideally, our experimental metric will have a **lower RMSE than the standard change-in-pain metric**, thus indicating a better prediction.

R was utilized to fit models, and its implementation is depicted in Appendix A.1. The evaluation of the metrics/generation of the RMSE plot is included in Appendix A.2. The "markovchain", "ggplot2", "patchwork", "Metrics", and "tidyverse" packages were used in the implementation of this code.

6 Results

6.1 Single-subject fit

This is the metric-calculation process for patient "SAT009" from the Levodopa (LDopa) Trial:



This is the pain trajectory for patient "SAT009", and within this trajectory it's clear to see the high variance in pain ratings day-to-day. This patient has a particularly impressive downward trajectory; if we were to use the conventional change-in-pain metric right now, they'd have an overall decrease of 8 pain units (starting pain "10", ending pain "2"; 2-10 = -8). We shall observe whether this is reflected in our proposed change-in-pain metric. This is the code used to calculate the transition matrix for patient SAT009 and the resulting matrix:

```
sub = "SAT009"

#extract sequence of pain ratings from subject
sequence <- round(as.numeric(subset(pain_ldopa,
        subject == sub)$pain), digits = 0)

#fit markov model
mcFitMLE <- markovchainFit(data = sequence)

transition_matrix <- mcFitMLE["estimate"]]@transitionMatrix

#round to two digits for demonstration
round(transition_matrix, digits = 2)
```

Through our code, we obtain the following transition matrix:

$$\begin{bmatrix}
0.77 & 0.01 & 0.01 & 0.02 & 0.10 & 0.02 & 0.04 & 0.01 & 0.02 & 0.00 \\
0.01 & & & & & & & & & \\
1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & & & & & & & & & \\
0.20 & 0.00 & 0.40 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.20 & 0.00 \\
0.20 & & & & & & & & & \\
0.50 & 0.00 & 0.00 & 0.25 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & & & & & & & & & \\
0.44 & 0.00 & 0.00 & 0.12 & 0.19 & 0.00 & 0.06 & 0.06 & 0.06 & 0.06 \\
0.00 & & & & & & & & & \\
0.75 & 0.00 & 0.25 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & & & & & & & & & \\
0.33 & 0.00 & 0.00 & 0.00 & 0.17 & 0.17 & 0.00 & 0.00 & 0.17 & 0.17 \\
0.00 & & & & & & & & & \\
1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & & & & & & & & & \\
0.50 & 0.00 & 0.00 & 0.00 & 0.00 & 0.17 & 0.17 & 0.00 & 0.17 & 0.00 \\
0.00 & & & & & & & & & \\
1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & & & & & & & & & \\
1.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
0.00 & & & & & & & & &
\end{bmatrix}$$

Note that rather than the columns being probability vectors (like in section 4), the **rows** are actually probability vectors. This is simply because it is the default for R packages and won't actually affect any calculations (simply consider the transpose of this matrix to get a stochastic matrix).

As previously stated, this is an 11 by 11 matrix as each row and column represent a rating from 0-10. The row-index is a pain rating m at time $t-1$ (aka what you rate your pain today), column-index is the following pain rating n at time t (aka what you will rate your pain tomorrow), thus meaning the matrix entries correspond to the probability that a pain rating will switch from m to n over time.

As expected from observing the steep declines within the trajectory, **many of the pain ratings have a high probability to go straight to 0** ("6", "8", and "9" at $t-1$ all have estimated probabilities of 1!).

This is the code used to calculate the steady-state vector for the transition matrix and the result, and as expected, we see a steady-state vector that holds a large probability for a pain rating of zero:

```
#obtain steady state vector
```

```
#left eigenspace decomposition (requires transpose)
#extract real value and scale to components to
        sum to 1 due to floating errors
eigenvector <- softmax(abs(Re(eigen(t(transition_matrix))
        [["vectors"]][,1])))

round(eigenvector, digits=2)
```

We get the resulting steady state vector:

$$\vec{v} = \begin{bmatrix} 0.70 & 0.01 & 0.02 & 0.03 & 0.11 & 0.03 & 0.04 & 0.01 & 0.04 & 0.01 \\ 0.01 \end{bmatrix}$$

The steady-state vector was obtained by calculating the eigenspace of the transpose of the transition matrix (note that we used the transpose of the matrix because, as previosuly stated, the rows rather than the columns are probabolity vectors), as this corresponds to the eigenspace of the transition matrix. We then scaled an eigenvector so that the entries summed to 1, just like we explained in section 4. Floating-point errors are also accounted for through the application of a softmax function.

Finally, this is the code used to calculate the metric of interest (aka our weighted average of the steady state vector):

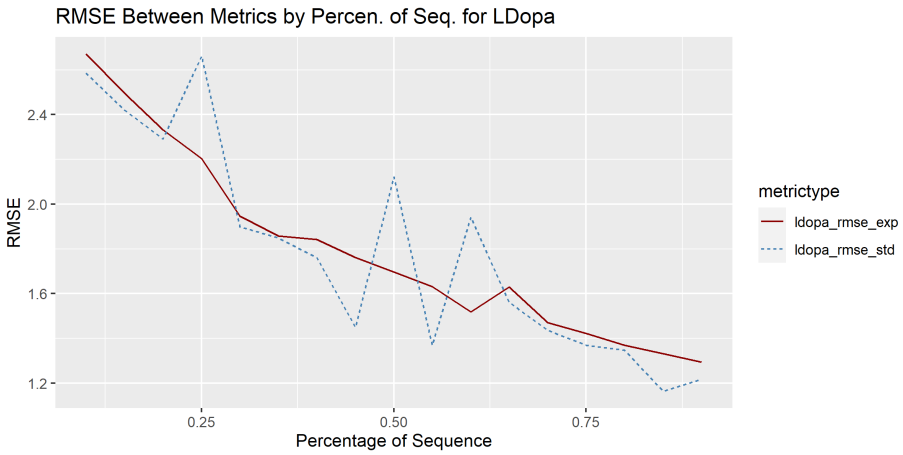
```
end_behavior <- (as.numeric(rownames(transition_matrix)) %*% eigenvector)

end_behavior

metric <- sequence[1] - end_behavior

metric
```

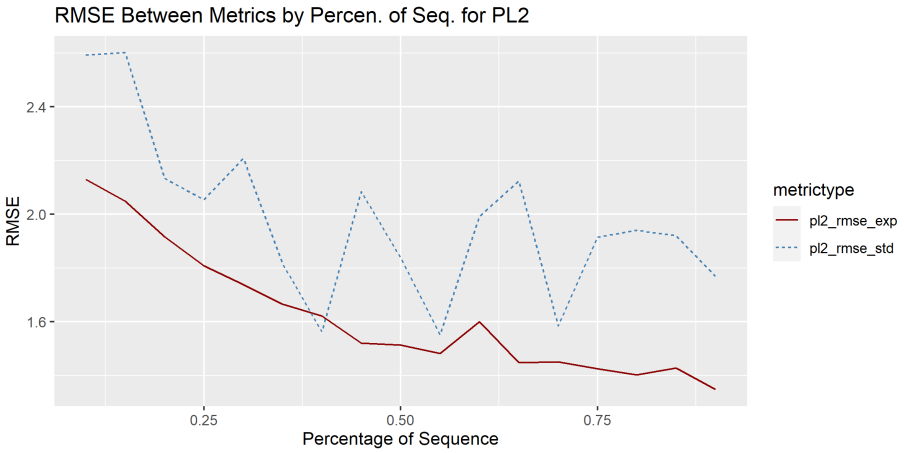
The code outputs a result of **8.57**!

This is similar to the traditionally calculated change-in-pain (-8), indicating that both metrics are in semi-agreement in the case of this subject. Differences in the two metrics will be explored between all subjects by replicating this process and testing for which reflects end behavior more accurately.

6.2 Metric evaluation

These are the following results for our experiment, detailed in "Methodology: The Experiment" (5.3). See the RMSE for the studies graphed below.





In both plots, the horizontal axis indicates the percentage of the full patient's time course (n) that was used for the calculating of our experimental metric and standard change-in-pain (step 1 in 5.3) and the vertical axis indicates the RMSE of both metrics at that specific percentage; **the experimental metric is colored in red and standard change-in-pain is indicated in blue**.

7 Discussion and Conclusion

Given the RMSEs obtained above, the usage of Markov models to model changes in pain **seems promising**. In Placebo II (the second chart), an apparent difference in RMSEs exist at most subset values, even as n approaches 1. At almost every point, the Markov model out-preforms the standard model. The difference of almost 0.5 in RMSE is notable, considering the metrics are calculated with data from 90% of the full sequence and the scale on which these metrics are being evaluated consists of 11 units.

RMSEs of both the Markov model and standard model are similar within the LDopa data (the first chart), indicating specific properties of the time series may specifically influence the accuracy of the model and that **further simulation and analysis of this model must be done** before any formal adoption or restructuring of research methods within the field. A possible direction includes **accounting for temporal resolution within the time-series**. Currently, models are fitted on discretely and evenly-spaced time points (pain per day), but somehow being able to transition between states 0 through 10 at non-evenly spaced time points would be beneficial.

With these considerations in mind, a goal to more accurately generalize and quantify patient outcomes within clinical trials and for patient insight still exists; with this understanding, the possibilities for understanding the nature of chronic pain conditions are vast. Characteristics of end-behavior (spread within the vector itself, the overall change in pain, etc.) can be related to clinical factors, demographic variables, etc. to understand the nature of pain trajectories. Specific types of end behavior can be explored between pain conditions to understand how the overall change in pain and total variabilities vary, and if clinical factors and demographic variables have further influences within these types of pain. Overall, we believe that the furthering of metrics to capture in pain are necessary for a better understanding of pain; without being able to properly quantify what it means for an individual's pain state to "improve", our understanding of chronic pain will be limited.

In conclusion, it is fascinating that while relatively straight forward, something as seemingly "simple" as a markov model can so drastically decrease errors in predicting chronic pain. We look forward to editing our model to better predict pain but also to applying markov models to other phenomena.

8 References

1. Apkarian, A. V. (n.d.). Open pain. Open Pain. Retrieved December 11, 2022, from https://www.openpain.org/

2. Applications to markov chains - MATH 1014. Math 1014. (n.d.). Retrieved December 12, 2022, from https://tqft.net/web/teaching/2015/1014/MATH1014-LinearAlgebra-Lecture10.handout.pdf

3. Chapter 8: Markov Chains - Auckland. Chapter 8 - Markov Chains. (n.d.). Retrieved December 12, 2022, from https://www.stat.auckland.ac.nz/ fewster/325/notes/ch8.pdf

4. Department of Electrical, Computer, & Biomedical Engineering - Toronto Metropolitan University. (n.d.). Retrieved December 11, 2022, from https://www.ecb.torontomu.ca/

5. Knill, O. (n.d.). Lecture 34: Perron Frobenius Theorem - Harvard University. Math 19b. Retrieved December 12, 2022, from https://people.math.harvard.edu/ knill/teaching/math19b_2011/handouts/lecture34.pdf

6. Levin, D. A. (n.d.). Markov Mixing - University of Oregon. Retrieved December 12, 2022, from https://pages.uoregon.edu/dlevin/MARKOV/markovmixing.pdf

7. Probability Vector, Markov Chains, Stochastic Matrix - UNESP. Probability Vector, Markov Chains, Stochastic Matrix. (n.d.). Retrieved December 12, 2022, from https://www.dcce.ibilce.unesp.br/ gsilva/math152/Ch4-9-lay-classnotes.pdf

8. R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

9. Reckziegel D, Ttreault P, Ghantous M, Wakaizumi K, Petre B, Huang L, et al. Sex-specific pharmacotherapy for back pain: a proof-of-concept randomized trial. Pain Ther. (2021). doi: 10.1007/s40122-021-00297-2

10. Vachon-Presseau E, Abdullah TB, Berger SE, Huang L, Griffith JW, Schnitzer TJ, et al. Validating a biosignature predicting placebo pill response in chronic pain in the settings of a randomized controlled trial. Pain. (2021). doi: 10.1097/j.pain.0000000000002450. [Epub ahead of print].

11. Yu. (2017, July 6). Eigenvalues of a stochastic matrix is always less than or equal to 1. Problems in Mathematics. Retrieved December 11, 2022, from https://yutsumura.com/eigenvalues-of-a-stochastic-matrix-is-always-less-than-or-equal-to-1/

A Implementation of Modeling

A.1 Metric calculation with n% of sequence

```
change_in_pain <- function(pain, n){

  #iterate through subjects to return matrix of scores
 scores <- sapply(unique(pain$subject), function(sub) {

   #extract sequence of pain ratings from subject
   full_sequence <- round(as.numeric(subset(pain,
           subject == sub)$pain), digits = 0)

   #calculate len of n% of sequence
   len <- round(n*length(full_sequence), digits = 0)
```

```
    #n% of sequence
    sequence <- full_sequence[1:len]

    #fit markov model
    transition_matrix <- markovchainFit(data =
            sequence)[["estimate"]]@transitionMatrix

    #obtain steady state vector
    #left eigenspace decomposition (requires transpose)
    #extract real value and scale to components to
    # sum to 1 due to floating errors
    eigenvector <- softmax(abs(Re(eigen(t(transition_matrix))
            [["vectors"]][,1])))

    #calculate metrics
    data.frame(subject = sub,
      #post - pre
      metric_tot = full_sequence[length(full_sequence)] - full_sequence[1],
      #end = weighted average of all possible pain values
      metric_exp = (as.numeric(colnames(transition_matrix))
            %*% eigenvector) - sequence[1],
      metric_std = sequence[length(sequence)] - sequence[1],

      #absolute difference between the metrics
      diff_exp = abs(full_sequence[length(full_sequence)] - full_sequence[1] -
            (as.numeric(colnames(transition_matrix))
            %*% eigenvector) - sequence[1]),
      diff_std = abs(full_sequence[length(full_sequence)] - full_sequence[1] -
            (sequence[length(sequence)] - sequence[1])))

  }
  )

  #reformat scores
  return(data.frame(t(scores)))
}
  )
```

A.2 Evaluation of metrics

```
  n_s <- c(0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45,
        0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9)

  #iterate through n's
  rmses <- sapply(n_s, function(n) {
```

```
    #obtain change-in-pain metrics for both studies
    ldopa <- change_in_pain(pain_ldopa, n)
    pl2 <- change_in_pain(pain_pl2, n)


    #calculate rmse for both metrics in both studies
    data.frame(
      n_s = n,
      ldopa_rmse_exp = as.numeric(rmse(as.numeric(c(ldopa$metric_tot)),
              as.numeric(c(ldopa$metric_exp)))),
      ldopa_rmse_std = as.numeric(rmse(as.numeric(c(ldopa$metric_tot)),
              as.numeric(c(ldopa$metric_std)))),
      pl2_rmse_exp = as.numeric(rmse(as.numeric(c(pl2$metric_tot)),
              as.numeric(c(pl2$metric_exp)))),
      pl2_rmse_std = as.numeric(rmse(as.numeric(c(pl2$metric_tot)),
              as.numeric(c(pl2$metric_std))))
    )

})

rmses <- as.data.frame(t(rmses))

ldopa_df <- rmses %>%
  select(n_s, ldopa_rmse_exp, ldopa_rmse_std) %>%
  gather(key = "metrictype", value = "rmse", -n_s)

ldopa_df <- as.data.frame(lapply(ldopa_df, unlist))

ldopa_plot <- ggplot(ldopa_df, aes(x = n_s, y = rmse)) +
   geom_line(aes(color = metrictype, linetype = metrictype)) +
scale_color_manual(values = c("darkred", "steelblue"))

pl2_df <- rmses %>%
  select(n_s, pl2_rmse_exp, pl2_rmse_std) %>%
  gather(key = "metrictype", value = "rmse", -n_s)

pl2_df <- as.data.frame(lapply(pl2_df, unlist))

pl2_plot <- ggplot(pl2_df, aes(x = n_s, y = rmse)) +
   geom_line(aes(color = metrictype, linetype = metrictype)) +
scale_color_manual(values = c("darkred", "steelblue"))


ldopa_plot+ xlab("Percentage of Sequence") + ylab("RMSE") +
      pl2_plot + xlab("Percentage of Sequence") + ylab("RMSE")
```

```
ggsave("fig.png", width=15, height=4)
```

A.3 Data Pre-processing

```
get_pl2 <- function(){
   ratings_pl2 <- read.csv("placeboII_ratings.csv",
           strip.white = TRUE, stringsAsFactors = FALSE)

   pain_pl2<-c()
   pain_pl2 <- (sapply(unique(ratings_pl2$subject), function(sub) {
     tmp = subset(ratings_pl2, subject == sub)
     tmp <- na.omit(tmp)

     if(sum(as.numeric(tmp$phase == 1)) > 0){
           cbind(subject = sub,
                 time = tmp$day,
                 pain = tmp$pain)
     }
   }))
   pain_pl2 <- do.call(rbind, pain_pl2)
   pain_pl2 <- na.omit(pain_pl2)
   pain_pl2 <- as.data.frame(pain_pl2)
   colnames(pain_pl2) <- c("subject","time", "pain")

   return(pain_pl2)
 }

 get_ldopa <- function(){
   ratings_ldopa <- read.csv("ldopa_ratings.csv",
           strip.white = TRUE, stringsAsFactors = FALSE)
   ratings_ldopa <- na.omit(ratings_ldopa)

   pain_ldopa <- (lapply(unique(ratings_ldopa$id), function(sub) {
     tmp = subset(ratings_ldopa, sub == id)
     tmp <- na.omit(tmp)

     start <- as.numeric(as.POSIXct(tmp[1,4], format="%d-%b-%Y"))

     cbind(
       subject = sub,
       time = (as.numeric(as.POSIXct(tmp[,4], format="%d-%b-%Y"))
               - start)/86400,
       pain = tmp$pain
     )
```

```
  }))

  pain_ldopa <- do.call(rbind, pain_ldopa)
  pain_ldopa <- na.omit(pain_ldopa)
  pain_ldopa <- as.data.frame(pain_ldopa)
  colnames(pain_ldopa) <- c("subject", "time", "pain")

  return(pain_ldopa)
}

softmax <- function(x) {x / sum(x)}

pain_pl2 <- get_pl2()
pain_ldopa <- get_ldopa()
```