CS 6601 Artificial Intelligence: Learning Portfolio

1 About Me

Hello. My name is Arvind Krishnaa Jagannathan. I am a MS CS student at Georgia Tech hoping to specialize in Interactive Intelligence.

I like to mess around with code and am passionate about Computer Science in general. This is my Learning Portfolio for the graduate class on Artificial Intelligence, taught by Dr. Thad Starner. This portfolio is intended to provide some useful insight to **ANYONE** who is interested in taking up this course in the future under Dr. Starner. I do not have a web page, but have an online presence at http://www.prism.gatech.edu/~ajagannathan3/. An external view of this document can also be found there.

2 FOUNDATIONS: READING AND ASSIGNMENTS

In the course, we completed 8 assignments on the foundations of AI, after reading the relevant material in the textbook.

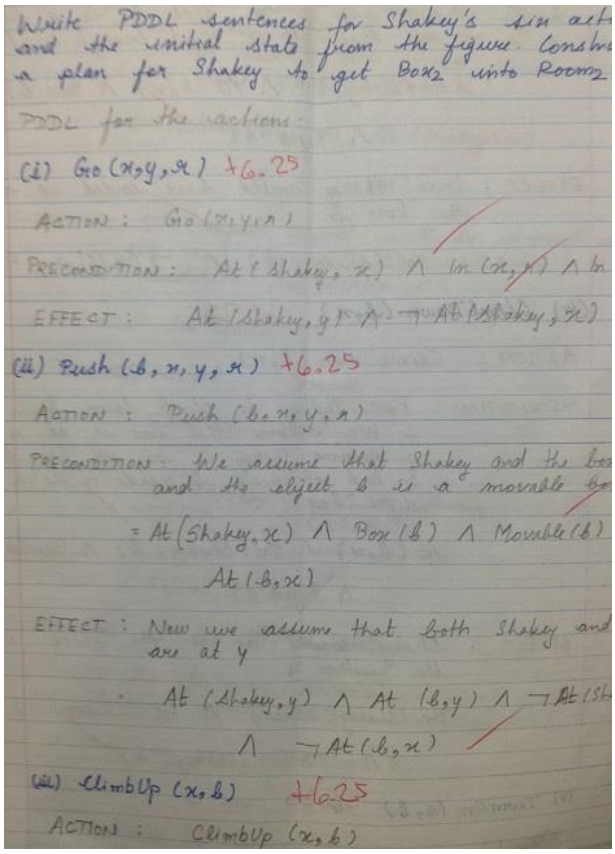
Although I had an attended an introductory course on Artificial Intelligence during my undergrad, this class presented a whole new refreshing perspective for me. We covered several aspects of AI, as reflected by the assignments. The 8 assignments covered several important topics, namely (in order), (1) Search (2) Game playing and constraint satisfaction (3) Logic (4) Planning and Bayes Nets (5) Hidden Markov Models (6) Learning (7) Reinforcement and (8) Natural Language Processing.

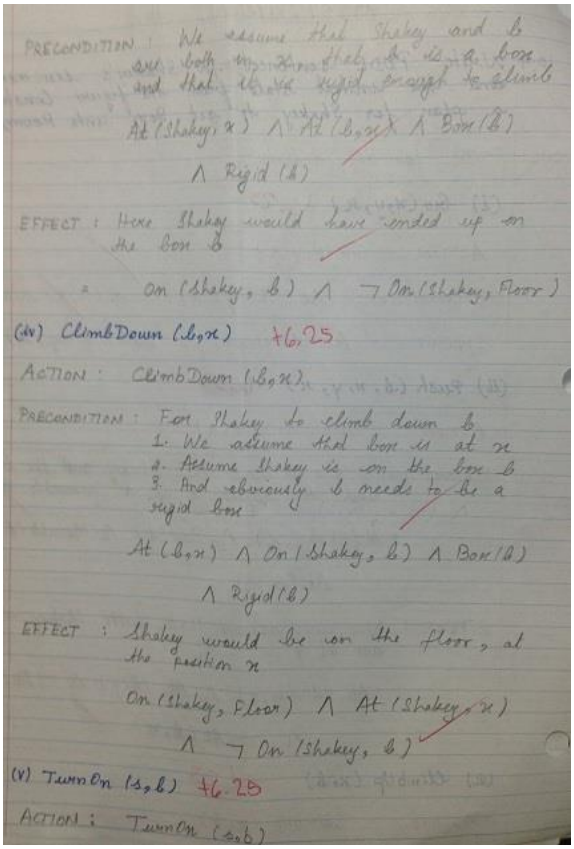
The most interesting aspect of the class was the way in which these assignments were structured in relation to the reading material and the class lectures. The readings required for the assignments were from (probably the best book on AI) the book, "Artificial Intelligence: A Modern Approach. 3rd Edition" by Stuart Russel and Peter Norvig. The readings and the assignments related to them had to be turned in before the lecture on that topic. The class lectures covered more advanced topics involving the concepts of that week's readings. This approach gave me a solid foundation about the basics of AI as well as an interest to participate in class. I learnt that the key challenge in AI is to take up intractable problems, ones which are NP-hard and obtain a good enough solution for them. I realized that almost all the problems encountered in class (programming or otherwise) had an exponential time complexity. A significant takeaway from the assignments was the "No free lunch theorem". It states that, no algorithm can be said to perform suboptimally, as there exists some dataset for which the algorithm under question will turn out to be the best. I found this out the hard way in the final NLP assignment where we had to generate text from a corpus using the unigram, bigram and trigram frequencies.

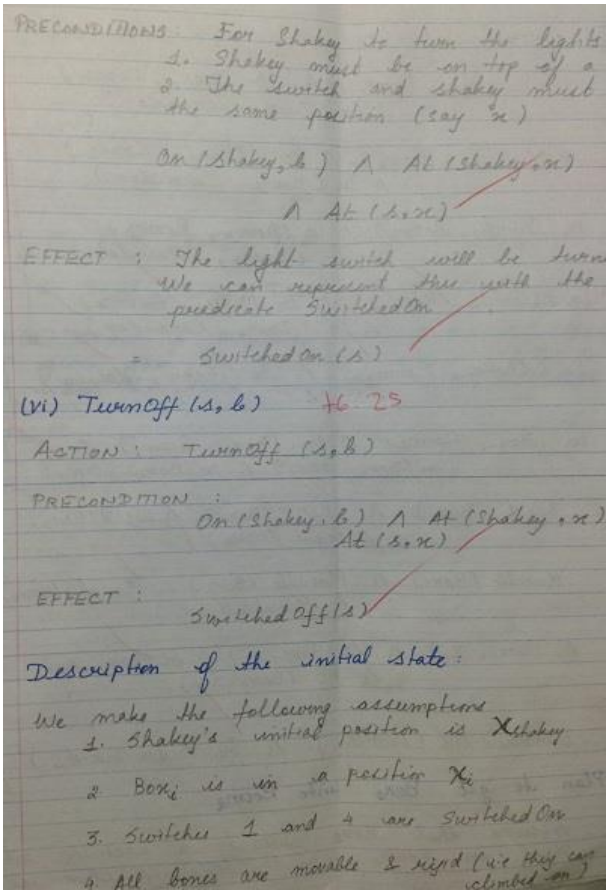
The later set of assignments were very time consuming, especially the ones on Hidden Markov Models and Reinforcement. They required some light programming, in order to solve problems such as that of the value iteration algorithm (which would probably take days for me to complete manually). However my favorite assignment was the one on planning and Bayes nets. I liked doing that assignment because of the nature of the first question. This was a question which required us to codify various moves that a monkey, "Shakey", could perform, such as moving from one spot to another, pushing a box and so on. Finally, we were given two situations, and we were asked to describe each of these situations and write the set of rules which would get the monkey from the initial state to the final state. All descriptions were to be codified in PDDL (Planning Domain Definition Language), which to me seems a very elegant way of representing information, in a form modeled almost exactly after First-Order Logic. This assignment really helped me in my critical thinking, and was probably the one which was closest to "classical AI" (which is my area of interest) among all the assignments. I learnt that describing scenarios in terms of logic, although not very natural for humans to understand, is a powerful technique that can be applied to computers. I understood that utmost care is required while coming up with representations for each action; even a minor mistake can end up "confusing" the rule engine, thus making the AI program loop infinitely without achieving the target. However, I also realized that if coded correctly, PDDL is extremely powerful. Using the approach of Forward Chaining, every inference which can possibly be drawn out of

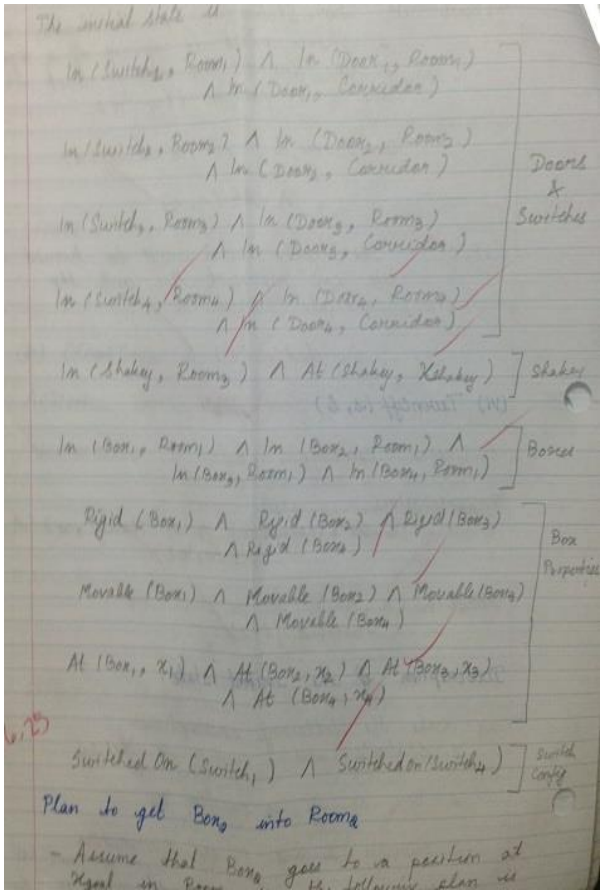
the knowledge base, can be extracted correctly. Using Backward chaining, one can quickly prove if a statement is true or false using the set of rules available.

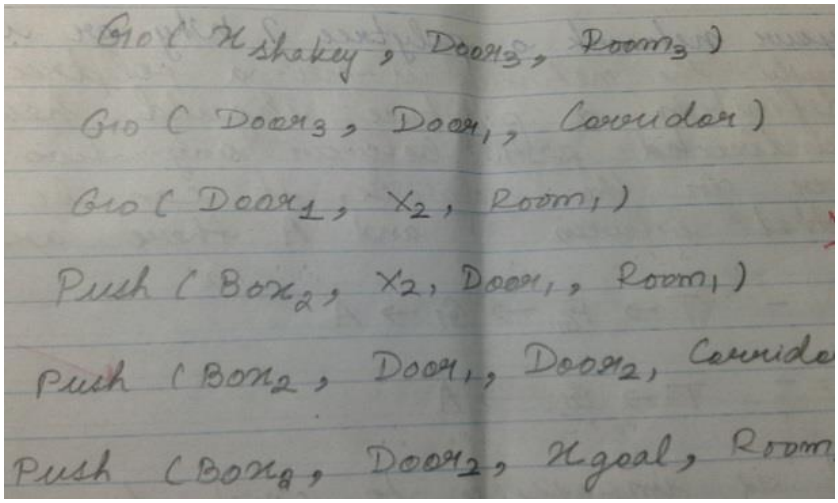
On the whole, I felt each assignment helped me gain a little more insight into the concepts involved. I performed fairly well on the assignments, but the most challenging assignments were the ones on constraint satisfaction and hidden markov models. The latter was really challenging because it involved several calculations. Also, because of the nature of the Viterbi Trellis algorithm, even if one of the calculations are wrong, it could create a cascading effect, making all the following steps incorrect. However it was fun to do the problem because it is possible to intuitively guess what the correct answer would be once the nature of the problem is well understood. I found the assignment on constraint satisfaction to be more challenging, since it required us to solve a cryptarithmetic problem and find all possible unique solutions. What was particularly hard about this problem was the manual enumeration of all possible solution sets; missing even one can lead to rapid convergence to an incorrect solution. The assignments I did the best on, as reflected by my score in those assignments, were the ones on NLP and Learning. The Learning assignment was very straightforward, and the concept of decision trees seemed very useful to me. Although deceptively simple, I came to know that decision trees are one of the best methods to determine the most "relevant" features in a problem with a large number of dimensions. The NLP assignment was the most "interesting" of the lot, as it involved quite a bit of programming. However a major chunk of this assignment was experimental, such as generating text from a large corpus and ranking the results returned by different search engines based on a set of queries. This assignment, I feel, was well placed on the schedule, and it has sparked my interest in pursuing research towards natural language and information retrieval.











Description of the "Shakey" problem in PDDL. Rules for various actions and finally the plan to transition from the initial state to the final state.

3 SKILLS: MINI PROJECTS

There were two mini-projects in which I chose to research a problem that was supposed to be relevant to my future career. For each of these two projects, I proposed a solution, implemented it, and described it in a mini-conference paper.

The two mini-projects I chose to do with partners covered two significant themes of the class. The first project was to develop an AI for an ancient Indian board game, called Chowka Bharra. My second project was to segment a unistroke mouse gesture sequence, drawn by a user, and identify each of the individual gestures in them. Obviously these projects gave me a hands-on experience in implementing the concepts that we learnt in class and map them to real-world applications.

Both these projects taught me a lot. In the first project, a key challenge I faced was that there was very little literature available about the game itself. The main goal of my partner and I, were to utilize strategies, such as alpha-beta pruning and expectiminimax, which have been used successfully with games such as Backgammon, to Chowka Bharra. However, a stumbling block to our progress was that besides the fact that Chowka Bharra was a partially observable and partially stochastic game (like Backgammon), there were no other games which had playing styles that could be applicable to this game. Hence we had to come up with our own optimal strategies of playing the game. An important lesson I learnt from designing an agent for this game was that, in general, any game playing algorithm relies heavily on the heuristic functions chosen to evaluate the state of the game. Depending on the heuristics chosen and the weights assigned to them, the performance of the AI player can be significantly altered. Also, a more obvious observation was that an agent which was able to look several levels down into the game performs better than one which looks only at the current state of the board. After several attempts of trial-and-error, we were able to develop an agent which had a success rate of over 90% against a random agent. Through this project, I also learnt how much of a performance improvement can be obtained if alpha-beta pruning can be applied to the game tree.

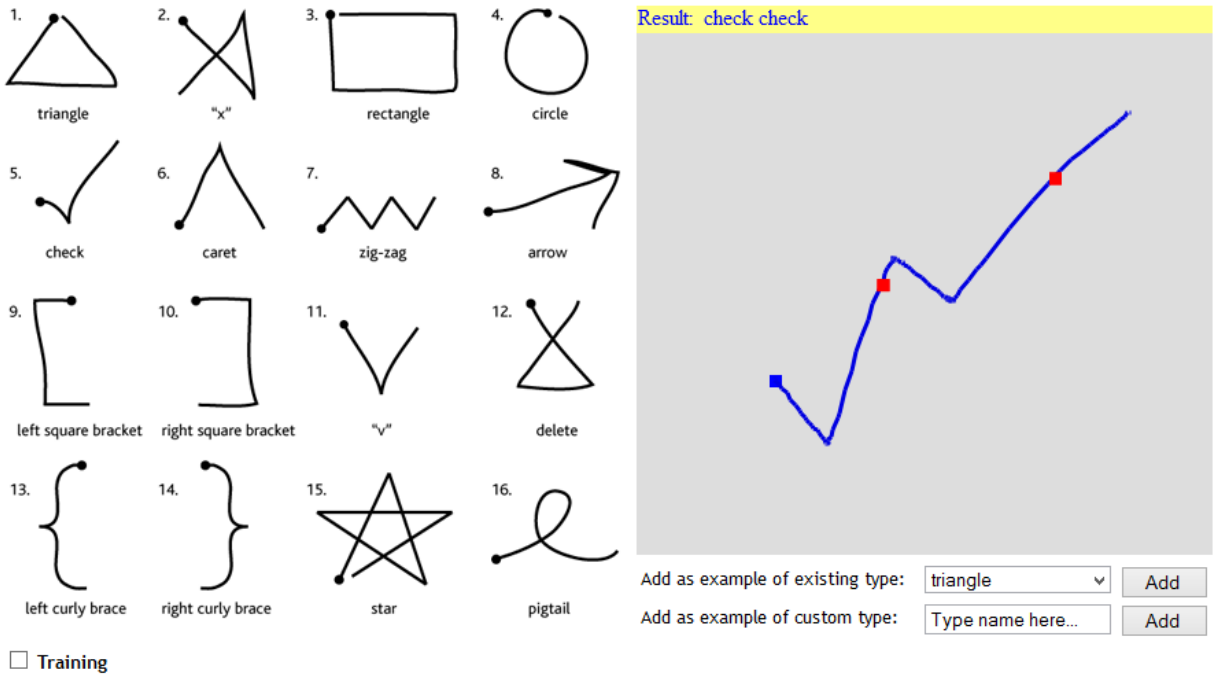
The second project posed a challenge of another kind. My partner and I decided to tackle the problem of gesture recognition in multi-gesture sequences through two approaches: first by applying dynamic time warping (DTW) and reducing the problem to that of a geometric match. The second approach was to train HMMs for individual gestures and then chain them up to form HMMs for every possible gesture sequence that a user could draw. The reason behind our approaches lies in our objective. We were clear that we did not want to "train" the system extensively. Our goal was to develop a system which can be minimally trained, and that too only on individual gestures, and based on this data, we planned to segment and recognize gesture sequences. Clearly, our objective made the problem much more complex and constrained the data which we could have used. We learnt that since we allowed users to give as input any arbitrary gesture, we were unable to leverage any sort of grammar from the input. Moreover, by not training on multiple gestures, we lost out important transition data, such as a pause, or a distinct change in direction of the mouse stroke, which often characterizes the "switch" from one gesture to another. Without this information, we had to algorithmically infer the segmentation points in the input gesture. Also, we were unable to represent the input data in a format suitable for the HMM toolkit which we experimented with (HTK). Since the process of constructing HMMs was too time consuming, and did not fit within the timeframe of the project, we decided to apply the DTW approach and we got fairly good results. I feel that although both the projects were equally fun, the second was

more challenging as well as the one which I felt I did better; it was challenging simply because of the constraints we imposed on ourselves, but we were still able to achieve reasonable results.

4 My Favorite Project: Recognition Of Unistroke Gesture Sequences

4.1 Problem Addressed

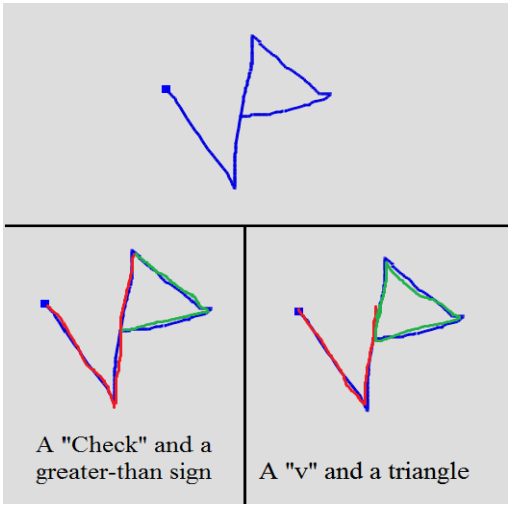
Gesture recognition is a sizable research area, due to the many uses and applications of gesture detection. Our problem is recognition of a unistroke gesture sequence, defined as a continuous stroke (unistroke) consisting of multiple gestures defined a priori (see the Figure below). This falls under a category of problems classified as "Inverse Perception Problem" and is notable "hard" problem in literature [1].



User Interface of the gesture recognizer. New gestures can be added to the system as well in addition to the 16 pre-defined ones.

The question we are asking is: given an input unistroke gesture sequence, which individual gestures comprise the input? A fundamental challenge with having no restriction on the input is the case of

ambiguity. There can be more than one combination of gestures which correspond to a given gesture sequence (refer Figure below)



Ambiguity in the recognition of similar gestures

4.2 Related Work

Yang et al [2] present work on recognition of individual gestures in unistroke and multistroke gesture sequences of digits by constructing an exhaustive set of HMMs. They train HMMs to recognize continuous gesture sequences defined a priori, which is a deficient approach for our problem. Robust individual gesture recognition systems exist, such as the mouse gesture recognition system developed by Tanguay [3]. However, the implementation is tailored to individual gesture recognition. Additionally, the $1 recognizer [4] is a single gesture recognition system which works with no training (the system has a built-in representative gesture training set), but the primary deficiency is its inability to recognize gesture sequences.

4.3 Approach

We created a prototype gesture recognizer with two modes of operation: (1) a training mode, in which the user selects a gesture and subsequently trains by providing additional gesture templates, and (2) a recognition mode, in which the user draws a unistroke gesture sequence for subsequent recognition. For evaluation purposes, we restrict the maximum number of gestures in the sequence to 3, but the system can accept any finite length input.

Our approach, at a high level, is to compare the visual similarity of an input with pre-existing templates: how similar is each perceived gesture comprising the input to a known gesture template? The first step is to match each template against a portion of the input in order to identify a "segmentation point" that splits the input into a matched gesture and the remainder. The second step is to remove the matched gesture from the input sequence and iterate, repeatedly searching for additional gestures in the remaining user input. A challenge to designing our approach was to account for variability in user input.

Segmentation

During recognition, the user draws a unistroke gesture sequence, which is recorded as an ordered list of (x,y) coordinates. The user's input, as noted above, may contain one or more individual gestures. However, the template, by definition, is one gesture. As a result, we need to compare a portion of the user's input with the template. The question is: what portion? An insight into our problem that helps answer this question is the observation that the same user provided both the template and the input, and that both are drawn to a similar scale. As a result, the user's input, if it contains this template, must have a path length approximately equal to or greater than the path length x of the template. Conversely, if the user's input has a path length less than x, then the input cannot contain the template. We repeatedly perform this calculation for each template; the templates that are a potential match are subsequently scored. We score the difference between a portion of the user's input and the given template using Dynamic Time Warping (DTW, see next section for details). DTW essentially performs a flexible comparison between two inputs that have been sampled over time. In order to generate input suitable for DTW, we re-sample both the user's input and the template to 64 equidistant points. Although the points are equidistant (due to the matching considerations discussed above), the points are also ordered with respect to time, which allows us to provide them as input to the DTW calculation. Additionally, there are two benefits to re-sampling: 1) re-sampling smooths out noise, and 2) re-sampling with 64 points is enough to retain the resolution necessary to disambiguate gestures but also low enough to be computationally inexpensive. Additionally, this number was successfully used by the $1 Recognizer. Our routine contains one other noteworthy optimization aimed at determining the most accurate segmentation point. After the routine determines an initial candidate segmentation point, as described above, it is possible that a nearby point is actually a better match. We look for a better segmentation point (indicated by a smaller DTW score) by searching backward and forward from the initial candidate point, repeatedly calculating the score. The segmentation point corresponding to the minimal score is marked as the point of segmentation, and the matched gesture is spliced from the input sequence.

4.4 EVALUATION

The unique component of our system is our segmentation routine, which is tightly coupled to (and dependent upon) the accuracy of the individual gesture recognition algorithm (DTW). As a result, we evaluated our recognition system via three metrics, which are, as whole, intended to provide insight into the system function.

1. Overall Accuracy: This is an all-or-nothing measure. For input sequence of lengths 1, 2 or 3, we measure the percentage of trials in which the system outputs both the correct number and correct identity of individual gestures (a perfect match).
2. Segmentation Accuracy: For input sequence of lengths 1, 2 or 3, we measure the percentage of trials in which the system outputs the correct number of gestures (regardless of correct identification of individual gestures).
3. Relaxed Accuracy: This is a relaxed version of Overall Accuracy relevant only to gesture sequences of length 2 or 3, where we measure the number of gestures in the output that actually appear in the input, accounting for order. The purpose of this metric is to determine the percentage of trials where a portion of the input was correctly segmented and identified. For this metric, we count each gesture that was correctly identified, accounting for gesture order.

We obtained the above results by performing a micro-study of 3 unrelated users. In addition we also measured how the accuracy of the system varied with the amount of training put into it by each user. The results are tabulated below.

Table 1: *Overall Accuracy*

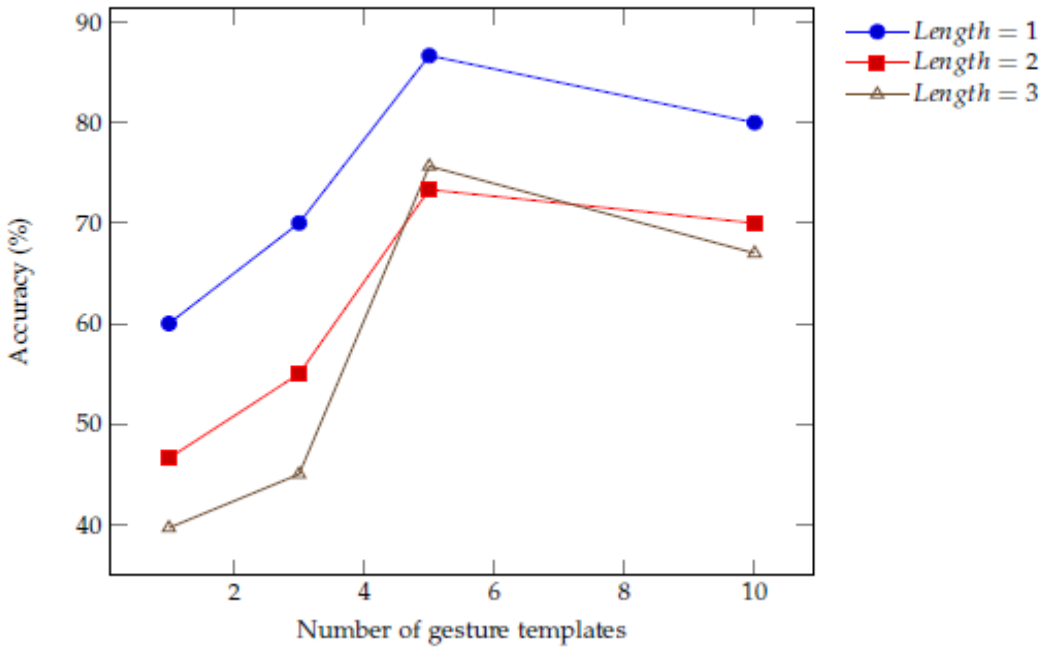
Sequence Length	Accuracy Rate
1	57.5%
2	32.16%
3	25.72%

Table 2: *Segmentation Accuracy*

Sequence Length	Segmentation Accuracy
1	71.5%
2	71.5%
3	52.27%

Table 3: *Relaxed Accuracy*

Sequence Length	Accuracy Rate
2	56.19%
3	56.37%



Recognition Rate vs. # of Gesture Templates

4.5 DISCUSSION

Tables 1, 2 and 3 reveal that our approach yields reasonable, but far from perfect, results. The accuracy rate for sequence of length 1 (57.5%) implies DTW is a useful approach for comparing two data sets that vary in time. This fact becomes clearer when we compare the DTW approach to the HMM approach taken by Tanguay, where he achieved a recognition accuracy of 50 - 60% on lower-case English alphabet letters after extensive training. Note that the accuracy is lower for our problem than for the simplified problem of 1-to-1 gesture template matching due to the complexity introduced by allowing a gesture sequence. In the training mode, where only sequences of length 1 are allowed, DTW excels at labeling the input correctly.

Table 1 shows a drop-off in accuracy as the sequence length increases. This drop-off indicates a fundamental shortcoming in our approach: improper recognition of a gesture is concomitant with improper segmentation of the gesture sequence, leading to mis-recognition of subsequent gestures. Yang et al report a nearly perfect recognition rate of up to 99.78% for sequences of digits up to length 2. The difference in approach is a key consideration when comparing these numbers. Yang et al train Hidden Markov models for both individual gestures as well as gesture sequences. It is probable that training on gesture sequences will yield significantly improved recognition rates. Table 2 shows the percentage of trials that were correctly segmented into the correct number of gestures. Note that, for each length, the trials in which complete recognition was achieved (table 1) are a subset of the trials in which the correct number of gestures was identified (table 2). As a result, one can conclude that the difference in Overall Accuracy as compared to Segmentation Accuracy is due to incorrect recognition of individual gestures. A more robust technique for individual gesture recognition may significantly boost the Overall Accuracy.

The Relaxed Accuracy results in Table 3 can be compared with the corresponding Overall Accuracy results in Table 1. The increase in recognition accuracy indicates that the system "recovers" after a gesture is incorrectly identified. This is noteworthy primarily because it implies that additionally training intended specifically to address gesture ambiguity (including ambiguity with the 11 gestures the user did not choose) may significantly increase Overall Accuracy. Figure 3 clearly indicates that additional training increases both Overall Accuracy and Relaxed Accuracy metrics. Importantly, "excessive" training (seen for 10 templates) may lead to incorrect recognition due to the "confusion" caused by having too many variations of an individual gesture; put another way, too many templates adds ambiguity, which supports our objective of training the system minimally.

4.6 References

[1] Zygmunt Pizlo. Perception viewed as an inverse problem. Vision Research, 41(24):3145–3161, November 2001.

[2] J. Yang, Y. Xu, and C. S. Chen. Gesture interface: Modeling and learning. In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on, pages 1747–1752, 1994.

[3] D. O. Tanguay Jr. Hidden markov models for gesture recognition. Master's thesis, Massachusetts Institute of Technology, 1995.

[4] J.O. Wobbrock, A.D. Wilson, and Y. Li. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In Proceedings of the 20th annual ACM symposium on User interface software and technology, pages 159–168. ACM, 2007.

4.7 Introspection

I am extremely proud of the work of both my projects. However, I am inclined towards the second project simply because I learnt more from it. I learnt that Dynamic Time Warping is a very quick and flexible method of comparing two signals (in my case the user input), when both vary over time. Hidden Markov Models are an appropriate tool to check for a "best" match between two inputs, when they vary by scale or over time. In other words, HMMs are a more robust method of discovering a known pattern

amongst previously unseen and potentially noisy input (which was the case with our user study). I only wish that we had more time, so that I could re-visit project 2 and try to implement our HMM idea.

5 SELF-ASSESSMENT

In addition to actually carrying out the project and drafting a mini-paper, the process of writing up a proposal for the two projects itself was a valuable learning experience for me. We had to draft both the proposal and the mini-paper, keeping in mind the rubric that was put up on the website. Our proposal and papers were graded by a peer review panel comprising of our classmates. This was something very new to me, and as the professor put it, gave me real-world experience at writing such proposals. Although I felt the review panel were overly critical at times, this was to be expected because a research grant review team or a conference publication selection panel would also be strict while going through a proposal/paper. I learnt that it is not only important to have an interesting problem, but is also necessary to elucidate it (in the proposal/paper) in a clear and concise manner. It is necessary to make sure our approach is easily understood and reproducible, otherwise it would not be rated well by the review panel; this was something I found out from my second project.

I felt that overall I managed to perform fairly well in this class and managed to take away a lot from the course. I think that both my projects have the scope and potential to be extended into full-fledged research. For instance, based on way in which we derived the heuristics for the game playing agent, I feel that I could work on a general game playing AI algorithm; although there are extremely specialized algorithms for particular games, a general algorithm is still a work in progress. I felt that we could have made the game more visual by actually incorporating a user-playable interface. Time permitting, I would have loved to find out how our AI player would have performed against a human opponent. With regards to our second project, we could have had more users participate in our study to ensure that our results did not show author bias. Moreover, as the professor suggested, if we had included a user orientation phase, where we specifically "train" the users to draw gestures accurately, I believe we could have obtained much more accurate results. This is what usually consumer-oriented products such as the Wii controller or the Kinect camera; they expect the users to perform certain pre-choreographed moves multiple times, before the sensors actually begin motion tracking. Time was the only constraint for both the projects, and I wish we were able to manage it better during the course of the project.

6 FINAL COMMENTS

This class was very unconventional to me. There were no mid-terms for which we had to cram whatever was taught in class, and neither is there any finals. However, this class was a great experience, both as a student as well as a potential AI researcher in the future. Our assignments and project were aligned in a manner to help our understanding of the subject. Even for a person like me, who has attended some introductory coursework on AI, this class has given me a sense of purpose in pursuing AI in my career. Some of the work that the professor presented during class lectures, were on the cutting edge of AI research at Georgia Tech, and it has given me a general idea about the direction in which AI is headed. Artificial Intelligence is a vast subject, with many of its sub-specialties having dedicated courses listed at Tech. Therefore, it is highly unlikely for every aspect of the subject to be covered in a class spanning just one semester. However, the class progressed at a fairly rapid pace and the professor did a great job of

introducing us to several tricks and techniques which we will find really useful during our future endeavors.

The prior readings and assignments were a good way to get a foothold of the discussions during class lectures, and the projects were a great tool in implementing the techniques we learnt and assessing our understanding of those concepts. The projects also helped me to communicate better with a partner; the proposal and paper drafts definitely improved my writing skills targeted at a scientific audience. Reflecting back on the course, I strongly believe that anyone who is interested in AI in general can take up this course without any reservation. Programming skills are preferred, but not really a requirement - what is more important is one's understanding of concepts and the desire to learn more. The class will definitely be more relatable if one has taken an introductory course on AI or machine learning, but the book is so well written that I think it is completely possible to learn the concepts on one's own. In addition, the class lectures ensure that the fundamental concepts are touched upon prior to advanced topics, and so no student should feel "lost" at any point of time during the course.