Applications of the Principle of Least Effort in Data Transformation

by

Luke Veenhuis

A thesis submitted in partial fulfillment of
the requirements for the degree of

Master of Science

Computer Science

At

The University of Wisconsin–Whitewater

December, 2019

Graduate Studies

The members of the Committee approve the thesis of
Luke Veenhuis presented on December 6th, 2019

Dr. Hien Nguyen, Chair

Dr. Athula Gunawardena

Dr. Jiazhen Zhou

Applications of the Principle of Least Effort in Data Transformation

By

Luke Veenhuis

The University of Wisconsin-Whitewater, 2019 Under the Supervision of Dr. Hien Nguyen

I am interested in applying the Principle of Least Effort to data transformation in an effort to solve three important challenges in using video games as a testbed for the study of inverse reinforcement learning. These challenges are as follows: the very large state space created by high granularity of time, the very large range of feature values provided by quantitative features, and the difficulty to measure similarity of trajectories. Through exploring The Principle of Least Effort from the Social Sciences, I propose a form of data transformation that categorizes data according to the familiarity and preferences of the modeled user. Furthermore I will also show how this approach can be used to create a similarity comparison between trajectories in accordance to The Principle of Least Effort. For the collection of test data I have utilized a reinforcement learning agent to play the minigame *BuildMarines* for StarCraft II as provided by DeepMind.

ACKNOWLEDGMENTS

I thank the Naval Research Laboratory in their instruction on deep reinforcement learning. Furthermore, I thank the faculty in the computer science department at the University of Wisconsin-Whitewater. Most notibly I thank my advisor, Hien Nguyen, for being patient when I was not.

DISCARD THIS PAGE

TABLE OF CONTENTS

Page

DISCARD THIS PAGE

LIST OF TABLES

DISCARD THIS PAGE

LIST OF FIGURES

Chapter 1

Introduction

In user modeling, the goal is to observe some agent's behavior and infer some explanation for why the behavior emerged as it did. Inverse reinforcement learning, first described by Russell [6], approaches this through an opposite of classical reinforcement learning where once given measurements of an agent's behavior over time, measure of an agent's sensory inputs, and a model of the physical environment in which this behavior is being observed, determinization of a reward function of that agent is attempted. Later, the first attempt at an IRL approach was attempted by Ng and Russell [5]. In this foundational approach, IRL would take as input an MDP (Markov Decision Process) without a reward function and attempt to approximate a reward function. Though they were not able to properly formulate a viable reward inference, the MDP was maintained as the primary structure on which inverse reinforcement would be attempted.

In this research, I have attempted to identify and solve a few problems introduced by very rich data, namely: the growth of state space, feature space, which causes complexity problems for IRL. The first problem of note comes from the continuous nature of many games. That is to say; there is no natural discrete representation of the flow of time, such as players taking turns; rather, time flows continuously from each second to the next. In order to describe the game data in a manner appropriate for an MDP, each discrete time step must be recorded as a separate node. Often these discrete time steps are very small, occurring approximately every second or less. This creates a very rapidly growing space in which each timestep, or state, is

represented (known as the state space). This is problematic because as the state space grows, the computational complexity of Inverse Reinforcement Learning begins to grow exponentially.

For the problem presented by the high granularity of time, it is reasonable to state that some amount of reduction is appropriate. It is intuitive to say that we, as human agents, rarely consider each brief time step individually, but rather, we lump time steps into ill-defined context-aware groups we might call moments. Some approach is needed to either remove states or combine states such that the granularity of time is reduced without sacrificing more information than is necessary.

A second problem comes from the high granularity introduced by quantitative features. That is to say, quantitative features often introduce a wide range of possible values. For example, this might apply to a representation of how much health an agent has or a representation of some number of needed resources that have been collected by an agent. In many domains, the range of possible values can be quite large. Previous experience has seen ranges thousands or tens of thousands in size though an upper limit cannot be stated as an upper limit would be defined by the rules that govern each domain. This contrasts with more nominal features in which the range of possible values rarely exceeds a dozen or so, though it should be stated that any upper limit is still defined by the rules of each domain. This becomes problematic when constructing an MDP because as each state becomes increasingly likely to become unique as the range of possible values increase per feature as well as the number of quantitative features used to represent each state.

Previous approaches have existed to "bin" univariate quantitative data, such as Jenks natural breaks optimization [3], though these approaches lack the nuance necessary to observe context given by time-series data. In some domains, for example, DOTA 2, it may be that 100 units of health may be a large number in the context of early in periods of each game but may be considered a very small number in later periods of a game. It would seem that a new approach is necessary to capture this need for context. A third difficulty, though not one particularly novel to the domain of video games, is the difficulty of determining the similarity between datasets, often called trajectories or narratives, used to construct MDPs. Though we

can intuitively describe some trajectories as being similar or different through our own experiences, this becomes very difficult to approach objectively without a single objective measure available. The ability to comparing trajectories becomes useful when one wishes to compare inferred reward values between trajectories that are very similar or perhaps very different.

To approach this problem, I am proposing a new data transformation approach. Each of these three difficulties stems from an insufficient approach to context-aware categorization. I wish to suggest that this context-aware categorization must be defined through ideas borrowed from previous work in the social sciences to describe human behavior for what better way to designate context than the designation already being done inside the mind.

From this need of context-aware categorization, I am drawing from an idea known as The Principle of least effort as described by Zipf in his book "Human Behavior and the Principle of Least Effort." [11] The Principle of least Effort is described as a dominating desire to select paths, or what we have defined as trajectories, in which the least amount of work is expended.

Not all trajectories are perceived as equal to the mentation of the decision-making agent, and thus preference is formed through the familiarity formed through repetitive trajectory exploration. If the preference towards certain trajectories is determined by which trajectories expend the least amount of effort, then it should be noted that the approximation of work expended is also the expenditure of work, and thus reliance on familiar trajectories is formed rather than attempting which trajectories will guarantee the highest completion of objectives. In other words, capturing a certain degree of laziness is necessary if we are to describe any context-aware categorization.

Statistical approaches have been used to study the Principle of Least Effort in the field of linguistics. When the frequency of words in a given text document or corpus of documents, a very clear trend in ranked distribution can be observed; this distribution is now referred to as the Zipf distribution. The cause of this trend has been described by Zipf as the economy of tools and jobs. Preference towards certain tools grows as the usefulness of those tools becomes recognized. It has been stated by Zipf that the presence of this distribution provides evidence

of preferred tools in the completion of an objective. It will be through these approximate distributions that the data transformation will be evaluated.

For the purposes of collecting data, I have elected to use the BuildMarines minigame for StarCraft II, as described by DeepMind in 2017 [8]. To ensure the availability of a large corpus of trajectories, I have elected to use a reconstruction of the FullyConv model described in the same publication. Normally this model is not sophisticated enough on its own to capture the complexities of this minigame, so I have created a series of transfer learning tasks to ensure a sufficient level of performance is achieved. These transfer learning tasks will be described in detail during the approach section to ensure reproducibility is possible.

This thesis is organized as follows: In chapter 2, I will briefly discuss some background in categorization, namely data clustering approaches, and why these approaches are not sufficient enough on their own for the purposes of clustering rich time-series data. In chapter 3, I will discuss a small history on The Principle of Least Effort, including early observations of this behavior in the laboratory and also how simple statistical approaches have observed the principle in a domain as seemingly complex as linguistics. In chapter 4, I will discuss the creation of the StarCraft II testbed and how it was particularly useful for studying this principle for the purposes of transformation. In chapter 5, I will briefly discuss how well the transformation performs and whether it can be considered viable. Finally, I will conclude by discussing how the transformation can be applied to solve the previously outlined problems.

Chapter 2

Background and Related Work

One approach to categorization problems is that of data clustering. In data clustering, data points are mapped to an n-dimensional space and are divided into naturally occurring groups. These groups are generally divided according to their density among one another measured by some predetermined distance function though Euclidean distance is a popular choice.

2.1 Partition Relocation Clustering

Partition-relocation clustering is done through the process of generating a random cluster and iteratively relocating data points to more appropriate clusters until satisfactory clusters have been determined [1]. The most popular clustering algorithm, k-means [4], is an example of this approach. In this approach, the centroid of each cluster is calculated as the average of each point belonging to it. The cluster each point belongs to is iteratively reevaluated by measuring the Euclidian distance to each centroid.

2.1.1 Liao's Method

In 2006 Liao proposed a 2-step partition relocation approach into clustering multivariate time series data [9]. In this approach, the time feature is first stripped from the data, which is then clustered through any preferred clustering technique. For the second step, all data sets are concatenated into a single file of univariate data representing the time-ordered clusters from the first step. Then a transition matrix is created to describe the probability of one cluster

transitioning into another. Finally, the number of converted clusters at once again clustered into a time-sensitive cluster.

While interesting, this approach is fundamentally lacking in data where there is no ground-truth specified number of clusters, as is often the case. If there is no ground truth for both the first step and the second step, then it can become very difficult to determine and accurate clustering model. As such, the need for a more robust approach is crucial in this area.

In the following sections, I discuss ideas borrowed from the field of psychology that attempt to describe biological behaviors in simple quantifiable terminology so that we can represent data in these terms. Later I will show that this terminology is applicable to artificial reinforcement learners in time series environments.

2.2 The Principle of Least Effort

Zipf [11] presented that for all available paths through which an individual can move, the selection of that path would be governed by the Principle of Least Effort. Through the Principle of Least Effort, "a person in solving his immediate problems as estimated by himself and attempts to solve his problems in a way that minimalizes total work towards immediate and probable future problems." The main idea here is that an intelligent agent (biological or artificial) does not desire to expend more energy than deemed necessary. Each agent may have different methods of determining predicted energy expenditure, but ultimately all decision-making will be governed through this.

This need to discuss the path selection process through the least effort comes from the concept of "The singleness of the superlative." Zipf stated that "no problem in dynamics can be properly formulated in terms of more than one superlative, whether the superlative in question is stated as a minimum or as a maximum. If this problem has more than one superlative, the problem itself becomes completely meaningless and indeterminate. It cannot be stated that an intelligent agent makes decisions on path selection through a multivariate consideration.

In the dynamic system that is the environment that the agent decides in, there must be one superlative that is maximized or minimized, this superlative then must be the average rate

of work expenditure. This particular superlative minimization is then independent upon the mentation of the individual. This mentation includes comprehending the relevant elements of the model, accessing the probabilities of those elements, and solving the problem in terms of least effort. This mentation is used to calculate the cost of each possible path. Of course, the cost of calculating the cost of the least effort must be factored into the overall cost of each path. This is an expensive process because cost calculation also requires energy, and thus, consistently taking familiar paths becomes preferable. This is the basis of forming a habit. This can be demonstrated in the laboratory experiments of mice.

2.2.1 Of Mice and Mazes

Experimental results have also shown the existence in the preference toward the least effort in laboratory settings. In work done by R. H. Waters [10], rats were introduced to a maze in which there were multiple equidistant minimal paths, as shown in the figure below. The question pursued by this experiment was, "Do rats prefer pathways containing the fewest turns?"

From the results, 4 statements of truth were given. In pattern A, rats preferred the paths containing the minimal number of turns. In pattern A, rats preferred the paths containing the minimal number of turns in pattern B rats to take the pathway containing the minimal number of turns. When presented, a straight path gradually becomes preferred. The gradual shift in preference is reduced when the maze is shifted.

2.2.2 Tsai: Relative Weight of Resistance

Another researcher of rats, Dr. Loh Seng Tsai [7], published a large series of experiments testing preferences towards decisions that either expend the least amount of effort or maximize a reward. For the purposes of this writing, I will discuss only the experiments that tested decision-making that prioritized the least effort expenditure.

In the first series of experiments, a group of fourteen rats was introduced to a simple maze in which there were only identical straight paths to a food source. Halfway through each pathway, a door had been installed to only open in one direction with the option of attaching

weighted resistance to the door. The question of this research was, “given an option between two paths, will a path of least resistance be chosen?” In this series, three different experiments were conducted.

In the first experiment, there was no resistance attached to the left path, and 20 grams of resistance were attached to the right. Each rat was given 10 trials to reach the end of the maze for a total of 140 trials. Of the 140 trials, 124, or 88.6%, thus resulting in the path of least resistance being chosen.

In the second experiment, the 50 grams of resistance were attached to the left path, and no resistance on the right path. The path of least resistance was intentionally switched from the previous experiment to account to break any established behaviors. To allow for the rats to learn that a new path of least resistance existed, 50 trials were given as opposed to the 10 trials of the previous experiment. After 50 trials, the path of least resistance was chosen 76.4

One final experiment in this series was conducted. In this experiment, 20 grams of resistance were attached to the left path, and 50 grams of resistance were attached to the right path. Again the path of least resistance was switched from the previous experiment to account for and previously established behavioral patterns. To account for the new selections in the path, another 50 trials were given to the rats. After 50 trials, the path of least resistance was chosen 82.9

2.2.3 Tsai: Relative Height of Obstacles

The second series of experiments consisted of 25 rats climbing over walls of different heights. The objective was “to test whether white rats would choose the shortest path in terms of height or vertical distance for an equivalent satisfaction of a certain organic need.” In other words, will rats prefer climbing over the shortest wall to obtain food?

In the first experiment, the rats were placed in a square enclosure in which each of the four walls was of different heights: 25, 35, 45, and 55 centimeters, respectively. Food was placed outside of each of the walls, and the rats were given 12 successive trails to climb over the walls

and obtain food. After all 12 trials were completed for each of 25 rats, for a total of 300 trials, 234 or 78% of all trials resulted in the shortest wall being climbed over.

In the second experiment in this series, a simplified approach was taken in which only a choice of two different heights was given. The rats were still placed inside of a square enclosure, but each pair of two walls shared the same height, which was 25 and 35 centimeters each. Another 12 trials were conducted on 25 rats for a total of 300 trials. Of these 300 trials, 96.3% selected the path of least resistance.

From these lessons, we can infer that, despite all else being equal, there is a strong preference for any path in which the least amount of work is expensed. This will be used as a strong motivator for describing behavior in terms of the Principle of Least Effort.

2.3 The Economy of Tools and Jobs

In this section, I will describe how decisions as tools can be observed to follow the Principle of Least Effort. In the second chapter of Zipf's book, he describes linguistics as an 'Economy of Words.' Language, being a key component in much of human behavior, is an incredibly useful tool to the point that going without language can be incredibly detrimental towards a good quality of life.

Instead of describing language itself as a tool, it may be more apt to describe it as a toolbox and the vocabulary used as the tools. Given a conversation between a listener and a speaker, there is an economy taking place in which both actors desire to expend the least amount of work possible.

From the perspective of the speaker, the ideal economy would consist of a one-word vocabulary in which all possible meaning can be conveyed. With such a simple vocabulary, we must assume that this one word must somehow be capable of all meaning possibly utterable despite how impractical this obviously would be in the real world. This economy would be ideal for the speaker because there would be no work necessary to formulate a sentence. A simple reliance on a single all-purpose tool is all that is needed.

From the perspective of the listener, the ideal economy would exist in the most opposite state as that of the speaker. The listener's ideal economy would consist of a vocabulary in which many words exist to match the many possible meanings utterable. In fact, if there would be a finite number, x, amount of utterable meanings, then there would be exactly x amount of words to utter.

This, of course, is where a compromise must be made, much like any other economy. In order to allow for communication to become a tolerably efficient enough task, the vocabulary used must be shaped in such a way that both the speaker and the listener need not expend too much work. As such, it can be assumed that the distribution of word use must form as some words take on many meanings and use, for the ease of the speaker, and other words become reserved for specific uses, for the ease of the listener. The desire for a small vocabulary is described as a Force of Unification, and the desire for a large vocabulary is described as a Force of Diversification.

2.3.1 Empirical Evidence of the Economy of Tools and Jobs

The primary piece of experimental evidence cam from the work of Dr. Miles L. Hanley, who performed an analysis of the novel Ulysses. This analysis is rather straight forward in that it observes the frequencies of word unique word uses throughout the book in what is now called the Hanley Index. As described by the Hanley Index, Ulysses contains 29,899 unique words throughout a running total of 260,430 words. It is important to note that of these words, approximately 20% of the most frequently used words account for approximately 80% of all words used. I will discuss this importance later on in the next section.

For a more holistic view of word distributions in the English language, I present another distribution from the Corpus of Contemporary American English. It can clearly be seen that the distribution followed here is quite comparable to that of Ulysses, and I have also included another visualization of the distribution of words that shows that the 20% most frequently used words account for approximately 80% of all occurrences.

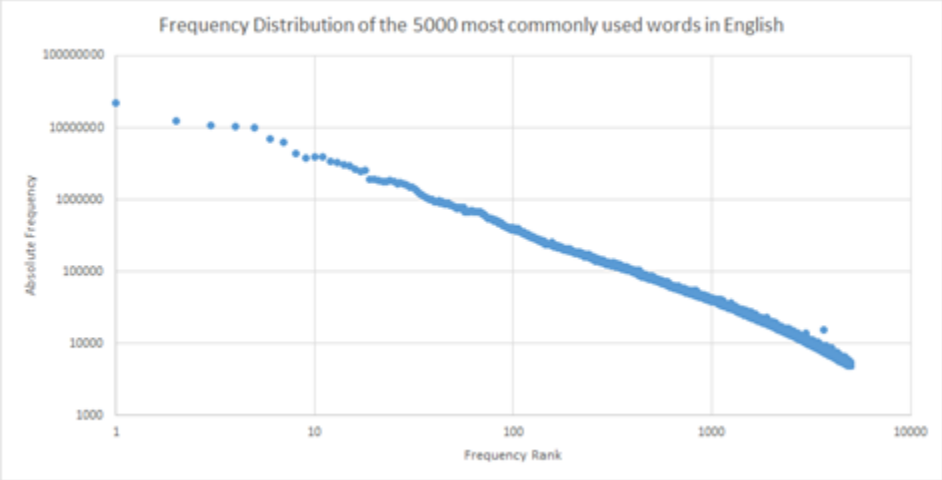

Figure 2.1 Frequency Distribution of 5000 most commonly used words in English

The recognition of this distribution will be necessary for any transformation that desires to capture an agent's mentation for context-aware categorization. First, a designation of "tool" is necessary to search for a Zipf distribution in the trajectory of a videogame. For this, I have chosen to describe a term I will refer to as a "feature-moment." A feature-moment can simply be described as the value of a feature at a given timestep. As an example, for a "collected resources" feature, a value of 50 at the 5th timestep is a feature-moment, and a value of 300 at the 10th timestep is another feature-moment. Later in this paper, I will show that these feature-moments will often approximately follow a Zipf distribution how that may be leveraged to solve the problems I have described with using inverse reinforcement learning in a video game domain

2.4 Confirmation of The Principle of Least Effort

An approximate confirmation of The Principle of Least effort can be done in two ways (outside of a simple qualitative examination of the log-log graph). First and foremost, according to Zipf's Law, the top 20% of most frequently occurring possible observations will account for approximately 80% of all observed observations. Conversely, the bottom 80% of possible

observations will account for approximately 20% of all observed observations. This verbiage may be difficult to understand but is quite clear with the previously shown examples.

Given the Corpus of Contemporary American English, the top 20% of the most frequently occurring words account for 83.2% of all word usages, and the bottom 80% of frequently occurring words account for 16.8% of all word usages.

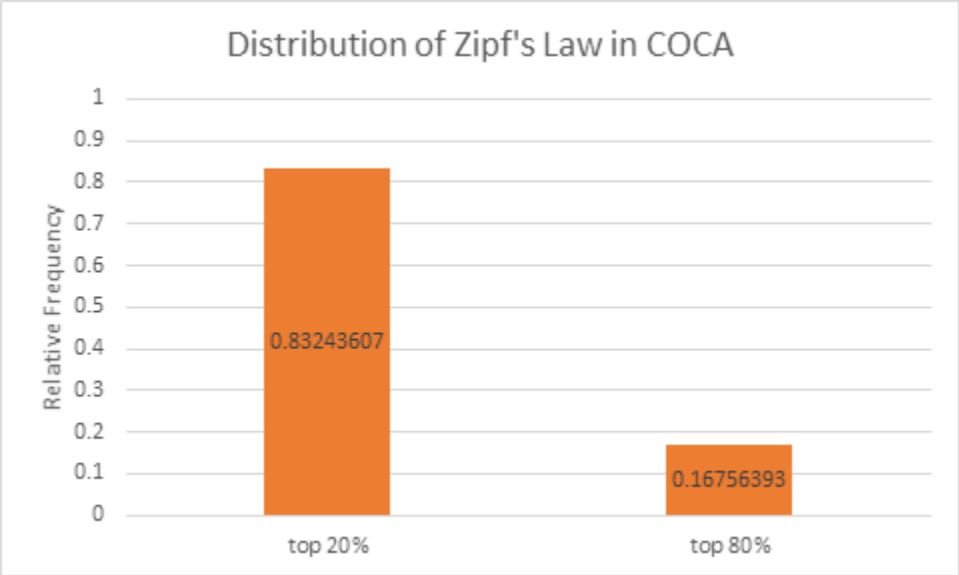

Figure 2.2 Distribution of Zips Law in COCA

Now it is true that the previously discussed experiments of mice and mazes do not immediately appear to follow this simple distribution. In experiments with only two options, one would assume that the option providing minimal effort should have been chosen approximately 93% of the time, whereas, in these experiments, the preference seems to hover at approximately 80%. At the time of this writing, there is not a clear method of determining the exact cause for this, but I would argue that it is likely due to mentation still in formation. That is, the experiments were designed in such a way that the rats were not able to rely on simple familiarity and in fact, were designed to confuse the rats into developing new mentations with each trial. It would be interesting to repeat the experiment to determine if the distribution follows Zipf's law upon settled mentation.

To ensure that this data transformation captures some aspect of the agent's mentation of the Principle of Least Effort, then it is pertinent that we ensure that our transformed data adheres to these rules of approximation.

Chapter 3

Testbed

To capture this Principle of Least Effort, I have elected to use the FullyConv [8] model to play a simple minigame as described by DeepMind [8]. In the FullyConv model, spatial actions are made within a state space where the state space is represented by a concatenation of metadata provided by the SC2LE Environment, a wrapper for StarCraft II for AI development.

Minigames were developed by DeepMind and used to establish baseline performances amongst different models in the video game StarCraft II. The minigames cover a range of tasks ranging from simple, such as MoveToBeacon, to complex, such as BuildMarines; the minigame task I've chosen to target is BuildMarines.

The primary challenge was to create a FullyConv model that can perform consistently better than random on the BuildMarines minigame. To accomplish this, I chose to do a series of transfer learning tasks and a small reduction in action space to encourage a policy to shape towards building many marines. To do this, I stripped down the action space immensely and build action space back up slowly over a period of time. First, let us understand what the original action space of build Marines looked like.

BuildMarines is comparatively simple to StarCraft II but is still complicated enough to be a challenge for a fully convolutional model. It requires the model to learn to gather resources, use those resources to build supply depots, use the supply depots to unlock building barracks, build barracks to train marines, and finally use the barracks to build as many marines as possible.

In order to facilitate the training process of the FullyConv model, a series of transfer learning tasks were utilized. A minigame was created for each step in the overall process of building

marines. First, the AI was taught to build Supply Depots, then Building Barracks, and finally on building Marines. These minigames also had a reduced action set to minimize training time.

In order to begin training on building supply depots, a map similar to the BuildMarines map was created called BuildSupplyDepots. In this map, the worker units were heavily modified such that they could not move to specified locations unless they were moving to collect resources or build supply depots. Furthermore, they were also modified not to be able to build barracks but only supply depots. The FullyConv model was given a +1 reward for every supply depot it built in a five-minute time frame.

Figure 3.1 Screenshot of BuildSupplyDepots Minigame

In order to begin training on building barracks, a map similar to the BuildSupplyDepots map was created called BuildSupplyDepots. In this map, the workers were granted a new action that allowed them to build the barracks building. The FullyConv model was given a +1 reward for every barracks it built in a fifteen-minute time frame.

Figure 3.2 Screenshot of BuildBarracks Minigame

In order to begin training on building marines, a map similar to the BuildBarracks map was created called BuildMarines-easy. In this map, the barracks were granted a new action that allowed them to train marines. This is referred to as BuildMarines-easy as the action space is still limited compared to the BuildMarines map provided by the DeepMind group. Workers still cannot move outside of collecting minerals and building structures. Furthermore, Marines have had all actions removed. These actions were not considered necessary to complete the objective and thus only served to increase training time beyond feasible limits. The FullyConv model was given a +1 reward for every marine it built in a fifteen-minute time frame.

Figure 3.3 Screenshot of BuildMarines Minigame

3.1 Data Representation

The data used in this study was in the traditional CSV format and was queried from the PySC2 library provided by DeepMind [2]. The information was recorded as a time series, and each state was recorded in 62.5-millisecond intervals. The features we all quantitative and consisted of the following set: SCVs, SupplyDepots, Barracks, Marines.

The features SupplyDepots, Barracks, Marines, keep a simple tally of how many of each unit has been built by the agent of the course of the game. These values cannot decrease as there is no adversary working against the agent, and the agent has no available action that can remove a unit.

3.2 Data Transformation

In this transformation, we are concerned with representing the features of high-granularity time-series datasets in accordance with the Principle of Least Effort. Any transformation designed to reflect The Principle of Least Effort must reflect the mentation of Tools and Jobs. For this, I am proposing an approach that treats "feature moments" as tools for the ultimate tool of completing some objective in any domain. These feature moments simply describe the value of a feature at a given time step. I suggest this as the proper mentation as it is flexible enough for most domains, and it stands to reason that this mentation ought to adhere to the forces of Unification and Diversification as discussed earlier. In any sufficiently complex domain, there will be no feature moment in which that moment is the only one that needs visitation (or even can be the only one visited), hence Diversification. Likewise, there ought not to be an equal distribution of feature moments as not all feature moments will be equally useful.

As a result of this transformation, time is no longer an explicit feature of the data set. First, we define a matrix Wk of size m^k x n^k for each numeric feature F^k, where W^k_{ij} for I = (0,1,2...max(m^k)) and j = (0,1,2,...,max(n^k)) is defined as the number of observations of m^k_i in F at time n^k_j. We now assign an entropic value to each element E of W^k, as the following equation dictates. In each trajectory, the numeric feature values will be replaced with the corresponding entropy value of each feature-moment inside the transformation matrix.

$$entropy(E) = -log_{10}(\frac{E}{\sum_{i=0,j=0}^{max(m),max(n)} W_{i,j}}) \quad [3.1]$$

Calculation of Entropy for final trajectory transformation

3.2.1 Transformation Walkthrough

For the purposes of example, I will present a walkthrough of this transformation applied to the SupplyDepots feature in the BuildMarines(easy) minigame. In order to begin, there is a need for some predetermined variables to instantiate the transformation matrix. First, it must be determined what the upper limit of the SupplyDepot value range is. As the rules of the game do not necessitate an upper-limit, it can simply be assumed that the largest observed value is a

sufficient upper limit. If so, this will be our value M. Another value needs to be determined, N which is representative of the maximum number of observed timesteps. In the domain of BuildMarines, this value is statically set at 901 as each episode runs for a set 15 minute period. In other domains this value may be dynamic and can be set to the max value observed amongst the corpus of trajectories. Once these values have been determined, we can instantiate a matrix, as shown below.

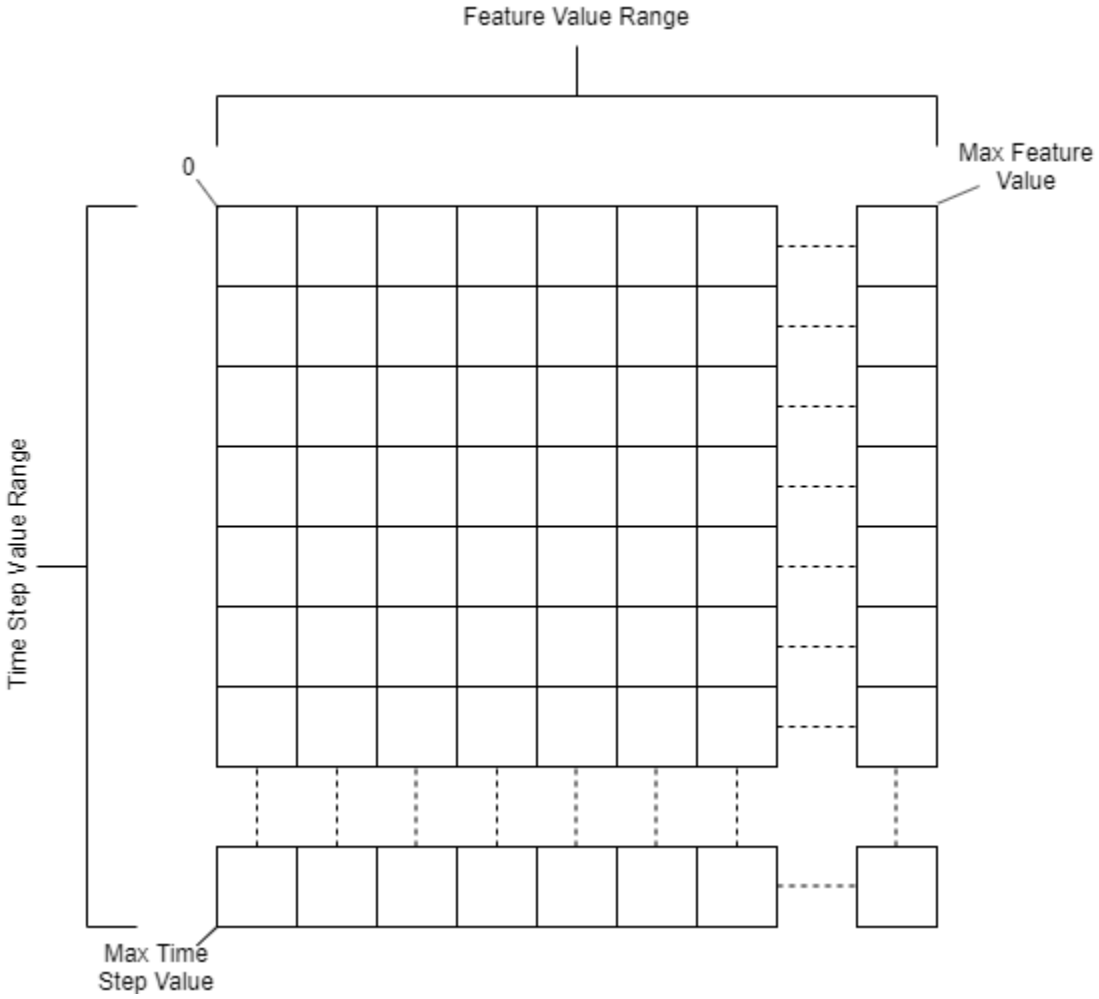

Figure 3.4 Visualization of Transformation Matrix

Once the matrix has been instantiated, it must then be filled with absolute frequency values from the entire corpus of trajectories. The process is relatively simple; each cell in the matrix can be described as a coordinate that keeps track of how frequently a specific feature value has

been observed at a specific time step. For example, if at time step 3 the feature value of 20 has been observed 57 times then 57 will be inserted into the cell at the location (20, 3) in the matrix. An example of this construction can be seen in the figure below with a corpus of 3 trajectories, only describing the SupplyDepot and TimeStep features.

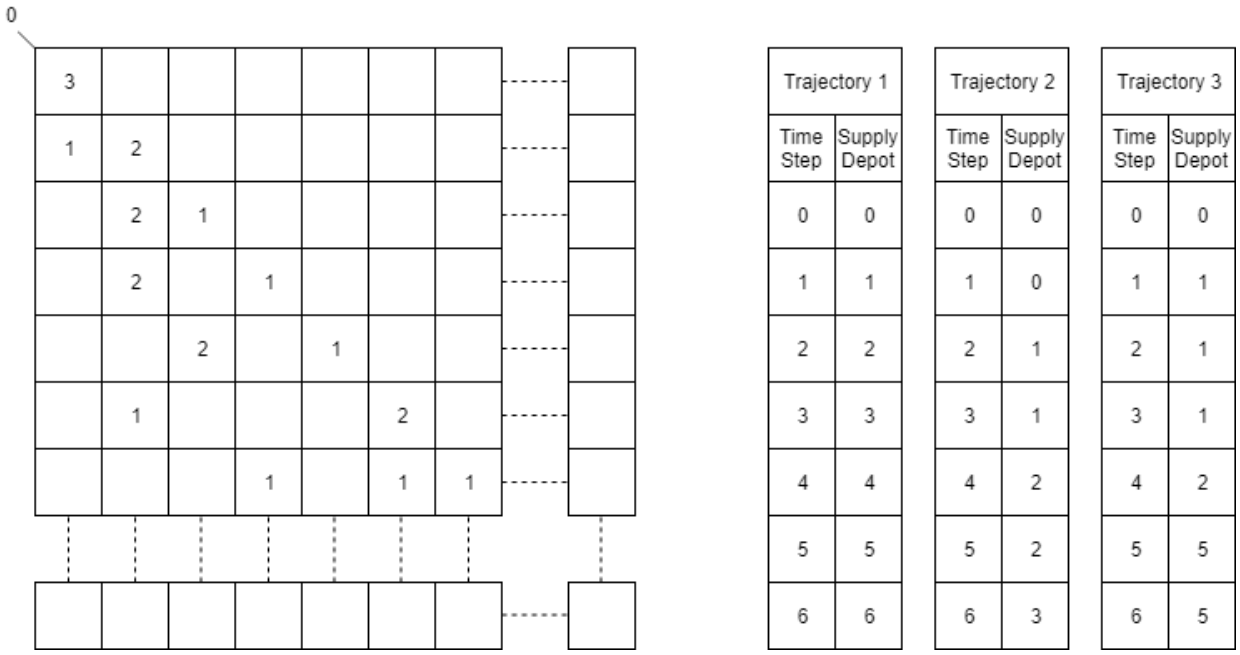

Figure 3.5 Filling in Values of Transformation Matrix

From here, a ranked frequency distribution of matrix cell values can be constructed. This distribution will ideally look very similar to that which has been observed in word distribution of various corpora of text. If this distribution has been found, then we can begin to assume that the Principle of Least Effort is at play.

Assuming we have confirmed that The Principle of Least Effort is at play, we calculate the probability of each nonzero cell occurring. It is important to note that the denominator of each probability is the count of all cells with nonzero observations and not simply the count of all cells within the matrix. From here, each cell of the matrix is given a negative log transformation, and with that, our data is transformed. Now, for each trajectory, we can simply replace each feature value with the corresponding value in the transformation matrix.

Chapter 4

Evaluation of the Transformation Matrix

It is the goal that the values of the transformation matrix will form a Zipf Distribution. As noted from statistical linguistics there are two relatively simple ways to test for a Zipfian Distribution. In the first case, we can note that the sum of the 20% of highest values should be approximately equal to 80% of the total sum of the entire matrix. There is also the predictive power of this distribution, in which the number of populated cells in the transformation can be approximated from the observed frequency values. First, we will begin by examining the distribution in figures 4.1 through 4.4 below.

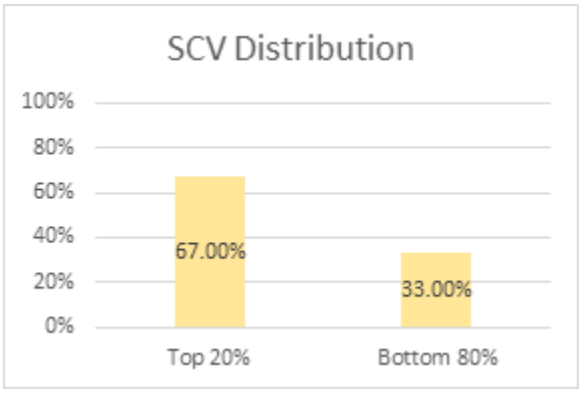

Figure 4.1 Distribution of SCV feature-moment frequencies

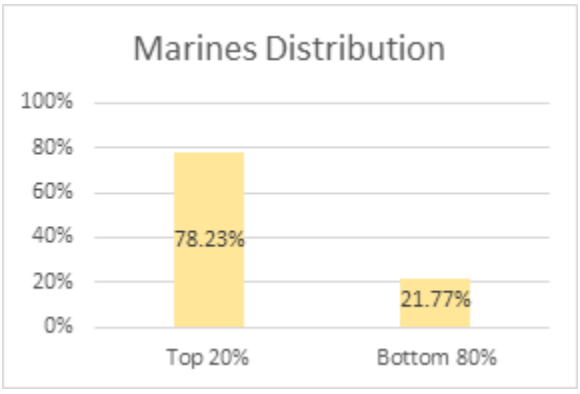

Figure 4.2 Distribution of Marine feature-moment frequencies

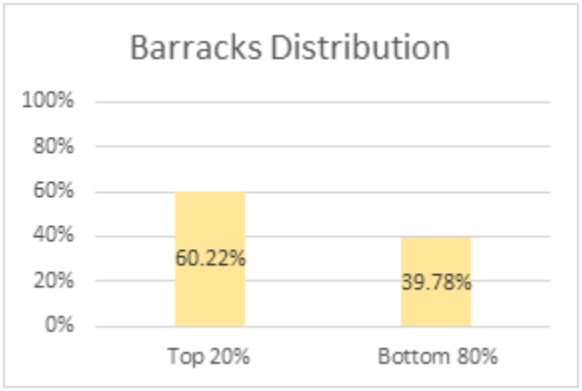

Figure 4.3 Distribution of Barracks feature-moment frequencies

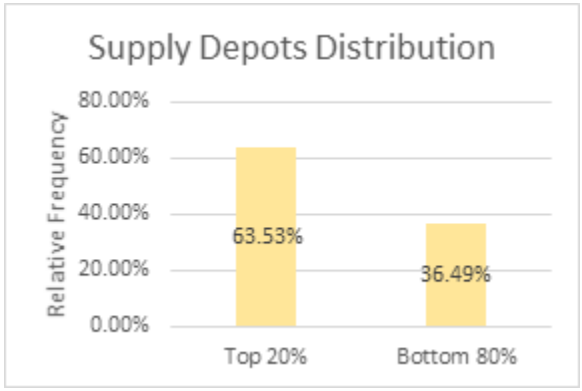

Figure 4.4 Distribution of Supply Depot feature-moment frequencies

First and foremost, it is quite clear that the expected distribution in which the top 20% most frequent feature-moments do not account for 80% of all feature-moments for most observed features (as observed by figures 4-1, 4-3, 4-4) as one would normally expect from theoretically perfect distribution. In fact, only the marines distribution seems to satisfy our expectations as observed by figure 4-2. It is of interest to note that while this distribution does not meet most feature-moments do seem to approximately hover around a distribution in which the top 20% most frequently observed feature-moments account for approximately 62.5% of feature-moments on average.

4.1 Distribution Discrepancy

It is here where the discrepancy between observed distribution and theoretical distribution must be discussed. In other fields, it is rather unlikely to observe such large differences such that the top 20% observable instances do not account for approximately 80% all observations. So, what is it about this application that suffers more from this gap between theory and practice? At the time of this writing, it would seem the best suggestion should be mentation formation. I have applied this transformation both to the previously described agent as well as an earlier saved version of this agent that had not yet finished training. The first difference

to note is outlined in the following histograms. These histograms represent the value of each feature in all games between each agent.

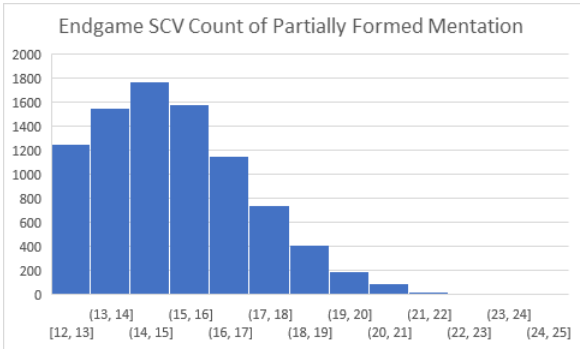

Figure 4.5 Distribution of SCV count at the end of each game for the partially trained agent

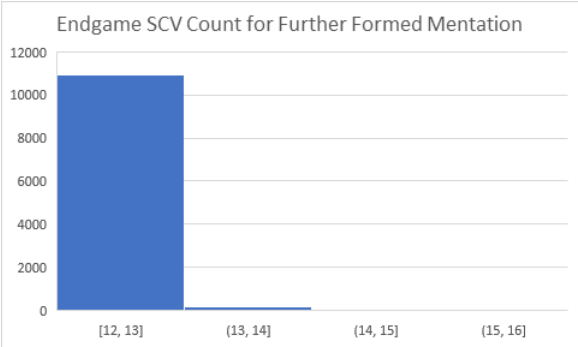

Figure 4.6 Distribution of SCV count at the end of each game for the more completely trained agent

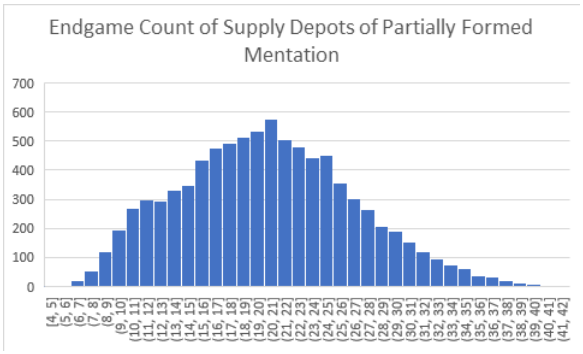

Figure 4.7 Distribution of supply depot counts at the end of each game for the partially trained agent

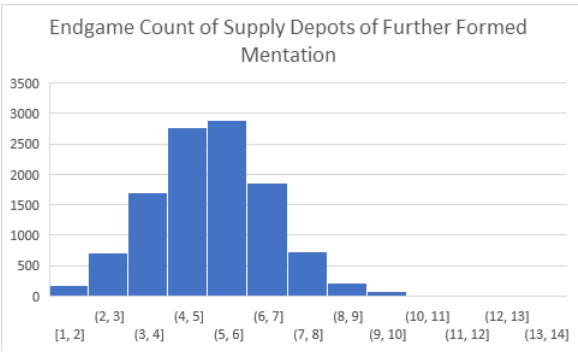

Figure 4.8 Distribution of supply depot counts at the end of each game for the more completely trained agent

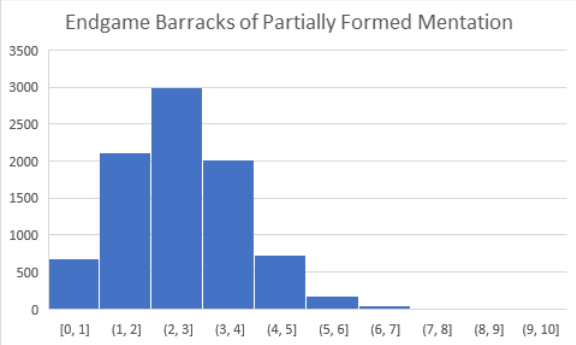

Figure 4.9 Distribution of supply depot counts at the end of each game for the partially trained agent

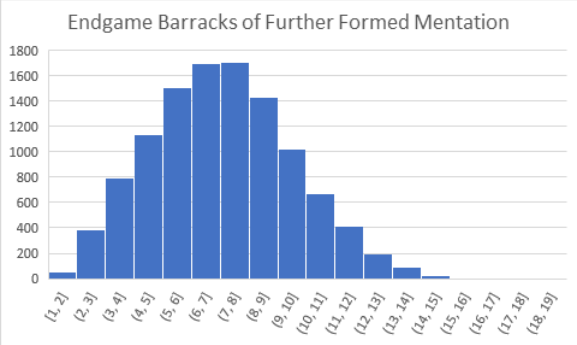

Figure 4.10 Distribution of supply depot counts at the end of each game for the more completely trained agent

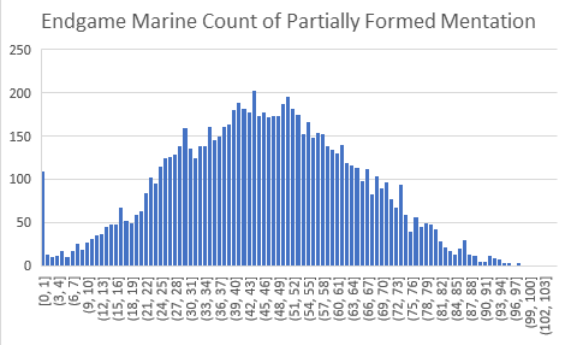

Figure 4.11 Distribution of supply depot counts at the end of each game for the partially trained agent

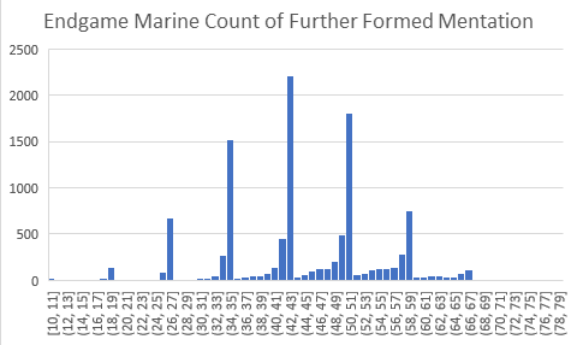

Figure 4.12 Distribution of supply depot counts at the end of each game for the more completely trained agent

	Supply Depot	**Barracks**	**SCV**	**Marines**
Partially Formed Mentation	6.34	1.226	2.007	18.652
Further Formed Mentation	1.532	2.486	.429	10.468
PercentChange	-75.836%	102.773%	-78.625%	-43.877%

Table 4.1 Change in Standard Deviation Between Endgame Feature Values of Each Agent

It can be seen that there seems to be a general trend that occurred as the agent gained more precise mentation of how to appropriately utilize feature-moments. In short, as the agent progressed, the amount of variation in feature values at the end of a game decreased. This

possibly suggests a more refined policy as the agent became more familiar with how to achieve its end goal. It is curious that the standard deviation of barracks has increased significantly rather than decreased like the others. It is impossible to ascertain and exact reason why without more research though it may suggest that the agent was exploring a policy change regarding barracks usage. This may be corroborated by the increase in mean barracks count which would allow an agent to create marines at a fast rate. Another point of note can be described in the following table, which outlines the distributions of each agent as well as the distance from theoretically perfect distribution.

	Supply Depot	**Barracks**	**SCVs**	**Marines**
Partially Formed Mentation Distribution of 20% most frequently observed feature moments	.57	.603	.573	.662
Partially Formed Mentation Distance from Theoretically Perfect Distribution	.23	.197	.227	.138
Further Formed Mentation Distribution of 20% most frequently observed feature moments	.635	.602	.67	.782
Further Formed Mentation Distance from Theoretically Perfect Distribution	.165	.198	.13	.018

Table 4.2 Difference Between Theoretical Distribution and Observed Distribution of Feature-Moments For Partially Formed Mentation and Further Formed Mentation

Now it stands to reason that as the endgame feature value standard deviation decreases, then the number of paths through the transformation matrix should decrease, and by extension

the number of visited feature-moments should decrease. If it is true that the standard deviation decreases as mentation forms, which seems intuitive, then this decrease should correlate with the decrease in distance from theoretical distribution of feature-moments. The following table demonstrates this with the change in standard deviation represented as a percentage in order to normalize the change. We can observe that with a correlation coefficient of .83, we can claim that there is a very strong correlation between these changes.

		Supply Depot	Barracks	SCV	Marines
Change in standard deviation	Partially Formed Mentation	6.34	1.226	2.007	18.652
	Further Formed Mentation	1.532	2.486	.429	10.468
	Percent Change	**-75.84%**	**102.773%**	**-78.625%**	**-43.877%**
Change in Distance from theoreticaly perfect distribution	**Distribution Distance Change**	**-1.066**	**-.999**	**-1.097**	**-1.1202**
	Further Formed Mentation	.1647	.1978	.13	.0177
	Partially Formed Mentation	0.23045012	0.196739	0.226811	0.137897
Correlation Coefficient: .83					

Table 4.3 Correlation Between Change in Standard Deviation during Training and Change in distance from Theoretical Distribution of Feature-Moments

Chapter 5

Conclusion and Related Work

With this work we can see that the transformation matrix works well with agents with sufficient mentation due to a strong correlation coefficient of .88. With this correlation coefficient we can begin to suggest that this transformation is an appropriate way to capture the Principle of Least Effort in decision-making as seen through trajectories with quantitative features. With increased mastery of an objective, it may be that the gap between observed distribution and theoretical distribution closes; this is even reminiscent of Tsai's experiments with rats in which a preference was still shown but did not fit Zipf distribution. This is in line with the intended use case of this transformation in the field of inverse reinforcement learning in which a masterful agent is usually the agent of choice of which observations are made. This may also have additional usages in indicating levels of mastery in human agents where that may otherwise be difficult to define.

Future work will primarily begin in the application of this transformation. Namely, this work will begin with the three problems originally outlined in the introduction section. The problems are state reduction space reduction, reduction of value ranges in quantitative features, and a similarity measure between trajectories.

5.1 State Space Reduction

In the interest of reducing the state space it may be that a simple partition-based clustering approach is viable (e.g. k-means). In this approach the transformed data can be clustered on to identify states based on similarity. This approach has a strength to it over Liao's method in

that it is more built upon a ground truth of familiarity. Furthermore, it would only require an estimation of the appropriate number of k clusters once rather than twice which would increase the validity likelihood of this clustering approach in this method.

With this clustering approach one may be able to represent a state as being dynamic in length on only ending upon the execution of some action or the entering of a new cluster. This would effectively reduce the state space to that of just time and cluster class. This reduction in feature space may not be ideal for applications that require the feature space be maintained and thus, the next approach may be considered.

5.2 Reduction of Value Range in Quantitative Features

This approach works relatively similar to the previous approach, but rather it is applied to each feature independently of one another. In this approach Jenks natural breaks optimization could be utilized on each transformed feature amongst the corpus or trajectories. Jenks natural breaks optimization functions more or less identically to k-means but it's optimized for univariate quantitative data. In short, it partitions univariate quantitative data into groupings based on distribution such that similar values that frequently occur are placed in the same partition.

This approach should provide a significantly reduced feature value range such that values are grouped based on feature-moment familiarity. This has the added benefit of reducing the state space as it could provide dynamic sized lengths of states just as it would for the k-means approach. This would not be quite as effective as k-means as there would still be a sizeable feature space but with the range of quantitative values reduced.

5.3 The Similarity of Trajectory Familiarity

In the event that a similarity function is needed between trajectories, this method could also be used. It starts with a similarity between states in which the average entropy of each feature is measured for each state. As trajectories naturally diverge further and further as they travel from their initial state it is expected that entropy would increase as a trajectory goes on.

In trajectories in which feature-moments are regularly visited it is intuitive to say that overall the trajectory has a rather low amount of entropy increase. This is opposed to unfamiliar trajectories that would have a very high entropy increase. With this in mind I propose that a similarity between trajectories can be defined as the average difference rate of increase in entropy per each trajectory.

LIST OF REFERENCES

[1] P. Berkhin. *A Survey of Clustering Data Mining Techniques*, pages 25–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[2] Deepmind. deepmind/pysc2, Sep 2019.

[3] G. F. Jenks. The data model concept in statistical mapping. *International Yearbook of Cartography*, 7:186–190, 1967.

[4] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297, Berkeley, Calif., 1967. University of California Press.

[5] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[6] Stuart Russell. Learning agents for uncertain environments (extended abstract). In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 101–103, New York, NY, USA, 1998. ACM.

[7] L. S. Tsai. *The Laws of Minimum Effort and Maximum Satisfaction in Animal Behavior*. Monographs of The National Research Institute of Psychology, 1932.

[8] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy P. Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. Starcraft II: A new challenge for reinforcement learning. *CoRR*, abs/1708.04782, 2017.

[9] T. Warren Liao. A clustering procedure for exploratory mining of vector time series. *Pattern Recogn.*, 40(9):2550–2562, September 2007.

[10] R. H. Waters. The principle of least effort in learning. *The Journal of General Psychology*, 16(1):3–20, 1937.

[11] G.K. Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Martino Fine Books, 2012.