PERSONALIZING TREATMENTS VIA CONTEXTUAL MULTI-ARMED BANDITS BY IDENTIFYING RELEVANCE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Cem Bulucu
August 2019

Personalizing Treatments via Contextual Multi-Armed Bandits by Identifying Relevance
By Cem Bulucu
August 2019

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Cem Tekin(Advisor)

Orhan Arıkan

Umut Orguner

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

PERSONALIZING TREATMENTS VIA CONTEXTUAL MULTI-ARMED BANDITS BY IDENTIFYING RELEVANCE

Cem Bulucu
M.S. in Electrical and Electronics Engineering
Advisor: Cem Tekin
August 2019

Personalized medicine offers specialized treatment options for individuals which is vital as every patient is different. One-size-fits-all approaches are often not effective and most patients require personalized care when dealing with various diseases like cancer, heart diseases or diabetes. As vast amounts of data became available in medicine (and other fields including web-based recommender systems and intelligent radio networks), online learning approaches are gaining popularity due to their ability to learn fast in uncertain environments. Contextual multi-armed bandit algorithms provide reliable sequential decision-making options in such applications. In medical settings (also in other aforementioned settings), data (contexts) and actions (arms) are often high-dimensional and performances of traditional contextual multi-armed bandit approaches are almost as bad as random selection, due to the curse of dimensionality. Fortunately, in many cases the information relevant to the decision-making task does not depend on all dimensions but rather depends on a small subset of dimensions, called the relevant dimensions. In this thesis, we aim to provide personalized treatments for patients sequentially arriving over time by using contextual multi-armed bandit approaches when the expected rewards related to patient outcomes only vary on a small subset of context and arm dimensions. For this purpose, first we make use of the contextual multi-armed bandit with relevance learning (CMAB-RL) algorithm which learns the relevance by employing a novel partitioning strategy on the context-arm space and forming a set of candidate relevant dimension tuples. In this model, the set of relevant patient traits are allowed to be different for different bolus insulin dosages. Next, we consider an environment where the expected reward function defined over the context-arm space is sampled from a Gaussian process. For this setting, we propose an extension to the contextual Gaussian

process upper confidence bound (CGP-UCB) algorithm, called CGP-UCB with relevance learning (CGP-UCB-RL), that learns the relevance by integrating kernels that allow weights to be associated with each dimension and optimizing the negative log marginal likelihood. Then, we investigate the suitability of this approach in the blood glucose regulation problem. Aside from applying both algorithms to the bolus insulin administration problem, we also evaluate their performance in synthetically generated environments as benchmarks.

Keywords: Online Learning, contextual multi-armed bandits, contextual Gaussian process bandits, relevance learning, personalized medicine.

ÖZET

İLGİ BELİRLEYEREK BAĞLAMSAL ÇOK KOLLU HAYDUTLAR İLE TEDAVİLERİ KİŞİSELLEŞTİRME

Cem Bulucu
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Cem Tekin
Ağustos 2019

Kişiselleştirilmiş tıp bireyler için özel tedavi seçenekleri sunar, bu da her hasta farklı olduğu için hayati bir önem taşır. Herkese uyması beklenen ortak yaklaşımlar genellikle etkili değildir ve çoğu hasta kanser, kalp hastalıkları ve diyabet gibi çeşitli hastalıklarda kişiselleştirilmiş bakıma ihtiyaç duyar. Tıpta (aynı zamanda ağ-tabanlı tavsiye sistemleri ve akıllı radyo ağları gibi diğer alanlarda) çok miktarda verinin elde edilebilmesi ile çevrimiçi öğrenme yöntemleri belirsiz ortamlarda hızlı öğrenme yetenekleri nedeniyle popülerlik kazanmaktadır. Bu tür uygulamalarda bağlamsal çok kollu haydut algoritmaları güvenilir karar verme seçenekleri sunar. Medikal uygulamalarda(ayrıca yukarıda belirtilen uygulamalarda), veriler (bağlamlar) ve eylemler (kollar) genellikle yüksek boyutludur ve çok boyutluluğun lanetinden dolayı geleneksel bağlamsal çok kollu haydut yöntemlerinin performansları neredeyse rasgele seçim kadar kötü olur. Neyse ki, çoğu zaman karar verme görevi ile ilgili bilgiler tüm boyutlara bağlı değildir, bunun yerine boyutların az sayıda eleman içeren ve ilgili boyutlar adı verilen bir alt kümesine bağlıdır. Bu tezde, hastaların sonuçlarına ilişkin beklenen ödüller bağlam ve kol boyutlarının yalnızca küçük bir alt kümesi üzerinde değişiklik gösterdiğinde bağlamsal çok kollu haydut yaklaşımları kullanarak zaman içinde ardışık olarak gelen hastalar için kişiselleştirilmiş tedaviler sağlamak hedeflenmiştir. Bu amaç için, ilk olarak bağlam-kol uzayı üzerinde yeni bir bölümlendirme stratejisi kullanarak ve bir aday ilgili boyut değişkenler grubu oluşturarak ilgi öğrenen ilgi öğrenmeli bağlamsal çok kollu haydut(contextual multi-armed bandit with relevance learning veya kısaca CMAB-RL) algoritması kullanılmıştır. Bu modelde, ilgili hasta özellikleri kümesinin farklı bolus insulin dozları için farklı olmasına izin verilen bir ortamı ele alınmıştır. Daha sonra, bağlam-kol uzayı üzerinde tanımlanan beklenen ödül fonksiyonunun bir

Gauss sürecinden örneklendiği bir ortam ele alınmıştır. Bu ortam için, bağlamsal Gauss süreci üst güven sınırı(contextual Gaussian process upper confidence bound veya kısaca CGP-UCB) algorithmasının bir uzantısı olan ve ilgi öğrenmeyi her boyut için bir ağırlık atama sağlayan çekirdek fonksiyonlarını entegre ederek ve negatif logaritmik marjinal olabilirliği eniyileyerek öğrenen ilgi öğrenmeli CGP-UCB(CGP-UCB with relevance learning veya kısaca CGP-UCB-RL) algoritması önerilmiştir. Sonrasında, bu yaklaşımın kan şekeri düzenlemesi problemine uygunluğu incelenmiştir. Bolus insulin düzenlenmesi problemine uygulanmalarının yanında, iki algoritmanın performansları referans olması için sentetik olarak yaratılmış ortamlarda değerlendirilmiştir.

Anahtar sözcükler: Çevrimiçi öğrenme, bağlamsal çok kollu haydutlar, bağlamsal Gauss süreci haydutlar, ilgi öğrenme, kişiselleşmiş tıp.

Acknowledgement

First, I would like to thank my advisor Dr. Cem Tekin, for being understanding and supportive. This thesis could not be completed without his patience and guidance.

Besides my advisor, I would also like to thank Prof. Orhan Arıkan and Prof. Umut Orguner for being in my thesis committee, for their time and their prized feedbacks.

I feel indebted to Selin Coşan for always being supportive and encouraging, as she helped me get back on my feet even when I felt desperate. I am also grateful to Eralp Turğay for our discussions about maths, physics and philosophy, Kubilay Ekşioğlu for keeping my (and everyone else's) spirits high with witty remarks and funny comments, Ümitcan Şahin for his great teamwork in projects. Additionally, I would like to thank Safa Onur Şahin, Anjum Qureshi, Alparslan Çelik and Andi Nika for relaxing coffee breaks, valued conversations and football matches.

Finally, I would like to thank my family for their love and support throughout my studies.

Part of the work that was included in this thesis was supported by TUBITAK Grant 215E342.

We thank Ohio University for providing us the Ohio T1DM dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Personalized medicine is a very important research area as the physiological properties of each individual is different and drugs or treatments often have varying effects on different people. In general, the personal medical histories, genetic vulnerabilities and ongoing treatments are different for distinct people. In fact, many people do not respond well to their first offered drug in treatments and it is argued that the reason behind this is due to their genetic heritage [1, 2]. This inspires researchers to develop adaptive and personalized strategies to optimize effectiveness. Although the concept is not new, it has gained popularity in recent years and welcomed in general for which the precision medicine initiative can be given [3] as an example, which was generally supported. Personalized medicine allowed improvements in the treatments of diseases cancer and HIV as targeted treatments are utilized that specifically addresses the root cause, as well as decreasing the occurrence of side effects. Moreover, the use of personalized medicine provides a better understanding for a wide range of diseases including cancer, asthma and sensory neuropathy by analyzing the symptoms, how the disease progresses and how different treatments work on different demographics [4]. In addition to treatment options, personalized healthcare also provides disease prevention and early diagnosis, as due to their medical histories and genetic heritage people may be prone to certain diseases. Identification of the proneness of patients helps in managing treatments accordingly or tackling problems they

may have before the problems surface.

In diabetes patients, determination of relevant patient traits is vital in order to provide individual based solutions. Regulation of blood glucose needs to be handled with utmost care since low blood glucose(hypoglycemia) can cause seizures, loss of consciousness and even death whereas high blood glucose(hyperglycemia) can cause skin infections, nerve damage and damage to eyes, blood vessels and kidneys. Different individuals should be subject to different insulin treatments to prevent these complications, as one-size-fits-all dosages may be too high or too low for distinct people, resulting in hypoglycemia or hyperglycemia. In this thesis, we aim to tackle this problem through the use of contextual multi-armed bandit approaches where we aim to use patient information as contexts and bolus insulin dosage as arms.

Like medical diagnosis and treatment recommendation, many real-world tasks also require taking actions in uncertain environments, requiring the learner to base its decisions on observations on the current information about the environment and its past experiences about similar decisions. These tasks are often associated with some notion of a reward, which is revealed after taking the action. The learner is then expected to update its decisions according to the reward and achieve higher rewards in upcoming decisions. In this thesis, our blood glucose regulation models consider the current information about the patient(such as current blood glucose levels, current basal insulin treatment, carbohydrate intake, exercise), recommend a bolus insulin dosage. The rewards are generated according to the effect of the decision on the blood glucose levels of the patient.

Reinforcement learning is an area of machine learning which models this decision-making process of making observations about the state of the environment, taking action, receiving reward and updating accordingly. The state of the environment changes after the action is taken and the reward is associated with this transition. In reinforcement learning models, the learner interacts with the environment repeatedly in discrete time steps, called rounds. The objective of the learner is to maximize its cumulative reward over all rounds. As this setting is very well suited for the blood glucose regulation task we aim to utilize

multi-armed bandit approaches, which will be introduced next.

Modeling the sequential decision-making problem with Multi-Armed Bandits(MABs) is common, as it provides a simple and structured way to model the problem. The importance of such a modeling for online learning tasks in uncertain environments is due to the fact that, in general, no data about the environment is available in advance. Thus, in such tasks the decision-maker is required to quickly learn about the environment and at the same time acquire high rewards. MABs represent a classical reinforcement learning problem which demonstrates the exploration-exploitation trade-off. The original MAB problem involves sequentially determining which slot machine to play in each round, in order to maximize the cumulative monetary gain over a set of rounds. The slot machine is also synonymously called "one-armed bandit" which gives rise to the term "arm"(in some works referred to as "action") in literature. Intuitively, this problem translates into a variety of applications, which we will provide examples of after explaining the generic framework for MAB. MABs can be used to model problems where a reward notion exists and the objective is to sequentially make the decisions that maximize the cumulative reward over all rounds. In MAB problem setting, the learner does not know the underlying reward distributions of arms and as feedback, the learner is revealed a noisy reward sampled from the reward distribution associated with the selected arm in each round. As the reward distributions are unknown, the learner has to utilize the information provided by its previous selections to optimize its decisions. Any learner then faces the aforementioned exploration-exploitation trade-off, since if the learner possesses a greedy approach that only selects seemingly best arm, then it misses the opportunity to give other arms a "fair" chance. For example, one arm may have a high sample mean reward when it is played only a few times, however, it may have a low true mean reward in which case the algorithm would be exploiting a sub-optimal arm. Conversely, an optimal arm may receive low rewards in a few rounds due to randomness, but since it would not be explored further, one may not be able to discover this seeming sub-optimal but truly optimal arm. On the other hand, exploring too much would simply mean unnecessarily selecting sub-optimal arms believing that they may be optimal and end up receiving

nearly equal amounts of high and low rewards. On the average such a learner performs poorly as the learner fails to take advantage of arms with high rewards, in limited amount of time. MAB model addresses this trade-off directly, and hence attracted much attention. One extension to the classical MAB model is the contextual MAB(CMAB) model, which allows the model to make use of side information about the environment. In each round, the learner observes a context from the context set and selects an arm from the arm set. The learner is required to optimize arm selection according to previous context observations, arm selections and rewards as well as the recently received context, because the expected reward of the arms depend on the current context. The applications of MAB and CMAB models include but not limited to hyper-parameter optimization in neural networks [5], intelligent radio networks [6], personalized content [7] and medicine [8, 9].

The performance of MAB and CMAB models are often measured by a metric called regret, which is essentially the difference between the performance of the learner and a strategy (often called an "oracle") that makes the optimal arm selection in every round. Since the oracle is the optimal strategy, minimizing regret is equivalent to maximizing cumulative reward.

Working with high dimensional data commonly causes problems for all machine learning tasks which is an infamous phenomenon, called the curse of dimensionality. In supervised learning, it leads to over-fitting as the model fails to identify actually meaningful features out of a vast set of variables. In unsupervised learning, similarity metrics fail to group similar instances together, since all samples appear to be dissimilar as the volume of the space increases rapidly with dimensionality.

High dimensionality have similar effects on CMAB problems. When the context and arm sets are finite, the effect of the high dimensionality often do not pose problems since commonly the cardinality of the context and arm sets are negligible compared to the number of rounds and the learner can observe enough samples from each context and arm. In a more general setting however, contexts and arms can be infinitely many and hence prevent the learner from observing

each context or selecting each arm even once. In such a setting the context and arm sets are modeled as multi-dimensional sets. Regret depends exponentially on the dimensionalities of these sets [10] and with increasing number of dimensions, the regret converges to a linear function of time. As linear regret is the worst that can be achieved in CMAB setting (for example, a strategy that makes random arm selections achieves linear regret), reducing the effect of dimensionality is of vital importance. Fortunately, in most cases, many of the context and arm dimensions have negligible to no effect on the reward distributions, which are referred to as "irrelevant" hereafter. In this thesis, we consider the CMAB problem in high-dimensional settings where the reward depends on only a subset of context and arm dimensions and we aim to exploit this information to provide personalized treatment in blood glucose regulation of type 1 diabetes mellitus patients.

We study this setting from two perspectives. First, we consider the setting given in [11] which is a classical CMAB setting where an upper bound on the number relevant dimensions for context and arm sets are available a priori to the learner and the set of relevant context dimensions are allowed to be different for different arms. As the number of possible context-arm pairs is infinite, any discretization technique requires some notion of a similarity constraint defined on the reward distributions and the context-arm pairs. This assumption is needed in order to ensure that expected rewards in a discretized region does not vary largely and estimations made for this region are accurate for most of the region. This sort of constraint on the relation between the expected rewards and the context-arm pairs is also needed in general to achieve sublinear regret. A nice example why this sort of assumption is needed is as follows. Consider a setting where, for all contexts, all arms have 0 expected reward except for a single arm that has expected reward of 1, which makes it optimal. Then any algorithm needs to identify this optimal arm in order to minimize regret. In finite amount of time, identifying the optimal arm in a set of infinitely many arms without any information gain from other arm selections is not possible, as the learner needs to play the exact optimal arm to realize its presence. Hence, in [11], the commonly

used Lipschitz continuity([12], [13]) is also assumed. Lipschitz continuity assumption states that the variation of expected rewards of any two context-arm pairs is bounded by the distance between the said context-arm pairs times a constant, called the Lipschitz constant. For this problem, a novel partitioning technique is proposed in [11], which discretizes subsets of dimensions of the context-arm space instead of naively applying discretization on the context-arm space. Then, it uses comparisons of the variation between the sample mean rewards of partitions to determine an estimated set of relevant context dimensions for each arm. The arm selection is done via employing optimism in the face of uncertainty in an effort to balance the exploration-exploitation trade-off. Optimism in the face of uncertainty can be explained as building optimistic indices for arms and selecting the arm which has the highest index when we lack the knowledge of true arm rewards. [11] adopts the classical regret definition as the performance metric. The performance of the proposed method is evaluated in experiment environments created synthetically and we apply it to the management of bolus insulin dosage in personalized treatments of diabetes patients.

Secondly, we consider the Bayesian version of the CMAB problem. In this setting, the underlying expected reward function is assumed to be sampled from a Gaussian process prior. We further assume that kernel function of the Gaussian process prior is such that any function sample is constant along some dimensions, yielding them irrelevant. For this problem, we propose an extension to the contextual Gaussian process upper confidence bound algorithm (CGP-UCB), originally introduced in [14]. Briefly, our extension is to consider kernels that can ignore certain dimensions(a discrete version of the automatic relevance determination model [15]), hence any estimate made with such kernels presume that expected reward is constant along ignored dimensions. During run-time, negative log marginal likelihood is optimized to determine which dimensions should be ignored based on the past context-arm-reward triplets. After the relevant dimensions are estimated, the UCBs of arms are estimated using the kernel associated with the estimated relevant dimensions and finally the arm with the highest UCB is selected. Again, we use the classical notion of regret and cumulative rewards as performance metrics. The comparison between CGP-UCB and our extension

is examined in settings with irrelevant context and arm dimensions, specifically in a synthetically created problem instance and in a problem instance which is based on real-world medical dataset collected from diabetes patients.

1.1 Our Contributions

The summary of contributions of this thesis are given below:

- In personalizing blood glucose control via contextual multi-armed bandits by identifying relevance;
  - We propose a way to integrate CMAB decision-making into blood glucose regulation of type 1 diabetes mellitus patients and show how data collected previously can be utilized in doing so.
  - We use CMAB-RL algorithm given in [11], an extension of HOO algorithm given in [16] and the IUP algorithm given in [9] to evaluate how different CMAB algorithms measure in bolus insulin administration task.
  - We also provide a comparison of these algorithms in a synthetic setting as a benchmark.
- In contextual Gaussian process bandit problem with relevance learning;
  - To the best of our knowledge, this work is the first to address relevance learning in a contextual Gaussian process bandit setting with infinitely many contexts and arms.
  - We propose contextual Gaussian process bandit upper confidence bound algorithm with relevance learning (CGP-UCB-RL), which does not require a priori information about the upper bounds on the number of relevant context and arm dimensions.
  - We compare the performance of CGP-UCB-RL with CGP-UCB numerically in a synthetic environment.

– We also investigate how suitable CGP-UCB-RL and CGP-UCB are for the bolus insulin administration task.

1.2 Organization of the Thesis

The organization of the thesis is as follows. In Chapter 2, we review the literature on MABs, CMABs, feature selection, personalized medicine and relevance learning in MABs and CMABs. Chapter 3 contains contextual multi-armed bandit problem with relevance learning given in [11] and our experiments about the personalized treatment using CMAB-RL and two other approaches. In Section 3.1, we give problem formulation. In Section 3.2, CMAB-RL algorithm is described including notes about the memory requirements as well as computational complexity and finally, in Section 3.3 we provide the illustrative experimental results on how CMAB-RL can be utilized for personalized treatment of diabetes patients. In Chapter 4, we introduce contextual Gaussian process bandit problem with relevance learning, with formal problem definition on Section 4.1, introduction of CGP-UCB-RL on Section 4.2 and numerical results about CGP-UCB-RL on Section 4.3. Lastly, Chapter 5 includes ideas for future work and concludes the thesis.

Chapter 2

Literature Review

In this chapter, we provide a review of literature on works that study MABs in various settings. Specifically, in Section 2.1 we present work on non-contextual bandits. Section 2.2 includes studies on contextual multi-armed bandits. Later, in Section 2.3, we discuss feature selection in offline and online settings. Section 2.4 contains work specifically on relevance learning in the non-contextual and contextual MAB settings. Finally, literature review on machine learning methods for medicinal applications is given in 2.5. A summary of this section and comparison of our work with the prior art is given in Table 2.1.

2.1 Non-contextual Multi-Armed Bandit

The earlier studies on MAB problems consider only a finite set of K arms, as introduced in [23], where the learner selects a single arm in each round and observes a noisy reward sampled from the reward distribution of the selected arm. This study shows that $O(\log T)$ is a lower bound on the regret up to a constant that is determined by the Kullback-Leibler divergence between the distributions of the optimal arm and the suboptimal arms. They also develop index policies that asymptotically match this lower bound. In [24], the aforementioned lower

Table 2.1: Summary of Related Work

Bandit algorithm	Contextual	Infinite arm set	Gaussian process prior	Relevance learning
UCB1 [17]	No	No	No	No
HOO [16]	No	Yes	No	No
GP-UCB [18]	No	Yes	Yes	No
Contextual Zooming Algorithm [12]	Yes	Yes	No	No
CGP-UCB [14]	Yes	Yes	Yes	No
Algorithm 3 [19]	Yes	Yes	Yes	No
SI-BO [20]	No	Yes	Yes	Yes
CAB [21]	No	Yes	No	Yes
RELEAF [22]	Yes	No	No	Yes
CMAB-RL [11]	Yes	Yes	No	Yes
CGP-UCB-RL(our work)	Yes	Yes	Yes	Yes

bound in matched by establishing index policies whose dependence on rewards of arms are only via their sample means. In a finite time setting, upper confidence bound(UCB) based index computation which only utilizes the current round number, sample mean rewards and selection counts for each arm is proposed in [17] achieving logarithmic regret. There are other works that study this setting, including [25] which achieves tighter bounds by using Kullback-Liebler divergence based UCB indices.

The problem is extended to infinite arms in many works as well, such as [26] which primarily considers one-dimensional continuum-armed bandit problem. In [26], the proposed algorithm divides the overall horizon into epochs where each epoch is twice as long the previous one. The arm space is partitioned into finer and finer grids with each epoch and in each epoch, an algorithm designed for a finite armed bandit setting is run. In [16], a more general setting is considered where the arms are allowed to be multi-dimensional. The hierarchical optimistic

optimization strategy builds non-uniform partitions structured as a binary tree and increases the granularity of the partitioning in regions where it believes to contain higher rewards, which in return enables it to identify the optimal arm with small discretization error. In continuum-armed problems, a notion of similarity is required to achieve sublinear regret. For example, in [26] Lipschitz continuity and in [16] a variant of Lipschitz continuity, which is called weak Lipschitz property, is assumed.

Another work, [18], assumes that the reward surface is a sample from a Gaussian process prior and utilizes Bayesian optimization to construct UCB bounds via the posterior distribution of the expected reward surface after observing samples.

2.2 Contextual Multi-Armed Bandit

In contextual MAB, the learner observes a context vector at the beginning of a round and has to shape its arm selection based on this context, as the expected reward of arms changes with different contexts. In general, the context set is considered to contain an infinite number of elements, whereas arm set is considered to have either finite or infinite number of elements in different studies. In CMAB setting, similar to the MAB setting with infinite number of arms, one requires further assumptions on the expected rewards and context-arm pairs. The studies in CMAB can be categorized into three main groups in terms of these assumptions.

First category includes the works which assume that the expected reward of an arm is given by a linear combination of elements of context and the arm. Although this model seems to consider a restricted set of CMAB problems, algorithms that adopt this model work well in practice and provide regret bounds with low dependence on dimensionality. LinUCB [7], which provides empirical results, and its modified version SupLinUCB [27], in which a theoretical analysis is given, are examples that adopt the linearity assumption. In [28], a variant of

the aforementioned works that makes use of kernel functions is proposed. This method achieves similar regret to [27] depending on the effective dimensionality of data instead of the total dimensionality. Effective dimensionality is a rough measure of number of dimensions in the reproducing kernel Hilbert space that data mostly resides in. Markedly, in [29], a better regret analysis is given through the use of more refined confidence sets.

The second category targets a more general set of CMAB problems. In general, the assumption is that the expected rewards associated with every context-arm pair form a Lipschitz continuous function with respect to the distances between contexts-arm pairs. In this setting, no statistical assumptions are made on context arrivals and the expected reward function is unknown to the learner but fixed. In [10], Query-Ad-Clustering algorithm that partitions the context space into subsets is proposed, where the model considers finitely many arms. The regret of Query-Ad-Clustering algorithm depends on the covering dimension of the context space, which is d for the d-dimensional Euclidean space. In the continuous context-arm space setting, Contextual Zooming Algorithm in [12] adaptively partitions the joint context-arm space, creating smaller sized sets around the high reward areas. For the Contextual Zooming Algorithm, a regret bound that depends on the zooming dimension of the context-arm space is derived, where the zooming dimension is determined by the size of near-optimal arm set. The problem is also considered in the Bayesian setting by [14] and [19], which assumes that the expected reward function is drawn from a Gaussian process prior. While CGP-UCB algorithm in [14] extends the work in [18] to the contextual setting intuitively, the work in [19] is inspired by the adaptive partitioning strategy given in [16].

Contexts and arm rewards are assumed to be jointly sampled from a fixed distribution in the last category. The studies in [30], [31] and [32] provide regret bounds that do not depend on the dimensionality of the context set in a setting with finitely many arms.

2.3 Feature Selection

The feature selection literature can be grouped into three as filter, wrapper and embedded approaches. Filter methods perform feature selection according to metrics such as the correlation coefficient or the mutual information of a feature with a target variable(class labels or target regression values). Wrapper methods work with a learning model, such as a classifier or a regression model, and selects the features according to the model's feedback. In embedded methods, feature selection is carried out simultaneously in the training process of the model.

As examples for the embedded approaches, [33] is a very famous paper that introduces LASSO regularization and the work in [34] utilizes regularized trees in order to perform feature selection. In [35], the model uses a modified objective function that penalizes the involvement of a feature.

As an example of wrapper methods, in [36], an iterative algorithm that removes features with the smallest ranking is proposed, with the ranks being calculated according to the feedback of a classifier. The help of a genetic algorithm is used in [37] to select features. [38] proposes two methods, namely SFFS and SBFS, in which the features are included and excluded according to their effect on the loss function, which can typically be the loss function of a classifier.

The filter methods do not make use of a classifier, as an example the weighting procedure in [39] can be given. In [40], an information-theoretic approach for determination of relevant features is given. [41] proposes a gradient based approach that looks to exploit the best of both worlds, aiming to merge the precision of wrapper methods with the speed of filter methods.

Apart from the above, there are papers that consider an online setting where the features are revealed sequentially. Generally filter methods are considered due to speed and compatibility in these settings. The methods given in [42], [43] and [44] can be considered as examples for this setting.

Even though there are plenty of feature selection methods in literature, unfortunately most of them can not be utilized in CMAB problems because in CMAB problems the aim is to maximize the cumulative reward but the aforementioned literature does not address this objective. We do, however, employ a version of the automatic relevance determination method given in [15] as part of our research in CGP-UCB-RL. This method allows different weights to be assigned to each feature and measure their relevance via these weights. In this sense, this method is similar to [39]. The optimal weights needs to be calculated according to a non-convex optimization on the negative log marginal likelihood.

2.4 Relevance Learning in Bandits

This section in literature review includes specific work on relevance learning in non-contextual and contextual MAB problems. Although not many sources are available in this area, there is a handful of important studies. First we consider non-contextual settings. In [21], a discretization strategy which cleverly partitions the arm space so that whatever arm is played the discretization error in relevant dimensions are small is proposed. This way [21] enjoys a regret bound with the time order that depends on the dimensionality of the relevant arm subspace, rather than the whole dimensionality of the arm space. Also in the Bayesian version of the CMAB problem, [20] proposes a model where the expected reward function essentially lives on a low-dimensional subspace of the original arm space and the transformation from the arm space to the low-dimensional subspace is given by a linear transformation. The proposed SI-BO algorithm first purely explores the arm space in order to learn the transformation matrix, then it uses the GP-UCB approach, given in [18], on the transformed space. These methods are not applicable in our setting as they do not consider the side information in the form of contexts.

Next we investigate the contextual models in relevance learning. In [45], a setting with finitely many arms and contexts is considered, whereas we consider sets for infinitely many contexts and arms. The proposed method determines

the relevant context dimensions with the use of conditional probabilities. The last work we consider is the RELEAF algorithm given in [22]. RELEAF adopts the Lipschitz continuity assumption and requires an upper bound on the number of relevant context dimensions. This approach is the closest work to [11] in the sense that it also creates a candidate set of relevant context dimensions in a similar way. Different from [11], it adaptively partitions the context space, their regret definition also considers the cost of observing rewards and most importantly, it diverges from [11] as it considers a finite set of arms whereas in [11] an infinite number of arms is considered.

2.5 Machine Learning for Personalized Medicine

The final section in literature review contains work on personalized medicine and machine learning methods in medicine in general.

In [46], C-Path method which analyzes cancer images and predicts prognosis is given and an analysis of morphological features in terms their relevance to the task is included. [47] and [48] uses deep learning and XGBoost methods, respectively, to predict blood glucose levels of patients. Although their methods do not directly come up with recommendations, their models can be used with various inputs to recommend treatments for different patients. In [49], the diagnosis of ischaemic heart disease using various machine learning approaches is considered. In [50], a comprehensive study with many different machine learning approaches is given for the heart failure subtype classification problem. In [51], a method that utilizes similarities between different patients and drugs to tackle the problem of personalized medicine is given. In [52], the DE method which discovers the relevant patient traits and utilizes them in making accurate diagnosis is proposed. They consider the case where for different treatments, different patient traits can be relevant. In [8], the infinite horizon Bayesian Bernoulli MAB and its finite horizon variant are used in the design and analysis of clinical trials. Another study is conducted on breast cancer diagnosis in [9], where an ensemble learning method called Hedged Bandits is proposed. Other than treatments of diseases

and diagnosis, [53] focuses on when patients should be admitted to the ICU by constructing personalized risk scores.

Chapter 3

Personalizing Blood Glucose Control via Contextual Multi-Armed Bandits by Identifying Relevance

In this chapter, we study how CMAB algorithms can be applied to the task of blood glucose regulation of diabetes patient via the optimization of bolus insulin dosage. We consider two approaches that do not take relevance information into account and the CMAB-RL [11] approach that considers relevance information. For completeness, we provide problem formulation and algorithm description for CMAB-RL in Sections 3.1 and 3.2, respectively. In Section 3.3, the evaluation of CMAB-RL and other approaches in personalized treatment is given. This section also includes a synthetic setup which allows us to investigate the performances of these algorithms outside of the scope of personalized treatment.

3.1 Problem Definition

In [11], the sequential decision-making problem is considered, where in each round $t \in \{1, 2, \ldots\}$, the learner observes a d_x-dimensional context $x(t) \in \mathcal{X}$ where $\mathcal{X} \coloneqq [0,1]^{d_x}$. Afterwards, it chooses a d_a-dimensional arm $a(t)$ from $\mathcal{A} \coloneqq [0,1]^{d_a}$. Let $\mathcal{F} \coloneqq \mathcal{X} \times \mathcal{A}$ denote the set of context-arm pairs and $\mu_a(x)$ denote the expected reward of a context-arm pair $(x, a) \in \mathcal{F}$. The random reward $r(t)$ is generated according to the context and arm selection in the given round, specifically given by $r(t) \coloneqq \mu_{a(t)}(x(t)) + \kappa(t)$. $\kappa(t)$ is the noise process which is conditionally 1-sub-Gaussian hence, $\forall \lambda \in \mathbb{R}$, it satisfies

$$\mathrm{E}[e^{\lambda \kappa(t)} \mid a_{1:t}, x_{1:t}, \kappa_{1:t-1}] \leq \exp(\lambda^2/2)$$

where for $b \in \{a, x, \kappa\}$, $b_{1:t} \coloneqq (b(1), \ldots b(t))$.

The set of arm dimensions is denoted with $\mathcal{D}_a \coloneqq \{1, \ldots, d_a\}$ and the $|z|$-dimensional subspace of $\mathcal{A}$ is denoted with $\mathcal{A}_{\boldsymbol{z}} \coloneqq [0,1]^{|\boldsymbol{z}|}$, for any $\boldsymbol{z} \subseteq \mathcal{D}_a$. Similarly, the $|z|$-dimensional subarm is denoted using $a_{\boldsymbol{z}} \in \mathcal{A}_{\boldsymbol{z}}$. An arm $a \in \mathcal{A}$ can be represented as an union of two components $a = \{a_{\boldsymbol{z}}, a_{\boldsymbol{z}'}\}$ where $\boldsymbol{z} \subseteq \mathcal{D}_a$ and $\boldsymbol{z}' \subseteq \mathcal{D}_a \setminus \boldsymbol{z}$. In this setting, it is assumed that for fixed $x \in \mathcal{X}$, expected reward $\mu_a(x)$ is constant along the irrelevant arm dimensions. In order to define this behaviour of the expected reward function mathematically, let $\boldsymbol{c} \subseteq \mathcal{D}_a$ denote the set of relevant arm dimensions. Then, $\mu_{\{a_{\boldsymbol{z}}, a_{\mathcal{D}_a \setminus \boldsymbol{z}}\}}(x) = \mu_{\{a'_{\boldsymbol{z}}, a_{\mathcal{D}_a \setminus \boldsymbol{z}}\}}(x)$, for all $\boldsymbol{z} \subseteq \mathcal{D}_a \setminus \boldsymbol{c}$, $a_{\boldsymbol{z}}, a'_{\boldsymbol{z}} \in \mathcal{A}_{\boldsymbol{z}}$, $a_{\mathcal{D}_a \setminus \boldsymbol{z}} \in \mathcal{A}_{\mathcal{D}_a \setminus \boldsymbol{z}}$ and $x \in \mathcal{X}$, is assumed.

Likewise, the set of context dimensions is denoted with $\mathcal{D}_x \coloneqq \{1, \ldots, d_x\}$ and the $|z|$-dimensional subspace of $\mathcal{X}$ is denoted with $\mathcal{X}_{\boldsymbol{z}} \coloneqq [0,1]^{|\boldsymbol{z}|}$, for any $\boldsymbol{z} \subseteq \mathcal{D}_x$. Also, the $|z|$-dimensional subcontext is denoted using $x_{\boldsymbol{z}} \in \mathcal{X}_{\boldsymbol{z}}$. A context $x \in \mathcal{X}$ can also be represented as a union of two components $x = \{x_{\boldsymbol{z}}, x_{\boldsymbol{z}'}\}$ where $\boldsymbol{z} \subseteq \mathcal{D}_x$ and $\boldsymbol{z}' \subseteq \mathcal{D}_x \setminus \boldsymbol{z}$. It is assumed that for fixed $a \in \mathcal{A}$, expected reward $\mu_a(x)$ only changes along the relevant context dimensions for arm a. The set of relevant context dimensions may correspond to a different subset of dimensions for different arms, hence the set of relevant context dimensions is defined as a function of an arm. More formally, let $\boldsymbol{c}_a$ denote the subset of $\mathcal{D}_x$ that contains the relevant

context dimensions for any $a \in \mathcal{A}$. Then, it is assumed that $\mu_a(\{x_{\boldsymbol{z}}, x_{\mathcal{D}_x \setminus \boldsymbol{z}}\}) = \mu_a(\{x'_{\boldsymbol{z}}, x_{\mathcal{D}_x \setminus \boldsymbol{z}}\})$ holds for all $a \in \mathcal{A}$, $\boldsymbol{z} \subseteq \mathcal{D}_x \setminus \boldsymbol{c}_a$, $x_{\boldsymbol{z}}, x'_{\boldsymbol{z}} \in \mathcal{X}_{\boldsymbol{z}}$ and $x_{\mathcal{D}_x \setminus \boldsymbol{z}} \in \mathcal{X}_{\mathcal{D}_x \setminus \boldsymbol{z}}$.

The optimal arm for a context x is defined as the arm that has the maximum expected reward given x, $a^*(x) \coloneqq \arg\max_{a \in \mathcal{A}} \mu_a(x)$. In CMAB problems a commonly considered performance metric is the contextual regret(see [12], [27]), given as

$$\mathrm{Reg}(T) \coloneqq \sum_{t=1}^{T} \mu_{a^*(x(t))}(x(t)) - \sum_{t=1}^{T} \mu_{a(t)}(x(t)).$$

which is also adopted in this work. Minimizing $\mathrm{Reg}(T)$ is equivalent to maximizing the expected cumulative reward over T rounds as $\mathrm{Reg}(T)$ essentially compares the expected cumulative reward of the learner against the best possible policy given a sequence of T contexts.

Unfortunately, since there are infinitely many arms and contexts, learning the optimal arm for each context is impossible. To solve this problem, a similarity structure is often assumed on expected reward function with respect to the set of context-arm pairs [12]. A modified version of this Lipschitz continuity assumption is utilized, which establishes a bound on the variation of the expected reward function between any two context-arm pairs in the relevant dimensions.

Assumption 1. *$\exists L > 0$ such that $\forall a, a' \in \mathcal{A}$ and $x, x' \in \mathcal{X}$, we have*

$$|\mu_a(x) - \mu_{a'}(x')| \leq L(\|x_{\boldsymbol{c}_a} - x'_{\boldsymbol{c}_a}\| + \|a_{\boldsymbol{c}} - a'_{\boldsymbol{c}}\|)$$

where $\|.\|$ represents the Euclidean norm.

Although Assumption 1, is stated according to the relevant context dimensions $\boldsymbol{c}_a$ of arm a, notice that since it is valid for all $a, a' \in \mathcal{A}$ it also implies

$$|\mu_a(x) - \mu_{a'}(x')| \leq L(\|x_{\boldsymbol{c}_{a'}} - x'_{\boldsymbol{c}_{a'}}\| + \|a_{\boldsymbol{c}} - a'_{\boldsymbol{c}}\|).$$

In this work, it is assumed that Lipschitz constant L is known by the learner, whereas $\mu_a(x)$ is not. Furthermore, let $\underline{d}_x \coloneqq \max_{a \in \mathcal{A}} |\boldsymbol{c}_a|$ and $\underline{d}_a \coloneqq |\boldsymbol{c}|$ denote the true number of relevant context and arm dimensions. It is assumed that the

upper bounds on the number of relevant context and arm dimensions, $\overline{d}_x$ and $\overline{d}_a$, are known and satisfy $\underline{d}_x \leq \overline{d}_x$ and $\underline{d}_a \leq \overline{d}_a$. When operating, CMAB-RL partitioning strategy requires partitions where $2\overline{d}_x$ dimensions are divided(see Section 3.2), $2\overline{d}_x \leq d_x$ is also needed. Overall the assumptions regarding the relevant number of dimensions can be summarized as

$$1 \leq \underline{d}_x \leq \overline{d}_x \leq d_x/2 \text{ and } 1 \leq \underline{d}_a \leq \overline{d}_a \leq d_a.$$

3.2 Contextual Multi-Armed Bandit with Relevance Learning Algorithm (CMAB-RL)

The pseudo-code for the CMAB-RL algorithm is given in Algorithms 1 and 2. In a nutshell, the CMAB-RL algorithm creates uniform partitions on $2\overline{d}_x$ and $\overline{d}_a$ dimensional subspaces of the context-arm space and forms a set of candidate dimensions that it regards to contain the relevant dimensions. Each set in the partitions uses the past contexts that arrived and arms that are selected to estimate the expected reward function.

In order to explain how CMAB-RL operates in detail, further notation needs to be introduced. Let $\wp(\mathcal{S})$ denote the power set of a set $\mathcal{S}$ and $\mathcal{V}_x^l \coloneqq \{\boldsymbol{v} \in \wp(\mathcal{D}_x) : |\boldsymbol{v}| = l\}$ and $\mathcal{V}_a^l \coloneqq \{\boldsymbol{v} \in \wp(\mathcal{D}_a) : |\boldsymbol{v}| = l\}$ denote the set of all l-tuples of context and arm dimensions for any $l \in \mathbb{Z}^+$, respectively. Also let $\mathcal{V}_x^l(\boldsymbol{v}) \coloneqq \{\boldsymbol{v}' \in \wp(\mathcal{D}_x) : |\boldsymbol{v}'| = l, \boldsymbol{v} \subseteq \boldsymbol{v}'\}$ for $\boldsymbol{v} \subseteq \mathcal{D}_x$ and $l \in \{|\boldsymbol{v}|, |\boldsymbol{v}| + 1, \ldots, \overline{d}_x\}$. In other words, for any $\boldsymbol{w} \in \mathcal{V}_x^l(\boldsymbol{v})$, we have the subset relation $\boldsymbol{v} \subseteq \boldsymbol{w}$.

CMAB-RL takes the context set $\mathcal{X}$, the arm set $\mathcal{A}$, the total number of rounds(horizon) T[1], the Lipschitz constant L given in Assumption 1, an integer upper bound on the number of relevant arm dimensions $\overline{d}_a \leq d_a$ and an

[1]Although a finite horizon T is needed as an input, CMAB algorithms can be run on infinite horizons by employing the well known doubling trick. The doubling trick involves setting exponentially increasing intervals one after another, typically as $T = 2T$. The doubling trick allows bandit algorithms designed for finite horizons to enjoy similar performance in an infinite horizon setup as in the finite horizon setup.

Algorithm 1 CMAB-RL

Input: $\mathcal{X}, \mathcal{A}, T, L, \overline{d}_x, \overline{d}_a$
Initialization: Set $m = \lceil T^{1/(2+2\overline{d}_x+\overline{d}_a)} \rceil$
$(\mathcal{C}(\mathcal{X}), \mathcal{Y}) = \text{Generate}(\mathcal{X}, \mathcal{A}, \overline{d}_x, \overline{d}_a, m)$
Set $\hat{\mu}_{y,p_{\boldsymbol{w}}}(0) = 0$, $N_{y,p_{\boldsymbol{w}}}(0) = 0$ for all $y \in \mathcal{Y}$, $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$, $p_{\boldsymbol{w}} \in \mathcal{P}_{\boldsymbol{w}}$
while $1 \leq t \leq T$ **do**
  Observe $x(t)$ and for each $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$, find $p_{\boldsymbol{w}}(t) \in \mathcal{P}_{\boldsymbol{w}}$ that $x(t)$ belongs to
  Compute $\mathcal{R}_y(t)$ for all $y \in \mathcal{Y}$ as given in (3.1)
  for $y \in \mathcal{Y}$ **do**
    if $\mathcal{R}_y(t) = \emptyset$ **then**
      Randomly select $\hat{\boldsymbol{c}}_y(t)$ from $\mathcal{V}_x^{\overline{d}_x}$
    else
      For each $\boldsymbol{v} \in \mathcal{R}_y(t)$, calculate $\hat{\sigma}^2_{y,\boldsymbol{v}}(t) = \max_{\boldsymbol{w},\boldsymbol{w}' \in \mathcal{V}_x^{2\overline{d}_x}(\boldsymbol{v})} |\hat{\mu}_{y,\boldsymbol{w}}(t) - \hat{\mu}_{y,\boldsymbol{w}'}(t)|$
      Set $\hat{\boldsymbol{c}}_y(t) = \arg\min_{\boldsymbol{v} \in \mathcal{R}_y(t)} \hat{\sigma}^2_{y,\boldsymbol{v}}(t)$
    end if
    Calculate $\hat{\mu}_y^{\hat{\boldsymbol{c}}_y(t)}(t) = \dfrac{\sum_{\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}(\hat{\boldsymbol{c}}_y(t))} \hat{\mu}_{y,\boldsymbol{w}}(t) N_{y,\boldsymbol{w}}(t)}{\sum_{\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}(\hat{\boldsymbol{c}}_y(t))} N_{y,\boldsymbol{w}}(t)}$
    Determine $\boldsymbol{w}_y(t) = \arg\max_{\boldsymbol{w}' \in \mathcal{V}_x^{2\overline{d}_x}} u_{y,\boldsymbol{w}'}(t)$
  end for
  Select $y(t) = \arg\max_{y \in \mathcal{Y}} \hat{\mu}_y^{\hat{\boldsymbol{c}}_y(t)}(t) + 5u_{y,\boldsymbol{w}_y(t)}(t)$
  Update estimates and the counters given for all $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$
end while

integer upper bound on the number of relevant context dimensions $\overline{d}_x \leq d_x/2$ as inputs. It sets $m = \lceil T^{1/(2+2\overline{d}_x+\overline{d}_a)} \rceil$.

During initialization CMAB-RL utilizes the similarity constraint given in Assumption 1, which assures context-arm pairs that are close to each other in $\mathcal{F}$ to have similar expected rewards and discretizes $\mathcal{X}$ and $\mathcal{A}$. However instead of doing the partitioning naively on all dimensions it creates partitions on the subsets of the context and arm spaces according to the upper bounds on the number of relevant context and arm dimensions as irrelevant dimensions need not be partitioned since the expected reward does not change along irrelevant dimensions. An example of how an arm space with $d_a = 2$ and $\overline{d}_a = 1$ would be discretized is given in Fig. 3.1. Expected reward function being constant along one of the dimensions means that the optimal arm is a line along the irrelevant dimension

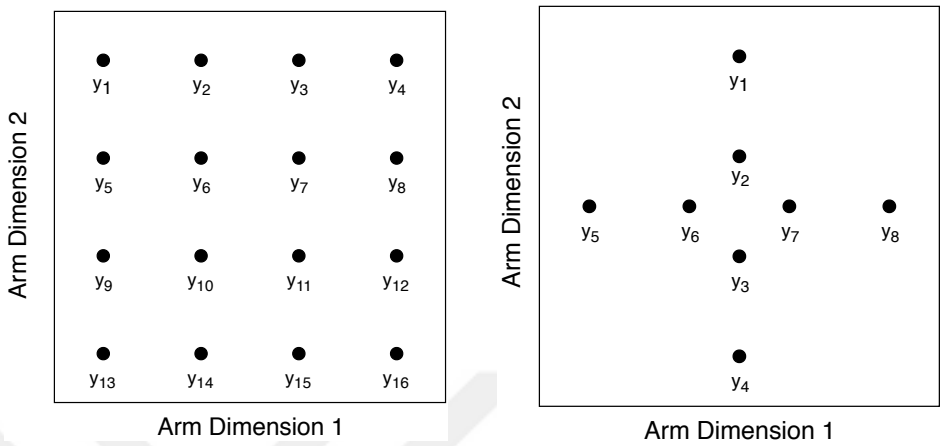
Algorithm 2 Generate

Input: $\mathcal{X}, \mathcal{A}, \overline{d}_a, \overline{d}_x, m$
Create $\mathcal{I}_i := \{[0, \frac{1}{m}], (\frac{1}{m}, \frac{2}{m}], \ldots, (\frac{m-1}{m}, 1]\}$ and $\mathcal{P}_i := \{[0, \frac{1}{m}], (\frac{1}{m}, \frac{2}{m}], \ldots, (\frac{m-1}{m}, 1]\}$
Generate $\mathcal{V}_a^{\overline{d}_a}$ and $\mathcal{V}_x^{2\overline{d}_x}$
for $\boldsymbol{v} \in \mathcal{V}_a^{\overline{d}_a}$ **do**
    $\mathcal{I}_{\boldsymbol{v}} = \prod_{i \in \boldsymbol{v}} \mathcal{I}_i$
end for
for $w \in \mathcal{V}_x^{2\overline{d}_x}$ **do**
    $\mathcal{P}_{\boldsymbol{w}} = \prod_{i \in \boldsymbol{w}} \mathcal{P}_i$
end for
$\mathcal{C}(\mathcal{A}) = \bigcup_{\boldsymbol{v} \in \mathcal{V}_a^{\overline{d}_a}} \mathcal{I}_{\boldsymbol{v}}$ and $\mathcal{C}(\mathcal{X}) := \bigcup_{\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}} \mathcal{P}_{\boldsymbol{w}}$
Index the geometric center of each set in $\mathcal{C}(\mathcal{A})$ by y and generate the set of arms $\mathcal{Y}$
return $\mathcal{C}(\mathcal{X})$ and $\mathcal{Y}$

for this 2-D example. Using CMAB-RL strategy, one guarantees that at least one of the arms is close to this optimal arm line with bounded discretization error. As it can be seen CMAB-RL strategy yields fewer number of arms without having an extra discretization error as the reward surface is constant either along arm dimension 1 or 2.

Precisely, first CMAB-RL generates the set $\mathcal{V}_a^{\overline{d}_a}$ and then partitions each dimension in the arm subspace $\mathcal{A}_{\boldsymbol{v}}$ into m equal intervals for all $\boldsymbol{v} \in \mathcal{V}_a^{\overline{d}_a}$. In this manner, $\mathcal{A}_{\boldsymbol{v}}$ is partitioned into $m^{\overline{d}_a}$ non-overlapping sets. Let $\mathcal{I}_{\boldsymbol{v}} := \prod_{i \in \boldsymbol{v}} \mathcal{I}_i$ denote the partition formed on $\mathcal{A}_{\boldsymbol{v}}$, where $\mathcal{I}_i := \{[0, \frac{1}{m}], (\frac{1}{m}, \frac{2}{m}], \ldots, (\frac{m-1}{m}, 1]\}$ is the partition for a single dimension $i \in \boldsymbol{v}$. As CMAB-RL applies this partition technique for all $\boldsymbol{v} \in \mathcal{V}_a^{\overline{d}_a}$, the collection of all partitions is then given by $\mathcal{C}(\mathcal{A}) := \cup_{\boldsymbol{v} \in \mathcal{V}_a^{\overline{d}_a}} \mathcal{I}_{\boldsymbol{v}}$ where $|\mathcal{C}(\mathcal{A})| = \binom{d_a}{\overline{d}_a} m^{\overline{d}_a}$. After the partitioning operation, we index the geometric centers of sets in the partitions in $\mathcal{C}(\mathcal{A})$ by y, and the set of all geometric centers is denoted by $\mathcal{Y}$. For an arm y that coincides with a geometric center in $\mathcal{I}_{\boldsymbol{v}}$, the values for the dimensions not included in $\boldsymbol{v}$ are set to 0.5.[2] Once arm discretization is completed, we essentially have a discrete arm set with $|\mathcal{C}(\mathcal{A})|$ elements.

[2]The value 0.5 is selected only for simplicity, certainly any value in $[0, 1]$ would work as the dimensions not included in $\boldsymbol{v}$ are not partitioned.



(a) Uniform discretization

(b) CMAB-RL discretization

Figure 3.1: Comparison of Uniform and CMAB-RL Discretization Strategies

For the partitioning of context space, CMAB-RL creates the set $\mathcal{V}_x^{2\bar{d}_x}$ similar to the case in arm partitioning. Then it partitions each dimension in the context subspace $\mathcal{X}_{\boldsymbol{w}}$ into m equal intervals for all $\boldsymbol{w} \in \mathcal{V}_x^{2\bar{d}_x}$. Hence $\mathcal{X}_{\boldsymbol{w}}$ is then partitioned into $m^{2\bar{d}_x}$ non-overlapping sets. Let $\mathcal{P}_{\boldsymbol{w}} := \prod_{i \in \boldsymbol{w}} \mathcal{P}_i$ denote the partition formed on $\mathcal{X}_{\boldsymbol{w}}$, where $\mathcal{P}_i := \{[0, \frac{1}{m}], (\frac{1}{m}, \frac{2}{m}], \ldots, (\frac{m-1}{m}, 1]\}$ is the partition for a single dimension $i \in \boldsymbol{w}$. Since CMAB-RL also applies this partition technique for all $\boldsymbol{w} \in \mathcal{V}_x^{2\bar{d}_x}$, the collection of all partitions is then given by $\mathcal{C}(\mathcal{X}) := \cup_{\boldsymbol{w} \in \mathcal{V}_x^{2\bar{d}_x}} \mathcal{P}_{\boldsymbol{w}}$ where $|\mathcal{C}(\mathcal{X})| = \binom{d_x}{2\bar{d}_x} m^{2\bar{d}_x}$.

For simplicity, let $x \in p_{\boldsymbol{w}}$ for $\boldsymbol{w} \in \mathcal{V}_x^{2\bar{d}_x}$ for any $x \in \mathcal{X}$ if $x_{\boldsymbol{w}} \in p_{\boldsymbol{w}}$ for $p_{\boldsymbol{w}} \in \mathcal{P}_{\boldsymbol{w}}$. Moreover, let $p_{\boldsymbol{w}}(t) \in \mathcal{P}_{\boldsymbol{w}}$ denote the set that $x_{\boldsymbol{w}}(t)$ belongs to.

CMAB-RL stores a sample mean estimate and a sample counter for each combination of elements of $\mathcal{C}(\mathcal{X})$ and $\mathcal{Y}$. For each $\boldsymbol{w} \in \mathcal{V}_x^{2\bar{d}_x}$, $p_{\boldsymbol{w}} \in \mathcal{P}_{\boldsymbol{w}}$ and $y \in \mathcal{Y}$, $N_{y,p_{\boldsymbol{w}}}(t)$ denotes the sample counter which simply keeps track of how many times a context was in $p_{\boldsymbol{w}}$ and arm y was selected before round t. Similarly, $\hat{\mu}_{y,p_{\boldsymbol{w}}}(t)$ keeps the sample mean reward estimate that is obtained from rounds before the t^{th} round, when a context was in $p_{\boldsymbol{w}}$ and arm y was selected. All counters and sample means are set to 0 during initialization.

As all aspects of the initialization is explained in detail, we next explain the algorithm details during run-time. The arm selection rule requires

one to calculate a term called uncertainty term, which is monotonically decreasing with the sample size. More formally, it is defined as $u_{y,p_{\boldsymbol{w}}}(t) \coloneqq \sqrt{(2+4\log(2|\mathcal{Y}|\binom{d_x-1}{2\overline{d}_x-1}m^{2\overline{d}_x}T^{3/2}))/N_{y,p_{\boldsymbol{w}}}(t)}$ for all $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$, $p_{\boldsymbol{w}} \in \mathcal{P}_{\boldsymbol{w}}$, $y \in \mathcal{Y}$. Notice that in each round t, $p_{\boldsymbol{w}}(t)$ is unique for each $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$, hence for simplicity of notation, let $\hat{\mu}_{y,\boldsymbol{w}}(t) \coloneqq \hat{\mu}_{y,p_{\boldsymbol{w}}(t)}(t)$, $u_{y,\boldsymbol{w}}(t) \coloneqq u_{y,p_{\boldsymbol{w}}(t)}(t)$ and $N_{y,\boldsymbol{w}}(t) \coloneqq N_{y,p_{\boldsymbol{w}}(t)}(t)$. The sample mean estimate for an arm $y \in \mathcal{Y}$ for the tuple of context dimensions $\boldsymbol{v} \in \mathcal{V}_x^{\overline{d}_x}$ in round t is defined as the weighted average over all sample means in counters for $\boldsymbol{w} \in \mathcal{V}_x^l(\boldsymbol{v})$,

$$\hat{\mu}_y^{\boldsymbol{v}}(t) \coloneqq \frac{\sum\limits_{\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}(\boldsymbol{v})} \hat{\mu}_{y,\boldsymbol{w}}(t) N_{y,\boldsymbol{w}}(t)}{\sum\limits_{\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}(\boldsymbol{v})} N_{y,\boldsymbol{w}}(t)}.$$

In every round t, context $x(t)$ is revealed to CMAB-RL which then determines the set $p_{\boldsymbol{w}}(t)$ in $\mathcal{P}_{\boldsymbol{w}}$ that $x(t)$ is contained in, for each $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$. Then for each arm $y \in \mathcal{Y}$, the set of candidate relevant tuples of context dimensions, each of which is a $\overline{d}_x$-tuple, are constructed using pairwise comparisons between $2\overline{d}_x$-tuples,

$$\begin{aligned}\mathcal{R}_y(t) \coloneqq \Big\{ &\boldsymbol{v} \in \mathcal{V}_x^{\overline{d}_x} : |\hat{\mu}_{y,\boldsymbol{w}}(t) - \hat{\mu}_{y,\boldsymbol{w}'}(t)| \\ &\leq 2L\sqrt{\overline{d}_x}/m + u_{y,\boldsymbol{w}}(t) + u_{y,\boldsymbol{w}'}(t), \forall \boldsymbol{w}, \boldsymbol{w}' \in \mathcal{V}_x^{2\overline{d}_x}(\boldsymbol{v}) \Big\}. \end{aligned} \tag{3.1}$$

The term $2L\sqrt{\overline{d}_x}/m + u_{y,\boldsymbol{w}}(t) + u_{y,\boldsymbol{w}'}(t)$ is the summation of the uncertainty due to randomness in rewards and the uncertainty due to discretization for the estimates $\hat{\mu}_{y,\boldsymbol{w}}(t)$ and $\hat{\mu}_{y,\boldsymbol{w}'}(t)$, called the joint uncertainty. If $|\hat{\mu}_{y,\boldsymbol{w}}(t) - \hat{\mu}_{y,\boldsymbol{w}'}(t)|$ is larger than the joint uncertainty, then for $\boldsymbol{w}, \boldsymbol{w}' \in \mathcal{V}_x^{2\overline{d}_x}(\boldsymbol{v})$, it is inferred that $\boldsymbol{v}$ does not contain all of the relevant dimensions. The reason behind this is that even though $\boldsymbol{v} \subset \boldsymbol{w}$ and $\boldsymbol{v} \subset \boldsymbol{w}'$ are satisfied, the sample mean reward estimated differ from each other largely. If $\boldsymbol{v}$ did contain all of the relevant context dimensions, then estimates for $\boldsymbol{w}$ and $\boldsymbol{w}'$ would be similar to each other as variance of expected reward along irrelevant dimensions is only due to random noise and discretization error. Any estimate $\hat{\mu}_{y,\boldsymbol{w}}(t)$ and uncertainty $L\sqrt{\overline{d}_x}/m + u_{y,\boldsymbol{w}}(t)$ such that $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}(\boldsymbol{v})$ would contain the variance along relevant dimensions.

The tuple of estimated relevant context dimensions to be selected from the set

$\mathcal{R}_y(t)$ is denoted by $\hat{\boldsymbol{c}}_y(t)$. If $\mathcal{R}_y(t) = \emptyset$, then $\hat{\boldsymbol{c}}_y(t)$ is selected from $\mathcal{V}_x^{\overline{d}_x}$ arbitrarily. Otherwise, CMAB-RL selects $\hat{\boldsymbol{c}}_y(t)$ by considering the variation of sample mean reward estimates which, for arm $y \in \mathcal{Y}$ and $\boldsymbol{v} \in \mathcal{R}_y(t)$ is defined as

$$\hat{\sigma}^2_{y,\boldsymbol{v}}(t) \coloneqq \max_{\boldsymbol{w},\boldsymbol{w}' \in \mathcal{V}_x^{2\overline{d}_x}(\boldsymbol{v})} |\hat{\mu}_{y,\boldsymbol{w}}(t) - \hat{\mu}_{y,\boldsymbol{w}'}(t)|.$$

CMAB-RL chooses the $\overline{d}_x$-tuple of context dimensions with minimum variation as the selection $\hat{\boldsymbol{c}}_y(t)$ for all $y \in \mathcal{Y}$, precisely $\hat{\boldsymbol{c}}_y(t) \coloneqq \arg\min_{\boldsymbol{v} \in \mathcal{R}_y(t)} \hat{\sigma}^2_{y,\boldsymbol{v}}(t)$.

After determining $\hat{\boldsymbol{c}}_y(t)$ for all $y \in \mathcal{Y}$, CMAB-RL calculates the mean reward estimates $\hat{\mu}_y^{\boldsymbol{v}}(t)$ using $\boldsymbol{v} = \hat{\boldsymbol{c}}_y(t)$ for all $y \in \mathcal{Y}$. When selecting an arm, CMAB-RL does not directly use the mean reward estimates, but instead uses an inflated term which is an upper confidence bound(UCB) on the expected rewards. For arm y, in round t, let $\boldsymbol{w}_y(t) \coloneqq \arg\max_{\boldsymbol{w}' \in \mathcal{V}_x^{2\overline{d}_x}} u_{y,\boldsymbol{w}'}(t)$ denote the $2\overline{d}_x$-tuple of context dimensions with the largest uncertainty due to randomness in rewards. The UCB term for an arm y in round t is defined as

$$\mathrm{UCB}_y(t) \coloneqq \hat{\mu}_y^{\hat{\boldsymbol{c}}_y(t)}(t) + 5u_{y,\boldsymbol{w}_y(t)}(t).$$

Finally, CMAB-RL selects the arm y with the highest UCB for round t i.e. $y(t) = \arg\max_{y \in \mathcal{Y}} \mathrm{UCB}_y(t)$. Note that, we have $a(t) = y(t)$ as well. CMAB-RL employs the usage of UCB to balance the exploration-exploitation trade-off which allows it to explore arms that are rarely selected(and thus have high uncertainty), even though they have low sample mean rewards.

Next, how update is executed after the selection of arm $a(t)$ and observation of reward $r(t)$ is explained. CMAB-RL updates the sample mean rewards and the sample counters for $y(t)$ and for all $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$ as follows

$$\begin{aligned} \hat{\mu}_{y(t),\boldsymbol{w}}(t+1) &= \frac{\hat{\mu}_{y(t),\boldsymbol{w}}(t) N_{y(t),\boldsymbol{w}}(t) + r(t)}{N_{y(t),\boldsymbol{w}}(t) + 1} \text{ and} \\ N_{y(t),\boldsymbol{w}}(t+1) &= N_{y(t),\boldsymbol{w}}(t) + 1. \end{aligned} \tag{3.2}$$

Notice that CMAB-RL updates statistics for all $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$, which is a nice consequence of the partitioning strategy as there is exactly one $p_{\boldsymbol{w}}(t)$ for each $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$. Context $x(t)$ falls into multiple partitions(across different elements of $\mathcal{V}_x^{2\overline{d}_x}$), and

since the reward is for context-arm pair $(x(t), y(t))$ is observed, it is allowed to update all partitions that contain $(x(t), y(t))$. However, it is also important to point out that $\hat{\mu}_{y,\boldsymbol{w}}(t)$ and $N_{y,\boldsymbol{w}}(t)$ for $y \neq y(t)$ are not updated, thus $\hat{\mu}_{y,p_{\boldsymbol{w}}}(t+1) = \hat{\mu}_{y,p_{\boldsymbol{w}}}(t)$, $N_{y,p_{\boldsymbol{w}}}(t+1) = N_{y,p_{\boldsymbol{w}}}(t)$ for $y \neq y(t)$.

3.2.1 Memory Requirements of CMAB-RL

The memory requirements analysis for CMAB-RL is relatively brief. For all $y \in \mathcal{Y}$, $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$ and $p_{\boldsymbol{w}} \in \mathcal{P}_{\boldsymbol{w}}$, CMAB-RL stores sample mean reward estimates and sample counters. Hence the memory requirement is

$$O\left(\tbinom{d_a}{\overline{d}_a}\tbinom{d_x}{2\overline{d}_x} m^{2\overline{d}_x+\overline{d}_a}\right),$$

substituting $m = \lceil T^{1/(2+2\overline{d}_x+\overline{d}_a)} \rceil$, sublinear memory requirements at the order of $O(T^{(2\overline{d}_x+\overline{d}_a)/(2+2\overline{d}_x+\overline{d}_a)})$ is achieved.

3.2.2 Computational Complexity of CMAB-RL

Next, the computational complexity of CMAB-RL is investigated during runtime, which requires a deeper analysis. In each round t, determining

- $p_{\boldsymbol{w}}(t) \in \mathcal{P}_{\boldsymbol{w}}$ for all $\boldsymbol{w} \in \mathcal{V}_x^{2\overline{d}_x}$ requires $O\left(d_x + \tbinom{d_x}{2\overline{d}_x}\right)$
- $\mathcal{R}_y(t)$ for all $y \in \mathcal{Y}$ requires $O\left(\tbinom{d_a}{\overline{d}_a} m^{\overline{d}_a} \tbinom{d_x}{\overline{d}_x} \tbinom{d_x-\overline{d}_x}{\overline{d}_x}^2\right)$
- $\hat{\sigma}^2_{y,\boldsymbol{v}}(t)$ for all $y \in \mathcal{Y}$, $\boldsymbol{v} \in \mathcal{R}_y(t)$ requires $O\left(\tbinom{d_a}{\overline{d}_a} m^{\overline{d}_a} \tbinom{d_x}{\overline{d}_x} \tbinom{d_x-\overline{d}_x}{\overline{d}_x}^2\right)$
- $\hat{\boldsymbol{c}}_y(t)$ for all $y \in \mathcal{Y}$ requires $O\left(\tbinom{d_a}{\overline{d}_a} m^{\overline{d}_a} \tbinom{d_x}{\overline{d}_x}\right)$
- $\hat{\mu}_y^{\hat{\boldsymbol{c}}_y(t)}(t)$ for all $y \in \mathcal{Y}$ requires $O\left(\tbinom{d_a}{\overline{d}_a} m^{\overline{d}_a} \tbinom{d_x-\overline{d}_x}{\overline{d}_x}\right)$
- $\boldsymbol{w}_y(t)$ for all $y \in \mathcal{Y}$ requires $O\left(\tbinom{d_a}{\overline{d}_a} m^{\overline{d}_a} \tbinom{d_x}{2\overline{d}_x}\right)$

- $y(t)$ requires $O\left(\binom{d_a}{\bar{d}_a} m^{\bar{d}_a}\right)$

computations. Thus, overall considering the largest terms, the computational complexity of CMAB-RL becomes

$$O\left(\binom{d_a}{\bar{d}_a} m^{\bar{d}_a} \binom{d_x}{\bar{d}_x} \binom{d_x - \bar{d}_x}{\bar{d}_x}^2\right),$$

once more by substituting $m = \lceil T^{1/(2+2\bar{d}_x+\bar{d}_a)} \rceil$, sublinear computational complexity at the order of $O(T^{\bar{d}_a/(2+2\bar{d}_x+\bar{d}_a)})$ is achieved.

3.3 Illustrative Results

In this section, we compare the performance of CMAB-RL against other algorithms in two experiments. The first experiment involves a synthetic simulation environment with a multi-dimensional arm set(5 dimensions with only one relevant) as well as a multi-dimensional context set(again with 5 dimensions, only one being relevant). In our second experiment, we test the performance and suitability of CMAB-RL in a medical treatment scenario where CMAB-RL and other competitor algorithms are employed for the problem of bolus insulin administration for type 1 diabetes mellitus(T1DM) patients based on the OhioT1DM dataset [54].

3.3.1 Competitor Learning Algorithms

We consider two competitor algorithms to compare CMAB-RL against, with one employing a uniform partitioning strategy whereas the other adaptively partitions the context-arm space. We also consider an approach that selects arms randomly to establish a benchmark where needed.

3.3.1.1 Instance-based Uniform Partitioning (IUP) [9]

IUP is a CMAB algorithm that uniformly partitions the context-arm space $\mathcal{F}$ into $m^{d_x+d_a}$ sets, where they show that the choice of $m = \lceil T^{1/(2+d_x+d_a)} \rceil$ minimizes regret. In each round, IUP determines the subset of hypercubes that context arrives in, and then it identifies the hypercube with the highest UCB among subset of hypercubes that contain context. It then selects the arm from the identified hypercube. Note that, IUP does not consider relevance information, and operates directly on $\mathcal{F}$.

3.3.1.2 Contextual Hierarchical Optimistic Optimization (C-HOO)

We propose an extension to the hierarchical optimistic optimization(HOO) strategy given in [16] which can be applied to CMAB problems as well. [3] In the vanilla version, HOO models the arm set $\mathcal{A}$ with a binary tree structure and adaptively partitions $\mathcal{A}$ by growing the tree in each round. The root node corresponds to the entire arm set $\mathcal{A}$ and each of the subsequent nodes is mapped to a subset of $\mathcal{A}$. Smaller subsets are represented by deeper nodes in the tree. Regions that are represented by the nodes in same depth establish a partition on $\mathcal{A}$. For a node n, the union of regions that correspond to the children of n is equal to the region n corresponds to.

In each round, HOO starts at the root node and builds a path which will eventually end at a leaf node. The criteria when creating the path is such that at every depth, the child with the higher UCB is added to the path, this goes on until a node with at most one child is reached. Provided that the node has no children, a child is created randomly, otherwise the second child is created. After the path terminates, GOO selects an arm from the recently created child's corresponding region. As time increases, HOO essentially zooms into regions where the expected rewards are likely to be high.

[3] [19] also proposes a contextual extension of HOO for the CMAB problem with Gaussian process prior.

We extend HOO to the contextual case, called C-HOO. The immediate difference is that the tree is now built on $\mathcal{F}$ rather than $\mathcal{A}$. In each round, as it is a contextual approach, C-HOO observes the context. Although the path construction is similar to that of HOO, the main distinction is that before a selection according to UCB takes place, first the availability of the children are determined. The availability in this case refers to whether the child contains the content or not. Notice that, at each level at least one child must contain the context. If only one child is available, then that child is selected and added to the path, otherwise the child with the highest UCB is added to the path.

One drawback of vanilla version of HOO is that the computational complexity increases quadratically with the number of rounds. In [16], a truncated version of HOO is also proposed which essentially achieves the same regret bound(additive factor of $4\sqrt{T}$ which does not change the time order of regret) as vanilla version of HOO. We based C-HOO on the truncated version of HOO in our experiments. HOO or C-HOO does not take relevance information into consideration either.

3.3.1.3 Uniform Random

This algorithm does not take the current context, past observations or the relevance information into account. It simply selects an arm randomly from $\mathcal{A}$ and is used as a benchmark.

3.3.2 Parameters Used in the Experiments

For both experiments, the set of all feasible context-arm pairs $\mathcal{F}$, time horizon T, dimensionality of context and arm sets, i.e., d_x and d_a, are made known to the algorithms as required. We also assume that L is not known by any of the algorithms, hence it is set to 1. For CMAB-RL, we set $\overline{d}_x = \underline{d}_x$ and $\overline{d}_a = \underline{d}_a$. As for C-HOO, we set $v_1 = 2\sqrt{d_x + d_a}$ and $\rho = 2^{(-1/(d_x+d_a))}$ so that Assumption A1 in [16] is satisfied. The IUP algorithm requires no further parameters. It is important to note that the confidence terms(uncertainty terms due to randomness

in rewards) are scaled so that they are smaller which forces algorithms to favor exploitation over exploration. At first, this might not seem as an appropriate manipulation however when experimenting, it is observed that confidence terms start much larger compared to the sample mean reward estimations and dominate the statistics involved in arm selection. Decreasing confidence terms, so that they are comparable with the sample mean reward estimations increases their cumulative rewards. For each algorithm we determined the best scaling factor via grid search over the set $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.25, 0.5, 1\}$. Also, results of both experiments represent the results obtained from 20 repetitions, with the aim of minimizing the effect of randomness in context arrivals, arm selection and reward generation when measuring the performances.

3.3.3 Experiments on a Synthetic Simulation Environment

For the experiment with the synthetic environment, we consider $d_x = 5$, $d_a = 5$, $\underline{d}_x = 1$ and $\underline{d}_a = 1$. It is also assumed that $\boldsymbol{c}_a = \boldsymbol{c}_{a'}$ for all $a, a' \in \mathcal{A}$, meaning that the set of relevant context dimensions is the same for all arms. We consider a Gaussian mixture model for the expected reward function $\mu_a(x)$. For a point $(x, a) \in \mathcal{F}$, $\mu_a(x)$ is given by

$$\mu_a(x) = \min \left\{ s \sum_{i=1}^{K} \rho_i f((x, a)|\theta_i, \Sigma_i), 1 \right\}$$

where $\min\{\cdot, \cdot\}$ function is required so that 1-sub-Gaussianity is satisfied and we have $\sum_{i=1}^{K} \rho_i = 1$ and $\rho_i > 0$, for $1 \leq i \leq K$. For the i^{th} component, the component weight is given by ρ_i, the mean vector is given by θ_i and the covariance matrix is given by Σ_i. Moreover, the number of components is denoted by K, s denotes the scaling factor and f denotes the probability density function of a multivariate Gaussian distribution. The parameters that we considered for our experiment are $s = 0.25$, $K = 2$, $\rho_1 = \rho_2 = 0.5$, $\theta_1 = [0.25, 0.75]^T$, $\theta_2 = [0.5, 0.5]^T$

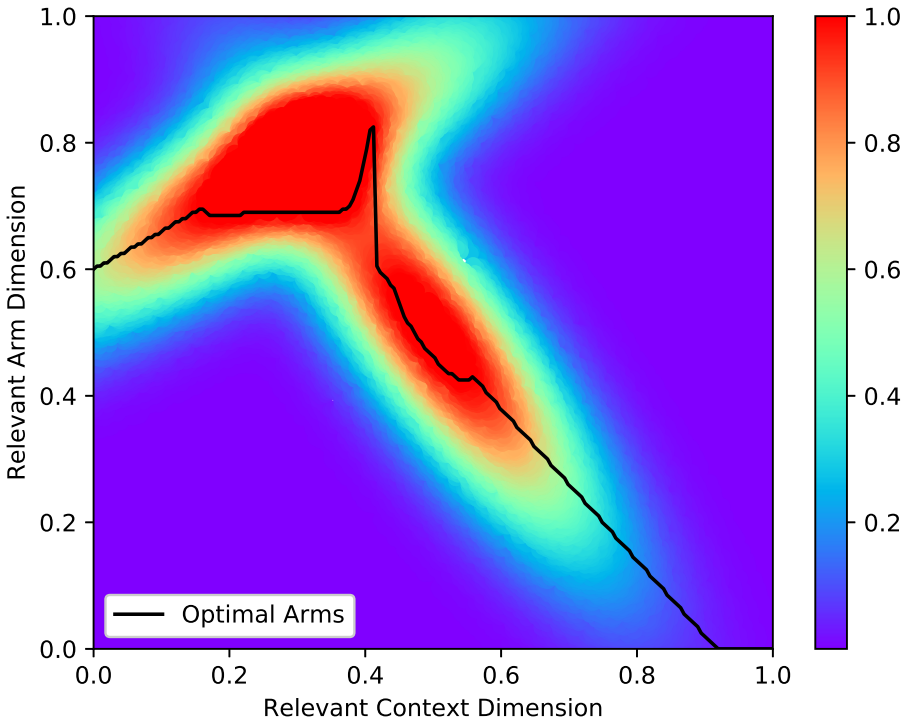


Figure 3.2: The expected reward defined over the relevant dimensions of the context-arm space

and

$$\Sigma_1 = \begin{bmatrix} 0.05 & 0.03 \\ 0.03 & 0.025 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 0.025 & -0.03 \\ -0.03 & 0.05 \end{bmatrix}.$$

The expected reward function defined on the relevant dimensions of the context-arm space can be found in Fig. 3.2, the black line shows the optimal arms for each context. In this setting, we consider $r(t) \sim \text{Bern}(\mu_{a(t)}(x(t)))$ and $r(t)$ are independent across rounds.

The algorithms are run for $T = 10^5$ rounds. The arrival of contexts are uniformly random and independent from other rounds. After grid search, the values 0.001 for CMAB-RL, 0.01 for IUP and 0.05 for C-HOO are found to achieve the highest cumulative rewards and thus the results that are reported are the results of the experiments using these values.

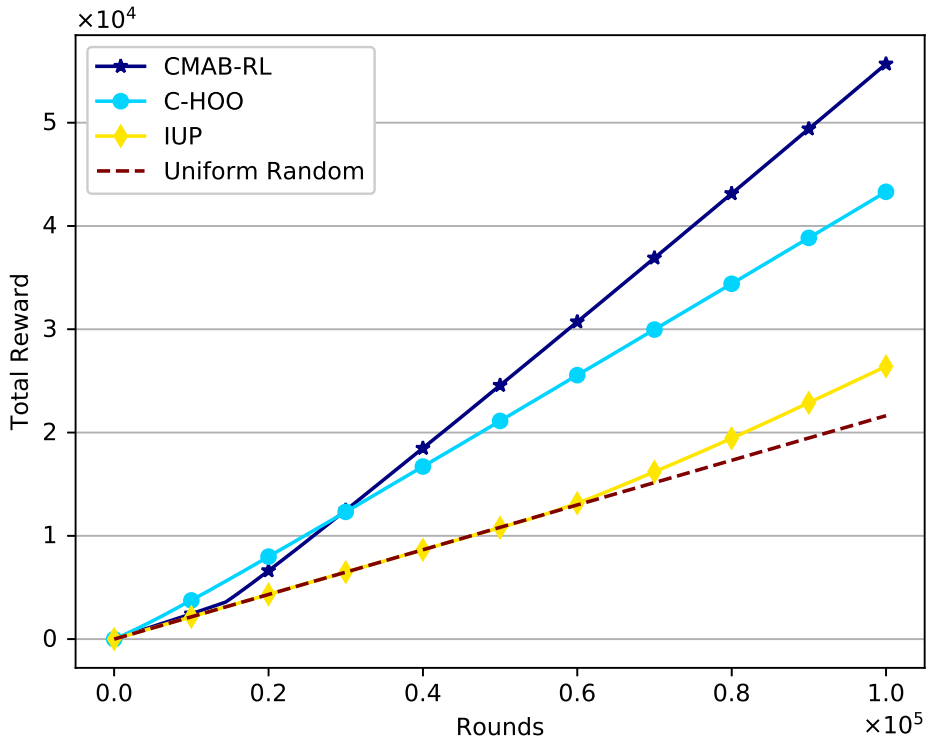


Figure 3.3: Comparison of cumulative rewards of CMAB-RL, C-HOO and IUP

The comparison of algorithms with respect to the cumulative rewards they achieve are given in Fig. 3.3. One can observe that CMAB-RL outperforms all its competitors by enjoying more than 29% and 100% improvement over the cumulative rewards of C-HOO and IUP respectively. Note that, even though C-HOO does not take relevance information into account, it still performs much better than IUP as it adaptively partitions the context-arm space. It can be seen that IUP performs only slightly better than Uniform Random algorithm, since the dimensionality of the problem is high and IUP need too many exploration rounds as a result.

We also compare the accumulated regrets of the algorithms, which are shown in Fig. 3.4. As contexts arrive uniformly random, CMAB-RL purely explores for about 15000 rounds, however after 15000 rounds, the regret accumulation rate drops significantly. Although the C-HOO performs better for the first 15000 rounds, as it does not utilize relevance information, it operates on a $d_x + d_a$

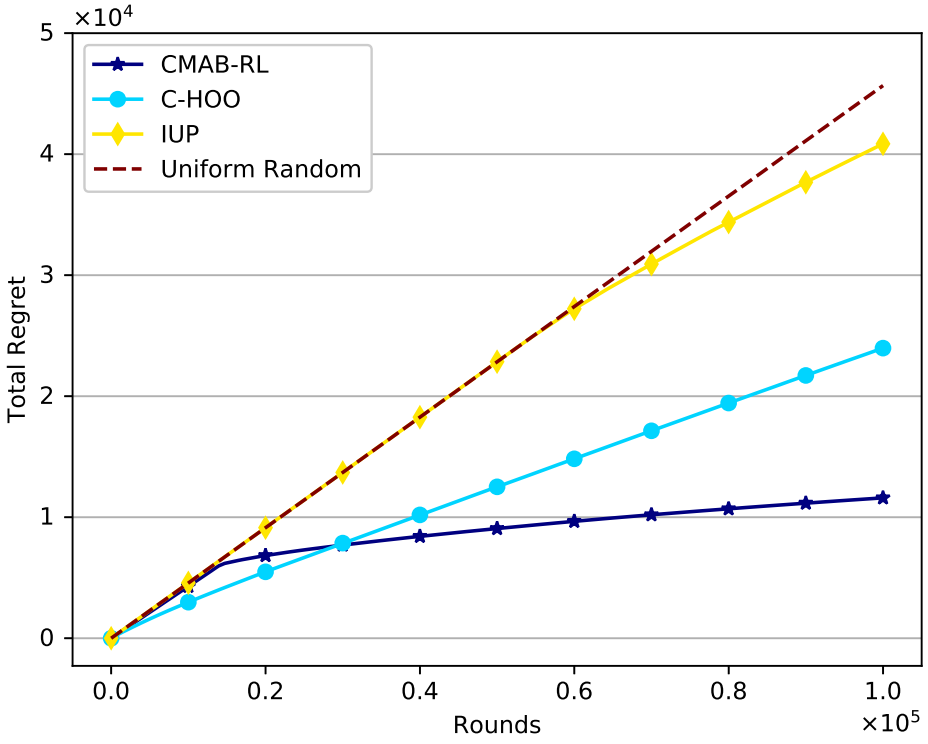


Figure 3.4: Comparison of regrets of CMAB-RL, C-HOO and IUP

dimensional space and fails to decrease regret due to dimensionality. Similarly IUP also suffers from curse of dimensionality and simply explores for too many rounds. We also investigate the effect of different time horizons on all algorithms by running them with different time horizons ranging from $T = 5000$ to $T = 10^5$. The result of this comparison can be seen in Fig. 3.5, where it is shown that CMAB-RL achieves the smallest regret for all time horizons. Note that since the synthetic environment is the same in all repetitions, the standard deviation of results are small compared to the values of cumulative rewards and cumulative regrets. The standard deviations of cumulative rewards and regrets over all repetitions, at the end of all rounds, for CMAB-RL are 255 and 184, respectively. The corresponding values for C-HOO are 224 and 191. Lastly, for IUP, we have 90 and 77.

At $T = 10^5$, CMAB-RL has a standard deviation of 255 and 184 for the final value of

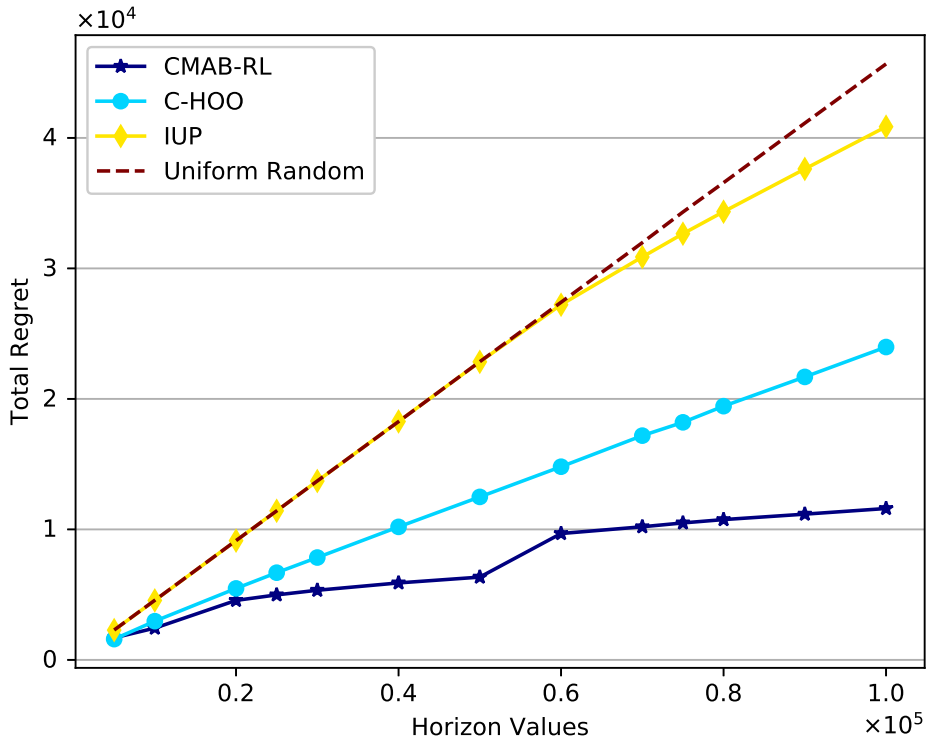


Figure 3.5: Comparison of regrets of CMAB-RL, C-HOO and IUP when they are run with different time horizons.

3.3.4 Experiments on the OhioT1DM dataset

In our second experiment, we use the OhioT1DM dataset which contains multiple physiological measurements taken from 6 T1DM patients is used. These patients were on continuous glucose monitoring and insulin pump therapy and the data that is taken is over a time interval of 8 weeks. [54] contains a detailed explanation for this dataset. As we are performing an online learning task, we merge the training and test sets which are pre-split in the dataset.

The purpose of this experiment is to learn the optimal bolus insulin dosage by utilizing the side(contextual) information such as the state of the patient and the ongoing basal insulin treatment. The optimal bolus insulin dosage is not a predetermined value, and we consider the whether a bolus treatment was proper or not by observing if the blood glucose levels of the patient remain within the

desired range $80 - 180$ mg/dL. This range is taken from [55].

We consider the following entries as the state of the patient: mean of continuous glucose measurements (CGMs), mean of heart rates, mean of skin temperatures, mean of air temperatures and mean of galvanic skin response measurements, sum of carbohydrate intake from meals, sums of exercise scores (multiplication of the duration and the intensity of an exercise session) and sums of number of steps taken for the last 30 minutes before a bolus injection. We take the mean of the basal insulin dosages again for the last 30 minutes and consider this as the basal insulin treatment. The entries listed above correspond to the context dimensions hence $d_x = 9$. Since the only action we consider the determination of dosages of bolus insulin treatments we have $d_a = \underline{d}_a = 1$.

As mentioned the goodness of a bolus dosage is determined by the CGM levels after the injection. For simplicity and convenience we will refer to the CGM values we use as contexts as past CGMs and we will refer to the CGM levels after a bolus injection as resulting CGMs. The resulting CGMs are the mean values of the CGM recording of the patient for the following 30 minutes to 2 hours after a bolus injection.

The data contains missing values, next we explain how we dealt with this issue. When extracting structured data from raw data, we first locate the bolus events and extract other variables near(past values for contexts, future values for resulting CGMs) the bolus events. Hence bolus injection information always exists, we do not need to impute. If no data is available for either the past or the resulting CGMs we omit that bolus event. As for other variables we impute by setting carbohydrate intakes, exercises and number of steps to zero since lack of data implies no activity. For heart rates, skin and air temperatures and galvanic skin response variables we take average value over the whole dataset as the value zero does not make sense in a human body for these variables.

The simulation setup requires the data to be in range $[0, 1]$ for all context and arm dimensions, hence first data is scaled. Afterwards, a Gaussian distribution is used to model a patient's context data. In order to mimic the dataset

Table 3.1: Percentages of samples for all approaches and individuals

		559	563	570	575	588	591	Overall
<80 mg/dL	CMAB-RL	00.25	00.04	00.13	00.47	00.24	00.71	00.27
	C-HOO	00.30	00.05	00.14	00.58	00.32	00.89	00.34
	IUP	00.35	00.05	00.18	00.69	00.38	00.96	00.38
	Dataset	01.97	00.25	01.59	04.50	00.00	03.41	01.75
80-180 mg/dL	CMAB-RL	66.92	70.49	66.64	88.65	69.06	79.31	72.78
	C-HOO	53.57	62.54	54.98	81.54	56.73	67.27	62.30
	IUP	50.10	56.08	51.76	77.11	52.39	64.25	57.99
	Dataset	36.84	56.78	37.93	62.00	39.81	57.95	49.06
>180 mg/dL	CMAB-RL	32.83	29.47	33.23	10.88	30.70	19.98	26.95
	C-HOO	46.12	37.41	44.88	17.87	42.95	31.84	37.36
	IUP	49.56	43.87	48.06	22.20	47.23	34.79	41.63
	Dataset	61.18	42.96	60.48	33.50	60.19	38.64	49.19

better, we also learn a prior distribution over the patients by observing their frequency of occurrence in the dataset. Since dataset only contains a finite amount of context-arm-reward instances, and as in our CMAB setting both the contexts and the arms are infinitely many, we also need to have a regression model that maps contexts-arm pairs to rewards. During run-time the model needs to return a reward for context-arm pairs that are not in the dataset. We use a Gradient Boosting regression model which has 100 decision trees with maximum level constraint of 5 as simple estimators and we use Huber loss in the model. As required, the inputs to the regression model consists of contexts and arms, and the outputs are the resulting CGMs. Before training the Gradient Boosting model, we employ oversampling(sampling with replacement) to have equal amounts of data for all patients.

In each round t, the prior distribution over the patients is used to select a patient. Next, the context vector $x(t)$ is sampled from the said patient's Gaussian distribution repeatedly, until a context vector that resides in $[0,1]^{(d_x+d_a)}$ is sampled. The context vector is then revealed to the CMAB algorithm, which in succession determines an arm $a(t)$. The bandit environment is then queried to generate the reward $r(t)$ according to $x(t)$ and $a(t)$. Bandit environment uses the regression model to generate the resulting CGM value and translates it into $r(t)$

by using the following mapping,

$$f(x) = \begin{cases} 0, & x \leq 80 \ \text{ (hypoglycemia)} \\ \frac{x-80}{10}, & 80 \leq x \leq 90 \\ 1, & 90 \leq x \leq 130 \\ \frac{180-x}{50}, & 130 \leq x \leq 180 \\ 0, & 180 \leq x \ \text{ (hyperglycemia)} \end{cases}$$

In order to add randomness to the reward generation process, a noise component with distribution $\mathcal{N}(0, 25)$ is added to the resulting CGM value.

In order to determine $\underline{d}_x$, we examine the average impurity decrease for each context dimension considering all trees which are then normalized so that the sum of the average impurities for all inputs add up to 1. The only context dimension that has a score higher than 0.5 is the past CGMs while the rest of the context dimensions yield scores lower than 0.1. [47] and [48] use this dataset for the forecasting problem and our result is consistent with theirs. Hence, it is rational to claim that the past CGM value affect the reward much more when compared with other variables in the dataset. Therefore we set $\underline{d}_x = 1$. Similar to the synthetic case, the horizon is set as $T = 10^5$, and for this experiment, the optimal confidence term multipliers found after grid search are given as 0.001 for CMAB-RL, 0.05 for IUP and 0.1 for C-HOO.

Since measuring the performance via cumulative rewards may not be meaningful for this setting, we directly examine the resulting CGMs values. Fig. 3.6 contains joint histograms for all patients, one for each algorithm including the original dataset as a benchmark. The histograms for each algorithm are normalized such that the area under the histograms sum up to 1, the reason is that since original dataset has much less instances when non-normalized histograms are plotted, the distribution for original dataset cannot be seen. Interestingly, all algorithms provide better administration of bolus dosage when compared with the method used when the data is gathered. To be able to investigate the effect of each algorithm on each patient, we also include the Table 3.1 where the percentages of samples are shown for each algorithm, for each patient and for each

blood glucose level. The header row indicates the patient identification numbers as encountered in the dataset. It can be observed that, CMAB-RL not only has the highest density in the desired range for all patients, it also has the lowest density in the hypoglycemia and hyperglycemia regions. The only exception to this for patient 588, where no hypoglycemic resulting CGMs are observed in the dataset.

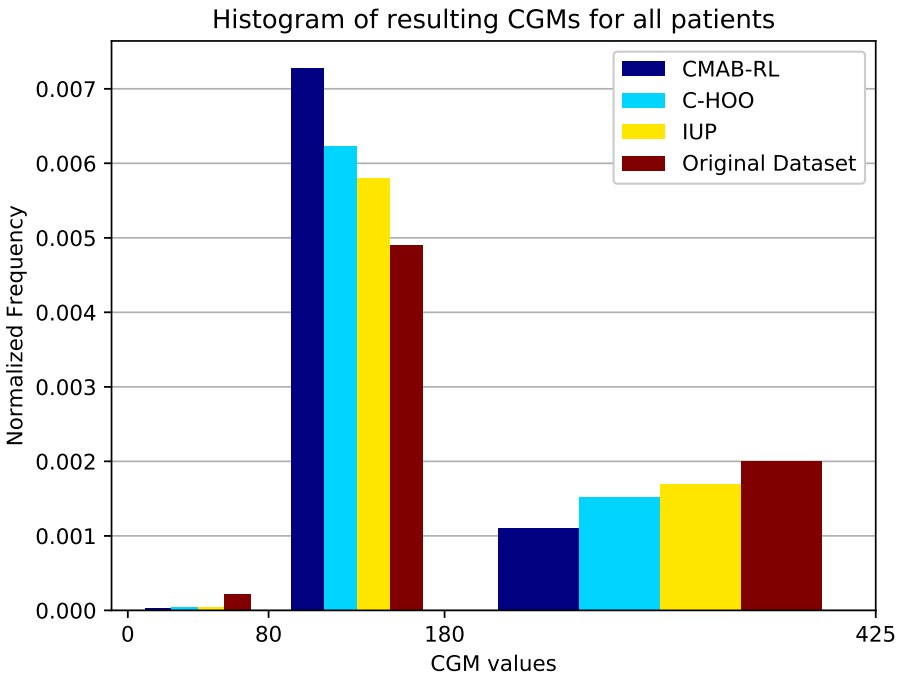


Figure 3.6: Joint Histograms of the resulting CGMs for all patients under different learning algorithms and the original dataset.

Chapter 4

Contextual Gaussian Process Bandit Problem with Relevance Learning

In this chapter, we study the CMAB problem with relevance learning in Bayesian setting. We do not require upper bounds on the number of relevant context and arm dimensions. In Section 4.1 we define the Bayesian CMAB setting with irrelevant context and arm dimensions, moreover we state our assumptions about this setting. We propose the CGP-UCB-RL algorithm in Section 4.2. We conclude this chapter with Section 4.3, in which we evaluate the performance of CGP-UCB-RL a numerically in a synthetic environment and in an experiment regarding the administration of bolus insulin for T1DM patients.

4.1 Problem Definition

The contextual multi-armed bandit(CMAB) problem involves sequential decision making in discrete rounds indexed by $t \in \{1, \ldots, T\}$. In each round, the learner observes the context vector $x(t) \in \mathcal{X}$, and selects an arm $a(t) \in \mathcal{A}$ by utilizing the

information of the current context and past observations. Once $a(t)$ is selected then a reward $r(t) = \mu(x(t), a(t)) + \epsilon(t)$ is observed, where $\mu(x, a)$ denotes the expected reward of a context-arm pair and $\epsilon(t) \sim \mathcal{N}(0, \sigma^2)$ is the random noise which is i.i.d across rounds and context-arm pairs.

The performance of a learner is measured by a metric called contextual regret. The instantaneous regret, regret for a fixed round t, is defined as $\Delta(x(t), a(t)) \coloneqq \mu(x(t), a^*(x(t))) - \mu(x(t), a(t))$ where $a^*(x) \coloneqq \arg\max_{a \in \mathcal{A}} \mu(x, a)$. The cumulative contextual regret at the end of T rounds is then given by $\text{Reg}(T) = \sum_{t=1}^{T} \Delta(x(t), a(t))$.

Let $d_{\mathcal{X}}$ and $d_{\mathcal{A}}$ denote the dimensionality of the context and arm spaces, $\mathcal{X}$ and $\mathcal{A}$, respectively. Also let $\mathcal{D}_{\mathcal{X}} = \{1, \ldots, d_{\mathcal{X}}\}$ and $\mathcal{D}_{\mathcal{A}} = \{1, \ldots, d_{\mathcal{A}}\}$ denote the set of dimensions for context and arm spaces, respectively. For any $q \subseteq \mathcal{D}_{\mathcal{A}}$, let the $|q|$-dimensional subspace of $\mathcal{A}$ be denoted by $\mathcal{A}_q$. Equivalently, for any $q \subseteq \mathcal{D}_{\mathcal{X}}$, let the $|q|$-dimensional subspace of $\mathcal{X}$ be denoted by $\mathcal{X}_q$. We denote the elements of $\mathcal{A}_q$ and $\mathcal{X}_q$ by adding the q subscript to the vectors a and x, i.e. $a_q \in \mathcal{A}_q$ and $x_q \in \mathcal{X}_q$.

We assume that the expected reward function, $\mu(x, a)$, is sampled from a Gaussian process where the kernel function is such that some dimensions of $\mathcal{D}_{\mathcal{X}}$ and $\mathcal{D}_{\mathcal{A}}$ do not affect the expected reward function and the mean is 0. In order to formally define this setting first we need some notation. Let $\wp(\mathcal{S})$ denote the set of all subsets of a set $\mathcal{S}$ excluding the empty set. Also let $\alpha_i^s = \mathbf{I}[i \in s]$ for $s \in \wp(\mathcal{S})$ and $i \in \mathbb{N}$, where $\mathbf{I}[\cdot]$ denotes the indicator function[1]. Consider the squared exponential kernel that inputs two context-arm pairs (x, a), (x', a') and returns a similarity measure, as given below:

$$k^{(v,w)}((x, a), (x', a')) \coloneqq \exp\left(-0.5 \sum_{i=1}^{d_{\mathcal{X}}} \alpha_i^v (x_i - x_i')^2\right) \exp\left(-0.5 \sum_{j=1}^{d_{\mathcal{A}}} \alpha_j^w (a_j - a_j')^2\right)$$

where (v, w) denote the context-arm dimensions tuple pair for $v \in \wp(\mathcal{D}_{\mathcal{X}})$ and

[1] α_i^s are used as inverse of parameters commonly referred to as length-scale in literature, which are utilized in automatic relevance determination(see [15] for details). Although length-scale parameters generally appear as denominators for each dimension, in our case it is more convenient to consider them as multipliers since we desire to completely ignore distances in some dimensions, requiring the multiplier to be 0.

$w \in \wp(\mathcal{D}_{\mathcal{A}})$. The choice of squared exponential is purely for simplicity purposes, in fact any kernel that enables a similar usage of weights per dimensions can be utilized. For any $v \in \wp(\mathcal{D}_{\mathcal{X}})$ and $w \in \wp(\mathcal{D}_{\mathcal{A}})$, let $GP_{(v,w)}$ be a Gaussian process such that $GP_{(v,w)} = GP(0, k^{(v,w)}((x,a),(x',a'))$. Let $\mathcal{C}_{\mathcal{X}}$ and $\mathcal{C}_{\mathcal{A}}$ denote the set of relevant context and arm dimensions respectively where $1 \leq |\mathcal{C}_{\mathcal{X}}| \leq d_{\mathcal{X}}$ and $1 \leq |\mathcal{C}_{\mathcal{A}}| \leq d_{\mathcal{A}}$. We assume that $\mu(x,a) \sim GP_{(\mathcal{C}_{\mathcal{X}},\mathcal{C}_{\mathcal{A}})}$. In Fig. 4.1, a sample from a Gaussian process with 4 dimensions, with the first 2 being relevant, is given where the expected reward function is viewed from all possible dimension combinations to observe the effect of such a kernel on the reward surface. Note that when two dimensions are plotted, values of the elements in other dimensions are taken to be zero, and due to memory constraints, first a sparse discretization of the function is realized then it is interpolated using linear interpolation.

For any noisy sample $Z_n = \{z(1), \ldots, z(n)\} \coloneqq \{(x(1), a(1)), \ldots, (x(n), a(n))\}$, $R_n = [r(1), \ldots, r(n)]^{\mathrm{T}}$, we define mean and variance estimations for point $(x,a) \in \mathcal{X} \times \mathcal{A}$, according to a context-arm dimension tuple pair (v,w) as,

$$\hat{\mu}_n^{(v,w)}(x,a) \coloneqq k_n^{(v,w)}(x,a)^{\mathrm{T}}(K_n^{(v,w)} + \sigma^2 I_{n\times n})^{-1} R_n$$
$$\hat{\sigma}_n^{(v,w)}(x,a) \coloneqq k^{(v,w)}((x,a),(x,a)) - k_n^{(v,w)}(x,a)^{\mathrm{T}}(K_n^{(v,w)} + \sigma^2 I_{n\times n})^{-1} k_n^{(v,w)}(x,a)$$

and when $(v,w) = (\mathcal{C}_{\mathcal{X}}, \mathcal{C}_{\mathcal{A}})$, then given estimations are parameters of the posterior distribution of $\mu(x,a)$. Here

$$k_n^{(v,w)}(x,a) \coloneqq [k^{(v,w)}((x,a),z(1)), \ldots, k^{(v,w)}((x,a),z(n))]^{\mathrm{T}}$$

and $K_n^{(v,w)} \coloneqq \{k^{(v,w)}(z,z')\}_{z,z' \in Z_n}$ is positive definite kernel matrix.

4.2 Contextual Gaussian Process Upper Confidence Bound Algorithm with Relevance Learning (CGP-UCB-RL)

The CGP-UCB-RL algorithm is described in algorithm 3. CGP-UCB-RL algorithm is an extension of CGP-UCB algorithm described in [14]. Unlike in [14],

in our approach we aim to exploit the existence of possible irrelevant dimensions that do not affect the reward distributions. The relevance learning is done via minimizing the negative log marginal likelihood according to different assumptions on which dimensions are relevant.

Algorithm 3 CGP-UCB-RL

Input: $\mathcal{X}$, $\mathcal{A}$, γ_t, τ
Initialization: Observe $x(1)$, select $a(1)$ arbitrarily, observe $r(1)$, set $Z_1 = \{(x(1), a(1)\}$ and $R_1 = [r(1)]^{\mathrm{T}}$
for $t = 2, 3, \ldots$ **do**
  Observe $x(t)$
  if $rem(t, \tau) = 0$ **then**
    Calculate $NLL^{(v,w)}(t)$ for all $v \in \wp(\mathcal{D}_{\mathcal{X}})$ and $w \in \wp(\mathcal{D}_{\mathcal{A}})$ according to 4.1
    Determine $(v(t), w(t))$ according to 4.2
  else
    Set $(v(t), w(t)) = (v(t-1), w(t-1))$
  end if
  Determine estimated optimal arm subspace $\mathcal{P}^{(v(t),w(t))}(t)$ according to 4.3
  Select arm $a(t)$ according to 4.4 (ties broken arbitrarily)
  Observe $r(t)$ and perform update on mean and variance estimations for all $v \in \wp(\mathcal{D}_{\mathcal{X}})$ and $w \in \wp(\mathcal{D}_{\mathcal{A}})$
end for

Let $rem(a, b)$ denote the remainder when a is divided by b, for $a, b \in \mathbb{N}$. If $rem(t, \tau) = 0$, CGP-UCB-RL calculates the negative log marginal likelihood(NLL) for all pairs of (v, w) where $v \in \wp(\mathcal{D}_{\mathcal{X}})$ and $w \in \wp(\mathcal{D}_{\mathcal{A}})$ and selects the (v, w) pair that minimizes the NLL, otherwise (v, w) pairs selected for the previous round is used. Although one can set $\tau = 1$ and perform optimization in every round, we include this option as one needs to calculate $(2^{d_{\mathcal{X}}} - 1)(2^{d_{\mathcal{A}}} - 1)$ NLL values and considering the matrix inverse operation this step may be desired to repeat every τ steps due to time constraints. In round t, for a pair (v, w), negative log marginal likelihood is given by,

$$NLL^{(v,w)}(t) = 0.5 \log \left(\left| K_{t-1}^{(v,w)} + \sigma^2 I_{(t-1)\times(t-1)} \right| \right) + 0.5 R_{t-1}^{\mathrm{T}} \left(K_{t-1}^{(v,w)} + \sigma^2 I_{(t-1)\times(t-1)} \right)^{-1} R_{t-1} + 0.5(t-1)\log(2\pi) \tag{4.1}$$

The (v, w) pair that minimizes the NLL is given as

$$(v(t), w(t)) \coloneqq \underset{\substack{v \in \wp(\mathcal{D}_{\mathcal{X}}) \\ w \in \wp(\mathcal{D}_{\mathcal{A}})}}{\arg\min} NLL^{(v,w)}(t) \tag{4.2}$$

Once the $(v(t), w(t))$ pair is determined, CGP-UCB-RL use this pair when calculating the estimation of the mean rewards and variances. Let upper confidence bound of a point (x, a), with respect to pair (v, w) be $U_t^{(v,w)}(x, a) \coloneqq \hat{\mu}_{t-1}^{(v,w)}(x, a) + \sqrt{\gamma_t \hat{\sigma}_{t-1}^{(v,w)}(x, a)}$ where γ_t is a function that can be optimized via prior knowledge about the problem instance or preliminary experiments on available data.

For round t, CGP-UCB-RL could then simply select any arm a that achieves largest $U_t^{(v(t),w(t))}(x(t), a)$, however notice that arms that are estimated to have the largest UCB actually form a $(d_{\mathcal{A}} - |w(t)|)$-dimensional hyperplane. This is because for any Z_n, R_n, σ^2, $x \in \mathcal{X}$, $v \in \wp(\mathcal{D}_{\mathcal{X}})$, $w \in \wp(\mathcal{D}_{\mathcal{A}})$ and $a, a' \in \mathcal{A}$ such that $a_w = a'_w$, we have $k_n^{(v,w)}(x, a) = k_n^{(v,w)}(x, a')$ and $k^{(v,w)}((x, a), (x, a')) = 1$. Hence, we have $\hat{\mu}_n^{(v,w)}(x, a) = \hat{\mu}_n^{(v,w)}(x, a')$ and $\hat{\sigma}_n^{(v,w)}(x, a) = \hat{\sigma}_n^{(v,w)}(x, a')$.

Instead of selecting any arm arbitrarily, CGP-UCB-RL aims to gather the maximum amount of information by selecting the arm that has the maximum variance among the set of arms that has the largest estimated UCB. Let

$$\mathcal{P}^{(v,w)}(t) \coloneqq \{a \in \mathcal{A} \mid U_t^{(v,w)}(x(t), a) = \max_{a' \in \mathcal{A}} U_t^{(v,w)}(x(t), a')\} \tag{4.3}$$

be the set of arms that have the largest estimated UCBs with respect to the pair (v, w). Then in round t, CGP-UCB-RL selects arm

$$a(t) = \underset{a' \in \mathcal{P}^{(v(t),w(t))}(t)}{\arg\max} \hat{\sigma}_t^{(\mathcal{D}_{\mathcal{X}}, \mathcal{D}_{\mathcal{A}})}(x(t), a'). \tag{4.4}$$

Finally, reward $r(t)$ is observed, the mean estimations $\hat{\mu}_t^{(v,w)}(x, a)$ and the variance estimations $\hat{\sigma}_t^{(v,w)}(x, a)$ are updated for all $v \in \wp(\mathcal{D}_{\mathcal{X}})$ and $w \in \wp(\mathcal{D}_{\mathcal{A}})$.

4.3 Illustrative Results

We evaluate the performance of CGP-UCB-RL by comparing it to the CGP-UCB approach in experiments with both synthetically generated data and data gathered from type 1 diabetes mellitus(T1DM) patients. CGP-UCB, proposed and discussed in detail in [14], assumes that the expected reward function $\mu(x, a)$ is sampled from a Gaussian process, however unlike in our setting, their work do not consider that one or more dimensions may be irrelevant. Thus, the kernel models they consider also do not allow different length-scale parameters to be learned for individual dimensions. During both experiments we assume that both algorithms do not have a priori information about the problem instance.

4.3.1 Competitor Learning Algorithms

We compare CGP-UCB-RL against CGP-UCB which does not take relevance information into account and an algorithm that makes uniformly random arm selections that is used as a benchmark.

4.3.1.1 Contextual Gaussian Process Bandit UCB Algorithm (CGP-UCB) [14]

The CGP-UCB algorithm is set in the Bayesian version of the CMAB problem which assumes that the expected reward function is sampled from a Gaussian process. In each round, CGP-UCB calculates statistics of posterior distribution over the expected reward function $\mu(x, a)$, which is again a Gaussian process distribution and applies UCB algorithm by identifying the arm with the highest UCB. The UCB is calculated using the posterior mean and standard deviation over $\mu(x, a)$.

4.3.1.2 Uniform Random

We include an algorithm that makes random arm selections without considering the context, past experiences or relevance information, in order to establish a benchmark in synthetic experiment setting.

4.3.2 Parameters Used in the Experiments

For both of our experiments, we used the square exponential kernel for both algorithms. As mentioned, we assume that both algorithms have no information available about the problem instance in advance. Hence, initial length-scale parameters are set to 1 for both algorithms as well as the noise variance σ^2. Also, both algorithms are given the context set $\mathcal{X}$ and the arm set $\mathcal{A}$. Additionally, for CGP-UCB, we set $\beta_t = 2\log\left(t^2 2\pi^2/(3\delta)\right) + 2(d_{\mathcal{X}} + d_{\mathcal{A}})\log\left(t^2(d_{\mathcal{X}} + d_{\mathcal{A}})br\sqrt{\log(4(d_{\mathcal{X}} + d_{\mathcal{A}})a/\delta)}\right)$ where we set $\delta = 0.01$(see [14])[2]. We use $r = 10$ in the synthetic experiment and in the T1DM experiment we set r to the largest value in any dimension in data after standardization. We also assume that $L = 1$ for the similarity assumption in Theorem 1 of [14] and then set $a = \delta e^{-1}$, $b = 1$ so that $ae^{-(L/b)^2} = \delta$. For CGP-UCB-RL, we set $\gamma_t = 1$(again, assuming we do no have a priori knowledge about the problem instance) and $\tau = 10$. Last but not least, although we set the values of β_t and γ_t as mentioned, during our studies we noticed that assigning smaller values tend to work better. In order to optimize these values, we scaled these values by a certain value selected from the set $\{0.01, 0.05, 0.1, 0.25, 0.5, 1\}$, using grid search. We report the results generated by the optimal scaling factor for each algorithm in both experiments. Also note that, reported performances are results acquired from 20 independent runs in an effort to minimize the effect of randomness which arises from context arrivals, arm selection and noisy rewards.

[2]The parameters of β_t are not extremely crucial as we optimize the uncertainty terms by scaling them specifically for each experiment

4.3.3 Experiments on a Synthetic Simulation Environment

In the synthetic setting, we consider an environment where the expected reward function is sampled from a Gaussian process prior where $d_x = 10$, $d_a = 2$, $|\mathcal{C}_\mathcal{X}| = 1$ and $|\mathcal{C}_\mathcal{A}| = 1$, with the relevant dimensions being the first dimensions for both the context and arm sets. Specifically, we have $\mu(x, a) \sim GP_{(\mathcal{C}_\mathcal{X}, \mathcal{C}_\mathcal{A})}$ where $\mathcal{C}_\mathcal{X} = \{1\}$ and $\mathcal{C}_\mathcal{A} = \{1\}$. We utilized squared exponential kernel as suggested in Section 4.1 to generate the environment with irrelevant context and arm dimensions. As expected reward function is randomly generated, it changes for each repetition of the experiment. However, to gain some perspective on the expected reward function instances, in Fig. 4.2 we provide several realizations of the reward surface with respect to the relevant context and arm dimensions. Reward $r(t)$ in round t, is generated using $\sigma^2 = 1$. Note that, here the relevant context dimension is the same for all arms.

The contexts are sampled independently from a uniform random distribution. Scale factor for β_t of CGP-UCB is set to 0.05, and scale factor for γ_t of CGP-UCB-RL is set to 0.5 after grid search over possible candidates. We set the horizon as $T = 100$ rounds for this setting.

We compare the algorithms by investigating their cumulative rewards and cumulative regrets. The plots of cumulative rewards and cumulative regrets as a function of time are given in Fig. 4.3 and in Fig. 4.4, respectively. It can be seen that CGP-UCB-RL achieves better performance by exploiting the fact that a total of 10 dimensions of the context-arm space do not affect the rewards, by learning to ignore them through the optimization of negative log marginal likelihood. For each round, we also provide 95% confidence intervals for cumulative rewards and regrets calculated over repetitions, shown in light-shaded regions. We provide this information because the synthetic environment is different for each repetition and hence the performance of the algorithms may vary due to the differences in the expected reward function. For the cumulative rewards, the confidence intervals we obtained were expected to be loose as in one repetition

the environment may be yielding higher rewards but in another repetition the rewards may be low. However, the confidence intervals for the cumulative regret were expected to be tighter as regret measures the difference between the optimal selection and the selection made by the learner. Thus, cumulative regret does not depend on whether the expected reward function yields high or low rewards. The confidence intervals we obtained for cumulative regret confirms this expectation. As both algorithms do not rely on T to optimize themselves, we do not include an experiment where the algorithms are run for different time horizons.

4.3.4 Experiments on the OhioT1DM dataset

We consider an experiment based on the OhioT1DM dataset, which is gathered from 6 patients whose various physiological traits and insulin treatments are monitored for 2 months(see [54] for a detailed review of the dataset). The dataset in vanilla version contains training and test sets separately, however we merge them as we are performing online learning.

We aim to use bandit algorithms to learn the optimal dosage of bolus insulin to regulate the blood glucose levels of a patient in the desire range, $80 - 180$ mg/dL given in [55]. A fixed bolus insulin dosage may be too high or too low to properly regulate the blood glucose levels depending on state of the patient and their current basal insulin treatment. In a CMAB setting, the state of the patient and the current basal treatment can be modeled as the contexts while the bolus insulin dosages can be modeled as the arms.

In our experiment we consider the state of the patient and basal insulin treatment for the last 30 minutes as contexts. Specifically, for the said 30 minutes the mean of continuous glucose measurements (CGMs), mean of heart rates, mean of skin temperatures, mean of air temperatures, mean of galvanic skin response measurements, sum of carbohydrate intake from meals, sums of exercise scores (multiplication of the duration and the intensity of an exercise session), sums of number of steps taken and the mean of the basal insulin dosages are extracted before a bolus injection event from the dataset as contexts, making a total of 9

Table 4.1: Percentages of samples for both approaches and all patients

		559	563	570	575	588	591	Overall
<80 mg/dL	CGP-UCB-RL	00.00	00.20	00.23	02.73	00.40	00.62	00.60
	CGP-UCB	00.44	00.79	00.45	03.12	00.00	02.16	01.10
	Dataset	01.97	00.25	01.59	04.50	00.00	03.41	01.75
80-180 mg/dL	CGP-UCB-RL	72.00	84.78	72.79	84.77	81.05	81.17	79.65
	CGP-UCB	68.44	77.47	66.89	79.30	71.77	68.52	72.20
	Dataset	36.84	56.78	37.93	62.00	39.81	57.95	49.06
>180 mg/dL	CGP-UCB-RL	28.00	15.02	26.98	12.50	18.55	18.21	19.75
	CGP-UCB	31.11	21.74	32.65	17.58	28.23	29.32	26.70
	Dataset	61.18	42.96	60.48	33.50	60.19	38.64	49.19

variables. We consider bolus injection dosage as the single arm variable.

The rewards are generated according to the CGM levels of the patient after a bolus injection, averaged over the following 30 minutes to 2 hours. We used the following mapping to translate CGMs to rewards

$$f(x) = \begin{cases} 0, & x \leq 80 \ \text{ (hypoglycemia)} \\ \frac{x-80}{40}, & 80 \leq x \leq 120 \\ \frac{180-x}{60}, & 120 \leq x \leq 180 \\ 0, & 180 \leq x \ \text{ (hyperglycemia)} \end{cases} \tag{4.5}$$

As data contains missing values, we need imputation to conduct experiments. While extracting data, first bolus events are located hence bolus injection events always exist. Then for missing contexts, we set the values of carbohydrate intake, exercise, basal insulin dosage and number of steps taken to 0, as no data implies no activity, but for heart rates, skin temperatures, air temperatures and galvanic skin responses we take the average over the dataset where imputing by 0 is not meaningful.

Before the experiments, we standardize the data so that each variable has zero mean and unit variance. Then, we construct a prior distribution over the patients by taking how many times they occur in the dataset into account. After this, we use a Gaussian distribution to model context data for each patient separately.

Since we assume that the expected reward function is sampled from a Gaussian process prior, we use Gaussian process regression to model the data(with contexts and arms as inputs and CGM levels after bolus injection as outputs) whose kernel is the squared exponential kernel and the length-scale parameters of the kernel are optimized during training. We use under sampling to create equal amounts of the data from each patient.

During run time, first a patient is randomly selected by sampling from the prior distribution. Afterwards, Gaussian distribution associated with the selected patient is used to sample the context vector. Then, the context vector is made available to the CMAB algorithm to observe which, in return, selects an arm. Finally, the CGMs are estimated according to the context and arms, a noise component with distribution $\mathcal{N}(0, 25)$ is added to the estimated CGM and the reward is generated according to function given in (4.5).

Since both algorithms are time consuming but converging fast, we run the experiments for $T = 100$ rounds. The optimal scale factors for CGP-UCB and CGP-UCB-RL are determined to be 0.01 and 0.05, respectively and the results are reported for these values. Examining the final CGMs instead of other statistics is a better way to analyze the results as it is hard to relate rewards or regret to the practical outcomes. In Fig. 4.5, the histograms for all approaches and the dataset are given, combining the results for all patients. Note that the histograms are normalized in order to eliminate the difference in sample sizes. We also give a more detailed view of the final distributions of the CGM levels after the injections in Table 4.1 where the header row contains the identification numbers of patients as they are given in the OhioT1DM dataset. CGP-UCB-RL outperforms CGP-UCB and achieves the largest density in the desired blood glucose level range over all patients. CGP-UCB-RL has less density in hypoglycemia and hyperglycemia ranges in general, except only for the hypoglycemia range for patient 588.

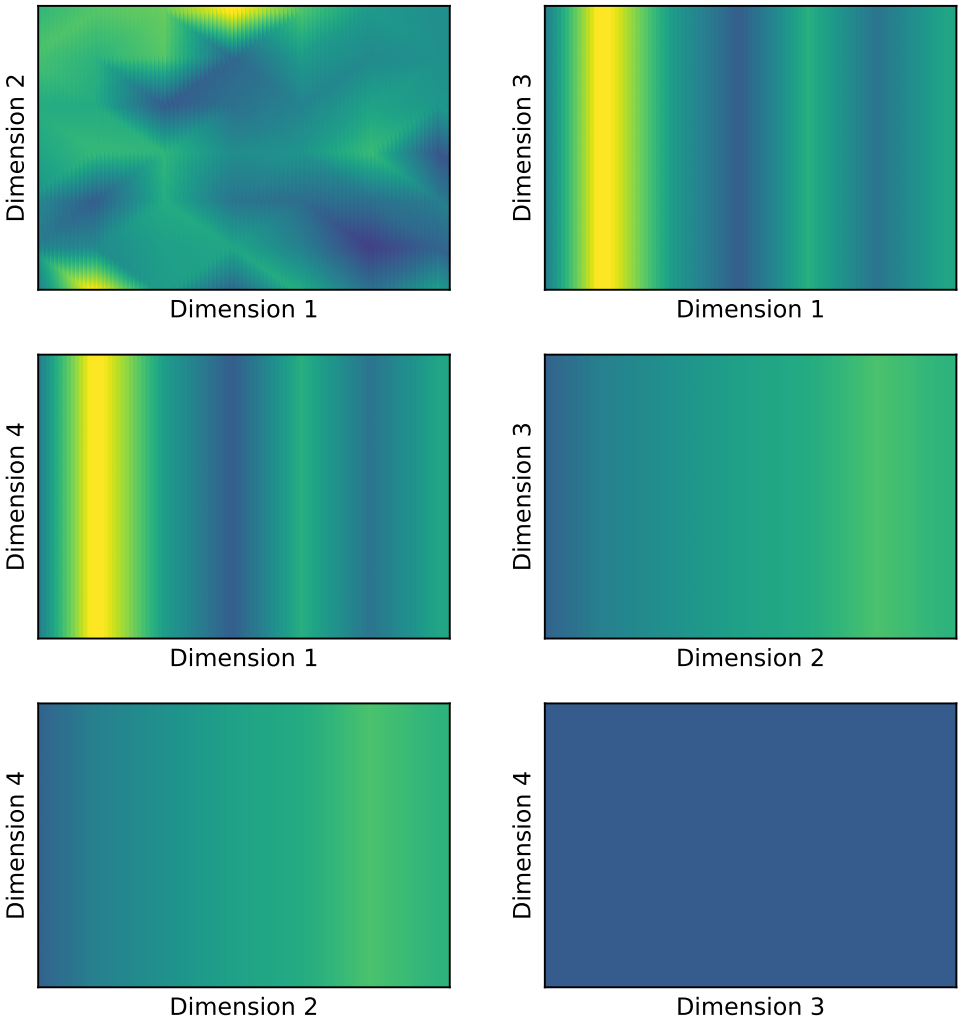


Figure 4.1: A sample of an expected reward function from a Gaussian process prior with 2 relevant and 2 irrelevant dimensions

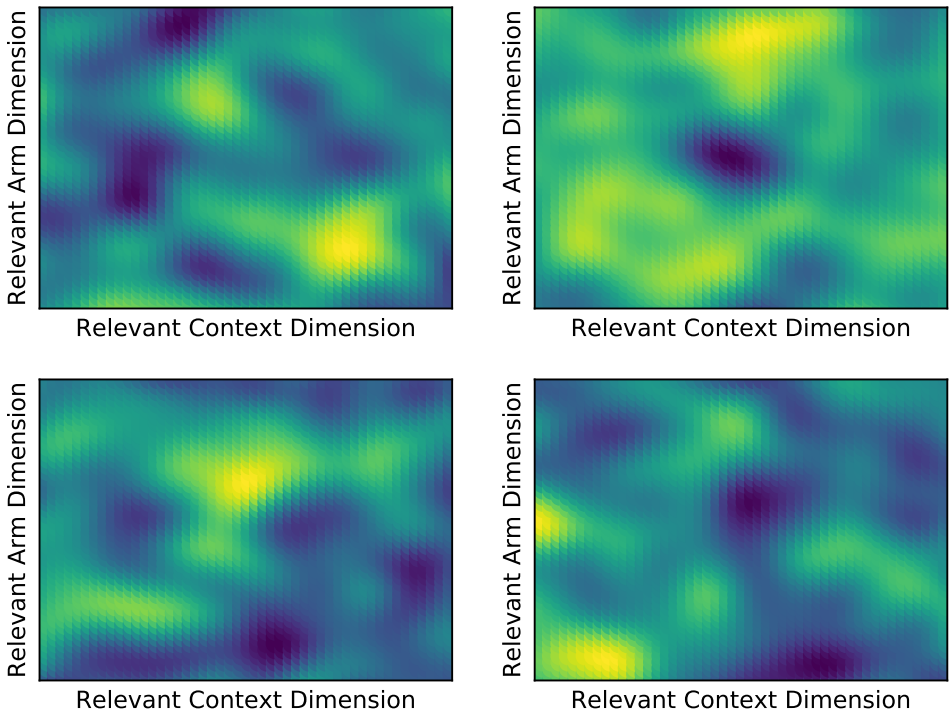


Figure 4.2: Four expected reward functions illustrated over the relevant dimensions of the context-arm space

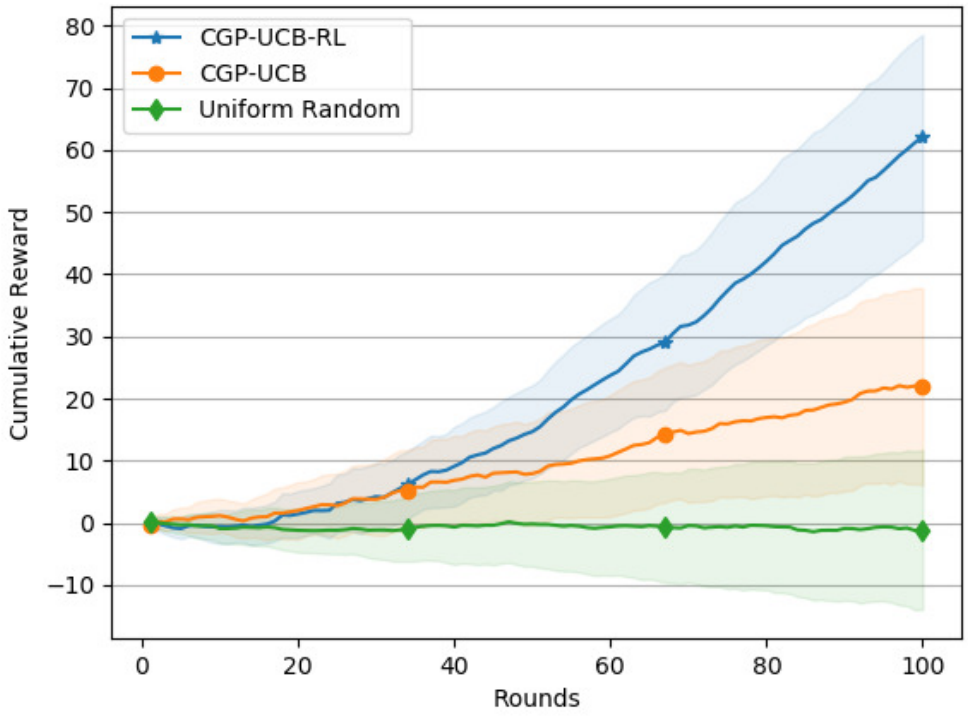


Figure 4.3: Comparison of cumulative rewards of CGP-UCB-RL, CGP-UCB and Uniform Random

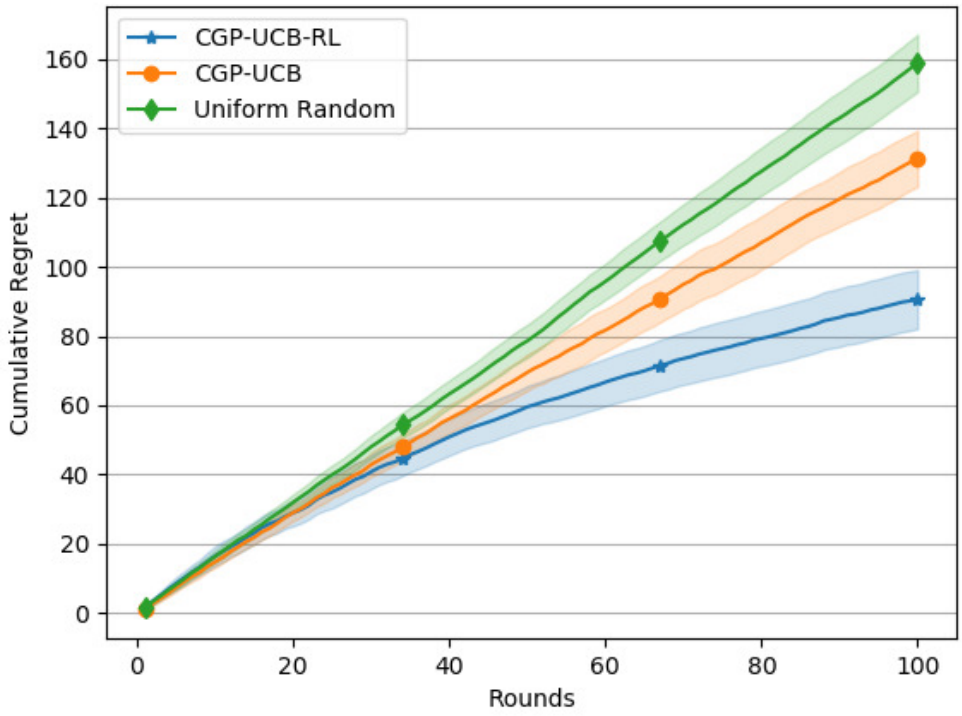


Figure 4.4: Comparison of regrets of CGP-UCB-RL, CGP-UCB and Uniform Random

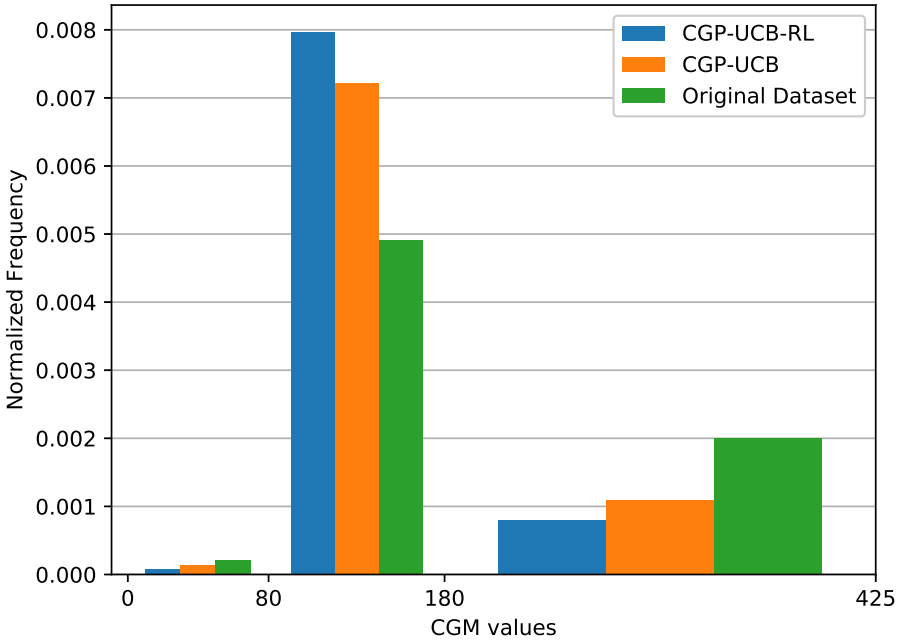


Figure 4.5: Histograms for CGP-UCB, CGP-UCB-RL and the dataset, combined for all patients

Chapter 5

Conclusion

In this thesis, we investigate the suitability of CMAB approaches to the blood glucose regulation problem in type 1 diabetes mellitus patients through the management of bolus insulin dosages. We make use of the OhioT1DM dataset, which is collected from diabetes patients, when setting up a realistic simulation scenario for the blood glucose regulation problem. When modeling the task as a CMAB problem, the contexts are considered to be the state of the patient, arms are considered to be different bolus insulin dosages and rewards are generated according to the blood glucose levels of patients after the injections. In both of the investigated problem settings, it is assumed that the expected reward function generated accordingly to the blood glucose levels is constant along a subset of dimensions of context and arm sets. Moreover, in the first problem, set of relevant patient traits are allowed to be different for different bolus insulin dosages and it is assumed that the expected reward function satisfies the Lipschitz continuity assumption. Upper bounds on the number of relevant context and arm dimensions are also assumed to be known in this setting. For this case, we use the CMAB-RL algorithm which includes a novel discretization strategy for the partitioning of context and arm sets. Empirically, we compare CMAB-RL against other CMAB approaches that do not take relevance into account in the setting of personalized treatment and show that CMAB-RL outperforms other approaches since irrelevant dimensions exist. CMAB-RL and other algorithms are also evaluated in a

synthetic environment where the expected reward function is generated using a Gaussian mixture model as a benchmark. In the second problem, we consider a Bayesian version of the CMAB problem, where the expected reward function is assumed to be a sample from a Gaussian process prior. We propose an extension of CGP-UCB, called CGP-UCB-RL, which learns the relevant context and arm dimensions by optimizing the negative log marginal likelihood of gathered samples. The performance of CGP-UCB-RL is numerically compared to CGP-UCB in two experimental settings and it is demonstrated that CGP-UCB-RL provides better performance when the number of relevant dimensions is small compared to the dimensionality of the context-arm space. The first experimental setting includes an environment sampled from Gaussian process prior and the second experiment involves the objective of this thesis, that is testing the algorithms in administration of bolus insulin to regulate the blood glucose levels of diabetes patients. For future work, we aim to provide theoretical guarantees for CGP-UCB-RL now that it is shown to perform well in practical settings. Furthermore, we aim to improve CMAB-RL and CGP-UCB-RL by considering adaptive partitioning strategies for relevance learning in CMAB problems. Moreover, we aim to extend this study to other personalized healthcare problems such as treatment of cancer and heart disease patients.

Bibliography

[1] B. B. Spear, M. Heath-Chiozzi, and J. Huff, "Clinical application of pharmacogenetics," *Trends in molecular medicine*, vol. 7, no. 5, pp. 201–204, 2001.

[2] L. Mangravite, C. Thorn, and R. Krauss, "Clinical implications of pharmacogenomics of statin treatment," *The pharmacogenomics journal*, vol. 6, no. 6, p. 360, 2006.

[3] D. J. Kaufman, R. Baker, L. C. Milner, S. Devaney, and K. L. Hudson, "A survey of us adults opinions about conduct of a nationwide precision medicine initiative® cohort study of genes and environment," *PLoS One*, vol. 11, no. 8, p. e0160461, 2016.

[4] A. of Medical Sciences, "Stratified, personalised or p4 medicine: a new direction for placing the patient at the centre of healthcare and health education," 2015.

[5] L. Li and K. Jamieson, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, pp. 1–52, 2018.

[6] Y. Gai, B. Krishnamachari, and R. Jain, "Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation," in *Proc. 4th IEEE Symposium on New Frontiers in Dynamic Spectrum*, pp. 1–9, 2010.

[7] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. on World Wide Web*, pp. 661–670, 2010.

[8] S. S. Villar, J. Bowden, and J. Wason, "Multi-armed bandit models for the optimal design of clinical trials: benefits and challenges," *Statist. sci.: a review J. Institute Math. Statist.*, vol. 30, no. 2, p. 199, 2015.

[9] C. Tekin, J. Yoon, and M. van der Schaar, "Adaptive ensemble learning with confidence bounds," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 888–903, 2017.

[10] T. Lu, D. Pál, and M. Pál, "Contextual multi-armed bandits," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, pp. 485–492, 2010.

[11] E. Turgay, C. Bulucu, and C. Tekin, "Exploiting relevance for online decision-making in high-dimensions," *arXiv preprint arXiv:1907.00783*, 2019.

[12] A. Slivkins, "Contextual bandits with similarity information," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2533–2568, 2014.

[13] E. Turgay, D. Oner, and C. Tekin, "Multi-objective contextual bandit problem with similarity information," in *Proc. 21st. Int. Conf. Artif. Intell. Statist.*, pp. 1673–1681, 2018.

[14] A. Krause and C. S. Ong, "Contextual gaussian process bandit optimization," in *Advances in Neural Information Processing Systems 24* (J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, eds.), pp. 2447–2455, Curran Associates, Inc., 2011.

[15] R. M. Neal, *Bayesian learning for neural networks*, vol. 118. Springer Sci. & Business Media, 2012.

[16] S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári, "X-armed bandits," *J. Mach. Learn. Res.*, vol. 12, pp. 1655–1695, 2011.

[17] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, pp. 235–256, 2002.

[18] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[19] S. Shekhar, T. Javidi, *et al.*, "Gaussian process bandits with adaptive discretization," *Electron. J. Stat.*, vol. 12, no. 2, pp. 3829–3874, 2018.

[20] J. Djolonga, A. Krause, and V. Cevher, "High-dimensional Gaussian process bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1025–1033, 2013.

[21] H. Tyagi, S. U. Stich, and B. Gärtner, "On two continuum armed bandit problems in high dimensions," *Theory Comput. Syst.*, vol. 58, no. 1, pp. 191–222, 2016.

[22] C. Tekin and M. van der Schaar, "RELEAF: An algorithm for learning and exploiting relevance," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 716–727, 2015.

[23] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, pp. 4–22, 1985.

[24] R. Agrawal, "Sample mean based index policies by o (log n) regret for the multi-armed bandit problem," *Adv. Appl. Prob.*, vol. 27, no. 4, pp. 1054–1078, 1995.

[25] A. Garivier and O. Cappé, "The KL-UCB algorithm for bounded stochastic bandits and beyond," in *Proc. 24th Annual Conf. on Learning Theory*, pp. 359–376, 2011.

[26] R. D. Kleinberg, "Nearly tight bounds for the continuum-armed bandit problem," in *Adv. Neural Inf. Process. Syst.*, pp. 697–704, 2005.

[27] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, pp. 208–214, 2011.

[28] M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini, "Finite-time analysis of kernelised contextual bandits," in *Proc. 29th Conf. Uncertainty Artif. Intell.*, pp. 654–663, 2013.

[29] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Adv. Neural Inf. Process. Syst.*, pp. 2312–2320, 2011.

[30] J. Langford and T. Zhang, "The Epoch-Greedy algorithm for contextual multi-armed bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, pp. 1096–1103, 2007.

[31] M. Dudik, D. Hsu, S. Kale, N. Karampatziakis, J. Langford, L. Reyzin, and T. Zhang, "Efficient optimal learning for contextual bandits," in *Proc. 27th Conf. Uncertainty Artif. Intell.*, pp. 169–178, 2011.

[32] A. Agarwal, D. Hsu, S. Kale, J. Langford, L. Li, and R. Schapire, "Taming the monster: A fast and simple algorithm for contextual bandits," in *Proc. 31st Int. Conf. Mach. Learn.*, pp. 1638–1646, 2014.

[33] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc.*, vol. 58, no. 1, pp. 267–288, 1996.

[34] H. Deng and G. Runger, "Feature selection via regularized trees," in *Proc. Int. Joint Conf. Neural Netw.,*, pp. 1–8, 2012.

[35] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, 1994.

[36] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, no. 1-3, pp. 389–422, 2002.

[37] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," in *Feature extraction, construction and selection*, pp. 117–136, Springer, 1998.

[38] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern recognition letters*, vol. 15, no. 11, pp. 1119–1125, 1994.

[39] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Conf. Mach. Learn.*, pp. 368–377, 1992.

[40] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, no. 8, pp. 1226–1238, 2005.

[41] S. Perkins and J. Theiler, "Online feature selection using grafting," in *Proc. 20th Int. Conf. Mach. Learn.*, pp. 592–599, 2003.

[42] J. Zhou, D. Foster, R. Stine, and L. Ungar, "Streaming feature selection using alpha-investing," in *Proc. 11th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 384–393, 2005.

[43] K. Yu, X. Wu, W. Ding, and J. Pei, "Towards scalable and accurate online feature selection for big data," in *Proc. 14th IEEE Int. Conf. Data Mining*, pp. 660–669, 2014.

[44] X. Wu, K. Yu, H. Wang, and W. Ding, "Online streaming feature selection," in *Proc. 27th Int. Conf. Mach. Learn.*, pp. 1159–1166, 2010.

[45] R. Féraud, R. Allesiardo, T. Urvoy, and F. Clérot, "Random forest for the contextual bandit problem," in *Proc. 19th Int. Conf. Artif. Intell. Statist.*, pp. 93–101, 2016.

[46] A. H. Beck, A. R. Sangoi, S. Leung, R. J. Marinelli, T. O. Nielsen, M. J. Van De Vijver, R. B. West, M. Van De Rijn, and D. Koller, "Systematic analysis of breast cancer morphology uncovers stromal features associated with survival," *Science translational medicine*, vol. 3, no. 108, pp. 108ra113–108ra113, 2011.

[47] T. Zhu, K. Li, P. Herrero, J. Chen, and P. Georgiou, "A deep learning algorithm for personalized blood glucose prediction," in *Proc. 3rd Int. Workshop Knowledge Discovery Healthcare Data*, pp. 74–78, 2018.

[48] C. Midroni, P. Leimbigler, G. Baruah, M. Kolla, A. Whitehead, and Y. Fossat, "Predicting glycemia in type 1 diabetes patients: Experiments with XGBoost," in *Proc. 3rd Int. Workshop Knowledge Discovery Healthcare Data*, pp. 79–84, 2018.

[49] M. Kukar, I. Kononenko, C. Grošelj, K. Kralj, and J. Fettich, "Analysing and improving the diagnosis of ischaemic heart disease with machine learning," *Artificial intelligence in medicine*, vol. 16, no. 1, pp. 25–50, 1999.

[50] P. C. Austin, J. V. Tu, J. E. Ho, D. Levy, and D. S. Lee, "Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes," *Journal of clinical epidemiology*, vol. 66, no. 4, pp. 398–407, 2013.

[51] P. Zhang, F. Wang, J. Hu, and R. Sorrentino, "Towards personalized medicine: leveraging patient similarity and drug similarity analytics," *AMIA Summits on Translational Science Proceedings*, vol. 2014, p. 132, 2014.

[52] J. Yoon, C. Davtyan, and M. van der Schaar, "Discovery and clinical decision support for personalized healthcare," *IEEE J. Biomedical and Health Informatics*, vol. 21, no. 4, pp. 1133–1145, 2016.

[53] A. M. Alaa, J. Yoon, S. Hu, and M. Van der Schaar, "Personalized risk scoring for critical care prognosis using mixtures of gaussian processes," *IEEE Trans. on Biomedical Engineering*, vol. 65, no. 1, pp. 207–218, 2017.

[54] C. Marling and R. Bunescu, "The OhioT1DM dataset for blood glucose level prediction," in *Proc. 3rd Int. Workshop Knowledge Discovery Healthcare Data*, 2018.

[55] "Standards of medical care in diabetes—2019 abridged for primary care providers," *Clin. Diabetes*, vol. 37, no. 1, pp. 11–34, 2019.