Song Twenty-One: To Listen or Not to Listen, That's the Data-Driven Decision

Siddhant Sanghi
Columbia University
New York, NY 10027
ss5943@columbia.edu

1 Introduction

It is often when we think about Music Recommendation Systems while thinking about data-driven decision-making on Song datasets. In this project, I am considering the decision of the user to listen or not to listen to the next song from a **40000+ song dataset**[1]. Imagine that it is the due-date of an assignment deadline, and a college student wishes to take a break from their studies by listening to some music with the aim of getting an energy-boost from it. **Inspired by the one-player version of the game twenty-one**[2] **from the last course assignment**[3]**, I will be modeling the decision of whether or not to play the next song as a Markov Decision Process (MDP)**.

The objective is for a college student to listen to songs during a study break on the due-date of an assignment, one at a time from a 40000+ song playlist and acquire a cumulative energy level as large as possible without going over the threshold value of 21, after which one is too distracted and energized to study, and then they cannot finish their assignment and they are "bust". This value is dependent on 'energy' audio feature from Spotify's API.

At each turn in the study break, the college student has one of two actions: **"play" to play the next song, and "stop" to stop listening to music**. We will only study scenarios in which the cumulative energy level is greater than 11, since the optimal decision is always to play the next song otherwise (since each song can at maximum have a energy level of 10, and the student won't get distracted, and they will go back to studying, hence finishing the assignment and not going "bust").

If the student has a late day for the assignment, that gives them the power like a usable ace in the game twenty-one not to get distracted and possibly fail to submit the assignment in time (not to go "bust"). **Just like a usable ace only in function (but not in any other likeness), we let the late day have either value 1 or 11**, as long as the cumulative energy level is not greater than 21 (going "bust"). We call the latter situation a "late day" and track this as part of the state.

2 Dataset

For my project, I have chosen **The Spotify Hit Predictor Dataset (1960-2019)**[1], a dataset of 41106 songs and their audio features from Spotify's API ranging from 1960 to 2019 (across six decades). These audio features are very informative and range from 'danceability' to 'key'. Each song has a corresponding 'energy' value between 0 and 1, that represents a perceptual measure of intensity and activity.[1]

I chose the **'energy'** feature as through exploratory data analysis, I found that 'energy' is positively linearly correlated with statistical significance ($p < 0.05$) to 'valence' or emotions that represent happy, cheerful songs closer to 1 and sad songs closer to 0, and I wanted to find a feature that would represent a mood-boost after listening to it during the study break.

Typically, **overly energetic tracks feel fast, loud and distracting**[1], and thus, there is a threshold value for energy level assumed as 21 (like twenty-one game) in cumulative energy level since then the student will get too energized and distracted and "bust" because they are likely not to finish the assignment by the deadline.

I have discretized and binned this continuous value into **10 energy levels**, from 1 to 10 as follows:

$$\text{Energy level} = \lceil \text{Song energy value} * 10 \rceil \tag{1}$$

In addition to linear regression of energy with valence, **I ran multinomial logistic regression of energy levels with valence to ensure that my discretization didn't change the positive correlation to valence**. I found the same statistically significant ($p < 0.05$) increasing positive correlations between increasing energy levels and 'valence' as compared to baseline energy level 1, as before the discretization.

3 Methods

3.1 Defining the Markov Decision Process (MDP)

My set of **states** will consist of tuples: a Boolean indicating whether we have a late day, and an integer between 12 and 21 indicating the current energy level sum of the college student. I generated a list of the states through taking the cross product of [0,1] x [12,...,21]. There are 20 states in total, seen below.

```
States = [(0, 12), (0, 13), ..., (0, 21), (1, 12), (1, 13), ..., (1, 21)]
```

We have just two **actions**, and we will define the **utilities** of each as follows:

- 0 or The **"stop"** action gives an instantaneous utility that equals the current energy level sum. Since the college student stops listening to music after this, there is zero future utility.
- 1 or The **"play"** action gives zero instantaneous utility plus a discounted expected future utility. The expected future utility can be found by computing a matrix-vector product of the transition matrix with the current state values.

The **transition matrix** T for the "play" action can be defined as $T_{ij} = P(j|i)$ and i and j index the states as described above. Each row corresponds to a current cumulative energy level state, and each column corresponds to the next cumulative energy level state after playing the next song.

3.2 Transition Matrix for the "Play" Action

For each energy level i, the probability that that level is seen is $P(i)$ stored as a length-10 vector called $probabilities$:

$$P(i) = \frac{\text{Number of songs with energy level i}}{\text{Total number of songs}} \text{ for } i \in [1, 10] \tag{2}$$

I constructed the 20×20 transition matrix T for the "play" action as described in Step 1.

The first 10 rows correspond to current energy level sums without a late day. From these states, there are valid transitions to any other state with a greater card sum and no late day, each with probability = $probabilites[b - a] = p_{ab}$where a, b ∈ [1,10], seen in the equations below. **We do not include any "bust" states (i.e. energy level sum greater than 21), so these rows will not sum to 1.**

The last 10 rows correspond to energy level sums with a late day. If our new energy level sum is $e \leq 21$, we have not had to use the late day to reduce the energy level sum to not go "bust" and so we still have a late day. As a result, the successor state is simply $(1, e)$. Otherwise $e > 21$ and the successor state is $(0, e - 10)$ since we have now used the late day and the energy level has reduced because we no longer have the 'veto power' of an extension and now from energy level 11 we are at 1. **These rows should all sum to 1, since no choice of next song can lead to distraction from on-time submission.**

$$T_{20\text{x}20} = \left(\begin{array}{c|c} \mathbf{A}_{10\text{x}10} & \mathbf{B}_{10\text{x}10} \\ \hline \mathbf{C}_{10\text{x}10} & \mathbf{D}_{10\text{x}10} \end{array}\right) \tag{3}$$

Here $\mathbf{A}$ shows the transitions from no usable late day state to no usable late day, which is the same as $\mathbf{D}$, representing the transitions from usable late day to usable late day. In both cases, we have only

p_{ab} for $b > a$ as described above. **B** shows transitions from no usable late day to usable late day, which is impossible assuming that you can't get an extra late day if you don't have any, and hence is a zero matrix. **C** shows transitions from usable late day to no usable late day, which is in the case that you use the late day, and hence we have a lower triangular matrix with all non-zero entries p_{ab} for $b \leq a$. Thus,

$$T = \left(\begin{array}{cc|cc} 0 & p_{ab} & \multicolumn{2}{c}{\multirow{2}{*}{\mathbf{0}}} \\ 0 & 0 & & \\ \hline p_{ab} & 0 & 0 & p_{ab} \\ p_{ab} & p_{ab} & 0 & 0 \end{array}\right) \tag{4}$$

where I'm just showing the 10x10 blocks **A, B, C, D** as 2x2 simplifications for representation, and p_{ab} is defined above.

I have displayed the row sums of my constructed transition matrix below, that follows what should be expected.

```
Sum across rows = [0.95951929 0.85165183 0.72595242 0.59421982 0.45874082 0.33231158
 0.2143726  0.1199095  0.05057656 0.         1.         1.
 1.         1.         1.         1.         1.         1.
 1.         1.        ]
```

3.3 Value Iteration Function

I wrote a value iteration function to find the values of each of the states corresponding to song energy level sum. The inputs are the transition matrix, discount factor, and convergence threshold.

The discount factor β, in my case is in terms of the constant **baseline stress level** η associated with the assignment, that the college student feels on the due-date.

$$\beta = 1 - \eta \tag{5}$$

The discount factor is a measure of optimism under uncertainty with β close to 1 showing optimism (risk-seeking) and β near 0 showing pessimism (risk-averse) behaviour. **The stress level η is inverse to** β, i.e, high η levels mean low discount factor β, which means more pessimism under future uncertainty.

The output is the set of converged energy level values stored in a length-20 array.

4 Results and Discussion

I have written code in a Jupyter notebook **SongTwentyOne.ipynb**, that implements Steps 2 and 3 outlined in the Methods section above.

Further, I ran instances of value iteration with the following values of baseline stress level: $\eta = 0$, $\eta = 0.1$, $\eta = 0.5$ and $\eta = 1.0$. For each one, I produced two scatter plots showing the optimal values and optimal policies of the states 12-21, overlaying both the sets of values/policies without and with a late day on the same plot. This is Figure 1 below.

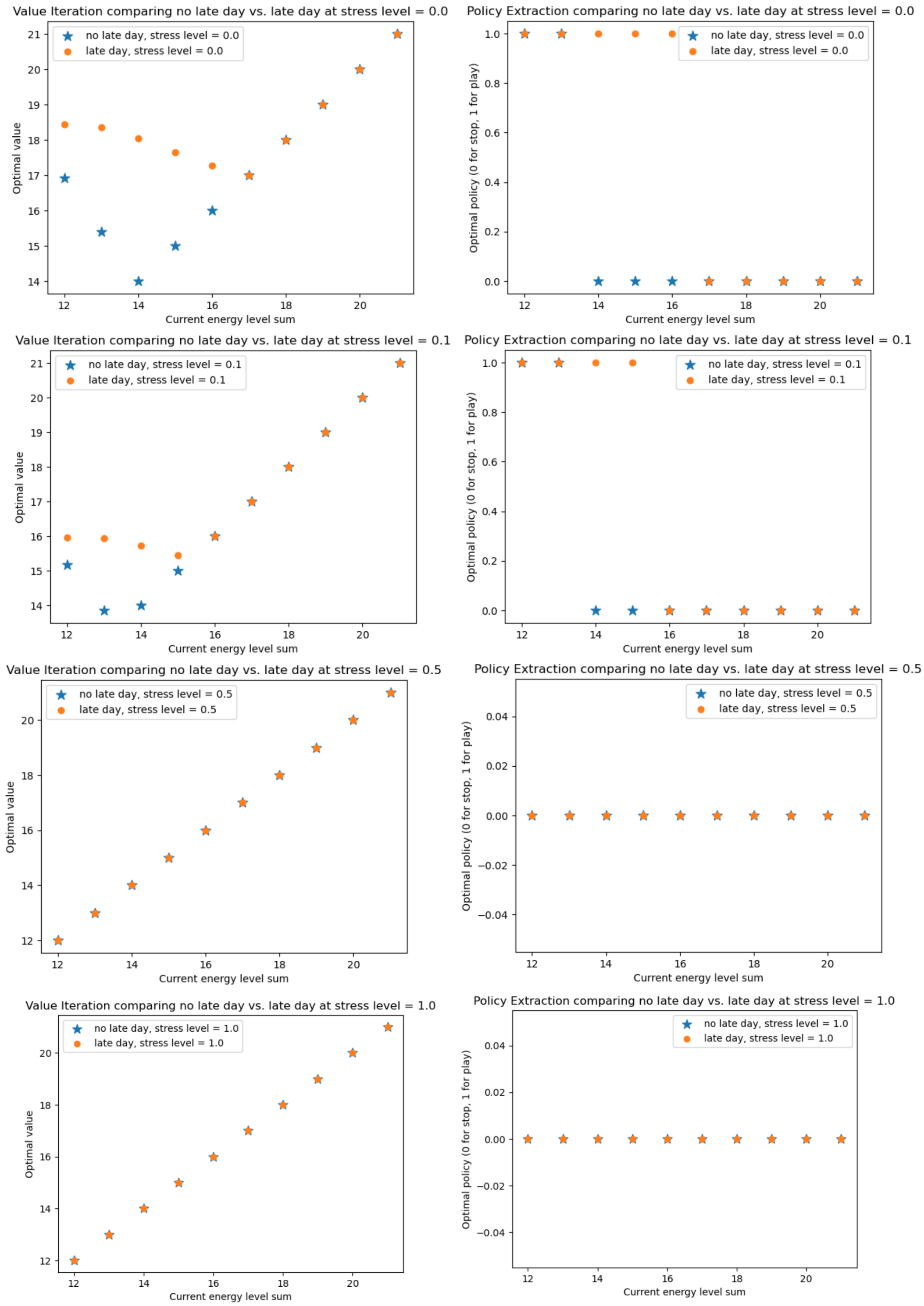

Figure 1: Scatter Plots for Value Iteration and Policy Extraction for Baseline Stress Levels $\eta = 0$, $\eta = 0.1$, $\eta = 0.5$ and $\eta = 1.0$ (from top panel to bottom panel) showing the values of the cumulative energy level states 12-21, overlaying both the sets of values without and with a late day on the same plot

Based on my plots in Figure 1, the optimal strategy is to **play the next song** until cumulative energy level 13 and **stop listening to music** with energy level sums ≥ 14 in the case of no late day, and low baseline stress levels $\eta = 0, 0.1$. This changes with an increase in baseline stress level $\eta \geq 0.5$, to make us always **stop listening to music** for all energy levels ≥ 12 because the college student is sufficiently stressed that they won't finish the assignment by the deadline, and won't have a late day to use.

For $\eta = 0.0$, the optimal actions in states with a usable late day are to **play the next song up until the sum of 16 and stop listening to a music for a sum** ≥ 17 for sums ≥ 12. For $\eta = 0.0$, no usable late day and energy level sums ≥ 12, the optimal action is **play the next song up until the sum of 13 and stop listening to a music for a sum** ≥ 14. The reason for the discrepancy in optimal values is because the college student is optimistic under uncertainty when they have a late day (as a kind of a veto power) and wants to play next song in their study break so they can possibly get a higher energy boost in in the future.

The optimal actions in states with usable late day (with sums ≥ 12) are to play the next song up to cumulative energy level 16 for $\eta = 0.0$ and stop listening to music for any sum greater than that, play the next song up to energy level 15 for $\eta = 0.1$ and stop for any sum greater than that, and stop listening to music for all cumulative energy levels for $\eta = 0.5$. **The optimal actions depend on the value of stress level η, as η represents pessimism under uncertainty.** Higher η close to 1 penalizes expected future utility highly if the decision made is to play the next song and so we always stop listening to music for $\eta \geq 0.5$, while lower baseline stress levels η closer to 0 mean the college student is optimistic under uncertainty and the optimal action is to play the next song. Thus, as η decreases we become less stressed and more optimistic under uncertainty and stop at an later cumulative energy level (≥ 17 for $\eta = 0.0$, rather than ≥ 16 for $\eta = 0.1$) as our optimal strategy.

5 Contributions, Limitations, Future Work

The idea of using MDP's to model the decision of a user listening to music, similar to the one-player game of twenty-one, is novel. I believe that this paper contributes to the realm of Data-Driven Decision Making because of this novelty. Further, I have utilised the Spotify audio feature of energy as a score for mood level due to its positive correlation with and better interpretability than the feature of valence, exploring in greater depth this feature from Spotify's audio analysis API.

In terms of limitations, I believe that **the single feature of energy may be too little to score a song by**, and I could try to leverage many other brilliant features from the dataset like 'danceability', 'valence', 'loudness' and so on. Further, the transition matrix probabilities in my project are constructed from data of all 40000+ songs, but should probably change to using a smaller subset like a user-defined playlist or a group of recommendations or songs by the same artist.

For future work, **I would like to make a simple app called 'Song Twenty-One' with a UI that simulates this decision making process like an interactive game for college students**, and further I would like to create the basic Music Recommendation System based on K-Nearest Neighbors for a song feature vector and a chosen metric, modifying the transition matrix with a smaller chosen song subset.

6 Acknowledgements

I would like to thank Professor Tony Dear and Professor Yi Zhang for the wonderful course, and the guidance through office hours in discussing ideas for the project.

References

[1] Ansari, F (2019). The Spotify Hit Predictor Dataset (1960-2019). `https://www.kaggle.com/datasets/theoverman/the-spotify-hit-predictor-dataset`

[2] Twenty-One (Banking Game). Wikipedia. `https://en.wikipedia.org/wiki/Twenty-One_(banking_game)`

[3] Dear, T & Zhang, Y (2022). Assignment 5, 'Twenty-One'. `https://edstem.org/us/courses/28294/lessons/49054/slides/279600`