Leveraging Q-Learning for Uncertainty within Medical Diagnosis

Stanford CS238 Project

David Maemoto
Department of Computer Science
Stanford University
davidmaemoto@stanford.edu

Daniel Yang
Department of Computer Science
Stanford University
dy92634@stanford.edu

Ethan Farah
Department of Biomechanical Engineering
Stanford University
efarah@stanford.edu

Abstract

This paper explores the use of Q-learning, a reinforcement learning algorithm, for providing discrete decision-making support in medical patient diagnosis. By modeling the diagnostic process as a sequential decision-making problem of asking questions to a patient, we aim to assist healthcare professionals and patients with an intelligent second opinion. We construct our knowledge base from numerous Kaggle datasets, textbooks, clinical guidelines, and PubMed articles, and standardizing medical terminology using SNOMED CT. Through this aggregation, our dataset is more comprehensive and expansive than previous studies. The input to our algorithm is a sequence of yes or no answers to various diagnostic questions and patient prompts. Thus, the number of potential states for our model grows exponentially with the number of symptoms. We propose new algorithmic features and enhancements to the base Q-learning algorithm to handle large complex decision trees and efficiently process exponential states. Our model outputs the most likely predicted diagnosis along with the associated probability and confidence. Initial testing demonstrates the potential of our approach in improving diagnostic accuracy (and ultimately patient comfort and satisfaction) while balancing exploration and exploitation in high-stakes environments. Our modifications and enhancements can be easily generalized to more complex tasks and decision-making scenarios. This project provides insights into the practical application of reinforcement learning in healthcare and introduces several novel modifications to existing Q learning algorithms.

1 Introduction

Diagnosing a variety of neurological and physical conditions is a complex – almost inhuman – process that requires expertise across a plethora of fields and often involves iterative questioning and reasoning. However, diagnostic errors remain a significant challenge, leading to suboptimal at best and deadly at worst patient results as well as increased healthcare costs both on the hospital and patient sides. Our motivation for pursuing this problem stems from the need to provide accurate, unbiased, and efficient diagnostic support for patient conditions, potentially improving patient care and reducing diagnostic errors.

The input to our algorithm is a sequence of responses to diagnostic questions prompted to a patient/user. These inputs include categorical answers such as symptom presence (e.g., "Yes" or "No"). We utilize a reinforcement learning Q-learning algorithm, where each diagnostic step is modeled as a

discrete state in a Markov Decision Process (MDP). The output of our system is a predicted patient diagnosis vector, designed to assist healthcare providers in their decision-making process.

Q-learning is particularly well-suited for this task as it allows us to optimally model the diagnostic process dynamically, enabling the system to adapt its recommendations based on sequential inputs. By integrating data from medical literature and clinical guidelines – and eventually real existing patient data – into a structured knowledge base, we ensure that our model operates within the boundaries of established medical practices.

2 Related Work

The task of developing intelligent decision support systems in healthcare has been explored through various approaches including within dataset development, reinforcement learning applications in medicine, and natural language processing (NLP) in medical diagnosis.

2.1 Dataset Development for Healthcare Applications

High-quality datasets form the backbone of decision-making models in medicine. The MIMIC-IV dataset Johnson et al. (2024) is a widely recognized benchmark, providing comprehensive patient health records for clinical research. Its focus lies within intensive care unit (ICU) data, but limits its applicability to understand specific domain knowledge. Similarly, disease symptom datasets such as the Disease Symptoms and Patient Profile Dataset UoM190346A (2023) and the Symptom2Disease Dataset Barman (2023) provide valuable symptom-disease mappings. However, due to the limited size of these datasets, each required significant preprocessing and enrichment. ENT-specific datasets like Lynx.MD Lynx.MD (2024) enhances domain-specific knowledge limitations but are relatively new and underutilized in reinforcement learning applications, which requires continued cleaning and preprocessing.

2.2 Reinforcement Learning in Medical Decision Making

Q-learning has been applied to optimize treatment strategies, showing promise in improving decision accuracy and consistency. Unlike traditional rule-based systems, RL models, such as those discussed by Helou in the context of uncertainty in medical decision-making Helou (2020), dynamically adapt to patient responses. While RL frameworks demonstrate strong theoretical potential, their implementation in high-stakes environments, like medical diagnostics, requires careful reward engineering and robust validation to mitigate risks associated with incorrect decisions. We've built upon this to ensure that the purpose of our RL model is to enhance medical diagnosis procedures, not fully replace them.

In establishing a robust foundation for comparison, it is essential to consider traditional machine learning models that have been extensively used in medical diagnosis. Kononenko (2001) provides a comprehensive review of machine learning algorithms applied to medical diagnosis, highlighting the effectiveness of classifiers such as Naïve Bayes and Decision Trees. According to Kononenko, Naïve Bayes classifiers typically achieve accuracies ranging from 78% to 85%, while Decision Tree classifiers exhibit accuracies between 82% and 88% across various medical diagnosis tasks Kononenko (2001). These models serve as standard baselines due to their simplicity, interpretability, and relatively low computational requirements.

2.3 Natural Language Processing for Medical Knowledge Extraction

Advances in NLP have facilitated the extraction and standardization of medical knowledge. Models like BioMedLM Bolton (2024); CRFM (2024), trained on large-scale biomedical corpora, have enabled the understanding and generation of domain-specific text. While these models excel in processing unstructured data and enhancing interoperability, they lack the inherent capability to model sequential decision processes, a key requirement for diagnostic support systems.

Overall, our work seeks to bridge these gaps by integrating domain-specific datasets, reinforcement learning techniques, and effective input/output tools into a cohesive framework for medical diagnosis. By leveraging Q-learning for sequential decision-making and incorporating knowledge from both

qualitative and quantitative patient data, we aim to provide accurate and reliable medical diagnosis backed by Q-learning algorithms.

3 Dataset and Features

Our dataset was comprised of multiple specialized datasets and case studies. On Kaggle, we found our base dataset with ten common diseases and the associated symptoms for each disease. For training purposes, we decided to standardize how symptoms are represented across each dataset. We have one column for each symptom and for each disease case, the presence of 1 means the symptom was observed and 0 indicates no symptom was observed in this case. Our next step was to continuously add datasets to expand the scope of our model. We found specialized datasets for cases of covid, malaria, zika, etc. and each dataset was passed through custom scripts for keyword extraction to format the dataset correctly before being appended. After exhausting all datasets we could find in Kaggle, we switched to clinical guidelines and case studies from PubMed. For each abstract of the article, we utilized LLMs (GPT and Llama) to extract symptoms. Each symptom mentioned would receive a 1 and the rest padded with zeros before being appended to our dataset. Eventually, after 14 Kaggle datasets and over 1500 PubMed articles, we amassed a dataset of over 80k individual cases of various diseases.

Our next step was to preprocess our dataset for training. The first step was to down sample our dataset so each disease has an equal number of representation in our training set. We had 400 diseases and downsampled to 16 cases for each disease, resulting in a final training set of 64k lines. Each line began with the disease column (representing the diagnosed case) followed by 660 collected symptoms through our webscraping. After coalescing similar symptoms and identifying identical symptom entries, we narrowed down our columns to 570. The diverse array of symptoms ranged from "localized back pain" to "weak urine flow." For later validation purposes, we set aside 1/16th of the dataset (exactly one entry per disease) for validation and accuracy assessment.

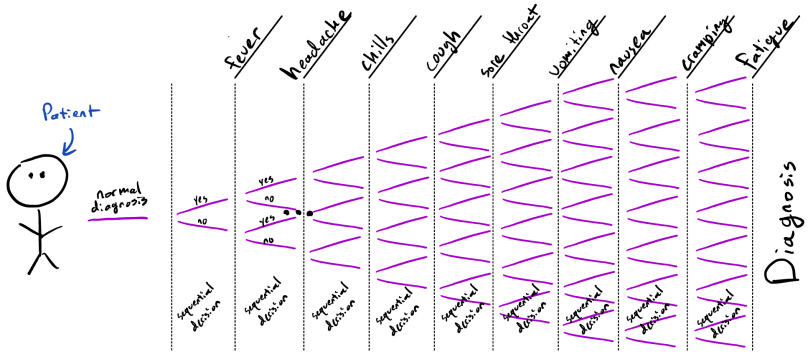

Figure 1: Sequence of Binary Representation of Symptoms.

4 Methods

The baseline Q learning model for this task involves creating our utility table or q_table of all states with their respective action. For a set number of iterations, usually in the thousands, we select a random state and either choose a random action or our best action according to preset parameters. The utility or reward of our given state is the reward from performing our best action combined with the best action reward of our next state discounted by Gamma, a preset parameter.

$$U_{a,s} = R_{a,s} + \gamma * \underset{a}{\text{Max}}(U_{next\_s}) \tag{1}$$

Given we have n = 570 symptoms and 400 diseases, this would require a q_table of dimensions 2^{570} by 400. This is because each symptom has two potential outcomes (1 or 0, present or not). The memory required for this baseline model was simply not possible. Every existing computing engine comes no where close to meeting the RAM requirement for this baseline Q learning algorithm. Thus, we propose the following modifications which attempt to address this issue.

4.1 Recursive Decision Trees

Our proposed method is for recurred subtrees. The idea is to narrow down the symptoms somehow and prepare a list of "candidates" to be run on the subset of symptoms. Our first step was to develop a function to map all 570 symptoms into 20 broad categories. After bucketing each symptom into its respective general category, we ran a script to reformat the data into a condensed version–one containing just 20 symptom categories instead of the full 570. Any presence of a specific symptom would result in the presence of the broader category. For example, "back pain" and "wrist pain" would be condensed into one symptom titled "localized pain." For each case, any presence of one of these specific symptoms would result in the presence of the broader symptom category.

$$\text{broad\_symptom}(U) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} \text{specific\_symptom}_i \geq 1, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Broad Symptom Categories
Localized Pain, Mobility and Muscular Issues, Gastrointestinal Discomfort, Localized Swelling, Cardiac Symptoms, Cognitive or Neurological Issues, Temperature Regulation Issues, Localized Tenderness, Skin Irritation or Abnormalities, Infections, Bleeding or Blood-Related Symptoms, Fatigue or Low Energy, Appetite and Weight Changes, Respiratory Issues, Sensory Issues, Vision and Hearing Issues, Mood and Behavioral Change, Bowel and Bladder Issues, Hair and Nail Abnormalities, Developmental Issues

After bucketing out 570 symptoms into 20, we could now train our Q learning algorithm. Our state space is now more manageable at 2^{20} and our action space consists of either asking about a symptom or making a diagnosis. For each iteration, the algorithm chooses an action that maximizes estimated utility guided by our q_table. An action to ask about a symptom advances the model to another state. During training, this action results in equal likelihood of 0 or 1 for any particular symptom. Rewards are only given if a model successfully diagnoses the correct disease from a state that is observed from our case studies. This forces the model to optimally ask for symptoms and eventually make an accurate diagnosis. Our utility function is as follows.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\big] \tag{3}$$

Once all symptoms are asked or if the minimum confidence in a diagnosis is met, the algorithm returns its top choice for a diagnosis and an array of other diseases that also bear strong probability confidence.

If multiple diseases likelihoods are similar, we move to the second part of the algorithm which narrows down the candidates using the full symptoms list. The idea here is to use the more specific symptoms like "knee swelling" or "sore throat" to isolate the true culprit instead of our broader categories such as "sensory problems." The large dataset is filtered out by candidate diseases and diseases not included in our candidates are discarded. To cut down on the symptoms, every insignificant column is removed. In other words, every column that has the same observance (1 or a 0) for all lines/cases is removed as it doesn't help distinguish our candidates. The resulting leftover data and table is fed into another iteration of Q learning which is usually fast, only consisting of less than three diseases and a handful of symptoms. The final output is the action with the highest reward from our second iteration.

Table 2: Before Filter for Secondary Training

Disease	Headache	Fatigue	Nausea	Juandice	Back Pain	Body Aches
Malaria	1	1	1	1	0	0
Yellow Fever	1	1	0	0	1	1

Table 3: After Filter for Secondary Training

Fatigue	Nausea	Juandice	Back Pain	Body Aches
Malaria	1	1	0	0
Yellow Fever	0	0	1	1

These tables show how developing a list of candidates and cutting the insignificant columns reduces the space and computation needed to further isolate decision making. This subtree method means we can provide an accurate diagnosis without having to expense the necessary computing power for the full decision tree. Our approach suggests a recursive approach to learning applied to large complex bayesian networks can provide fast and accurate decision making.

5 Experiments, Results, and Discussion

From a qualitative standpoint, our model performs reasonably well for the naked eye. Picking a disease at random and trying to isolate for it through the symptoms usually bears fruit. Our quantitative tests also show promise. From our test set which comprises one medical record per disease, we achieve an overall correct top confidence accuracy in 86.25% of all cases. And in every case, the correct diagnosis shows up in our top three most confident candidate diseases.

During training, we set certain hyperparameters to guide our model's final results. To balance old information with new information, we set our alpha level at 0.4. This means new information during training has a near equal weight to existing information. A gamma of 0.9 to discount future action rewards incentivizes a faster diagnosis of disease rather then running through all the symptoms. Because of the large state space, we opted for a low epsilon, choosing actions that we previously believed to yield the most reward.

When contextualizing our results against established baselines from the literature, we find that our Q-learning based approach not only demonstrates strong performance but also surpasses most classical machine learning models reported by Kononenko (2001). In particular, Kononenko's review notes that Naïve Bayes and Decision Tree classifiers typically achieved accuracies in the range of 78–85% and 82–88%, respectively, for general medical diagnosis tasks. Our model, which incorporates a two-stage approach of coarse-to-fine symptom selection, reached an accuracy of 86.25%, placing it in the upper end of established baselines. This improvement suggests that strategically incorporating sequential decision-making and reinforcement learning principles can yield more accurate diagnoses than static, single-pass classification approaches while also significantly reducing training time.

Table 4: Comparison of Diagnostic Accuracy Between Q-Learning Model and Baseline Classifiers

Method	**Accuracy (%)**	**Reference**
Naïve Bayes	86.0	Kononenko (2001)
Semi-naïve Bayes	85.0	Kononenko (2001)
k-Nearest Neighbors (k-NN)	72.0	Kononenko (2001)
Neural Network	76.5	Kononenko (2001)
Q-Learning (Ours)	86.25	Medical TWEAKinator

6 Conclusion and Future Work

Our Q-learning approach, enhanced by recursive symptom grouping and candidate refinement, demonstrates feasibility and strong performance in a complex medical diagnosis task. We reduce computational intractability by leveraging broad symptom categories and then eventually zoom into specific symptoms only for closely competing diseases. This dual-stage method yielded high accuracy and extremely practical training times for a task that historically has required massive amounts of data, complex algorithmic design, and overwhelming training time.

Regarding future work, we plan to integrate real patient data from clinical notes (in collaboration with software at Stanford Medicine that uses speech-to-text to write clinical notes), improve symptom categorization, and refine confidence thresholds We will also explore alternative RL algorithms and potentially non-model-free approaches. Ultimately, our approach opens pathways for more efficient, accurate, and unbiased AI-assisted diagnosis with the goal of eliminating distrust in patient-doctor interactions.

7 Contributions

Daniel Yang: I researched majority of the datasets and created all the code to format, clean, and compile the training dataset. This included downsampling, keyword extractions, manual labeling, column formatting, etc. I also brainstormed and engineered the new modifications for the Q learning algorithm. Specifically, the mapping, recursive elements, candidate selection, thresholding, etc. I wrote the abstract (partially), introduction (partially), methods, and results section. Also, to account for my additional course unit, I put in 30 more hours than the other contributors.

David Maemoto: I worked partially on researching the datasets and medical articles and related works that were key inspirations for our paper. In the final paper, I developed the first Q-learning model that ran on a conglomerate of Kaggle datasets that can be found here: CS 238 First Q-learning draft Github. I also wrote the abstract, introduction, and related works sections.

Ethan Farah: I scraped through the PubMed database of articles related to key words we identified in the Kaggle dataset aggregation that Daniel created in order to create a sort of prior knowledge from existing literature and research regarding proper protocols for specific symptoms and identify symptoms as "priors" for specific diseases. I interviewed some physicians at Stanford hospital to understand proper use cases for a model in their daily lives, and also successfully submitted an IRB to Stanford Medicine in order to attain access to (PIH scrubbed) real patient data and clinical notes (which contain symptoms and diagnoses in a very concise form). I also wrote the conclusion, parts of abstract/intro/related works/discussion, and the baseline.

References

Niyar R Barman. 2023. Symptom2disease dataset. `https://www.kaggle.com/datasets/niyarrbarman/symptom2disease/data`. Accessed: 2024-12-06.

Elliot Bolton. 2024. Biomedlm: A 2.7b parameter language model trained on biomedical text. *arXiv preprint arXiv:2403.18421*. Accessed: 2024-12-06.

Stanford CRFM. 2024. Biomedlm. `https://huggingface.co/stanford-crfm/BioMedLM`. Accessed: 2024-12-06.

Marieka A. Helou. 2020. Uncertainty in decision making in medicine: A scoping review and thematic analysis of conceptual models. *Academic Medicine: Journal of the Association of American Medical Colleges*, 95.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Brian Gow, Benjamin Moody, Steven Horng, Leo Anthony Celi, and Roger Mark. 2024. Mimic-iv. *Physio Net*.

Igor Kononenko. 2001. Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109.

Lynx.MD. 2024. Ent data. `https://lynx.md/ent-data/`. Accessed: 2024-12-06.

UoM190346A. 2023. Disease symptoms and patient profile dataset. `https://www.kaggle.com/datasets/uom190346a/disease-symptoms-and-patient-profile-dataset/data`. Accessed: 2024-12-06.