ABSTRACT

YANG, XI. Time-aware Hierarchical EM Energy-based Subsequence Clustering for Inducing Offline Decision-making Policies with Heterogeneous Reward Functions. (Under the direction of Dr. Min Chi.)

Reward function plays a critical role in Reinforcement Learning (RL). Many tasks involve manually crafted reward functions, which require expertise and have multiple factors involved that are often difficult to balance. **Apprenticeship Learning (AL)** addresses this problem by learning reward functions and inducing policies from observing expert agents' behaviors. Most existing AL methods assume a *homogeneous* latent reward function and are designed for online tasks, however, in human-centric tasks, *e.g.*, education and healthcare, the reward functions are often *heterogeneous across different sub-populations and varying over time* and need to be learned *offline*.

We propose a general offline AL framework named **T**ime-aware **H**ierarchical **EM** **E**nergy-based **S**ubsequence (**THEMES**) clustering, which aims at inducing more accurate decision-making policies with *evolving*-heterogeneous reward functions varying over time. Our framework consists of two components: *(I) sub-trajectory partitioning* based on observations; and *(II)* Inducing policies with heterogeneous reward functions varying *across sub-trajectories*. The two components can be conducted iteratively to refine the sub-trajectory partitioning and to induce more accurate policies.

In *Component I* (*i.e.*, **sub-trajectories partitioning**), considering the multi-series nature and irregular intervals in real-world scenarios, we proposed a Multi-series Time-aware Toeplitz Inverse Covariance-based Clustering (**MT-TICC**) algorithm. Based on observed states, MT-TICC can partition and cluster sub-trajectories, simultaneously. The effectiveness of MT-TICC was first validated over a hand gesture dataset with ground truth. Then we further applied it to learn the progressive stages of an extremely challenging disease — sepsis, taking electronic healthcare records (EHRs) as the testbed. The results showed that the MT-TICC-learned clusters can effectively unveil interpretable stages during the sepsis progression and assist in the early diagnosis of septic shock.

In *Component II* (*i.e.*, inducing policies with **across-heterogeneous** reward functions), we employed an Expectation Maximization Inverse RL (**EM-IRL**) algorithm, which can cluster the trajectories and learn the policy for each cluster, simultaneously. The effectiveness of EM-IRL was first validated in three simulation environments with ground truth. Then we further validated it using students' data collected from a real-world intelligent tutoring system. The results showed the EM-IRL can effectively cluster students with various learning intentions into different subtypes, including a "learning-oriented" subtype that aims to learn regardless of time; a "no-learning" subtype that spends less time and fails to learn; and an "efficient-oriented" subtype that not only learns significantly comparing to the "no-learning" but also spends less time than the "learning-oriented".

Taking respective powers of the above two components, we further designed a unified **THEMES** framework to model **evolving-heterogeneous** reward functions, which will iteratively refine the learned sub-trajectories and induce more fine-grained policies accordingly. Specifically, based on MT-TICC, we proposed a Reward-regulated MT-TICC (**RMT-TICC**), which can encode the *inter-*

ventional patterns by introducing a reward regulator because interventions have a great impact on sub-trajectories partitioning and are latently driven by rewards. Besides, based on the EM-IRL and an offline AL method of Energy-based Distribution Matching (**EDM**), we propose an **EM-EDM**, which can induce policies *offline* and efficiently model large continuous state spaces. The whole framework was evaluated over the EHRs by 1) inducing heterogeneous policies when treating sepsis at different progressive stages and 2) modeling the progression of sepsis by incorporating THEMES-derived progressive patterns as well as interventional patterns as supplementary features. Based on THEMES, more fine-grained policies can be derived, which can better evaluate clinicians' interventions and also recommend more personalized treatments to clinicians in time.

Time-aware Hierarchical EM Energy-based Subsequence Clustering for Inducing Offline Decision-making Policies with Heterogeneous Reward Functions

by
Xi Yang

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2022

APPROVED BY:

Dr. David L. Roberts	Dr. Xipeng Shen
Dr. Ranga Raju Vatsavai	Dr. Min Chi Chair of Advisory Committee

DEDICATION

To my beloved family and friends, for their support, encouragement, and love.

BIOGRAPHY

Xi Yang was born in Shaanxi, China. In 2008, she got early admission to Xi'an Jiaotong University with a major in Automation. After four years of study, in 2012, she obtained her Bachelor's degree and got a postgraduate recommendation to the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University to pursue her Master's degree with a major in Pattern Recognition and Intelligent Systems. During this period, she got a chance to learn machine learning and data mining techniques and developed her interest in artificial intelligence via a project on face recognition and facial age estimation. After achieving her Master's degree in 2014, she decided to continue her research work. From 2015 to 2016, she had an internship in the IDEA lab of the University of North Carolina at Chapel Hill and worked on a project about using machine learning to analyze medical imaging data. In 2016, she was admitted to the Department of Computer Science at North Carolina State University to continue her research.

During her doctoral research, Xi worked at the NC State Center for Educational Informatics under the supervision of Dr. Min Chi. Being involved in several projects with real-world applications, she enhanced her research interests in machine learning and data mining and further expanded her interests in deep learning and reinforcement learning. Specifically, based on these techniques, she developed novel methods in the field of both education and healthcare. Besides her academic work at school, Xi also explored utilizing advanced machine learning techniques to solve industrial problems. In the summer of 2020 and 2021, she interned at Visa Research and IBM Research, respectively, working with anomaly detection projects using machine learning methods. After her internship in 2021, she got a return offer. In May of 2022, she successfully defended her Ph.D. thesis and started working as a research scientist at IBM Research.

ACKNOWLEDGEMENTS

I'm grateful for the life-changing choice in 2016 when I decided to pursue my Ph.D. studies. Looking back on the past years, it has not always been smooth sailing, but I experienced, learned, and grew a lot. During this journey, I luckily met great teachers who guided and enlightened my academics and life, made sincere friends who spent many happy moments together, and got endless love from my dearest families as always. Without all these people's kind support and dedicated assistance in every step of this journey, this dissertation would have never been accomplished.

First of all, I would like to express my deepest appreciation to my academic advisor, Dr. Min Chi, for her invaluable guidance, strong encouragement, and profound trust. I'm always feeling extremely lucky to have Dr. Chi as my advisor. For me, she is not only an advisor but also a mentor and a role model. Her extensive expertise and enthusiasm toward research, her conscientiousness and patience for every student, and her kindness and thoughtful personality always inspire me to keep on improving myself to become a powerful and caring person like her.

Many thanks to my dissertation committee members, Dr. David L. Roberts, Dr. Xipeng Shen, and Dr. Ranga Raju Vatsavai, for their insightful and valuable suggestions, which are pretty helpful for me to reflect on the aspects that need to be improved, and I appreciate their support in rescheduling their life to accommodate my final defense time. Besides, I would like to thank Dr. Roger Azevedo, Dr. Michelle Taub, and Megan Wiedbusch from the University of Central Florida, and Dr. Soonhye Park, Vance Kite, and Amanda Hall from the STEM Education Department at NC State. They helped me a lot in interpreting the specialized knowledge in the MetaDash project. I am also thankful to Dr. George Rouskas, who always gives prompt support for my academic records. Additionally, I'm grateful to Fei Wang from Visa Research, Yu Deng, Larisa Shwartz, and Ruchi Mahindru from IBM Research, who give me valuable advice and assistance during my internships.

Thanks also go to former lab members: Hamoon Azizsoltani, Guojing Zhou, Liana Lin, Shitian Shen, Yuan Zhang, Song Ju, Farzaneh Khoshnevisan, Ye Mao, and Esha Sharma, and my fellow Ph.D. students: Yeojim Kim, Markel Sanz Ausin, Hyunwoo Sohn, Mark Abdelshiheed, John Hostetter, Ge Gao, Mitchell Plyler, Louise Rabinowitz, and Angela Zhang. I appreciate their insightful feedback and enjoy the discussions and cooperation with them. I would like to extend my sincere thanks to my friends: Xiao Zhang, Junqiu Guo, Holly Maguire, Rosemary Hart, Tianpei Xia, Shudi Shao, Shailaja Mallick, Manish Tripathy, Zifan Nan, Fan Zhang, Yuchen Sun, Jingzhu He, Peipei Wang, and Rui Shu. I feel grateful for their persistent support and companionship. The pleasant times we spent together and the beautiful sceneries we enjoyed together are all my precious memories.

Finally, words cannot express my gratitude to my family. I will give special thanks to my grandparents who brought me up, my parents and parents-in-law who always provide me with their selfless dedication and unconditional love, and my husband Zhengwang Wu who always puts up with me, protects me, and supports me. Also, thanks to my unborn baby girl, Luna, who accompanies me day and night during the most challenging time in my life, gives me the courage and strength to confront all difficulties, and encourages me to become a better person and a more powerful mom.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 Background and Motivation

Reinforcement Learning (RL) has experienced significant development in recent years. Specifically, it has been extensively applied in various *human-centric* applications, *e.g.*, education [Igl09a; Chi11a; Chi11b; She16b] and healthcare [Wan18; Yu21], where the intelligent agent in RL aims at learning a decision-making policy to maximize the expected cumulative *rewards* in the long run to benefit humans in better interacting with a certain system or with other humans [Ju22]. As a fundamental element in RL, the *reward function* defines the goals to achieve when the agent interacts with the environment, which provides the agent incentive to adopt better decision-making behaviors [Kae96]. Specifically, the reward function maps each perceived state (or state-action pair) of the environment to a scalar value and gives it as a credit or punishment to the RL agent, and the agent will learn a decision-making policy accordingly to maximize the expected cumulative rewards. Despite the importance of the reward function in RL, it is usually difficult to design, especially for *human-centric* tasks, where multiple factors need to be covered and traded off. In general, the reward function is hand-crafted beforehand, separate from the policy induction. However, manually designing an appropriate reward function is always labor-intensive and time-consuming [Goy19]. It commonly depends on the domain knowledge, hard to avoid the expertise blind spots [Abb04b]. Moreover, the manually specified reward function is likely to be misspecified, which can turn out to be inconsistent with the expected policy [Amo16].

To handle these difficulties, a family of Apprenticeship Learning (AL) algorithms have been proposed. Instead of inducing the policy under the guidance of a pre-defined reward function, AL aims at learning via observing and imitating a set of demonstrations provided by expert agents, who make decisions optimally or near-optimally with respect to an unknown underlying reward function [Abb04b]. By learning from demonstrations, the learner agent, *i.e.*, the apprentice, aims to learn a decision-making policy to behave as well as the experts. AL generally follows the procedure of 1) taking the demonstrations as input to infer the latent reward function, based on which a policy can be induced; then 2) rolling out the learned policy to update the reward function by minimizing the divergence between the rolled-out behaviors versus the demonstrations. This procedure will be

conducted iteratively until convergence. Getting rid of the difficulties in reward function design in RL, AL has been widely applied in various applications [Pan19; Raf15; Aso13; Wan20a].

When applying AL to model *human-centric* tasks, there are in general *two challenges*: 1) the existing AL are commonly *online*, requiring interacting with the environment iteratively for collecting new data and then updating the model accordingly [Abb04b; Zie08; Ho16; Fin16]. However, since the execution of a bad policy can be costly or even dangerous in human-centric tasks [Lev20], *e.g.*, healthcare and education, it is highly desired to induce the policies purely based on the demonstrated behaviors in an **offline** manner. 2) Most of the existing AL assume a *homogeneous* reward throughout and across different demonstrations. Whereas in real-world scenarios, the reward function can be **heterogeneous** either ***across*** the demonstrations or ***evolving*** over time. For example, in the education domain, when students make decisions during learning, their reward functions can be learning-oriented, efficient-oriented, or not learning; In the healthcare domain, the clinicians have different reward functions when improving patients from different progressive stages, *e.g.*, infection, inflammation, and shock.

Regarding the two factors of 1) *online* vs. *offline* and 2) homogeneous vs. heterogeneous reward functions either varying across trajectories (*across*-heterogeneous) or evolving over time (*evolving*-heterogeneous), AL approaches can be categorized into:

- *Online* AL with 1) a *homogeneous* reward function [Ng00; Abb04a; Sye07; Zie08; Bou11; Ho16; Fin16]; 2) *across*-heterogeneous reward functions [Aro20]; and 3) *evolving*-heterogeneous reward functions [Kri16; Hau17; Li17; Wan21; Wan22];

- *Offline* AL with 1) a *homogeneous* reward function [Ram07; Jar20]; 2) *across*-heterogeneous reward functions [Dim11; Cho12; Bab11]; and 3) *evolving*-heterogeneous reward functions.

Among the above AL categories, the **offline** setting with **evolving**-heterogeneous reward functions is the most generic case, while few of the previous methods have been specifically proposed for handling it. Therefore, to fill this gap, our major motivation in this thesis is *developing an AL framework to induce policies in an offline manner from given demonstrations, which are assumed to be driven by multiple evolving-heterogeneous reward functions*. It is especially beneficial for real-world **human-centric** tasks, where the *offline* setting without rolling out the policy is highly desired, and humans usually make decisions adaptively with much of the *heterogeneity*. By doing so, more accurate and fine-grained policies can be induced, which is enlightening for not only evaluating the humans' decision-making behaviors but also enabling the system to provide humans with proper recommended actions. For example, in education, students' learning behaviors can be better evaluated, so that intelligent tutors can accordingly tailor their assistance; In healthcare, clinicians' decision-making policies can be more accurately induced, so that proper individualized interventions can be suggested to improve the clinicians' treatment plans.

1.2 Research Questions

To achieve the task of offline learning for the *evolving*-heterogeneous reward functions, the key challenge lies in *how to effectively partition the trajectories*. If the sub-trajectories with different progressive patterns can be derived, then the problem converts to learning the *across*-heterogeneous reward functions over the sub-trajectories, which is more tractable to be addressed. The more precisely the sub-trajectories are partitioned, the more accurate the induced policies will be. Therefore, we focus on fulfilling two components of 1) *sub-trajectory partitioning*; and 2) offline learning *across*-heterogeneous reward functions. If these two components can be resolved, the ultimate task of offline learning for the *evolving*-heterogeneous reward functions can be fulfilled accordingly. We summarize this procedure as three research questions detailed as follows:

RQ1: *How can we learn **sub-trajectory partitioning** regarding different progressive patterns?*

We model the *sub-trajectory partitioning* as an unsupervised problem, which aims at partitioning and clustering the sub-trajectories, simultaneously. A recently proposed approach named Toeplitz Inverse Covariance-based Clustering (**TICC**) [Hal17] has shown great success in solving this problem by outperforming competitive baseline methods, *e.g.*, dynamic time warping [Rak13] and Gaussian mixture models [Rey09]. It has been successfully applied in various applications, *e.g.*, analyzing physical activities for patients with Alzheimer's [Li18] and segmenting the critical stages in disease progression [Gao22]. When applying TICC to more general real-world scenarios, there are two challenges: 1) TICC takes a single trajectory as input, while the real-world dataset usually consists of *multiple time series*; 2) The states in real-world datasets are usually recorded *irregularly*, while TICC treats irregular intervals equally and always encourages the consecutive states to be assigned into the same cluster. Therefore, the motivation for this research question lies in developing an approach, which can handle the *multi-series* input and can take *time-awareness* into consideration. The analyses will be carried out first over a hand gesture dataset with moment-by-moment ground-truth labels which indicate the progressive stages. Afterward, we will analyze a healthcare application with the data collected from electronic healthcare records (EHRs), to model the progressive stages of an extremely challenging disease — sepsis.

RQ2: *How can we model the **across-heterogeneous** reward functions?*

We model the learning of across-heterogeneous reward functions as a problem of clustering the trajectories and inferring the respective reward function for each cluster, simultaneously. Taking advantage of an offline Expectation Maximization Inverse Reinforcement Learning (**EM-IRL**) [Bab11], which has demonstrated significantly better performance compared to using a homogeneous reward function, we adapt it to a challenging scenario in education to analyze the heterogeneous reward functions across different students. Herein, various reward functions can reflect different students' learning intentions when they interact with an intelligent tutoring system. If a student's intention is detected to be off the track from learning, the tutors can accordingly provide he/she with proper personalized assistance. Since it is hard to figure out students' authentic learning intentions, there

are no ground-truth labels for directly evaluating the results. As a result, we will carry out our analysis over three simulation environments first, which have similar properties to the students' data, *i.e.*, temporal with diverse reward functions across different trajectories, while having the ground-truth labels. After evaluating the effectiveness of the algorithm, we will further deploy it to real-world students' data. By statistically comparing the learning gain and time across different student groups with varying reward functions, we aim to infer the students' underlying learning intentions.

RQ3: *How can we model the **evolving-heterogeneous** reward functions?*

Taking advantage of the above two research questions, *i.e.*, learning **sub-trajectory partitioning** and **across-heterogeneous** reward functions, we aim at further incorporating them into a unified framework to model the **evolving-heterogeneous** reward functions. Specifically, when conducting the RQ1 and RQ2 separately: 1) existing techniques for partitioning sub-trajectories mainly focus on *observed states* from the environment when learning the progressive stages, while the humans' *interventions* over the environment, which can greatly affect the partitioning results, have not been adequately explored; 2) The techniques for learning across-heterogeneous reward functions can capture such interventional patterns, while they commonly assume a *constant* reward function inside each trajectory, which cannot model the *evolving* progressive patterns. Therefore, a unified framework that can take the respective powers of RQ1 and RQ2 is highly desired: referring to the progressive stages indicated by sub-trajectory partitioning, more fine-grained interventional policies that evolve over time can be learned; Meanwhile, the derived interventional patterns can, in turn, contribute to refining the sub-trajectory partitioning results to indicate more accurate progressive stages. The effectiveness of the unified framework will be evaluated in healthcare by 1) inducing the clinicians' heterogeneous decision-making policies when treating sepsis patients, and 2) utilizing the extracted progressive and interventional patterns for better modeling the sepsis progression.

1.3 Contributions

Based on the three research questions specified in Section 1.2, we carry out our analysis and experiments accordingly. The contributions of this thesis lie in the following three aspects:

• To derive *sub-trajectory partitioning* with different progressive patterns, based on TICC [Hal17], we propose a Multi-series Time-aware TICC (**MT-TICC**) [Yan21], by taking *multi-series* nature of input trajectories and the *time irregularity* into consideration. In a case study with real-world hand gesture datasets, our experimental results show that MT-TICC can significantly outperform the state-of-the-art models, including the original TICC. Then in a healthcare application for modeling sepsis progression using EHRs data, we treat the MT-TICC-derived progressive stages as additional features for the task of septic shock early prediction. The results show that by incorporating MT-TICC-learned patterns which indicate the progressive stages, the early prediction performance can be significantly improved. Additionally, the sub-trajectory clusters derived by MT-TICC can convey meaningful insights and shed some light for clinicians to better understand the sepsis progression.

• To learn the *across*-heterogeneous reward functions, we employ the **EM-IRL** algorithm [Bab11]. In the original EM-IRL, it is required to pre-define the number of clusters for reward functions, which is usually difficult to be properly assigned. Herein, we embedded the original EM-IRL into a more general heuristic framework which can automatically determine the optimal number of clusters from the data [Yan20]. As we know, this is the first IRL-based work for analyzing the heterogeneous reward functions in a *human-centric* task. Our experimental results in three simulation environments with ground-truth labels show that EM-IRL can accurately cluster the data with different decision-making policies. Additionally, in the education domain, the EM-IRL can effectively derive student subtypes with different learning intentions, including a "learning-oriented" subtype of students that learn the material as much as possible regardless of the time, an "efficient-oriented" subtype of students that learn efficiently, and a "no learning" subtype that spends less amount of time and fails to learn.

• Equipped with MT-TICC [Yan21] and EM-IRL [Yan20], we further propose an AL framework named Time-aware Hierarchical EM Energy-based Sub-trajectory (**THEMES**) clustering. Specifically, based on MT-TICC, we propose a Reward-regulated MT-TICC (**RMT-TICC**), which can encode the *interventional* patterns by introducing a reward regulator, motivated by the fact that interventions are latently driven by rewards. Using our RMT-TICC, more fine-grained sub-trajectories can be derived to indicate different progressive stages. Besides, based on an offline AL named energy-based distribution matching (EDM) [Jar20] and the EM-IRL [Bab11; Yan20], we propose an Expectation Maximization EDM (**EM-EDM**). Utilizing our EM-EDM, decision-making policies over different progressive stages can be effectively induced offline, without the need of rolling out the policies iteratively. Unlike original EM-IRL which takes discrete states as input, EM-EDM can efficiently model large continuous state spaces. In a healthcare application for sepsis progression modeling with EHRs data, our THEMES framework demonstrates better performance compared to competitive baselines in both 1) *policy induction* from demonstrated trajectories and 2) *early prediction* of septic shock by incorporating the THEMES-derived progressive patterns and interventional patterns as supplementary features when modeling the sepsis progression. To the best of our knowledge, this is the first work incorporating AL-derived interventional patterns for disease progression modeling.

The rest of this thesis is structured as follows: Chapter 2 describes the background techniques that would be covered throughout this thesis. Chapter 3 and Chapter 4 launch the RQ1 and RQ2 as specified in Section 1.2, respectively. In each chapter, the overview, related work, methodology, case study, experiments, and a summary will be presented. Based on Chapter 3 and Chapter 4, in Chapter 5, a unified framework, *i.e.*, THEMES, will be introduced. To evaluate the THEMES framework in the healthcare domain with EHRs data, we first apply it to induce clinicians' decision-making policies when treating sepsis patients in Chapter 5; Then we further take THEMES-derived patterns as additional features for a disease progression modeling task to early predict the septic shock in Chapter 6. Finally, Chapter 7 concludes this thesis and proposes potential future works.

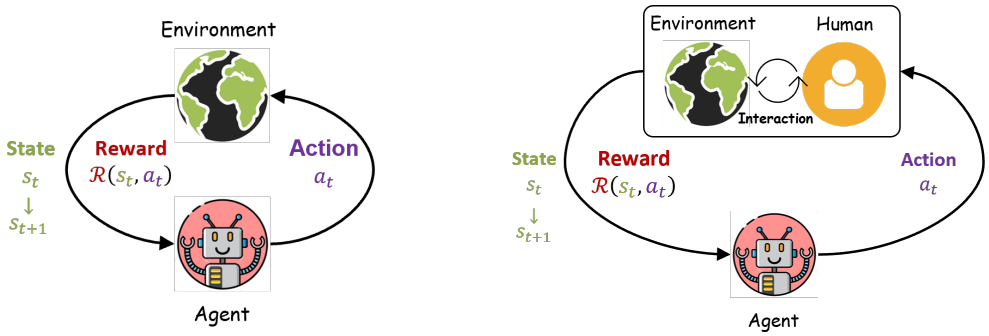
2 BACKGROUND

This chapter will elaborate on the background of this thesis. Specifically, we will first introduce an important factor – *reward function*, in reinforcement learning. To handle the difficulties in designing proper reward functions, different categories of apprenticeship learning (AL) algorithms have been proposed, and they will be presented in this chapter. AL is especially beneficial for complex *human-centric* tasks, where the reward functions are hard to handcraft. We will illustrate how AL has been successfully applied in human-centric applications, for example, education and healthcare.

2.1 Reward Function

Reinforcement learning (RL) can be modeled as a computational problem to find an optimal policy for choosing actions in a Markov decision process (MDP). An MDP describes a stochastic control process using a 5-tuple $< S, A, T, R, \gamma >$, where S denotes the states; A represents the actions; T is the probabilities transiting from one state to another when taking different actions; R is a reward function, and $\gamma \in [0, 1)$ denotes a discount factor for future rewards. As illustrated in Figure 2.1(a), at the t-th timestamp, the environment is in state s_t; the intelligent agent executes action a_t and receives reward $r_t = R(s_t, a_t)$, then the environment will transit to the state s_{t+1}. RL aims at training the agent to make optimal decisions, so as to maximize the expected cumulative rewards. The RL techniques experienced explosive development in recent years, and they have been extensively applied in various applications. Especially, it has shown great potential in *human-centric* tasks [Chi11a; She16b; Wan18; Yu21], *e.g.*, education and healthcare. As illustrated in Figure 2.1(b), in *human-centric* RL, the agent aims at inducing a decision-making policy to gain maximum expected rewards, to benefit the human in the loop in better interacting with the environment [Ju22].

The reward function R plays a critical role in RL to praise/punish the agent when deriving the optimal policy. As mentioned in [Abb04b], "the entire field of RL is founded on the presupposition that the reward function, rather than the policy, is the most *succinct, robust, and transferable* definition of the task." Generally, the reward function requires to be specified before the policy induction and it is elaborately hand-crafted to reflect certain tasks. Since there are usually multiple factors involved in decision-making, especially for *human-centric* tasks, manually designing an



(a) MDP in RL

(b) MDP in Human-centric RL

Figure 2.1 MDP for Traditional RL vs. Human-centric RL.

appropriate reward function to trade off these factors can be not only expertise-intensive but also time-consuming [Goy19]. For example, when inducing the decision-making policy in a human-driving environment, different factors (*e.g.*, speed, distance to other cars, gasoline consumption) need to be considered, which is challenging to be balanced via a unified reward function. Besides, if the reward function is misspecified, the induced behaviors may turn out to be divergent from the expected behaviors [Amo16]. As a result, the requirement of manually specifying the reward function poses a significant barrier to the broader applicability of RL for complex human-centric tasks, while a more robust data-driven approach, which can integrate the reward function learning during the process of policy induction, is highly desired.

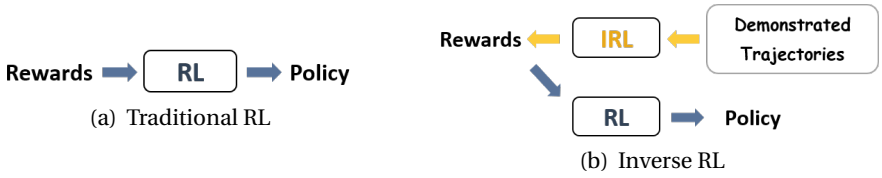
2.2 Apprenticeship Learning (AL)

The difficulties in reward function design in RL triggered the development of apprenticeship learning (AL) [Abb04b; Ng00]. AL is also known as imitation learning [Sch99] or learning from demonstrations [Arg09]. In traditional RL, the reward function R is a part of the input, based on which the agent will induce an optimal policy; While in AL, the input is a 4-tuple, *i.e.*, $MDP\backslash R$ without knowing the R, together with a set of demonstrated trajectories $\mathcal{T}$, which are assumed to be illustrated by expert agents, by rolling out a certain optimal or near-optimal policy.

2.2.1 AL Algorithms

The existing AL algorithms in general can be divided into the following three categories:

• **Behavior Cloning**: The most intuitive AL method is behavior cloning, which is modeled as a supervised learning problem [Sye12; Raz12], by building up a mapping from states to actions to greedily imitate the demonstrated actions, without reasoning about the distributions for the actions [Pom91]. Since it is usually costly to collect a large amount of experts' data, it would be hard to learn a robust behavior cloning model to fully capture the experts' behavioral patterns given the limited size of state-action pairs. In this case, the compounding errors may lead the agent to drift away



(a) Traditional RL

(b) Inverse RL

Figure 2.2 Traditional RL vs. Inverse RL.

Algorithm 1 General Process of IRL

1: Input: $MDP\backslash R =< S, A, T, \gamma >$ and demonstrated trajectories $\mathscr{T}$
2: Initialize: Parameter θ in reward function R_θ
3: **repeat**
4: **Step 1** Learn the policy π by RL
5: **Step 2** Update θ to minimize the divergence between $\mathscr{T}$ and behaviors rolled out by π
6: **until** Convergence
7: Output: R_θ, π

from the demonstrations [Ros11]. Once the agent drifts and encounters certain states that are not covered in training data, they may be confused about what action should be taken to return to the demonstrated states.

To handle the issues in behavior cloning, some Inverse Reinforcement Learning (IRL) as well as Adversarial Imitation Learning (AIL)-based methods have been proposed. Instead of merely imitating the demonstrated actions, these methods can consider the causality of the actions, by explicitly or implicitly learning the reward function in a data-driven manner.

• **Inverse Reinforcement Learning (IRL)**: As illustrated in Figure 2.2, IRL follows a reverse procedure compared to the traditional RL. Most IRL methods follow the general framework as shown in Algorithm 1 [Aro21]: Taking the 4-tuple $MDP\backslash R$ together with experts' demonstrated trajectories $\mathscr{T}$ as input, the parameter θ for the reward function R_θ is initialized; Then in Step 1, given the R_θ, general RL methods can be applied to induce a policy π; In Step 2, the divergence between the behaviors rolled out by the learned policy π versus the given demonstrations $\mathscr{T}$ is minimized to update the θ. Step 1 and Step 2 are repeated until the divergence is reduced to the desired level. Once the reward function is learned, it can be further employed to induce the policy based on traditional RL. Finally, the algorithm can output both the reward function R_θ and policy π.

The existing IRL algorithms can be generalized into two categories: *maximum margin-based* methods and *probabilistic model-based* methods. In *maximum margin-based* methods, the solver aims at finding a reward function that enables the demonstrations to be better than other possible behaviors with a margin. For example, in [Abb04b], Abbeel and Ng formulated the IRL as a quadratic programming problem, maximizing the minimal margin between the feature expectation of the expert's demonstrations and other alternative behaviors. In [Rat06], Ratliff *et al.* proposed a maximum margin planning algorithm over a space of policies, which learns mappings from features to rewards, so that an optimal policy in an MDP with these rewards mimics the demonstrated

behaviors. The maximum margin-based methods are generally plagued by the *ill-posed issue of uncertainty* [Ng00], since there can be multiple reward functions explaining the expert's behaviors. The *probabilistic model-based* methods can handle this issue by modeling the uncertainty by a probability distribution. Specifically, it models the demonstrations to be exponentially proportional to the rewards, *i.e.*, the larger the reward, the higher the probability for a demonstration to be observed. For example, Ramachandran and Amir [Ram07] proposed a Bayesian IRL (BIRL), which combined prior knowledge and evidence from the expert's demonstrations to derive a probability distribution over the reward functions. Similarly, Ziebart *et al.* [Zie08] proposed a Maximum Entropy IRL (MaxEntIRL). Since the larger the entropy, the more information could be retrieved from data, the MaxEntIRL gave the least biased estimate given the demonstrations [Zhi12]. Boularias *et al.* [Bou11] further extended the MaxEntIRL with a relative entropy version when dynamics of the environment were not given. The aforementioned works generally represented the reward function as a weighted linear combination of features. To model a more complex nonlinear reward function, based on BIRL, Levine *et al.* [Lev11] proposed a Gaussian process IRL, which modeled the reward function non-linearly by Gaussian process and learned the kernel hyperparameters to capture the structure of the rewards. Additionally, Wulfmeier *et al.* [Wul15] extended the MaxEntIRL paradigm using deep neural networks to approximate the reward functions in large state spaces.

• **Adversarial Imitation Learning (AIL)**: As we can see in Algorithm 1, the IRL generally involves an iterative loop for explicitly inferring a reward function and inducing a policy by traditional RL. The iteratively conducted RL in each loop hinders it to be efficient. To handle this issue, recently there have been some AIL approaches proposed [Ho16; Fin16]. Unlike IRL which infers the reward function *explicitly* in each loop, in AIL, the reward function is *implicitly* learned together with the policy induction. Specifically, it follows the general adversarial idea, with 1) a *generator* indicating the policy to be learned, which can be rolled out to collect data, and 2) a *discriminator* to distinguish the divergence between the rolled-out behaviors versus the experts' demonstrations to update both the generator and discriminator, simultaneously. Herein, the output of the discriminator can be considered as an implicit reward signal to guide the induction of the policy.

2.2.2 AL Applications in Human-centric Tasks

AL has been widely applied in various *human-centric* applications, *e.g.*, traffic [Zie08; Bou11; Pan19; Oh19; Ram07], education [Raf15; Raf16b], and healthcare [Aso13; Wan20a; Jar20; Wan21; Wan22], and it has achieved promising performance and demonstrated practical values. In this thesis, we mainly focus on applying AL to the domains of *education* and *healthcare*.

• **Education**: IRL has been employed in education for assessing learning status, diagnosing misconceptions, and providing informative feedback. Anna *et al.* [Raf15] applied IRL to automatically infer learners' knowledge of the environment in educational games. In another of their works [Raf16b], they employed IRL to model how a learner solves algebraic equations to estimate their proficiency in several skills. The personalized feedback then focused on the skill which was the least proficient and introduced a combination of existing educational content and scaffolded practice.

• **Healthcare**: Asoh *et al.* [Aso13] applied Bayesian IRL to medical records and explored the reward function that doctors have in mind when treating diabetes. Wang *et al.* [Wan20a] employed a generative adversarial AL to infer continuous dosage when giving treatment recommendations for sepsis patients. Jarrett *et al.* [Jar20] proposed an offline AL, *i.e.*, EDM, and applied it to the electronic healthcare records consisting of patients treated in intensive care units from the Medical Information Mart for Intensive Care. More recently, Wang *et al.* used hierarchical AL methods for managing sepsis patients under various progressive stages with multiple reward functions [Wan21; Wan22].

3 MT-TICC: SUB-TRAJECTORY PARTITIONING

* *This chapter has been published as*:

> **X. Yang**, Y. Zhang, and M. Chi, "Multi-series Time-aware Sequence Partitioning for Disease Progression Modeling." In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

In this chapter, we focus on learning **sub-trajectory partitioning** in a *healthcare* application with electronic healthcare records (EHRs). EHRs are comprehensive longitudinal collections of patient data that play a critical role in modeling disease progression to facilitate clinical decision-making. Specifically, based on EHRs, we focus on *sepsis* – a broad syndrome that can develop from nearly all types of infections (*e.g.*, influenza, pneumonia). The symptoms of sepsis, such as elevated heart rate, fever, and shortness of breath, are vague and common to other illnesses, making the modeling of its progression extremely challenging. Motivated by the recent success of a novel sub-trajectory clustering approach: Toeplitz Inverse Covariance-based Clustering (TICC) [Hal17], we model the sepsis progression as a sub-trajectory partitioning problem and propose a Multi-series Time-aware TICC (**MT-TICC**), which incorporates *multi-series nature* and *irregular time intervals* of EHRs. The effectiveness of MT-TICC is first validated via a case study using a real-world hand gesture dataset with ground-truth labels. Then we further apply it for sepsis progression modeling using EHRs. The results suggest that MT-TICC can significantly outperform competitive baseline models, including the TICC. More importantly, it unveils interpretable patterns, which sheds some light on a better understanding of sepsis progression.

3.1 Overview

Electronic health records (EHRs) are comprehensive longitudinal collections of patients' health data. The extensive application of EHRs in medical systems has accelerated the development of various computational methods for understanding patients' medical history, identifying interesting cohorts, predicting potential risks, and evaluating interventions [Che15; Cho16b; Zha19]. Among them, *disease progression modeling* (DPM) is a crucial task, which monitors disease development,

predicts future risks, and assists clinicians in making effective interventions. Given its importance, many recent works aim to develop automated solutions for DPM using machine learning techniques [Cho16a; Cho16b; Bay17]. In this chapter, we focus on modeling the progression of an extremely challenging disease – *sepsis*, which is a life-threatening organ dysfunction and a leading cause of death worldwide [Sin16]. Without timely diagnosis and intervention, sepsis can progress from infection to septic shock, which is the most severe stage with a mortality rate as high as 50%.

Despite the great importance, modeling the sepsis progression with EHRs is particularly challenging. Specifically, whether a patient has sepsis is not directly observable, and its symptoms are often hidden by medical "expert blind spots" [Tin85]. Moreover, different patient groups may show diverse symptoms. For example, although one common sign of sepsis is fever, for young, old, or immune system-weakened patients, their body temperature may be low or normal when sepsis is present. Thus, our motivation is to explore *whether the sepsis progression modeling can be automated*. So far, the DPM of sepsis is generally modeled by supervised learning [Fle20], which relies on a large amount of *fine-grained moment-by-moment labeled data*. Such labeled data are *not only* time and expertise intensive, *but also* often infeasible to be acquired [Giu07; Sin16].

In this chapter, we utilize an unsupervised learning approach to automatically model the sepsis progression, which is formulated as a *sub-trajectory partitioning* problem to partition and cluster the sub-trajectories in EHRs simultaneously. More importantly, we expect the discovered sub-trajectory clusters to be interpretable because, in healthcare domains, it is usually more essential to learn discriminative and interpretable patterns that reflect the disease progression than to merely induce a prediction model. Recently, Severson *et al.* employed a hidden Markov model (HMM)-based method for learning Parkinson's progression, which encoded prior knowledge to learn the latent states and then utilized posthoc analysis to interpret the states [Sev20]. In this chapter, we learn the interpretable DPM by leveraging a novel sub-trajectory clustering method, *i.e.*, Toeplitz Inverse Covariance-based Clustering (TICC) [Hal17]. TICC employs inverse covariance matrices and constrains these matrices to be block-wise Toeplitz to model the time-invariant structural patterns in each cluster. It has demonstrated better performance compared to both *distance-based* methods, *e.g.*, dynamic time warping (DTW) [Cut11] or rule-based motif discovery [Li12]), and *model-based* methods, *e.g.*, Gaussian mixture models (GMM) [Rey09] or hidden Markov models [Smy97].

When applying TICC to EHRs data, there are two major challenges: 1) TICC takes a single time series as input, whereas most EHRs consist of multiple time series as a collection of visits from different patients. Applying TICC to each visit independently may lead to inconsistent patterns across different visits, while concatenating all visits to be a single series may introduce some undesired patterns at the joints between adjacent visits. Therefore, in this chapter, we extend the TICC by considering multi-series inputs and refer to it as Multi-series TICC (M-TICC); 2) The records in EHRs are generally collected with *irregular time intervals*, varying from seconds to days. For example, the interval between two consecutive records in our EHRs ranges from 0.94 seconds to 28.19 hours. Hence, it is essential to consider irregular time intervals for capturing latent progressive patterns of a targeted disease [Bay17]. However, the TICC ignores the intervals and encourages the consecutive

records to be assigned to the same cluster. Consequently, we further extend M-TICC by incorporating time-awareness for the consistency between consecutive records, which is denoted as **M**ulti-series, **T**ime-aware TICC (**MT-TICC**).

The effectiveness of MT-TICC is first validated with a case study involving a real-world hand gesture dataset (sEMG) with moment-by-moment ground-truth labels. Like EHRs, sEMG is human-oriented with multi-series and irregular time intervals. Our results show that MT-TICC significantly outperforms the state-of-the-art models, including the TICC. Then we applied MT-TICC for sepsis progression modeling using the real-world EHRs. To evaluate the MT-TICC derived clusters, we incorporate them into the original EHRs for the task of septic shock early prediction. The results show significantly improved prediction performance compared to using original EHRs or using the clusters learned by TICC. Furthermore, the clusters derived by MT-TICC convey meaningful insights and shed some light on a better understanding of sepsis progression.

3.2 Related Work

3.2.1 Time-awareness in EHRs

Previous works have demonstrated time-aware mechanisms to be a critical factor when handling irregular time intervals in EHRs. Baytas *et al.* proposed a time-aware Long Short-Term Memory (LSTM) in an auto-encoder to learn representation for sequential records of patients, which are then utilized to cluster patients into clinical subtypes [Bay17]. Zhang *et al.* proposed to use attention-based time-aware LSTM networks for disease progression modeling, which can effectively improve the interpretability of LSTM and identify critical previous events for current diagnosis by modeling the inherent time irregularity [Zha19]. Pham *et al.* introduced an end-to-end deep dynamic neural network named DeepCare that introduced time parameterizations to handle irregular timed events by moderating the forgetting and consolidation of memory cells [Pha16]. The efficacy of DeepCare was demonstrated via disease progression modeling, intervention recommendation, and future risk prediction. The time-awareness has also demonstrated effectiveness when handling missing data in EHRs. Kim and Min proposed a temporal belief memory for handling missing data in EHRs with recurrent neural networks. It considers time continuity and captures latent missing patterns based on irregular real-time intervals of the inputs [Kim18]. Yang proposed a time-aware subgroup matrix decomposition [Yan18] to impute missing data and forecast future events. The time-awareness can effectively constrain the smoothness when filling in missing data and forecasting the subsequent data and can help to improve the performance in septic shock early prediction.

3.2.2 Sub-trajectory Partitioning

Some prior works have been developed for sub-trajectory partitioning, including *distance-based* methods and *model-based* methods. Examples of distance-based methods include dynamic time warping and rule-based motif discovery. Dynamic time warping (DTW) is a generalization of classical

clustering algorithms for measuring the similarity between two temporal trajectories. It is an efficient time-series similarity measurement, which minimizes the effects of shifting and distortion and allows elastic transformation of time series so that the similarity between two trajectories with different phases can be detected [Ber94]. However, it is generally employed for clustering the whole trajectory and cannot effectively detect the sub-trajectory partitions. The rule-based motif discovery can handle this problem: it will first leverage motif discovery to segment the trajectories precisely for seeking recurring patterns as antecedents or consequents, and then investigate the underlying temporal relationships between motifs by combing motifs as rule candidates and ranking them based on the similarities [Das98]. The *model-based* sub-trajectory partitioning methods include Gaussian mixture models and hidden Markov models [Smy97], *etc.* Gaussian Mixture Models (GMM) assume that there are a finite number of mixed Gaussian distributions, and each distribution represents a cluster. It can group the data belonging to the same distribution into one cluster [Rey09]. Compared to static clustering methods, *e.g.*, k-means, it can incorporate information about the covariance of data and the center of latent Gaussians. Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process [Gha14]. Pavlovic proposed a recursive EM algorithm for a mixture of Gaussian HMMs where each trajectory has a prior probability of being generated by the HMM [Pav04].

Compared to distance-based methods, model-based methods can introduce structures in the data, and therefore it is more reliable. Recently, Hallac *et al.* proposed a novel model-based Toeplitz Inverse Covariance-based Clustering (TICC) [Hal17], which employed inverse covariance matrices to model the patterns in each cluster and constrained the matrices to block Toeplitz. It can successfully capture time-invariant patterns and effectively prevent overfitting. More importantly, the discovered patterns for the learned clusters can be interpreted. TICC has shown great success in finding meaningful patterns in both traffic [Hal17] and healthcare [Li18] data.

3.3 Methodology

3.3.1 Toeplitz Inverse Covariance Clustering (TICC)

Given a dataset with N multivariate sequences, denoting the n-th sequence as $\{\mathbf{x}_1^n, \ldots, \mathbf{x}_{T^n}^n\}$, where $\mathbf{x}_t^n \in \mathbb{R}^m$ is the t-th event and T^n is the length of the sequence, we aim to simultaneously *partition and cluster* the events based on their latent patterns. Without loss of generality, supposing there are K clusters in the dataset, we will learn a mapping from each event $\mathbf{x}_t^n$ to a certain cluster $k \in \{1, \ldots, K\}$. Since the events in a sequence are consecutive and each event is dependent on its neighbors, instead of mapping each event independently, we investigate events in a sliding window $\omega \ll T^n$. For $\mathbf{x}_t^n$, its preceding events within ω, *i.e.*, $\mathbf{X}_t^n = \{\mathbf{x}_{t-\omega+1}^n, \ldots, \mathbf{x}_t^n\}$, are extracted for determining which cluster k the event $\mathbf{x}_t^n$ belongs to.

To learn the clustering mapping in an unsupervised manner, we treat each $\mathbf{X}_t^n$ as a $m\omega$-dimension random variable (obtained by concatenating the ω events in $\mathbf{X}_t^n$) and optimally fit all variables into K Gaussian distributions, with the k-th fitted distribution corresponding to the k-th cluster. Of note,

each $\mathbf{X}_t^n$ will only contribute (belong) to one distribution (cluster). The TICC [Hal17] characterizes each distribution by determining its mean and inverse covariance matrix. Specifically, for all K distributions, determining the optimal mean vectors $\{\mu_k | k = 1, \ldots, K\}$ is equivalent to matching each event to an optimal cluster, which leads to the clustering assignment results $\mathbf{P} = \{P_k | k = 1, \ldots, K\}$, where $P_k \subset \{1, \ldots, T^n\}$ denotes the indices of (sliding window) events belonging to the cluster k; Meanwhile, determining the optimal $\{\mathbf{\Theta}_k | k = 1, \ldots, K\}$ is to estimate K inverse covariance matrices with block-wise Toeplitz constraints. It is worth noting that the inverse covariance matrix is used rather than the covariance matrix because it models conditional dependencies and can easily introduce a graph structure during the matrix learning [Hal17], which can substantially decrease the number of parameters to reduce the risk of overfitting [Mei06]. Each $\mathbf{\Theta}_k$ for a cluster k is constrained to be block-wise Toeplitz, which composes of ω sub-blocks $A^{(i)} \in \mathbb{R}^{m \times m}$, $i \in [0, \omega - 1]$:

$$\mathbf{\Theta}_k = \begin{bmatrix} A^{(0)} & (A^{(1)})^\top & \cdots & & (A^{(\omega-1)})^\top \\ A^{(1)} & A^{(0)} & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \\ & & & & (A^{(1)})^\top \\ A^{(\omega-1)} & \cdots & & A^{(1)} & A^{(0)} \end{bmatrix}$$

The sub-block $A^{(i)}$ represents the partial correlations among m features between timestamps t and $t + i$. For example, the (p, q)-th entry in $A^{(i)}$ indicates the partial correlation between the p-th feature at t and the q-th feature at $t + i$, where $p, q \in \{1, \ldots, m\}$. The block-wise Toeplitz constraints enable $\mathbf{\Theta}_k$ to capture time-invariant structural patterns within $\mathbf{X}_t^n$.

3.3.2 M-TICC

The original TICC learns time-invariant structural patterns to simultaneously partition and cluster over a single sequence. To apply it to multi-series input, we can either treat each sequence independently or concatenate them as one sequence. However, both strategies are not optimal. When treating each sequence independently, the learned patterns across different sequences can be inconsistent due to sequence discrepancies, while concatenating all sequences will introduce undesired patterns due to the artificial joints between neighboring sequences. To handle this issue, we adapt the TICC for jointly partitioning and clustering across difference sequences to explicitly learn the shared patterns, which is denoted as *Multi-series* TICC (**M**-TICC), as formulated in Eq. 3.1. Note that when $N = 1$, Eq. 3.1 will degenerate into the original TICC.

$$\underset{\mathbf{\Theta}, \mathbf{P}}{\operatorname{argmin}} \sum_{k=1}^{K} \Bigg[\sum_{n=1}^{N} \sum_{\mathbf{X}_t^n \in P_k} \Big(\overbrace{-\ell\ell(\mathbf{X}_t^n, \mathbf{\Theta}_k)}^{\text{Log-likelihood}} + \overbrace{c(\mathbf{X}_{t-1}^n, P_k)}^{\text{Consistency}} \Big) + \lambda \overbrace{||\mathbf{\Theta}_k||_1}^{\text{Sparsity}} \Bigg] \tag{3.1}$$

The roles of three terms in Eq. 3.1 are detailed as follows:

• **Log likelihood term**: The log likelihood term represents the probability that $\mathbf{X}_t^n$ belongs to the cluster k. Assume that $\mathbf{X}_t^n \sim N(\mu_k, \mathbf{\Theta}_k^{-1})$, then the $\ell\ell(\mathbf{X}_t^n, \mathbf{\Theta}_k)$ can be calculated as:

$$\ell\ell(X_t^n, \mathbf{\Theta}_k) = -\frac{1}{2}(\mathbf{X}_t^n - \mu_k)^T \mathbf{\Theta}_k (\mathbf{X}_t^n - \mu_k) + \frac{1}{2}\log|\mathbf{\Theta}_k| - \frac{n}{2}\log(2\pi) \tag{3.2}$$

• **Consistency term**: The consistency term encourages the neighboring events $\{\mathbf{X}_{t-1}, \mathbf{X}_t\}$ to be assigned into the same cluster. It is defined as:

$$c(\mathbf{X}_{t-1}^n, P_k) = \beta \mathbb{1}\{t-1 \notin P_k\} \tag{3.3}$$

Herein, $\mathbb{1}\{t-1 \notin P_k\}$ is an indicator function, which is 1 if $\mathbf{X}_{t-1}^n$ does not belong to the same cluster as $\mathbf{X}_t^n$, otherwise it is 0. By minimizing Eq. 3.1, neighboring events belonging to different clusters will be penalized. β is a weight parameter.

• *Sparsity term* controls the sparseness of $\mathbf{\Theta}_k$ via a l_1-norm, which selects the most significant variables to represent the time-invariant structural patterns that can effectively prevent overfitting. λ is a sparsity regularization coefficient.

3.3.3 MT-TICC

Both M-TICC and TICC assume neighboring events having equal time intervals. However, the intervals between neighboring events can vary greatly, ranging from seconds to days in EHRs. Specifically, two events with a shorter interval would more likely belong to the same cluster compared to those with longer intervals. Thus, it is essential to consider the time interval irregularity in the consistency term. To address this problem, we incorporate *Time-awareness* into the consistency term to make it interval-dependent, which is denoted as **MT**-TICC. The modified objective function is presented in Eq. 3.4.

$$\underset{\mathbf{\Theta},\mathbf{P}}{\operatorname{argmin}} \sum_{i=1}^{K} \Big[\sum_{n=1}^{N} \sum_{\mathbf{X}_t^n \in \mathbf{P}_k} \Big(\overbrace{-\ell\ell(\mathbf{X}_t^n, \mathbf{\Theta}_k)}^{\text{log likelihood}} + \overbrace{c(\mathbf{X}_{t-1}^n, \mathbf{P}_k, \Delta T^n)}^{\text{time-aware consistency}} \Big) + \overbrace{\left\| \lambda \circ \mathbf{\Theta}_k \right\|_1}^{\text{sparsity}} \Big] \tag{3.4}$$

• *Time-aware consistency term* encourages the consecutive events $\{\mathbf{X}_{t-1}, \mathbf{X}_t\}$ with shorter time interval to be assigned into the same cluster, which is defined as:

$$c(\mathbf{X}_{t-1}^n, P_k, \Delta T_t^n) = \frac{\beta \mathbb{1}\{t-1 \notin P_k\}}{\log(e + \Delta T_t^n)} \tag{3.5}$$

Herein, we introduce a decay function, *i.e.*, $1/log(e + \Delta T_t^n)$, which can adaptively relax the penalization of the consistency constraint as the interval ΔT_t^n between neighboring events becomes larger [Bay17]. The nonlinear monotonically decreasing manner of the decay function enables us to control the impact of intervals over the consistency term.

To solve the objective function Eq. 3.4, we followed the expectation-maximization (EM) procedure to iteratively learn the cluster belongings $\mathbf{P}$ and the clusters' structural patterns $\mathbf{\Theta}$ until

convergence. In *E-step*, $\mathbf{\Theta}$ is fixed as a constant to learn the $\mathbf{P}$. The objective function degrades into the form with only log-likelihood and consistency terms. It can be solved by dynamic programming, which is equivalent to finding the minimum cost Viterbi path [Vit67] for each trajectory. To be specific, the cost for an event in the current timestamp belonging to a cluster is described by the negative log-likelihood, and the cost of transiting to a different cluster in the next timestamp is described by the consistency term. In *M-step*, $\mathbf{P}$ is fixed to learn the $\mathbf{\Theta}$. The objective function degrades into the form with only log-likelihood and sparsity terms. It can be modeled as a graphical lasso problem with the Toeplitz constraint over the $\mathbf{\Theta}$. We employed the alternating direction method of multipliers (ADMM) method to solve the inverse covariances [Hal17].

3.4 A Case Study

Since our EHRs have no moment-by-moment ground-truth labels, we first carried out MT-TICC in a case study to validate its effectiveness, using a real-world hand gesture dataset (sEMG). The sEMG dataset has similar properties to EHRs, *i.e.*, human-oriented consisting of trajectories with irregular intervals, but with moment-wise ground-truth labels.

3.4.1 Data Preprocessing

The dataset contains surface electromyographic (sEMG) signals for 36 participants collected when they are performing different hand gestures [Lob19]. Specifically, the multichannel sEMG signals were collected via eight sensors embedded in an MYO Thalmic bracelet worn on the participants' right forearm. Each participant performed a series of six or seven gestures and repeated the series twice, which yielded 72 trajectories in total. Each gesture lasted for 3 seconds with a pause of 3 seconds before the next gesture. We preprocessed the sEMG data with three steps:

• **Smoothing the signals**: The raw sEMG signals were recorded per millisecond, characterized by high volatility. Due to the insensitivity of sEMG sensors, the peaks and troughs in the recorded signals tend to be truncated [Voi20]. To obtain more stable data, a general approach is smoothing the signals by a sliding window [Lob18; Voi20]. Herein, we applied a $100ms$ window with a step size of $50ms$ following the settings in [Voi20]. Inside the sliding window, the recordings from each channel were smoothed by the root-mean-squared value [Lob18; Voi20].

• **Slicing and shuffling the gestures**: We first excluded the unlabeled events during the pause in between different gestures. Since the gesture *extends the palm* is performed by a few participants ($\sim$ 0.3%), it is excluded from the analysis, together with its counterpart gesture *clenched in the fist*. Then, five out of seven gestures were sliced for analysis, including 1) *hand at rest* (rest), 2) *wrist flexion* (left), 3) *wrist extension* (right), 4) *radial deviation* (up), and 5) *ulnar deviation* (down). Moreover, in the original sEMG, all participants followed the same order to perform the gestures. To enable the data to cover more complex scenarios for learning a more robust model, we randomly shuffled the order of the gestures. Note that the order of timestamps inside a gesture segment will not change. For each gesture segment, the time interval to previous gesture will be maintained during shuffling.

• **Selecting the features**: The sEMG signals were mainly contributed by four sensors out of the eight channels [Lob18], which monitored the electrodes of four muscles: 1) *flexor carpi radialis* (FR), 2) *extensor carpi radialis longus* (ER), 3) *extensor carpi ulnaris* (EU), and 4) *flexor carpi ulnaris* (FU). When performing different gestures, these four muscles can act either *synergistically* or *antagonistically*. The signals collected from these four muscles were taken as features and normalized to the range of $[0,1]$. Finally, our dataset consists of 72 trajectories with 14,441 events (timestamps). The intervals between consecutive events ranged from [50, 5450] milliseconds.

3.4.2 Experimental Setup

Based on the sEMG data, we evaluated the effectiveness of MT-TICC. Specifically, for the training data, we learned cluster assignment for each timestamp and compared it against the ground truth; For the test data, based on the learned clustering model, we followed Eq. 3.2 to calculate the probability belonging to each cluster and then assigned the data to the cluster with the maximal probability.

• **Baselines**: We compared our *MT-TICC* to 1) *M-TICC* which takes multi-series as input without time-awareness and 2) six other baselines including *TICC* which randomly concatenates all trajectories as a single input, *I-TICC* which treats each trajectory independently, *TICC(β=0)* which is a competitive baseline reflected in [Hal17] without the consistency term, model-based Gaussian mixture model (*GMM*) [Rey09] and a hidden Markov model using GMM for emissions (*GMM-HMM*) [Yan17], and a distance-based dynamic time warping (*DTW*) [Cut11].

• **Parameters**: All the model parameters were determined by Bayesian information criteria (BIC) [Fri01]. In MT-TICC, the cluster number K was 11; the window size ω was 2; the sparsity and consistency coefficients λ and β were 1e-5 and 4, respectively. For a fair comparison, the optimal parameters in other methods were determined by BIC as well.

• **Metrics**: The results were evaluated via 1) *classification metrics*: accuracy (Acc), recall (Rec), precision (Prec), and F1-score (F1), which treated the clustering as a multi-class classification to compare the results against the ground-truth labels. Specifically, the metrics were weighted by the respective size of each label; and 2) *clustering metrics*: normalized mutual information (NMI), adjusted random index (ARI), homogeneity score (Hom), and completeness score (Com). We repeated the 5-fold cross-validation ten times and conducted a corrected paired t-test [Nad03] to compare MT-TICC and M-TICC against the TICC.

• **Interpretation**: We also explored interpreting the MT-TICC-learned clusters, which are delineated by the inverse covariance matrices $\mathbf{\Theta}$. Specifically, we calculated the PageRank in $\mathbf{\Theta}$ to measure the importance of the features contributing to each cluster.

3.4.3 Results

We compared MT-TICC against other methods for both training data and test data first across the whole trajectories and then specifically around the *joints* connecting two sub-trajectories with different ground-truth labels. Since the events in a trajectory are consecutive and dependent on neighboring events, the switch of patterns around the joints is more challenging to be identified. To

Table 3.1 Clustering results in sEMG *Training Data* with (a) *overall* data and (b) data over *joints* connecting two sub-trajectories.

Method	Training Data							
	Acc	Rec	Prec	F1	NMI	ARI	Hom	Com
DTW	.208	.211	.229	.186	.002	.000	.002	.002
GMM	.556	.609	.584	.573	.387	.296	.369	.408
GMM-HMM	**.610**	**.610**	**.637**	**.616**	**.422**	**.318**	**.415**	**.429**
TICC($\beta = 0$)	.686	.708	.686	.677	.453	.405	.446	.461
I-TICC	**.845**	**.912**	**.878**	**.879**	**.810**	**.750**	**.781**	**.845**
TICC	.696	.721	.696	.687	.475	.423	.467	.484
M-TICC	.710	.735	.710*	.702	.491	.442†	.484†	.499*
MT-TICC	**.728***	**.747***	**.728***	**.721***	**.512***	**.473***	**.505***	**.519***

(a) Clustering results for **overall** data

Method	Training Data							
	Acc	Rec	Prec	F1	NMI	ARI	Hom	Com
TICC	.549	.618	.549	.527	.297	.224	.281	.315
M-TICC	.570	.634	.570*	.552	.318	.246†	.303†	.335*
MT-TICC	**.602***	**.647***	**.602***	**.584***	**.344***	**.294***	**.332***	**.357***

(b) Clustering results for data over **joints** connecting two sub-trajectories

$\star$ denotes p-value < 0.01 and † denotes p-value < 0.05 when comparing M-TICC and MT-TICC against TICC, respectively. The best results in test data are in bold.

Table 3.2 Clustering results in sEMG *Test Data* with (a) *overall* data and (b) data over *joints* connecting two sub-trajectories.

Method	Test Data							
	Acc	Rec	Prec	F1	NMI	ARI	Hom	Com
DTW	.212	.213	.322	.232	.016	.000	.004	.017
GMM	.563	.621	.674	.628	.430	.316	.399	.468
GMM-HMM	**.614**	**.642**	**.660**	**.635**	**.447**	**.373**	**.444**	**.450**
TICC($\beta = 0$)	.674	.694	.674	.664	.444	.393	.436	.451
I-TICC	.554	.554	.558	.517	.399	.318	.374	.430
TICC	**.677**	**.698**	**.677**	**.668**	**.451**	**.398**	**.444**	**.459**
M-TICC	.685	.709	.685†	.677	.458	.408	.450	.465
MT-TICC	**.699**†	**.721***	**.699***	**.690***	**.474***	**.430***	**.466***	**.481***

(a) Clustering results for **overall** data

Method	Test Data							
	Acc	Rec	Prec	F1	NMI	ARI	Hom	Com
TICC	.519	.621	.519	.499	.288	.213	.273	.304
M-TICC	.533	.648	.533*	.516	.316	.195	.292	.316
MT-TICC	**.571***	**.656***	**.571***	**.557***	**.338***	**.242***	**.321***	**.357***

(b) Clustering results for data over **joints** connecting two sub-trajectories

$\star$ denotes p-value < 0.01 and † denotes p-value < 0.05 when comparing M-TICC and MT-TICC against TICC, respectively. The best results in test data are in bold.

Table 3.3 The PageRank of the EMGs features (FR, ER, EU, and FU) and their functionalities of Synergistic (S) or Antagonistic (A).

Idx	Cluster Label	PageRank				Functionalities [Lob18]			
		FR	ER	EU	FU	FR	ER	EU	FU
1	rest	.55	.12	.12	.20	-	-	-	-
2	left	.01	.63	.22	.14	S	A	A	S
3		.01	.67	.29	.01				
4		.22	.22	.32	.24				
5		.01	.97	.01	.01				
6	right	.80	.02	.08	.10	A	S	S	A
7		.88	.01	.01	.10				
8	up	.45	.01	.06	.47	S	S	A	A
9		.01	.01	.06	.92				
10	down	.97	.01	.01	.01	A	A	S	S
11		.78	.18	.02	.03				

- The highlighted numbers show the most important feature learned by the MT-TICC model for each cluster.

do so, we defined a *tolerance window* (tol) around the joints and evaluated the results within the tol. In this study, tol is set as 5, with 39.4% of the overall data counted as joints.

Table 3.1(a) and Table 3.2(a) report the results of different methods for *overall training* data and *test* data, respectively. Among the six baselines, I-TICC performs the best in training data while the second worst in test data. It might be because the individual trajectory cannot provide adequate information to cover all variations across different trajectories. Taking together both training and test data, TICC performed the best among the baselines. When comparing M-TICC and MT-TICC against TICC: $\star$ denotes p-value < 0.01 and † denotes p-value < 0.05. The results demonstrated that both MT-TICC and M-TICC outperformed the TICC by taking multi-series as input. Furthermore, equipped with time-awareness, MT-TICC performed the best among all methods. Specifically, compared to TICC, MT-TICC improved by $\sim 3\%$ and $\sim 2\%$ in training and test data, respectively.

Table 3.1(b) and Table 3.2(b) show the results for *training* data and *test* data over *joints* within the tol. Herein, we focused on comparing the best baseline, *i.e.*, TICC. As expected, clustering over joints is much harder compared to overall data. The improvement of MT-TICC versus TICC was $\sim 6\%$ in training data and $\sim 5\%$ in test data. The results suggested that the time-awareness of MT-TICC could effectively capture the switch of sub-trajectory clusters.

Table 3.3 shows the PageRanks of different features in each cluster, with the highest PageRank highlighted in shade. Referring to [Lob18], the functionalities of the four muscles can be either synergistic (*S*) or antagonistic (*A*) when performing different gestures. For example, in gesture *left*, the ER and EU were *A* while FR and FU were *S*. Based on the PageRanks calculated from our model, we can also infer which muscle works in the first place in *S* or *A* functionalities: For example, ER plays a more important role than EU in *left*.

Table 3.4 Missing rate for each feature and their average.

Category	Feature	Missing Rate (%)
Vital Signs	Systolic Blood Pressure (SBP)	63.91
	Mean Arterial Pressure (MAP)	68.86
	Respiratory Rate (RR)	56.10
	Pulse Oxygen saturation (PulOx)	58.26
	Heart Rate (HR)	53.41
	Temperature (Temp)	73.19
	Fraction of inspired Oxygen (FiO2)	84.76
Lab Values	White Blood Cell count (WBC)	93.07
	Bilirubin(Bili)	97.99
	Blood Urea Nitrogen (BUN)	92.84
	Lactate (Lac)	97.83
	Creatinine (Create)	92.84
	Platelet (Plat)	93.07
	Band cells of neutrophils (Bands)	99.08
Average		**80.37**

3.5 Experiments with Healthcare Data

The EHRs applied in this chapter were extracted from the Christiana Care Health System (CCHS) from July 2013 to December 2015. Each trajectory is a visit that consists of a series of temporal events. In total, the dataset contains 210,289 visits with 10,412,729 events. Since the moment-by-moment status for sepsis is unknown, when applying MT-TICC to learn the cluster belongings for each event, there were no available labels to directly evaluate the performance. Instead, when referring to the ICD-9 code and the domain experts' rules, we can tag the septic shock for each trajectory. As a result, in the experiment, we took MT-TICC-learned patterns as additional features and evaluated the performance via a septic shock early prediction task. Besides, based on the MT-TICC learned clusters, we referred to some criteria provided by experts and got some interesting insights.

3.5.1 Data Preprocessing

In our dataset, 52,919 visits with 4,224,567 events have suspected infection, which was defined as the sepsis-related study cohort. Note that the rules for identifying the suspected infection and for tagging the septic shock were determined by two leading clinicians with extensive experience from Mayo Clinic and CCHS. The selected cohort was preprocessed with the following steps:

• **Selecting features**: We selected 14 features related to the sepsis progression as suggested by the clinicians: 1) *Vital signs*: systolic blood pressure (SBP), mean arterial pressure (MAP), respiratory rate (RR), oxygen saturation (PulOx), heart rate (HR), temperature (Temp), the fraction of inspired Oxygen (FiO2); and 2) *Lab results*: white blood cell count (WBC), bilirubin (Bili), blood urea nitrogen (BUN), lactate (Lac), creatinine (Creat), platelet (Plat), neutrophils (Bands).

• **Handling missing data**: The events in each visit were collected at irregular time intervals, which ranged from 0.94 seconds to 28.19 hours. The missingness in EHRs is generally caused by the different recording frequencies among the features. For instance, vital signs are usually measured

Table 3.5 Intersection of ICD-9 code and experts' rules for tagging the septic shock visits. The shock and non-shock visits were balanced by stratified sampling.

		Experts' Rules	
		Shock (+)	Non-shock (-)
ICD-9	Shock (+)	1,869	
	Non-shock (-)		23,901 → 1,869

every 8 hours, while lab values are measured every 24 hours. Thus, there may not be available readings for lab results when a new record is created for the vital signs. On average, the missing rate for our data is 80.37%, as illustrated in Table 3.4. We handled the missing values by carrying forward, *i.e.*, filling the missing entries as the last observation until the next observed value, with the remaining missingness filled with the mean value for each feature.

• **Tagging the septic shock visits**: Identifying the septic shock and non-shock visits is a challenging task. Though the diagnosis codes, such as the International Classification of Diseases, Ninth Revision (ICD-9), are widely used for clinical labeling, solely relying on the ICD-9 can be problematic: the ICD-9 has been proven to have limited reliability since its coding practice is mainly utilized for administrative and billing purpose. Based on the Third International Consensus Definitions for Sepsis and Septic Shock [Sin16], our domain experts identified septic shock when *any* of the following two conditions were met:

- Persistent hypertension through two consecutive readings ($\leq$ 30 minutes apart), including
  1) SBP $<$ 90 mmHg;
  2) MAP $<$ 65mmHg; and
  3) Decrease in SBP $\geq$ 40 mmHg within an 8-hour period; or
- Any vasopressor administration.

By combining both ICD-9 and the domain experts' rules, we identified 1,869 shock visits and 23,901 non-shock visits, as shown in Table 3.5. Considering the highly imbalanced ratio between shock and non-shock visits, we conducted a stratified random sampling on non-shock visits while keeping the same underlying distribution of age, sex, ethnicity, and duration of stay. Finally, the dataset was narrowed down to 3,738 visits (1,869 shocks and 1,869 non-shocks) with 145,421 events.

3.5.2 Experimental Setup

To evaluate the effectiveness of our MT-TICC, we first conducted an early prediction task by taking MT-TICC-derived patterns as additional features and compared them against competitive baselines. Then we further explored ways of interpreting and differentiating the clusters learned by MT-TICC.

3.5.2.1 Task I: Septic shock early prediction

Herein, our goal is to predict septic shock as early as possible, which is defined as: given the observation of a patient's visit until τ hours before an endpoint, we will predict whether or not the

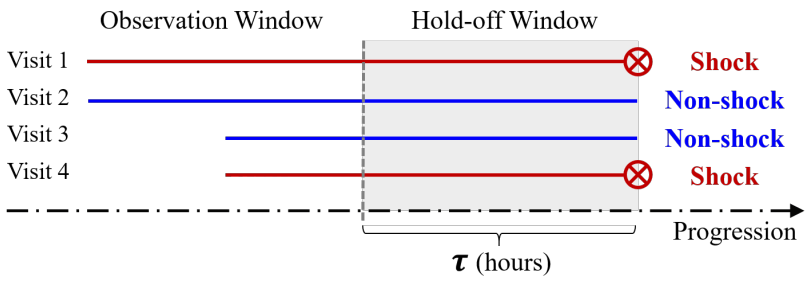


Figure 3.1 Illustration of septic shock early prediction.

visit will develop into septic shock τ hours later. For septic shock visits, the endpoint is the onset time of septic shock while for non-shock visits, the endpoint is the end of sequences. As shown in Figure 3.1, the τ-hour window leading up to the endpoint is denoted as *hold-off window*.

We employed long short-term memory (LSTM) as the prediction model since extensive previous works have demonstrated its preferable performance in EHRs modeling [Lip15; Bay17; Soh20a]. Different inputs to the LSTM were compared, including 14 **O**riginal features **(O)** vs. original features with additional **C**luster-based features **(O+C)** learned from three TICC-based methods, *i.e.*, *TICC*, *M-TICC*, and *MT-TICC*. In MT-TICC, the cluster number K was determined as 6 via BIC; therefore, six additional features were generated for each event, which measures the probabilities of the event belonging to each cluster based on Eq. 3.2. For test data, the additional features were generated based on the clustering models learned from training data. BIC determined the K for M-TICC and TICC as 7 and 9, respectively. The other parameters involved in the three TICC-based methods were tuned based on BIC as well: for example, in MT-TICC, the window size ω was 3; the sparsity and consistency coefficients λ and β were 1e-8 and 10, respectively. We implemented LSTM with Keras and tuned the parameters by grid search. All models were evaluated by repeating the 3-fold cross-validation ten times. The results were compared over $\tau \in [12,24)$ and $\tau \in [24,36]$ to validate whether septic shock can be predicted at least 12 hours or a day in advance. The metrics of Acc, Rec, Prec, F1, and AUC were employed for evaluation.

3.5.2.2 Task II: Interpretation for MT-TICC patterns

To further analyze the structural patterns of each cluster learned by our MT-TICC, we first inspected how each feature deviates from its normal range per cluster. By doing so and referring to experts' defined rules, we could gain interpretations of what the disease progressive stage that each cluster stands for. Besides, we also calculated the PageRank of the $\boldsymbol{\Theta}$ to measure the importance of features contributing to each cluster and analyzed whether the missing rates of features will affect their importance. Additionally, based on our interpretations of the clusters, we ranked their severeness and calculated the transition probabilities among them. The transitional patterns could be further analyzed over the shock vs. non-shock patients to better understand the disease progression.

Table 3.6 Early prediction using original features (O) with additional TICC-learned cluster-based features (O+C) when $\tau \in [12,24)$ and $\tau \in [24,36]$.

Features		Hold-off Window $\tau \in [12,24)$				
		Acc	Rec	Prec	F1	AUC
(O)	Original	.743(.016)	.728(.022)	.753(.022)	.739(.015)	.813(.015)
(O+C)	TICC	.757(.012)	.748(.016)	.765(.021)	.755(.010)	.825(.014)
	M-TICC	.772(.011)*	.767(.020)†	.778(.011)	.771(.013)*	.837(.010)
	MT-TICC	**.790(.010)***	**.806(.025)**†	**.784(.015)**	**.793(.011)***	**.852(.012)**†

Features		Hold-off Window $\tau \in [24,36]$				
		Acc	Rec	Prec	F1	AUC
(O)	Original	.704(.012)	.701(.020)	.709(.017)	.703(.013)	.778(.018)
(O+C)	TICC	.713(.013)	.717(.025)	.715(.015)	.714(.015)	.783(.018)
	M-TICC	.730(.014)*	.735(.022)†	.732(.016)†	.731(.015)*	.797(.017)*
	MT-TICC	**.761(.010)***	**.773(.023)**†	**.758(.023)**†	**.763(.007)***	**.825(.011)***

$\star$ denotes p-value < 0.01 and † denotes p-value < 0.05 when comparing M-TICC and MT-TICC to TICC. The best results are in bold.

3.5.3 Results

3.5.3.1 Task I: Septic shock early prediction

The early prediction results are shown in Table 3.6, which suggested the effectiveness of (O+C) features: (O+C) learned with the three TICC-based methods all outperformed the (O)-only. Especially, for the recall, MT-TICC improved by $\sim 8\%$ and $\sim 7\%$ compared to (O) when $\tau \in [12,24)$ and $\tau \in [24,36]$, respectively. When comparing M-TICC and MT-TICC to TICC: $\star$ denotes the p-value < 0.01 and † indicates the p-value < 0.05. The results showed that the multi-series input can effectively learn the shared patterns across different sequences since M-TICC outperformed the TICC, and the time-awareness is effective in modeling irregular intervals since MT-TICC performed better than M-TICC. Equipped with both multi-series and time-awareness, MT-TICC performed the best.

We further visualized the F1 score and AUC when varying τ from 1 hour to 36 hours before the septic shock onset, as shown in Figure 3.2. As τ increases, it becomes harder for early prediction across all models. The figures showed the advantage of (O+C) learned by three TICC-based methods compared to the Original (O)-only. Especially, it is demonstrated that the MT-TICC performed the best. When τ is larger, the gaps between MT-TICC and other methods are more apparent.

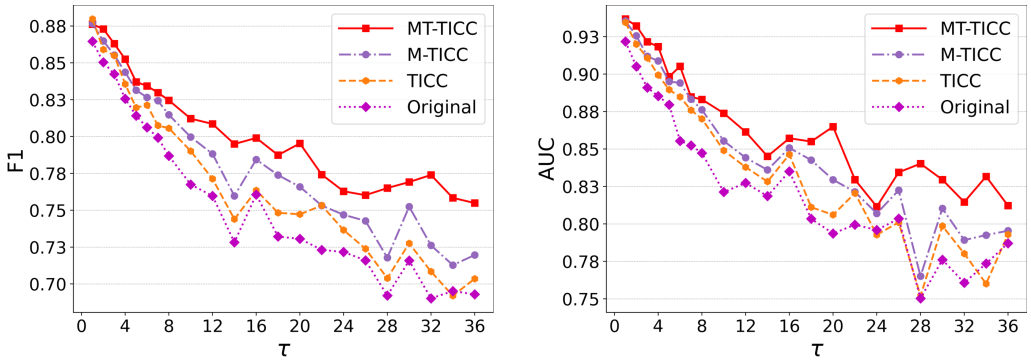


Figure 3.2 Early prediction F1 & AUC given original features with additional features learned by different TICC-based methods when $\tau \in [1,36]$.

Table 3.7 Interpretations of the MT-TICC-learned clusters.

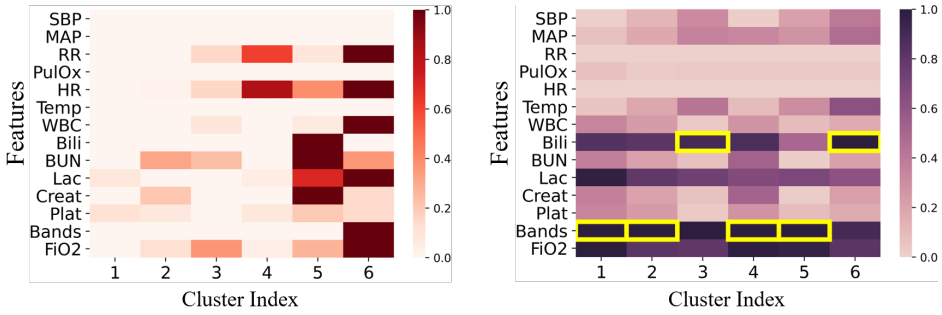
Cluster Idx	Interpretations
1	Metabolic Dysfunction
2	Renal Dysfunction
3	Non-temperature Physiological Response, Cellular Response, Renal Dysfunction
4	Non-temperature Physiological Response, Metabolic Dysfunction
5	Non-temperature Physiological Response, Metabolic Dysfunction, Renal Dysfunction, Gastrointestinal Dysfunction
6	Non-temperature Physiological Response, Cellular Response, Metabolic Dysfunction, Renal Dysfunction

3.5.3.2 Task II: Interpretation for MT-TICC learned patterns

For each cluster learned by MT-TICC, we calculated the mean value of each feature: if the value is abnormal, we measured its deviation from the normal range. As shown in Figure 3.3(a): the darker the color, the more abnormal the feature is. We ranked the 6 clusters from the least severe (*Cluster_1*) to the most severe (*Cluster_6*) based on the deviations. Referring to the criteria suggested by our domain experts, we obtained interpretations for each cluster as shown in Table 3.7. The clusters reflected the complications that could happen simultaneously.

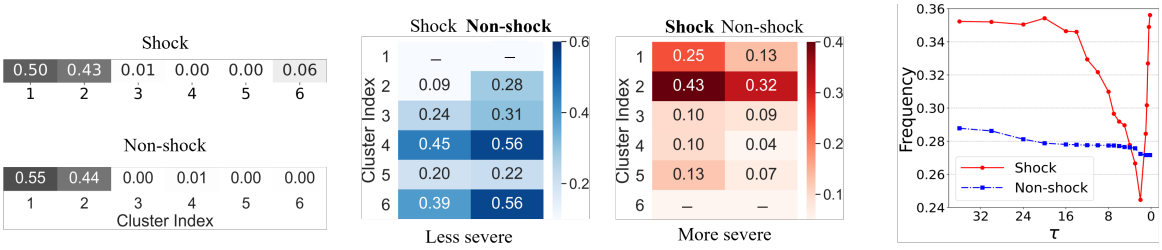
The missing rates for the features in each cluster are visualized in Figure 3.3(b): the darker the color, the higher the missing rate. Then we analyzed how the missing rates are related to the patterns learned by MT-TICC. Since the structural patterns $\mathbf{\Theta}$ can be represented as graphs, we calculated the PageRanks [Ber05] to measure the importance of features in each cluster. The features with the maximum PageRanks are highlighted with yellow boxes in Figure 3.3(b). An interesting finding is that the features with the maximum PageRanks usually have the highest missing rates, which indicates the MT-TICC has similar effects with missing indicators [Lip16] to capture structural patterns.

We further analyzed the transitions between clusters. Figure 3.4(a) indicates that the initial probabilities of shock and non-shock visits are quite similar. Figure 3.4(b) displays the probabilities of transiting to *Less* severe or *More* severe clusters. For example, in *Cluster_2*: for a shock visit, the probability to transit to less severity (*Cluster_1*) is 0.09, while for the non-shock visit, the probability is 0.28; Meanwhile, for a shock visit, the probability of transiting to more severity (*Cluster_3*, *Cluster_4*, *Cluster_5*, and *Cluster_6*) is 0.43, while for the non-shock visit, the probability is 0.32. In general,



(a) Deviations from normal ranges (b) Missing rate

Figure 3.3 Analysis of the MT-TICC-learned structural patterns.



(a) Initial Probabilities (b) Transition probabilities to less/more severe clusters (c) Transition frequency

Figure 3.4 Transitions between the MT-TICC-learned clusters.

non-shock visits have higher probabilities of transiting to less severity and lower probabilities of transiting to more severity compared to shock counterparts. Figure 3.4(c) shows the transition frequencies when sepsis progresses, with τ decreasing from 36 hours to 0.2 hours. The non-shock visits are stable when τ varies. In contrast, the shock visits have high frequencies at the beginning; when τ decreases, the visits are more likely to turn into the severe cluster and hardly get out, thus the transition frequencies decrease; the frequencies surge in 2 hours before the onset of septic shock, which possibly arises from augmented clinical interventions. Thus, MT-TICC can effectively cluster sub-trajectories and capture the differences in cluster transitions between shock and non-shock visits.

3.6 Summary

In this chapter, we proposed a method named **MT-TICC** to learn **sub-trajectory partitioning** in a healthcare application for sepsis progression modeling using EHRs. Compared to the original TICC, we made two improvements, including 1) making the objective function more general to handle the multi-series input, and 2) introducing the time-awareness in the consistency constraint. We first validated its effectiveness in a case study with the ground truth. Afterward, we applied MT-TICC to EHRs for learning the structural patterns and taking them as additional features to conduct a septic

shock early prediction task. The results showed the patterns derived from MT-TICC can effectively improve the prediction performance. Besides, the learned clusters conveyed interpretable insights which could help clinicians gain a better understanding of the sepsis progression.

4

EM-IRL: ACROSS-HETEROGENEOUS REWARD FUNCTIONS

* *This chapter has been published as*:

> **X. Yang**, G. Zhou, M. Taub, R. Azevedo, and M. Chi, “Student Subtyping via EM-Inverse Reinforcement Learning." In *International Educational Data Mining Society (EDM)*, 2020.

In this chapter, we focus on learning **across-heterogeneous** reward functions in an *education* application to infer the students' various learning intentions when interacting with an intelligent tutoring system. In learning science, heterogeneity among students usually leads to different learning strategies and may require different types of instructional interventions. Therefore, it is important to investigate student subtyping, which is to group students into subtypes based on their learning patterns. Subtyping from complex student learning processes is often challenging because of the information heterogeneity and temporal dynamics. Various inverse reinforcement learning (IRL) algorithms have been successfully employed in many domains for inducing policies from the trajectories and recently have been applied for analyzing students' temporal logs to identify their mastery of different learning skills. Traditionally, IRL was designed to model the data by assuming that all trajectories have *homogeneous* patterns, with a constant decision-making strategy. However, due to the heterogeneity among students, their decision-making strategies can vary greatly and the design of traditional IRL may lead to suboptimal performance. In this chapter, we applied a novel Expectation Maximization IRL (**EM-IRL**) to extract heterogeneous learning strategies from sequential data collected from three simulation environments and a real-world scenario with longitudinal students' logs. Experiments on simulation environments showed that EM-IRL can successfully identify the heterogeneity among sequences with different decision-making strategies. Furthermore, experimental results from our educational dataset showed that EM-IRL can be used to obtain different student subtypes: a *“learning-oriented"* subtype who learned the material as much as possible regardless of the time, *i.e.*, the students spent significantly more time than the other two subtypes and learned significantly; an *“efficient-oriented"* subtype who learned efficiently, *i.e.*, the students not only learned significantly but also spent less time than “learning-oriented"; a *“no learning"* subtype who spent less amount of time than “learning-oriented" and failed to learn.

4.1 Overview

With the rapid development of educational technologies, longitudinal students' learning trajectories are increasingly available. However, analyzing large-scale heterogeneous trajectories to infer high-level information embedded in student subgroups is often challenging. This challenge motivates the development of student subtyping [Cor94; Ale02; Gra04; GE07] for adaptive instructional interventions [Igl09b; Raf16a; Chi11b; Zho17b]. Student subtyping is to group students with similar learning progressions, which is crucial to personalize instructions so that the students can be provided with tailored interventions based on their unique learning status. From the data mining perspective, student subtyping is posed as an unsupervised clustering task of grouping students according to their historical records. Since these records are longitudinal and interrelated, it is important to capture the dependencies among the elements of the recorded sequence to learn more effective and robust representations, which can be utilized in the clustering stage to obtain the student subgroups.

In this chapter, we aim at investigating *student subtyping* based on their pedagogical strategies, which can be seen as a process of self-regulated learning [Aze19; Win98; Win12; Gre07; Wei00] by setting one's learning goals and ensuring the goals are attained. Specifically, we focus on students' pedagogical decision-making strategies during their interactions with an intelligent tutoring system (ITS) to learn probability. In this ITS, once a problem is presented, the students will decide whether they want the ITS to directly *tell* them how to solve the whole problem or to provide step-wise solutions, by presenting them with worked examples, or they want the ITS to *elicit* this problem or next step through problem-solving. When making pedagogical decisions, the students have to self-regulate their learning process, which may change the learning outcomes even though the instructional content is controlled. We believe that students' pedagogical strategies are closely related to metacognition, *i.e.*, the processes involved in thinking about thinking [Tau17].

Reinforcement learning (RL) offers one of the most promising approaches to induce effective pedagogical strategies directly from data. Some researchers have studied applying RL to improve the effectiveness of ITSs, *e.g.* [Chi11a; Chi11b; Dor15; Koe13; Man14; Row15; She16a; Zho17b; Zho19; Zho20], and much of the prior work focused on inducing effective policies that determine the best action for the ***ITS*** to take in any given situation to maximize the cumulative rewards, which is often the student learning gain. However, in this chapter, our goal is to infer the ***students'*** pedagogical strategies based on their behaviors and decisions while interacting with the ITS. To do so, we applied *inverse RL (IRL)*. Unlike traditional RL, where the reward function is explicitly given, IRL takes a bunch of trajectories as input and from which a reward function will be inferred. Given this inferred reward function, we can further employ it to induce the decision-making policy by RL. Since the students' decisions are generally made based on a trade-off among various complex factors, *e.g.*, time, learning gain, the difficulty of problems, *etc.*, merely taking the learning gain as the reward cannot reflect the actual decision-making strategies. As a result, we employed IRL to learn about students' strategies based on their behavioral data.

Recently, IRL has been widely employed in various domains to understand how decisions are

made in the given trajectories [Zie08; Aso13]. Specifically, it has been employed in educational domains to analyze students' temporal log data to identify their mastery of different learning skills [Raf15; Raf16b]. However, IRL was originally designed to model the data by assuming that all trajectories share a *homogeneous* pattern or strategy. Considering the heterogeneity among students, their pedagogical strategies can vary greatly and the design of traditional IRL may lead to suboptimal performance. To model such heterogeneity, though we can apply IRL individually for each student, it will forfeit our goal of revealing general and meaningful patterns from students' trajectories in consideration of the heterogeneity among subgroups of students.

To better capture the heterogeneity among student subtypes, we employed a novel Expectation-Maximization IRL (EM-IRL) algorithm [Bab11] by assuming that different student subtypes have different pedagogical strategies and that students within each subtype share the same strategy. The EM-IRL would recursively cluster students into different subgroups and induce a policy for each group by IRL until both clusters and policies converge. The original EM-IRL requires the number of clusters to be pre-defined [Bab11]. However, when applying it to student subtyping in education, it is often hard to figure out beforehand how many types of strategies are involved in students' trajectories. Therefore, we embedded the original EM-IRL into a general framework that can automatically determine the optimal number of clusters from the data.

We evaluated our framework first over three simulation environments: Grid World, Highway, and Mountain Car. The results showed that EM-IRL could accurately cluster the data with different decision-making strategies. Then we further applied it to real-world longitudinal students' logs collected from an ITS. The experiments showed that EM-IRL could be easily employed to obtain the student subtypes. Specifically, we got three student subtypes: a *"learning-oriented"* subtype who tries to learn the material as much as possible regardless of the time spent and learned significantly from pre- to post-test; an *"efficient-oriented "* subtype who learns efficiently in that they not only learned significantly but also spent significantly less time than the first subtype; a *"no learning"* subtype who spent less time and failed to learn. The clustering results suggested the potential of targeting the students who are not using effective pedagogical strategies, adapting the interventions, and offering the students effective pedagogical skill training through the ITS.

4.2 Related Work

4.2.1 Student Subtyping

Previous research has widely explored the modeling of student subtyping to assist teachers in providing more targeted interventions at the right time. Generally, student subtyping was analyzed via unsupervised clustering methods. For example, Lopez *et al.* employed an expectation maximization clustering method to determine if students' participation in the course Moodle forum could be a good predictor of the final marks [Lop12]. Durairaj and Vijitha applied K-means clustering to predict the pass/fail percentage of the students who appeared for a particular examination [Dur14]. Khalil and Ebner clustered the students into appropriate categories based on their level of engage-

ment [Kha17] so that the teachers could increase retention and improve interventions for specific sub-population. All of these methods were based on *static* data, without considering the *dynamic* properties during the learning process.

With the rapid development of e-learning, an increasing amount of sequential data was collected via ITSs. In general, the clustering methods to handle sequential data could be generalized into three categories: proximity-based, feature-based, and model-based [Li06]. Specifically, 1) proximity-based methods measure the similarity between the pair-wise data via the distance calculated by the longest common subsequence, dynamic time warping, *etc*. For example, Shen and Chi proposed a temporal clustering framework that measured pair-wise distance between the students by dynamic time warping and then clustered them by hierarchical clustering [She17]. Their method identified some distinctive patterns among the clusters, which could provide benefits to personalized learning. 2) Feature-based approaches would first compress the sequential data to be static, then the clustering methods taking static data as input could be further employed. For example, in [Ame06] and [Ame09], the authors aggregated the students' activities to a feature vector and then applied K-means clustering to recognize learner groups in exploratory learning environments. 3) In the model-based methods, the similarity of two data could be calculated based on the likelihood of one of them given the model derived from the other. For example, Li and Yoo proposed the use of a Markov chain-based clustering methodology to model the students' online learning behaviors collected during the learning process for more effective and adaptive teaching [Li06]. Additionally, Kock and Paramythis proposed a method combining K-means clustering with discrete Markov models to identify new, semantically meaningful problem-solving styles of the learners [Köc11].

4.2.2 Students Pedagogical Strategies

A number of researchers have investigated students' pedagogical decision-making [Ale00; Rol14; Mit03; Lon16; Zho16; Zho17a]. Previous research has shown that students make pedagogical decisions strategically. For example, Aleven *et al.* conducted a study to investigate students' hint usage behavior [Ale00]. Results showed that students used the easy-to-apply intelligent help more often than the Glossary. However, students often waited long before asking for a hint. When requesting hints, they often skipped the intermediate hints to reach the bottom-out hint which showed the solution directly. The results suggested that students preferred less effort-taking help (intelligent help and bottom-out hint), and oftentimes, they used the help less than they needed.

Additionally, prior research showed that providing students with pedagogical decision-making assistance could result in better decision-making skills or learning performance. Roll *et al.* [Rol14] examined the relationship between students' help-seeking patterns and learning performance. They found that asking for help on challenging steps was generally productive while help-abusing behaviors were correlated with poor learning. Mitrovic *et al.* [Mit03] compared three types of decision-making modes: system control, student control, and faded control. Under faded control, the system selected the problem for the student at the beginning of the training and gave explanations of why the problems should be selected. As the training proceeded, the control right was given to the

students. Results showed that the faded control group demonstrated improved problem selection skills and achieved better learning gains than the other two groups. Long *et al.* [Lon16] compared an assistance condition for selecting problems vs. a standard condition (no assistance). Their results showed the assistance condition achieved significantly better learning performance and better declarative knowledge of a key problem-selection strategy compared to the standard condition.

4.2.3 AL with Across-heterogeneous Reward Functions

Some Apprenticeship Learning (AL) algorithms have been proposed to learn the decision-making policies with *across*-heterogeneous reward functions. Dimitrakakis and Rothkopf proposed a Bayesian multi-task IRL [Dim11]. To model the heterogeneity in reward functions, they formalized the problem as a statistical preference elicitation, via a joint reward-policy prior. Choi and Kim proposed to integrate a Dirichlet process mixture model into Bayesian IRL to cluster the demonstrations [Cho12]: based on a Bayesian approach, the domain knowledge of different reward functions can be incorporated during learning. Note that the above two methods are both generalized from the original Bayesian IRL [Ram07]. Babes *et al.* proposed a more flexible EM-based IRL (EM-IRL) framework, which clusters the given demonstrations considering their heterogeneous reward functions via an EM algorithm [Bab11]. Specifically, in the E-step, given the initialized parameters for reward functions, EM-IRL will learn the cluster belongings across different demonstrations; Then in the M-step, it will infer the reward function and induce the policy for each cluster via IRL. The E-step and M-step will be conducted interactively until convergence. Finally, the EM-IRL will output the demonstration clusters with their respective policies. The flexibility of the EM-IRL framework lies in the fact that the M-step can use any AL algorithm. Herein, [Bab11] employed an offline maximum-likelihood IRL (MLIRL) in the M-step, which uses a gradient ascent method to optimize the reward parameters.

4.3 Methodology

In this work, we adapted the EM-IRL [Bab11] to model the across-heterogeneous reward functions. To figure out a suitable IRL for EM-IRL in the M-step, we compared some commonly utilized IRL methods including: maximum margin IRL [Abb04b], maximum entropy IRL [Zie08], Bayesian IRL [Ram07], and maximum likelihood IRL (MLIRL) [Bab11] over three simulation environments (*i.e.*, Grid World, Highway, and Mountain Car). The results showed MLIRL always outperformed others and it was also more efficient. Thus, we took MLIRL for the IRL-based analysis in this chapter.

4.3.1 Maximum Likelihood IRL

To formally define the MLIRL, we denote the input N demonstrated trajectories as $\mathcal{T} = \{\xi_1, ..., \xi_N\}$ and each trajectory is composed of a set of state-action pairs: $\xi_i = \{(s_1, a_1), (s_2, a_2), ...\}$. The reward function is defined as the linear function of a feature vector for state-action pairs: $r_\theta(s,a) = \theta^T \phi(s,a)$.

Algorithm 2 MLIRL

1: Input: $MDP\backslash R$, trajectories $\mathscr{T}$, trajectories' weights ω_i, $i = 1, ..., N$, learning rate α
2: Initialize: Reward parameter θ
3: **repeat**
4: Learn the policy π_θ
5: Compute $L = \sum_i \sum_{(s,a)\in\xi_i} \omega_i log(\pi_\theta(s,a))$
6: Update $\theta = \theta + \alpha \bigtriangledown L$
7: **until** the target number of iterations is completed

Then the Q-value can be calculated as:

$$Q_\theta(s,a) = \theta^T \phi(s,a) + \gamma \sum_{s'} T(s,a,s') \bigotimes_{a'} Q_\theta(s',a') \tag{4.1}$$

$$\text{where } \bigotimes_{a} Q_\theta(s,a) = \frac{\sum_a Q_\theta(s,a) \exp(\beta Q_\theta(s,a))}{\sum_{a'} \exp(\beta Q_\theta(s,a'))} \tag{4.2}$$

Eq. 4.1 shows the Boltzmann exploration. Compared to the standard Bellman equation, it enables the likelihood to be differentiable, thus the objective function can be easier optimized. β represents the degree of confidence and it is set as 0.5 in our experiments. The Boltzmann exploration policy parameterized by θ is:

$$\pi_\theta(s,a) = \frac{\exp(\beta Q_\theta(s,a))}{\sum_{a'} \exp(\beta Q_\theta(s,a'))} \tag{4.3}$$

Then the log-likelihood of trajectories $\mathscr{T}$ is calculated as:

$$L(\mathscr{T}|\theta) = log \prod_{i=1}^{N} \prod_{(s,a)\in\xi_i} \pi_\theta(s,a)^{\omega_i} = \sum_{i=1}^{N} \sum_{(s,a)\in\xi_i} \omega_i \log \pi_\theta(s,a) \tag{4.4}$$

Herein, ω_i denotes the weight for ξ_i, which can be estimated by its frequency of occurrence. By maximizing Eq. 4.4, the parameter θ enables the trajectories $\mathscr{T}$ to have the highest probability to be observed given the reward function R_θ. Once the reward function is learned, the strategy followed by $\mathscr{T}$ can be further induced by any RL method, *e.g.*, policy iteration that we employed in this work.

In general, IRL methods assume the reward function to be unique for all input trajectories. However, it is often the case that the trajectories are heterogeneous and have various reward functions. For example, in ITS, students' learning intentions, which actuate their decision-making behaviors, can be different, which is hard to be captured by a homogeneous reward function. Therefore, a model suitable for modeling multiple reward functions is favored.

4.3.2 Expectation Maximization IRL (EM-IRL)

To deal with trajectories with multiple reward functions, *i.e.*, multiple strategies, based on IRL algorithms, Babes-Vroman *et al.* [Bab11] proposed an expectation maximization IRL (EM-IRL). Herein, we adapted the original EM-IRL to automatically determine the optimal number of clusters.

Algorithm 3 EM-IRL

Input: $MDP\backslash R$, trajectories $\mathcal{T}$, the maximal number of clusters K_{max}
Initialize: $k=2$
while $k \leqslant K_{max}$ **do**
  Initialize: ρ_j and θ_j, $j=1,...,k$
  repeat
    E Step: Compute the z_{ij}, $i=1,...,N$
    M Step: Update the prior probability ρ_j and learn reward parameter θ_j via MLIRL
  until the target number of iterations is completed
  if *stop_criterion* is True: **break**; **else**: $k=k+1$
end while

Instead of directly assigning the cluster number, we considered a possibly maximal number of clusters, *i.e.*, K_{max}, and a variable k initialized as 2 indicating the current cluster number.

Specifically, to determine the optimal number of clusters, starting from the cluster number $k=2$, we iteratively implemented the EM procedure, until a pre-defined *stop_criterion* was met. The *stop_criterion* was defined as: either there were some empty clusters generated or the log-likelihood (LL) of the clustering results defined in Eq. 4.5 varied smaller than a pre-defined threshold compared to the last iteration, which we set as 10. The LL reflected the clustering performance by measuring the accordance of learned clusters with the correspondingly induced cluster-wise strategies. In Eq. 4.5, N_j stands for the number of trajectories in cluster j.

$$LL=\sum_{j=1}^{k}\sum_{i=1}^{N_j}\log(z_{ij}) \tag{4.5}$$

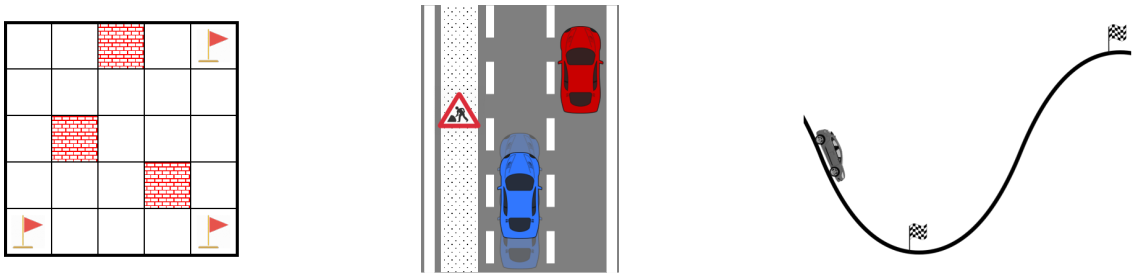
$$z_{ij}=Pr(\xi_i|\theta_j)=\prod_{(s,a)\in\xi_i}\frac{\pi_{\theta_j}(s,a)\rho_j}{Z} \tag{4.6}$$

Before the EM loop, parameters ρ_j and θ_j, $j=1,..,k$, which denoted the estimated prior probability and reward parameter for the j^{th} cluster were randomly initialized.

In the ***E step***, the probability that trajectory i belongs to cluster j was calculated by Eq. 4.6, in which Z is a normalization factor; In the ***M step***, the prior probability of cluster is updated by Eq. 4.7. Meanwhile, the reward parameter θ_j can be learned by any IRL, and herein we employed the MLIRL with weights of trajectories being z_{ij}.

$$\rho_j=\sum_i\frac{z_{ij}}{N} \tag{4.7}$$

The E step and M step will be iteratively executed until a target number of iterations is completed, which was set as 80 in this work to ensure convergence. Finally, we found k clusters when LL converged, with each cluster standing for a group of trajectories with a unique reward function. Based on these reward functions, we could further induce cluster-wise strategies.



(a) Grid World (b) Highway (c) Mountain Car

Figure 4.1 Three simulation environments: (a) Grid World; (b) Highway; (c) Mountain Car.

4.4 Case Studies

4.4.1 Data Preprocessing

Since the ground truth labels of students' subtypes were unknown in advance, it is difficult to directly evaluate the EM-IRL-learned clusters from the students' data. Therefore, we first carried out EM-IRL in three simulation environments that had decided ground-truth labels. If different strategies could be accurately distinguished by EM-IRL in simulations, then we would be more confident to further deploy it in the real-world ITS environment.

4.4.2 Experimental Setup

We explored three simulation environments: Grid World, Highway, and Mountain Car, as illustrated in Figure 4.1. For each environment, we generated trajectories using different strategies. Then taking these trajectories as the input for EM-IRL, we aimed at clustering them into different subgroups.

4.4.2.1 Description of Three Simulation Environments

• **Grid World**: Our Grid World simulation environment was adapted from [Bab11], in which three grids were randomly chosen as puddles indicated by bricks, as shown in Figure 4.1(a).

- **States (25)**: 5×5 grid-size.
- **Actions (4)**: Moving up, down, left, or right.
- **Strategies (3)**: Moving to 1) the upper-right corner; 2) the lower-left corner; or 3) the lower-right corner.

The rewards are designed for the following three decision-making strategies: 1) Upper-right corner has the reward of 10; 2) Lower-left corner has the reward of 10; 3) Lower-right corner has the reward of 10. Otherwise, each state was punished -1.

• **Highway**: Our Highway environment was adapted from a three-lane highway scenario introduced in [Kle12], in which the agent controlled a blue car with three speed levels, switching between the

three lanes or going off-road on either side, as shown in Figure 4.1(b). At all timestamps, there would be a red car in one of the three lanes.

- **States (729)**: The blue car's speed had 3 levels and could move horizontally in 9 locations; The red car could move vertically in 9 locations and horizontally in 3 locations.
- **Actions (5)**: Maintaining, speeding up, slowing down, moving left, or moving right.
- **Strategies (2)**: 1) Keeping off the left lane (imagining a scenario where the left lane is under construction); 2) Driving at the fastest speed.

The rewards are designed for the following two strategies: 1) Driving on the left lane has a reward of -10; 2) Driving with the lowest level of speed has a reward of -10. In both strategies, off-road is punished -0.5, collision is punished -5, and maintaining the state has no reward.

• **Mountain Car**: Our Mountain Car environment was adapted from the MountainCar-v0 in OpenAI Gym [Pal18], in which a car was on a one-dimensional track and moved between two mountains, as shown in Figure 4.1(c).

- **States (80)**: 10 horizontal positions with 8 levels of speed.
- **Actions (3)**: Pushing left, no pushing, or pushing right.
- **Strategies (2)**: 1) Reaching the right mountain top (the car needs to drive back and forth to build up enough momentum to push up); 2) Parking at the valley bottom.

The rewards are generated for the following two strategies: 1) Right mountaintop has a reward of +10; 2) Valley bottom has a reward of +10. Otherwise, each state is punished -1.

In each simulation environment, the initial states were randomly assigned, and the transitions between states were stochastic: in Grid World, the transitions were manually designed to have an 85% probability move to the expected state and a 15% probability move to the other three directions; the transitions in Highway and Mountain Car were estimated from the data.

4.4.2.2 Collecting Trajectories

For each strategy in the three simulation environments, we induced a policy via policy iteration and employed it to collect the trajectories. In Grid World, we got 500 trajectories for each strategy; in Highway, 1000 trajectories were generated per strategy; and in Mountain Car, 1000 for each strategy were collected. In each environment, trajectories with various strategies were mixed, with the total sizes being: Grid World (1500), Highway (2000), and Mountain Car (2000), as illustrated in Figure 4.2.

4.4.2.3 Metrics to Evaluate the Clustering Results

Given the ground truth of cluster belongings in simulation environments, the results of EM-IRL were evaluated by the purity of each cluster and across overall clusters. Denote the size of i^{th} cluster as

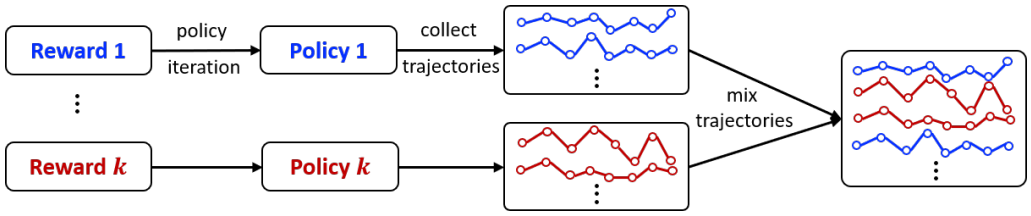


Figure 4.2 The process of collecting mixed trajectories given different rewards.

Table 4.1 Cluster-wise and overall purities learned by EM-IRL clustering in three simulation environments.

Environment	**Cluster-wise**		**Overall**
	Strategy Idx	Purity (%)	Purity (%)
Grid World	1	100	100
	2	100	
	3	100	
Highway	1	100	100
	2	100	
Mountain Car	1	100	96.4
	2	93.2	

N_i with ground-truth labels $\mathbf{L}_i$, then the cluster-wise purity is calculated as the number of majority labels divided by the cluster size:

$$\text{purity}_i = \frac{\text{majority}(\mathbf{L}_i)}{N_i} \tag{4.8}$$

and the overall purity is calculated by the mean of purity among all clusters:

$$\text{purity} = \frac{1}{k}\sum_{i=1}^{k}\text{purity}_i \tag{4.9}$$

4.4.3 Results

The EM-IRL clustering results for the three simulation environments are shown in Table 4.1, in which the first column is the environment; the second and third columns show the index of strategy and the corresponding cluster-wise purity; the last column shows the overall purity among all clusters.

In Grid World, all strategies could be accurately clustered by EM-IRL, with both cluster-wise purities and overall purity being 100%. Likewise, in Highway, the two strategies were accurately clustered with a purity of 100%. In Mountain Car, a few trajectories of driving to the right mountaintop (Strategy 0) were mis-clustered to the parking at the valley (Strategy 1). This is because the mis-clustered trajectories tried to move to the left to collect enough momentum, which showed similar behaviors to reach the valley. Overall, the results suggested the effectiveness of EM-IRL in accurately distinguishing subtypes of trajectories with different strategies in simulation environments.

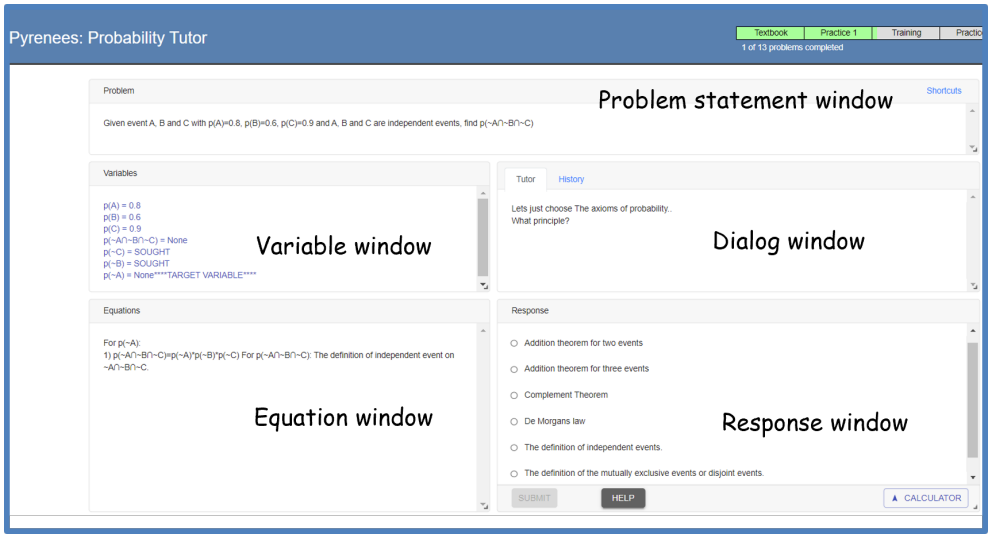


Figure 4.3 The interface of the Pyrenees ITS.

4.5 Experiments with Education Data

Our data was collected by letting students work on a web-based ITS called *Pyrenees*, which taught college students the probability, *e.g.*, the Addition Theorem and the Bayes' Theorem. The instruction was conducted by guiding students through training problems. For each problem, the tutor provided step-by-step instruction, immediate feedback, and on-demand help. The help was provided via a sequence of increasingly specific hints. The last hint in the sequence, *i.e.*, the bottom-out hint, told the student exactly what to do. During training, the students could make pedagogical decisions on whether to solve the next step by themselves or observe the tutor solve it. If they choose to solve by themselves, the tutor will *elicit* the solution from them by asking questions; otherwise, the tutor will show or *tell* them the solution directly.

The interface of the ITS is shown in Figure 4.3. It consists of multiple windows. Specifically, the problem statement window shows the problem to be solved. The dialog window shows the message the tutor provides to the students, such as the explanation for a *tell* step or the prompt for an *elicit* step. In the response window located in the right-bottom section, the student enters responses such as writing an equation or selecting a choice. Any variables or equations generated during the learning process are shown in the variable window and equation window for reference.

4.5.1 Data Preprocessing

4.5.1.1 Learning Process

All students participating in our data collection went through *four phases*: textbook, pre-test, training, and post-test. The details of the four phases are as follows: 1) **Textbook**: The students studied the domain principles from a probability textbook. They read a general description of each principle,

reviewed some examples of it, and solved some single- and multiple-principle problems; 2) **Pre-test**: The students took a *pre-test* which contained 14 problems. During this phase, they would not be given feedback on their answers, nor be allowed to go back to earlier questions (this was also true for the post-test); 3) **ITS training**: During training, students received 12 problems in the same order. Each main domain principle was applied at least twice. The minimal number of steps needed to solve each training problem ranged from 20 to 50. Such steps included variable definitions, principle applications, and equation solving. The number of domain principles required to solve each problem ranged from 3 to 11; 4) **Post-test**: The students took the *post-test* which contained 20 problems in total. 14 of the problems were isomorphic to the problems given in the pre-test phase, while the remaining 6 were harder non-isomorphic multiple-principle problems.

4.5.1.2 Metric of Learning Performance

The pre- and post-tests required students to derive an answer by writing and solving one or more equations. We used three scoring rubrics: binary, partial credit, and one-point-per-principle. Under the binary rubric, a solution was worth 1 point if it was completely correct or 0 if not. Under the partial credit rubric, each problem score was defined by the proportion of correct principle applications evident in the solution. A student who correctly applied 4 of 5 possible principles would get a score of 0.8. The one-point-per-principal rubric in turn gave a point for each correct principle application. All of the tests were graded in a double-blind manner by a single experienced grader. The results we presented were based upon the partial-credit rubric but the same results hold for the other two. For comparison purposes, all test scores were normalized to the range of $[0,100]$.

We measure students' learning performance using *normalized learning gain* (NLG), which measures their gain *irrespective of their incoming competence*. It is calculated as:

$$\text{NLG} = \frac{\text{post} - \text{pre}}{100 - \text{pre}} \tag{4.10}$$

where *pre* and *post* refer to the students' test scores before and after the ITS training respectively and 100 is the maximum score. Herein, for the post-test, we considered all 20 problems that are either isomorphic or non-isomorphic. Specifically, NLG measured that regarding the largest possible improvement the participant could make (*i.e.*, $1-\text{pre}$), and how much improvement that has been actually made (*i.e.*, post$-$pre). Additionally, we measured an *isomorphic NLG* (Iso_NLG). Unlike NLG, the Iso_NLG was calculated based on the pre- and isomorphic post-test scores, which contained only 14 isomorphic multiple-principle problems.

4.5.1.3 States & Actions

Our dataset contains 127 students. Each student spent ~ 2 hours on the system and completed around 400 steps. According to the logs collected by ITS, we represented the students' *states* and *actions* as follows:

• **States**: 142 state features were extracted from the student-system interaction log data. Specifically,

the features can be grouped into five categories (the numbers in the brackets indicate the number of corresponding features):

- **Autonomy** (10): the amount of work done, such as the number of elicits since the last tell;
- **Temporal** (29): time-related information, such as the average time per step;
- **Problem Solving** (35): current problem-solving context, such as problem difficulty;
- **Performance** (57): student's performance so far, such as the percentage of correct entries;
- **Hints** (11): student's hint usage, such as the total number of hints requested.

For each category, we employed K-means clustering to get the discretized states. By selecting an elbow of errors when the clustering results converged, the number of states for each category of features was determined as follows: Autonomy (3 states), Temporal (4), Problem Solving (3), Performance (4), and Hints (3). As a result, we got 432 discrete states in total. Based on the discretized states, we estimated the transition probabilities from all available data.

• **Actions**: The students can take two actions of *elicit/tell*, *i.e.*, to *elicit* the solution by themselves by asking questions, or to let the tutor *tell* them the solution directly.

4.5.2 Experimental Setup

4.5.2.1 Student Subtyping by EM-IRL

Based on the EM-IRL-learned clusters, we first conducted analyses by checking the statistical significance among different clusters' learning performance, including the pre-test scores, isomorphic NLG (Iso_NLG), NLG, students' learning time on the training task (Time), and the percentage of eliciting in students' decisions (Elicit_Perc).

4.5.2.2 Student Subtyping by Other Methods

• **Clustering by Traditional Methods**: To evaluate the clustering performance of EM-IRL, we compared it with three other clustering methods: two K-means-based approaches which took the pre-test scores and the learning state in the final step as the input respectively, and a K-medoids based approach which took dynamic time warping (DTW) [Ber94] distance between trajectories as the input. The K-means-based approaches were static-information-based clustering while the K-medoids-based DTW considered dynamic state transitions in the trajectories. In our experiments, each of these methods generated three clusters, and for each cluster, the MLIRL was employed to learn a strategy. Based on the learned strategies, we calculated the log-likelihood (LL, referring to Eq. 4.5) of observing such clustering results.

• **Clustering by Matching RL/IRL Policies**: We further explored whether RL/IRL policies could model heterogeneity in student decision-making. The RL/IRL policies were induced as follows:

1) *Inducing the RL policy*: To investigate whether students' learning strategies could be distinguished from the *tutor's* perspective, we compared students' decisions to an RL-induced pedagogical

policy and clustered the students based on the matching rate. Since the RL policy was induced to improve students' learning performance, it is expected that the group with a higher matching rate with the RL policy would have better learning performance.

Specifically, we applied RL to learn a pedagogical policy that determines whether the next step should be elicit or tell (same as our ITS). The training data set contained 1,118 students' interaction logs collected from a series of seven prior studies which followed the identical procedure and learning materials as the students in this study described in Section 4.5.1. The same 142 features used by EM-IRL were extracted from the logs and used to induce the policy. In an empirical classroom study, the policy was compared with a deep Q-network (DQN) induced policy and a random policy. Results showed that the RL policy significantly outperformed both of them [Zho19].

Once the RL policy was induced, we applied it to the student decision-making data (127 students) to see what decision the RL policy would make on each step. Then, we calculated the matching rate between students' decisions and the RL policy individually for each student. Based on the matching rates, the students were split into three groups via K-means clustering, denoted as High, Medium, or Low based on the average matching rate of the group.

2) *Inducing the IRL Policy*: Similarly, to investigate whether students' learning strategies could be distinguished from their perspective, we applied IRL to induce a policy from student decision-making data and compared students' decisions with the IRL policy. Given that our data analysis showed that most of the students learned significantly from ITS training, herein, we assumed that a majority of students completed the training to learn. Thus, we expected that the group with a higher matching rate with the IRL policy would have better learning performance.

The IRL policy was induced based on the 127 students who were given the opportunities to make pedagogical decisions during training. Herein, the MLIRL algorithm [Bab11] was utilized for policy induction. Similar to the RL-based method, the IRL policy was applied back to students' data to calculate the matching rate between students' decisions and the IRL policy. Then, K-means clustering was applied to the matching rate to cluster students into High, Medium, or Low groups

4.5.3 Results

4.5.3.1 Student Subtyping by EM-IRL

Fitting students' data to the EM-IRL framework in Section 4.3.2, when *stop_criterion* was met, we got three clusters. Table 4.2 shows the EM-IRL subtyping results. From left to right, it shows the students' subtypes, the number of students (# Stu), pre-test score (Pre), isomorphic NLG (Iso_NLG), NLG, time on the training task (Time), and percentage of *elicit* in students' decisions (Elicit_Perc). Based on statistical analysis, we named the three resulting clusters as: *learning-oriented*, *efficient-oriented*, and *no-learning*, respectively.

A one-way ANOVA analysis on pre-test scores showed no significant difference among the three clusters: $F(2,124) = 1.36$, $p = 0.260$, $\eta = 0.022$. This suggested that students in the three clusters were balanced in incoming competence. To measure students' learning gain in training, we

Table 4.2 EM-IRL clustering results in an ITS environment.

Subtype	#Stu	Pre	Iso_NLG	NLG	Time	Elicit_Perc (%)
learning-oriented	50	73.9(16.8)	55.9(45.3)	23.4(53.6)	2.52(.70)	87.53(13.40)
efficient-oriented	64	76.2(14.5)	43.9(92.4)	-4.4(127.2)	2.18(.45)	84.93(15.02)
no learning	13	81.9(17.4)	-21.1(212.1)	-98.4(340.4)	2.10(.50)	77.06(20.04)

conducted analyses on their Iso_NLG and NLG. A one-way ANOVA analysis on Iso_NLG showed a significant difference among the three clusters: $F(2,124)=3.24$, $p=0.042$, $\eta=0.050$. Subsequent contrast analysis revealed that *learning-oriented* > *no learning*: $t(124)=2.54$, $p=0.012$, $d=0.75$ and *efficient-oriented* > *no learning*: $t(124)=2.19$, $p=0.030$, $d=0.54$. Similar results were found for NLG in that a one-way ANOVA analysis showed a significant difference among the three clusters: $F(2,124)=3.73$, $p=0.027$, $\eta=0.057$. Subsequent contrast analysis revealed that *learning-oriented* and *efficient-oriented* significantly outperformed *no learning*: $t(124)=2.73$, $p=0.007$, $d=0.77$ and $t(124)=2.15$, $p=0.033$, $d=0.52$, respectively.

In terms of time on task, a one-way ANOVA analysis showed a significant difference among the three clusters: $F(2,124)=5.81$, $p=0.004$, $\eta=0.086$. Subsequent contrast analysis indicated that *learning-oriented* took longer time on task than the other two clusters: $t(124)=-3.11$, $p=0.002$, $d=0.58$ for *efficient-oriented* and $t(124)=2.37$, $p=0.019$, $d=0.63$ for *no learning*. A contrast analysis on the percentage of elicit in students' decisions revealed that *learning-oriented* took significantly more elicit actions than *no learning*: $t(124)=2.24$, $p=0.027$, $d=0.70$.

To summarize, the *learning-oriented* subtype spent significantly more time than the other two groups on the training task and achieved the best performance on both Iso_NLG and NLG (significantly higher than *no learning*). This suggested that learning-oriented students mainly focused on learning the materials, regardless of the time they may spend. The *efficient-oriented* subtype significantly outperformed *no learning* on learning performance and at the same time spent significantly less time than *learning-oriented*. This suggested that *efficient-oriented* students could balance learning gain and time on task. Finally, the *no learning* subtype spent less amount of time and achieved the lowest learning outcomes.

4.5.3.2 Student Subtyping by Other Methods

• **Clustering by Traditional Methods**: We compared our EM-IRL with three traditional baseline clustering methods, namely K-means on the pre-test score (K-means on Pre); K-means on the learning state (142 features) in the final step (K-means on Final Step); K-medoids on the DTW distance among trajectories [Ber94], which is calculated based on the 142 features (K-medoids on DTW). The results are shown in Table 4.3, with the two columns being the clustering method and the resulting log-likelihood (LL).

Overall, results showed that the dynamic-information-based clustering approaches (K-medoids on DTW and EM-IRL) performed better than static-information-based approaches (K-means on

Table 4.3 Comparison of the log-likelihood (LL) for different clustering methods.

Method	LL ($\times 10^3$)
K-means on Pre	-10.68
K-means on Final Step	-9.60
K-medoids on DTW	-8.83
EM-IRL	**-6.36**

Pre and K-means on Final Step). Between the two static-information-based approaches, K-means on the final Step performed better than K-means on the pre-test. This is not surprising because the state in the final step included information generated during training while the pre-test score only included information till the end of the pre-test. Between the two dynamic-information-based approaches, EM-IRL outperformed K-medoids on DTW. A possible reason is that EM-IRL took both states and actions into account while K-medoids on DTW considered only the states in trajectories.

• **Clustering by Matching RL/IRL Policies**:

1) *Results of Matching with the RL Policy*: Based on the matching rate with RL policy, we got three clusters by K-means: High ($M = .84, SD = .05$), Medium ($M = .70, SD = .05$), and Low ($M = .52, SD = .07$). A one-way ANOVA analysis over the matching rate showed a significant difference: $F(2,124) = 339.87$, $p < 0.0001$, $\eta = 0.846$. Subsequent contrast analysis showed that: High > Medium: $t(124) = 4.38$, $p < 0.0001$, $d = 0.99$ and Medium > Low: $t(124) = 8.01$, $p < 0.0001$, $d = 1.70$.

A one-way ANOVA analysis on pre-test showed there was no significant difference among the three groups: $F(2,124) = 0.26$, $p = 0.771$, $\eta = 0.004$. Analyses on Iso_NLG (calculated based on pre-test and isomorphic post-test) and NLG (calculated based on pre-test and full post-test, which contains 6 additional hard problems) also showed no significant difference among the three groups. For the time on the training task, there was a significant difference among the three groups: High ($M = 2.40, SD = .50$), Medium ($M = 2.42, SD = .66$), and Low ($M = 1.88, SD = .40$). A one-way AVONA on time shows: $F(2,124) = 9.21$, $p = 0.0002$, $\eta = 0.129$. Subsequent contrast analysis revealed that the High and Medium groups spent significantly more time than the Low group: $t(124) = 3.85$, $p = 0.0002$, $d = 1.11$ and $t(124) = 3.99$, $p = 0.0001$, $d = 0.92$, respectively. An analysis on the percentage of elicit in students' decisions showed a significant difference among the three groups: $F(2,124) = 66.97$, $p < 0.0001$, $\eta = 0.519$. Subsequent contrast analysis revealed that High > Medium: $t(124) = 4.38$, $p < 0.0001$, $d = 0.99$ and Medium > Low: $t(124) = 8.01$, $p < 0.0001$, $d = 1.70$.

The results showed that by matching with the RL strategy, we could differentiate students' time-consuming strategies from time-efficient strategies. However, it was not able to identify the student subtypes that made a difference in the learning performance. This suggested the presence of a gap between the tutor's and the students' strategies. Specifically, compared to taking actions following the tutor's decisions passively, the students might prefer to actively direct their own learning process. Therefore, when deploying the tutor's strategy to students, it might not promote the learning performance as expected.

2) *Results of Matching with the IRL Policy*: Based on the matching rate with the IRL policy,

we got three clusters by K-means: High ($M = .86, SD = .05$), Medium ($M = .71, SD = .05$), and Low ($M = .54, SD = .06$). A one-way ANOVA analysis over the matching rate showed a significant difference among the three groups: $F(2,124) = 360.99$, $p < 0.0001$, $\eta = 0.853$. Subsequent contrast analysis showed that: High > Medium: $t(124) = 15.92$, $p < 0.0001$, $d = 3.37$ and Medium > Low: $t(124) = 13.52$, $p < 0.0001$, $d = 3.23$.

A one-way ANOVA analysis on pre-test showed there was no significant difference among the three groups: $F(2,124) = 1.17$, $p = 0.314$, $\eta = 0.019$. Analyses of the Iso_NLG and NLG also showed no significant difference among the three groups. In terms of time on the training task, there was a significant difference among the three groups: High ($M = 2.44, SD = .54$), Medium ($M = 2.27, SD = .68$), and Low ($M = 2.08, SD = .42$). A one-way AVONA on time shows: $F(2,124) = 3.11$, $p = 0.048$, $\eta = 0.048$. Subsequent contrast analysis showed that the High group spent significantly more time than the Low group: $t(124) = 2.43$, $p = 0.017$, $d = 0.70$. An analysis on the percentage of elicit in students' decisions showed a significant difference among the three groups: $F(2,124) = 93.92$, $p < 0.0001$, $\eta = 0.602$. Subsequent contrast analysis revealed that High > Medium: $t(124) = 7.95$, $p < 0.0001$, $d = 1.83$ and Medium > Low: $t(124) = 7.08$, $p < 0.0001$, $d = 1.43$.

The results showed that IRL-based policy matching was able to cluster the students' strategies in time. However, it was unable to learn specific subtypes of students whose strategy will lead to better learning outcomes. One possible reason that the IRL-based analyses could not identify the learning-performance-impactful strategies is that a single policy was insufficient to effectively generalize the decision-making patterns for the overall students. Different students might follow heterogeneous decision-making strategies.

In short, the results suggested that EM-IRL could effectively conduct student subtyping that reflects different decision-making strategies. In contrast, clustering by traditional methods or by matching RL/IRL policies could not find desired student subtypes.

4.6 Summary

In this chapter, we investigated the **across-heterogeneous** reward functions via an **EM-IRL** algorithm in an application of students' subtyping. By analyzing students' subtyping, we aimed at putting ourselves in the shoes of students to better understand their decision-making. To evaluate the performance of EM-IRL, we first applied it to three simulation environments, where the EM-IRL displayed robust performance to accurately cluster the trajectories with different strategies. Given the accurate clustering results in simulators, we were more confident to further apply EM-IRL to real-world longitudinal students' logs collected from an ITS. The results suggested that the EM-IRL could effectively group students with different subtypes, *e.g.*, learning-oriented, efficient-oriented, and no-learning. In contrast, clustering by traditional methods or by matching RL/IRL policies could not find desired subtypes. The subtyping results showed the potential of providing tutors evidence to give more customized interventions to better assist students' learning.

5 THEMES: EVOLVING-HETEROGENEOUS REWARD FUNCTIONS

Based on the analysis in Chapter 3 and Chapter 4, in this chapter, we develop an apprentice learning (AL) framework to learn the **evolving-heterogeneous** reward functions in a *healthcare* application. Existing apprenticeship learning (AL) methods commonly require *online* interactions, and the majority of them assume the experts' demonstrations are driven by a *single* reward function. To handle more general cases in real-world human-centric applications, *e.g.*, healthcare, where the policy is usually desired to be learned in an *offline* manner, and the experts' demonstrations generally involve *multiple* reward functions *evolving* over time, in this chapter, we propose an apprenticeship learning (AL) framework named Time-aware Hierarchical EM Energy-based Sub-trajectory (**THEMES**) clustering. The effectiveness of the THEMES framework is evaluated via an extremely challenging task – sepsis treatment. Our results showed that THEMES can accurately learn the treatment policies compared to competitive baselines, which can shed some light on evaluating clinicians' interventions and assisting them to carry out personalized treatments in time.

5.1 Overview

Apprenticeship learning (AL) aims at inducing decision-making policies via observing and imitating experts' demonstrations [Abb04b]. The existing AL approaches are commonly *online*, requiring interaction with the environment iteratively for collecting new data and then updating the model accordingly [Abb04b; Zie08; Ho16; Fin16]. Since the execution of a potential bad policy can be costly or even dangerous in many real-world human-centric tasks [Lev20], *e.g.*, healthcare, it is highly desired to induce the policies purely based on the demonstrated behaviors in an *offline* manner. The recently proposed energy-based distribution matching (EDM) algorithm [Jar20] has advanced the state-of-the-art in *offline* AL. Despite the great success of EDM, there are two major challenges when applying it to real-world applications such as healthcare: 1) EDM assumes all demonstrations to be generated with a homogeneous policy driven by a *single* reward function, whereas when treating the patients under different disease progressive stages, *e.g.*, common fever or severe shock, clinicians

will adaptively carry out different policies with *multiple* varying reward functions [Wan21; Wan22]. In order to capture the disease progression stages, we propose a *hierarchical* AL approach that automatically partitions trajectory into sub-trajectories by taking account of the reward function; 2) EDM assumes all the demonstrations to be regularly collected with equal time intervals. The real-world data, however, are generally collected irregularly. For example, the time intervals in electronic healthcare records (EHRs) can be varying from seconds to days [Bay17; Zha19]. To handle this issue, we develop a *time-awareness* AL framework, which can capture the latent progressive patterns to induce more accurate policies.

Specifically, we propose a **T**ime-aware **H**ierarchical **EM** **E**nergy-based **S**ubtrajectory (**THEMES**) AL framework to address the *multiple* reward functions *evolving* in an *offline* manner. Many previous AL works primarily focused on *single* reward function [Abb04b; Zie08; Ho16; Fin16; Jar20]. In order to model the *multiple* reward functions varying across trajectories (while remaining the same within each trajectory), Babes *et al.* proposed an offline EM-based inverse reinforcement learning (EM-IRL) [Bab11], which can cluster the demonstrated trajectories and learn the respective policies for each cluster, simultaneously [Bab11]. To tackle the multiple reward functions *evolving* over time, the key challenge lies in how to effectively partition the trajectories. For example, in healthcare, the more precisely the partitioned sub-trajectories reflect the actual disease progression stages, the more accurate the induced policies will be. Recently, a few AL approaches have been proposed to handle this issue [Kri16; Hau17; Sha18; Wan21; Wan22]. Among them, Wang *et al.* explored breaking the trajectory into *fixed-length* sub-trajectories for hierarchical AL [Wan22]. For more flexible partitioning of the trajectories, Wang *et al.* proposed to learn sub-trajectories via contextual bandits [Wan21]; Krishnan *et al.* proposed a hierarchical inverse reinforcement learning (HIRL) [Kri16] to learn the sub-trajectories by Gaussian mixture model; Hausman *et al.* developed a multi-modal imitation learning (MIL) [Hau17] to segment the trajectories via InfoGAN [Sha18].

The core of THEMES consists of two components: 1) a Reward-regulated Multi-series Time-aware Toeplitz Inverse Covariance-based Clustering (**RMT-TICC**), which can incorporate the *time-awareness* when partitioning the trajectories. By introducing a reward regulator learned in a hierarchical manner, the RMT-TICC can capture decision-making patterns to derive fine-grained sub-trajectories; and 2) an *offline* EM Energy-based Distribution Matching (**EM-EDM**) to cluster the partitioned sub-trajectories and induce policy for each cluster, simultaneously. The original EDM [Jar20] has been compared against the state-of-art offline AL methods such as adversarial imitation learning [Kos19] on a series of benchmarks, *e.g.*, Acrobot, BeamRider, LunarLander, and BeamRider [Bro16]. In this chapter, we focus on experts' demonstrations with *multiple* reward functions *evolving* over time, thus we leverage EHRs on an extremely challenging task – sepsis prevention. As introduced in Section 3.1, sepsis is a life-threatening organ dysfunction and a leading cause of death worldwide [Sin16]. Without proper interventions, sepsis can progress from infection to septic shock, which is the most severe stage with a mortality rate as high as 50% [Mar03]. Contrarily, 80% of sepsis deaths can be prevented with timely interventions [Kum06]. Our results show that THEMES outperforms baseline methods which assume the reward function to be either *single* or *multiple*

varying across demonstrations as well as competitive ablation methods.

The major contributions in this chapter are three-folded: 1) By incorporating the *time-awareness* and *decision-making patterns* in RMT-TICC, fine-grained sub-trajectories can be derived to indicate different progressive stages; 2) Utilizing our offline EM-EDM, decision-making policies over different progressive stages can be efficiently induced, without the need of rolling out the policies iteratively; 3) The THEMES framework is helpful to model complex human-centric decision-making tasks. Specifically, more accurate policies induced over the EHRs data shed some light on evaluating the clinicians' interventions and assisting them to carry out personalized treatments in time.

5.2 Related Work

5.2.1 Offline AL

The existing AL approaches are commonly *online*, requiring interaction with the environment iteratively for collecting new data and then updating the model accordingly [Abb04b; Zie08; Ho16; Fin16]. Since the execution of a potential bad policy can be costly or even dangerous in many real-world *human-centric* tasks [Lev20], *e.g.*, healthcare and education, it is highly desired to induce the policies purely based on the demonstrated behaviors in an *offline* manner. Though behavior cloning (Refer to Section 2.2) can handle this issue by directly learning the mapping from states to actions, without the need of rolling out the policy online, it highly depends on the quality and quantity of demonstrations. For better modeling the offline AL, some inverse reinforcement learning (IRL)-based and adversarial imitation learning (AIL)-based offline adaptations [Abb04b; Zie08; Bab11; Rag17; Kos18; Kos19] have been proposed. However, these adapted methods will inherit the disadvantages of the original IRL/AIL. More recently, an energy-based distribution matching approach [Jar20] was developed, which can overcome the disadvantages of adapted IRL/AIL.

• **IRL-based and AIL-based Offline Adaptations**: When re-purposing the *IRL* to offline, its original computational and theoretical disadvantages would be inherited accordingly. For example, like the original IRL, the IRL-based adaptations generally suffer from inefficiency in inferring an explicit reward function and inducing a policy by RL for each loop. Additionally, IRL methods usually model the reward via a certain tractable format, *e.g.*, linear function [Abb04b; Zie08; Bab11] that mapping from states/state-action pairs to reward values. Without excessive feature engineering, the underlying rewards might be imperfectly inferred, thereby rendering ineffective policies, while this issue might be inherited in IRL-based offline adaptations. Besides, to avoid rolling out the learned policy, there have also been some batch-IRL proposed [Rag17], while they usually require off-policy evaluation to update the reward function, which is itself nontrivial with imperfect solutions. Likewise, When re-purposing the *AIL* for the offline setting, the original computational and theoretical disadvantages in AIL would be inherited accordingly. For example, under the batch setting, to avoid rolling out the policy, some off-policy AIL methods have been proposed based on off-policy actor-critic [Kos18; Kos19]. However, these methods would inherit the complex alternating max-min optimization from the original AIL [Ho16].

• **Energy-based Distribution Matching (EDM)**: To better handle the offline setting, recently an EDM algorithm [Jar20] was proposed, and it has demonstrated better performance compared to both *IRL*- and *AIL-based* adaptations. The major limiting factor when adapting IRL- and AIL-based methods to offline settings lies in their RL-centric structure, which is intrinsically online while relying on off-policy techniques and may introduce some variance to affect the policy-learning performance. To clarify the requirements in the offline setting, Jarrett *et al.* defined a strictly batch imitation learning (SBIL) problem [Jar20] with three criteria: 1) *Policy:* learning a policy directly, without explicitly inferring an intermediate reward function; 2) *Occupancy:* fully exploiting the information contained in the demonstrations; and 3) *Intrinsically batch:* working offline without knowing/inferring the dynamics, and without leaning on off-policy evaluation in an iterative loop or in max-min optimizations. To solve the SBIL problem, they developed a strictly batch imitation learning algorithm named energy-based distribution matching (EDM) [Jar20], which handles the scenario when there is no access to reinforcement signals, no knowledge of transition dynamics, and no interaction with the environment. Based on a joint energy-based modeling (JEM) approach [Gra19], the EDM simultaneously learns a classifier to model the policy and learns a density to model the distribution of states visited by the demonstrators. Compared to rigid behavior cloning, it can take the distribution of states into account to fully exploit the information conveyed by demonstrations; Meanwhile, compared to IRL- and AIL-adapted methods, it can learn a parameterized policy directly without specifying any assumptions or constraints over the rewards, and it is also strictly offline, without any off-policy evaluations and complex max-min optimization in each iteration.

5.2.2 AL with Evolving-Heterogeneous Reward Functions

In existing AL methods, it is commonly assumed that all demonstrations are generated with a homogeneous policy driven by a *single* reward function. However, it is usually not the case for real-world *human-centric* tasks. For example, in education, the reward functions driving the decision-makings can be varying across different students [Mit03]; In healthcare, when treating patients under different disease progressive stages, *e.g.*, common fever or severe shock, clinicians will adaptively carry out different policies with multiple reward functions evolving over time [Wan21; Wan22]. To model such heterogeneous decision-making patterns in AL, in Chapter 4, we have analyzed the scenario when the reward functions are varying across different demonstrations, *i.e.*, *across*-heterogeneous reward functions; Herein, we mainly focus on a more complex case when multiple reward functions are varying over time, *i.e.*, *evolving*-heterogeneous reward functions.

As specified in Section 4.2.3, some prior works have been proposed for handling the heterogeneous reward functions that vary across different demonstrations, *i.e.*, *across*-heterogeneous reward functions. Especially, the EM-IRL [Bab11] algorithm has attracted attention due to its flexibility to be adapted using different AL algorithms for its M-step. In both [Bab11] and our Chapter 4, the maximum-likelihood IRL (MLIRL) was employed in the M-step, which uses a gradient ascent method to optimize the reward parameters. The MLIRL can be deployed in the offline setting; however, it is model-based, requiring knowing the state transition probabilities. Meanwhile, the states in

EM-IRL are required to be discrete and not scalable for large state space. Therefore, it is highly desired to replace the original MLIRL in the M-step of EM-IRL with a more advanced AL approach. Additionally, the EM-IRL cannot handle the *evolving*-heterogeneous reward functions.

Specifically, to model the *evolving*-heterogeneous reward functions varying over time, there have been some prior works proposed. Krishnan *et al.* developed a hierarchical inverse reinforcement learning (HIRL) for long-horizon tasks with delayed rewards [Kri16]. Each trajectory is assumed to complete a certain task and can be decomposed into sub-tasks with shorter horizons based on transitions that are consistent across demonstrations. By identifying the changes in local linearity, the sub-tasks can be identified, which will be further utilized for constructing the local rewards concerning the global sequential structure. Hausman *et al.* proposed a multi-modal imitation learning (MIL) [Hau17] framework to segment and imitate skills (*i.e.*, policies) from unlabelled and unstructured demonstrations by learning policy segmentation and imitation learning jointly via an InfoGAN [Sha18]. Based on the generative adversarial imitation learning, MIL augmented the input with a latent intention distributed by a categorical or uniform distribution to model the heterogeneity of policies varying over time. More recently, Wang *et al.* explored breaking the trajectory into *fixed-length* sub-trajectories for hierarchical AL [Wan22]. In another work proposed by Wang *et al.*, the sub-trajectories can be more flexibly partitioned, by learning with contextual bandits [Wan21]. These methods are generally developed based on generative adversarial imitation learning [Ho16], which requires *online* interactions or off-policy evaluations to update the model.

5.3 Methodology

5.3.1 THEMES Framework

The overview of the THEMES framework is illustrated in Figure 5.1 and detailed in Algorithm 4. Taking experts' demonstrated trajectories as input, an **RMT-TICC** will partition and cluster the sub-trajectories over the states, so that each cluster of sub-trajectories can share the same time-invariant patterns, which will be considered as a *high-level state*; Then regarding the state-action pairs over the partitioned sub-trajectories, an **EM-EDM** will cluster and induce their policies, so that each cluster can share the same decision-making patterns, which will be considered as *high-level actions*. We denote the RMT-TICC derived sub-trajectory cluster indexes as $\{S_k | k = 1, ..., K\}$, and EM-EDM learned policy cluster indexes as $\{\pi_g | g = 1, ..., G\}$. Based on the high-level state-action pairs, a *high-level reward regulator* $\overline{R}(\cdot)$ can be learned based on an EM-IRL [Bab11] and fed back to refine the RMT-TICC. The whole procedure will be conducted interactively until reaching a pre-defined threshold of Thres. Finally, the framework will output the fine-grained sub-trajectories clusters with their respective policies. The two key components, i.e., RMT-TICC and EM-EDM, will be detailed in the following sections.

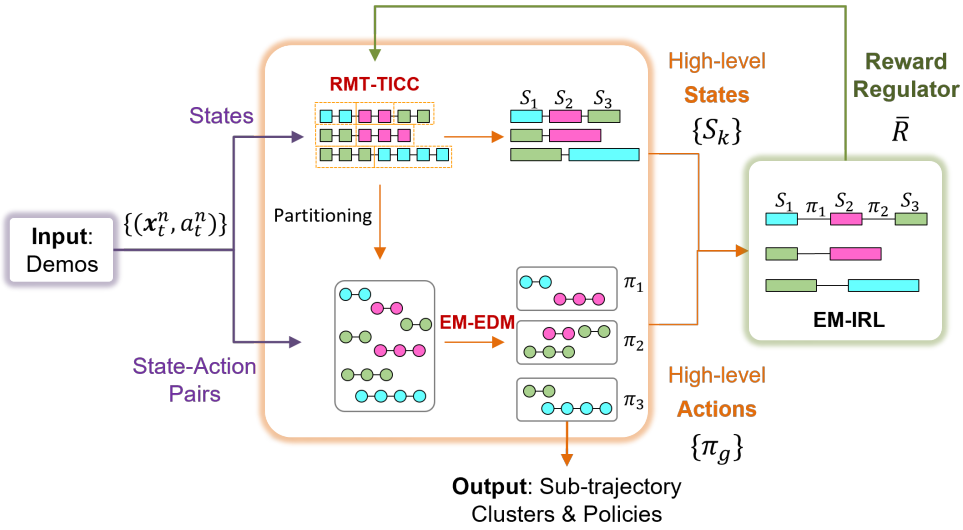


Figure 5.1 Overview of THEMES framework.

Algorithm 4 THEMES

Input: Demonstrated trajectories with state-action pairs: $\{(\mathbf{x}_t^n, a_t^n)|t=1,...,T^n\}$, $n \in \{1,...,N\}$
Initial: Reward regulator $\bar{R}(\cdot)=1$
while iter < Thres **do**
    Conduct **RMT-TICC** to derive *high-level states*, with the state indexes of $\{S_k|k=1,...,K\}$
    Conduct **EM-EDM** to derive *high-level actions*, with the action indexes of $\{\pi_g|g=1,...,G\}$
    Conduct EM-IRL over high-level state-action pairs to derive a reward regulator $\bar{R}(\cdot)$
end while
Output: Sub-trajectory clusters with respective policies

5.3.2 RMT-TICC: Sub-trajectory Partitioning

5.3.2.1 Preliminaries

Given N trajectories, the *states* in n-th trajectory are denoted as $\{\mathbf{x}_t^n|t=1,...,T^n\}$, with $\mathbf{x}_t^n \in \mathbb{R}^m$ being the t-th multivariate state with m features. Herein, our motivation is to simultaneously *partition and cluster* the trajectories based on their latent patterns. Specifically, we will learn a mapping from each state $\mathbf{x}_t^n$ to a certain cluster $\{k|k=1,...,K\}$. Since the states in trajectories are consecutive and each of them is dependent on its neighbors, instead of treating each state independently, we explore the patterns in a sliding window $\omega \ll T^n$. For $\mathbf{x}_t^n$, its preceding states within ω, *i.e.*, $\mathbf{X}_t^n = \{\mathbf{x}_{t-\omega+1}^n,...,\mathbf{x}_t^n\}$, are extracted for determining which cluster k the state $\mathbf{x}_t^n$ belongs to. To learn the clustering mapping in an unsupervised manner, each state $\mathbf{X}_t^n$ is treated as a $m\omega$-dimension random variable (obtained by concatenating the ω states in $\mathbf{X}_t^n$), and all states are then optimally fit into K Gaussian distributions, with the k-th fitted distribution corresponding to the k-th cluster.

To characterize the cluster distributions, Hallac *et al.* proposed a Toeplitz inverse covariance-based clustering (TICC) [Hal17] to learn the mean and inverse covariance matrix for each cluster.

More specifically, determining the optimal mean vectors $\{\mu_k|k=1,...,K\}$ is equivalent to matching each state to an optimal cluster, which leads to the clustering assignments $\mathbf{P}=\{P_k|k=1,...,K\}$, where $P_k \subset \{1,...,T\}$ denotes the indices of states (sliding windows) belonging to the cluster k; Meanwhile, determining the optimal $\{\mathbf{\Theta}_k|k=1,...,K\}$ is to estimate K inverse covariance matrices with block-wise Toeplitz constraints. Note that the inverse covariance matrix is used rather than the covariance matrix because it models conditional dependencies and can easily introduce a graph structure during the matrix learning, which can substantially decrease the number of parameters to reduce the risk of overfitting [Mei06]. Each $\mathbf{\Theta}_k$ for a cluster k is constrained to be block-wise Toeplitz, which is composed of ω sub-blocks $A^{(i)} \in \mathbb{R}^{m\times m}$, $i \in [0, \omega-1]$. The sub-block $A^{(i)}$ represents partial correlations among m features between timestamp t and $t+i$. For instance, the (p,q)-th entry in $A^{(i)}$ indicates the partial correlation between the p-th feature at t and the q-th feature at $t+i$, where $p,q \in \{1,...,m\}$. The block-wise Toeplitz constraints enable $\mathbf{\Theta}_k$ to capture time-invariant structural patterns within $\mathbf{X}_t^n$. More details can be found in Section 3.3.1.

When applying TICC to real-world scenarios, *e.g.* healthcare, there are two challenges: 1) TICC takes a single trajectory (*i.e.*, N=1) as input, while the real-world dataset usually contains multiple time series; 2) The states in real-world datasets are usually collected irregularly, while TICC treats irregular intervals equally and always encourages the consecutive states to be assigned to the same cluster. To address these issues, based on TICC, we proposed multi-series time-aware TICC (MT-TICC) in Chapter 3, which takes multi-series as input and incorporates the *time-awareness*. Specifically, it estimates the mean and inverse covariance matrix based on the states from different trajectories and employs a decay function to re-evaluate the cluster consistency over time. Based on these improvements, MT-TICC got significantly better performance compared to TICC. Despite the success of MT-TICC, it mainly focuses on *observed states*, while the *actions* are not fully explored. Since the *interventions* usually have a great impact on the progression of trajectories [Kom18], it is highly desired to incorporate them when learning the sub-trajectory clusters.

5.3.2.2 RMT-TICC

To incorporate the decision-making patterns learned from state-action pairs, based on MT-TICC, we propose a *Reward-regulated* MT-TICC (RMT-TICC), considering the fact that the actions are driven by the reward function. The objective function is shown in Eq. 5.1:

$$\underset{\mathbf{\Theta},\mathbf{P}}{\operatorname{argmin}} \sum_{k=1}^{K}\Bigg[\sum_{n=1}^{N}\sum_{\mathbf{X}_t^n\in\mathbf{P}_k}\Bigg(\overbrace{-\ell\ell(\mathbf{X}_t^n,\mathbf{\Theta}_k)}^{\text{Log-likelihood}}+\overbrace{c(\mathbf{X}_{t-1}^n,\mathbf{P}_k,\Delta T_t^n,\overline{r}_t^n)}^{\text{Reward-regulated Time-aware Consistency}}\Bigg)+\lambda\overbrace{||\mathbf{\Theta}_k||_1}^{\text{Sparsity}}\Bigg] \tag{5.1}$$

• *Log-likelihood term* measures the probability that $\mathbf{X}_t^n$ follows the $\mathbf{\Theta}_k$ and belongs to the cluster k. Assuming $\mathbf{X}_t^n \sim N(\mu_k, \mathbf{\Theta}_k^{-1})$, $\ell\ell(\mathbf{X}_t^n,\mathbf{\Theta}_k)$ is defined as:

$$\ell\ell(\mathbf{X}_t^n,\mathbf{\Theta}_k) = -\frac{1}{2}(\mathbf{X}_t^n-\mu_k)^T\mathbf{\Theta}_k(\mathbf{X}_t^n-\mu_k)+\frac{1}{2}\log|\mathbf{\Theta}_k|-\frac{m}{2}\log(2\pi) \tag{5.2}$$

• *Reward-regulated Time-aware Consistency term* encourages the consecutive states $\{\mathbf{X}_{t-1},\mathbf{X}_t\}$

to be assigned to the same cluster, while considering the time interval and the corresponding rewards. It is defined as:

$$c(\mathbf{X}_{t-1}^n, \mathbf{P}_k, \Delta T_t^n, \overline{r}_t^n) = \frac{\beta \mathbb{1}\{t-1 \notin P_k\}}{\Phi(f(\overline{r}_t^n), \log(e+\Delta T_t^n))} \tag{5.3}$$

where β is a weight parameter; $\mathbb{1}\{t-1 \notin P_k\}$ is an indicator function, with the value of 1 if $\mathbf{X}_{t-1}^n$ does not belong to the same cluster with $\mathbf{X}_t^n$, otherwise its value is 0; $\log(e+\Delta T_t^n)$ is a decay function, which can adaptively relax the penalization over the consistency constraint when the interval ΔT_t^n between consecutive states becomes larger [Bay17; Zha19].

To incorporate the decision-making patterns to refine the sub-trajectory clustering, we employ a *hierarchical* structure to infer the reward $\{\overline{r}_t^n | t = 1, ..., T^n\}$ for each state-action pair and then utilize them to regulate the consistency constraint as illustrated in Figure 5.1. Given the high-level states learned by RMT-TICC and the high-level actions learned by EM-EDM, a high-level reward function $\overline{R}(\cdot)$ can be inferred accordingly. To learn the $\overline{R}(\cdot)$, here we employ an EM inverse reinforcement learning (EM-IRL) [Bab11] algorithm. Given the high-level reward function, we can get the reward for each state-action pair via $\overline{r}_t^n = \overline{R}(\mathbf{x}_t^n, a_t^n) = 1/G \sum_{g=1}^{G} z_{tg}^n \pi_g(\mathbf{x}_t^n, a_t^n) r_{tg}^n(\mathbf{x}_t^n, a_t^n)$, where G is the number of decision-making policies, *i.e.*, high-level actions; z_{tg}^n is the probability $(\mathbf{x}_t^n, a_t^n)$ in accordance with the g-th policy; $\pi_g(\mathbf{x}_t^n, a_t^n)$ denotes the probability of taking a_t^n at $\mathbf{x}_t^n$ under the g-th policy; and $r_{tg}^n(\mathbf{x}_t^n, a_t^n) = \overline{R}(S_k, \pi_g)$ denotes the rewards learned at high-level when $\mathbf{x}_t^n$ falls into the sub-trajectory cluster S_k. In the first iteration of THEMES, the function $\overline{R}(\cdot)$ is initialized as a constant, *i.e.*, $\overline{R}(\cdot) = 1$. Instead of directly learning the rewards from observed state-action pairs, we employ such *hierarchical* structure for two reasons: 1) the decision-making patterns across different sub-trajectories are of different importance, while directly learning from state-action pairs will treat them equally; 2) learning from state-action pairs in a flattened structure cannot capture the transitional patterns across different policies.

Based on the inferred rewards for each state-action pair, we further define a function $f(\cdot)$ as shown in Eq. 5.3 to reflect whether the consecutive states have the same decision-making patterns, with its scale indicating the importance of such patterns: $f(\overline{r}_t^n) = 1/\omega \sum_{i=\omega}^{0} \overline{r}_{t-i}^n$. By minimizing Eq. 5.3, the neighboring states belonging to different clusters will be penalized. To balance the effects of time-awareness patterns, *i.e.*, $log(e+\Delta T_t^n)$, and decision-making patterns, *i.e.*, $f(\overline{r}_t^n)$, here we employ a bivariate Gaussian distribution $\Phi(.)$ to formulate their interactional regularizations.

• *Sparsity term* controls the sparseness based on a l_1-norm, *i.e.*, $\lambda||\mathbf{\Theta}_k||_1$, with λ being a coefficient. It can select the most significant variables to effectively prevent overfitting.

To solve the objective function Eq. 5.1, we employ the EM to learn the cluster assignments $\mathbf{P}$ and the structural patterns $\mathbf{\Theta}$ iteratively until convergence. Specifically, *in E-step*, by fixing $\mathbf{\Theta}$ to learn $\mathbf{P}$, Eq. 5.1 degenerates into a form with only the log-likelihood term and the consistency term. It can be solved by dynamic programming to find a minimum cost *Viterbi* path [Vit67]; *In M-step*, by fixing $\mathbf{P}$ to learn the $\mathbf{\Theta}$, Eq. 5.1 degenerates into a form with only the log-likelihood term and the sparsity term. It can be formulated as a typical graphical lasso problem [Fri08] with a Toeplitz constraint over $\mathbf{\Theta}$ and be solved by an alternating direction method of multipliers (ADMM) [Boy11].

5.3.3 EM-EDM: Inducing Policies over the Sub-trajectories

5.3.3.1 Preliminaries

We first assume the demonstrated trajectories follow a single reward function. Herein, for simplicity, we represent the state-action pairs as $(\mathbf{x}, a)$ by omitting their indexes when it does not cause ambiguity. Denote π_θ as the policy to be learned parameterized by θ. ρ_D and ρ_{π_θ} are the occupancy measures for the demonstrations and the learned policy, respectively. The probability density for state-action pairs can be measured as: $\rho_{\pi_\theta}(\mathbf{x}, a) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}_{\{\mathbf{x}_t=\mathbf{x}, a_t=a\}}]$, then the probability density for the states can be measured via: $\rho_{\pi_\theta}(\mathbf{x}) = \sum_a \rho_{\pi_\theta}(\mathbf{x}, a)$. To induce the π_θ, our goal is: $\text{argmin}_\theta D_{KL}(\rho_D || \rho_\theta) = \text{argmin}_\theta -\mathbb{E}_{\mathbf{x},a\sim\rho_D} \log \rho_\theta(\mathbf{x}, a)$. Considering $\pi(a|\mathbf{x}) = \rho_\pi(\mathbf{x}, a)/\rho_\pi(\mathbf{x})$, we can formulate the objective function as:

$$\mathrm{L}(\theta) = -\mathrm{E}_{\mathbf{x}\sim\rho_D} \log \rho_{\pi_\theta(\mathbf{x})} - \mathrm{E}_{\mathbf{x},a\sim\rho_D} \log \pi_\theta(a|\mathbf{x}) \tag{5.4}$$

When there is no access to roll out the policy π_θ with an *online* manner, the first term $\rho_{\pi_\theta}(\mathbf{x})$ in Eq. 5.4 is difficult to estimate. To handle this, Jarrett *et al.* proposed an energy-based distribution matching (EDM) [Jar20]. Based on energy-based models (EBMs) [Gra19], the probability density $\rho_{\pi_\theta}(\mathbf{x}) \propto e^{-E(\mathbf{x})}$. The occupancy measure for state-action pairs can be represented as: $\rho_{\pi_\theta}(\mathbf{x}, a) = e^{f_{\pi_\theta}(\mathbf{x})[a]}/Z_{\pi_\theta}$, then the occupancy measure for states can be obtained by marginalizing out the a: $\rho_{\pi_\theta}(\mathbf{x}) = \sum_a e^{f_{\pi_\theta}(\mathbf{x})[a]}/Z_{\pi_\theta}$. Herein, Z_{π_θ} is a partition function, and f_θ denotes the logits for action conditionals. Then the parameterization of π_θ implicitly defines an EBM of state visitation distributions, with the energy function $E_\theta : \mathbb{R}^{|X|} \to \mathbb{R}^{|\mathbb{A}|}$ defined as: $E_\theta(\mathbf{x}) = -\log \sum_a e^{f_\theta(\mathbf{x})[a]}$. Under the scope of EBMs, the first term in Eq. 5.4 can be reformulated as an "occupancy" loss:

$$\mathscr{L}_\rho(\theta) = \mathbb{E}_{\mathbf{x}\sim\rho_D} E_\theta(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\sim\rho_\theta} E_\theta(\mathbf{x}) \tag{5.5}$$

where $\nabla_\theta \mathscr{L}_\rho(\theta) = -\mathbb{E}_{\mathbf{x}\sim\rho_D} \nabla_\theta \log \rho_\theta(\mathbf{x})$ [Jar20] can be solved by existing optimizers, *e.g.*, stochastic gradient Langevin dynamics (SGLD) [Wel11]. Thus, by substituting the first term in Eq. 5.4 as Eq. 5.5, we can derive a *surrogate objective function* to derive the optimal solution without the need of *online* rolling out the policy. Specifically, the EDM algorithm is strictly *offline*, without requiring any knowledge of model transitions or conducting off-policy evaluations, also it can efficiently handle continuous state spaces. Despite the power of EDM in *offline* learning, it assumes the demonstrations to follow a *homogeneous* reward function, which is infeasible for many real-world human-centric tasks, *e.g.*, healthcare.

5.3.3.2 EM-EDM

To deal with *multiple* reward functions varying across the demonstrations, Babes-Vroman *et al.* [Bab11] proposed an EM inverse reinforcement learning (EM-IRL), by iteratively learning to cluster the demonstrations in *E-step* and inducing pollies for each cluster by IRL in *M-step*. In EM-IRL, an offline maximum-likelihood IRL (MLIRL) [Bab11; Yan20] was employed, which models discrete

states – not scalable for large state spaces. Enlightened by the recent success of the EDM, based on the EM-IRL framework, we propose an EM-EDM, by upgrading the MLIRL to EDM in the *M-step*.

For both EM-IRL and EM-EDM, the input is a set of trajectories $\{\xi_i\}$. Denote G as the number of clusters and N_g as the number of trajectories in g-th cluster, respectively. ρ_g and θ_g, $g = 1, .., G$, are randomly initialized, denoting the prior probability and the policy parameter for the g-th cluster. The objective function is to maximize the log-likelihood (LL) as specified in Eq. 5.6:

$$LL = \sum_{g=1}^{G} \sum_{i=1}^{N_g} \log(u_{ig}) \tag{5.6}$$

$$u_{ig} = Pr(\xi_i|\theta_g) = \prod_{(\mathbf{x},a)\in\xi_i} \frac{\pi_{\theta_g}(\mathbf{x},a)\rho_g}{U} \tag{5.7}$$

In *E-step*, the probability that trajectory ξ_i belonging to cluster g is calculated by Eq. 5.7, in which U is a normalization factor; Then in *M-step*, the prior probabilities are updated via $\rho_g = \sum_i u_{ig}/N_g$. In EM-IRL, the policy parameters θ_g can be learned via MLIRL; while in EM-EDM, we learn the θ_g by EDM. The *E-step* and *M-step* are iteratively executed until converged. Finally, the output of EM-EDM is the clustered trajectories with their respective policies.

Note that when applying EM-EDM to THEMES, the input $\{\xi_i\}$ are MT-TICC-learned sub-trajectories. Except for the difference in M-step, EM-IRL can return the policy as well as the reward function for each cluster, while EM-EDM will directly learn the policy. That is the reason we employ the EM-IRL to learn the reward regulator $\overline{R}(\cdot)$. Since our high-level states are discrete, the EM-IRL can be effectively conducted to infer the reward functions.

5.4 Experiments with Healthcare Data

5.4.1 Data Preprocessing

5.4.1.1 EHRs Dataset

The EHRs dataset we utilized to evaluate the proposed THEMES comes from the same CCHS sepsis-related study cohort as the one we employed in Section 3.5.1. Specifically, following the same data preprocessing procedures, *i.e.*, selecting features, handling missing data, and tagging the septic shock visits, finally, our dataset contains 3,738 visits (1,869 shock and 1,869 non-shock) with 145,421 events as shown in Table 3.5. Based on this selected cohort, we defined states and actions as follows:

• **States**: Here we selected 14 sepsis-related continuous features, including 1) *Vital signs*: systolic blood pressure (SBP), mean arterial pressure (MAP), respiratory rate (RR), oxygen saturation (PulOx), heart rate (HR), temperature (Temp); 2) *Lab results*: white blood cell count (WBC), bilirubin (Bili), blood urea nitrogen (BUN), lactate (Lac), creatinine (Creat), platelet (Plat), neutrophils (Bands); and 3) *Intervention*: fraction of inspired Oxygen (FiO2).

Based on the observed features, previous works generally employed an aggregation manner to

represent the states. For example, based on a Medical Information Mart for Intensive Care [Joh16] EHRs dataset, [Kom18; Jar20; Cha21] averaged the observed measurements per 24 hours to represent the states, and [Wan21; Wan22] chose the window as 4 hours to aggregate the states. To avoid the large aggregation window over-smoothing the state representation, in this work, we employed a much smaller aggregation window of 10 minutes. Additionally, when aggregating the observations, we considered not only the averaged values but also the maximum and minimum values that occurred in the past 24 hours. By doing so, our states turn out to be 42-dim continuous vectors.

• **Actions**: We employed binary actions (1/0) to indicate whether any antibiotic administration (*e.g.*, clindamycin, daptomycin) is taken or not. As suggested in [Gau13], antibiotic therapy plays a critical role in improving the clinical outcomes for sepsis patients. The medical interventions are usually sparse over the trajectories. For example, in our dataset, the original ratio of the two actions (1/0) is $1:6.4$. For two closely consecutive timestamps, their underlying states can be quite similar; therefore, the action taken over one of them can be replicated in another. As suggested by [Gau13], early antibiotic therapy can improve clinical outcomes and should be given within *1 hour* of suspected sepsis. Therefore, to balance the two actions, we carried forward the positive action, *i.e.*, the usage of antibiotics, for an hour.

5.4.1.2 Positive vs. Negative Trajectories

One major assumption in AL is that the experts are performing optimally or near-optimally driven by the reward functions [Abb04a]. As a result, the quality of experts' demonstrations matters in order to induce an accurate policy. If there are many imperfect or noisy behaviors, the learning model would be confused and hardly converge. When analyzing the EHRs data by AL, most of the previous works, *e.g.*, [Wan21; Jar20; Cha21; Wan22], utilized all available EHRs for policy induction, without considering whether the interventions are carried out optimally/near-optimally or not. Recently, in [Wan20b], the authors distinguished the *Positive* trajectories, *i.e.*, survived patients, from the *Negative* trajectories, *i.e.*, deceased patients, when learning the AL model. An adversarial cooperative imitation learning model was proposed to learn the treatment policy, which aimed at learning a policy to mimic the *Positive* trajectories while staying far away from the *Negative* ones. However, the *Negative* trajectories may not always have imperfect actions, thus when they behave similarly to the *Positive* data, the model will be confused.

As suggested by [Jar20], the EDM algorithm is capable of inducing policy from a small dataset. Taking this advantage, we considered improving the quality of the demonstrations by filtering out the *Negative* trajectories. Specifically, we distinguished the *Positive* vs. *Negative* trajectories based on a septic shock early prediction task. As illustrated in Table 5.1, comparing the prediction results *before the first treatment* versus the *onset of shock*: if a patient turns from shock to non-shock, then the treatments are considered to be effective and the trajectory is identified as *Positive*; Otherwise, if a patient develops from non-shock to shock, then the trajectory is identified as *Negative*.

Herein, we employed long-short term memory [Hoc97] as the early prediction model. The overall framework for distinguishing the *Positive* vs.*Negative* trajectories are detailed in Table 5.2.

Table 5.1 Rules for distinguishing the *Positive* & *Negative* trajectories by comparing septic shock early prediction using the data *truncated before the first treatment* vs. the *onset of shock/non-shock*.

		Onset of Shock/Non-shock	
		Non-shock	Shock
Truncated before	Non-shock	–	Negative
the First Treatment	Shock	Positive	–

Specifically, given a buffer $\mathscr{B}$ with all available trajectories, we followed three steps to get the *Positive* trajectories: 1) filtering out the ambiguous data from the input buffer $\mathscr{B}$ and get a subset of $\mathscr{B}'$; 2) Learning the early prediction results over the truncated trajectories in the filtered buffer $\mathscr{B}'$; and 3) Distinguishing the *Positive* & *Negative* trajectories in the filtered buffer $\mathscr{B}'$.

5.4.2 Experimental Setup

5.4.2.1 Positive vs. Negative Demonstrations

We applied the THEMES to partition the sub-trajectory clusters and learn the clinician's decision-making patterns in each cluster, simultaneously. Specifically, we fixed the policy induction model to be EDM and compared different combinations of training and test data. When the test data is set as either Positive/Negative, we compare the training data of: 1) **Positive**: trajectories transiting from Shock at *early prediction before the first treatment* to Non-shock *at the onset*; 2) **Positive+Negative**: trajectories either from Shock to Non-shock (Positive) or from Non-shock to Shock (Negative); 3) **Random**: Randomly chosen trajectories, which can be any one of the four cases, *i.e.*, Shock to Non-shock, Non-shock to Shock, Shock to Shock, and Non-shock to Non-shock; 4) **Negative**: trajectories from Non-shock to Shock. For each training and test data combination, we keep the number of training and test trajectories to have the same amount, *i.e.*, 156 trajectories for training and 39 for testing, and repeat experiments with each combination for 500 times.

The results for filtering the demonstrations are first evaluated over the *event-level* – once a policy is induced, we will compare whether the suggested action is consistent with the real action in the test set. The results are evaluated using metrics: Accuracy (ACC), Recall (Rec), Precision (Prec), F1-Score (F1), and AUC. Besides, we calculated the average precision (AP) and Jaccard scores. The AP will summarize a precision-recall curve as the weighted mean of precision achieved at different thresholds, where the increases in recall between consecutive thresholds are used as weights. The Jaccard is a similarity coefficient score. It is calculated as the intersection divided by the union: $\frac{|y \cap y'|}{|y \cup y'|}$, where the y and $y^{'}$ stand for the actual and predicted actions, respectively.

Then we further evaluated the performance over the *visit-level* with two metrics of Accuracy (Acc) and a self-defined Early Time Before the First Treatment (ETBFT). Herein, the Acc for each trajectory is defined as the percentage of predicted actions aligned with the actual actions. The ETBFT measures how early the first treatment can be suggested before the actual first treatment is taken, with the unit of hours. A larger ETBFT indicates the actual first treatment is more delayed.

Table 5.2 The framework for distinguishing the *Positive* & *Negative* trajectories.

Framework: Distinguishing the *Positive* & *Negative* Trajectories
Input: A buffer $\mathscr{B}$ with all trajectories
Output: A filtered buffer $\mathscr{B}'$ with the labels of *Positive* & *Negative*
1. *Filtering out the ambiguous data from the buffer* $\mathscr{B}$
– For the trajectories (`traj`) before the onset
1) Build LSTMs with `traj` to predict Shock/Non-shock for 100 times
2) Predict the label via majority-voting among the 100 LSTMs
3) Compare predicted label vs. ground-truth label at the onset:
· **If** correct → Keep the data in $\mathscr{B}$
· **Else** → Remove the data from $\mathscr{B}$
– For the *truncated* trajectories (`traj_trunc`) before the first treatment
While *iteration_num* < 10:
1) Build LSTMs with `traj_trunc` to predict Shock/Non-shock for 100 times
2) Predict the label via majority-voting among the 100 LSTMs
3) Check the percentage of predicted Shock (*Shock_perc*) among the 100 LSTMs:
· **If** *Shock_perc* < 0.2 or *Shock_perc* > 0.8 → Keep the data in $\mathscr{B}$
· **Else** → Remove the data from $\mathscr{B}$
4) Check the convergence:
· **If** Δ*Shock_perc* < 1e-3 → Break
– Get a filtered buffer $\mathscr{B}'$ after conducting the above two steps
2. *Retraining the early prediction over* `traj_trunc` *in* $\mathscr{B}'$
– Build LSTMs with `traj_trunc` to predict Shock/Non-shock for 100 times in $\mathscr{B}'$
– Get the early prediction labels (`early_pred`) among 100 LSTMs via majority-voting
3. *Distinguish the Positive & Negative Trajectories in* $\mathscr{B}'$
– Compare the early predicted label `early_pred` vs. ground-truth onset label (`onset`):
· **If** `early_pred` = Shock & `onset` = Non-shock → *Positive*
· **Elif** `early_pred` = Non-shock & `onset` = Shock → *Negative*
· **Else** → Excluded

5.4.2.2 Policy Induction

Based on the filtered *Positive* trajectories, to evaluate the effectiveness of the THEMES, we compare it against the following baseline methods:

• Two traditional methods, including: 1) **GP+DQN**, which employs the Gaussian process (GP) [Azi18] to infer the rewards for each timestamp from the outcomes, *i.e.*, shock/non-shock, and then apply the deep q-network (DQN) [Fan20] to learn the policy; and 2) **LSTM** [Zha19], which is a behavior cloning for learning a mapping from states to actions.

• The state-of-the-art offline **EDM** [Jar20] assumes the demonstrations to follow a *single* reward function. Since it has been demonstrated in EDM work that it can outperform many other competitive cutting-edge AL methods with a *single* reward function, we will not repetitively conduct more comparisons here.

• Our **EM-EDM**, which assumes the input data follow *multiple* reward functions varying across trajectories (while remaining the same within each trajectory).

• Two state-of-the-art methods assuming *multiple* reward functions *evolving* over time, including: 1) **Adapted HIRL**, which is an Adapted hierarchical inverse reinforcement learning based on a framework proposed in [Kri16], by learning sub-trajectory clusters utilizing Gaussian mixture model first and then applying EDM to learn the cluster-wise policies; and 2) **MIL**, which is a multi-modal imitation learning [Hau17], using infoGAN to learn the sub-trajectories with their respective policies.

• THEMES and its ablation methods, which assume the demonstrations following *multiple* reward functions *evolving* over time, including 1) **MT-TICC+EDM**, with the policy for each MT-TICC-learned sub-trajectory cluster learned by EDM; 2) **MT-TICC+EM-EDM**, which more flexibly assumes the MT-TICC-learned sub-trajectory clusters can follow *multiple* reward functions learned by EM-EDM; and 3) **THEMES** which utilizes RMT-TICC to partition the trajectories with hierarchically learned reward regulator and induces policies over the sub-trajectory via EM-EDM.

Each method was repeated 10 times by randomly splitting 80% data for training and 20% for testing. We employ the metrics of Accuracy (Acc), Recall (Rec), Precision (Prec), F1-score (F1), AUC, Average Precision (AP), and Jaccard score to evaluate the performance. All the model parameters are determined by 5-fold cross-validation. Specifically, in THEMES, for RMT-TICC, based on Bayesian information criteria (BIC) [Fri01], we determine the cluster number K as 11, the window size ω as 2, and the sparsity and consistency coefficients λ and β as 1e-5 and 4, respectively. For EM-EDM, the optimal cluster number is determined heuristically as 3, by iteratively implementing the EM, until empty clusters are generated or the log-likelihood of the clustering results varied smaller than a pre-defined threshold. The iterations for the overall THEMES framework have the Thres = 10, based on our observation that the clustering likelihood for both MT-TICC and EM-EDM converge within 10 iterations. For fair comparisons, the optimal parameters in other baseline methods are determined by across-validation as well.

Table 5.3 Comparing the training data of Positive vs. Positive+Negative vs. Random vs. Negative when fixing the test data as either Positive or Negative (*event-level*).

Training	Test	Acc	Rec	Prec	F1	AUC	AP	Jaccard
Positive	Positive	**.749(.020)**	**.748(.080)**	**.480(.037)**	**.581(.040)**	**.821(.012)**	**.715(.015)**	**.411(.037)**
Positive+Negative		.738(.020)*	.749(.097)	.466(.035)*	.570(.047)*	.812(.013)*	.704(.014)*	.400(.041)*
Random		.737(.021)*	.747(.101)	.464(.033)*	.568(.047)*	.811(.013)*	.703(.013)*	.398(.040)*
Negative		.722(.023)*	.744(.119)*	.447(.035)*	.553(.059)*	.792(.015)*	.686(.015)*	.384(.049)*
Positive	Negative	.702(.030)	.595(.080)	.456(.052)	.513(.055)	.748(.030)	.671(.025)	.347(.048)
Positive+Negative		**.703(.030)**	.647(.103)*	.461(.048)*	.533(.057)*	.760(.028)*	.682(.023)*	.365(.050)*
Random		.702(.030)	.645(.106)*	.456(.047)	.529(.059)*	.758(.028)*	.679(.023)*	.362(.051)*
Negative		**.703(.003)**	**.663(.131)***	**.463(.046)***	**.537(.064)***	**.767(.027)***	**.691(.022)***	**.369(.056)***

⋆ p-value < 0.01 compared to the Positive training data when test data is Positive/Negative.

Table 5.4 Comparing the training data of Positive vs. Positive+Negative vs. Random vs. Negative when fixing the test data as either Positive or Negative (*visit-level*).

Training	Test	Acc	ETBFT
Positive	Positive	**.735(.082)**	5.409(16.976)
Positive+Negative		.722(.084)*	**5.506(16.806)**
Random		.720(.084)*	5.292(16.435)
Negative		.702(.087)*	4.920(16.241)*
Positive	Negative	.606(.207)	**12.130(55.717)**
Positive+Negative		.622(.190)*	11.735(55.000)
Random		.627(.201)*	11.227(55.371)
Negative		**.644(.175)***	10.562(55.655)

⋆ p-value < 0.01 compared to the Positive training data when test data is Positive/Negative.

5.4.3 Results

5.4.3.1 Positive vs. Negative Demonstrations

The results at the event-level are reported in Table 5.3. According to the table, when fixing the test data to be Positive and training with different combinations of data, the training with Positive got the best performance. This indicates the Positive trajectories we found are more compatible. Likewise, when we fixed the test data to be Negative and trained with different combinations, training with Negative got the best performance. This indicates the Negative trajectories that we identified are more comparable. Then we further compared both the training and testing to be Positive or Negative, we can see that using Positive for both training and testing performed better. This indicates the Positive trajectories are more compatible while the Negative trajectories are relatively more diverse.

Table 5.4 reports the experimental results at the visit-level. When fixing the test data to be

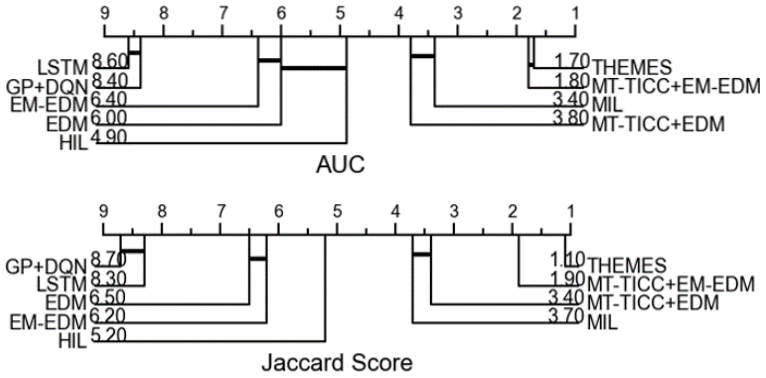


Figure 5.2 Critical Difference Diagram with Wilcoxon Signed-rank Test over AUC and Jaccard Scores.

Table 5.5 Comparing THEMES with baseline methods.

Methods	Acc	Rec	Prec	F1	AUC	AP	Jaccard
GP+DQN	.588(.042)	.342(.033)	.246(.045)	.286(.045)	.552(.037)	.533(.015)	.167(.028)
LSTM	.553(.043)	.441(.080)	.244(.034)	.312(.044)	.537(.033)	.522(.014)	.186(.031)
EDM	.740(.022)	.760(.058)	.462(.034)	.572(.014)	.817(.014)	.709(.013)	.400(.014)
EM-EDM	.748(.018)	.748(.036)	.473(.033)	.578(.020)	.807(.012)	.702(.014)	.407(.019)
Adapted HIRL	.788(.014)	.733(.016)	.520(.015)	.608(.012)	.844(.014)	.737(.011)	.437(.012)
MIL	.803(.013)	.762(.042)	.538(.032)	.631(.012)	.872(.012)	.768(.013)	.476(.014)
MT-TICC+EDM	.836(.019)	.659(.045)	.650(.037)	.646(.014)	.869(.011)	.808(.012)	.481(.014)
MT-TICC+EM-EDM	.865(.013)	.781(.058)	.678(.038)	.723(.012)	.904(.010)	**.851(.010)**	.566(.015)
THEMES	**.867(.012)**	**.830(.010)**	**.685(.020)**	**.751(.010)**	**.905(.008)**	.838(.007)	**.600(.013)**

Positive, the better performance for Positive training data indicates high compatibility among the Positive data. Meanwhile, we found that Positive+Negative can give earlier suggested treatment with a larger ETBFT, while there's no significant difference between the Positive+Negative vs. Positive. When fixing the test data to be Negative, similarly, we found testing with Negative performed the best, which indicates the Negative data is more compatible with itself. Meanwhile, training with Positive data can give earlier suggested treatment with a larger ETBFT. Finally, as we can see, testing with Negative data in general got much higher ETBFT than Positive data, which means their first treatment is much more delayed compared to Positive data.

5.4.3.2 Policy Induction

The results for policy induction are reported in Table 5.5, and we also report the critical difference diagram [IF19] with Wilcoxon signed ranks tests over AUC and Jaccard scores in Figure 6.1: the unconnected models mean pairwise significance, with the confidence level of 0.05. The results show that THEMES in general performs the best. Specifically, it outperforms the MT-TICC+EM-EDM, which indicates the effectiveness of the hierarchically learned reward regulator to incorporate the

decision-making patterns during partitioning. Meanwhile, our THEMES, MT-TICC+EM-EDM, and MT-TICC+EDM outperform the competitive baselines of Adapted HRL and MIL, which has the same assumption of *multiple* reward functions *evolving* over time. It indicates the effectiveness of RMT-TICC and MT-TICC in capturing *time-awareness* during partitioning. Comparing THEMES and its ablation methods (*i.e.*, MT-TICC+EM-EDM and MT-TICC+EDM) against EM-EDM, there are large improvements, which indicates the power of RMT-TICC and MT-TICC in contributing to capture the time-varying progressive patterns across the partitioned sub-trajectories. Additionally, Adapted HIL, MIL, MT-TICC+EDM, MT-TICC+EM-EDM, and THEMES perform better than other methods, which indicates *multiple* reward functions *evolving* over time can be a better way for modeling human-centric tasks such as healthcare. Likewise, EDM and EM-EDM do not show significant differences, and their performances are not as good as THEMES, MT-TICC+EM-EDM, and MT-TICC+EDM, which indicates the assumption of *multiple across-heterogeneous* reward functions (remaining the same within each trajectory) cannot fully model the time-varying patterns.

5.5 Summary

In this chapter, we aim at modeling the **evolving-heterogeneous** reward functions by AL. Specifically, we propose a **THEMES** framework to induce policies from the experts' demonstrations (*Positive*) in an *offline* manner when there are *multiple* reward functions *evolving* over time. The effectiveness of the proposed framework is evaluated via a healthcare application to induce policies from clinicians when treating an extremely challenging disease, *i.e.*, sepsis. The results demonstrate that THEMES can outperform competitive baselines in policy induction: Empowered by time-awareness and incorporated decision-making patterns encoded in hierarchically learned rewards, more fine-grained sub-trajectory partitions can be learned, based on which more accurate policies can be induced.

6 INCORPORATING THEMES FOR DISEASE PROGRESSION MODELING

In this chapter, taking advantage of the **THEMES** framework proposed in Chapter 5, we incorporate the THEMES-learned patterns for a disease progression modeling (**DPM**) task. DPM plays an important role in characterizing historical progressive pathways and predicting potential future risks. Herein, to conduct DPM, we investigate incorporating the patterns extracted from Apprenticeship Learning (AL), which induces decision-making policies via observing and imitating expert clinicians' demonstrations. More specifically, given the nature of DPM, we employ the THEMES clustering to extract these patterns. To the best of our knowledge, this is the first work incorporating AL-derived intervention patterns for DPM. The effectiveness of THEMES is evaluated on the task of early prediction for an extremely challenging condition, *i.e.*, septic shock. The results suggest that by incorporating the THEMES-derived features, more refined DPM can be achieved, as the septic shock can be predicted more accurately in earlier stages, which can shed some light on assisting clinicians to carry out personalized treatments in time.

6.1 Overview

Disease progression modeling (DPM) aims at characterizing patients' longitudinal medical records, identifying disease progressive stages, and evaluating factors affecting the progression pathways [Suk12; Coo16]. Empowered by the increasing availability of large medical datasets, *e.g.*, electronic health records (EHRs), various deep learning models [Cho16a; Cho16b; Zha17] have been developed for DPM based on the *recorded observations* in EHRs, *e.g.*, vital signs and laboratory test results. Nevertheless, merely relying on monitored observations may not fully characterize the disease progression as many diseases don't have clear observations and even medical experts often miss them [Tin85]. Meanwhile, different patient groups may also exhibit different observations of the same disease: a common sign that appears abnormal in one group may be normal in another [Bay17]. Some recent DPM works have used *sub-trajectory clustering* for deriving *progressive stages* automatically in a data-driven manner from various observations [Gao22], *e.g.*, our MT-TICC proposed in Chapter

3 [Yan21]. Other than using observations for DPM, *interventions* also have a significant impact on disease progression [Kom18; Azi19], and some prior works have taken clinicians' interventions as additional features for DPM [Est16; Goh21]. Clinical interventions are usually carried out based on prior medical knowledge, which is difficult to be conveyed by observation alone. Additionally, interventions tend to take hours or days to manifest in observations [DA20]. However, individual clinical interventions on DPM cannot be fully explained without assessing their effectiveness, since different patient groups may respond to the same treatment differently and patients at different stages of DPM may respond differently to the same treatment [Cli16]. For this reason, when learning progressive patterns in DPM, it is essential to incorporate the *effectiveness of interventions*.

Reinforcement learning (RL), especially deep RL, has shown great success in effectively inducing clinicians' interventional policies [Wan18; Azi19; Yu21]. A common challenge when applying RL to healthcare is the design of the *reward function*, which serves as an incentive to induce the policy [Azi19]. Apprenticeship learning (AL) was proposed to address this issue [Abb04b]: instead of taking an explicitly delineated reward function as input, AL can learn the policy by imitating the demonstrated behaviors provided by clinicians. Recently, an AL method named energy-based distribution matching (EDM) has shown great success in inducing effective clinical intervention policies directly from EHRs [Jar20]. In EDM, the demonstrations given by clinicians are assumed to be generated with a uniform policy, latently driven by a *single* reward function; while when managing patients under various progressive stages, such as common fevers or severe organ failures, clinicians may adopt hierarchical policies with *multiple rewards* [Wan21; Wan22].

In this chapter, we *incorporate the interventional patterns extracted by a hierarchical AL with evolving reward functions* for DPM. Specifically, we employ the THEMES framework introduced in Chapter 5, which can incorporate the *interventional* patterns learned by AL, taking the power of *sub-trajectory clustering*. Referring to the progressive stages indicated by sub-trajectory clustering, more fine-grained interventional patterns that evolve over time can be extracted by AL; Meanwhile, the derived interventional patterns can in turn contribute to refining the sub-trajectory clustering results to indicate more accurate progressive stages.

The effectiveness of the THEMES framework in DPM is evaluated by modeling an extremely challenging disease – *sepsis*. As specified in Section 3.1, sepsis is a life-threatening organ dysfunction and a leading cause of death worldwide [Sin16]. Without giving timely interventions, patients can progress to septic shock, which is the most severe condition with a mortality rate as high as 50% [Mar03]. Contrarily, 80% of sepsis deaths can be prevented if proper diagnosis and interventions can be conducted in time [Kum06]. Therefore, modeling the sepsis progression to more accurately predict the septic shock in an early stage is a crucial task. Based on THEMES, we take the latent *progressive stages* extracted by RMT-TICC as well as *interventional* patterns derived by EM-EDM as supplemental features to the *observations* in septic shock early prediction. Our results show that by incorporating such additional features, more accurate prediction can be achieved in earlier stages.

The contributions in this work are three-folded: 1) To the best of our knowledge, this is the first work incorporating AL-derived interventional patterns for DPM; 2) We effectively model the

evolving reward functions in an offline manner for healthcare data taking advantage of our THEMES, which consists of two key components for taking the power of sub-trajectory clustering and robustly learning the interventional patterns by AL; 3) We deploy the THEMES to a challenging disease, *i.e.*, sepsis, for learning its progressive stages and predicting its most severe condition, *i.e.*, septic shock, in early stages, which can shed light on more personalized timely treatments.

6.2 Related Work

6.2.1 Disease Progression Modeling (DPM)

Disease progression modeling (DPM) is a task of monitoring the time course of disease stages and tracking disease severity over time, which is crucial for forecasting future risks and providing prompt medications [Suk12; Coo16; Sev20]. Recently, the extensively collected longitudinal information in electronic health records (EHRs) has provided a rich source of data for DPM. Based on EHRs, plenty of previous works have been developed based on deep learning models. Especially, the recurrent neural networks-based approaches [Lip15; Saq18; Soh20b], *e.g.*, long-short term memory (LSTM), which can capture the sequential patterns over a long period of historical records, have achieved great success. In general, these deep learning models take monitored *observations* as input and rely on neural networks to extract the latent progressive patterns for DPM.

To better capture the progressive stages, there have been some *sub-trajectory clustering* approaches proposed, aiming at partitioning and clustering the sub-trajectories, simultaneously. As specified in Section 3.2.2, in general, the sub-trajectory clustering approaches can be categorized into *distance-based*, *e.g.*, dynamic time warping [Gia18], and *model-based*, *e.g.*, Gaussian mixture models [Far21] and hidden Markov models [Sun19; Kwo20]. Compared to distance-based, model-based methods are usually more reliable, since they can introduce the structures of the data. Recently, a model-based Toeplitz inverse covariance-based clustering (TICC) [Hal17] has attracted attention in accurately partitioning sub-trajectories in various applications, *e.g.*, analyzing physical activities for patients with Alzheimer's [Li18] and segmenting the critical stages for sepsis patients [Gao22]. Based on TICC, in Chapter 3, we proposed a multi-series time-aware TICC (MT-TICC) [Yan21], by taking multi-series as input and using the time-awareness mechanism to deal with irregular intervals in EHRs. The MT-TICC achieved preferable performance in DPM: by supplementing the MT-TICC-derived patterns to observations, the septic shock can be predicted more accurately in earlier stages, outperforming competitive baselines including the original TICC.

The above techniques mainly focus on *observations* monitored from the patients. Other than observations, clinicians' *interventions* have shown a significant impact on disease progressions [Kom18; Azi19]. Some previous works have explored incorporating interventions in DPM by directly taking them as additional features [Est16; Lin19; Goh21]. Since clinicians usually carry out interventions via a series of actions in an interactive manner, a more robust way to extract the interventional patterns is highly desired.

6.2.2 RL & AL for EHRs

Reinforcement learning (RL) has been widely applied in EHRs for dynamic treatment regimes, which aims at inducing a decision-making policy to dictate how the interventions should be executed, so that the patients can gain improved outcomes [Yu21]. As an input of RL, the reward function plays a critical role in praising/punishing the learning model to derive an optimal policy. However, manually specifying an appropriate reward function is usually expertise-intensive and time-consuming, which poses a significant barrier to the broader applicability of RL [Abb04b]. As introduced in Section 2.2, apprenticeship learning (AL) [Ng00] was proposed to handle this issue: instead of taking an explicitly specified reward function as input, AL learns from a set of demonstrations, which are assumed to be optimally executed following an implicit reward function. The most intuitive AL method is behavior cloning [Raz12], *i.e.*, building up a mapping from states to actions to greedily imitate the demonstrated behaviors, whereas it highly relies on both the quality and quantity of the demonstrations [Ros11]. To more robustly learn from the demonstrations, some inverse RL (IRL) [Abb04b; Zie08] and adversarial imitation learning [Ho16; Fin16] approaches have been proposed. However, these methods are commonly *online*, which requires iteratively executing the latest policy to collect data for updating the model. Since the execution of a bad policy can be costly or even dangerous in healthcare [Lev20], it is highly desired to learn the model purely based on demonstrations in an *offline* manner. Though some online methods have been adapted to offline [Cha21; Kos18], they usually rely on off-policy evaluation, which itself is a nontrivial problem with imperfect solutions.

A recently proposed AL approach named energy-based distribution matching (EDM) [Jar20] can cope with the *offline* setting well by fully exploiting the data distribution in the given demonstrations. It has got state-of-the-art performance compared to competitive offline methods. When applying the EDM to EHRs for modeling the clinicians' interventions, it poses a strong assumption that all demonstrations are generated with a unified policy, following a *single* reward function. However, when treating patients under different disease progressive stages, clinicians will adaptively execute different policies with *multiple* reward functions [Wan21; Wan22]. To model such *multiple* reward functions, some AL approaches have been proposed [Dim11; Cho12; Bab11]. For example, [Bab11] developed an EM-based IRL (EM-IRL), which assumes reward functions to be diverse across different demonstrations, while within each demonstrated sequence, the reward function is still assumed to be unified. To further model the multiple reward functions *evolving* over time, some more recent works have been proposed [Kri16; Hau17; Wan21; Wan22]. However, these methods generally partition the demonstrations into *fixed-length* sub-trajectories to learn their respective reward functions; Meanwhile, the irregular intervals are not considered during the partitioning.

Furthermore, the existing AL generally learns the interventional patterns for either *inducing* a decision-making policy or *evaluating* whether the interventions are carried out in an expected manner. None of them have been incorporated for learning the progressive stages during the DPM.

6.3 Experiments with Healthcare Data

The EHRs dataset we utilized for DPM analysis comes from the same CCHS sepsis-related study cohort as the one we employed in Section 3.5.1. Specifically, following the same data preprocessing procedures, we narrowed down the cohort to 3,738 visits (1,869 shock and 1,869 non-shock) with 145,421 events. Based on the selected demonstrations, we defined states and actions the same way as Section 5.4.1.1. Following the framework for filtering the demonstrations introduced in Section 5.4.1.2, 195 trajectories are identified as the experts' demonstrations.

6.3.1 Experimental Setup

Based on the output of THEMES, we extract additional features and supplement them to the observed state features for septic shock early prediction. Herein, we fix the long short-term memory (LSTM) as the prediction model, considering that extensive works have demonstrated its preferable performance in EHRs modeling [Lip15; Zha17; Lin18], then different input features generated by various methods will be compared, including:

• **Observations**: Only observations are employed during the DPM, without considering the impact of interventions: 1) **Original:** Except for the *Original* observed state features, none of the other additional information will be incorporated; 2) **MT-TICC:** Observed state features are supplemented with additional progressive features learned by *MT-TICC* [Yan21]. The additional features are extracted as the probabilities belonging to each sub-trajectory cluster.

• **Observations+Interventions** Both observations and interventions are employed for DPM: 1) **Action:** Interventional *Actions* are directly taken as additional features for the observed state features; 2) **EDM:** Based on the policy learned by *EDM* [Jar20], an additional feature is extracted as the probability following such policy; 3) **MT-TICC+EM-EDM (Proposed):** The probabilities belonging to *MT-TICC*-learned sub-trajectory clusters as well as the probabilities following *EM-EDM*-learned policies are taken as additional features. Here the interventional patterns will not feed back to refine the sub-trajectory clustering; 4) **THEMES (Proposed):** Based on the THEMES framework, the probabilities belonging to *RMT-TICC*-learned sub-trajectory clusters as well as the probabilities following *EM-EDM*-learned policies are taken as additional features.

The metrics utilized include: Accuracy (Acc), Recall (Rec), Precision (Prec), F1-score (F1), and AUC. All parameters in THEMES are determined by 5-fold cross-validation: for RMT-TICC, based on Bayesian information criteria (BIC) [Fri01], we determine the cluster number K as 11, the window size ω as 2, and the sparsity and consistency coefficients λ and β as 1e-5 and 4. For EM-EDM, the optimal cluster number is determined heuristically as 3, by iteratively implementing the EM, until empty clusters are generated or the log-likelihood varies smaller than a pre-defined threshold. The iterations for the overall THEMES framework have a threshold of 10, based on our observation that the clustering likelihood for both MT-TICC and EM-EDM converge within 10 iterations. For fair comparisons, optimal parameters in other baselines are also determined by across-validation.

We implemented LSTM with Keras and tuned the parameters by grid search. The results are

Table 6.1 Early prediction using original features with additional features learned by different methods when $\tau = 36$ and $\tau \in [12, 36]$.

(a) **Hold-off Window** $\tau = 36$.

	Acc	Rec	Prec	F1	AUC
Original	.754(.013)	.737(.012)	.763(.014)	.750(.013)	.827(.014)
MT-TICC	.803(.010)	.802(.012)	.802(.011)	.802(.012)	.861(.013)
Action	.757(.013)	.754(.013)	.759(.012)	.756(.013)	.832(.014)
EDM	.765(.013)	.753(.012)	.772(.013)	.762(.011)	.821(.013)
MT-TICC+EM-EDM	.809(.011)*	**.837(.012)***	.793(.012)*	.814(.013)*	.871(.012)*
THEMES	**.834(.011)***	.820(.012)*	**.843(.012)***	**.832(.011)***	**.891(.012)***

(b) **Hold-off Window** $\tau \in [12, 36]$

	Acc	Rec	Prec	F1	AUC
Original	.724(.018)	.714(.021)	.731(.020)	.721(.033)	.812(.026)
MT-TICC	.776(.020)	.790(.024)	.771(.019)	.778(.021)	.839(.020)
Action	.724(.018)	.717(.020)	.723(.019)	.722(.029)	.814(.024)
EDM	.727(.017)	.718(.020)	.736(.019)	.726(.031)	.829(.024)
MT-TICC+EM-EDM	.784(.016)*	.807(.021)*	.783(.018)*	.795(.017)*	.850(.019)*
THEMES	**.801(.015)***	**.816(.021)***	**.789(.019)***	**.802(.016)***	**.860(.017)***

$\star$ denotes p-value < 0.01 comparing THEMES and MT-TICC+EM-EDM vs. Original. The best results are in bold.

compared over $\tau = 36$ and $\tau \in [12, 36]$ to validate whether septic shock can be predicted earlier before the onset. Each method was repeated 10 times by randomly splitting 80% of data for training and 20% for testing and we conducted a corrected paired t-test [Nad03] to compare two of our methods, *i.e.*, MT-TICC+EM-EDM and THEMES versus the Original. Besides, we reported the critical difference diagram [IF19] with Wilcoxon signed ranks tests over F1 and AUC when the hold-off window $\tau = 36$.

6.3.2 Results

The experimental results are reported in Table 6.1, and the critical difference diagram with Wilcoxon signed ranks tests over F1-score and AUC for $\tau = 36$ are shown in Figure 6.1. Regarding the additional features extracted from progressive/interventional patterns learned from different methods: a) *Progressive patterns*: among the two observation-based methods (*i.e.*, Original and MT-TICC), by introducing progressive patterns, MT-TICC performs much better. Meanwhile, among the four methods using observations+interventions (*i.e.*, Action, EDM, MT-TICC+EM-EDM, and THEMES), our two methods with progressive patterns (*i.e.*, MT-TICC+EM-EDM and THEMES) outperforms the two without (*i.e.*, Action and EDM). Thus, by incorporating the progressive patterns, patients' progressive stages can be better captured to improve the early prediction; b) *Interventional patterns*: comparing Action and EDM vs. Original, Action has similar results with Original, while EDM gets

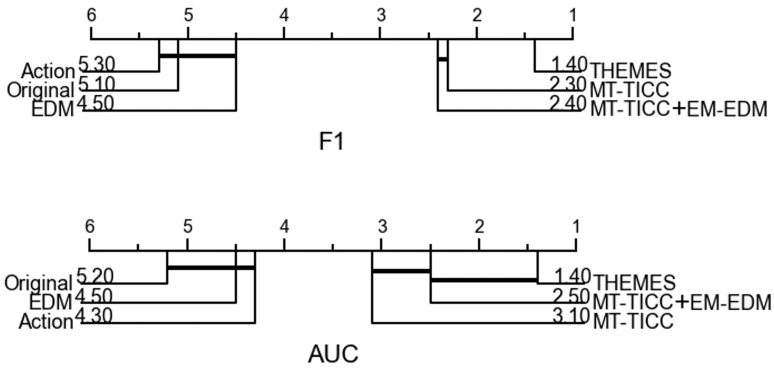


Figure 6.1 Critical Difference Diagram with Wilcoxon Signed-rank Test over F1 and AUC ($\tau = 36$).

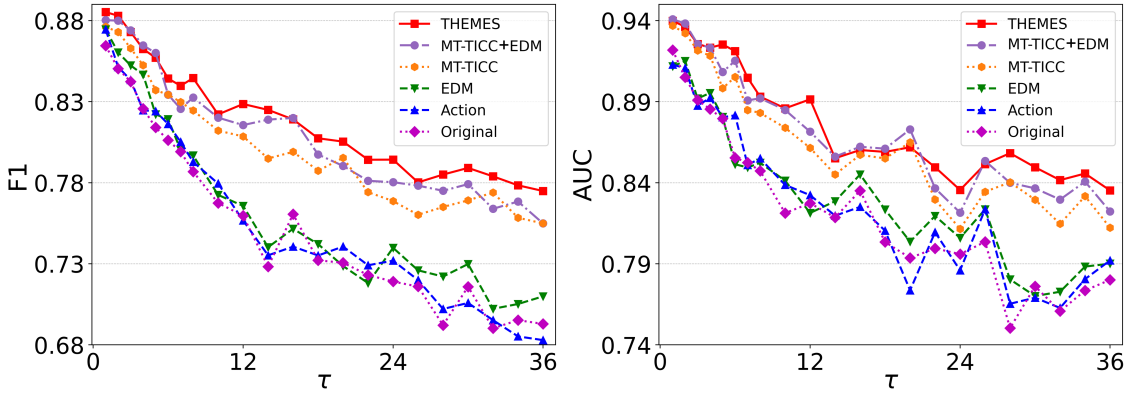


Figure 6.2 Early prediction F1 & AUC given original features with additional features learned by different methods when $\tau \in [1, 36]$.

slightly better performance. Meanwhile, comparing MT-TICC+EM-EDM and THEMES vs. MT-TICC, MT-TICC+EM-EDM is marginally better than MT-TICC, while THEMES improves the performance. Thus, directly taking actions as additional features via Action cannot fully capture the clinicians' decision-making patterns, while incorporating AL-learned patterns can better reflect the effectiveness of interventions; c) *Progressive and interventional patterns*: among two of our proposed methods with both patterns (*i.e.*, MT-TICC+EM-EDM and THEMES), THEMES is better, which indicates the effectiveness of using interventional patterns to refine the learned progressive stages.

Figure 6.2 shows the F1 and AUC when varying $\tau \in [1, 36]$ hours before the septic shock onset. As τ increases, it becomes harder for early prediction across all models. The figures show the advantage of progressive patterns in improving the performance, since MT-TICC, MT-TICC+EM-EDM, and THEMES outperform Original, Action, and EDM. Meanwhile, incorporating interventions is essential, while how they are incorporated also matters, as Action and EDM get similar results with Original, while MT-TICC+EM-EDM and THEMES improve the results.

Figure 6.3 shows the projection of THEMES-derived features for 100 patients whose interventions have the highest matching rate with the learned policies (High Match) and 100 patients

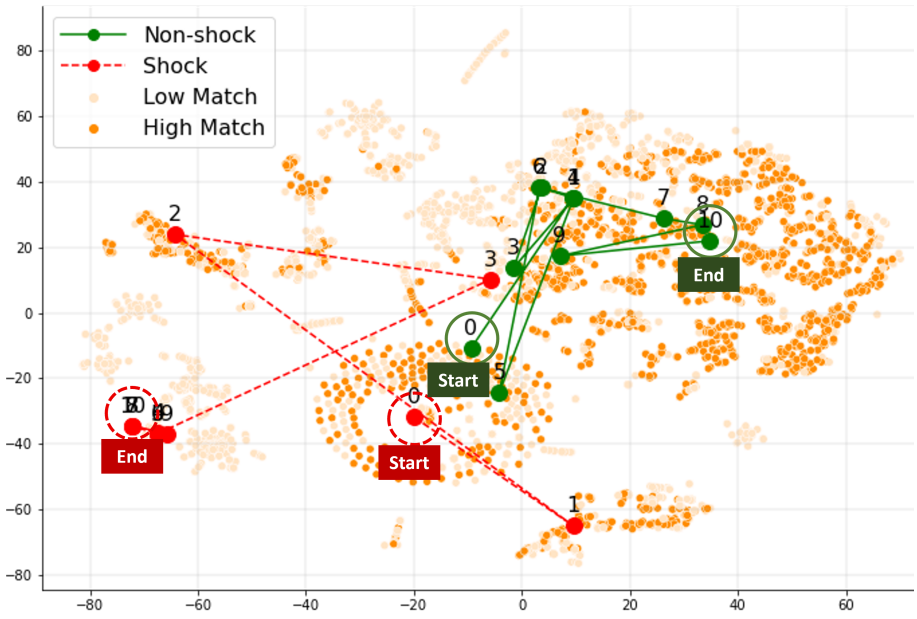


Figure 6.3 Visualizing the progression of patients.

with the lowest (Low Match) on 2D scatter plot using t-SNE [Maa08]. One shock (red) and one non-shock (green) patient with the same length are randomly sampled. Although the start points for the two patients are similar, they drift apart as the non-shock patient's treatments (green) align with the High-matched group while the shock patient's treatments (red) align with the Low-matched group, especially after time point 3. It demonstrates that THEMES can capture the effectiveness of treatments of progressive patterns for distinguishing shock patients from non-shock patients.

6.4 Summary

In this chapter, we aim at incorporating the impact of clinicians' interventions learned by our **THEMES** framework when conducting the DPM. Taking the power of sub-trajectory clustering, which has shown great success in DPM, we propose to utilize THEMES framework to capture the interventional patterns by AL and utilize the extracted patterns to refine the DPM. The experimental results suggest that by incorporating the features extracted from THEMES-derived patterns, more refined DPM can be achieved, as the septic shock can be predicted more accurately in earlier stages, which can shed some light on assisting clinicians to carry out personalized treatments in time.

7 CONCLUSIONS & FUTURE WORK

The major motivation of this thesis lies in developing an AL framework to induce policies in an **offline** manner from the demonstrations driven by *multiple* **evolving-heterogeneous** reward functions. It is especially beneficial for *human-centric* tasks, where the *offline* setting without rolling out the policy is highly desired and humans commonly carry out actions adaptively with *heterogeneous* decision-making strategies varying over time. To handle this issue, we propose a framework named: Time-aware Hierarchical EM Energy-based Sub-trajectory (**THEMES**) clustering. Specifically, to learn the *evolving*-heterogeneous reward functions, we modeled it hierarchically with two components: *(I) sub-trajectories partitioning* based on observed states; and *(II)* Inducing heterogeneous policies varying across sub-trajectories. The two components were conducted iteratively in THEMES to refine the sub-trajectory partitioning, so as to induce more fine-grained and accurate policies.

In *Component I*, *i.e.*, **sub-trajectory partitioning**, based on a novel sub-trajectory clustering approach: Toeplitz Inverse Covariance-based Clustering (TICC), we proposed a Multi-series Time-aware TICC (**MT-TICC**) and deployed it in a *healthcare* application to model the progression for a challenging disease – sepsis. Compared to the original TICC, our MT-TICC made improvements in two aspects: 1) the objective function is more general to handle the *multi-series input*, and 2) the *time-awareness* is introduced in the consistency constraint to incorporate the irregular time intervals. To validate the effectiveness of MT-TICC, we first employed it in a case study using a hand gesture dataset, which has similar properties to the healthcare data while having ground truth labels. Afterward, we further applied it to electronic healthcare records (EHRs) data, taking the MT-TICC-derived progressive patterns as additional features when conducting the septic shock early prediction task. The results showed the patterns learned by MT-TICC can effectively improve the prediction performance. Additionally, the MT-TICC-learned clusters conveyed interpretable insights which could help clinicians gain a better understanding of the sepsis progression.

In *Component II*, *i.e.*, modeling **across-heterogeneous** reward functions, we adapted an expectation maximization IRL **(EM-IRL)** framework and used it for students' subtyping in an *education* application. The EM-IRL can capture different decision-making strategies across different trajectories driven by various reward functions and group the trajectories into different clusters. Herein, we aim to cluster the students based on their learning intentions, *i.e.*, reward functions, when they inter-

act with an intelligent tutoring system. To the best of our knowledge, this is the first IRL-based work for analyzing the heterogeneous reward functions in a *human-centric* task. To evaluate the performance of EM-IRL, we first applied it to three simulation environments. The results demonstrated the EM-IRL has robust performance in accurately clustering the trajectories with different reward functions. Then we further applied it to real-world students' data. The results suggested that the EM-IRL could effectively group students into different subtypes, *e.g.*, learning-oriented, efficient-oriented, and no-learning. The subtyping results showed the potential to give individualized interventions to better assist students' learning.

Equipped with the above two components, we further proposed a unified **THEMES** framework for modeling the **evolving-heterogeneous** reward functions with *healthcare* data. Specifically, based on MT-TICC, we propose a reward-regulated MT-TICC (**RMT-TICC**), which can encode the clinical *interventional* patterns during the modeling by introducing a reward regulator, considering the fact that such clinical interventions are latently driven by rewards. Using our RMT-TICC, more fine-grained sub-trajectories can be obtained to indicate different progressive stages. Besides, based on EM-IRL and an offline AL method of energy-based distribution matching (EDM), we propose an expectation maximization EDM (**EM-EDM**). Utilizing our EM-EDM, diverse decision-making policies over different progressive stages can be effectively induced offline, without the need of rolling out the policies iteratively. Besides, EM-EDM can efficiently model continuous state spaces. When applying the THEMES framework to EHRs data, it demonstrated better performance compared to competitive baselines in both: 1) *policy induction* from demonstrated trajectories with evolving-heterogeneous reward functions; and 2) *early prediction* of septic shock by incorporating THEMES-derived progressive patterns as well as clinical interventional patterns as additional features. The fine-grained policies derived by THEMES can better evaluate the clinicians' interventions, and it can also recommend clinicians with more personalized and timely treatments.

There are multiple promising directions for future work: 1) For MT-TICC, we will explore more general models, in which the context window for extracting the time-invariant structural patterns can have a dynamic size. In addition, the attention for different timestamps can be incorporated when extracting such patterns; 2) For EM-IRL-based student subtyping, instead of utilizing the whole trajectory till the end of their learning to learn the student subtypes, we can use the data collected before a certain time point, so that the subtypes can be reliably detected earlier and the tutor can carry out proactive interventions accordingly; 3) For THEMES, since there are no standard human-centric benchmarks with multiple reward functions evolving over time, in this thesis, we validated the performance based on a complex real-world EHRs dataset. In future work, we will explore other possible simulated/real-world applications to further evaluate the overall framework.

BIBLIOGRAPHY

[Abb04a] Abbeel, P. & Ng, A. Y. "Apprenticeship Learning via Inverse Reinforcement Learning". *ICML*. Banff, Alberta, Canada, 2004.

[Abb04b] Abbeel, P. & Ng, A. Y. "Apprenticeship learning via inverse reinforcement learning". *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 1.

[Ale00] Aleven, V. & Koedinger, K. R. "Limitations of student control: Do students know when they need help?" *International conference on ITS*. Springer. 2000, pp. 292–303.

[Ale02] Aleven, V. A. & Koedinger, K. R. "An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor". *Cognitive science* **26**.2 (2002), pp. 147–179.

[Ame06] Amershi, S. & Conati, C. "Automatic recognition of learner groups in exploratory learning environments". *International Conference on ITS*. Springer. 2006, pp. 463–472.

[Ame09] Amershi, S. & Conati, C. "Combining unsupervised and supervised classification to build user models for exploratory". *JEDM| Journal of Educational Data Mining* **1**.1 (2009), pp. 18–71.

[Amo16] Amodei, D. et al. "Concrete problems in AI safety". *arXiv preprint arXiv:1606.06565* (2016).

[Arg09] Argall, B. D. et al. "A survey of robot learning from demonstration". *Robotics and autonomous systems* **57**.5 (2009), pp. 469–483.

[Aro21] Arora, S. & Doshi, P. "A survey of inverse reinforcement learning: Challenges, methods and progress". *Artificial Intelligence* **297** (2021), p. 103500.

[Aro20] Arora, S. et al. "Maximum Entropy Multi-Task Inverse RL". *arXiv preprint arXiv:2004.12873* (2020).

[Aso13] Asoh, H. et al. "An application of inverse reinforcement learning to medical records of diabetes treatment". *ECMLPKDD2013 workshop on reinforcement learning with generalized feedback*. 2013.

[Aze19] Azevedo, R. et al. "Self-regulation in computer-assisted learning systems." (2019).

[Azi18] Azizsoltani, H. & Sadeghi, E. "Adaptive sequential strategy for risk estimation of engineering systems using Gaussian process regression active learning". *Engineering Applications of Artificial Intelligence* **74** (2018), pp. 146–165.

[Azi19] Azizsoltani, H. et al. "Unobserved is not equal to non-existent: using Gaussian processes to infer immediate rewards across contexts". *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press. 2019, pp. 1974–1980.

[Bab11] Babes, M. et al. “Apprenticeship learning about multiple intentions”. *Proceedings of the 28th International Conference on Machine Learning*. 2011, pp. 897–904.

[Bay17] Baytas, I. M. et al. “Paient subtyping via time-aware LSTM networks”. *SIGKDD*. ACM. 2017, pp. 65–74.

[Ber05] Berkhin, P. “A survey on PageRank computing”. *Internet mathematics* **2**.1 (2005), pp. 73–120.

[Ber94] Berndt, D. J. & Clifford, J. “Using dynamic time warping to find patterns in time series.” *KDD workshop*. Vol. 10. 16. Seattle, WA, USA: 1994, pp. 359–370.

[Bou11] Boularias, A. et al. “Relative entropy inverse reinforcement learning”. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. 2011, pp. 182–189.

[Boy11] Boyd, S. et al. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[Bro16] Brockman, G. et al. “Openai gym”. *arXiv preprint arXiv:1606.01540* (2016).

[Cha21] Chan, A. J. & Schaar, M. van der. “Scalable bayesian inverse reinforcement learning”. *arXiv preprint arXiv:2102.06483* (2021).

[Che15] Che, Z. et al. “Deep computational phenotyping”. *SIGKDD*. ACM. 2015, pp. 507–516.

[Chi11a] Chi, M. et al. “An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach”. *International Journal of Artificial Intelligence in Education* **21**.1-2 (2011), pp. 83–113.

[Chi11b] Chi, M. et al. “Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies”. *User Modeling and User-Adapted Interaction* **21**.1-2 (2011), pp. 137–180.

[Cho16a] Choi, E. et al. “Doctor ai: Predicting clinical events via recurrent neural networks”. *MLHC*. 2016, pp. 301–318.

[Cho16b] Choi, E. et al. “Retain: An interpretable predictive model for healthcare using reverse time attention mechanism”. *NIPS*. 2016, pp. 3504–3512.

[Cho12] Choi, J. & Kim, K.-E. “Nonparametric Bayesian inverse reinforcement learning for multiple reward functions”. *Advances in Neural Information Processing Systems*. 2012, pp. 305–313.

[Cli16] Clifford, K. M. et al. “Challenges with diagnosing and managing sepsis in older adults”. *Expert review of anti-infective therapy* **14**.2 (2016), pp. 231–241.

[Coo16] Cook, S. F. & Bies, R. R. “Disease progression modeling: key concepts and recent developments”. *Current pharmacology reports* **2**.5 (2016), pp. 221–230.

[Cor94] Corbett, A. T. & Anderson, J. R. “Knowledge tracing: Modeling the acquisition of procedural knowledge”. *User modeling and user-adapted interaction* **4**.4 (1994), pp. 253–278.

[Cut11] Cuturi, M. “Fast global alignment kernels”. *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 929–936.

[Das98] Das, G. et al. “Rule Discovery from Time Series.” *KDD*. Vol. 98. 1. 1998, pp. 16–22.

[Dim11] Dimitrakakis, C. & Rothkopf, C. A. “Bayesian multitask inverse reinforcement learning”. *European workshop on reinforcement learning*. Springer. 2011, pp. 273–284.

[Dor15] Doroudi, S. et al. “Towards Understanding How to Leverage Sense-Making, Induction and Refinement, and Fluency to Improve Robust Learning.” *International Educational Data Mining Society* (2015).

[DA20] Dulac-Arnold, G. et al. “An empirical investigation of the challenges of real-world reinforcement learning”. *arXiv preprint arXiv:2003.11881* (2020).

[Dur14] Durairaj, M & Vijitha, C. “Educational data mining for prediction of student performance using clustering algorithms”. *International Journal of Computer Science and Information Technologies* **5**.4 (2014), pp. 5987–5991.

[Est16] Esteban, C., Staeck, O., et al. “Predicting clinical events by combining static and dynamic information using recurrent neural networks”. *ICHI*. IEEE. 2016, pp. 93–101.

[Fan20] Fan, J. et al. “A theoretical analysis of deep Q-learning”. *Learning for Dynamics and Control*. PMLR. 2020, pp. 486–489.

[Far21] Faruqui, S. H. A. et al. “A functional model for structure learning and parameter estimation in continuous time Bayesian network: An application in identifying patterns of multiple chronic conditions”. *IEEE Access* **9** (2021), pp. 148076–148089.

[Fin16] Finn, C. et al. “Guided cost learning: Deep inverse optimal control via policy optimization”. *International conference on machine learning*. PMLR. 2016, pp. 49–58.

[Fle20] Fleuren, L. M. et al. “Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy”. *Intensive care medicine* (2020), pp. 1–18.

[Fri01] Friedman, J. et al. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[Fri08] Friedman, J. et al. “Sparse inverse covariance estimation with the graphical lasso”. *Biostatistics* **9**.3 (2008), pp. 432–441.

[Gao22] Gao, G. et al. “A Reinforcement Learning-Informed Pattern Mining Framework for Multivariate Time Series Classification” (2022).

[Gau13] Gauer, R. “Early recognition and management of sepsis in adults: the first six hours”. *American family physician* **88**.1 (2013), pp. 44–53.

[Gha14] Ghassempour, S. et al. “Clustering multivariate time series using hidden Markov models”. *International journal of environmental research and public health* **11**.3 (2014), pp. 2741–2763.

[Gia18] Giannoula, A. et al. “Identifying temporal patterns in patient disease trajectories using dynamic time warping: a population-based study”. *Scientific reports* **8**.1 (2018), pp. 1–14.

[Giu07] Giuliano, K. K. “Physiological monitoring for critically ill patients: testing a predictive model for the early detection of sepsis”. *American Journal of Critical Care* **16**.2 (2007), pp. 122–130.

[Goh21] Goh, K. H. et al. “Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare”. *Nature communications* **12**.1 (2021), pp. 1–10.

[GE07] González-Espada, W. J. & Bullock, D. W. “Innovative applications of classroom response systems: Investigating students’ item response times in relation to final course grade, gender, general point average, and high school ACT scores”. *Electronic Journal for the Integration of Technology in Education* **6** (2007), pp. 97–108.

[Goy19] Goyal, P. et al. “Using natural language for reward shaping in reinforcement learning”. *arXiv preprint arXiv:1903.02020* (2019).

[Gra04] Graesser, A. C. et al. “AutoTutor: A tutor with dialogue in natural language”. *Behavior Research Methods, Instruments, & Computers* **36**.2 (2004), pp. 180–192.

[Gra19] Grathwohl, W. et al. “Your classifier is secretly an energy based model and you should treat it like one”. *arXiv preprint arXiv:1912.03263* (2019).

[Gre07] Greene, J. A. & Azevedo, R. “A theoretical review of Winne and Hadwin’s model of self-regulated learning: New perspectives and directions”. *Review of educational research* **77**.3 (2007), pp. 334–372.

[Hal17] Hallac, D. et al. “Toeplitz inverse covariance-based clustering of multivariate time series data”. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, pp. 215–223.

[Hau17] Hausman, K. et al. “Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets”. *Advances in neural information processing systems* **30** (2017).

[Ho16] Ho, J. & Ermon, S. “Generative adversarial imitation learning”. *Advances in neural information processing systems* **29** (2016).

[Hoc97] Hochreiter, S. & Schmidhuber, J. “Long short-term memory”. *Neural computation* **9**.8 (1997), pp. 1735–1780.

[Igl09a] Iglesias, A. et al. “Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning”. *Applied Intelligence* **31**.1 (2009), pp. 89–106.

[Igl09b] Iglesias, A. et al. “Reinforcement learning of pedagogical policies in adaptive and intelligent educational systems”. *Knowledge-Based Systems* **22**.4 (2009), pp. 266–270.

[IF19] Ismail Fawaz, H. et al. “Deep learning for time series classification: a review”. *Data Mining and Knowledge Discovery* **33**.4 (2019), pp. 917–963.

[Jar20] Jarrett, D. et al. “Strictly batch imitation learning by energy-based distribution matching”. *Advances in Neural Information Processing Systems* **33** (2020), pp. 7354–7365.

[Joh16] Johnson, A. E. et al. “MIMIC-III, a freely accessible critical care database”. *Scientific data* **3**.1 (2016), pp. 1–9.

[Ju22] Ju, S. et al. “Student-Tutor Mixed-Initiative Decision-Making Supported by Deep Reinforcement Learning”. *International Conference on Artificial Intelligence in Education*. Springer. 2022, pp. 440–452.

[Kae96] Kaelbling, L. P. et al. “Reinforcement learning: A survey”. *Journal of artificial intelligence research* **4** (1996), pp. 237–285.

[Kha17] Khalil, M. & Ebner, M. “Clustering patterns of engagement in Massive Open Online Courses (MOOCs): the use of learning analytics to reveal student categories”. *Journal of Computing in Higher Education* **29**.1 (2017), pp. 114–132.

[Kim18] Kim, Y.-J. & Chi, M. “Temporal Belief Memory: Imputing Missing Data during RNN Training.” *In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-2018)*. 2018.

[Kle12] Klein, E. et al. “Inverse reinforcement learning through structured classification”. *Advances in NIPS*. 2012, pp. 1007–1015.

[Köc11] Köck, M. & Paramythis, A. “Activity sequence modelling and dynamic clustering for personalized e-learning”. *User Modeling and User-Adapted Interaction* **21**.1-2 (2011), pp. 51–97.

[Koe13] Koedinger, K. R. et al. “New potentials for data-driven intelligent tutoring system development and optimization”. *AI Magazine* **34**.3 (2013), pp. 27–41.

[Kom18] Komorowski, M. et al. “The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care”. *Nature medicine* **24**.11 (2018), pp. 1716–1720.

[Kos18] Kostrikov, I. et al. “Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning”. *arXiv preprint arXiv:1809.02925* (2018).

[Kos19] Kostrikov, I. et al. “Imitation learning via off-policy distribution matching”. *arXiv preprint arXiv:1912.05032* (2019).

[Kri16] Krishnan, S. et al. “Hirl: Hierarchical inverse reinforcement learning for long-horizon tasks with delayed rewards”. *arXiv preprint arXiv:1604.06508* (2016).

[Kum06] Kumar, A., Roberts, D., et al. “Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock”. *Critical care medicine* (2006).

[Kwo20] Kwon, B. C. et al. “Modeling disease progression trajectories from longitudinal observational data”. *AMIA Annual Symposium Proceedings*. Vol. 2020. American Medical Informatics Association. 2020, p. 668.

[Lev11] Levine, S. et al. “Nonlinear inverse reinforcement learning with gaussian processes”. *Advances in Neural Information Processing Systems*. 2011, pp. 19–27.

[Lev20] Levine, S. et al. “Offline reinforcement learning: Tutorial, review, and perspectives on open problems”. *arXiv preprint arXiv:2005.01643* (2020).

[Li06] Li, C. & Yoo, J. “Modeling student online learning using clustering”. *Proceedings of the 44th annual Southeast regional conference*. 2006, pp. 186–191.

[Li18] Li, J. et al. “Tatc: Predicting alzheimer’s disease with actigraphy data”. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 509–518.

[Li12] Li, Y. et al. “Visualizing variable-length time series motifs”. *Proceedings of the 2012 SIAM*. SIAM. 2012, pp. 895–906.

[Li17] Li, Y. et al. “Infogail: Interpretable imitation learning from visual demonstrations”. *Advances in Neural Information Processing Systems* **30** (2017).

[Lin18] Lin, C. et al. “Early Diagnosis and Prediction of Sepsis Shock by Combining Static and Dynamic Information Using Convolutional-LSTM”. *ICHI*. IEEE. 2018, pp. 219–228.

[Lin19] Lin, C. et al. “Multi-layer facial representation learning for early prediction of septic shock”. *2019 IEEE International Conference on Big Data (Big Data)*. IEEE. 2019, pp. 840–849.

[Lip16] Lipton, Z. et al. “Directly Modeling Missing Data in Sequences with RNNs: Improved Classification of Clinical Time Series”. *JMLR* **56** (2016).

[Lip15] Lipton, Z. C., Kale, D. C., et al. “Learning to diagnose with LSTM recurrent neural networks”. *arXiv preprint arXiv:1511.03677* (2015).

[Lob18] Lobov, S. et al. “Latent factors limiting the performance of sEMG-interfaces”. *Sensors* **18**.4 (2018), p. 1122.

[Lob19] Lobov, S. et al. *UCI Machine Learning Repository. EMG data for gestures Data Set*. `https://archive.ics.uci.edu/ml/datasets/EMG+data+for+gestures#`. 2019.

[Lon16] Long, Y. & Aleven, V. “Mastery-oriented shared student/system control over problem selection in a linear equation tutor”. *International conference on intelligent tutoring systems*. Springer. 2016, pp. 90–100.

[Lop12] Lopez, M. I. et al. “Classification via clustering for predicting final marks based on student participation in forums.” *International Educational Data Mining Society* (2012).

[Maa08] Maaten, L. Van der & Hinton, G. “Visualizing data using t-SNE.” *Journal of machine learning research* **9**.11 (2008).

[Man14] Mandel, T. et al. “Offline policy evaluation across representations with applications to educational games”. *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. 2014, pp. 1077–1084.

[Mar03] Martin, G. S. et al. “The epidemiology of sepsis in the United States from 1979 through 2000”. *New England Journal of Medicine* **348**.16 (2003), pp. 1546–1554.

[Mei06] Meinshausen, N. & Bühlmann, P. “High-dimensional graphs and variable selection with the Lasso”. *Ann. Statist.* **34**.3 (2006), pp. 1436–1462.

[Mit03] Mitrovic, A. & Martin, B. “Scaffolding and fading problem selection in SQL-Tutor”. *Proceedings of the 11th International Conference on Artificial Intelligence in Education*. 2003, pp. 479–481.

[Nad03] Nadeau, C. & Bengio, Y. “Inference for the generalization error”. *Machine learning* **52**.3 (2003), pp. 239–281.

[Ng00] Ng, A. Y. & Russell, S. J. “Algorithms for inverse reinforcement learning.” *ICML*. Vol. 1. 2000, p. 2.

[Oh19] Oh, M.-h. & Iyengar, G. “Sequential Anomaly Detection using Inverse Reinforcement Learning”. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 1480–1490.

[Pal18] Palanisamy, P. *Hands-On Intelligent Agents with OpenAI Gym: Your guide to developing AI agents using deep reinforcement learning*. Packt Publishing Ltd, 2018.

[Pan19] Pan, M. et al. “Dissecting the learning curve of taxi drivers: A data-driven approach”. *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM. 2019, pp. 783–791.

[Pav04] Pavlovic, V. “Model-based motion clustering using boosted mixture modeling”. *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* Vol. 1. IEEE. 2004, pp. I–I.

[Pha16] Pham, T. et al. “Deepcare: A deep dynamic memory model for predictive medicine”. *PAKDD*. Springer. 2016, pp. 30–41.

[Pom91] Pomerleau, D. A. “Efficient training of artificial neural networks for autonomous navigation”. *Neural computation* **3**.1 (1991), pp. 88–97.

[Raf16a] Rafferty, A. N, Brunskill, E., et al. “Faster teaching via pomdp planning”. *Cognitive science* **40**.6 (2016), pp. 1290–1332.

[Raf15] Rafferty, A. N. et al. “Inferring learners’ knowledge from their actions”. *Cognitive Science* **39**.3 (2015), pp. 584–618.

[Raf16b] Rafferty, A. N. et al. “Using Inverse Planning for Personalized Feedback.” *EDM* **16** (2016), pp. 472–477.

[Rag17] Raghu, A. et al. “Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach”. *Machine Learning for Healthcare Conference*. PMLR. 2017, pp. 147–163.

[Rak13] Rakthanmanon, T. et al. “Addressing big data time series: Mining trillions of time series subsequences under dynamic time warping”. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **7**.3 (2013), pp. 1–31.

[Ram07] Ramachandran, D. & Amir, E. “Bayesian Inverse Reinforcement Learning.” *IJCAI*. Vol. 7. 2007, pp. 2586–2591.

[Rat06] Ratliff, N. D. et al. “Maximum margin planning”. *Proceedings of the 23rd international conference on Machine learning*. ACM. 2006, pp. 729–736.

[Raz12] Raza, S. et al. “Teaching coordinated strategies to soccer robots via imitation”. *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2012, pp. 1434–1439.

[Rey09] Reynolds, D. A. “Gaussian Mixture Models.” *Encyclopedia of biometrics* **741** (2009).

[Rol14] Roll, I. et al. “On the benefits of seeking (and avoiding) help in online problem-solving environments”. *Journal of the Learning Sciences* **23**.4 (2014), pp. 537–560.

[Ros11] Ross, S. et al. “A reduction of imitation learning and structured prediction to no-regret online learning”. *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 627–635.

[Row15] Rowe, J. P. & Lester, J. C. “Improving student problem solving in narrative-centered learning environments: A modular reinforcement learning framework”. *AIED*. Springer. 2015, pp. 419–428.

[Saq18] Saqib, M. et al. “Early prediction of sepsis in EMR records using traditional ML techniques and deep learning LSTM networks”. *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2018, pp. 4038–4041.

[Sch99] Schaal, S. “Is imitation learning the route to humanoid robots?” *Trends in cognitive sciences* **3**.6 (1999), pp. 233–242.

[Sev20] Severson, K. A. et al. “Personalized input-output hidden Markov models for disease progression modeling”. *Machine Learning for Healthcare Conference*. PMLR. 2020, pp. 309–330.

[Sha18] Sharma, A. et al. “Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information”. *arXiv preprint arXiv:1810.01266* (2018).

[She16a] Shen, S. & Chi, M. “Aim Low: Correlation-Based Feature Selection for Model-Based Reinforcement Learning.” *International Educational Data Mining Society* (2016).

[She16b] Shen, S. & Chi, M. “Reinforcement Learning: the Sooner the Better, or the Later the Better?” *Proceedings of the 2016 Conference on UMAP*. ACM. 2016, pp. 37–44.

[She17] Shen, S. & Chi, M. “Clustering Student Sequential Trajectories Using Dynamic Time Warping.” *International Educational Data Mining Society* (2017).

[Sin16] Singer, M. et al. “The third international consensus definitions for sepsis and septic shock (Sepsis-3)”. *Jama* **315**.8 (2016), pp. 801–810.

[Smy97] Smyth, P. “Clustering sequences with hidden Markov models”. *Advances in neural information processing systems*. 1997, pp. 648–654.

[Soh20a] Sohn, H. et al. “MuLan: Multilevel Language-based Representation Learning for Disease Progression Modeling”. *IEEE International Conference on Big Data*. 2020.

[Soh20b] Sohn, H. et al. “Mulan: Multilevel language-based representation learning for disease progression modeling”. *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 1246–1255.

[Suk12] Sukkar, R. et al. “Disease progression modeling using hidden Markov models”. *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2012, pp. 2845–2848.

[Sun19] Sun, Z. et al. “A probabilistic disease progression modeling approach and its application to integrated Huntington’s disease observational data”. *JAMIA open* (2019).

[Sye07] Syed, U. & Schapire, R. E. “A game-theoretic approach to apprenticeship learning”. *Advances in neural information processing systems* **20** (2007).

[Sye12] Syed, U. & Schapire, R. E. “Imitation learning with a value-based prior”. *arXiv preprint arXiv:1206.5290* (2012).

[Tau17] Taub, M. et al. “Strategies for designing advanced learning technologies to foster self-regulated learning”. *Strategies for deep learning with digital technology: Theories and practices in education* (2017), pp. 137–170.

[Tin85] Tintinalli, J. E. et al. *Emergency medicine: a comprehensive study guide*. Mcgraw-hill New York, 1985.

[Vit67] Viterbi, A. “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm”. *IEEE transactions on Information Theory* **13**.2 (1967), pp. 260–269.

[Voi20] Voisin, P. S. “An Application of Graph Diffusion for Gesture Classification”. PhD thesis. Duke University, 2020.

[Wan20a] Wang, J. et al. “Inferring Continuous Treatment Doses from Historical Data via Model-Based Entropy-Regularized Reinforcement Learning”. *Asian Conference on Machine Learning*. PMLR. 2020, pp. 433–448.

[Wan18] Wang, L. et al. “Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation”. *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 2447–2456.

[Wan20b] Wang, L. et al. “Adversarial Cooperative Imitation Learning for Dynamic Treatment Regimes”. *Proceedings of The Web Conference 2020*. 2020, pp. 1785–1795.

[Wan21] Wang, L. et al. “Hierarchical Imitation Learning with Contextual Bandits for Dynamic Treatment Regimes”. *Reinforcement Learning for Real Life (RL4RealLife) Workshop in the 38 th International Conference on Machine Learning*. 2021.

[Wan22] Wang, L. et al. “Hierarchical Imitation Learning via Subgoal Representation Learning for Dynamic Treatment Recommendation”. *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 2022, pp. 1081–1089.

[Wei00] Weinstein, C. E. et al. “Self-regulation interventions with a focus on learning strategies”. *Handbook of self-regulation*. Elsevier, 2000, pp. 727–747.

[Wel11] Welling, M. & Teh, Y. W. “Bayesian learning via stochastic gradient Langevin dynamics”. *Proceedings of the 28th international conference on machine learning (ICML-11)*. Citeseer. 2011, pp. 681–688.

[Win98] Winne, P. H. & Hadwin, A. F. “Studying as self-regulated learning.” *Metacognition in educational theory and practice, The educational psychology series.* 1998.

[Win12] Winne, P. H. & Hadwin, A. F. “The weave of motivation and self-regulated learning”. *Motivation and self-regulated learning*. 2012, pp. 309–326.

[Wul15] Wulfmeier, M. et al. “Maximum entropy deep inverse reinforcement learning”. *arXiv preprint arXiv:1507.04888* (2015).

[Yan17] Yang, J. et al. “sEMG-based continuous hand gesture recognition using GMM-HMM and threshold model”. *IEEE ROBIO*. 2017.

[Yan18] Yang, X. et al. “Time-aware subgroup matrix decomposition: Imputing missing data using forecasting events”. *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 1524–1533.

[Yan20] Yang, X. et al. “Student Subtyping via EM-Inverse Reinforcement Learning.” *International Educational Data Mining Society* (2020).

[Yan21] Yang, X. et al. “Multi-series time-aware sequence partitioning for disease progression modeling”. *IJCAI*. 2021.

[Yu21] Yu, C. et al. “Reinforcement learning in healthcare: A survey”. *ACM Computing Surveys (CSUR)* **55**.1 (2021), pp. 1–36.

[Zha17] Zhang, Y. et al. “LSTM for septic shock: Adding unreliable labels to reliable predictions”. *Big Data*. IEEE. 2017, pp. 1233–1242.

[Zha19] Zhang, Y. et al. “ATTAIN: Attention-based Time-Aware LSTM Networks for Disease Progression Modeling.” *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2019, pp. 4369–4375.

[Zhi12] Zhifei, S. & Meng Joo, E. “A survey of inverse reinforcement learning techniques”. *International Journal of Intelligent Computing and Cybernetics* **5**.3 (2012), pp. 293–311.

[Zho17a] Zhou, G. & Chi, M. “The Impact of Decision Agency & Granularity on Aptitude Treatment Interaction in Tutoring.” *CogSci*. 2017, pp. 3652–3657.

[Zho16] Zhou, G. et al. “The Impact of Granularity on the Effectiveness of Students’ Pedagogical Decisions”. *CogSci*. 2016, pp. 2801–2806.

[Zho17b] Zhou, G. et al. “Towards Closing the Loop: Bridging Machine-induced Pedagogical Policies to Learning Theories”. *EDM*. 2017.

[Zho19] Zhou, G. et al. “Hierarchical reinforcement learning for pedagogical policy induction”. *International conference on artificial intelligence in education*. Springer. 2019, pp. 544–556.

[Zho20] Zhou, G. et al. “Improving Student-Tutor Interaction Through Data-driven Explanation of Hierarchical Reinforcement Induced Pedagogical Policies”. *Proceedings of the 28th Conference on User Modeling, Adaptation and Personalization*. ACM. 2020.

[Zie08] Ziebart, B. D. et al. “Maximum entropy inverse reinforcement learning” (2008).