Thesis for the degree of Doctor of Philosophy

Knowledge Transfer and Active Learning for Representation Learning in the Absence of Sufficient Supervision

Arman Rahbar

Department of Computer Science and Engineering
Division of Data Science and AI
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2025

Knowledge Transfer and Active Learning for Representation Learning in the Absence of Sufficient Supervision
Arman Rahbar
ISBN 978-91-8103-239-0

© Arman Rahbar, 2025.

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5697
ISSN 0346-718X

Department of Computer Science and Engineering
Chalmers University of Technology
SE–412 96 Göteborg, Sweden
Telephone + 46 (0) 31 – 772 1000

Typeset by the author using LaTeX.

Printed by Chalmers Reproservice
Göteborg, Sweden 2025

To my beloved Anahita, my light in life.
To my parents and my brother, my first supporters.
To Iran, my root and origin.

Abstract

In many machine learning applications, labels or feature values may be unavailable or expensive to obtain. This motivates the need to develop methods for learning data representations under limited supervision, where supervision refers to either labels or feature values. In this thesis, we study this important problem from two distinct perspectives.

First, we explore knowledge transfer to improve representation learning in the target domain by leveraging existing sources of knowledge. These include (i) kernel and neural embeddings, which we use to guide learning in wide neural networks based on insights from neural network theory, and (ii) labeled data from a different source domain. We propose an optimal transport framework that enables the use of source domain labels when target labels are not available.

Second, we study active feature acquisition, where the goal is to select a cost-efficient subset of features for each data point under a budget constraint. We review recent works in this area, and then focus on the online setting, where data arrives as a stream and no training data is available in advance. We introduce a general framework for this setting, using methods from online learning and bandits, such as Thompson sampling, to efficiently learn from streaming data. We present two formulations of the problem, based on a partially observable Markov decision process and a combinatorial multi-armed bandit. The latter allows us to generalize our framework to stochastic feature costs. These formulations enable us to theoretically analyze the performance of our online framework. Our experiments show that the proposed framework is competitive in both feature acquisition cost and prediction accuracy.

Keywords: representation learning, supervision, knowledge transfer, kernel, domain adaptation, active feature acquisition, online learning, combinatorial multi-armed bandit, Thompson sampling

Acknowledgments

I am deeply grateful to my supervisor, Morteza, for his continuous support, guidance, and encouragement throughout my PhD journey. Your support, profound knowledge, and positive attitude made this thesis possible. I am also thankful to my co-supervisor, Hossein. I am grateful to my examiner, Graham, for taking the time to provide valuable feedback.

I have been fortunate to collaborate with many talented researchers during my PhD. I would like to thank Ashkan, Devdatt, Niklas, Emilio, Linus, Chiranjib, Ziyu, Yuxin, and Hamid for the inspiring collaborations.

I would like to thank my opponent, Prof. Sindri Magnússon, and the grading committee members, Prof. Cassio de Campos, Prof. Joni Pajarinen, Prof. Josephine Sullivan, and Prof. Fredrik Johansson, for their time and effort in reviewing my thesis.

I am also grateful to the many wonderful people I have met and worked with, including Hampus, Deepthi, Jack, Emil, Shirin, Newton, Tobias, David, Adam, Adel, Birgit, Divya, Hampus, Lovisa, Mohammad, Richard, Simon, Hannah, Christopher, Anton, Nasser, Yossra, Alexander, Tobias, Simon, Dag, Rocio, Markus, Arsham, Andrea, Peter, Alexander, Selpi, Lena, Kolbjörn, Juan, and Ola. I apologize if I have unintentionally missed anyone.

I also would like to thank my new colleagues at Volvo Group for their kindness and support as I transitioned into a new role while completing this thesis.

I deeply appreciate the friendship and support of Seyed Mohammad, Firooz, Mehrdad, Ehsan, Amir, Karolina, Mehrzad, Farzaneh, Siavash, Samira, Ali, Peter, Mo, Milad, Ramin, Elham and Yoosof, who have made this journey more enjoyable.

I am forever grateful to my parents, Mahmood and Sakineh, and my brother, Erfan, for their endless love and support. I also thank my wife's parents, Nazi and Dariush, and my brother-in-law, Danial, for their kindness and encouragement.

Finally, I dedicate my deepest love and thanks to my wife, Anahita, whose patience, understanding, and unwavering support have been my greatest strength.

This research has been supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, project "Under-supervised Representation Learning" grant nr. 37200022.

Arman Rahbar
Göteborg, July 2025

List of Publications

This thesis is based on the following appended papers:

Paper 1. **Arman Rahbar**, Emilio Jorge, Devdatt Dubhashi, and Morteza Haghir Chehreghani. *Do Kernel and Neural Embeddings Help in Training and Generalization?* Neural Processing Letters (NPL), 55, 1681–1695, 2023.

Paper 2. **Arman Rahbar**, Ashkan Panahi, Morteza Haghir Chehreghani, Devdatt Dubhashi, and Hamid Krim. *Recovery Bounds on Class-Based Optimal Transport: A Sum-of-Norms Regularization Framework.* 40th International Conference on Machine Learning (ICML), 2023.

Paper 3. **Arman Rahbar**, Linus Aronsson, and Morteza Haghir Chehreghani. *A Survey on Active Feature Acquisition Strategies.* Under Review.

Paper 4. **Arman Rahbar**, Ziyu Ye, Yuxin Chen, and Morteza Haghir Chehreghani. *Efficient Online Decision Tree Learning with Active Feature Acquisition.* 32nd International Joint Conference on Artificial Intelligence (IJCAI), 2023.

Paper 5. **Arman Rahbar**, Niklas Åkerblom, and Morteza Haghir Chehreghani. *Cost-Efficient Online Decision Making: A Combinatorial Multi-Armed Bandit Approach.* Transactions on Machine Learning Research (TMLR), 2025.

My contributions to the papers are as follows:

- **Paper 1**: I contributed to the study design, empirical evaluation, and writing of the manuscript.
- **Paper 2**: I contributed to the empirical evaluation and writing of the manuscript.
- **Paper 3**: I contributed to the study design and writing of the manuscript. The first two authors contributed equally.
- **Paper 4**: I contributed to the study design, empirical evaluation, theoretical analysis, and writing of the manuscript.
- **Paper 5**: I contributed to the study design, empirical evaluation, theoretical analysis, and writing of the manuscript. The first two authors contributed equally.

The following paper was also published during my PhD studies, but it is not appended to the thesis:

Ashkan Panahi, **Arman Rahbar**, Chiranjib Bhattacharyya, Devdatt Dubhashi, and Morteza Haghir Chehreghani. *Analysis of Knowledge Transfer in Kernel Regime.* 31st ACM International Conference on Information and Knowledge Management (CIKM) [Full research paper], 2022.

List of Important Acronyms

AFA	–	Active Feature Acquisition
CMAB	–	Combinatorial Multi-Armed Bandit
DA	–	Domain Adaptation
EC^2	–	Equivalent Class Edge Cutting
IG	–	Information Gain
LCB	–	Lower Confidence Bound
MAB	–	Multi-Armed Bandit
OT	–	Optimal Transport
TS	–	Thompson Sampling
UCB	–	Upper Confidence Bound

Contents

Part I

Introductory Chapters

Chapter 1

Introduction

Machine learning algorithms rely heavily on features, which are pieces of information used to make predictions or decisions. Traditionally, finding good features requires domain knowledge and expert effort, a process known as *feature engineering*. However, this approach can be time-consuming and expensive. To address this challenge, a line of research has focused on methods to automate the discovery of useful features, a field known as *feature learning* or *representation learning*.

Many representation learning techniques rely on labeled data to learn effective representations. These techniques are usually referred to as *supervised* representation learning. However, obtaining labeled data is not always easy, as it often requires significant human effort, expert knowledge, or costly annotation processes. As a result, in many real-world scenarios, we have access to only a limited set of labeled examples. Similarly, acquiring the feature values for a data point can be expensive, as it may involve complex measurements, specialized equipment, or extensive manual effort. As an example, consider the medical domain. Diagnosing a disease (i.e., labeling) often requires expert annotations from trained physicians, which can be time-consuming and costly to obtain. Additionally, collecting feature values, such as laboratory test results, can be challenging, as it may require expensive equipment or cause patient discomfort.

In this thesis, we address both challenges: the scarcity of labeled data and the difficulty of obtaining complete feature values.

To learn effectively when sufficient labeled data is not available, we employ *knowledge transfer*. In simple words, knowledge transfer aims to leverage information from another task or domain to enhance the learning process in the task or domain of interest. We explore two settings for knowledge transfer.

In the first setting, we utilize the knowledge embedded in *kernel and neural embeddings* to improve the training of wide neural networks. Our results demonstrate that incorporating this knowledge enhances both optimization (i.e., convergence speed) and generalization (i.e., performance on unseen data) properties in neural networks.

In the second setting, we focus on transferring knowledge from a source domain with abundant labeled data to a target domain where labeled data is scarce. This is commonly known as *domain adaptation*. In this thesis, we propose a method that transforms data points from the source domain so that their distribution matches that of the target domain. This transformation enables us to use the labels from the source data to train models for the target domain. Our results show that our method is competitive when we have no labels available in the target domain.

When obtaining feature values is costly, it is common to use *feature selection*. Traditionally, feature selection involves choosing the same set of features for all data points, regardless of their individual properties. While this approach can reduce the overall cost of feature acquisition, it may not always be optimal, as different data points might benefit from observing different subsets of features. Therefore, in this thesis, we focus on methods that can adaptively acquire a subset of features for each data point. These methods aim to balance the trade-off between acquisition cost and informativeness. In this thesis, we review recent methods developed for this purpose.

Many existing approaches operate in an *offline* setting, where it is assumed that a training dataset is fully available to learn feature selection strategies. However, access to such a training set is not always feasible in practice. Consequently, we propose methods for an *online* setting, where no training dataset is available in advance. Instead, data points arrive sequentially as a stream, and feature selection must be performed on each received data point while simultaneously updating the models based on the information obtained at each time step.

In our online methods, we employ *active learning* at each time step to select the features of the data point received at that time step. We propose different formulations for this online problem, which enable us to theoretically analyze the performance of our methods. Moreover, we develop methods that are designed to operate in settings where the costs of obtaining features are not fixed and may be subject to uncertainty.

In the first part of this thesis, Chapter 2, we present explanations of key concepts that are used in the appended papers. Chapter 3 provides summaries of the appended papers, and Chapter 4 concludes the thesis.

In the second part of this thesis, we have appended the papers that I worked on during my PhD studies. These papers present methods for learning representations in settings with limited supervision, utilizing techniques from knowledge transfer and active learning:

- **Paper 1.** Demonstrates that leveraging knowledge from kernel and neural embeddings improves both optimization and generalization in neural networks.

- **Paper 2.** Proposes a novel framework for domain adaptation and demonstrates that it preserves class structure, thereby leading to improved performance.

- **Paper 3.** Provides a review of feature selection techniques that adaptively select features for each data point.

- **Paper 4.** Develops an online learning framework that employs active learning to select the most informative features to query, optimizing the trade-off between performance and acquisition cost.

- **Paper 5.** Reformulates the online learning problem introduced in Paper 4 to better exploit structural properties and provides a theoretical analysis based on the new formulation.

Chapter 2

Background

In this chapter, we discuss the main concepts and methods that are used throughout this thesis.

2.1 Knowledge Transfer

One of the main approaches used in this thesis to learn appropriate representations is knowledge transfer. In general, knowledge transfer means using information from another task or domain to improve the learning process in the task we are interested in, so that learning does not have to start from scratch (Niu et al. 2021). Knowledge transfer has been used in various application areas, including recommendation systems (W. Pan 2016), health care (Yu et al. 2022), and signal processing (Xie et al. 2016).

More formally, in a learning task T within a domain $\mathcal{D}$, we are given a dataset consisting of feature-label pairs $(\mathbf{x}_i, y_i)$. The goal is to use this dataset to learn a predictive function f such that $f(\mathbf{x}) = y$ for a new example $(\mathbf{x}, y)$. In knowledge transfer, we consider a source domain and task, denoted by $\mathcal{D}_s$ and T_s, and a target domain and task, denoted by $\mathcal{D}_t$ and T_t, where $\mathcal{D}_s \neq \mathcal{D}_t$ or $T_s \neq T_t$. The objective is to use the knowledge from $\mathcal{D}_s$ and T_s to improve the learning of the predictive function f in $\mathcal{D}_t$ (Lin and Jung 2017).

There are several research directions in knowledge transfer. One widely used approach is (partial) *fine-tuning*[1], which has gained significant popularity with the rise of large language models (LLMs); see, for example, (Houlsby et al. 2019). Another approach to knowledge transfer is to use representations learned from one domain in a target domain. This is sometimes referred to as *feature-based transfer learning*; see, for instance, (J. Pan 2010).

In this thesis, we focus on two specific approaches for knowledge transfer: *kernel methods* and *domain adaptation*. In the following sections, we review these two approaches and explain how they can be used for knowledge transfer.

[1]Fine-tuning means continuing the training of (parts of) a model that has already been trained (for example, a large neural network) using our own data.

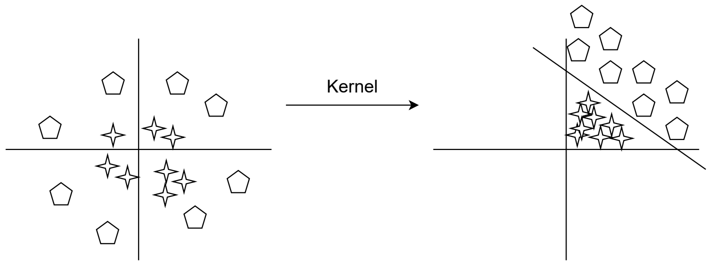

Figure 2.1: An example showing how mapping data to a new feature space can make it linearly separable. Image adapted from (Murphy 2022).

2.1.1 Kernel Methods

In machine learning, kernel methods are a set of techniques that help learning algorithms find patterns in complex input data. In simple words, kernel methods allow a learning algorithm to work *indirectly* in another (possibly higher-dimensional) feature space, which can make learning easier for certain tasks. For example, in a classification problem where the data is not linearly separable in the original feature space, applying a kernel may transform the data so that it becomes linearly separable in the new feature space (see Figure 2.1). Here, indirectly means that we do not need to explicitly compute the target feature vectors. Instead, we use a function called a *kernel function*. A kernel function measures the similarity between two input feature vectors as if they were mapped to the target space, without actually computing their coordinates in that space.

Now, we provide a more formal definition of kernels in machine learning. Suppose that in a learning problem (e.g., classification or regression), each feature vector $\mathbf{x}$ in the data set belongs to a feature space $\mathcal{X}$. As a concrete example, consider Ridge regression, where the input feature space is $\mathbb{R}^d$ with $d > 0$. In this case, the *hypothesis space*, that is, the set of functions that the algorithm can learn, is given by

$$\mathcal{H} = \left\{ \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle + b \;\middle|\; \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R} \right\},$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^d$, and $\mathbf{w}$ and b are parameters learned during training.

As mentioned before, sometimes it is beneficial to transform the input features to improve the learning procedure (e.g., to achieve linear separability). To formalize this idea, we introduce the concept of a *feature map*. A feature map is a function $\phi : \mathcal{X} \to \mathcal{F}$ that transforms the original input space $\mathcal{X}$ into a new feature space $\mathcal{F}$, which may have higher dimensionality. After applying the feature map, the learning algorithm operates on $\phi(\mathbf{x})$ instead of $\mathbf{x}$. Accordingly, the hypothesis space becomes

$$\mathcal{H}_\phi = \{ \mathbf{x} \mapsto \langle \mathbf{w}, \phi(\mathbf{x}) \rangle + b \mid \mathbf{w} \in \mathcal{F},\ b \in \mathbb{R} \}.$$

In many cases, however, explicitly computing the feature map $\phi(\mathbf{x})$ can be computationally expensive or even infeasible, especially if $\mathcal{F}$ is very high-dimensional or infinite-dimensional. To address this issue, a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is used. The kernel function satisfies

$$k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle,$$

for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. This approach allows algorithms to work with inner products in the feature space without explicitly computing the mapping ϕ. This is known as the *kernel trick*. Specifically, a learning algorithm can apply the kernel trick if the input features always appear inside an inner product. Such a learning algorithm is said to be *kernelized*.

In this thesis, we do *not* apply the kernel trick. Instead, we exploit the knowledge already available in kernels to construct new input features for training a neural network. The theoretical results on the optimization and generalization properties of neural networks in (Arora et al. 2019) demonstrate how these properties depend on the input features through a Gram matrix. Inspired by these results, we employ kernels to generate data representations that enhance both optimization and generalization in neural networks.

Since the kernels used in this thesis are associated with feature maps of infinite dimension, we approximate the feature representations. The first approximation method is *random Fourier features* (RFF) (Rahimi and Recht 2007), which constructs an explicit, low-dimensional and randomized feature map. The resulting inner product closely approximates the original kernel function. The second method is based on the Nyström approximation (Williams and Seeger 2000), which approximates the kernel matrix[2] computed over the data points. In this method, the data is used to construct a low-rank approximation of the full kernel matrix. The result is a matrix $\mathbf{\Phi}$ such that

$$K \approx \mathbf{\Phi}\mathbf{\Phi}^T,$$

where K is the original kernel matrix, and the rows of $\mathbf{\Phi}$ are approximate kernel feature representations.

In what follows, we briefly review the kernels used in this thesis.

Gaussian Kernel. The Gaussian kernel, which is also known as squared exponential kernel (SE) or RBF kernel, is one of the most commonly used kernels for real-valued input features (Murphy 2022). This kernel is defined by

$$k_{RBF}(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right),$$

where γ controls the shape of the kernel.

Arc-Cosine Kernel. The next type of kernel we use in this thesis is the arc-cosine kernel (Cho and Saul 2009). The n-th order *arc-cosine* kernel between two inputs x and x' is given by

$$k_n(x, x') = \frac{1}{\pi} \|x\|^n \|x'\|^n J_n(\theta),$$

[2]The kernel matrix is a symmetric positive semi-definite matrix where each entry is computed by applying the kernel function to a pair of data points.

where θ is the angle between the feature vectors x and x', and $J_n(\theta)$ is defined as

$$J_n(\theta) = (-1)^n (\sin\theta)^{2n+1} \left(\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\right)^n \left(\frac{\pi - \theta}{\sin\theta}\right).$$

In (Cho and Saul 2009), it is shown that this kernel models the behavior of a neural network with a single hidden layer and threshold activation functions.

Optimal Kernel. Finally, we use an *optimal* kernel proposed in (Cortes et al. 2012), which aligns the data representations with the label vector (a vector that contains the labels assigned to each data point). We use this kernel because theoretical results in (Arora et al. 2019) suggest that high alignment between input features and labels improves both optimization and generalization in neural networks. This kernel is based on the concept of *kernel-target alignment* (see, for example, (Cristianini et al. 2001)), and the work of (Cortes et al. 2012) proposes an algorithm to *learn* this kernel from a group of *base kernels*. This is achieved by formulating the learning problem as a Quadratic Programming (QP) problem.

2.1.2 Domain Adaptation

One of the main challenges in machine learning is that the data distribution can change between training and prediction stages. This issue is often called *data drift*. The way drift appears can vary depending on the application domain. For example, in computer vision, it can happen because of changes in the background or because different cameras are used for training and prediction (Courty et al. 2016). In healthcare, data drift might occur when a model trained on data from one hospital is used in another hospital, where patient populations or medical equipment may be different.

More precisely, in the data drift problem, the Probability Density Functions (PDFs) of the data are different during training and prediction. Because of this, domain adaptation focuses on finding ways to transfer knowledge from a source domain to a target domain where the source and target have different PDFs. In this thesis, we concentrate on *unsupervised domain adaptation*, where the aim is to use labeled data from the source domain together with unlabeled data from the target domain to train a predictor that performs well on the target domain (Z. Wang and Mao 2023; Courty et al. 2016). An example is shown in Figure 2.2.

Similar to (Courty et al. 2016), in this thesis, we study domain adaptation using *optimal transport*. In particular, optimal transport gives a transport map that moves the source data distribution to match the target data distribution. We use the transport map obtained from optimal transport to modify the source data samples so that they follow the target distribution. After that, we train a predictor using the transformed source data along with the original source labels.

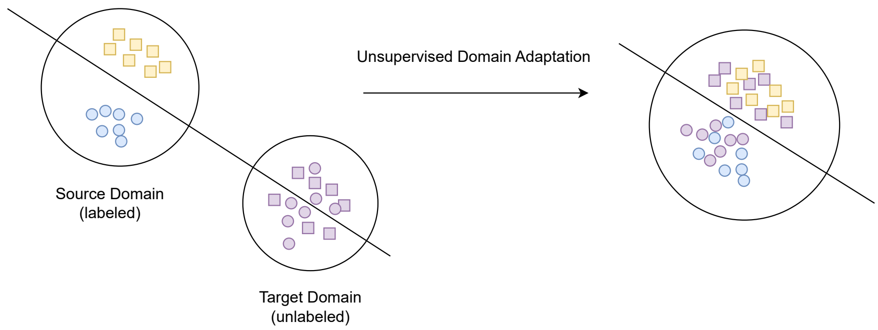

Figure 2.2: An illustration of unsupervised domain adaptation. A classifier trained solely on labeled data from the source domain performs poorly when applied to the target domain (left). In contrast, a classifier trained on domain-adapted source data achieves better performance. Image adapted from (Goel and Ganatra 2023).

2.2 Active Learning

Active learning is another approach we use in this thesis to deal with situations where we do not have enough labeled data. Often, we have a large set of data points without labels, and labeling them comes with a cost. For example, think about a medical diagnosis problem. In this case, we may have many patient profiles, but we do not know the correct diagnosis for each patient (that is, we do not know their true label). Getting the correct diagnosis is expensive because it requires the help of medical experts. Therefore, we need to carefully choose which patient profiles to label so that we gain the most useful information while keeping the labeling cost low. This is exactly where active learning becomes helpful.

Active learning aims to select the most informative data points to label, instead of labeling data randomly. By doing this, we can train models that reach high accuracy with fewer labeled examples. In medical diagnosis, for instance, active learning algorithms can prioritize labeling patient profiles that are uncertain or unusual, where a correct label would greatly improve the model's understanding. This strategy saves expert time and reduces overall costs, while still building effective models even with limited supervision.

There are two main types of active learning discussed in the literature. The first type is called *streaming active learning*. In this setting, data points arrive one by one, like a stream, and at each time step, the active learner decides whether to label the current data point or to skip it. For example, consider an email spam filter that receives emails in real time. As each email arrives, the system must decide whether to ask a human to label it as spam or not, or simply ignore it and move on. The second type is called *pool-based active learning*. Here, the learner has access to a large pool of unlabeled data points all at once and can choose which points to label. An example of this is having a database of medical images without diagnosis labels. The active learner can look at the entire collection and select the most uncertain or informative images to label first, helping to build a better diagnostic model more efficiently. For more information, refer to (Settles 2009).

We now give a more formal definition of active learning, following the description in (Golovin, Krause, and Ray 2010). In general, in active learning (both pool-based and streaming settings), the goal is to find the best *hypothesis* (such as a classifier or a scientific theory) by using a low-cost set of *tests* (such as requesting a label from an expert or conducting an experiment).

Formally, let $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ represent the set of all possible hypotheses. We are also given a set of available tests $T = \{1, 2, \ldots, N\}$. For instance, in a classification problem, each $h \in \mathcal{H}$ corresponds to a classifier (that is, a function), and each test $t \in T$ represents a data point for which we can request a label. Performing a test $t \in T$ comes with a cost $c(t)$, where $c : T \to \mathbb{R}^+$.

The goal in active learning is to find the true underlying hypothesis $h^* \in \mathcal{H}$ by selecting a set of tests $A \subseteq T$ such that the total cost $\sum_{t \in A} c(t)$ is as small as possible. In other words, active learning aims to find a policy π^* that chooses the next test to perform in a way that eventually identifies the true hypothesis while minimizing the total cost of the tests. The policy π^* is obtained by solving the following optimization problem:

$$\pi^* = \underset{\pi \text{ is feasible}}{\operatorname{argmin}} \; cost(\pi),$$

where $cost(\pi)$ denotes the expected total cost of the tests performed by the policy π, and the expectation is taken over the possible hypotheses. Here, we assume that there is a random variable H that represents the true underlying hypothesis we want to identify. Furthermore, we define a policy as *feasible* if it is guaranteed to find the true hypothesis, although it may not do so with the minimum possible cost.

Unfortunately, solving the optimization problem described above is computationally intractable. Even finding an approximate policy π that achieves $cost(\pi) \leq cost(\pi^*) \times o(\log n)$ is NP-hard (Chakaravarthy et al. 2007). In the noiseless version of the problem, where the outcome of each test is known deterministically given the true hypothesis, there are some heuristics, such as *information gain* and *generalized binary search*, that are shown to produce near-optimal policies. However, these heuristics can perform arbitrarily poorly when the setting is noisy. To address this, (Golovin, Krause, and Ray 2010) proposes the *equivalence class edge cutting* (EC^2) algorithm based on *adaptive submodularity* (Golovin and Krause 2011), which achieves near-optimal performance even in the presence of noise.

In short, EC^2 defines an objective function based on a graph, where each node represents a possible hypothesis and edges are weighted according to the probabilities of the connected hypotheses. The objective function takes as input a set of performed tests. Using this objective, EC^2 employs a greedy algorithm that selects the test with the highest gain. Here, the gain of a test is the expected increase in the objective function when the test is added to the current set of performed tests. In (Golovin, Krause, and Ray 2010), it is shown that the EC^2 objective is adaptive submodular, which means that the gain of a test does not increase when more tests are added to the set of performed tests. Due to adaptive submodularity, the EC^2 algorithm is guaranteed to have near-optimal performance. More details about EC^2 can be found in [Paper 4].

2.2.1 Feature Selection via Active Learning

In many real-world applications, we need to work with high-dimensional datasets that contain a large number of features. Obtaining the values of all features for every data point in order to make a prediction is often impractical or computationally expensive. Returning to the medical diagnosis example, collecting all possible test results for a patient to make a diagnosis (i.e., a prediction) is time-consuming and may cause unnecessary discomfort for the patient.

To address this issue, a line of research has focused on the sequential selection of features for each data point (e.g., a patient), aiming to maintain high prediction accuracy while minimizing the acquisition cost. This approach is known as *dynamic feature selection* (DFA), and is also referred to as *active feature acquisition* (AFA) or *active feature selection*. Note that AFA is different from traditional feature selection methods, where a static subset of features is selected once and used for *all* data points.

AFA can be viewed as a special case of the general active learning problem discussed earlier. Specifically, in AFA, each feature corresponds to a test, and the cost of the test is the cost of acquiring that feature. In this setting, the goal is to predict a label for each data point, and the set of all possible labels corresponds to the set of all possible hypotheses. It is important to note that in a general label prediction problem, we are in a noisy setting, since the value of a feature cannot necessarily be determined based on the true label.

In this thesis, we review recent methods developed for the AFA problem. In summary, there are three main approaches to address AFA, and the key differences between them lie in how they design the feature selection policy. The main approaches can be summarized as follows:

- **Greedy Active Feature Acquisition.** These approaches define an objective function that measures the informativeness of a feature with respect to the prediction task. For each data point, the goal is to optimize this objective function using a greedy strategy, selecting the next most informative feature at each step.

- **Embedded Methods for Active Feature Acquisition.** Another important line of work integrates feature selection directly into the learning algorithm. In these methods, the feature acquisition process is embedded within the prediction model, allowing the algorithm to actively decide which features to acquire at each step based on their estimated contribution to improving predictive performance.

- **MDP-Based Active Feature Acquisition.** Many existing methods suffer from the limitation of relying on greedy, local informativeness criteria, which can lead to myopic decisions and suboptimal long-term outcomes. To overcome this, some approaches model the AFA problem as a Markov Decision Process (MDP). This allows them to leverage Reinforcement Learning (RL) to learn non-myopic policies that optimize feature acquisition over a sequence of decisions, leading to better long-term predictive performance.

An important example of greedy methods for AFA is the EC^2 algorithm, which we introduced earlier. In this thesis, we propose a framework for learning from a stream of data points, where at each time step we use the EC^2 algorithm to actively select a cost-efficient sequence of features for the data point received at that time. Furthermore, we model this framework as a *combinatorial multi-armed bandit* problem (see Section 2.3.3), which enables us to provide a rigorous theoretical analysis of the framework's performance.

2.3 Online Learning

In online learning, the goal is to learn from a sequence of data points that arrive over time in a streaming fashion. At each time step, the learner receives a new data point and is required to make a prediction or take an action (e.g., assigning a label in a classification task). After that, the learner receives some form of feedback about the correctness of its decision. This feedback may not always be the exact correct answer, but it provides a signal to guide future predictions (Shalev-Shwartz et al. 2012). Over time, the learner aims to improve its performance by continuously adapting to the incoming data stream.

2.3.1 General Online Learning

We now present a formal description of the general online learning problem, adapted from (Cesa-Bianchi and Lugosi 2006; Shalev-Shwartz et al. 2012). At each time step $t \in [T]$, the learner receives an input instance $\mathbf{x}_t$ and must make a prediction (or take an action) $\hat{y}_t$ from a set of possible predictions $\mathcal{Y}$. After the prediction is made, the environment provides feedback in the form of loss values $l_t(y)$ for all $y \in \mathcal{Y}$. In some cases, this feedback may come as the true label or outcome instead, from which the losses can be computed. The learner then incurs the loss associated with its prediction, namely $l_t(\hat{y}_t)$. This interaction protocol is outlined in Algorithm 1.

Algorithm 1 Online Learning Protocol

for $t = 1$ to T **do**
  Learner receives input $\mathbf{x}_t$
  Learner makes prediction $\hat{y}_t \in \mathcal{Y}$
  Environment reveals loss values $l_t(y)$ for all $y \in \mathcal{Y}$
  Learner suffers loss $l_t(\hat{y}_t)$
end for

In the online learning literature, it is common to assume the presence of a set of N distinct *experts*, where each expert $i \in [N]$ provides a prediction $\hat{y}_t^i$ at each time step t. The *regret* with respect to expert i is defined as:

$$Regret_i = \sum_{t=1}^{T} \Big(l_t(\hat{y}_t) - l_t(\hat{y}_t^i) \Big)$$

In general, the goal in online learning is to maintain low regret relative to all experts. That is, we desire that

$$\lim_{T\to\infty} \max_{i\in[N]} \frac{Regret_i}{T} = 0.$$

This ensures that, in the long run, the learner performs as well as the best expert in hindsight.

2.3.2 Multi-Armed Bandits

In the general online learning setting described in Section 2.3.1, the learner receives the losses for all possible predictions after making its own prediction at each time step. However, this level of feedback is often unrealistic in real-world scenarios. Moreover, we make no assumptions about how the losses are generated. These losses can be chosen by an *adversary* aiming to increase the learner's regret. These considerations motivate special cases of the general online learning problem. In this section, we focus on an important variant known as the *multi-armed bandit* (MAB) problem which was first studied in (Thompson 1933).

In the basic form of the MAB problem, the learner does not receive a data point at each time step. Equivalently, we can think of the input as being the same across all time steps. At each time step t, the learner selects an *arm* $a_t \in \mathcal{A}$, where $\mathcal{A}$ is the set of all available arms (or actions). After the selection, the environment reveals the reward (or loss) associated *only* with the chosen arm, denoted by $r_t(a_t)$. The learner receives this feedback but gains no information about the rewards of the other arms. The interaction protocol is described in Algorithm 2.

Algorithm 2 Multi-Armed Bandit Protocol

```
1: for t = 1 to T do
2:     Learner selects arm a_t ∈ A
3:     Environment reveals reward r_t(a_t)
4:     Learner observes reward r_t(a_t)
5: end for
```

Without any assumption on how the environment generates the rewards for the arms, the MAB problem is referred to as the *adversarial* bandit problem (see, for example, (Lattimore and Szepesvári 2020; Slivkins 2019)). Here, we focus on the *stochastic* bandit problem. In the stochastic setting, each arm $a \in \mathcal{A}$ is associated with a fixed distribution D_a, and the environment draws the reward $r_t(a_t)$ from D_{a_t} whenever arm a_t is selected. That is, for each arm, the rewards are i.i.d. samples from a fixed distribution that is unknown to the learner (Slivkins 2019).

To evaluate and compare different arm selection algorithms, it is common to use the notion of regret, similar to the definition in Section 2.3.1. Let $\mu_a = \mathbb{E}[r_t(a)]$ denote the expected reward of arm a, and let $\mu^* = \max_{a\in\mathcal{A}} \mu_a$ be the highest expected reward among all arms. The regret up to time T is defined as:

$$Regret_T = T \cdot \mu^* - \sum_{t=1}^{T} \mu_{a_t}.$$

This quantity measures how much less reward the algorithm accumulates compared to always selecting the best arm. Since there is randomness in the choice of arm a_t, we are interested in the expected regret, $\mathbb{E}[Regret_T]$ (Slivkins 2019).

In stochastic bandit problems, a key challenge is the trade-off between *exploration* and *exploitation* under uncertainty about the rewards of actions (arms). Exploration involves trying out different arms to gather information about their rewards, while exploitation means choosing the arm that currently seems best in order to gain the highest reward. For instance, in a recommendation system where each item is an arm, exploration would involve showing a user a new item to learn their preferences, and exploitation would mean recommending an item already known to match their interests. Bandit algorithms are designed to manage this trade-off effectively. In the following, we briefly review two well-known algorithms for arm selection in stochastic bandits.

Upper Confidence Bound. The first algorithm we present is based on the principle of *optimism in the face of uncertainty*. According to this principle, the algorithm should act as if the environment will give the best possible outcome (Lattimore and Szepesvári 2020). Algorithms that follow this principle are called *upper confidence bound* (UCB) algorithms. Here, we focus on a specific version called UCB1, introduced in (Auer et al. 2002).

In UCB1, at each time step t, the algorithm computes an upper confidence bound for each arm $a \in \mathcal{A}$ as follows:

$$UCB_{t,a} = \hat{\mu}_{t,a} + \sqrt{\frac{2\ln(t)}{n_{t,a}}},$$

where $\hat{\mu}_{t,a}$ is the average reward obtained from arm a before time t, and $n_{t,a}$ is the number of times arm a has been chosen before time t. The algorithm starts by selecting each arm once. After that, at each time step, it selects the arm with the highest $UCB_{t,a}$. The full procedure is given in Algorithm 3.

Algorithm 3 UCB1

```
for all a ∈ A do
    Play arm a once
end for
for t = 1 to T do
    For each arm a, compute UCB_{t,a} = μ̂_{t,a} + sqrt(2 ln(t) / n_{t,a})
    Select arm a_t = argmax_a UCB_{t,a} and observe reward r_t(a_t)
    Update μ̂_{t,a_t} and n_{t,a_t}
end for
```

Intuitively, in UCB1 an arm a is selected either because it has a high average reward $\hat{\mu}_{t,a}$ or because it has been played only a few times, that is, it has a low $n_{t,a}$. Choosing an arm with a high average reward means we are exploiting what we have already learned. On the other hand, choosing an arm with a low $n_{t,a}$ helps us explore less familiar options. In this way, the algorithm balances exploration and exploitation (Slivkins 2019).

Thompson Sampling. The second algorithm we present assumes some prior knowledge about the reward distributions of the arms. This prior knowledge is in the form of prior distributions over the parameters of the reward distributions, such as their means. The algorithm is known as *Thompson sampling* and was originally introduced in (Thompson 1933). It is also referred to as *posterior sampling* in the literature.

At each time step, Thompson Sampling draws a sample of the reward distribution parameter for each arm from its posterior distribution. It then selects the arm with the highest expected reward under the sampled parameters. After observing the reward, the posterior distribution of the selected arm is updated based on the new data. A sketch of the Thompson sampling algorithm is given in Algorithm 4.

Algorithm 4 Thompson Sampling

```
Input: Prior distributions over parameters
for t = 1 to T do
    for each arm a ∈ 𝒜 do
        Sample parameter θ_{t,a} from the posterior of arm a
    end for
    Select arm a_t = argmax_a 𝔼[r_t(a) | θ_{t,a}]
    Play arm a_t and observe reward r_t(a_t)
    Update the posterior distribution of arm a_t using r_t(a_t)
end for
```

2.3.3 Combinatorial Multi-Armed Bandits

In this section, we briefly describe an extension of the MAB problem from Section 2.3.2, known as the *combinatorial multi-armed bandit* (CMAB) problem (Cesa-Bianchi and Lugosi 2012). In the CMAB setting, instead of selecting a single arm at each time step, a *set* of arms is selected. This set is referred to as a *super arm*. After playing the super arm, the environment reveals the reward for that set. In addition, the environment may provide some feedback about the individual arms, known as *base arms*, that are part of the super arm. This type of feedback is called *semi-bandit* feedback.

We now give a formal definition of the CMAB problem, adapted from (Chen et al. 2013). Let $S \subseteq 2^{\mathcal{A}}$ be the set of all possible super arms. At each time step t, the algorithm selects a super arm $s_t \in S$. The environment then reveals the reward for the selected super arm, denoted by $R_t(s_t)$. In the semi-bandit feedback setting, the environment also provides feedback (such as rewards) for each base arm $a \in s_t$. The total reward of the super arm depends on the rewards of its base arms. It is usually assumed that the expected reward of the super arm s, $\mathbb{E}[R_t(s)]$, can be written as a function of the expected rewards of the base arms in s. That is, there exists a function f such that

$$f(\boldsymbol{\mu}, s) = \mathbb{E}[R_t(s)],$$

where $\boldsymbol{\mu}$ is the vector of expected rewards for all base arms.

In the literature, it is commonly assumed that there is access to an oracle that can find the optimal super arm s^* given a reward vector $\boldsymbol{\gamma}$ such that

$$f(\boldsymbol{\gamma}, s^*) = \max_{s \in S} f(\boldsymbol{\gamma}, s).$$

The algorithm that selects super arms does not know the true expected reward vector $\boldsymbol{\mu}$. Therefore, similar to the regret definition in Section 2.3.2, the regret in CMAB problems is defined using the reward of the best super arm under $\boldsymbol{\mu}$. This optimal super arm is given by

$$s^*_{\boldsymbol{\mu}} = \operatorname*{argmax}_{s \in S} f(\boldsymbol{\mu}, s).$$

The algorithms discussed in Section 2.3.2 have been extended to the CMAB setting. For instance, see (Chen et al. 2013) for an extension of the UCB algorithm, and (S. Wang and Chen 2018) for an extension of Thompson Sampling.

Chapter 3

Summary of Included Papers

As previously mentioned, this thesis focuses on two different approaches to learning data representations when sufficient supervision is not available. [Paper 1] and [Paper 2] explore the first approach, which involves using knowledge transfer for representation learning. In [Paper 3] - [Paper 5], we examine how active learning can be used to learn representations in the absence of sufficient supervision. Figure 3.1 illustrates the relationships between these papers (the underlying concepts will be discussed in more detail later).

In the following, we briefly summarize the contributions of each paper included in this thesis.

3.1 Paper 1

In [Paper 1], we experimentally explore how the knowledge embedded in kernel and neural representations can be used to improve both optimization and generalization

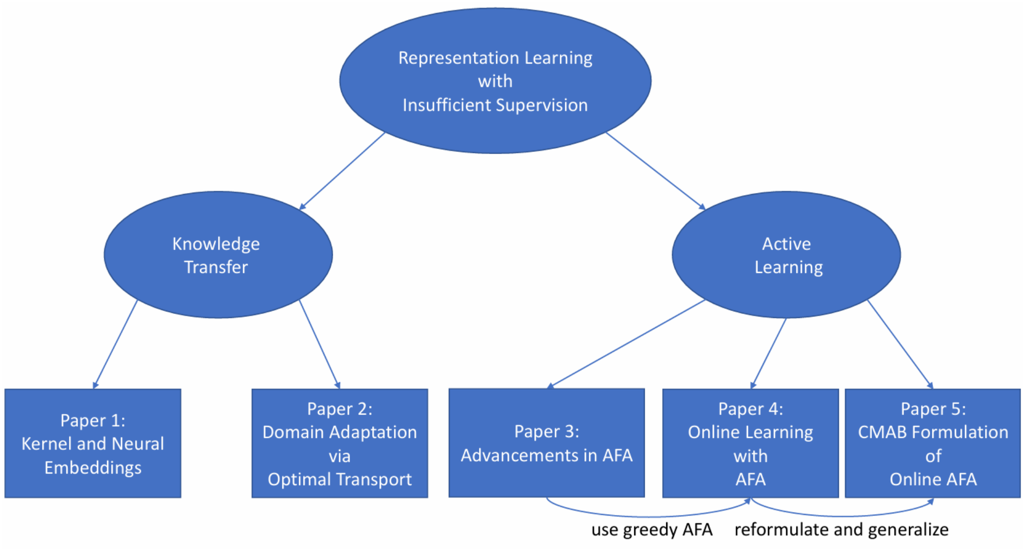

Figure 3.1: An overview of the papers and their contributions within the two main research directions.

in wide neural networks. Our studies are motivated by the theoretical findings of (Arora et al. 2019), which, as mentioned earlier, explain why training with true labels leads to faster convergence compared to random labels. Specifically, their results show that true labels have stronger alignment with the eigenvectors of a Gram matrix. In other words, they demonstrate that the relationship between data representations and true labels plays a key role in achieving better convergence (and generalization) during neural network training. Building on these insights, we carefully select a set of kernels to construct data representations that are more aligned with the labels.

Among the selected kernels, we achieve the best results in both optimization and generalization using an optimal kernel that we learn from a set of base kernels. We use the algorithm proposed in (Cortes et al. 2012) to learn this kernel, which ensures optimal alignment with the labels. This alignment is the main reason we observe superior performance with this kernel. We can interpret the other kernels as approximations of this optimal kernel.

3.2 Paper 2

In [Paper 2], we introduce a novel framework for optimal transport, where the goal is to learn a transport map from a source domain to a target domain while minimizing a transport cost. A common assumption in the literature is that the optimal transport map preserves the underlying structure in the source and target domains. In the context of domain adaptation, this typically refers to an implicit class structure in both domains. We adopt the same assumption in our framework and provide a theoretical analysis that characterizes conditions on this class structure under which the transport map can be computed.

Our optimal transport framework provides a way to map each class in the source domain to its corresponding class in the target domain. The main difficulty lies in the fact that the class components are hidden. To address this, our framework operates in two stages: first, we recover the association between components across domains; once this association is recovered, we apply standard optimal transport separately to each component pair. In the paper, we formally analyze the conditions under which the first step can be done efficiently.

Our experimental results demonstrate that our proposed framework performs competitively in the setting of unsupervised domain adaptation where we do not have access to data labels in the target domain.

3.3 Paper 3

In [Paper 3], we present a comprehensive survey of AFA strategies (see Section 2.2.1). We begin by providing a general formulation of the AFA problem, followed by a structured overview of recent approaches. In particular, we examine three main categories of AFA methods and highlight their key differences.

First, we examine greedy methods, which acquire features based on an objective function that evaluates the informativeness of each feature for the prediction task.

We focus on two widely used objectives: conditional mutual information (CMI) and equivalence class edge cuts (EC^2). Within CMI-based methods, we distinguish between *generative approaches*, which employ models such as variational autoencoders (VAEs) to estimate CMI values, and *discriminative approaches*, which either directly predict the next most informative feature or approximate CMI values. While generative methods offer greater modeling flexibility, they are generally harder to train, which has contributed to recent progress in discriminative techniques.

Second, we review embedded methods, where the feature acquisition strategy is incorporated into the learning algorithm itself. These methods jointly optimize for both the predictive task and the feature acquisition process. We discuss a range of such approaches, including tree-based models (e.g., decision trees and random forests), Naïve Bayes classifiers, and margin-based methods.

Third, we explore MDP-based methods, which frame AFA as an MDP to better reflect its inherently sequential nature. This formulation allows the use of reinforcement learning to learn optimal feature acquisition policies. Methods in this category differ in how they define state transitions, model reward functions, and apply various reinforcement learning algorithms.

Beyond this taxonomy, we also highlight key practical challenges in AFA and identify open problems and directions for future research.

3.4 Paper 4

In [Paper 4], we propose a framework for learning a classifier from a stream of data points with initially unknown features. Our method integrates an active planning oracle into an online learning scheme. At each time step, the oracle performs AFA on the incoming data point by sequentially selecting features until sufficient information is gathered for prediction. The framework is flexible and supports different oracle strategies. For example, in our experiments, we evaluate CMI (also known as information gain), EC^2, and a few other baselines.

To minimize regret in the online setting, we employ posterior sampling (see Section 2.3.2), relying solely on prior knowledge about the data distribution, which is gradually refined through interactions with the environment. Specifically, we assume access to prior distributions over the parameters of feature distributions conditioned on labels. At each time step, we sample these parameters from the current posterior distributions and apply the oracle based on the sampled parameters.

Our empirical results show that the EC^2 oracle achieves the best trade-off between acquisition cost and prediction accuracy, with performance comparable to CMI. These findings align with the theoretical results of (Golovin, Krause, and Ray 2010).

We also introduce several algorithmic contributions. To support real-valued features, we extend the original EC^2 algorithm, which was originally designed for discrete settings. Furthermore, we present a variant of our framework that adapts to concept drift, allowing it to handle changes in the relationship between features and labels over time. We achieve this by applying *non-stationary* posterior sampling (Russo et al. 2018) to our framework.

Finally, we provide a theoretical analysis of our framework, establishing both prior-independent and prior-dependent regret bounds.

3.5 Paper 5

In [Paper 5], we build upon the framework introduced in our previous work in [Paper 4], where we formulated the problem as a partially observable Markov decision process (POMDP) and provided a theoretical analysis based on that formulation. Although the POMDP framework offers a principled approach, it leads to a regret bound that is *exponential* in the number of features.

To address this limitation, in [Paper 5], we reformulate the problem as a CMAB (see Section 2.3.3). This new formulation enables us to tackle a more general online decision-making problem, where the objective is to make cost-efficient decisions by sequentially performing tests on incoming data points. Unlike [Paper 4], which assumes deterministic costs for each test (i.e., feature), we allow for *stochastic* test costs that may depend on the test outcomes or the correct decision.

Modeling the problem as a CMAB requires a careful definition of base arms, super arms, and reward functions. We adapt standard algorithms such as UCB and Thompson Sampling to our setting and extend cost-efficient information acquisition methods like EC^2 to handle stochastic test costs. This CMAB-based approach allows us to derive a regret bound that is *linear* in the number of tests.

Chapter 4

Concluding Remarks and Future Work

In this thesis, we study how to learn appropriate representations when we do not have enough supervision for the task of interest. This lack of supervision can mean either missing labels for data points or missing feature values. We tackle this problem using two main approaches: knowledge transfer and active learning.

Knowledge Transfer. From the knowledge transfer perspective, our goal is to use existing knowledge to help learning in the domain or task of interest. We consider two sources of such knowledge in this thesis: (i) information captured in (learned) kernel and neural embeddings, and (ii) labeled data from a different but related source domain.

In the first case, we explore how kernel and neural embeddings can be used to improve the training of neural networks. Our experiments show that these embeddings not only speed up training but also lead to better generalization on unseen data. Combined with theoretical work connecting neural networks to kernel methods, our findings suggest that it may be possible to extend existing theory to deeper neural architectures.

In the second case, we focus on the challenging problem of unsupervised domain adaptation, where we aim to train a model for a target domain that has no labeled data, using a labeled source data set. We propose a new method for optimal transport, which allows us to shift the source data distribution to better match the target distribution. By learning a transport plan that aligns the source and target domains, we can leverage the labeled source data to train a model that performs well on the target domain. Our method also preserves the class structure during transport, meaning that examples from the same class in the source are mapped to their corresponding class in the target. Our experiments show that our optimal transport approach is effective for real-world domain adaptation tasks across several data sets.

Active Learning. From the active learning perspective, we consider the setting where the feature values of data points are not fully available. This scenario is highly relevant in practice, especially when acquiring feature values is expensive or time-consuming.

We review recent advances in methods that actively select which features to acquire for each data point. We provide a comparative study of greedy approaches and MDP-based methods for this task.

We then consider an online version of the problem, where data points arrive sequentially and we have to learn a policy to acquire feature values for each incoming data point. This setting is more challenging than the offline case, as decisions must be made on the fly without access to a training data set. We propose a new online learning framework designed to make predictions on a stream of data. For each incoming data point, our framework selects a cost-efficient sequence of features to acquire before making a prediction. After the prediction is made, the true label is revealed, and we update our knowledge using both the acquired features and the observed label. Our approach for updating the knowledge is based on a variant of Thompson sampling, tailored to handle the challenge of learning in this online setting. We also introduce a method to handle concept drift in non-stationary environments. Our experiments demonstrate that the proposed framework achieves accurate predictions while keeping feature acquisition costs low.

We first formulate the above problem as a POMDP, which enables us to theoretically analyze the performance of our framework and derive bounds on its regret. However, since the results for POMDPs depend on the number of states, the derived bounds are exponential in the number of features. Thus, we also provide a CMAB formulation, which enables us to derive a tighter bound on the regret. The CMAB formulation also allows us to solve a more general version of the problem where the cost of features can be stochastic. For this version, we propose weighted versions of active feature acquisition methods to handle stochastic costs.

An important direction for future research is to extend our online learning framework to settings with more limited feedback, where the true label may not be available at the end of each time step. Another promising direction is to incorporate function approximation into the active learning component. For example, one could design a deep neural network to approximate the gains of different features, such as the gain defined by the EC^2 objective, which satisfies adaptive submodularity and provides theoretical guarantees on acquisition cost.

Our theoretical analysis is based on Thompson sampling, but experiments show that alternatives like BayesUCB can also perform well in terms of both prediction accuracy and cost. This suggests that future work could explore a theoretical analysis of BayesUCB in our setting.

Bibliography

Arora, Sanjeev, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang (2019). "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks". In: *International Conference on Machine Learning*. PMLR, pp. 322–332 (cit. on pp. 9, 10, 20).

Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer (2002). "Finite-time analysis of the multiarmed bandit problem". In: *Machine Learning* 47, pp. 235–256 (cit. on p. 16).

Cesa-Bianchi, Nicolo and Gábor Lugosi (2006). *Prediction, learning, and games*. Cambridge University Press (cit. on p. 14).

Cesa-Bianchi, Nicolo and Gábor Lugosi (2012). "Combinatorial bandits". In: *Journal of Computer and System Sciences* 78.5, pp. 1404–1422 (cit. on p. 17).

Chakaravarthy, Venkatesan T, Vinayaka Pandit, Sambuddha Roy, Pranjal Awasthi, and Mukesh Mohania (2007). "Decision trees for entity identification: Approximation algorithms and hardness results". In: *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 53–62 (cit. on p. 12).

Chen, Wei, Yajun Wang, and Yang Yuan (2013). "Combinatorial multi-armed bandit: General framework and applications". In: *International Conference on Machine Learning*. PMLR, pp. 151–159 (cit. on pp. 17, 18).

Cho, Youngmin and Lawrence K. Saul (2009). "Kernel Methods for Deep Learning". In: *Advances in Neural Information Processing Systems 22*. Ed. by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta. Curran Associates, Inc., pp. 342–350 (cit. on pp. 9, 10).

Cortes, Corinna, Mehryar Mohri, and Afshin Rostamizadeh (2012). "Algorithms for learning kernels based on centered alignment". In: *The Journal of Machine Learning Research* 13.1, pp. 795–828 (cit. on pp. 10, 20).

Courty, Nicolas, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy (2016). "Optimal transport for domain adaptation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.9, pp. 1853–1865 (cit. on p. 10).

Cristianini, Nello, John Shawe-Taylor, Andre Elisseeff, and Jaz Kandola (2001). "On kernel-target alignment". In: *Advances in Neural Information Processing Systems* 14 (cit. on p. 10).

Goel, Parth and Amit Ganatra (2023). "Unsupervised domain adaptation for image classification and object detection using guided transfer learning approach and JS divergence". In: *Sensors* 23.9, p. 4436 (cit. on p. 11).

Golovin, Daniel and Andreas Krause (2011). “Adaptive submodularity: Theory and applications in active learning and stochastic optimization”. In: *Journal of Artificial Intelligence Research* 42, pp. 427–486 (cit. on p. 12).

Golovin, Daniel, Andreas Krause, and Debajyoti Ray (2010). “Near-optimal bayesian active learning with noisy observations”. In: *Advances in Neural Information Processing Systems* 23 (cit. on pp. 12, 21).

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly (2019). “Parameter-efficient transfer learning for NLP”. In: *International Conference on Machine Learning*. PMLR, pp. 2790–2799 (cit. on p. 7).

Lattimore, Tor and Csaba Szepesvári (2020). *Bandit algorithms*. Cambridge University Press (cit. on pp. 15, 16).

Lin, Yuan-Pin and Tzyy-Ping Jung (2017). “Improving EEG-based emotion classification using conditional transfer learning”. In: *Frontiers in Human Neuroscience* 11, p. 334 (cit. on p. 7).

Murphy, Kevin P. (2022). *Probabilistic Machine Learning: An introduction*. MIT Press (cit. on pp. 8, 9).

Niu, Shuteng, Yongxin Liu, Jian Wang, and Houbing Song (2021). “A decade survey of transfer learning (2010–2020)”. In: *IEEE Transactions on Artificial Intelligence* 1.2, pp. 151–166 (cit. on p. 7).

Pan, Jialin (2010). *Feature-based transfer learning with real-world applications*. Hong Kong University of Science and Technology (Hong Kong) (cit. on p. 7).

Pan, Weike (2016). “A survey of transfer learning for collaborative recommendation with auxiliary data”. In: *Neurocomputing* 177, pp. 447–453 (cit. on p. 7).

Rahimi, Ali and Benjamin Recht (2007). “Random Features for Large-Scale Kernel Machines”. In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20. Curran Associates, Inc. (cit. on p. 9).

Russo, Daniel J, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. (2018). “A tutorial on thompson sampling”. In: *Foundations and Trends® in Machine Learning* 11.1, pp. 1–96 (cit. on p. 21).

Settles, Burr (2009). *Active learning literature survey*. Technical Report 1648. University of Wisconsin-Madison, Department of Computer Sciences (cit. on p. 11).

Shalev-Shwartz, Shai et al. (2012). “Online learning and online convex optimization”. In: *Foundations and Trends® in Machine Learning* 4.2, pp. 107–194 (cit. on p. 14).

Slivkins, Aleksandrs (2019). “Introduction to Multi-Armed Bandits”. In: *Foundations and Trends® in Machine Learning* 12, pp. 1–286 (cit. on pp. 15–17).

Thompson, William R (1933). “On the likelihood that one unknown probability exceeds another in view of the evidence of two samples”. In: *Biometrika* 25.3/4, pp. 285–294 (cit. on pp. 15, 17).

Wang, Siwei and Wei Chen (2018). “Thompson sampling for combinatorial semi-bandits”. In: *International Conference on Machine Learning*. PMLR, pp. 5114–5122 (cit. on p. 18).

Wang, Ziqiao and Yongyi Mao (2023). "Information-Theoretic Analysis of Unsupervised Domain Adaptation". In: *The Eleventh International Conference on Learning Representations* (cit. on p. 10).

Williams, Christopher and Matthias Seeger (2000). "Using the Nyström Method to Speed Up Kernel Machines". In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press (cit. on p. 9).

Xie, Michael, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon (2016). "Transfer learning from deep features for remote sensing and poverty mapping". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1 (cit. on p. 7).

Yu, Xiang, Jian Wang, Qing-Qi Hong, Raja Teku, Shui-Hua Wang, and Yu-Dong Zhang (2022). "Transfer learning for medical images analyses: A survey". In: *Neurocomputing* 489, pp. 230–254 (cit. on p. 7).