Thesis for The Degree of Doctor of Philosophy

Offline and Online Models for Learning Pairwise Relations in Data

Fazeleh Hoseini

Division of Data Science and AI
Department of Computer Science & Engineering
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden, 2023

Offline and Online Models for Learning Pairwise Relations in Data

Fazeleh Hoseini

ISBN 978-91-7905-820-3
Doktorsavhandlingar vid Chalmers tekniska högskola, Ny serie nr 5286.
ISSN 0346-718X

Department of Computer Science & Engineering
Division of Data Science and AI
Chalmers University of Technology and Gothenburg University
Gothenburg, Sweden

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice,
Gothenburg, Sweden 2023.

To my parents for their endless love.
تقدیم به مادرم فرخنده و پدرم جعفر.

Abstract

Pairwise relations between data points are essential for numerous machine learning algorithms. Many representation learning methods consider pairwise relations to identify the latent features and patterns in the data. This thesis, investigates learning of pairwise relations from two different perspectives: offline learning and online learning.

The first part of the thesis focuses on offline learning by starting with an investigation of the performance modeling of a synchronization method in concurrent programming using a Markov chain whose state transition matrix models pairwise relations between involved cores in a computer process.

Then the thesis focuses on a particular pairwise distance measure, the minimax distance, and explores memory-efficient approaches to computing this distance by proposing a hierarchical representation of the data with a linear memory requirement with respect to the number of data points, from which the exact pairwise minimax distances can be derived in a memory-efficient manner. Then, a memory-efficient sampling method is proposed that follows the aforementioned hierarchical representation of the data and samples the data points in a way that the minimax distances between all data points are maximally preserved. Finally, the thesis proposes a practical non-parametric clustering of vehicle motion trajectories to annotate traffic scenarios based on transitive relations between trajectories in an embedded space.

The second part of the thesis takes an online learning perspective, and starts by presenting an online learning method for identifying bottlenecks in a road network by extracting the minimax path, where bottlenecks are considered as road segments with the highest cost, e.g., in the sense of travel time. Inspired by real-world road networks, the thesis assumes a stochastic traffic environment in which the road-specific probability distribution of travel time is unknown. Therefore, it needs to learn the parameters of the probability distribution through observations by modeling the bottleneck identification task as a combinatorial semi-bandit problem. The proposed approach takes into account the prior knowledge and follows a Bayesian approach to update the parameters. Moreover, it develops a combinatorial variant of Thompson Sampling and derives an upper bound for the corresponding Bayesian regret. Furthermore, the thesis proposes an approximate algorithm to address the respective computational intractability issue.

Finally, the thesis considers contextual information of road network segments by extending the proposed model to a contextual combinatorial semi-bandit framework and investigates and develops various algorithms for this contextual combinatorial setting.

Keywords

Representation Learning, Pairwise Relations, Minimax Distance, Online Learning, Mulit-Armed Bandit, Thompson Sampling, Bottleneck Identification, Memory Efficiency, Motion Trajectory Clustering, Concurrent Programming, Performance Modeling

Acknowledgment

I would like to start by expressing my deepest gratitude to my thesis advisor, Morteza Haghir Chehreghani, for his valuable guidance, flexibility towards all different case, and encouragement along the way. It has been an enjoyable experience with a lot of knowledge. I would also like to thank my co-supervisor Alexander Schliep and my examiner Peter Damaschke. I would also like to express my heartfelt appreciation to Agneta Nilsson and Tomas Olovsson for helping me on my PhD journey.

I am honored to have Prof. Bengt J. Nilsson as my faculty opponent for my PhD defense. I would also like to thank the members of the grading committee: Prof. Niklas Lavesson, Prof. Anne Håkansson, Prof. Paul Davidsson and Prof. Richard Johansson. I would also like to express my gratitude to the Chair of the defense session, Prof. Graham Kemp.

I am grateful that during these years as a doctoral student I have met and worked with many extraordinary people who have helped me in various ways: Mehrzad, Shirin, Nicklas, Arman, Firooz, Ivan, and Aras.

I am lucky to have met such nice colleagues and friends in Chalmers who made work a fun place for me. Many thanks to Adam, Adones, Amir, Anton, Ashkan, Babi, Bastian, Birgit, Carlo, Christopher, Christos, Dag, David, Debabrota, Devdatt, Dimitris, Divya, Elad, Elena, Emil, Emilio, Fatemeh, Francisco, Fredrik, Georgia, Hamid, Hampus, Hannah, Iosif, Joris, Juan, Karl, Katerina, Kolbjörn, Lena, Linus, Lovisa, Magnus, Marika, Markus, Marwa, Mehrdad, Mohammad, Nasser, Newton, Olaf, Oliver, Romaric, Selpi, Siavash, Simon, Thomas, Tobias, Valentin P., Valentin T. , Vladimir and Wissam. I would also like to thank the administrative staff, especially Clara, Marianne, Rebecca, Monica, Fatima and Eva.

I would like to express my deepest gratitude to my parents for their endless love and the countless sacrifices they made to ensure that I could follow the path I wanted. To my sister and brother who always support me and help me overcome challenges and difficulties. To my wonderful friends who provide me with pleasant distractions. And of course to my incredible partner Lucas for his love and belief in me.

This research has been funded by Swedish Research Council, project "Models and Techniques for Energy-Efficient Concurrent Data Access Designs" grant nr. 201605360, as well as faculty funds, IKB 37403230, KST 372300.

Fazeleh Hosini
Göteborg, Jan 2023

List of Publications

Appended publications

This thesis is based on the following publications:

[A] Fazeleh Hoseini, Aras Atalar, and Philippas Tsigas "Modeling the performance of atomic primitives on modern architectures"
Proceedings of the 48th International Conference on Parallel Processing (ICPP), 2019.

[B] Fazeleh Hoseini, Morteza Haghir Chehreghani "Memory-Efficient Minimax Distance Measures"
Proceedings of the 26-th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2022.

[C] Fazeleh Hoseini, Sadegh Rahrovani, Morteza Haghir Chehreghani "Vehicle Motion Trajectories Clustering via Embedding Transitive Relations"
Proceedings of the 24th IEEE International Conference on Intelligent Transportation Systems (ITSC), 2021.

[D] Niklas Åkerblom, Fazeleh Hoseini, Morteza Haghir Chehreghani "Online Learning of Network Bottlenecks via Minimax Paths"
Machine Learning (2023): 131-150.

[E] Fazeleh Hoseini, Niklas Åkerblom, Morteza Haghir Chehreghani " A Contextual Combinatorial Semi-Bandit Approach to Network Bottleneck Identification "
Submitted to Recommender Systems (RecSys), 2023 .

Contents

Part I

Thesis Overview

Chapter 1

Introduction

Depending on how the data representation is derived, the difficulty of different data processing tasks can vary greatly, making an appropriate data representation essential. Therefore, representation learning is the first step in many machine learning and data analysis processes with the goal of extracting and recognizing useful data representations and latent features.

Some representation learning algorithms rely on pairwise similarities or dissimilarities between data points. If the data points can be represented by a graph, these (dis)similarities can be expressed by link-based distances. Link-based distances are distance measures where all paths between a pair of nodes in a graph are considered to calculate the distance between the pair. Minimax distance measure is a link-based distance that allows us to extract elongated manifolds and structures in the data in an unsupervised manner.

This thesis studies pairwise relations from different perspectives: from the perspective of offline learning, where the data and their pairwise relations are known, and from the perspective of online learning, where the pairwise relations are stochastic and unknown and need to be learned. It also investigates and proposes different learning frameworks for real-world applications, especially in the domain of autonomous vehicles.

Numerous industries have undergone revolutionary changes as a result of digitalization. In the automotive industry, these forces are leading to four disruptive technology-driven megatrends that are mutually reinforcing: diverse mobility, autonomous driving, electrification and connectivity [1]. The future of the automotive industry depends on how successfully it can adapt to a market that has evolved and where consumer acceptance and willingness to pay is highly dependent on the implementation of sustainable, flexible, reliable and safe technical solutions that deliver a world-class user experience. The safety and navigation of autonomous vehicles are two topics explored in this thesis.

An efficient navigation or route planning system for autonomous vehicles aims to optimize a combination of factors (travel time, energy consumption, etc.). Bottleneck identification in a road network is a navigation problem where the bottleneck on a path between a source node and a destination node in a network is defined as the road segment with the maximum cost. The goal of identifying and avoiding bottlenecks is then to find a path whose bottleneck is minimal. Thus, one can model bottleneck identification as the problem of

computing the minimax edge over the given network/graph to obtain an edge with the minimum largest gap between the source and destination nodes.

In the last decade, several studies have investigated the navigation problem networks in different applications. However, most existing methods assume that the necessary information to compute the optimal path is given. This assumption may not be true in many situations where the parameters of the models are unknown and stochastic. Therefore, the algorithm must learn the parameters of the model while solving the navigation problem. Using Bayesian methods to model these parameters allows us to use prior knowledge to update and learn the model parameters.

The contributions of this thesis, which are carried by five papers, are as follows:

- **Paper A** investigates a synchronization method on multiprocessors and modeling its performance by considering the core pairwise relations.
- **Paper B** proposes two memory-efficient approaches to compute minimax distances that requires a linear memory with respect to the size of the data points.
- **Paper C** develops an unsupervised framework to cluster vehicle motion trajectories. This framework is based on minimax distances and does not require the specification of hyper-parameters and can be used as a validation tool to assess the quality of synthetic trajectories.
- **Paper D** develops a combinatorial semi-bandit framework for learning minimax distances (bottleneck) over stochastic networks, including a combinatorial version of Thompson Sampling. Since the problem is computationally intractable, the paper proposes an alternative problem formulation that approximates the original objective. It also establishes an upper bound for the corresponding Bayesian regret.
- **Paper E** proposes a unified online learning framework based on contextual combinatorial semi-bandits that enables bottleneck identification in addition to learning the specifications of the underlying network. This framework adapts to several combinatorial semi-bandit algorithms. It also proposes an extension of the neural network approach for combinatorial bandits.

The rest of the thesis is organized as follows. Chapter 2 provides background information on pairwise relations, online learning and concurrent programming. Chapter 3 describes the research challenges addressed in each paper and the contributions made to address these challenges. Finally, Chapter 4 concludes the thesis and provides an outlook on future work.

Chapter 2

Background

The following chapter presents some of the topics and concepts used throughout this thesis.

2.1 Representation Learning

The first phase in many machine learning and data analysis tasks is to apply one or more representation learning methods to the input data points to find meaningful data representations and discover hidden patterns. The success of this step can greatly affect the performance of data processing tasks. One category of representation learning methods focuses on a pairwise relation of data points, such as kernel methods, PCA, LDA, etc. A powerful structure for representing relational data is a graph, such as information networks, social networks, and biological networks, where the links between data points are represented by the edges of the graph. Link-based distances can be defined between data points represented by a weighted graph where nodes are associated with data points and edges indicate the similarity/dissimilarity of data point pairs.

Link-based distances are based on all possible paths between pairs of nodes. For example, shortest path distance is a link-based distance in which the similarity of a pair of nodes is derived from the shortest path between the pair. Minimax distance is another link-based distance determined by the minimum largest distance along all possible paths between a pair of data points and captures the transitive dissimilarity of data points. It is therefore well suited for separating data clustered in non-convex manifolds.

2.1.1 Minimax Distances

Minimax distance is a distance based on transitive connections, used in applications where the underlying structures and patterns in the data are better represented by extracting transitive relations, than by direct (dis)similarities. The work in [2] used minimax distances in path-based clustering applications, and [3]–[5] investigated minimax distances in K-nearest neighbor search.

Given a data set D with N data points, the first step in minimax distance computation is to construct a weighted graph where the nodes indicate the

data points and the edges indicate the direct (dis)similarities of a pair of nodes using a dissimilarity function, e.g., squared Euclidean distance. More formally: Given a data set $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_N\}$ where $\mathbf{d}_i$ is the feature vector of data point $i \in \mathbf{V}$ and $\mathbf{V} = \{1, 2, \ldots, N\}$ is the corresponding index set, $\mathbf{D}$ can be represented by a weighted graph $\mathcal{G}(\mathbf{V}, \mathbf{E}, f)$ where the nodes indicate the data points indices $\mathbf{V}$, and the edge weights are determined by a dissimilarity function f, where $\mathbf{E}_{ij} = f(\mathbf{d}_i, \mathbf{d}_j)$, for all $i, j \in \mathbf{V}$. The dissimilarity function f must satisfy semi-metric requirements (which for all $j, i \in \mathbf{V}$: i) $f(\mathbf{d}_i, \mathbf{d}_j) \geq 0$, ii) if $f(\mathbf{d}_i, \mathbf{d}_j) = 0$, then $i = j$, and iii) $f(\mathbf{d}_i, \mathbf{d}_j) = f(\mathbf{d}_j, \mathbf{d}_i)$) .

The minimax distance between a pair of nodes in graph $\mathcal{G}$, $i, j \in \mathbf{V}$ is the smallest largest edge among all possible paths between i and j and can be expressed as Equation 2.1, where $\mathbf{M}_{ij}$ indicates the minimax distance between i and j, for all $i, j \in \mathbf{V}$; and $P_{i,j}$ is the set of all possible paths between i and j in graph $\mathcal{G}$. A path $p \in P$, is characterized by a set of consecutive edges, each denoted by $(n, m) \in \mathbf{E}$ where $n, m \in \mathbf{V}$ and $n \neq m$ is a pair of vertices at two ends of edge $\mathbf{E}_{n,m}$ with a weight of $f(\mathbf{d}_n, \mathbf{d}_m)$.

$$\mathbf{M}_{ij} = \min_{p \in P_{i,j}} \max_{(n,m) \in p} \mathbf{E}_{n,m}, \tag{2.1}$$

The Floyd-Marshall algorithm [6] is the classic approach to find pairwise distances within a data set. It tracks all possible paths between each pair of nodes, so the complexity of the algorithm is $\mathcal{O}\left(N^3\right)$ for a data set with N data points. Studies in [7]–[9] show that the minimax distance of any pair of nodes in graph $\mathcal{G}$ is equal to the minimax distance of the same pair in the minimum spanning tree of $\mathcal{G}$. A minimum spanning tree (MST) of $\mathcal{G}$ is an acyclic subgraph of $\mathcal{G}$ where the sum of the edge weights is minimal and there is a unique path between arbitrary pairs of nodes. In a weighted graph with positive edges, an MST is not necessarily unique (if the edge weights are not identical), but the sum of edges of different MSTs over $\mathcal{G}$ is the same. Thus, an efficient approach to computing minimax distances is to first find the MSTs of graph $\mathcal{G}$. Prim's is an efficient algorithm for a dense graph with runtime complexity of $\mathcal{O}(|\mathbf{E}| + |\mathbf{V}| \log |\mathbf{V}|)$, and Kruskal algorithm is the preferred algorithm for a sparse graph with runtime of $\mathcal{O}(|\mathbf{E}| \log |\mathbf{V}|)$. [9] proposed an efficient algorithm to compute the minimax distance with a runtime of $\mathcal{O}(N^2)$.

2.1.2 Minimax Distances in Application

Minimax distances provide an efficient way to determine the transitive-aware relation of data points. This thesis investigates this property of minimax distances in two different applications: i) the use of minimax distances in vehicle motion trajectories, and ii) the use of minimax distances in identifying bottlenecks in road networks. This subsection provides some background information on these two applications.

Vehicle motion trajectories

Autonomous vehicles are often considered as an important direction for the future of transportation, where consumer acceptance and willingness to pay is highly dependent on the development of safe and reliable technical solutions that can deliver a satisfactory user experience. With the increasing complexity

of Autonomous Driving (AD), field operational testing is no longer sufficient to ensure their safety. Therefore, the development of data-driven approaches in AD safety becomes one of the most challenging tasks in the area of AD.

Nowadays, various sensors are installed in vehicles, such as LiDAR and cameras. These sensors provide a huge amount of raw data that needs to be processed to be useful for the task at hand, such as a scenario database for scenario-based verification tasks. Depending on the use case, different approaches [10] can be used to assign the data collected by the sensors to specific scenarios: Rule-based approaches are used to find standard scenario classes where prior knowledge is required to set rules and thresholds to label each scenario. However, these approaches can be inaccurate where the parameters are close to their corresponding thresholds or the thresholds are not set correctly for some reason (e.g., because there are not enough samples). To overcome the above challenges, machine learning approaches have been proposed as complementary methods, especially to handle unknown scenarios.

This thesis proposes an unsupervised approach to cluster motion trajectories that exploits the transitive-aware relation of trajectories. Given a dataset of trajectories, where a trajectory is a time series of the relative position of a nearby vehicle with respect to an ego vehicle, the goal is to associate each time series with a particular scenario that is unknown to the algorithm.

Regardless of whether safety testing is performed in-the-field or via simulations, the testing is more focused on naturalistic scenarios. However, it is essential to test the safety of AD in some high-risk and diverse scenarios that rarely occur in real life. Consequently, there is a need to augment real-world data with generated trajectories. For this reason, studies have used [11], [12] Generative Adversarial Networks (GANs) to generate synthetic scenarios. GANs have two main components: a generative model that attempts to generate a syntactic pattern, and a discriminator model that attempts to distinguish its input as either real (from the domain) or fake (synthetic) [13].

Road networks bottleneck identification

A road network can be mathematically represented in different ways depending on the desired use case. This thesis studies bottleneck identification problems in road networks represented by a weighted graph $\mathcal{G}(\mathcal{V},\mathcal{E},\mathbf{w})$, where nodes $\mathcal{V}$ correspond to intersections and edges $\mathcal{E}$ to road segments.

Nodes and edges in a road network can have various properties and attributes, such as coordinates for intersections and length for road segments. Edge weights $\mathbf{w}$, where w_e denotes the edge weight of $e \in \mathcal{E}$, is defined by a particular attribute of edges. Considering edge $(i,j) \in \mathcal{E}$ denotes an edge between nodes i and j where $i,j \in \mathcal{V}$, and graph $\mathcal{G}$ is directed if $w_{(i,j)} \neq w_{(j,i)}$. It is assumed that graph $\mathcal{G}$ has no self-loop.

The bottleneck in a road network between a specific source and a destination, is the edge with the maximum edge weight that one cannot avoid. Therefore, a bottleneck over the network is equal to the minimax edge. By negating the edge weights, the identification of the bottleneck can also be considered as the widest path or maximum capacity path problem [14].

Based on Section 2.1.1, the first step in finding minimax distances between nodes in a graph $\mathcal{G}$ is to find a minimum spanning tree (MST). The common

algorithms to find an MST (Prim's and Kruskal's) are applied to an undirected graph; however, the direction of a segment is critical information in most road network applications. A modification of Dijkstra's algorithm [15] can be used to find the MST when $\mathcal{G}$ is directed.

In classical bottleneck identification problems, the edge weights are given. However, in real-world applications, the edge weights of a road network may be uncertain and stochastic. In this thesis, the edge weights are assumed to be stochastic, with fixed and unknown distribution parameters, and the bottleneck identification problem is formulated as an online learning problem, as explained in Section 2.2, to learn the distribution parameters while solving the bottleneck identification problem.

2.2 Online Decision Making

Online learning provides a framework to analyze sequential decision making. The goal is to derive a policy π that maximizes a long-term goal in horizon T. It is common to consider the long-term goal to be *regret minimization*, where at time $t \in [T]$ the instantaneous regret is the difference between the algorithm's performance and the optimal performance, and the *cumulative regret* is the sum of regrets received up to time t.

In an online learning setting, a learner (agent) and its environment interact at each time step t of horizon T. In particular, at time step t, the environment reveals some information about the current state $S_t \in \mathcal{S}$ to the learner. Then, the learner chooses an action $a \in \mathcal{A}_t$ where $\mathcal{A}_t$ are all possible actions at time step t, based on the current state S_t and the learner's policy π. Consequently, the learner receives a reward R_t from the environment. Then, the learner updates its policy π based on R_t and the environment also enters a new state S_{t+1}.

Multi-armed bandit is a special framework for making decisions over time under uncertainty. A bandit agent does not consider state transitions, which means that the learner's available choices and rewards for the next step are not affected by its decisions in the current time step.On the contrary, when the environment is influenced by the agent's choices (i.e., state transitions), reinforcement learning models are applicable [16].

2.2.1 Multi-armed Bandit Algorithm

The term *multi-armed bandit* is originated from a gambling scenario in which a gambler faces several identical slot machines, each with possibly different payoffs. In this scenario, the gambler aims to play an arm that maximizes the cumulative payoff. However, since the gambler receives a reward only for the chosen arm, they have to explore and try different arms to acquire new information. Therefore, in each round, the gambler should decide whether to try to make the optimal short-term decision based on the available information or to explore to gain more knowledge.

The problem of multi-armed bandits has many different variants and setting. This thesis focuses on the stochastic bandit problem represented by Algorithm 1, where the agent has the same set of possible actions $\mathcal{A}_t$ with size K to choose from for all $t \in [T]$. In this thesis, the concept of arms is used interchangeably

Algorithm 1 Stochastic multi-armed bandits problem

Input: $\mathcal{A}$, T, and $\mathcal{D}_a$

1: **for** $t \leftarrow 1, \ldots, T$ **do**
2: $a_t \leftarrow$ Choose arm $a_t \in \mathcal{A}_t$ to play according to bandit algorithm
3: $r(a_t) \leftarrow$ Observe reward based on $\mathcal{D}_{a_t}$
4: Update algorithm's parameter using observed reward $r(a_t)$
5: **end for**

with actions. Then, the agent chooses action a_t to play according to its algorithm and observes a corresponding reward $r(a_t)$, where $r(a_t)$ is *i.i.d.* drawn from a fixed and unknown distribution $\mathcal{D}_{a_t}$. The goal of the algorithm is to maximize its total reward over the time horizon T.

However, maximizing total reward is often framed as a regret minimization problem. Assuming mean reward vector $\boldsymbol{\mu} \in \mathbb{R}^K$, where $\mu(a) := \mathbb{E}[\mathcal{D}_a]$, the best mean reward is $\mu^* := \max_{a \in \mathcal{A}} \mu(a)$, and the gap of arm a is defined by $\Delta(a) := \mu^* - \mu(a)$, which describes how bad arm a is compared to the best arm. Then the cumulative regret at time T, $\mathcal{R}(T)$ is defined by Eq. (2.2).

$$\mathcal{R}(T) = \sum_{t=1}^{T} \big(\mu^* - \mu(a_t)\big) \tag{2.2}$$

Due to possible randomness in actions and rewards, usually the expected regret $\mathbb{E}[\mathcal{R}(t)]$ is of interest. Different bandit algorithms are compared based on their dependencies between expected regret and time horizon (and sometimes number of arms). Some of the most important algorithms for bandit problems are briefly presented here.

Epsilon-greedy algorithm

The ϵ-greedy algorithm is a simple approach that optimizes for short term benefits (exploitation) with probability of $1-\epsilon$ and uniformly randomly chooses an arm $a \in \mathcal{A}$ with probability of ϵ (exploration). The probability of random choice can be constant, ϵ, or decreasing by time t, ϵ_t. Assuming that $\epsilon_t \sim t^{-1/3}$, then the ϵ_t-greedy algorithm achieves the regret bound $\mathbb{E}[\mathcal{R}(t)] \leq t^{2/3}\mathcal{O}(K \log t)^{1/3}$, at every time $t \leq T$. [17].

Upper confidence bound algorithms

Upper Confidence Bound (UCB) is a class of bandit algorithms based on the principle of taking optimistic decisions in uncertain environments. That is, at each time step t, the algorithm plays the arm with the highest empirical average of reward up to time t plus a *confidence radius* which is inversely proportional to the number of times the arm has been played [17], [18].

The UCB policy leads to sublinear regret because an arm is selected based on either a high empirical average of reward or a high confidence radius. If a non-optimal arm is played due to its high confidence radius, it is later discarded due to the inverse relationship between the confidence radius and the number of times an arm was played. The notion of confidence radius itself is often derived from the concentration properties of the reward distributions. Suppose

Algorithm 2 Upper Confidence Bound

Input: $\mathcal{A}$, T, δ, and $\mathcal{D}_a$

1: **for** $t \leftarrow 1, \ldots, T$ **do**
2: $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \mathrm{UCB}_a(t-1, \delta)$
3: Observe reward $r(a_t)$
4: Update upper confident bounds
5: **end for**

that for each arm $a \in \mathcal{A}$, the reward is a 1-subgaussian random variable with mean $\mu(a)$ and $\hat{\mu}(a) = \frac{1}{n}\Sigma_{i=1}^{n} r(a_i)$ where n indicates how many times that arm was played. Then, by Chernoff-Hoeffding bound Eq. (2.3) is derived for all confidence levels $\delta \in (0,1)$. The confidence level should be small enough to ensure optimism with high probability, but not too small, which leads to trying suboptimal arms for too many time steps [16], [17].

$$\mathcal{P}\Big(\mu(a) \geq \hat{\mu}(a) + \sqrt{\frac{2\log(1/\delta)}{N(t,a)}}\Big) \leq \delta \tag{2.3}$$

Algorithm 2 shows a UCB agent [17]. In line 2 of the algorithm, $\mathrm{UCB}_k(t-1,\delta)$ is defined by Eq. (2.4), where $\hat{\mu}(a)_{(t-1)}$ indicates the empirical average of arm a' reward by time t, and $N(t,a)$ is a random variable indicating how many times arm $a \in \mathcal{A}$ has been played by time t.

$$\mathrm{UCB}_a(t-1,\delta) = \begin{cases} \infty, & \text{if } N(t,k) = 0; \\ \hat{\mu}(a)_{(t-1)} + \sqrt{\frac{2\log(1/\delta)}{N(t,a)}}, & \text{otherwise.} \end{cases} \tag{2.4}$$

However, there are many variants of the UCB algorithms; all (that I am aware of) consist of a term for exploitation and a term for exploration based on optimism. The exact regret bounds of these algorithms can be discussed individually.

Thompson Sampling algorithm

The Bayesian bandit is another variant of the stochastic bandit in the presence of prior knowledge, assuming that the parameters of the arms' reward distributions are drawn from a known prior distribution. The Thompson Sampling (TS) algorithm proposed by Thompson [19] is a natural randomized Bayesian algorithm for minimizing regret by assuming the existence of uncomplicated prior distributions over the parameters of the arms' reward distributions, where the agent chooses an arm based on the arms' posterior probabilities of being the optimal arm at time $t \in [T]$.

Recently, TS has attracted considerable attention in various problems, including online learning with derived theoretical guarantees [20], with diffrent applications such as finding the shortest paths in graphs [21]–[24].

Algorithm 3 shows TS for stochastic multi-armed bandit (MAB) where the reward distributions of the arms are Gaussian, with the reward means θ_a drawn from a Gaussian prior distribution $\mathcal{N}(\mu_{a,0}, \sigma_{a,0})$ and a known variance ς^2 (for simplicity, as in other works [25], [26]). Given a Gaussian prior and a Gaussian likelihood, the posterior has the same parametric form as the prior

Algorithm 3 Thompson Sampling

Input: $\mathcal{A}$, T, $\mathcal{N}(\mu_{a,0}, \sigma_{a,0})$

for $t \leftarrow 1, \ldots, T$ **do**
  for $a \in \mathcal{A}$ **do**
    $\tilde{\theta}_a \leftarrow$ Sample from posterior $\mathcal{N}(\mu_{a,t-1}, \sigma^2_{a,t-1})$
  end for
  $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \tilde{\theta}_a$
  $r(a_t) \leftarrow$ observe reward of playing a_t
  $\mu_{a_t,t}, \sigma^2_{a_t,t} \leftarrow$ Update the parameters of the posterior distribution for arm a_t
end for

due to conjugacy, and the update of the parameters of the Gaussian posterior is derived in a closed form.

2.2.2 Combinatorial Bandit Setting

This thesis focuses on stochastic combinatorial semi-bandit setting [27], where the agent chooses a super arm $\boldsymbol{a}_t \subseteq \mathcal{A}$ at each time step t in horizon T. A super arm $\boldsymbol{a}_t$ is a subset of base arms $a \in \mathcal{A}$. Sometimes there are constraints on super arms; therefore $\mathcal{I}$ is called the set of *feasible* super arms where $\boldsymbol{a}_t \in \mathcal{I} \subseteq 2^{\mathcal{A}}$.

Many of the bandit algorithms mentioned in Section 2.2.1 have been adapted to the combinatorial setting, such as UCB for combinatorial setting [28], and its Bayesian variants [29]. Bottleneck avoidance problems have been studied with a variant of UCB in a combinatorial pure exploration setting [30] with a different problem formulation and method, compared to what is presented in this thesis. The main distinction between their environment and the classical combinatorial semi-bandit environment in terms of how agents interact with the environment is that they allow agents to test individual arms sequentially to find the best feasible solution to the combinatorial problem, as opposed to being constrained to select a subset of actions at once for each time step. In order to define the setting as explained, the feedback from each edge should be observable.

Similar to UCB, Thompson Sampling has been adapted to combinatorial bandit setting with intact theoretical guarantees [20], one application being to finding shortest paths in graphs [21]–[24].

2.3 Concurrent Programming

Multiprocessor architecture leads to more efficient data processing by providing the capability for parallel computations. A multiprocessor contains two or more processing units that can execute threads simultaneously. These units share some resources, such as caches, at some level and interact through shared memory. In asynchronous thread execution, the threads can be executed at different speeds so that the steps of the threads can be interleaved. By using appropriate synchronization methods, the correctness of these threads is maintained.

Algorithm 4 Compare And Swap (CAS)

function CAS (int^* p, int old, int new)

1: **if** $^*p = old$ **then**
2: **return** *False*
3: **end if**
4: $^*p \leftarrow new$
5: **return** *True*

2.3.1 Atomic Primitives

Hardware instructions, known as *atomic primitives*, consist of a series of steps that are executed sequentially (without interruption). With concurrent accesses, atomics provide consistency and accuracy. The most frequent application of atomics is when multiple threads modify a single shared object. Atomics can be viewed as a congestion that can affect the performance and scalability of multithreaded applications because the changes made by each thread must be serialized to ensure correctness. Atomics are used extensively in the development of concurrent data structures [31]–[33].

The maximum number of threads that can reach an agreement with the atomic primitive without waiting is called the *consensus number* and is a crucial aspect of an atomic. Compare And Swap (CAS), Fetch And Increment (FAI), Test And Set (TAS) presented respectively in Algorithm 4, Algorithm 5, and Algorithm 6 are the most common atomics with consensus number equal to ∞, 2, and 2, respectively [31].

2.3.2 Cache Coherence

In a multiprocessor system, each core has its own cache, known as the L1 cache. A core can have more than one exclusive cache. When a core executes a thread on a particular memory location, the target memory location should be available in the core's L1 cache. However, copies of that memory location might be in other caches. Cache coherence techniques are required to ensure consistency of shared data resources that are available locally in multiple caches. Depending on the number of defined cache states, cache coherence protocols are classified into different categories. The focus of this thesis is on MESI [34] and MESIF [35].

The MESI protocol consists of four states, Modified (M), Exclusive (E), Shared (S), Invalid (I). In MESIF there is a fifth state called Forward (F). In the Modified state, the content is exclusive to the current cache and has been modified from the value in main memory. In the Exclusive state, the content is exclusive to the current cache but is identical to the value in main

Algorithm 5 Test And Set (TAS)

function TAS (int^* p)

1: int $old \leftarrow^* p$
2: $^*p \leftarrow True$
3: **return** old

Algorithm 6 Fetch And Increment (FAI)

function FAI (int^* p, int add)

1: int $old \leftarrow^* p$
2: $^*p \mathrel{+}= add$
3: **return** old

memory. The Shared state indicates that other caches have an identical copy of this cache line and that it matches the value in main memory. The Invalid state means that the value in the current cache is invalid. In MESIF there is another state called Forward. If there are multiple copies of a memory address in different caches, one of them holds the F state to forward the cache value in the response to a copy request.

2.3.3 Execution Time

Atomics are typically used to provide exclusive access when modifying the state of a shared object, making them a natural candidate to become the bottlenecks that hinder the scalability of multithreaded programs. In Chapter 5, we model atomics' execution time, which consists of two factors:

Stall time: Thread contention occurs when one or more threads are waiting to access and modify a particular cache line. When contention occurs, one thread gains exclusive access to the cache line and executes an atomic, so that preventing the other threads from having exclusive access until the current atomic execution is ended. When the current execution is finished, another thread accesses the cache line in accordance with the scheduling strategy of the hardware. This time interval that a thread must wait to access the cache line is called the *stall time* and is the first element of the execution time.

Atomic execution time: The period of time that begins with the acquisition of the cache line and ends with the completion of atomic execution, and is the second element of execution time. This component is affected by the state of the system when the thread acquires the cache line. For example, instead of having to fetch the specified cache line from memory, execution will be faster if it is already in the L1 of the core where the thread is currently executing.

Chapter 3

Thesis Challenges and Contributions

This thesis explores the pairwise relations between data points in the context of offline and online learning models. The following chapter provides a brief overview of the problems studied in the two settings, the methods used, and the contributions developed in each of the attached papers.

3.1 Offline Learning Models

3.1.1 Paper A

Atomic primitives are essential for concurrent programming. Therefore, several studies introduce benchmarks to investigate the behavior of atomic primitives and their latency/bandwidth when reading and writing data in multi-core architectures with complex memory hierarchies [36]–[39].

In these studies, however, the cache coherence states of the data are explicitly specified, and the measurements are made with complete control over the state of the system, including the location of the cache containing a copy of the data and the order of the threads running on the shared memory locations. For example, to measure the latency of an atomic primitive on a cache line in the modified state (Section 2.3.1), in the studies cited, a specific location in the L1 cache should first be brought into the modified state by a thread modifying the shared memory location. Then another waiting thread executes the atomic primitive on that line, and a timer counts the execution time of that thread. While this measurement gives us some insight, it does not show the performance metrics in uncontrolled instances like real applications.

In *paper A*, we study the performance of atomic primitives in a less controlled setting. Inspired by the results of [39] showing that the latency of atomic primitives strongly depends on the location of the targeted data and the cache coherence state, we propose a method based on modeling cache line bounces between threads (more precisely, between cores executing the threads) accessing the shared cache lines. Thus, to model the performance of atomics, we observe a sequence of threads executing the atomics, and then model the sequence of

events with a stochastic process that satisfies the Markov property.

This paper proposes a representation of the process by a Markov chain with a transition matrix determined by some properties of the target architecture: the number of cores and their arrangement, and the degree of cache sharing. The focus is on two hardware platforms, a medium-scale multi-core platform (Intel Xeon E5) and a large-scale many-core platform (Intel Xeon Phi). Then, the proposed method measures the cost of each state transition in terms of latency, throughput, fairness, and energy consumption. Finally, using the state transition matrix and the cost of each transition, we model the performance of atomics, and provide a detailed evaluation of the desired performance metrics to demonstrate that our model accurately captures the performance.

3.1.2 Paper B

The transitive relations between data points rather than their direct (dis)similarities provide a better representation of the underlying patterns of data in various applications, e.g., minimax distances in clustering problems. Minimax distance measure is a link-based distance, where to calculate the minimax distance between two data points, all different paths between these two points should be considered [40].

Therefore, the first step in computing the minimax distance of a pair of data points is to find all possible paths between them. This step is computationally and memory-wise intensive. However, there are some studies [41]–[43] that provide computationally efficient methods for computing pairwise minimax distances; to our knowledge, there is no other study on memory-efficient methods for pairwise minimax distances, and no improvement in the memory requirement of $\mathcal{O}(N^2)$ has been obtained for this problem.

In *paper B*, we present two memory-efficient methods to reduce memory requirements and achieve linear space complexity: i) At the expense of higher computational cost, we propose a method to determine the exact minimax distances between each pair of objects. ii) With the goal of reducing the computational cost, we introduce an effective sampling strategy based on minimax distances where the use of minimax distances is limited to the sample space. We show that the required information for minimax distances between any data points can be captured by the proposed sampling method.

Assuming that it is not possible to store the pairwise distances of a data set with size N directly in linear memory, in the first approach we propose an algorithm that arranges the data in a hierarchy of components. The relations between intra-component data and inter-component data allow us to encode pairwise minimax distances of N objects, and store the compressed information with linear memory requirements. We also present a method for decoding the minimax distance of arbitrary nodes.

While the proposed algorithm can compute exact pairwise distances, the memory efficiency comes at the price of a heavy computational burden. To reduce this, we also propose an approximate sampling method based on the hierarchy of components mentioned above. A sample in this context refers to any data point within a component. We claim that the minimax distances between our samples coincide with the pairwise minimax distances of the entire data points. We demonstrate the sampling framework for the downstream task

of clustering, for which it is well suited, but note that it could be similarly applied to many tasks. To evaluate our method, we apply a clustering task to the samples and then generalize the labels of the samples to the objects outside the samples.

3.1.3 Paper C

To ensure the safety of Autonomous Driving (AD) vehicles, they would have to be driven more than a hundred million kilometers. However, the efficiency of in-the-field testing of AD is arguable. An alternative approach is scenario-based verification and validation, where a driving scenario trajectory is a time series consisting of information about surrounding vehicles relative to an ego-vehicle over different time periods. It is common to consider the relative lateral and longitudinal positions of the vehicle w.r.t. the ego vehicle as elements of a trajectory. To create such a dataset, collected raw sensor data must be processed and then categorized into different driving patterns (e.g. cut- in, overtaking, etc.). However, there are many scenarios that are not very frequent but often hazardous. Therefore, the collected data is not sufficient and should be augmented by synthetic data, which can be generated by generative models such as Generative Adversarial Networks (GAN) [13] or other variants of GAN such as Recurrent Auto-Encoder GAN [44]–[46].

Knowledge-based approaches allow us to categorize given trajectories based on established rules and thresholds. These rules are known in advance and are determined by a team of experts. Although this method is simple and effective in many cases and uses expert knowledge, it can be error-prone in cases where the thresholds have too small margins and can also omit unknown driving events. Therefore, as a complementary approach, we investigate the use of machine learning tools, in particular the clustering approach, to speed up and verify the obtained scenario labels. However, these approaches have their own challenges due to the nature of the problem, where we need to label a huge amount of data in the form of time series with different lengths. For example, padding is used to address varying length of each time-series in some domains, but this is non-trivial to define neutral padding for trajectories [47], [48].

To deal with trajectories of different lengths, model-based clustering techniques such as Mixture of Hidden Markov Models [49] have been used for trajectories, where each trajectory is appointed to a generative model. However, the weakness of this approach is the sensitivity of the models to the initialization step; also, it is very computationally intensive. Later, deep neural networks were used for clustering trajectories, where a neural network first transforms the trajectories by summarizing their critical parts and enriching them with the context-derived features of their geographical location. Then, it learns a powerful representation of the trajectories in latent space and finally clusters the trajectories [50]. Despite some improvements by the above study, neural network approaches still suffer from the high computational cost and sensitivity to hyper-parameter tuning.

In *Paper C*, we propose a non-parametric framework to cluster vehicle trajectories with different lengths. In our approach, we first determine the temporal relations of the trajectories and then embed the dissimilarities of the

trajectories with t-SNE into a vector space. We then extract the transitive relations in the embedded space using minimax distances and apply multidimensional scaling to represent the trajectories in a vector space, allowing us to apply clustering methods such as the Gaussian Mixture Models (GMM). We evaluate our approach on real-world and synthetic trajectory data sets and obtain promising results, despite the complexity introduced by trajectories of different lengths. Furthermore, we extend our approach to validate the augmentation of the real datasets with the synthetic data generated by variations of GAN concluding the consistency of the generated and real-world trajectories.

3.2 Online Learning Models

3.2.1 Paper D

A bottleneck in a path connecting a source and a destination is defined as the edge with the highest cost or weight according to some defined criteria. In a bottleneck identification problem, the objective is to find the path with the smallest bottleneck. Therefore, a bottleneck identification problem can be considered as a minimax problem. The widest path problem and the maximum capacity path problem [14] are equivalent to the bottleneck identification problem.

The applications of the mentioned problems can be defined in a specific context. For example, in the context of a road network, a bottleneck between a source and a destination is a road segment along a path between them, the cost of which is simultaneously maximal w.r.t. the edges on the path, and minimal w.r.t. maximum edge of all possible paths. For example, if weights indicate travel time, the bottleneck is the largest road segment that cannot be avoided when commuting between source and destination.

A classic problem setting for identifying bottlenecks in a network assumes that the network is deterministic and the edge weights are given. However, this assumption does not hold for many applications where the network is unknown or stochastic and a learning algorithm must learn unknown parameters while finding the bottleneck.

In *Paper D*, we propose an online bottleneck identification framework with the goal of finding a path that minimizes the maximum stochastic edge weight along that path (the minimax path). We treat the problem as a combinatorial semi-bandit problem, where choosing a path is viewed as playing a super-arm. We use combinatorial Thompson Sampling to solve the problem. We make the Bayesian assumption that the mean values of the edge weights are drawn from a known prior distribution. We then derive an upper bound of $\tilde{\mathcal{O}}(K\sqrt{T})$ where K is the number of base arms and T is the horizon, with a detailed proof in *Appendix A*.

The mentioned objective requires computing the expected maximum of Gaussian random variables which is computationally intractable when there are many base arms. Therefore, we use an approximate objective [51] modified to find a super-arm that minimizes the maximum expected edge weight of the base arm. We then evaluate combinatorial Thompson Sampling with our proposed formulation in two real-world networks, a road network, and a social network.

3.2.2 Paper E

Paper E builds on the work in *Paper D*, where we formulate a bottleneck identification problem in a combinatorial semi-bandit environment. In this work, however, we aim to exploit additional information available for each base arm, referred to as its context. For example, in road networks, each road segment may have some attributes such as length, coordinates, speed limit, etc., which may affect the reward. For a contextual multi-armed bandit, the reward for each time step depends on the context and the action played at that time.

The relations between the expected rewards of the arms and the corresponding observed feature vectors serve as inspiration for several typical assumptions made in contextual bandit problems regarding the expected reward function. Studies in [52]–[54] use ridge regression to estimate the parameters of the reward function under the assumption that the reward function is linear with context. Other works [55], [56] have proposed LinUCB-based combinatorial MAB algorithms with application-specific objectives. The Thompson Sampling algorithm with linear arm rewards has been explored by [26], [57].

In *paper E*, we develop a unified online learning method based on contextual combinatorial semi-bandits, learning the properties of the underlying network in addition to identifying bottlenecks. We consider the approximate objective mentioned in Section 3.2.1, where the goal is to find a super-arm that minimizes the maximum expected loss of the base arm given the contextual information. In this framework, we study an ϵ-greedy agent, LinUCB, BayesUCB, as well as Thompson Sampling in a contextual combinatorial semi-bandit environment. We also propose an extension of the neural network approach, NeuralUCB [58], for a combinatorial environment. We then evaluate our approach on a real-world application, a road network.

Chapter 4

Concluding Remarks & Future Works

This thesis studies offline and online models for learning pairwise relations in data. From the perspective of offline learning, it introduces a performance model built upon the bounces of cache lines around executing cores, where various performance metrics such as energy consumption, throughput, latency, and fairness are modeled and discussed. Evaluation of the model has shown that the model is capable of capturing atomic's behavior on two modern multicore processors.

Then, the thesis focuses on minimax distance as a pairwise measure capable of detecting latent features and patterns in a data set. First, two memory-efficient approaches were presented to reduce the memory required to compute minimax distances. Both methods are based on the proposed hierarchical representation, where it assigns data points into hierarchical components, and the relations of a pair of data can be interpreted differently depending on whether they are in the same component or not. Later in this thesis, minimax distances were explored in clustering motion trajectories, leading to a clustering framework that does not require the specification of hyper-parameters.

The second part of the thesis focuses on the use of online learning approaches to introduce two frameworks to find a minimax distance of any pair of data points when the edge weights are not deterministic. Then, the proposed methods were applied to the problem of identifying bottlenecks in networks, mainly in road networks.

Both frameworks (*Paper E and D*) use different bandit algorithms in the context of stochastic combinatorial multi-armed bandits. We considered a stochastic criterion as weights of edges in the networks and use a Bayesian approach for probabilistic modeling of the criterion associated with each edge. In *Paper E*, we used combinatorial versions of Thompson Sampling and BayesUCB. Due to interactivity of computing the expected maximum of Gaussian random variables when there are many of such variables, we employed an approximate problem formulation alternating maximization and expectation operations. We derived an upper bound for the Bayesian regret of combinatorial Thompson Sampling applied to the online minimax path problem. Finally, we evaluated the approximate method on several real-world networks and datasets.

The second online framework (*Paper D*) is a unified contextual combinatorial bandit framework for identifying bottlenecks in networks that takes into account the available contextual information. Again, we used the minimax concept to find the path containing a bottleneck edge given contextual information. We adapted Thompson Sampling, BayesUCB, LinUCB, and ϵ_t-greedy to our framework, and proposed a NeuralUCB method for a combinatorial setting. We evaluated the performance of the proposed framework on a real-world application for a road network.

The MAB neglects several important factors of the environment in many real-world applications. For example, the number of available arms at each time step is considered fixed in a simple MAB environment. However, in many applications, such as identifying bottlenecks in a road network, the available arms at time t may be a subset of a known finite set of arms, or even a subset of an infinite number of arms (in other applications). Moreover, the assumption of an *i.i.d.* reward model may not be realistic for many real-world problems where there is a latent structure that induces correlations between options. By learning and utilizing the correlation or structure between arms, the agent can reduce the exploitation in bandit algorithms.

Another possible research direction is to investigate how the contextual bandit can address the cold-start in online minimax distance problems, where the agent starts with a very limited amount of contextual information. This research direction is correlated to the problem of dealing with missing contextual data.

References

[1] P. Gao, H.-W. Kaas, D. Mohr, and D. Wee, "Automotive revolution–perspective towards 2030 how the convergence of disruptive technology-driven trends could transform the auto industry," *Advanced Industries, McKinsey & Company*, 2016.

[2] B. Fischer and J. M. Buhmann, "Path-based clustering for grouping of smooth curves and texture segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 4, pp. 513–518, 2003.

[3] K.-H. Kim and S. Choi, "Neighbor search with global geometry: A minimax message passing algorithm," in *Proceedings of the 24th international conference on machine learning*, 2007, pp. 401–408.

[4] K.-H. Kim and S. Choi, "Walking on minimax paths for k-nn search.," in *AAAI*, 2013.

[5] M. H. Chehreghani, "K-nearest neighbor search and outlier detection via minimax distances," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, SIAM, 2016, pp. 405–413.

[6] R. W. Floyd, "Algorithm 97: Shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.

[7] R. B. Zadeh and S. Ben-David, "A uniqueness theorem for clustering," *arXiv preprint arXiv:1205.2600*, 2012.

[8] T. Hu, "The maximum capacity route problem," *Operations Research*, vol. 9, no. 6, pp. 898–900, 1961.

[9] M. Haghir Chehreghani, "Efficient computation of pairwise minimax distance measures," in *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 799–804.

[10] T. Menzel, G. Bagschik, and M. Maurer, *Scenarios for development, test and validation of automated vehicles*, 2018.

[11] A. Demetriou, H. Alfsvåg, S. Rahrovani, and M. H. Chehreghani, *A deep learning framework for generation and analysis of driving scenario trajectories*, 2020.

[12] P. Dendorfer, S. Elflein, and L. Leal-Taixé, "Mg-gan: A multi-generator model preventing out-of-distribution samples in pedestrian trajectory prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 158–13 167.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[14] M. Pollack, "The maximum capacity through a network," *Operations Research*, pp. 733–736, 1960.

[15] O. Berman and G. Y. Handler, "Optimal minimax path of a single service unit on a network to nonservice destinations," *Transportation Science*, vol. 21, no. 2, pp. 115–122, 1987.

[16] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[17] A. Slivkins *et al.*, "Introduction to multi-armed bandits," *Foundations and Trends® in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[18] P. Auer, "Using confidence bounds for exploitation-exploration tradeoffs," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 397–422, 2002.

[19] W. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3–4, pp. 285–294, 1933.

[20] S. Wang and W. Chen, "Thompson sampling for combinatorial semi-bandits," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, 2018, pp. 5114–5122.

[21] K. Liu and Q. Zhao, "Adaptive shortest-path routing under unknown and stochastically varying link states," in *2012 10th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2012, pp. 232–237.

[22] Y. Gai, B. Krishnamachari, and R. Jain, "Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1466–1478, 2012.

[23] Z. Zou, A. Proutiere, and M. Johansson, "Online shortest path routing: The value of information," in *2014 American Control Conference*, 2014, pp. 2142–2147.

[24] N. Åkerblom, Y. Chen, and M. Haghir Chehreghani, "An online learning framework for energy-efficient navigation of electric vehicles," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 2020, pp. 2051–2057.

[25] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Mathematics of Operations Research*, vol. 39, no. 4, pp. 1221–1243, 2014.

[26] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International conference on machine learning*, PMLR, 2013, pp. 127–135.

[27] N. Cesa-Bianchi and G. Lugosi, "Combinatorial bandits," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1404–1422, 2012, JCSS Special Issue: Cloud Computing 2011, ISSN: 0022-0000.

[28] W. Chen, Y. Wang, and Y. Yuan, "Combinatorial multi-armed bandit: General framework and applications," in *International conference on machine learning*, PMLR, 2013, pp. 151–159.

[29] E. Kaufmann, O. Cappé, and A. Garivier, "On bayesian upper confidence bounds for bandit problems," in *Artificial intelligence and statistics*, PMLR, 2012, pp. 592–600.

[30] Y. Du, Y. Kuroki, and W. Chen, *Combinatorial pure exploration with bottleneck reward function*, 2021.

[31] M. Herlihy, N. Shavit, V. Luchangco, and M. Spear, *The art of multiprocessor programming*. Newnes, 2020.

[32] B. Norris and B. Demsky, "Cdschecker: Checking concurrent data structures written with c/c++ atomics," in *Proceedings of the 2013 ACM SIGPLAN international conference on Object oriented programming systems languages and applications*, 2013, pp. 131–150.

[33] M. M. Michael and M. L. Scott, "Simple, fast, and practical non-blocking and blocking concurrent queue algorithms," in *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, 1996, pp. 267–275.

[34] G. F. Pfister, *In search of clusters*. Prentice Hall PTR Upper Saddle River, NJ, 1998, vol. 2.

[35] R. Intel, "Intel 64 and ia-32 architectures optimization reference manual," *Intel Corporation, Sept*, 2014.

[36] D. Molka, D. Hackenberg, R. Schöne, and M. S. Müller, "Memory performance and cache coherency effects on an intel nehalem multiprocessor system," in *PACT*, IEEE Computer Society, 2009, pp. 261–270.

[37] D. Hackenberg, D. Molka, and W. E. Nagel, "Comparing cache architectures and coherency protocols on x86-64 multicore SMP systems," in *MICRO*, ACM, 2009, pp. 413–422.

[38] D. Molka, D. Hackenberg, and R. Schöne, "Main memory and cache performance of intel sandy bridge and AMD bulldozer," in *MSPC@PLDI*, ACM, 2014, 4:1–4:10.

[39] H. Schweizer, M. Besta, and T. Hoefler, "Evaluating the cost of atomic operations on modern architectures," in *PACT*, IEEE Computer Society, 2015, pp. 445–456.

[40] A. Moscovich, A. Jaffe, and N. Boaz, "Minimax-optimal semi-supervised regression on unknown manifolds," in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 933–942.

[41] M. Haghir Chehreghani, "K-nearest neighbor search and outlier detection via minimax distances," in *SIAM International Conference on Data Mining (SDM)*, 2016, pp. 405–413.

[42] M. Haghir Chehreghani, "Efficient computation of pairwise minimax distance measures," in *IEEE International Conference on Data Mining, ICDM*, 2017, pp. 799–804.

[43] M. H. Chehreghani, "Efficient computation of pairwise minimax distance measures," in *2017 IEEE International Conference on Data Mining (ICDM)*, IEEE, 2017, pp. 799–804.

[44] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[45] D. Donahue and A. Rumshisky, "Adversarial text generation without reinforcement learning," *ArXiv*, vol. abs/1810.06640, 2018.

[46] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*, PMLR, 2019, pp. 7354–7363.

[47] T. W. Liao, "Clustering of time series data—a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[48] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[49] J. Martinsson, N. Mohammadiha, and A. Schliep, "Clustering vehicle maneuver trajectories using mixtures of hidden markov models," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 3698–3705.

[50] W. Wang, F. Xia, H. Nie, *et al.*, "Vehicle trajectory clustering based on dynamic representation learning of internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3567–3576, 2021.

[51] N. Åkerblom, F. S. Hoseini, and M. Haghir Chehreghani, "Online learning of network bottlenecks via minimax paths," *Machine Learning*, pp. 131–150, 2023.

[52] Y. Abbasi-yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011.

[53] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proceedings of the Annual Conference on Learning Theory*, 2008.

[54] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 661–670.

[55] L. Qin, S. Chen, and X. Zhu, "Contextual combinatorial bandit and its application on diversified online recommendation," in *Proceedings of the 2014 SIAM International Conference on Data Mining*, SIAM, 2014, pp. 461–469.

[56] C. Zhang, P. Ren, and Q. Du, "A contextual multi-armed bandit approach to caching in wireless small cell network," in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, IEEE, 2017, pp. 1–6.

[57] M. Abeille and A. Lazaric, "Improved regret bounds for thompson sampling in linear quadratic control problems," in *International Conference on Machine Learning*, PMLR, 2018.

[58] D. Zhou, L. Li, and Q. Gu, "Neural contextual bandits with ucb-based exploration," in *International Conference on Machine Learning*, PMLR, 2020, pp. 11 492–11 502.

Part II

Collection of Papers