Goal-oriented Modeling for Data-driven Decision Making

データ駆動型意思決定のための目的指向モデリング

by

Akira Tanimoto

谷本 啓

A Doctor Thesis

博士論文

Submitted to

the Graduate School of Kyoto University

on September 3rd, 2021

in Partial Fulfillment of the Requirements

for the Degree of Doctor of Informatics

Thesis Supervisor: Professor Hisashi Kashima　鹿島 久嗣

ABSTRACT

Machine learning-based systems are rapidly implemented in broad areas. Notably, predictive analytics has penetrated a variety of decision-making situations in the enterprise. Automated machine learning (AutoML) technologies have reduced the burden of the modeling process, such as the feature (predictor) generation, which has further accelerated this movement. However, while AutoML has its focus on maximizing a given metric of prediction performance, selecting the evaluation metric and linking the learned models to decision-making remain to be the work of human expert analysts. We cannot discuss the goodness of a choice of evaluation metric on its own but need to consider how it leads to the quality of subsequent decision-making. For further advancement of machine learning towards automated decision-making support beyond prediction, we propose directly modeling the utility of each action. This framework is related to offline reinforcement learning or causal inference. Compared to typical application domains of these approaches, real-world business decision-making problems often have (1) a vast (combinatorial) action space and (2) scarce supervision. We examine the challenges posed by these differences with real-world examples and discuss how to deal with them.

The first challenge is (1-1) computational complexity due to large combinatorial action spaces. In sequential decision-making settings, the outcome is evaluated with a time delay, and the utility of action in each time step should be evaluated with the action at the next time step being optimized. When the action space is combinatorial, the computational complexity of action optimization in each learning iteration would be a barrier to taking the utility modeling approach. However, the dependency of each dimension of action sometimes has a desirable property of locality. We discuss how to utilize this property by designing the utility function and optimization procedure with a realistic example of road infrastructure maintenance planning.

The second challenge is (2) the sample-efficiency. The outcome is often scarce, and thus supervision might be weak in direct utility modeling. In predictive modeling, on the other hand, expert analysts can isolate a prediction subproblem out of whole decision-making so that the supervision would be rich. We investigate how we can incorporate such intermediate supervision while our model directly predicts the outcome. One of the simplest examples is the imbalanced classification with numerical labels. The final prediction target is a binary label with class imbalanced, i.e., the positive label is scarce, but we can utilize the numerical labels of how likely each instance was to be positive.

Finally, we discuss (1-2) the biased sampling problem out of vast action space. The supervision of the outcome is only for the actual action that the past decision-makers have taken, and other potential outcomes of counterfactual actions are missing in general. Nonetheless, the model is expected to evaluate the utilities of all possible actions, including rare actions such as prescribing strong medicine to healthy people. Since such strong medicine tends to be prescribed to unhealthy patients, the prognosis for those prescribed such treatment may be poor despite the effects of the medicine. Typical supervised learning methods can be misled by such a spurious correlation. We reformulate causal effect inference as a decision-making problem and extend it into larger action spaces. We also apply it to a combinatorial (set-wise) recommendation problem.

Acknowledgements

I would like to thank the following people who have helped me undertake this research project.

First, I am extremely grateful to my supervisor Dr. Hisashi Kashima, for his valuable advice and encouragement. The meetings and conversations were vital for directing this project and improving its presentation. I would like to express my deepest appreciation to Dr. Akihiro Yamamoto and Dr. Hidetoshi Shimodaira, the referee of this thesis, for the thorough review and helpful comments especially to improve the organization of this thesis.

I also wish to thank Dr. Takehisa Yairi and Dr. Akira Iwasaki, the advisors in my master course in the Department of Aeronautics and Astronautics in the University of Tokyo, for helping me take my first steps as a researcher.

I would like to thank the co-authors of my earlier articles, Dr. Takashi Takenouchi, Dr. Masashi Sugiyama, Mr. So Yamada, Dr. Tomoya Sakai, and Dr. Shogo Hayashi, for their valuable discussion, worthy ideas, contributions to the articles, and other helpful support.

I would also like to extend my gratitude to my managers in NEC Corporation, Mr. Yosuke Motohashi, Dr. Ryohei Fujimaki, Dr. Satoshi Morinaga, Mr. Yoshio Kameda, Mr. Junji Sakai, Mr. Shinji Nakadai, Mr. Norio Yanagi, and Mr. Atsushi Kashitani, for giving me the opportunity, various support, and encouragement for my working on this doctoral thesis. I also would like to thank NEC Corporation for the financial support and other various support for the doctoral course.

Finally, I would like to thank my family for their devotion and support over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the rise of Big Data technologies, various kinds of data are being observed, stored, and analyzed for decision-making in governments and enterprises [Chen and Zhang, 2014]. Machine learning-based modeling technology is at the core of this. Among its diverse objectives, predicting future outcomes, called predictive analytics, is one of its main approaches [Gandomi and Haider, 2015]. It has been applied to predict equipment failures, advertisement clicks, product demand, and others. While predictive analytics has an extensive range of applications and is achieving a certain level of success, it requires special skills to handle. The lack of human resources to handle it, called data scientists, has become a bottleneck. Automated machine learning technology (AutoML) [Yao et al., 2018, He et al., 2021] and neural architecture search (NAS) [Elsken et al., 2019] have reduced the burden of data science so that experts in each application domain can easily handle analytics.

However, designing an ML problem remains an essential human task, i.e., choosing the target variable, loss function, and evaluation metrics, which is not a straightforward process. In order to properly choose an evaluation metric or loss function, we need to look beyond the problem setting of prediction—how the learned model is utilized. Fig. 1.1 shows a typical pipeline of analytics. Trained models can only have an impact by improving the quality of decision-making. Therefore, it is vital to align the loss and metric for training with utility, the performance measure of decision.

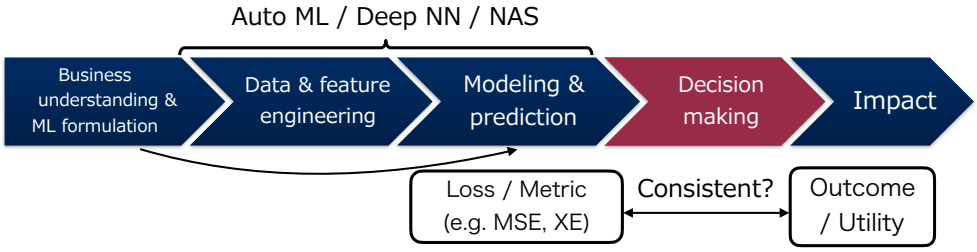


Figure 1.1: An overview of modeling and decision-making pipeline.

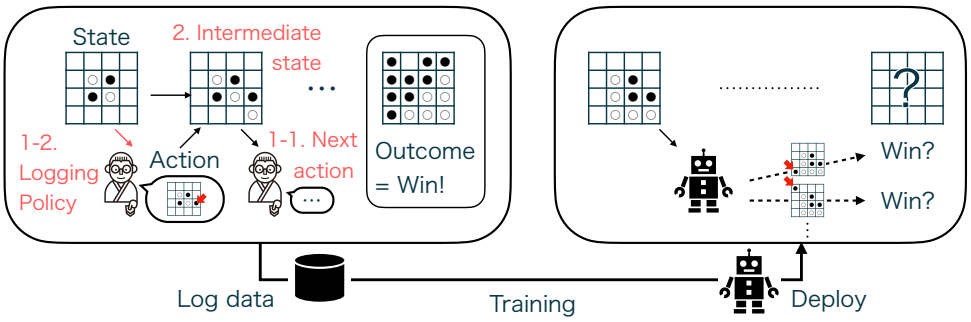


Figure 1.2: Overview of our problem setting of data-driven decision-making. The challenges we will tackle in this thesis are highlighted.

For example, in movie box-office revenue forecasting, they sometimes do not employ the mean squared error (MSE) for the prediction error. Still, the target variable (revenue) is logarithmized in advance, or classification losses are employed even though the actual outcomes are real [Lash and Zhao, 2016]. This means that the choice of evaluation metric and loss function is not self-evident from the data alone. Taking the logarithm for the target variable may reflect that mispredicting \$10M as \$11M is acceptable, but mispredicting \$0.1M as \$1.1M is a serious problem for the decision-maker, even for the same \$1M error. Furthermore, when the box-office forecast is utilized to make an investment decision, it is crucial to know whether the sales will exceed the expenses, and an accurate real-valued forecast may not be necessary. It is vital to formulate a necessary and sufficient problem setting, and predicting a numerical value is generally more complicated than classification and thus redundant.

Another example is the cost-sensitive classification [Elkan, 2001]. When the positive class corresponds to rare phenomena such as accidents or diseases, a trivial model that classifies all instances as negative achieves high accuracy but is useless to decision-makers. This mismatch is because the cost of a false negative (i.e., misclassifying a positive instance as negative) is typically much higher than that of a false positive. A simple workaround is a two-phase approach. After learning the data as it is, we can adjust the classification threshold so that even slightly suspicious instances are classified as positive. However, this approach is known to be suboptimal in general (i.e., when the model class is misspecified). On the other hand, the cost-sensitive learning considers the costs of misclassification in the training phase, which is shown to be preferable [Dmochowski et al., 2010]. We further discuss this class-imbalance problem in Chapter 3.

As discussed above, the predictive modeling approach may be suboptimal for the subsequent decision, raising the need for careful formulation as an ML problem by expert analysts. In this thesis, therefore, we consider the general framework illustrated in Fig. 1.2, where we involve the action and utility into the

Application domain	Business decision-making	Robotics	Game	Political decision-making
Example applications	Marketing, Maintenance, Risk-management	Self-driving, Humanoids	Atari, Chess	Prohibiting smoking, Job training
Primary approach	Predictive modeling (supervised)	Reinforcement learning (policy-based)	Reinforcement learning (value-based)	Causal inference
Sample size (of outcome)	Small (offline)	Large (sim-to-real)	Large (self-play)	Small (offline)
Action space	Large (combinatorial)	Medium —Large	Small —Medium	Small (binary)

Table 1.1: Comparison of application domains.

Approach	Predictive modeling	Reinforcement learning	Causal inference
Supervision	Rich (any future states)	Scarce (reward: maximization objective)	Scarce (outcome)

Table 1.2: Comparison of approaches.

model. We assume historical decision-making data, including its outcome (or the outcome can be calculated with the data as in the classification problem). Then we train a utility model which evaluates each possible action (decision) under a given situation (state) in terms of the outcome. We aim at building a useful model in terms of utility, i.e., the goal is the decision-making performance when following the model's recommendation.

Reinforcement learning and bandit have been formulated for such decision-making problems, and there has been active research in recent years on offline settings [Levine et al., 2020]. Also, causal effect inference aims at estimating the outcome of intervention [Rubin, 2005]. Nevertheless, direct utility modeling is still challenging for real-world decision-making problems in governments or enterprises compared to predictive modeling in some aspects. Table 1.1 compares various business decision-making applications that we are targeting with other domains where reinforcement learning and causal inference are successfully applied. Reinforcement learning usually requires large sample sizes, which might be fulfilled by simulation or self-play, and causal inference assumes a small action space such as a binary one. On the other hand, real-world business decision-making problems often have (1) a vast action space, such as a combinatorial one, and (2) a limited sample size. We hypothesize that these are the sources of difficulty for applying utility-level modeling to such business decision-making problems.

Also, Table 1.2 compares predictive modeling and other approaches. While reinforcement learning and causal inference assume the reward or outcome as

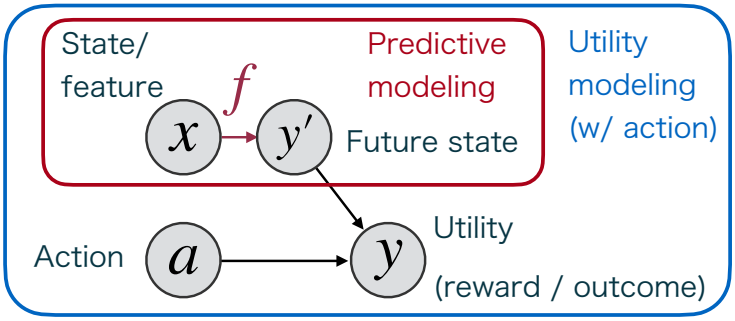


Figure 1.3: Predictive modeling can be seen as a subproblem of utility modeling.

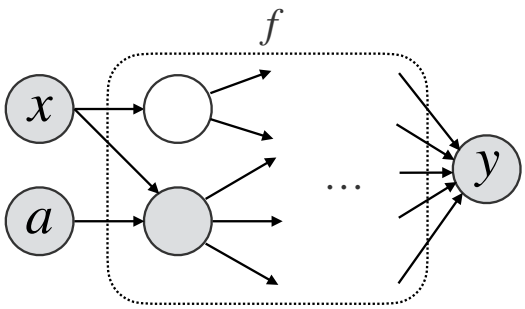


Figure 1.4: Direct modeling for utility. Each node represents a (possibly vectored) variable, and the edges represent cause-effect relationships. x denotes a given feature, a denotes an action chosen by past decision-makers, and y denotes the outcome (utility). Shaded nodes denote the accessible variables in training. We do not model each of the causal relations; instead, we directly model the evaluator f.

their ultimate supervision, which are the very target of decision-making and is often scarce, predictive modeling can be supervised by any future state of interest. Therefore, expert analysts can carefully isolate a prediction problem out of decision-making with uncertainty so that the supervision would be relatively plentiful, as illustrated in Fig. 1.3. This might be the reason of success of predictive modeling in the business domains.

Here, we have a choice between direct prediction and stepwise prediction of each phenomenon. In this regard, we would like to refer to Vapnik's principle in the field of statistical machine learning [Vapnik, 2013]:

> When solving a given problem, try to avoid solving a more general problem as an intermediate step.

Following this principle, in this thesis, moving away from the problem setting of modeling or predicting the data as it is, we consider the outer problem of predicting the utility directly as illustrated in Fig. 1.4.

Fig. 1.5 summarizes the discussion so far. We aim at automated data-driven support for decision-making by direct modeling of action and its utility. This approach is challenging in terms of (1) vast (combinatorial) action spaces and

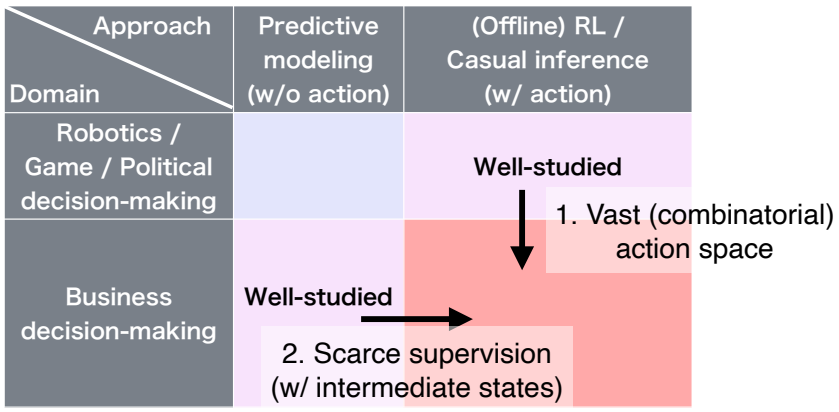


Figure 1.5: Target of this thesis and challenges.

(2) scarce supervision with relatively affluent intermediate results. We discuss these challenges and solutions with concrete example problems in the following chapters.

First, we discuss (1-1) the computational challenge due to vast action spaces. In sequential decision-making, the outcome may be evaluated with a time delay, and the utility of action in each time step cannot be assessed directly but in the long run. Thus, the utility of each action should be evaluated under the optimal subsequent actions, which may include a computationally expensive action optimization. We discuss this point in Chapter 2, taking up the infrastructure maintenance planning as a concrete example.

In multi-component maintenance planning, opportunistic maintenance is often adopted. That is, simultaneous maintenance costs less than independent maintenance for each deteriorated component due to setup costs. Suppose that we divide a continuous infrastructure (such as a road surface) into sufficiently short patches. Simultaneous maintenance for neighboring patches is economical due to the maintenance team's traveling (or setup) costs. Each maintenance action (a combination of patches maintained in a single time-step) is evaluated by the action-value function (Q-function). This function evaluates the long-term benefits of the action. Here, optimizing the Q-function with respect to the future action, which requires heavy computation, is required not only in the actual planning phase but also in the training phase of the Q-function. However, by utilizing a locality in the cost-saving in this problem, we can realize linear-time optimization, enabling Q-learning on the combinatorial action space.

Next, we discuss (2) the statistical difficulty of scarce outcomes. In real-world problems, often more detailed label information is given than the final outcome. For example, the real-valued box-office revenue is observed before being classified as success or failure. If we consider the classification accuracy as the final utility and train a classifier directly, ignoring the information of such intermediate

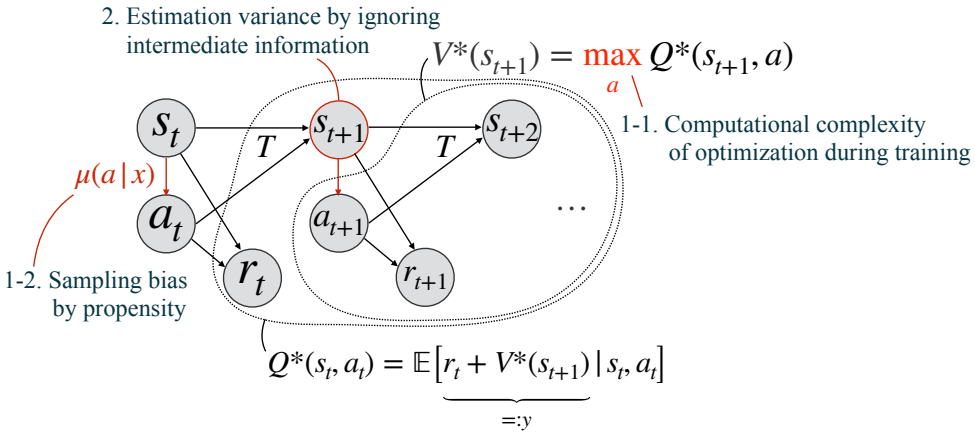


Figure 1.6: An example data-generation structure of our scope (the Markov decision process) that involves all the challenges we tackle in this thesis.

labels will lead to the loss of important label information and an increase in the estimation variance. In Chapter 3, we discuss how to utilize these intermediate labels, especially for the imbalanced classification problem, in which the positive labels (succeeded movies) are rare, as a typical case.

Finally, we discuss (1-2) the statistical challenge due to vast action spaces. When we train an evaluator f from observational data, in which actions are chosen by past decision-makers instead of randomized trials, there is a risk of biased estimation caused by the so-called spurious correlation. For example, since unhealthy people have a greater probability of being prescribed medication, the effectiveness of the treatment would be underestimated simply by comparing the healthiness of those who are prescribed with those who are not. In Chapter 4, we discuss debiased evaluation methods of the causal effect of an intervention. While most of the existing causal effect inference methods assume a binary intervention, i.e., whether to prescribe a certain kind of medication or not, many real-world problems have larger action space, e.g., choosing a combination of medication. We extend causal effect inference for such cases. Also, in Chapter 5, we discuss a combinatorial item recommendation problem as a concrete example. We consider recommendation as an intervention and extend it to the combinatorial recommendation, in which multiple items are recommended to a user simultaneously, and the responses to them are dependent.

To summarize the above, this thesis will address the following issues as challenges towards more principled analysis for decision-making through direct utility modeling, as illustrated in Fig. 1.6.

1-1 Computational challenge due to vast action space: optimization in learning iteration discussed with an example of infrastructure maintenance planning (Chapter 2)

1-2 Statistical challenge due to vast action space: estimation bias caused by

sampling policy discussed for cases with large treatment space (Chapter 4) and with an example of combinatorial item recommendation (Chapter 5)

2 Statistical challenge due to the scarce outcome: estimation variance caused by summarized instances discussed with an example of imbalanced classification with intermediate real-valued labels (Chapter 3)

We present solutions to each of these issues in the following chapters. These investigations are only for limited situations that highlight each issue; though, we hope to represent a direction towards utility modeling.

Chapter 2

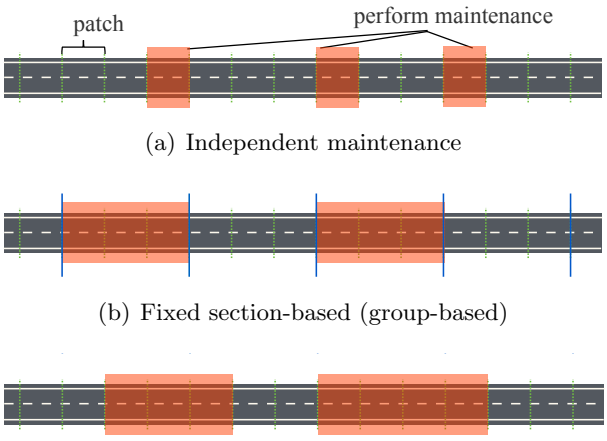
Reinforcement Learning for Maintenance Planning

2.1 Introduction

We consider an infrastructure maintenance planning problem for the road surfaces of highways; water, oil, and gas pipelines; and so on. At each discretized time step, the maintenance decision-maker considers which components, or the small patches of the road surface, should be maintained on the basis of the regularly observed condition of each component. If a number of patches have almost deteriorated and are geospatially neighboring, simultaneous maintenance (as shown in Fig. 2.1) is economical. In highway maintenance, for example, the traveling cost of a maintenance team to the site and the setup costs associated with putting up lane restrictions are incurred once for the simultaneous maintenance of a larger section consisting of contiguous small patches. Similarly, in underground pipeline maintenance, the cost of drilling vertically is incurred only once for the simultaneous maintenance of a larger section, while the cost of drilling horizontally is incurred for each patch [Papadakis and Kleindorfer, 2005].

A huge maintenance cost is paid to keep the infrastructure in good condition since its condition is critical in terms of safety, conformity, and the prevention of economic loss caused by emergent corrective maintenance or availability loss. We focus on reducing the total cost, i.e., the sum of the maintenance and condition cost (risk) caused by a deteriorated infrastructure.

Maintenance planning for minimizing the total cost has been extensively investigated in prior work [Jardine and Tsang, 2005]. For multi-component systems, i.e., those with multiple maintenance targets, the so-called economic dependency of targets and group-based maintenance is often discussed [Dekker et al., 1997, Nicolai and Dekker, 2008]. Infrastructure maintenance can also be regarded as multi-component maintenance by considering small patches as components. In



(a) Independent maintenance

(b) Fixed section-based (group-based)

(c) Dynamic grouping by combinatorial action optimization (proposed)

Figure 2.1: Comparison of road maintenance policies. Performing maintenance of longer sections that cover multiple deteriorated patches may cost less in the long run. That is, when multiple components are maintained simultaneously, overall maintenance costs are reduced since traveling costs of the maintenance team and/or setup costs are saved; this is called economic dependency. Thus, a fixed section-based maintenance policy (b) is preferable to an independent maintenance policy (a). The proposed dynamic grouping policy (c) is computationally expensive but is more flexible than the two baselines thanks to its consideration of the dependency of maintenance cost with the increased spatial resolution to small patch levels.

road maintenance, for example, cost savings can be achieved by maintaining larger sections instead of small patches [Nicolai and Dekker, 2008]. In [Papadakis and Kleindorfer, 2005], a maintenance optimization technique for an infrastructure network was proposed. They formalized a special type of economic dependency for an infrastructure network, namely, the network topology dependency (NTD), and proposed an optimization method under the benefit of maintenance for each component given. The NTD assumption reflects the locality of the economic dependency in infrastructure maintenance; i.e., the cost reduction is achieved only when the neighboring components are maintained simultaneously. To consider complex economic dependency such as NTD, combinatorial optimization is required, and the computational complexity is high. The proposed optimization method in [Papadakis and Kleindorfer, 2005] exploits the submodularity in NTD for computational efficiency. We also consider such locality in economic dependency. That is, we can assume that simultaneous maintenance is beneficial only when the maintenance target is spatially neighboring.

These maintenance optimization methods for multi-component systems are mostly built on the basis of time-based maintenance (TBM), in which each component has a predefined lifetime. Thus, the benefit of maintenance for each component can be calculated, but the uncertainty in the deterioration process

is not considered. On the other hand, recent developments in health condition monitoring technologies have enabled the actual condition of each component of an infrastructure to be observed in a timely manner. Examples of such technologies include image processing [Chambon and Moliard, 2011] and sensor networks [Kim et al., 2007] for road surfaces, and fiber optic sensing for pipelines [Li et al., 2004, Inaudi and Glisic, 2010]. These sensing technologies contribute to cost savings since only deteriorated components are maintained regardless of their age through a policy known as condition-based maintenance (CBM). Note that CBM includes a wide range of maintenance concepts, which are characterized as predictive maintenance aided by condition monitoring technologies.

These capabilities for health monitoring pose challenges to the subsequent stages of the information processing pipeline, i.e., analyzing the data and making a decision [Bousdekis et al., 2018]. In particular, optimization for multi-component CBM is not straightforward due to the economic dependency and the uncertainty in condition degradation. The optimization for this setting is computationally more challenging than TBM when taking the uncertainty into account. Studies for CBM of large-scale multi-component systems such as those for infrastructures are limited. Existing work in this context [Tian and Liao, 2011, Nguyen et al., 2015, Van Horenbeek and Pintelon, 2013] for systems such as those for heavy vehicles assumes simple economic dependency, i.e., constant maintenance costs or cost reductions, regardless of the number of components or which components are to be maintained. Since infrastructures are geospatially distributed systems with large numbers of components, the locality of economic dependency such as NTD should be considered.

A simple heuristic approach to avoid the whole combinatorial optimization with respect to locality is to divide the whole infrastructure into larger local sections in advance, which is called a fixed section-based maintenance policy (illustrated in Fig. 2.1(b)). However, this simplified approach lacks flexibility in optimization, which leads to limited performance.

To fully consider the local economic dependency and optimize large-scale maintenance actions efficiently, we utilize two dynamic programming techniques for temporal and spatial scalability. For temporal scalability, we implement the direct modeling approach of a cost-benefit evaluator, that is, Q-learning [Watkins and Dayan, 1992]. Q-learning aims to learn the total cost-benefit in the long run under the observed conditions as the state-action value function (known as the Q-function), $Q(s, a)$. Once the Q-function is learned, the maintenance action can be quickly evaluated without assessing the uncertain future degradation. For the spatial scalability of the combinatorial optimization of actions, we propose an approximated Q-function model and a linear-time optimization algorithm that exploits the locality in the economic dependency. The scalable action optimiza-

tion is also necessary for learning the Q-function, since the Q-learning requires the optimal value $\min_a Q(s, a)$ in each learning iteration. Although our Q-function is simple, the dynamic grouping of neighboring maintenance targets (shown in Fig. 2.1(c)) was significantly better than that of the fixed section-based approach.

In addition to the performance, our proposed method also provides an interpretation of the solution. Since maintenance decision-makers are often responsible for safety, the interpretability of an optimized solution matters. In our parameterized Q-function, the maintenance benefits for each component and cost are separated. Thus, the estimated benefit and condition for each component can be shown in the same figure, which enables the decision-makers to assess the cost-benefit tradeoff. A detailed discussion is provided in Section 2.5.

In our experiments, we compare our dynamic grouping approach with the fixed section-based approach, since the independent maintenance policy shown in Fig. 2.1(a) is included in the fixed section-based maintenance policy where the section length (window width) is set to one. The optimized maintenance history provides an intuitive explanation of the advantage of determining groups dynamically.

For the geospatial structure of the maintenance targets, we focus on one-dimensional (1-D) cases such as highways and pipelines, which is the simplest way to demonstrate the advantage of our approach. In addition, most parts of a highway, for example, are 1-D. For the highway network, it would be effective to combine a fixed section-based policy and dynamic grouping for intersections and branching parts, and the remaining parts, respectively.

2.2 Related Work

Condition-based infrastructure maintenance planning at scale has yet to be fully investigated. We introduce some related work and clarify the differences from our setting.

2.2.1 Multi-component maintenance planning

One of the most related areas is multi-component maintenance planning. In [Nicolai and Dekker, 2008], various types of component dependency, including economic dependency, are reviewed. NTD [Papadakis and Kleindorfer, 2005] is related the most to our local economic dependency, in particular. However, TBM is assumed, in which the maintenance benefit is given or is easily calculated without uncertainty since the aging process is deterministic. That is, we have to estimate the benefit of maintenance, which is assumed to be independent and explicitly given in [Papadakis and Kleindorfer, 2005]. To the best of our knowledge, condition-based multi-component maintenance at scale is a novel setting.

2.2.2 Condition-based maintenance planning

Both TBM and CBM aim at proactive maintenance to extend the lifetime of the entire system or to reduce accidents, downtime, and emergency maintenance costs due to unexpected failures [Peng et al., 2010]. While TBM policies tend to be too conservative with failures, resulting in high maintenance costs, CBM policies are more economical because health monitoring enables unnecessary maintenance to be controlled [Peng et al., 2010]. In the problem of maintenance planning based on CBM, it is basically assumed that the condition is measured regularly at a sufficient frequency or even continuously, except in a few studies that include the optimization of inspection policies in the problem setting [Andriotis and Papakonstantinou, 2021], and methods for prognosis and decision-making based on the measured current condition and historical data are discussed [Bousdekis et al., 2018]. Research on CBM-based maintenance planning can be broadly divided into two types of policies. The model-based approach, which optimizes after prognosis with respect to the condition, is reviewed in Section 2.2.3, and the model-free (reinforcement learning) approach, which optimizes decision policies without explicit modeling of the deterioration process, is reviewed in Section 2.2.4.

2.2.3 Model-based predictive control for maintenance planning

While we adopt a model-free approach, Q-learning, model-based approaches have also been studied. In a model-based approach, the transition model $\boldsymbol{s}_t = M(\boldsymbol{s}_{t-1}, \boldsymbol{a})$ is first estimated, and then, on the basis of the estimated model, the action optimization and future prediction to a prediction horizon are iteratively performed. Since this approach is computationally complex, existing work [Tian and Liao, 2011, Nguyen et al., 2015, Van Horenbeek and Pintelon, 2013] assumes simple economic dependencies. In railway infrastructure maintenance, applying the model-predictive control (MPC) is discussed [Su et al., 2019, 2017, Verbert et al., 2017], which is computationally expensive and does not scale to a massive number of components. In MPC, the future degradation up to the prediction horizon is predicted by using the estimated transition model, and then the maintenance action of not only the current but also the future maintenance plan up to the planning horizon is jointly optimized in each time step. In addition, the uncertainty of the model estimation should be considered in this approach. In [Su et al., 2019, 2017], the chance-constrained optimization approach is proposed. They impose a constraint to be satisfied with high probability with respect to the model uncertainty. To evaluate the constraint, they have to make multiple predictions (called "scenarios") with parameters sampled from the posterior probability of the transition model. Even though we only consider unconstrained optimization of the expected total cost, such uncertainty evaluation is generally

necessary in model-based optimization as long as the cost (risk) evaluation has nonlinearity with respect to the condition. On the other hand, our method has an advantage in that the evaluation of uncertainty is included in the training of the model, i.e., the objective function of the optimization is modeled directly, so that the evaluation of uncertainty, such as based on scenario sampling, is not necessary during training or testing. We further discuss this point in Section 2.4.1. In other areas related to maintenance, rebalancing in bike-sharing is considered a maintenance task in that the bike inventory in each 2-D distributed station is maintained to be sufficiently stocked [Liu et al., 2016]. In [Liu et al., 2016], combinatorial optimization is based on predicted values for such a problem; however, stations are clustered in advance. The advantage of our approach is that maintenance groups are determined dynamically, i.e., combinatorial optimization is performed at every time step.

2.2.4 (Deep) Reinforcement learning for maintenance planning

The application of model-free reinforcement learning (RL) to maintenance has been explored recently. Examples include on-policy RL (e.g., SARSA algorithm [Singh and Sutton, 1996, Sutton and Barto, 2018]) proposed for a petroleum industry production system [Aissani et al., 2009], for opportunistic maintenance of a fleet of military trucks [Barde et al., 2019], for minimizing the forced outage in gas turbine maintenance [Compare et al., 2018], for minimizing the average inventory level and the average number of backorders by optimizing production/maintenance policy in manufacturing [Xanthopoulos et al., 2017], and for minimizing the maintenance cost and downtime in manufacturing [Kuhnle et al., 2019].

In addition, especially since the successes of the deep Q-network (DQN) [Mnih et al., 2013], applying off-policy (deep) reinforcement learning has been actively studied. The method corresponding to an off-policy configuration is superior in that it can utilize historical data of past maintenance by human experts to implement an optimized decision-making policy that is different from the policy in the past history immediately after offline training. DQN applications to maintenance include road pavement maintenance [Yao et al., 2020], bridge maintenance [Wei et al., 2020b], and general multi-component condition-based maintenance [Zhang and Si, 2020]. In [Zhang and Si, 2020], stochastic and economic dependencies among multiple components are taken into account by DQN. DQN takes the same approach as ours in terms of Q-learning, and while its model is flexible enough to fully capture these dependencies, it is too complex to scale with respect to the number of components. The number of components assumed in [Zhang and Si, 2020] is around ten, while we assume up to thousands or more. DQN utilizes a multi-head neural network that outputs Q-values for each combination

of actions (thus, it has 2^n heads for the number of components n), while we have too many components ($n = 1000$ or more) to apply this approach in terms of statistical and computational complexity. Although DQN was successfully applied to bridge maintenance in [Wei et al., 2020b], in which a large number of components ($k = 263$) are encoded as independent Q values (thus the network only has $k \times |\mathcal{A}|$ heads, where $\mathcal{A}$ denotes the candidate maintenance actions for each component), it would be suboptimal when the action is optimized independently for each component (as in [Wei et al., 2020b]) and when the true Q value (e.g., the maintenance cost) has high dependency among actions for each component, as in our setting presented in Section 2.3.3.

One possible approach for maintaining the combinatorial optimization of components is the actor-critic (AC) algorithm as in [Liu et al., 2020], in which an actor-network that outputs an approximately optimal combinatorial action, as well as a critic network (single-head Q-network) with the action as its input, are trained. Although AC provides an approximated solution for the action optimization after training, training an actor is another big issue in terms of computation, and thus its scalability with respect to the number of components is limited. Also, these approaches face difficulties in terms of interpretability. Our approach combines a simple Q-function with dynamic programming-based optimization to resolve the scalability and interpretability issues.

Another important possibility in applying RL to maintenance is the integrated planning of the inspection policy. Andriotis and Papakonstantinou [2021] formulated maintenance decision-making as a partially observable Markov decision process (POMDP) and proposed evaluating the value of inspection, i.e., observing latent states (conditions). Although we assume regular inspection with sufficient frequency, which leads to the Markov decision process (MDP) without unobserved conditions, this would be an important direction for future research.

2.3 Problem Setting

2.3.1 Problem description

The problem can be described as optimizing which components (small patches of a 1-D structured infrastructure) to be maintained for minimizing the sum of the maintenance and condition cost (or risk) caused by deteriorated components in the long run based on the current observed condition of each component. The condition cost is a predefined non-decreasing function of the condition (degree of deterioration). The deterioration speed varies in every small patch, and thus it is inefficient to maintain by large sections, as in Fig. 2.1(b). This implies the need to divide the whole infrastructure into small patches in sufficient spatial resolution, which leads to a large number of components as a whole. Then, each component

is a small patch, and the cost of sending a maintenance team and setting up (traveling/setup cost) is relatively higher than the cost of maintaining each component (working cost), which indicates the efficiency of the dynamic group-based maintenance in Fig. 2.1(c) compared to the independent maintenance in Fig. 2.1(a). Traveling cost is assumed to occur once for neighboring components maintained simultaneously at the same time step and the working cost is proportional to the number of components maintained. The consideration of traveling cost incurs the economic dependency, i.e., the total cost of maintenance cannot be written as the summation of independent maintenance costs for each component.

We address these problems, namely, minimizing the economically dependent maintenance cost and the condition cost in the long run. For the other points, we make the following simplifying assumptions.

- Complete maintenance by replacement: the condition is fully recovered after maintenance.
- Stochastic independence: each component deteriorates independently from other components.
- Regular (real-time) inspection: the latest condition is always observed for each component.

2.3.2 Markov decision process

Our problem is sequential maintenance decision-making aimed at long-term cost minimization under imperfect knowledge of condition degradation, which can be modeled as a reinforcement learning problem. The Markov decision process (MDP) is a formalism of reinforcement learning to describe a discrete-time decision-making process with a stochastic environment. At each time step t, the decision-maker observes the state s_t (the condition of each component) and decides on an action a_t (which components to perform maintenance). At the same time, the decision-maker receives a reward (cost) $R(s_t, a_t)$, which consists of the condition cost and the maintenance cost. The state (condition) transits to the next state s_t stochastically according to an (unknown) conditional probability $p(s_{t+1}|s_t, a_t)$ depending on the current state s_t and the action a_t.

Formally, MDP consists of the following five parts.

- $\mathcal{S}$ is a set of states of the environment.
- $\mathcal{A}$ is a set of actions that can be taken as a result of decision-making.
- $p(s_{t+1}|s_t, a_t)$ is the state transition probability that means the action $a_t \in \mathcal{A}$ in the state $s_t \in \mathcal{S}$ will lead to the next state $s_{t+1} \in \mathcal{S}$.

- $R(s_t, a_t)$ is the immediate reward function (for maximization problem) or cost function (for minimization problem) of the action a_t in the state s_t.
- $\beta \in [0, 1]$ is the discount parameter of future rewards.

The aim is to optimize a (deterministic) policy π that maximizes (or minimizes) the discounted total reward (cost) in the long run, i.e.,

$$\arg\max_{\pi} \mathop{\mathbb{E}}_{\pi(a_t|s_t), p(s_{t+1}|s_t, a_t)} \left[\sum_{t'=1}^{T} \beta^{t'} R\left(s_{t'}, \pi(a_{t'}|s_{t'})\right) \right].$$

The state is what determines the reward (cost) along with the action, i.e., the condition of the components. Here, an important assumption in MDP is the Markov property for the transition, i.e., the next state only depends on the current state (condition) and the action $p(s_{t+1}|s_1, a_1, \ldots, s_t, a_t) = p(s_{t+1}|s_t, a_t)$. We assume that the state (condition) is the representative of the entire past information for both the future states and rewards.

2.3.3 Problem formulation

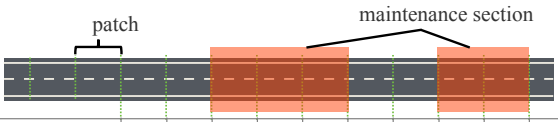
We determine when and which maintenance targets (small patches of road or pipeline) should be maintained to minimize the cumulative cost including future maintenance cost and condition cost. We assume the current cost is given explicitly as the cost function $\mathrm{Cost}(\boldsymbol{s}, \boldsymbol{a})$, where $\boldsymbol{s} = \{s_i\}_i, s_i \in \mathbb{R}$ is the state (condition) and $\boldsymbol{a} = \{a_i\}_i, a_i \in \{0, 1\}$ is the action taken at each time step ($a_i = 1$ represents that the maintenance is performed for the i-th patch).

The final goal is as follows. At each time step t, given the observed states (or the condition) $\boldsymbol{s}_t \in \mathbb{R}^n$, where n is the number of maintenance targets, we determine which targets are to be maintained to minimize the expected (discounted) total cost in the long run with regard to future actions assumed to be optimized. Thus, the optimal action for the time step t is

$$\begin{aligned} \boldsymbol{a}_t^* = \mathop{\arg\min}_{\boldsymbol{a} \in \Gamma(t) \subseteq \{0,1\}^n} \Bigg\{ & \mathrm{Cost}(\boldsymbol{s}_t, \boldsymbol{a}) \\ & + \min_{\{\boldsymbol{a}_{t'}\}_{t+1}^{t+H}} \sum_{t'=t+1}^{t+H} \beta^{t'-t} \mathop{\mathbb{E}}_{\boldsymbol{s}_{t'}|\boldsymbol{s}_t, \boldsymbol{a}_t, \ldots, \boldsymbol{a}_{t'-1}} \left[\mathrm{Cost}(\boldsymbol{s}_{t'}, \boldsymbol{a}_{t'}) \right] \Bigg\}, \end{aligned} \tag{2.1}$$

where $\beta \in [0, 1]$ is the discount parameter, $H \in \mathbb{N} \cup \{\infty\}$ is the prediction horizon, and $\Gamma(t)$ is the feasible set of actions. $a_{t,i}$ is the maintenance action for the i-th target at t. In the following sections, we assume $\Gamma(t) = \{0, 1\}^n$.

The cost function can be separated into maintenance (action) cost and con-



patch id	…	i	i+1	i+2	i+3	i+4	…			…
$s_{t,i'}$	…	$s_{t,i}$	$s_{t,i+1}$	$s_{t,i+2}$	$s_{t,i+3}$	$s_{t,i+4}$	…			…
$a_{t,i'}$	…	0	1	1	1	0	…	1	1	…
working cost	…	0	c_w	c_w	c_w	0	…	c_w	c_w	…
traveling cost	…	0	c_t			0	…	c_t		…
subtotal cost	…	0	$c_t + 3c_w$			0	…	$c_t + 2c_w$		…

Figure 2.2: Maintenance cost assumed in 1-D target environment.

dition (state) cost; namely,

$$\mathrm{Cost}(\boldsymbol{s}, \boldsymbol{a}) = C_a(\boldsymbol{a}) + C_s(\boldsymbol{s}).$$

The local economic dependency in the action cost is formalized as

$$C_a(\boldsymbol{a}) \coloneqq a_1(c_w + c_t) + \sum_{i=2}^{n} a_i\{c_w + (1 - a_{i-1})c_t\}, \tag{2.2}$$

where c_t and c_w are given constants that represent the traveling costs occurring once for neighboring patches maintained simultaneously and the working costs for each patch, respectively. Fig. 2.2 illustrates the calculation of the action cost. The interaction term $-a_i a_{i-1} c_t$ represents the local economic dependency, which comes from the traveling cost savings, i.e., the traveling cost is incurred only once for the contiguous section maintained at the same time. Although only the dependency of one-neighboring components is modeled in (2.2), the length of the locality considered can easily be extended, i.e., the maintenance cost is assumed to be decomposed as $C_a(\boldsymbol{a}) = \sum_i f_i(a_{i-k}, \ldots, a_i)$, where $\{f_i\}$ is a set of (possibly nonlinear) functions and k denotes the width of locality considered. The benefit of simultaneous maintenance is considered to have such locality ($k \ll n$), which is the key assumption that we exploit to achieve the computationally efficient algorithm described in Section 2.4. By assuming this locality, we can exploit the dynamic programming by memorizing the optimal subtotal action costs not for each full combination of sub-actions $(a_1, \ldots, a_{i-1}) \in \{0, 1\}^{i-1}$ but only for each combination of local actions $(a_{i-k}, \ldots, a_{i-1}) \in \{0, 1\}^k$ to compute the optimal subtotal action costs for the $1, \ldots, i$-th components, which results in the computational complexity of $O(n2^k)$. In this chapter, we assume $k = 1$ for simplicity. Other global nonlinearity in the maintenance cost C_a is ignored, such as the workload capacity in each time step [Nicolai and Dekker, 2008], which might matter when the resources are not sufficient.

For the state (condition) cost function, we assume the independence of each component. The dependent state cost setting has also been studied as a stochastic

dependency in [Van Horenbeek and Pintelon, 2013], although here we focus on economic dependency. For the state cost of each component, it is reasonable to assume a non-decreasing function. In our experiment, we set the following hinge cost:

$$C_s(\boldsymbol{s}) \coloneqq c_s \sum_{i}^{n} (s_i - \alpha)_+, \tag{2.3}$$

where $(x)_+ \coloneqq \max\{x, 0\}$, c_s, and α are given constants.

In addition, we assume a sufficient amount of training data $\mathcal{D}$ of the maintenance history under an unknown policy given instead of an accurate prediction of the condition degradation or the benefit of maintenance for each component. That is, we assume an off-policy setting; we do not experiment in the real environment to learn the objective in (2.1), but rather learn it from a recorded dataset.

2.4 Dynamic Group-based Maintenance by Combinatorial Q-learning

The general framework we adopted for this problem is the fitted Q-iteration [Riedmiller, 2005] described in Algorithm 1. The difference from the original work is the combinatorial optimization in the loop $\min_{\boldsymbol{a}'} Q(\boldsymbol{s}_t, \boldsymbol{a}')$ and the model of the Q-function tailored for our problem setting.

Fitted Q-learning is an off-policy Q-learning method; namely, only training data generated from an unknown policy are needed for training, while on-policy learning updates its parameters through experiments in a real environment. In mission-critical systems such as infrastructure maintenance, online updates are not feasible, and the maintenance history by human experts is often available and utilized. The future value $\min_{\boldsymbol{a}'} Q_\theta(\boldsymbol{s}_{t+1}, \boldsymbol{a}')$ is not differentiable with respect to θ due to the discrete optimization in a. Thus, in fitted-Q learning, the derivative is taken only for the current value, and the future value is fixed in each iteration.

2.4.1 Q-learning

Let the optimal state-action value function Q be the objective function of the total cost in the long run (2.1) and let $\tilde{Q}$ be the terms that exclude $C_s(\boldsymbol{s}_t)$, which is not involved in the optimization of the current action $\boldsymbol{a}$, i.e.,

$$\begin{aligned}
\boldsymbol{a}_t^* &= \arg\min_{\boldsymbol{a}} Q(\boldsymbol{s}_t, \boldsymbol{a}) \\
&= \arg\min_{\boldsymbol{a}} Q(\boldsymbol{s}_t, \boldsymbol{a}) - C_s(\boldsymbol{s}_t) \\
&= \arg\min_{\boldsymbol{a}} \tilde{Q}(\boldsymbol{s}_t, \boldsymbol{a}).
\end{aligned}$$

Algorithm 1 Fitted-Q for maintenance optimization

Input: $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}_t, \beta, C_a(\cdot), C_s(\cdot)$
Output: Trained Q-function parameter θ

Initialize θ.
$k \leftarrow 0$
while Convergence is not met **do**
  Get $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ from $\mathcal{D}$ in random order.
  Calculate the empirical target y with minimizing Q_θ by Algorithm 2:
  $y \leftarrow C_a(\boldsymbol{a}_t) + \beta C_s(\boldsymbol{s}_{t+1}) + \beta \min_{\boldsymbol{a}'} Q_\theta(\boldsymbol{s}_{t+1}, \boldsymbol{a}')$
  $L_{\theta'} \coloneqq \frac{1}{2}(y - Q_{\theta'}(\boldsymbol{s}_t, \boldsymbol{a}_t))^2$
  $\gamma_k \leftarrow (2+k)^{-1/2}$
  $\theta \leftarrow \theta - \gamma_k \frac{\mathrm{d}L_{\theta'}}{\mathrm{d}\theta'}|_{\theta'=\theta}$
  $k \leftarrow k + 1$
end while
return θ

Then, we have an optimal substructure:

$$\tilde{Q}(\boldsymbol{s}_t, \boldsymbol{a}_t) = C_a(\boldsymbol{a}_t) + \beta \mathop{\mathbb{E}}_{\boldsymbol{s}_{t+1}|\boldsymbol{a}_t, \boldsymbol{s}_t} \left[C_s(\boldsymbol{s}_{t+1}) + \min_{\boldsymbol{a}_{t+1}} \tilde{Q}(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}) \right]. \tag{2.4}$$

The first term represents the cost of maintenance at time t and the rest represent the benefit of maintenance, i.e., if we perform maintenance at time t, the condition cost at time $t+1$ (the second term), the need for maintenance and the condition costs afterwards (the third term) will decrease.

Our adopted fitted-Q learning [Riedmiller, 2005] minimizes the empirical inconsistency between both sides of Eq. (2.4) in terms of MSE (called the mean-squared Bellman error), which is the objective function L_θ in Algorithm 1. This consequently enables (approximate) minimization of (2.1) through minimizing the learned Q-function Q_θ as a proxy. Although there is no rigorous guarantee that the estimated Q function using the Bellman error will converge to the true Q function (except in special cases [Watkins and Dayan, 1992]), empirical evidence shows success in many fields [Arulkumaran et al., 2017].

Note that the Q-learning approach also handles the uncertainty in future condition degradation. In the model-based approach described in Section 2.2.3, a state transition model $\hat{M}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ is trained to predict the future state $\hat{\boldsymbol{s}}_{t+1}$, and the uncertainty in the future state $\hat{\boldsymbol{s}}_{t+1}$ has to be considered for unbiased estimation of the expectations in (2.4) due to the nonlinearity in C_s and $\min \tilde{Q}$. That is, even if the state prediction $\hat{\boldsymbol{s}}_{t+1}$ is an unbiased estimator of the expected future state $\mathbb{E}[\boldsymbol{s}_{t+1}]$, a simple plug-in estimation $C_s(\hat{\boldsymbol{s}}_{t+1})$ is biased for the term $\mathbb{E}[C_s(\boldsymbol{s}_{t+1})]$ when C_s is nonlinear. This is why the model-based approach needs to take uncertainty into account explicitly. In contrast, our q function is trained to approximate the expectation terms directly, and thus we can simply minimize Q_θ as an empirical estimate of (2.4).

2.4.2 Q-function approximation by cost and component-wise benefit decomposition

We approximate the $\tilde{Q}$ function (2.4) with a parametric model Q_θ. For Q_θ, we approximately assume the component-wise independence for these benefit terms in (2.4) (as assumed in [Papadakis and Kleindorfer, 2005]), which enables the fast optimization. The second term is component-wise independent under the component-wise transition (i.e., $s_{t+1,i} = M_i(s_{t,i}, a_{t,i})$), that is, the second term can be decomposed into the sum of functions of $a_{t,i}$ as $\sum_i \mathbb{E}\left[C_s(M_i(s_{t,i}, a_{t,i}))\right]$. Therefore, the approximation of component-wise independence corresponds to ignoring the dependencies in the third term. This approximation is accurate when β is sufficiently small*. When β is not small enough, the future rewards are taken into account and approximated to be independent for each component. There would be some planning ability lost through this approximation, e.g., clustering the degraded components left not maintained close together so that they can be maintained together in the future. On the other hand, it does not lose opportunistic planning ability in the sense of maintaining components that are likely to deteriorate in the near future.

After summing up terms that are not involved in optimizing a as a constant, we have the following parametric Q-function that represents the cost-benefit tradeoff of maintenance:

$$Q_{\boldsymbol{\theta}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \coloneqq C_a(\boldsymbol{a}_t) + \sum_i^n (1 - a_{t,i}) q(s_{t,i}; \boldsymbol{\theta}) + \theta_0, \tag{2.5}$$

where the component-wise function q represents the benefit for performing maintenance of each component, i.e., the cost of not performing maintenance. We will discuss the specific design of the component-wise benefit q in Section 2.4.4.

This component-wise separation of Q function also contributes to the interpretability in optimization. In this formulation, the value $q(s_i)$ can be interpreted as the priority of performing maintenance on the i-th component. The detailed discussion is in Section 2.6.2.

Here, the constant θ_0 in (2.5) represents a baseline cost. It does not directly affect the optimization; nonetheless, it contributes to the learning phase. Considering that the Q_θ function approximates the expected total cost in the long run (2.4), there may remain other terms besides the maintenance cost and benefit (the cost of not performing maintenance). That is the future cost that remains even when the maintenance is performed. Let us consider an extreme case where the condition cost C_s is so high or the maintenance operation is so imperfect that

*When that is not the case (e.g., $\beta = 0.99$), we can derive a variant of (2.4) by using a function $\tilde{Q}(\boldsymbol{s}_t, \boldsymbol{a}_t, \ldots, \boldsymbol{a}_{t+H})$, and the component-wise dependent term would be sufficiently small (by the factor of β^H), after which the computational complexity would be 2^H times higher.

it is optimal to perform maintenance for almost all components in every time step. Since the immediate maintenance cost is the same in both Q_θ and (2.4), the remaining terms in (2.4) would be the (expected) future condition and maintenance costs and those in (2.5) would be the benefit q and the constant term θ_0. Without the constant term ($\theta_0 = 0$), we need to express all the future costs by the benefit terms of the few components that are not maintained, which causes over-estimation of the maintenance benefit. In other words, θ_0 is the constant that summarizes terms that are not involved in the current action optimization with respect to the cost-benefit tradeoff.

2.4.3 Q-function optimization by dynamic programming

Our approximated Q-function can be optimized with respect to the action in linear time by means of dynamic programming. This is because the locality of economic dependency enables the optimal action of a patch to depend only on the optimal action of the neighbors; i.e., it has an optimal substructure property, as shown below.

First, we define the partial value function $v_i(a)$ as

$$v_1(a_{t,1}) \coloneqq a_{t,1}(c_w + c_t) + (1 - a_{t,1})q(s_{t,1})$$

and for $i = 2, \ldots, n$,

$$v_i(a_{t,i}) \coloneqq \min_{a_{t,1},\ldots,a_{t,i-1}} \Bigg\{ a_{t,1}(c_w + c_t) + \sum_{i'=2}^{i} a_{t,i'}\{c_w + (1 - a_{t,i'-1})c_t\} + \sum_{i'=1}^{i}(1 - a_{t,i'})q(s_{t,i'}) \Bigg\}.$$

Note that, the minimization of $v_n(a_{t,n})$ is equivalent to that of the whole Q-function.

$$\min_{a_{t,n}} v_n(a_{t,n}) + \theta_0 = \min_{a_{t,1},\ldots,a_{t,n}} Q_\theta(\boldsymbol{a}_t, \boldsymbol{s}_t)$$

The partial value $v_i(a_{t,i})$ depends on the combination of actions $\{a_{t,i}\}_i$ only through the neighboring partial values $\{v_{i-1}(a_{t,i-1})\}_{a_{t,i-1}}$; namely,

$$\begin{aligned} v_i(a_{t,i} = 0) = \min_{a_{t,i-1}} \{ & v_{i-1}(a_{t,i-1} = 0) + q(s_{t,i}), \\ & v_{i-1}(a_{t,i-1} = 1) + q(s_{t,i})\}, \\ v_i(a_{t,i} = 1) = \min_{a_{t,i-1}} \{ & v_{i-1}(a_{t,i-1} = 0) + c_t + c_w, \\ & v_{i-1}(a_{t,i-1} = 1) + c_w\}. \end{aligned}$$

Algorithm 2 Dynamic programming for optimizing $\boldsymbol{a}$

Input: $\boldsymbol{s}_t, \boldsymbol{\theta}$
Output: $\boldsymbol{a}_t^* = \underset{\boldsymbol{a}' \in \{0,1\}^n}{\arg\min}\, Q_\theta(\boldsymbol{s}_t, \boldsymbol{a}')$

% forward step
$v_1(a_{t,1} = 0) \leftarrow q(s_{t,1}; \boldsymbol{\theta})$
$v_1(a_{t,1} = 1) \leftarrow c_w + c_t$
for $i = 2, ..., n$ **do**
  $v_i(a_{t,i} = 0) \leftarrow \min_{a' \in \{0,1\}} v_{i-1}(a_{t,i-1} = a') + q(s_{t,i}; \boldsymbol{\theta})$
  $a_{t,i-1}(a_{t,i} = 0) \leftarrow \underset{a' \in \{0,1\}}{\arg\min} \; v_{i-1}(a_{t,i-1} = a') + q(s_{t,i}; \boldsymbol{\theta})$
  $v_i(a_{t,i} = 1) \leftarrow \min_{a' \in \{0,1\}} v_{i-1}(a_{t,i-1} = a') + (1 - a')c_t + c_w$
  $a_{t,i-1}(a_{t,i} = 1) \leftarrow \underset{a' \in \{0,1\}}{\arg\min} \; v_{i-1}(a_{t,i-1} = a') + (1 - a')c_t + c_w$
end for
% backward step
$a_{t,n}^* \leftarrow \underset{a' \in \{0,1\}}{\arg\min}\, v_n(a_{t,n} = a')$
for $i = n - 1, ..., 1$ **do**
  $a_{t,i}^* \leftarrow a_{t,i}(a_{t,i+1} = a_{t,i+1}^*)$
end for
return $\boldsymbol{a}_t^* = \left(a_{t,i}^*\right)_i$

This property means that we only have to calculate the partial values $\{v_i(a_{t,i} = 1), v_i(a_{t,i} = 0)\}_{i \in [n]}$ to obtain the optimal action $\boldsymbol{a}_t^*$, which takes only linear time with respect to the number of components n. The detailed algorithm is described in Algorithm 2.

2.4.4 Modeling q_i: the maintenance priority of i-th target

The component-wise value $q_i = q(s_{t,i}; \boldsymbol{\theta})$ in (2.5) represents the priority (or the benefit) of performing maintenance for the i-th component. In this section, we design the hypothesis space of the function q specifically using domain knowledge of desirable properties as a benefit function.

First, the benefit of maintenance should be non-negative, i.e., $q(s_{t,i}) \geq 0$ should hold. Since the state cost $C_s(\boldsymbol{s}_t)$ is non-decreasing in $s_{t,i}$ and the condition does not improve (at least without maintenance), $q(s_{t,i})$ should also be non-decreasing. Considering these properties, we utilize the following parameterization for q, which is an extension of the softplus (smoothed ReLU) function [Zheng et al., 2015]:

$$q(s_{t,i}; \boldsymbol{\theta}) \coloneqq \frac{\theta_3}{\theta_1} \log\left(1 + \exp(\theta_1(s_{t,i} - \theta_2))\right). \tag{2.6}$$

The parameter θ_1 controls the smoothness. Since we adopt non-linear parameterization for q, convergence is not guaranteed [Getoor and Taskar, 2007]. Thus, we try several initial parameters for $\boldsymbol{\theta}$.

Here, we explain how to design the q function given the condition cost C_s in (2.3). Since the third term in (2.4) is greater than 0, the q function should be greater than the expected condition cost in the next time step, i.e., $q(s_{t,i}) \geq \beta\,\mathbb{E}\left[C_s(s_{t+1,i})|s_{t,i}\right]$ should hold. Furthermore, the benefit asymptotes to zero when the condition is good, i.e., $\lim_{s_{t,i}\to-\infty} q(s_{t,i}) = 0$, and asymptotes to the condition cost in the next time step $(t+1)$ plus a constant that represents the (averaged) maintenance cost in $t+1$ since it must be maintained in $t+1$, i.e., $\lim_{s_{t,i}\to\infty} q(s_{t,i}) - \mathbb{E}\left[C_s(s_{t+1,i})|s_{t,i}\right] = \text{Const}$. Our adopted softplus function reflects these properties under the definition of C_s in (2.3).

2.5 Experiment

We investigated the effectiveness of this approach with experiments in a simulated environment.

2.5.1 Setup

Since degradation proceeds at an accelerated rate, a log-linear model is often assumed [Srivastava and Mondal, 2016, Zhou et al., 2011, Famurewa et al., 2015]: $s_t = e^{\beta t+\alpha} = e^{\beta} \cdot s_{t-1}$. This represents that the degraded condition itself causes further degradation. Also, we consider a stochastic degradation model with heteroscedastic noise, i.e., the degradation rate depends on its location (component) i. The heteroscedasticity of road pavement, for example, is caused by the difference in traffic conditions, material properties, construction quality, and other geometric conditions [Adlinge and Gupta, 2013, Hong and Wang, 2003]. This difference in the degradation rate is the very reason CBM, in which the component to be maintained is determined in accordance with its degradation condition, is superior to TBM, which assumes a pre-determined lifetime. We also take into account the skewness of the degradation rate distribution [Peng and Tseng, 2013], i.e., several components show very fast degradation rates and need frequent maintenance. To reproduce these conditions, we use the following transition models $\{M_i\}$ for each component (position) i as the environment:

$$M_i(s_{t,i}, a_{t,i}) = \begin{cases} 1.1 s_{t,i} + \epsilon\Delta_i & (a_{t,i} = 0) \\ 1.0 & (a_{t,i} = 1), \end{cases} \tag{2.7}$$
$$\epsilon \sim \exp(\mathcal{N}(0,1)),$$

where Δ_i is the characteristic excess degradation rate for the i-th target, which is generated from the log-normal distribution $\Delta_i^{(base)} \sim \exp(\mathcal{N}(0, 1.3))$ followed by the application of a Gaussian filter ($std = 2$) for smoothness. We fixed the

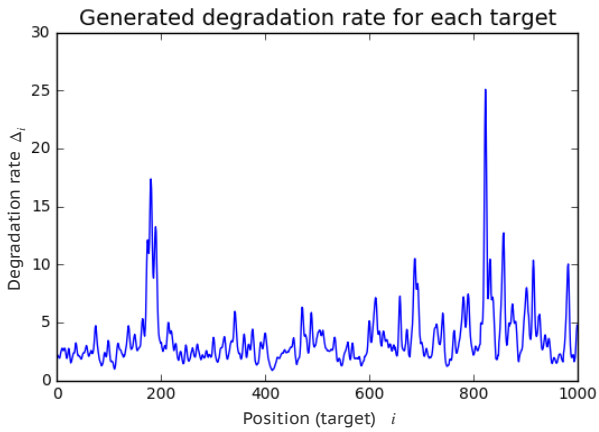


Figure 2.3: Generated position-specific average degradation rates. Degradation phenomena often have this kind of local and skewed distribution.

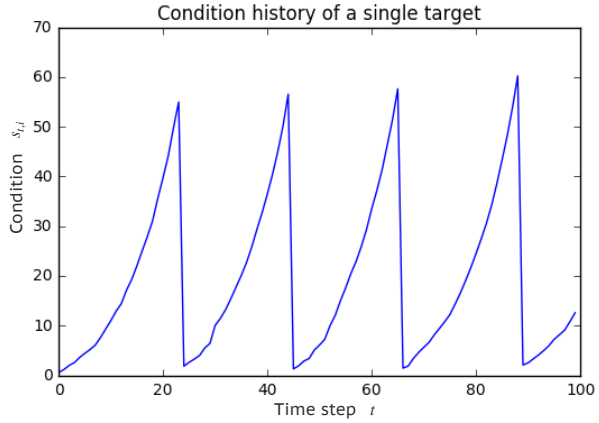


Figure 2.4: Condition and maintenance history of specific target generated from (2.7) and the fixed section-based CBM policy (2.8). Condition degrades gradually, then returns to a good condition when maintenance is performed.

average degradation rates $\{\Delta_i\}$ once after sampling; thus the average frequency of maintenance needed for the i-th component is constant for the entire training and test periods. The resulting degradation rates $\{\Delta_i\}$ are shown in Fig. 2.3 and the condition history for a specific component is shown in Fig. 2.4.

For the cost function, we used the state cost function C_a in (2.3) with the parameters $\alpha = 50, c_s = 1$ and the action cost function C_s in (2.2) with the parameters $c_w = 2, c_t = 10$.

2.5.2 Training and testing settings

We set the number of targets $n = 1000$ and the training and testing periods $T_{\text{train}} = \{0, \ldots, 1000\}, T_{\text{test}} = \{1001, \ldots, 2000\}$, respectively. To generate the training data, we adopted the fixed section-based policy in (2.8) with the parameters $w = 10, \theta_t = 45$. The random values Δ_i and $\epsilon_{t,i}$ are the same for all policies tested, i.e., CBM with various parameters and the proposed policy. We

Table 2.1: Initial parameters tested

θ_0 (Baseline cost)	{0.1, 1}
θ_1 (Smoothness)	{0.05, 0.1, 0.2, 0.5, 1, 2}
θ_2 (Threshold)	{20, 25, 30, 35, 40, 45, 50}
θ_3 (Slope)	{0.1, 0.3, 1, 2}

set the discount parameter $\beta = 0.9$, since a too-large discount parameter causes divergence. After training, we fixed the estimated Q-function during the test phase and ran a simulation. The test evaluation was done by the total cost in the entire test period T_{test}.

We tested the initial parameters of the Cartesian product of the candidate shown in Table 2.1 and selected the best parameter that minimizes the training objective $\sum_{t \in T_{\text{train}}} L_\theta$. These initial parameters were selected considering the environment to ensure that we had a good parameter near one of the initial parameters. Let us consider a greedy policy as a baseline that considers only the action cost and condition cost in the next step, i.e., one that ignores the third term in (2.4). Further suppose that the expectation and the cost function C_s in the second term can be approximately exchangeable, i.e.,

$$\begin{aligned} \mathop{\mathbb{E}}_{\boldsymbol{s}_{t+1}|\boldsymbol{a}_t,\boldsymbol{s}_t}[C_s(\boldsymbol{s}_{t+1})] &\simeq C_s(\mathbb{E}[\boldsymbol{s}_{t+1}|\boldsymbol{a}_t,\boldsymbol{s}_t]) \\ &= \sum_i (1-a_i) C_s(1.1 s_{t,i}). \end{aligned}$$

Then, (2.4) can be expressed by our model with the parameters $\theta_0 = 0, \theta_1 \to \infty, \theta_2 = \alpha/1.1 \simeq 45$, and $\theta_3 = 1.1\beta c_s \simeq 1.0$. The optimal Q function should be larger than this greedy Q function. The third term in (2.4) may include the baseline cost $\theta_0 > 0$. Due to the convexity of C_s, the second term will gradually increase near the threshold $s_{t,i} = 45$, i.e., the smoothness should be introduced as $\theta < \infty$. Also, considering the future cost (the third term in (2.4)), the threshold might be smaller: $\theta_2 < 45$. The condition may not exceed the threshold θ_2 so many times because the maintenance is performed preventively, and there are seldom instances in a region such as $s > 50$, where the slope parameter θ_3 alone is dominant, so the optimal slope θ_3 depends on the interaction with other parameters. Although we chose the candidate initial parameters taking these properties into account, it may be possible to choose them using a black-box optimization such as the Bayesian optimization [Snoek et al., 2012, Shahriari et al., 2015].

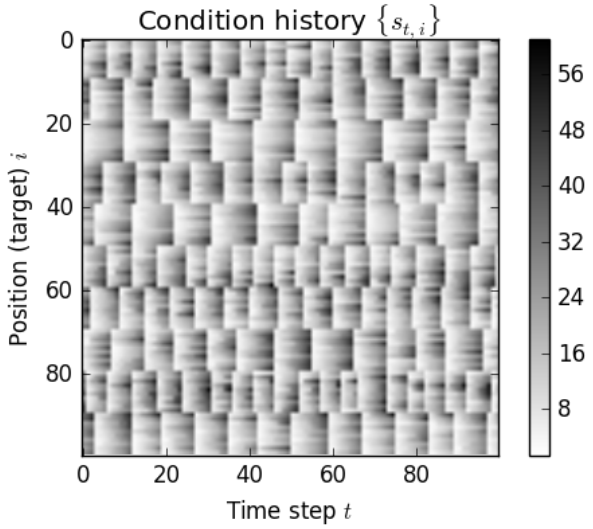


Figure 2.5: State history $\{s_{t,i}\}_{t,i}$ under fixed section-based CBM policy (in Fig. 2.1(b) and Eq. (2.8)). Dark regions are degraded and thus need maintenance.

2.5.3 Baseline method: fixed section-based CBM

As the baseline method, we examined the fixed section-based CBM approach (Fig. 2.1(b)). With the parameter of window width w, the targets are split into intervals in advance, and the action is taken for all targets in the section if the most degraded target in it is greater than the threshold θ_t:

$$\pi_{\mathrm{CBM}}(a_{t,i} = 1 | \boldsymbol{s}_t) = \begin{cases} 1 & \left(\max_{j \in A_i} \{s_{t,j}\}_j \geq \theta_t \right) \\ 0 & (\text{otherwise}), \end{cases} \tag{2.8}$$

where $A_i = \{j \mid \lfloor j/w \rfloor = \lfloor i/w \rfloor\}$ is the set of components in the same section as the i-th component. The resulting condition history with parameters $(w = 10, \theta_t = 50)$ is shown in Fig. 2.5, which is also used for generating training data. Performance under this policy was sensitive to the parameters as shown in Fig. 2.6. These parameters have to be appropriately optimized using the training data, which is another issue. To simplify the discussion, we used the optimal parameters selected by the test performance and demonstrate that our method with learned parameters still outperforms the baseline with the optimal parameters.

2.6 Results and Discussion

2.6.1 Discussion on performance

The proposed method outperformed the baseline approach even when the best parameters (w, θ_t) in the test period were chosen for the baseline, as shown in Table 2.2.

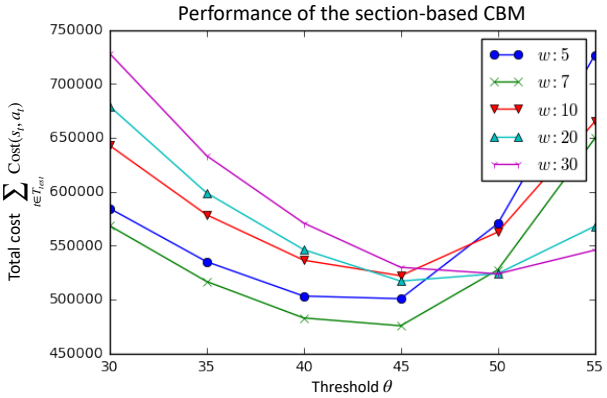


Figure 2.6: Performance of fixed section-based CBM with various parameters. The performance is strongly dependent on the parameters, the window width w and the threshold θ_t, and the means to optimize them beforehand is not straightforward. Nonetheless, as a baseline method, we can assume these parameters are optimized appropriately using the training data; thus we compare our method with the baseline method under the best parameters in the test period ($w = 7, \theta_t = 45$).

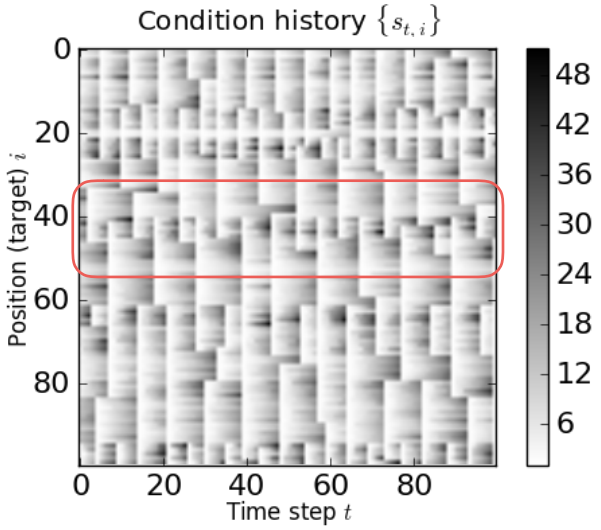
Table 2.2: Performance comparison. We ran simulations for each policy and evaluated the test performance in T_{test}. For the proposed method, we learned the Q-function with data in T_{train} and fixed the Q-function in the test phase. For the baseline (section-based CBM), we studied several parameters (as in Fig. 2.6) and showed the best performance in the test period.

	Section-based CBM with best parameters	Proposed method
Total cost	4.76×10^5	$\mathbf{4.31 \times 10^5}$

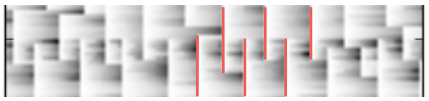
A possible explanation of the performance of dynamic grouping is illustrated in Fig. 2.7. Rapidly degrading targets ($i \in [40, 45]$) are frequently maintained with negligible expense by selecting sections that cover such targets alternately (indicated by red lines in Fig. 2.7(b)). This alternate selection of sections cannot be achieved in the fixed section-based approach, and we consider this is a key benefit of the flexibility of dynamic grouping.

2.6.2 Interpretability in optimization

The advantage of the separability approximation of the state cost function, i.e., $C_s(\boldsymbol{s}_t) = \sum_i q_i(s_{t,i})$, is not only the computational efficiency but also the interpretability in optimization. Black-box optimization is difficult to accept for maintenance decision-makers in the field since they are responsible for the safety or have motivation for factors other than minimizing the explicitly defined cost function with observed data. As shown in Fig. 2.8, $q(s_{t,i})$ can be interpreted as the maintenance priority of the i-th target. We can plot it in the same graph as observed physical quantities, which maintainers are familiar with.



(a) Part of state history under the proposed policy



(b) Framed section in (a) extracted

Figure 2.7: Condition history under dynamic grouping policy with learned parameters (a). The better performance of our approach (in Table 2.2) possibly comes from the exploitation of the local economic dependency and the variety of degradation rates. Rapidly degrading targets (extracted in (b)) are maintained frequently with a small number of groups by selecting groups alternately (indicated by red lines).

2.7 Conclusion

In this chapter, we presented a condition-based infrastructure maintenance planning problem as a sequential and combinatorial optimization problem. This problem setting requires large-scale combinatorial optimization for the combination of current and future actions of each component, considering the uncertainty of the future conditions. To achieve the dynamic grouping of small components of large infrastructures, we introduced the local economic dependency assumption for maintenance cost. We proposed a number of approximations, namely, the Q-learning approach for temporal scalability and uncertainty and a parameterized Q-function and dynamic programming for spatially scalable optimization of the Q-function, which exploits the locality in economic dependency.

We investigated the performance in a simulated environment. The resulting condition history showed the advantage of dynamic grouping; that is, rapidly degrading targets could be maintained frequently by selecting alternate sections with a small extra expense only in working cost. The proposed method is not only has a superior performance but is also interpretable, which we feel is important for

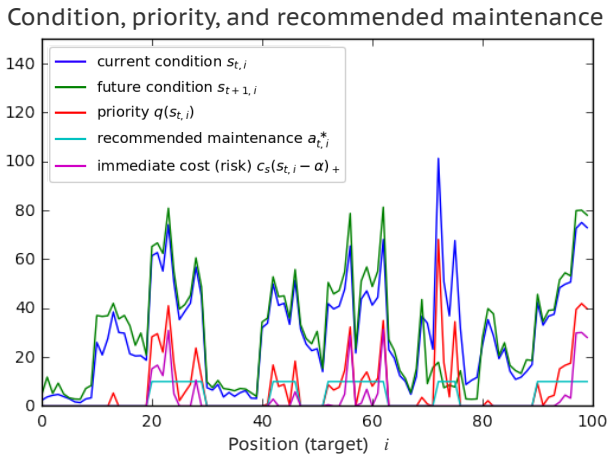


Figure 2.8: Possible user interface of the maintenance recommendation system showing the current condition, current state cost, the estimated priority of maintenance, and recommended maintenance groups. $q(s_{t,i})$ can be interpreted as the maintenance priority of the i-th target, and we can plot it with observed physical quantities to explain the reason for the recommendation.

maintenance decision-makers to accept the recommended action. This is achieved by separating the objective function of action optimization (Q-function) into the action cost and sum of maintenance priority for each component, which can be indicated in the same figure as the observed condition of each component. Comparing the cost and maintenance priority enables the maintenance planner to make a reasonable decision.

There are a number of remaining issues or limitations with our method, as well as possible extensions. In real applications, historical data is sometimes limited. Since the transition of each target at a single time step is summarized into one sample in our approach, our method may not be sample-efficient. Thus, in those cases, we have to consider incorporating a model-based approach, as in [Gu et al., 2016], in which the transition for each target is learned as a prediction model $\hat{M}(s_i, a_i)$. Also, in our experiment, we assumed that the condition observations are noise-free, but in the maintenance field, they often have severe noise or outliers. Therefore, estimating the true condition s_t, or calculating q_i from many observations (e.g., a CNN-like model $q_i(s_{t-\tau:t,i-k:i+k})$), is an important possible extension. In addition, we focused on 1-D infrastructures. Other possible applications of the dynamic grouping approach include whole network settings such as NTD and two-dimensionally distributed targets such as machine maintenance and inventory management of vending machines, ATMs, and so on.

Chapter 3

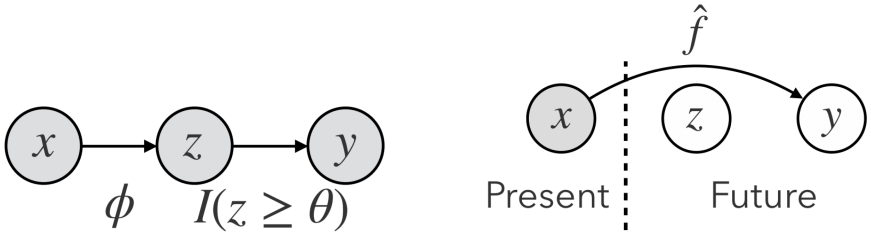
Incorporating Intermediate Labels for Sample-efficient Imbalanced Direct Classification

3.1 Introduction

Class imbalance is often a major problem in real-world data analysis [Japkowicz and Stephen, 2002, Haixiang et al., 2017], since the class of interest (i.e., the positive class) often corresponds to rare events, such as disasters, accidents, diseases [Haixiang et al., 2017], abnormalities [Fuqua and Razzaghi, 2020], or conversions in advertisement recommendation tasks [Lee et al., 2012]. In such cases, the performance will be limited by the size of the positive training sample. However, among such real-world imbalanced problems, there are cases where "near-miss" instances, i.e., negative but nearly-positive instances, are relatively plentiful.

In flood prediction [Cloke and Pappenberger, 2009], for example, actual floods are rare, while there are relatively many near-miss cases where the water level approached the height of the riverbank. Also, in condition-based maintenance, the condition of each piece of equipment is monitored regularly, and the maintenance is carried out to keep the condition not to reach an alarm-level [Lee et al., 2006]. While actual accidents are rare, there are many near-miss incidents where the condition approaches the alarm-level [van der Schaaf, 1995, Li and Nilkitsaranont, 2009]. Furthermore, sales forecast for new products such as songs [Herremans et al., 2014] or books [Chang and Lai, 2005] are difficult due to the skewness of the sales distribution [Hendricks and Sorensen, 2009]. If one needs only to know whether the sales exceed a threshold, such as a break-even point for deciding to publish, the task would be a classification task. While hit books are rare, we often have plentiful records of near-miss hit books whose sales are slightly below the break-even point.

This chapter is based on Akira Tanimoto, So Yamada, Takashi Takenouchi, Masashi Sugiyama, and Hisashi Kashima, Improving imbalanced classification using near-miss instances, submitted to Expert Systems with Applications.



(a) Data generation model (for the training phase).

(b) Prediction model (for the inference phase).

Figure 3.1: Our assumed graphical models for training and inference. Gray nodes represent observed variables at each phase. (a) Our assumed data generation model. $x \in \mathbb{R}^d$ is a feature vector, $z \in \mathbb{R}$ is a numerical mediator variable that represents "positivity," I is an indicator function, θ is a threshold, and $y := I(z \geq \theta)$ is the binary label. (b) Our employed prediction model. z typically represents a future condition; thus it is not available in the test phase, and need not be predicted. The only prediction target y is whether or not the condition z exceeds a given threshold. Thus, we do not predict z; rather, we predict y directly.

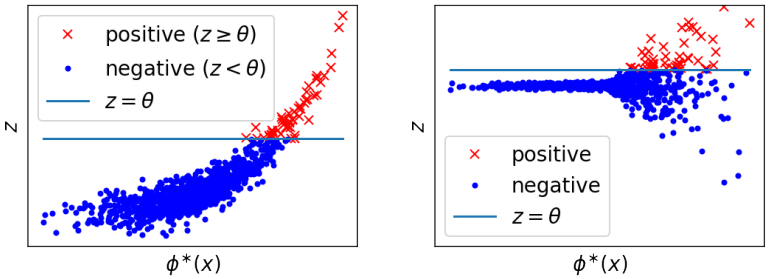
Exploiting such near-miss data is a well-known heuristic in the field of accident prevention. Heinrich et al. (1980), Jones et al. (1999), and Barach and Small (2000) argued the importance of collecting data not only regarding actual accidents but also regarding near-miss incidents and suggested to take measures to prevent them. To the best of our knowledge, exploiting near-miss data has not yet been sufficiently investigated in machine learning literature. We therefore show that this lesson in accident prevention applies to machine learning, i.e., even when the number of true positive cases is quite limited, the accuracy can be improved by obtaining additional information to identify the near-miss cases.

Such additional information we assume is "positivity" $z \in \mathbb{R}$ given in the training phase as in Fig. 3.1(a). The label y is defined by whether or not z exceeds a given threshold θ. Fig. 3.2 shows synthetic examples. Positivity z represents, for example, the future water level in flood prediction, the future condition of equipment in condition-based maintenance, or the sales of the new book. Note that, since z typically denotes some future condition, z is not available in the inference phase.

Since the final goal is to predict the binary label y, a naive approach is to throw away z and train a classifier only from (x, y) pairs.

Imbalanced classification using binary labels has been actively studied [Haixiang et al., 2017, Leevy et al., 2018].

In particular, when the number of positive data is small, cost-sensitive learning [Elkan, 2001] is often used to cancel the estimation bias due to the class imbalance, in which misclassification costs for false positives and false negatives



(a) $p(z|x) = \mathcal{N}(\mu(\phi^*(x)), \psi)$. (b) $p(z|x) = \mathcal{N}(\mu, \psi(\phi^*(x)))$.

Figure 3.2: Toy examples for the setting illustrated in Fig. 3.1(a). $\phi^*(x) = w^{*\top}x$ is the true scoring function. (a) Toy data generated by a generalized linear model that we used in our experiments. (b) Another data with heteroscedastic noise, showing how regression- and rank-based approaches may fail.

are unequal. While it converges asymptotically to the Bayes optimal solution, estimation variance is high, as we theoretically prove in Section 3.4 and experimentally demonstrate in Section 3.5.3.

Many methods have been proposed in this context, including those based on under- and oversampling with synthetic data generation [Chawla et al., 2002, He et al., 2008, Barua et al., 2012, Wei et al., 2020a] and hybrid/ensemble methods [Seiffert et al., 2009, Chawla et al., 2003, Kim and Sohn, 2020]. We also make comparisons with representative ones of these in Section 3.5.4.

A tempting approach for avoiding high estimation variance is regression, i.e., estimating the generative model ϕ in Fig. 3.1(a). While here we never confront the imbalance issue, naive regression methods cannot convey information other than the conditional mean $\mathrm{E}[z|x]$, and fail when the noise level is not constant, as illustrated in Fig. 3.2(b). Further discussion of this approach and the relation of z and $p(y|x)$ is provided in Section 3.3.2.

We therefore take a direct modeling approach, as in Fig. 3.1(b), and exploit z as side-information to alleviate the estimation variance. Then, provided that the near-miss positive instances are relatively plentiful with respect to the real positives, we can increase the effective positive rate by regarding the near-miss positive instances as being partly positive. This makes it possible for our method to enjoy reduced estimation variance, as is proved in Section 3.4.2, in exchange for additional bias, as in Section 3.4.3. Experimental results given in Section 3.5.4 indicate the effectiveness of our approach.

Our main contributions are three-fold. First, we propose a new learning algorithm to exploit the positivity z, which is model-agnostic, i.e., it can be incorporated into many off-the-shelf implementations of classifiers. Second, we derive a non-asymptotic bound, which shows the mechanism that our method can reduce the estimation variance via increasing the effective size of the positive

sample with the help of near-miss instances, in exchange for additional bias. The bound of the additional bias also gives a characterization of effective positivity information. Lastly, our extensive experiments illustrate the effectiveness of our method compared to the conventional classification methods and the regression- and rank-based approaches.

3.2 Problem Setting

We want to learn a scoring function (decision function) $g : \mathcal{X} \to \mathbb{R}$ that defines a plug-in binary classifier $\hat{y} = I(g(x) \geq 0)$, where $\mathcal{X} \subset \mathbb{R}^d$ is the feature space and I is the indicator function. Given a task-specific threshold θ we learn from the data set $S = \{d_1, d_2, \ldots, d_N\}$, where N is the sample size, and $d_n = (x_n, z_n)$ consists of a feature vector $x_n \in \mathcal{X}$ and a mediator variable $z_n \in \mathbb{R}$, which we refer to as positivity. Note that the positivity z_n is accessible only in the training phase, and we *cannot* use z_n in the test phase. A class label is determined as

$$y_n \coloneqq I(z_n \geq \theta).$$

Without loss of generality, we hereafter assume $\theta = 0$ (i.e., let $z_n - \theta$ be the new z_n).

Positivity z_n is considered related to a "probabilistic label (soft-label)" $p_n = p(y = 1|x_n)$; however, p_n itself is not given, which represents the difference from existing soft-label studies [Nguyen et al., 2011b, 2014, Peng et al., 2014]. A detailed discussion of this is given in Section 3.3.2.

For the evaluation, we adopt a cost-sensitive metric called the weighted accuracy (WA) [Cohen et al., 2006]:

$$\begin{aligned}\mathrm{WA}_N(g) \coloneqq \frac{1}{N}\sum_{n}^{N} \big\{ & C_+ I(z_n \geq 0 \wedge g(x_n) \geq 0) \\ & + C_- I(z_n < 0 \wedge g(x_n) < 0) \big\},\end{aligned}$$

where C_+ and C_- are task-specific constants for the positive class and the negative class, respectively, as introduced in the cost-sensitive learning framework [Elkan, 2001, Ling and Sheng, 2010, Vasile et al., 2017], and $\wedge$ represents the logical AND. Since we consider the imbalanced case, the accuracy for the rare positive class is usually emphasized, i.e., $C_- < C_+$. We also consider a special case of WA, letting $C_+ = N/2N_+$ and $C_- = N/2N_-$, where $N_+ := \sum_n^N I(z_n \geq 0)$ and $N_- := \sum_n^N I(z_n < 0)$, as balanced accuracy (BA). Here, $(1 - \mathrm{BA})$ is the balanced error rate (BER), which is often adopted in imbalanced problems [Chen and Wasikowski, 2008]. We evaluated the performance of a classifier with respect

to BA in our experiments:

$$\begin{aligned}\mathrm{BA}_N(g) \coloneqq \frac{1}{N}\sum_n^N \Bigg\{ &\frac{N}{2N_+} I(z_n \geq 0 \wedge g(x_n) \geq 0) \\ &+ \frac{N}{2N_-} I(z_n < 0 \wedge g(x_n) < 0) \Bigg\}.\end{aligned} \tag{3.1}$$

3.3 Learning with Positivity

In this section, we propose a proxy loss, a generalization of the cost-sensitive learning to the case in which positivity is obtained, and compare it with another approach, i.e., the generative modeling.

3.3.1 Proposed loss function

A naive approach for this problem is the cost-sensitive learning which minimizes the convex relaxation of (const. $-$ WA_N) [Dmochowski et al., 2010], i.e., its empirical risk is

$$\hat{L}(g) = \frac{1}{N}\sum_n^N \left\{C_+ y_n \ell(g(x_n)) + C_-(1-y_n)\ell(-g(x_n))\right\},$$

where $\ell(g)$ is the instance-wise loss such as the hinge loss or the negative log-likelihood. As we prove in Section 3.4, however, the estimation variance is high, and thus the performance would be poor under the limited size of the positive training sample. To overcome this limitation, we propose the following proxy loss that treats near-miss instances as being partly positive:

$$\hat{L}_T(g) \coloneqq \frac{1}{N}\sum_n^N \left\{C_{T,+}\sigma(z_n/T)\ell(g(x_n)) + C_{T,-}\sigma(-z_n/T)\ell(-g(x_n))\right\}, \tag{3.2}$$

where $\sigma(a) \coloneqq 1/(1+\exp(-a))$ is the sigmoid function, T is a hyperparameter called temperature, $C_{T,+} \coloneqq C_+ \frac{N_+}{N_{T,+}}$ and $C_{T,-} \coloneqq C_- \frac{N_-}{N_{T,-}}$ are rebalanced cost parameters, and $N_{T,+} \coloneqq \sum_n^N \sigma(z_n/T)$. We refer to $\sigma(z/T)$ as the soft-label. Considering that the soft-label goes to the original hard label in the limit of $T \to 0$, i.e, $\lim_{T\to 0}\sigma(z/T) = y$ (except for $z=0$), our loss function includes the cost-sensitive learning as the limit of $T \to 0$.

Our loss function (3.2) can be implemented as instance weighting; namely, we duplicate the whole training set for positive and negative parts with weights $C_{T,+}\sigma(z_n/T)$ and $C_{T,-}\sigma(-z_n/T)$, respectively. Then we can train any off-the-shelf base learner $\mathcal{A}$ with duplicated instances and weights. The detailed algorithm for the setting of BER minimization is described in Algorithm 3.

One benefit of introducing the soft-label is increasing the effective positive

Algorithm 3 Learning with positivity

Input: $D = \{(\mathbf{x}_n, z_n)\}_n$, θ, T, a base learner $\mathcal{A}$
Output: Trained model M

1: **for** $i = 1$ to N **do**
2: $\quad s_n \leftarrow \frac{1}{1+\exp\left(-\frac{z_n-\theta}{T}\right)}$
3: **end for**
4: $p_{T,+} \leftarrow \frac{\sum_i^N s_n}{N}$
5: $D' \leftarrow \left\{\left(\mathbf{x}_n, y=1, \text{weight} = \frac{s_n}{p_{T,+}}\right)\right\}_n \cup \left\{\left(\mathbf{x}_n, y=0, \text{weight} = \frac{1-s_n}{1-p_{T,+}}\right)\right\}_n$
6: $M \leftarrow \mathcal{A}(D')$
7: **return** M

sample size, i.e., $N_+ < N_{T,+}$ for some proper $T > 0$, as is described in Section 3.4. By increasing the effective positive sample size $N_{T,+}$ and rebalancing the effective total costs of each class, we can reduce the imbalance of cost parameters $C_{T,+}$ and $C_{T,-}$, which results in the reduction of the estimation variance as we prove theoretically in Section 3.4.2.

3.3.2 Comparison with the generative modeling approach

In this section, we explain the relationship between the positivity z and the conditional probability $p(y|x)$ and clarify the reason why the naive generative modeling approach is not always suitable.

In a similar and well-studied setting called learning on probabilistic labels, the conditional probability $p_n := p(y = 1|x_n)$ or its estimation is given as the label for each instance. The probabilistic labels are typically attained by averaging crowd-sourced labels over annotators. For this setting, regression-based [Nguyen et al., 2011a, Peng et al., 2014] and rank-based methods [Nguyen et al., 2011b,a, Xue and Hauskrecht, 2016, 2017] are proposed.

In our setting, the conditional probability is not directly given, but can be expressed as $p(y = 1|x) = \int I(z \geq 0)p(z|x)\mathrm{d}z$. Thus, it might be tempting to model $\hat{p}(z|x)$ and then plug-in as

$$\hat{p}(y = 1|x) = \int I(z \geq 0)\hat{p}(z|x)\mathrm{d}z. \tag{3.3}$$

Then, since the positivity z is a continuous variable, one need never confront the imbalance issue.

However, this indirect modeling of z is not always suitable. For example, regression methods with homoscedastic noise (i.e., $\mathrm{Var}[z|x]$ is assumed constant) fail if the assumption is not satisfied, as with the distribution illustrated in Fig. 3.2(b). In this case, these methods tend to learn a constant model $\hat{p}(z|x) = c$ and the plug-in classification model in (3.3) also ends up in a constant model $\hat{p}(y = 1|x) = c'$, while the true conditional probability $p(y = 1|x)$ is not con-

stant in x. Modeling conditional variance is not always sufficient, either, due to higher moments of $p(z|x)$. We are particularly interested here in the tails of distributions, and, therefore, the higher moments are often dominant for evaluating $p(z \geq 0|x)$. This is why the direct modeling approach is superior in terms of versatility for distributions. The experimental results in Section 3.5.4 also support the versatility of the proposed method.

3.3.3 Choice of the soft labeling function σ and the noise robustness

Here we make a note on the noise in the training data and the choice of the soft labeling function σ. Addressing noise is considered important in the imbalanced classification field [Napierała et al., 2010, Sáez et al., 2015, Natarajan et al., 2017]. Generally speaking, our approach is considered to be relatively robust to noise. That is, when the true positivity $z = 0.1$ is observed as $z_{\text{obs}} = -0.1$ as a result of noise on z, the binary label y changes abruptly from 0 to 1, while the soft label $\sigma(z/T)$ in the proposed method only changes from 0.48 to 0.52 under the temperature $T = 1$. Here, even when the noise is added in the input x, if the degree of noise is small, and if we further assume that the conditional probability $p(y = 1|x)$ is continuous (e.g., in the sense of Lipschitz) in x, it can be regarded as equivalent to a small noise on z.

On the other hand, for the case of severe noise, e.g., a completely negative instance $z = -10$ is sometimes observed as completely positive $z_{\text{obs}} = 10$, the noise robustness of our proposed method is only comparable to that of the conventional cost-sensitive learning. One possible solution for such cases is to incorporate a label smoothing technique in the learning from the binary label setting [Natarajan et al., 2017, Szegedy et al., 2016], in which the label is smoothed from $\{0, 1\}$ to (e.g.) $\{0.05, 0.95\}$. Our approach can incorporate this by, e.g., setting the soft labeling function as $\tilde{\sigma}(z/T) = 0.9\sigma(z/T) + 0.05$. The optimal labeling scheme depends on the joint distribution $p(x, z)$. It is desirable to reduce the variance analyzed in Section 3.4.2 while minimizing the increase in bias analyzed in Section 3.4.3. This direction, i.e., improving the soft labeling function to increase noise robustness, is a promising future work.

3.3.4 Comparison with synthetic oversampling methods

Our proposed method extends the cost-sensitive learning, which is called the algorithm-level approach in the imbalanced classification field [Krawczyk, 2016]. Another well-studied direction is the data-level approach, i.e., synthetic oversampling of positive instances. This direction was pioneered by the synthetic minority over-sampling technique (SMOTE) [Chawla et al., 2002] and has been actively studied [Fernández et al., 2018].

While simple over-sampling of positive instances is equivalent to the cost-sensitive learning at the level of its loss function, SMOTE and its variants are clearly distinguished in that they utilize additional inductive biases. For example, SMOTE treats interpolations of neighboring positive instances as positive, which may reflect the convexity of the support of conditional distribution $p(x|y = 1)$ or the cluster assumption [Chapelle et al., 2006]. Also, Ali-Gombe and Elyan [2019] proposed generating positive instances by training a generative adversarial network (GAN) for image data. GANs can incorporate with unlabeled instances for generating realistic images, which highlights a new approach of semi-supervised learning for imbalanced classification. It has been suggested that GANs can utilize some kind of inductive bias common to images [Zhao et al., 2018].

While data augmentation methods have been repeatedly shown to be promising, careful consideration should be given as to whether the inductive biases behind them are still valid in our problem setting. A significant difference may come from the direction of causality. Our typical setting is prediction, i.e., the input feature x causes the outcome y with positivity z observed as a mediator variable as in Fig. 3.1. This is called a causal setting, as opposed to an anti-causal setting, where the label y causes the feature or image x. Schölkopf et al. [2012] have revealed that incorporating the cluster assumption by semi-supervised learning can be helpful only in anti-causal settings. In causal settings, the marginal distribution of the feature $p(x)$ contains no information about the conditional distribution $p(y|x)$. In fact, our experimental results in Section 3.5.4 also show that SMOTE only achieves comparable or inferior performance for cost-sensitive learning.

The inductive bias we are utilizing is in a different direction from this data augmentation approach in the input space. As we analyze in Section 3.4.3, we assume that the larger positivity values indicate the larger possibility of being positive, which reflects a kind of continuity assumption of the conditional distribution in the positivity space. Therefore, our approach may not only be effective for the settings where SMOTE and its variant are not effective but may also incorporate with them. Investigating the key success factor of these synthetic oversampling methods and extend them to prediction or regression problems is a promising direction as discussed in Krawczyk [2016].

3.4 Theoretical Analysis

In this section, we describe the performance of the proposed method, which includes the conventional cost-sensitive learning method as a special case.

3.4.1 Setup

We analyze the excess risk, i.e., the difference in the expected risks of estimated and optimal models, using the population version of the proposed loss (3.2)

$$L_T(g) = \mathop{\mathbb{E}}_{x,z}\left[C_{T,+}\sigma(z/T)\ell(g(x)) + C_{T,-}\sigma(-z/T)\ell(-g(x))\right] \tag{3.4}$$

and the cost-sensitive one

$$L(g) = \mathop{\mathbb{E}}_{x,y}\left[C_+ y\ell(g(x)) + C_-(1-y)\ell(-g(x))\right]. \tag{3.5}$$

When ℓ is the hinge loss or the negative log-likelihood, (3.5) can be seen as a tight convex upper bound of (const. $-$ WA) [Dmochowski et al., 2010], and thus good performance is expected asymptotically. Although, when the size of the positive sample is small and its weight C_+ is set large, the estimation variance is high. Our proposed loss (3.4) treats near-miss instances as being partly positive through soft-labeling function σ, and relaxes the imbalance between the class weights, resulting in reduced estimation variance, as we prove in this section.

The excess risk with respect to the cost-sensitive loss (3.5) can be decomposed as

$$\begin{aligned}\mathop{\mathbb{E}}_{S}[L(\hat{g}) - L(g^*)] = &\underbrace{\mathop{\mathbb{E}}_{S}[L(\hat{g}) - L_T(\hat{g})]}_{\text{bias 1}} + \underbrace{\mathop{\mathbb{E}}_{S}[L_T(\hat{g})] - \min_{g\in\mathcal{G}} L_T(g)}_{\text{variance}} \\ &+ \underbrace{\min_{g\in\mathcal{G}} L_T(g) - L_T(g^*)}_{\leq 0 \text{ by definition}} + \underbrace{L_T(g^*) - L(g^*)}_{\text{bias 2}},\end{aligned} \tag{3.6}$$

where S is the training set, $\hat{g} \coloneqq \arg\min_{g\in\mathcal{G}} \hat{L}_T(g)$ is the empirical proxy loss minimizer, which depends on S, and $g^* \coloneqq \arg\min_{g\in\mathcal{G}} L(g)$ is the optimal model in assumed model class $\mathcal{G}$.

Although the proposed method is model-agnostic, we add some technical assumptions here for theoretical analysis.

Assumption 1. *$\mathcal{G}$ is a bounded linear class; namely, $\mathcal{G} = \{g : g(x;w) = w^\top x, \|w\|_2 \leq B\}$.*

Assumption 2. *The support of $p(x)$ is bounded; namely, $p(\|x\|_2 \leq X) = 1$.*

Assumption 3. *ℓ is 1-Lipschitz and satisfies $\max_{a,a'\in[-BX,BX]} |\ell(a) - \ell(a')| \leq c$.*

In addition, we replace the cost parameter settings with the population versions:

$$C_{T,+} = C_+\frac{p_+}{p_{T,+}} \text{ and } C_{T,-} = C_-\frac{p_-}{p_{T,-}}, \tag{3.7}$$

where p_+ and p_- are the expected positive and negative rates, $p_{T,+}$ and $p_{T,-}$ are the expected effective rates of positive and negative, namely, $p_{T,+} = \mathbb{E}_z[\sigma(z/T)]$ and $p_{T,-} = \mathbb{E}_z[\sigma(-z/T)]$. This is because the expectation $\mathbb{E}_S[N_+/N_{T,+}]$ may not exist. Similarly, when we discuss the BER minimization setting in cost-sensitive learning, we set the cost parameters as

$$C_+ = \frac{1}{2p_+} \text{ and } C_- = \frac{1}{2p_-}. \tag{3.8}$$

3.4.2 Variance reduction

Let us first evaluate the excess risk for our proxy loss, which is denoted as variance in (3.6).

Theorem 3.4.1 (Proxy loss minimization bound)**.** *Let $\hat{w}$ be a minimizer of the empirical proxy loss $\hat{L}_T$ (3.2) with cost parameters (3.7) and w_T^* be a minimizer of the expected proxy loss L_T. Suppose that $\mathcal{G}$, $p(x)$ and ℓ satisfy Assumptions 1–3. The excess risk for L_T will then be bounded as follows:*

$$\mathop{\mathbb{E}}_{S}[L_T(\hat{w}_T) - L_T(w_T^*)] \leq \frac{2BX}{\sqrt{N}} \sqrt{C_+^2 \frac{p_+^2}{p_{T,+}} + C_-^2 \frac{p_-^2}{p_{T,-}}}.$$

This is given by element-wise upper bounding of the Rademacher complexity, i.e.,

$$\begin{aligned} R(\ell \circ A) \coloneqq & R(\{(\ell_1(a_1), \ldots, \ell_N(a_N)) : \mathbf{a} \in A \subset \mathbb{R}^N\}) \\ \leq & R(\{(\rho_1 a_1, \ldots, \rho_N a_N) : \mathbf{a} \in A\})), \end{aligned}$$

where $\mathbf{a} \coloneqq (a_1, \ldots, a_N)$ and ρ_n is the Lipschitz constant of ℓ_n. The detailed proof is given in the appendix. This element-wise evaluation of the Lipschitz constants is the key for a tighter bound since our loss function consists of a small number of element-wise losses that have a large Lipschitz constant $C_{T,+}$ and a large number of one with a small Lipschitz constant $C_{T,-}$.

For the BER minimization setting (3.8), the bound is rewritten as follows.

Corollary 3.4.1.1 (Balanced loss minimization bound)**.** *Under the same condition with Theorem 3.4.1, the excess risk for L_T with the parameters* (3.8) *is bounded as*

$$\mathop{\mathbb{E}}_{S}[L_T(\hat{w}) - L_T(w_T^*)] \leq \frac{BX}{\sqrt{N}} \sqrt{\frac{1}{p_{T,+}} + \frac{1}{p_{T,-}}}. \tag{3.9}$$

So long as the effective positive rate is much smaller than the effective negative rate, namely, $p_{T,+} \ll p_{T,-}$, the term $1/p_{T,+}$ is dominant in (3.9). This is why reducing the imbalance between $p_{T,+}$ and $p_{T,-}$ has a critical impact on the

variance reduction. From the definition of the soft-label $\sigma(z/T)$, we observe

$$\lim_{T\to 0} p_{T,+} \to p_+ \text{ and } \lim_{T\to\infty} p_{T,+} \to \frac{1}{2}.$$

Therefore, by using proper $T > 0$, we can increase the effective positive rate $p_{T,+}$, and can attain variance reduction.

Corollary 3.4.1.1 is also useful to predict the limitation of conventional cost-sensitive learning. Let us assume that $T \to 0$ (then $Np_{T,+} \to Np_+ \simeq N_+$), the model complexity $B = 1$, and the size of the feature space $X = \sqrt{d}$ (each dimension is normalized). Since $p_+ \ll p_-$ holds, the r.h.s. of (3.9) would be simplified as follows:

$$\text{r.h.s. of (3.9)} \simeq \sqrt{\frac{d}{N_+}}. \tag{3.10}$$

Therefore, when the size of the positive sample is smaller than the feature dimension ($N_+ < d$), the variance term would be larger than 1, which is no longer meaningful as an upper bound of the BER. Assuming the bound is tight enough, this implies that there is plenty of room for performance improvement by tuning T when $N_+ < d$ holds, and also experimental results in Section 3.5.3 agree to this. That is, the conventional cost-sensitive method significantly underperforms the proposed method when $N_+ < d$.

3.4.3 Bias bound

We next give an upper bound of the bias terms in (3.6). To simplify the notation, we introduce a random variable η that depends on the soft label $\sigma(z/T)$ as

$$p(\eta|z) \coloneqq mathrmBernoulli(\sigma(z/T)).$$

The random variable η can be seen as "a potential label that might have been under the given z," and $p(\eta = y) = 1$ when $T \to 0$. By using η, we can bound the bias as follows:

Proposition 3.4.2 (Bound of the bias of the proxy loss)**.** *Suppose that $\mathcal{G}$, $p(x)$, and ℓ satisfy Assumptions 1–3. The bias terms in (3.6) in the BER minimization setting (3.8) is upper-bounded as*

$$\begin{aligned}(\text{bias } 1 + \text{bias } 2) \leq c\big\{&\mathrm{TV}\left(p(x|\eta = 1), p(x|y = 1)\right) \\ &+ \mathrm{TV}\left(p(x|\eta = 0), p(x|y = 0)\right)\big\},\end{aligned}$$

where $\mathrm{TV}(p(x), q(x)) \coloneqq \frac{1}{2}\int |p(x) - q(x)|\,\mathrm{d}x$ *is the total variation distance.*

If we set $T > 0$, the bias might increase, which is bounded using the TV

distances, and which depends on the joint distribution $p(x, z)$ and the temperature T. Differently from the distance between the conditional label probabilities $\mathrm{TV}(p(y|x), p(\eta|x))$, these TV distance terms do not necessarily increase as do $p_{T,+} = p(\eta = 1)$. Thus, in the range of reasonably small T, and provided that a reasonable z is given such that $\sigma(z/T)$ is highly correlated to $p(y = 1|x)$, the proposed method attains reasonable variance reduction in exchange for additional bias. Conversely, if z has no additional information to y, that is, for example, z is determined by y as $z = 2y - 1$, the TV distance terms immediately increase when $T > 0$, and we cannot attain significant variance reduction. Note that the soft-label itself need not necessarily be a good estimator of $p(y = 1|x)$, which is a difference from the probabilistic label $p_n = p(y = 1|x_n)$.

3.4.4 Connection to the learning using privileged information (LUPI)

Learning using privileged information (LUPI) is a general problem setting that aims to utilize additional information like z. Privileged information was first proposed in Vapnik and Vashist [2009], in which it was assumed that additional features were provided for each training instance and that the features were strongly related to the label but not available in the test phase. They argued that a faster learning rate could be obtained by using privileged information to estimate the slack variables in the SVM. Generalized distillation (GD) [Lopez-Paz et al., 2016] enables model-agnostic learning with privileged information using a similar procedure to the distillation [Hinton et al., 2015]. The basic procedure of GD is to first learn a "teacher model" $g_t(z)$ from the privileged features $z \in \mathbb{R}^m$ and the original labels, and then learn a "student model" with the original features $x \in \mathbb{R}^d$ and soft-labels given by the teacher model using the following proxy loss*:

$$\hat{L}_T^{\mathrm{GD}}(g) = \frac{1}{N} \sum_n^N \left\{ \sigma(g_t(z_n)/T)\ell(g(x_n)) + \sigma(-g_t(z_n)/T)\ell(-g(x_n)) \right\}.$$

While those methods are aimed at fast learning rates in terms of the sample size, we utilize soft-labels given by a similar procedure for lessening the imbalance. To the best of our knowledge, this is the first work that utilizes privileged information for imbalanced classification problems. Without cost rebalancing in (3.7), GD cannot attain the variance reduction analyzed in Section 3.4.2. The key advantage of privileged information in the application to the class-imbalanced problems comes from the reduction of the instance-wise Lipschitz constants by rebalanced costs, which highlights a new aspect of LUPI.

*In the original paper, they used a mixed label of the true label y and the teacher label $\sigma(g_t/T)$, by means of a so-called imitation parameter λ. We do not need λ since positivity z includes the whole information of y.

3.5 Experiments

In this section, on the basis of extensive experiments on synthetic and real datasets, we demonstrate the characteristics and the performance of the proposed method.

3.5.1 Setup

Here we describe experimental settings briefly.

Datasets

Since our method (as do regression and rank-based methods) requires positivity z, we used datasets originally designed for use in regression problems. Each dataset has a numerical target attribute, which we regarded as positivity z, and we set the task-specific threshold θ such that the top-$100p_+\%$ would be positive.

Evaluation

We used balanced accuracy (BA) for the performance evaluation, as explained in Section 3.2. For the regression-based methods, we applied the original threshold to the prediction to evaluate BA, i.e., $\hat{y} = I(\hat{z} \geq 0)$. For the rank-based method, we set θ such that the top $100p_+\%$ predicted scores would be positive.

In the experiments in Section 3.5.3 and Section 3.5.4, we evaluated BA using nested cross-validation [Varma and Simon, 2006]. The outer cross-validation loop was 5-fold, and the inner one for hyperparameter selection was 2-fold. For the Gaussian process (GP), we applied the maximum likelihood estimation for hyperparameter selection to avoid heavy computation. In both training and test data, the ratio of positive and negative sample was maintained, i.e., stratified sampling was performed. We repeated this process four times, changing the split of the outer loop (thus, there were 20 results for the test data).

3.5.2 Performance variation in temperature T

First, we investigated the effect of introducing a soft-label using the hyperparameter T. Since a soft-label with a small T goes to a hard label y, the change in metrics for various T values demonstrates the benefit of utilizing positivity information. We used the toy data in Fig. 3.2(a) and logistic regression with l2 and l1 regularizers. The regularization strength was fixed to 1.0.

Results with respect to BA are shown in Fig. 3.3. The best T is neither zero nor infinity, which indicates the variance reduction in small T and the bias increase in large T. The difference between the best performance and the performance in $T \to 0$ illustrates the benefit of introducing the soft-label. Also,

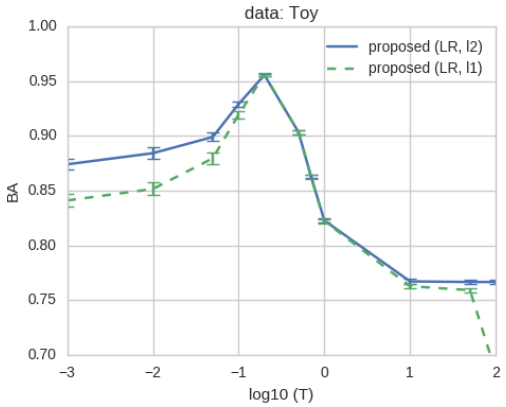


Figure 3.3: Performance in various T on toy data in Fig. 3.2(a). As shown, there exist here some moderate temperatures that perform better than $T \to 0$ or $T \to \infty$.

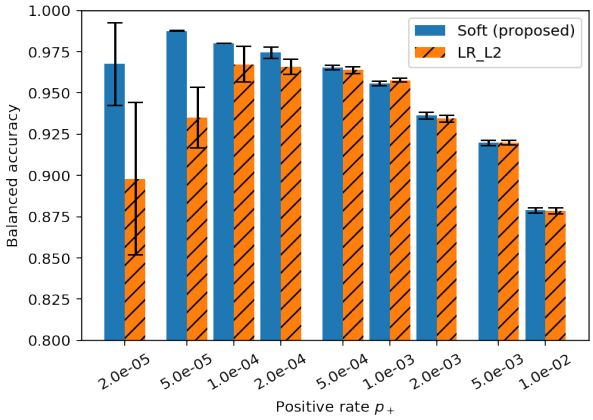


Figure 3.4: Performance of the proposed method and conventional cost-sensitive classification with respect to BA for the GPU kernel performance dataset under highly imbalanced conditions. Positive rate $p_+ := \sum I(z \geq 0)/N$ ranged from 2×10^{-5} to 1×10^{-2}. Error bars indicate standard error.

$T \to \infty$ means treating near-miss and far-miss, i.e., the other negative instances equally, which induces a large bias as analyzed in Section 3.4.3 and degrades the performance. This illustrates the importance of treating only near-miss instances as being partly positive.

3.5.3 Comparison with conventional classification under highly imbalanced conditions

To demonstrate the benefit of our method under highly imbalanced conditions, we compared it with conventional cost-sensitive learning for various positive rates p_+ (and thus N_+). The base learner was a logistic regression model with the L2 regularizer.

We used the GPU kernel performance dataset [Nugteren and Codreanu, 2015, Ballester-Ripoll et al., 2017], which is a large-scale dataset with real-valued tar-

get attributes. It had 14 features of GPU kernel parameters and four target attributes of elapsed times in milliseconds for four independent runs under the same parameters, and the number of instances was 241.6k. We transformed the problem for elapsed time regression into a classification for finding good parameters, i.e., we used the average speed $z = \frac{4}{\sum y_i}$, where $\{y_i\}_{1:4}$ are the original elapsed times.

The results given in Fig. 3.4 show that the conventional cost-sensitive logistic regression worsened when highly imbalanced, while the proposed method worked well. The performance gap is particularly large when $p_+ \leq 5 \times 10^{-5}$, which means the size of the positive training sample $N_+ \leq 10 < d$. This is in good agreement with the theoretical prediction in (3.10). The results with respect to AUC in the same setting and the results in fixed p_+ and various N are also shown in the appendix, which presents similar trends.

3.5.4 Comparison with various baseline methods and datasets

We are also able to demonstrate the versatility of our proposed method for various datasets. Positive rate p_+ was fixed to 5% since the sample sizes are not so large in most of the datasets we prepared. We compared the proposed method with three base learners (logistic regression with L1 and L2 regularizers, and SVM with an RBF kernel) and baseline methods, namely, the conventional cost-sensitive classification, oversampling-based classification (SMOTE) [Chawla et al., 2002], undersampling ensemble classification (RUSBoost) [Seiffert et al., 2009], regression-based methods (ridge, lasso, and GP with an RBF kernel), and Rank-SVM (with a linear kernel, as proposed in Xue and Hauskrecht [2016]).

Table 3.1 and Table 3.2 show the overall results with respect to BA and ROC-AUC, respectively. Also, Fig. 3.5 summarizes the comparison between our proposed method and existing classification and regression-based approaches. Our approach outperformed or was at least comparable to the regression and the rank-based baselines for properly chosen base learners, while regression-based approaches failed for some data including Diabetes and Puma32H. The student performance dataset had a quite limited number of instances for its dimensions, which may be a reason why the regression-based baseline worked better.

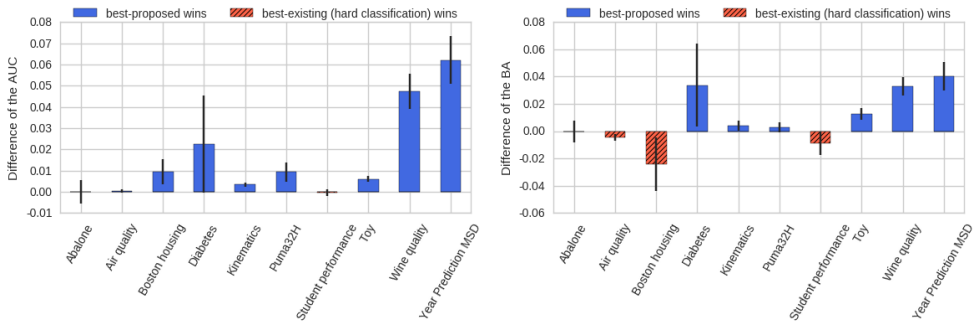
To investigate the performance on high-dimensional and large-scale data, we also employed the GPU dataset used in Section 3.5.3 with an expanded binary feature set of up to second-order interaction terms of the original features. The resulting number of features was 335. Due to the large data size ($N = 241,600$), we compared methods excluding the kernel-based and pairwise ranking-based methods. The performance comparison under various positive rate is shown in Table 3.3. The resulting performance illustrates that our proposed method outperforms baseline methods in a highly imbalanced setting ($p_+ = 0.005\%$) and is

Table 3.1: Balanced accuracy (3.1) for the experiment in Section 3.5.4. The best method is in bold, and the second place is in italic and underlined. Datasets indicated by (*1) are from [Dheeru and Karra Taniskidou, 2017], (*2) is from [Efron et al., 2004] and (*3) are from [Torgo, 2018]. The positive rate was fixed to 5% (the imbalance ratio of 1:19).

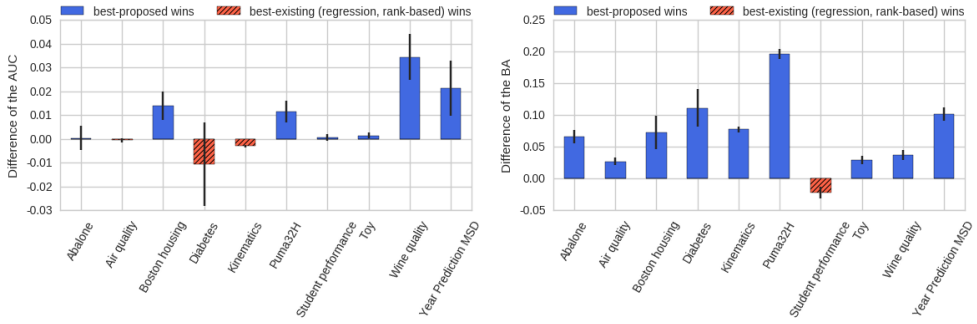
Dataset			Cost-sensitive classification			SMOTE			Ensemble	Regression-based			Order-based	Proposed		
Name	N	d	LR (l1)	LR (l2)	SVM	LR (l1)	LR (l2)	SVM	RUSBoost	Lasso	Ridge	GP	Rank-SVM	LR (l1)	LR (l2)	SVM
Abalone (*1)	4177	9	0.835	0.836	0.827	**0.837**	*0.837*	0.820	0.724	0.566	0.769	0.565	0.671	0.833	0.835	0.833
Air quality (*1)	6941	11	**0.963**	0.959	0.946	0.958	0.958	0.950	0.870	0.877	0.932	0.912	0.844	0.958	*0.959*	0.956
Boston housing (*1)	506	13	0.895	0.909	**0.942**	0.892	0.900	0.885	0.892	0.519	0.845	0.677	0.834	*0.917*	0.914	0.886
Diabetes (*2)	442	10	0.745	0.736	0.728	0.755	0.743	0.727	0.608	0.501	0.667	0.500	0.605	**0.778**	*0.757*	0.734
Kinematics (*3)	8192	8	0.859	0.861	*0.947*	0.859	0.863	0.935	0.858	0.500	0.735	0.873	0.714	0.863	0.864	**0.951**
Puma32H (*3)	8192	32	0.868	0.866	0.875	0.876	0.873	0.862	**0.921**	0.500	0.681	0.507	0.632	0.868	0.867	*0.878*
Student performance (*1)	1044	43	0.959	0.895	0.929	*0.964*	0.914	0.878	0.952	**0.972**	0.952	0.953	0.878	0.950	0.941	0.881
Wine quality (*1)	6497	12	0.724	0.732	0.729	0.724	0.725	*0.737*	0.702	0.500	0.728	0.680	0.563	0.724	0.736	**0.765**
Year Prediction MSD (*1)	10000	90	0.613	0.612	0.598	0.622	0.621	0.583	0.556	0.515	0.552	0.511	0.504	*0.652*	**0.654**	0.642
Toy (synthetic data)	3000	100	0.944	0.950	0.932	0.938	0.940	0.621	0.705	0.585	0.934	0.579	0.863	**0.963**	*0.962*	0.899

Table 3.2: ROC-AUC for the experiment in Section 3.5.4. The best method is in bold, and the second place is in italic and underlined.

Dataset			Cost-sensitive classification			SMOTE			Ensemble	Regression-based			Order-based	Proposed		
Name	N	d	LR (l1)	LR (l2)	SVM	LR (l1)	LR (l2)	SVM	RUSBoost	Lasso	Ridge	GP	Rank-SVM	LR (l1)	LR (l2)	SVM
Abalone (*1)	4177	9	0.916	0.916	0.912	**0.916**	0.916	0.910	0.809	0.894	0.906	0.915	0.915	0.916	*0.916*	0.906
Air quality (*1)	6941	11	0.991	0.991	0.991	0.991	0.991	0.991	0.956	0.991	0.992	**0.992**	0.986	*0.992*	0.991	0.989
Boston housing (*1)	506	13	*0.981*	0.950	0.976	0.952	0.949	0.959	0.966	0.946	0.977	0.928	0.976	0.959	0.940	**0.991**
Diabetes (*2)	442	10	0.826	0.833	0.805	0.816	0.824	0.810	0.714	**0.866**	*0.860*	0.572	0.833	0.853	0.855	0.819
Kinematics (*3)	8192	8	0.932	0.933	0.986	0.932	0.933	0.986	0.929	0.919	0.930	**0.993**	0.932	0.932	0.933	*0.990*
Puma32H (*3)	8192	32	0.908	0.900	0.910	0.907	0.902	0.900	**0.971**	0.908	0.903	0.907	0.898	0.908	0.902	*0.920*
Student performance (*1)	1044	43	**0.995**	0.982	0.981	0.994	0.981	0.980	0.985	0.994	0.994	0.992	0.993	*0.995*	0.993	0.972
Wine quality (*1)	6497	12	0.799	0.801	0.802	0.799	0.801	0.806	0.797	0.790	0.805	*0.815*	0.801	0.796	0.800	**0.850**
Year Prediction MSD (*1)	10000	90	0.654	0.652	0.635	0.684	0.675	0.636	0.582	0.673	0.692	0.695	0.605	*0.702*	0.702	**0.716**
Toy (synthetic data)	3000	100	0.988	0.988	0.975	0.986	0.987	0.975	0.782	0.993	0.983	0.989	0.987	*0.994*	**0.994**	0.985



(a) Proposed "soft" method vs. conventional "hard" cost-sensitive classification



(b) Proposed method vs. regression- and rank-based methods

Figure 3.5: Pairwise accuracy comparison between existing approaches and ours. The Y-axes indicate the differences in BA and AUC between the best performance of the proposed methods (with three base learners) and that of the baseline methods (cost-sensitive classification, regression-based and rank-based methods with seven base learners in total). Error bars indicate standard errors. Our soft-classification approach outperforms or at least compares favorably in most of the datasets and metrics, while existing approaches (cost-sensitive hard classification, regression- and rank-based methods) significantly underperform in some datasets or metrics.

Table 3.3: Balanced accuracy comparison in the large-scale dataset (GPU kernel performance). The best method is in bold, and the second place is italic and underlined.

Dataset	Cost-sensitive classification		SMOTE		RUSBoost	Regression-based		Proposed	
	LR (l1)	LR (l2)	LR (l1)	LR (l2)		Lasso	Ridge	LR (l1)	LR (l2)
GPU-interaction-0.1%	<u>*0.961*</u>	0.959	0.958	0.959	**0.969**	0.500	0.500	0.951	0.951
GPU-interaction-0.005%	0.894	0.878	0.887	0.898	0.968	0.505	0.505	<u>*0.986*</u>	**0.986**

comparable in a mildly imbalanced setting ($p_+ = 0.1\%$, which means the positive sample size of $N_+ = 242$, equivalent to the dimension in terms of order). When the positive rate is 0.1%, the positive sample size would be $N_+ = 241$, which is about the same as the dimensions. Thus, it is thought that the variance was not dominant when the linear model with a regularizer was used, and the difference from the cost-sensitive learning did not appear. It is again confirmed that the proposed method performs as well as the cost-sensitive learning when the estimated variance is not dominant and improves on the cost-sensitive learning when

the sample size of positive examples is small, and the estimated variance becomes dominant.

3.6 Summary

In this chapter, we have introduced a novel problem setting, imbalanced classification with positivity, and proposed a versatile method for dealing with it, which highlighted the usefulness of the positivity information. The key advantage of our method is exploiting near-miss positive instances, which are specified by positivity, to lessen the class imbalance.

We have investigated the loss theoretically for the proposed method and for conventional cost-sensitive learning in consideration of the degree of imbalance, and have shown that our method lessens the imbalance with the help of near-miss positive instances. Extensive experiments have illustrated that our method outperforms the conventional cost-sensitive classification under highly imbalanced conditions and is more versatile than are existing regression or rank-based approaches.

Chapter 4

Causality-aware Utility Modeling

4.1 Introduction

Predicting individualized causal effects is an essential issue in many domains for decision-making. For example, a doctor considers which medication would be the most effective for a patient, a teacher finds which problems are most effective for helping students learn, and a retail store manager considers which assortment of items would improve the overall store sales. To support such decision-making, we advocate providing a prediction of which actions will lead to better outcomes.

Recent efforts in causal inference and counterfactual machine learning have focused on making predictions of the potential outcomes that correspond to each action for each individual target on the basis of observational data. Observational data consists of features of targets, past actions actually taken, and their outcomes. We have no direct access to the past decision-makers' policies, i.e., the mechanism of how to choose an action under a given target feature. Unlike in normal prediction problems, pursuing high-accuracy predictions only with respect to the historical data carries the risk of incorrect estimates due to the sampling bias in the past policies. This bias may cause *spurious correlation* [Simon, 1954, Pearl, 2009], which might mislead the decision-making. For those cases where real-world experiments such as randomized controlled trials (RCTs) or multi-armed bandit are infeasible or too expensive, causal inference methods provide debiased estimation of potential outcomes from observational data.

While most of the existing approaches assume limited action spaces such as a binary one, as in conditional average treatment effect (CATE) estimation, there are many real-world situations where the number of options is large. For example, doctors need to consider which combination of medicines will best suit a patient.

For such cases, it is difficult to apply existing methods (as in [Shalit et al., 2017, Yoon et al., 2018, Schwab et al., 2018, Lopez et al., 2020]) for two rea-

This chapter is based on Akira Tanimoto, Tomoya Sakai, Takashi Takenouchi, and Hisashi Kashima, Regret minimization for causal inference on large treatment space, in International Conference on Artificial Intelligence and Statistics (AISTATS), 2021.

sons. One is the issue of sample-efficiency for large action spaces. Since the sample sizes for each action would be limited, building models for each action (or using a multi-head neural network), which existing methods adopt, is not sample-efficient. The other reason is the gap between the decision-making performance and the regression accuracy of the potential outcome. Even if we manage to achieve the same level of regression accuracy as when the action space is limited, the same decision-making performance is no longer guaranteed in a large action space, as we demonstrate in Section 4.4. This is because, in a nutshell, the over-estimated potential outcome of only a single action may mislead the decision, even though it has only a small impact on the mean regression accuracy over all actions.

To achieve informative causal inference for decision-making in a large action space, we propose solutions for the above two issues. For the sample-efficiency, we propose extracting representations not only from features but also from actions. We extend two existing representation-based causal effect inference methods, respectively, to balance the representation distribution to be similar to that in the randomized trials.

For the gap between the decision performance and the regression accuracy, we prove that we can further improve the decision performance by minimizing the classification error of whether or not each action is relatively good for each target, in addition to the regression error (MSE). Unlike the recommendation problems in which ranking losses can be used, we cannot directly observe whether the action is relatively good or not since only one action and its outcome is observed for each target. We therefore propose a proxy loss that compares the observed outcome to the estimated conditional average performance of the past decision-makers, which is estimated by regular supervised learning.

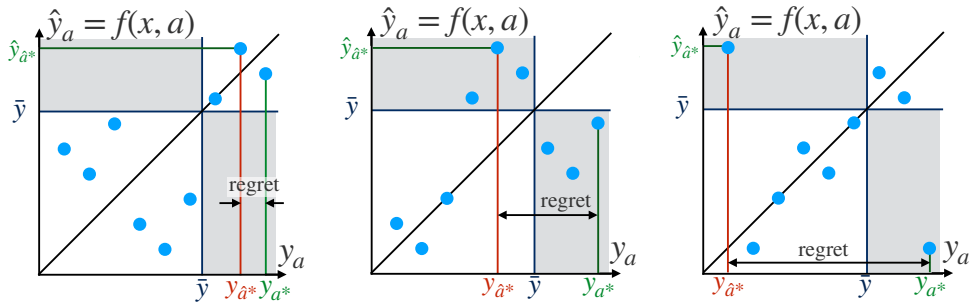
In summary, our proposed method minimizes both the classification error and the MSE by using debiased representations of both the features and the actions. We demonstrate the effectiveness of our method through extensive experiments with synthetic and semi-synthetic datasets.

4.2 Problem Setting

In this section, we formulate our problem and define a decision-focused performance metric. Our aim is to build a predictive model to inform decision-making. Given a feature vector $x \in \mathcal{X} \subset \mathbb{R}^d$, the learned predictive model f is expected to correctly predict which action $a \in \mathcal{A}(x)$ leads to a better outcome $y \in \mathcal{Y} \subset \mathbb{R}$, where $\mathcal{A}(x)$ is a feasible subset of a finite action space $\mathcal{A}$ given x. We hereafter assume that the feasible action space does not depend on the feature, i.e., $\mathcal{A}(x) = \mathcal{A}$, for simplicity. A typical case of large action spaces is when an action

x	a		Y_a								y
		a_0	0				1				
		a_1	0		1		0		1		
		a_2	0	1	0	1	0	1	0	1	
x_1	$(0,0,1)$		–	1	–	–	–	–	–	–	1
x_2	$(0,1,0)$		–	–	3	–	–	–	–	–	3
x_3	$(0,0,0)$		4	–	–	–	–	–	–	–	4
x_4	$(1,0,1)$		–	–	–	–	–	6	–	–	6

Figure 4.1: An example data table for our causal inference on a combinatorial action space. Dashes indicate missing entries. Only factual outcomes are observed (when $a = a'$, $y_{a'}$ is observed) and the counterfactual records $\{y_a\}_{a \neq a'}$ are missing.



(a) Zero ER^u_μ (b) High ER^u_μ (c) Low ER^u_μ, high MSE^u

Figure 4.2: Example scatter plots of true vs. predicted potential outcomes for a target (fixed x) for different models. Each point corresponds to an action. ER^u_μ corresponds to the rate of instances in the shaded areas. Assuming that the predicted best action $\hat{a}^* := \arg\max_a f(x,a)$ is adopted, minimizing the difference between its outcome $y_{\hat{a}^*}$ and the true optimal outcome y_{a^*} (regret) is our aim (see the definition in Section 4.4).

consists of multiple causes, i.e., $\mathcal{A} = \{0,1\}^m$ (combinatorial action space).

With an unknown policy of past decision-makers $\mu(a|x)$, which is a conditional distribution called propensity, we assume there exists a joint distribution $p(x, a, y_1, \ldots, y_{|\mathcal{A}|}) = p(x)\mu(a|x)p(y_1, \ldots, y_{|\mathcal{A}|}|x)$, where $y_1, \ldots, y_{|\mathcal{A}|}$ are the potential outcomes corresponding to each action. The observed (factual) outcome y is the one corresponding to the observed action a, i.e., a training instance is the triplet (x_n, a_n, y_{a_n}), where n denotes the instance index, and the other (counterfactual) potential outcomes are regarded as missing, as shown in Fig. 4.1.

We make the following assumptions on the distributions of the observational data.

- $(y_1, \ldots, y_{|\mathcal{A}|}) \perp a|x$ (unconfoundedness)
- $\forall a \in \mathcal{A}$ and $\forall x$, $0 < \mu(a|x) < 1$ (overlap)

These are commonly required to identify causal effects [Imbens and Wooldridge, 2009, Pearl, 2009].

4.3 Regret Minimization Network: Debiased Potential Outcome Regression and Classification

For this problem of estimating the action evaluation model, we propose our regret minimization network (RMNet), which consists of two parts: 1) a decision-focused risk to reduce the gap between the decision-making performance and the regression accuracy, and 2) representation balancing methods for debiased and sample-efficient learning.

4.3.1 Decision-focused risk

Most of the existing causal effect inference methods aim at minimizing the MSE of the treatment effect (a.k.a. the precision in the estimation of heterogeneous effect (PEHE) [Hill, 2011]) in the binary treatment setting. In multiple treatment settings, a typical performance measure is the MSE averaged uniformly over all the actions [Schwab et al., 2018, Yoon et al., 2018]:

$$\mathrm{MSE}^u(f) \coloneqq \mathop{\mathbb{E}}_{x}\left[\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\mathop{\mathbb{E}}_{y_a|x}\left[(y_a - f(x,a))^2\right]\right]. \tag{4.1}$$

We refer to MSE^u as MSE, or specifically the uniform MSE, in this chapter.

On the other hand, there is a gap between the decision performance and the regression accuracy (MSE^u). Specifically, we do not necessarily have to accurately estimate the outcomes of candidate actions, but it is enough to identify better actions among others to achieve a higher decision-making performance. This is analogous to the personalized ranking approach in recommender systems [Rendle et al., 2009], in which pairwise comparison of the item preference for each target user is considered.

The pairwise ranking approach [Joachims, 2002, Burges et al., 2005] measures the consistency between the actual and predicted orders by means of the error rate of pairwise comparison as

$$\mathrm{ER}_{\mathrm{rank}}(f) = \mathop{\mathbb{E}}_{i,j}\left[I(y_i \geq y_j \oplus f(x_i) \geq f(x_j))\right],$$

where $\oplus$ denotes the logical XOR. However, we cannot apply a regular pairwise loss, since we typically only have the outcome for one action observed among the feasible actions. Instead, we propose minimizing the following comparison loss to the average performance of the past decision-makers as the personalized baseline for the target (x):

$$\mathrm{ER}^u_\mu(f) \coloneqq \mathop{\mathbb{E}}_{x}\left[\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} I(y_a \geq \bar{y} \oplus f(x,a) \geq \bar{y})\right], \tag{4.2}$$

where $\bar{y} = \mathbb{E}_{a\sim\mu(a|x)}[Y_a|x]$ is the average performance of the past decision-makers under x. As shown in Fig. 4.2, minimizing ER^u_μ leads to better models in terms of decision performance. The MSE is the same in Fig. 4.2(a) and Fig. 4.2(b), and thus MSE cannot be used to determine which of these prediction models is better. Minimizing ER^u_μ enables us to correctly choose the model in Fig. 4.2(a) with a high decision performance (small regret).

Replacing the expected value $\bar{y}$ with its estimation $g(x)$ and the 0-1 loss with cross entropy, we get the following risk:

$$\widetilde{\mathrm{ER}}^u_g(f) \coloneqq \mathop{\mathbb{E}}_x\left[-\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}}\{s\log v + (1-s)\log(1-v)\}\right], \tag{4.3}$$

where $s \coloneqq I(y - g(x) \geq 0)$ and $v \coloneqq \sigma(f(x,a) - g(x))$, and $g(x) \simeq \mathbb{E}_{a\sim\mu(a|x)}[Y_a|x]$ is the estimated average performance of the past decision-makers. We first fit g with the standard supervised learning procedure from $\{(x_n, y_n)\}$ and then plug it into (4.3).

Not only the classification error but also the regression error (MSE) matters to the decision-making performance. This is because even with high classification accuracy, decisions might be misleading if only one misclassified action a is predicted as the best ($\hat{y}_a$ is the highest among others $\{\hat{y}_{a'}\}_{a'}$) but is actually quite bad (y_a is quite low), as in Fig. 4.2(c).

Therefore, we propose minimizing a combination of both the regression and classification risks, i.e., the geometric mean of them:

$$L^u(f;g) = \sqrt{\widetilde{\mathrm{ER}}^u_g(f)\cdot \mathrm{MSE}^u(f)}. \tag{4.4}$$

The reason we chose the geometric mean will be explained theoretically in Section 4.4. Intuitively, it is sufficient to make one of these losses small, e.g., if the classification loss is zero, good decisions can be made even if the MSE is large. As shown in Fig. 4.2(a), a model that achieves $\mathrm{ER}^u_\mu = 0$ (thus the geometric mean is also zero) can at least outperform the past decision-makers on average ($y_{\hat{a}^*} \geq \bar{y}$) no matter how large the MSE^u is.

4.3.2 Debiased and sample-efficient learning

While accessible observational data taken from $p(x,a)$ is biased by the propensity $\mu(a|x)$, our target expected risk $L^u(f;g)$ is averaged over all actions uniformly, i.e., $p^u(x,a) = p(x)p^u(a)$, where $p^u(a) = \mathrm{Unif}(\mathcal{A})$ is the discrete uniform distribution. In this section, therefore, we construct two debiasing methods for the sampling bias that performs domain adaptation from $p(x,a)$ to $p^u(x,a)$ as extensions of two existing approaches. Also, we propose network architectures that extract representations from both the feature and the action for better general-

ization in a large action space.

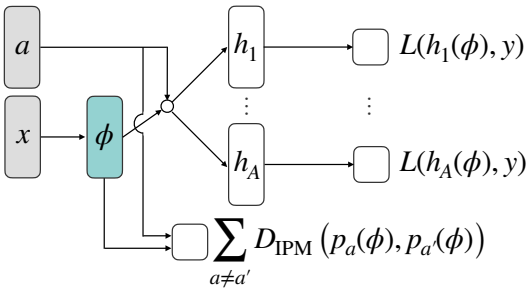
There are two major approaches for debiased learning in individual-level causal inference. One is a density estimation-based method called inverse probability weighting using propensity score (IPW) [Austin, 2011], in which each instance is weighted by the inverse propensity $1/\mu(a_n|x_n)$. Since the expected risk matches that of the RCT, a good performance can be expected asymptotically under accurate estimation of μ or when it is recorded as in logged bandit problems. However, in observational studies where the propensity has to be estimated and plugged in, its efficacy would easily decrease [Kang et al., 2007]. It becomes further difficult when it comes to a large treatment space. Zou et al. [2020] proposed assuming an intrinsic low-dimensional structure for combinatorial treatment assignments (bundle treatments) $a \in \{0,1\}^p$ and estimating weights on the latent space. While in this study we examine a general case of large treatment spaces without additional assumptions, it may be necessary to introduce such assumptions to consider such a huge treatment space of combinatorial interventions.

The other approach is representation balancing [Shalit et al., 2017, Johansson et al., 2016, Lopez et al., 2020], in which a representation extractor of the feature ϕ_x is trained to eliminate the effect of confounding as well as to preserve the relation to the outcome. Shalit et al. [2017], Johansson et al. [2016] proposed regularizing the conditional probabilities of representations $\{p(\phi_x|a)\}_a$ to be similar to each other by means of the integral probability metric (IPM) regularizer [Müller, 1997, Sriperumbudur et al., 2012] (as in Fig. 4.3(a)) for limited action spaces such as the binary space $\mathcal{A} = \{0,1\}$. Lopez et al. [2020] proposed regularizing the representation ϕ_x to be independent from the action a by means of the Hilbert-Schmidt Independence Criterion (HSIC) [Gretton et al., 2005, 2008] for real-valued action space $\mathcal{A} \subset \mathbb{R}$. We extend this approach to large treatment spaces.

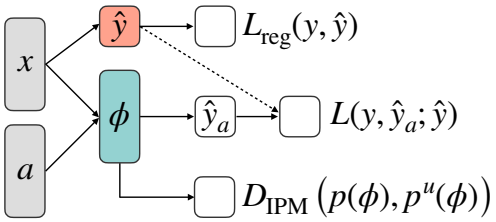
To deal with a large treatment space, we propose performing representation extraction from the treatment a as well as the feature x. RMNet-IPM (Fig. 4.3(b)) extracts the joint representation $\phi_{x,a}$ from x and a, which is regularized to be distributionally similar to that of the RCTs $p^u(\phi_{x,a})$. That is, IPM measures the discrepancy between the distributions

$$p(\phi_{x,a}) \coloneqq \int \sum_{a'} p(\phi_{x,a}|x',a')\mu(a'|x')p(x')\mathrm{d}x',$$
$$p^u(\phi_{x,a}) \coloneqq \int \sum_{a'} p(\phi_{x,a}|x',a')p^u(a')p(x')\mathrm{d}x',$$

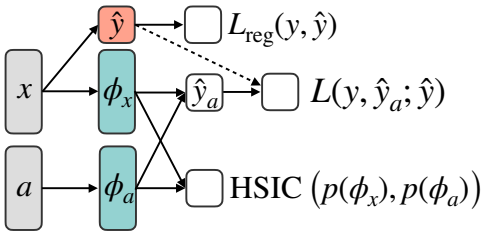
where $p(\phi_{x,a}|x',a') = \delta(\phi_{x,a} - \phi(x',a'))$. IPM is defined for a pair of distributions



(a) Counterfactual regression (CFR)



(b) RMNet-IPM (proposed)



(c) RMNet-HSIC (proposed)

Figure 4.3: Network structures of counterfactual regression for CATE [Shalit et al., 2017, Schwab et al., 2018] and our proposed methods. A broken line indicates no backpropagation.

(p_1, p_2) over $\mathcal{S}$ and a function family G as

$$\mathrm{IPM}_{\mathrm{G}}(p_1, p_2) = \sup_{g \in \mathrm{G}} \left| \int_{\mathcal{S}} g(s)\,(p_1(s) - p_2(s))\,\mathrm{d}s \right|.$$

We adopt the set of 1-Lipschitz functions as G (as in [Shalit et al., 2017]), after which IPM is equivalent to the Wasserstein distance. Specifically, we use an entropy relaxation of the exact Wasserstein distance, called Sinkhorn distance [Cuturi, 2013], to ensure compatibility with the gradient-based optimization. This discrepancy upper-bounds the gap between our target risk (4.4), which is averaged over the uniform distribution with respect to action $p^u(x, a) = p(x)p^u(a)$, and the one of observational distribution $p(x, a)$. Theoretical analysis for this point can be found in Appendix B.1.2.

Note that minimizing the discrepancy between $p(\phi_{x,a})$ and $p^u(\phi_{x,a})$ and preserving the causal relation are not necessarily incompatible. In this sense, our approach, which directly regularizes the representation distribution $p(\phi_{x,a})$ to be similar to that taken from RCTs $p^u(\phi_{x,a})$, provides a weaker and sufficient

condition for this domain adaptation problem. We discuss this point in Appendix B.1.3.

RMNet-HSIC (Fig. 4.3(c)) extracts each representation ϕ_x and ϕ_a from x and a separately, and they are regularized to be independent from each other by minimizing $\mathrm{HSIC}(p(\phi_x), p(\phi_a))$. HSIC can be defined as a special case of the (squared) maximum mean discrepancy (MMD), which is an instance of the IPM with the class of norm-1 reproducing kernel Hilbert space (RKHS) functions, as follows:

$$\mathrm{HSIC}(p(\phi_x), p(\phi_a)) = \mathrm{MMD}^2(p(\phi_x, \phi_a), p(\phi_x)p(\phi_a)).$$

This means the joint distribution is being separated, i.e., $p(\phi_x, \phi_a) = p(\phi_x)p(\phi_a)$, but it does not mean the consistency with the RCTs $p^u(\phi_{x,a}) = p(\phi_x)p^u(\phi_a)$. To compensate $p(\phi_a)$, we weight the loss according to the estimated marginal probability of the actions $\beta = 1/\hat{p}(a)$.

The resulting objective is

$$\min_f \frac{1}{N} \sum_n L(f(x_n, a_n), y_n; g(x_n), \beta_n) + \alpha \cdot D_{\mathrm{bal}}\left(\{\phi(x_n, a_n)\}_n\right) + \mathfrak{R}(f), \quad (4.5)$$

where L is the empirical instance-wise version of (4.4), D_{bal} is the balancing regularizer (IPM or HSIC), and $\mathfrak{R}$ is a regularizer. The resulting learning flow is shown in Algorithm 4.

4.4 Relation Between Prediction Accuracy and Decision-making Performance

In this section, we analyze our decision-focused performance metric. This analysis demonstrates the difficulty of maximizing the decision performance only by minimizing the regression error when the action space is large. At the same time, however, it is shown that we can further minimize the upper-bound of the regret by minimizing a classification error, which justifies our proposed loss (4.4) in Section 4.3.1.

Here we define the decision performance of a model f as the simple average of the potential outcomes for the top-k predicted actions by f. We call that performance metric the mean cumulative gain (mCG), and also define its difference

Algorithm 4 Regret minimization network

Input: Observational data $D = \{(x_n, a_n, y_n)\}_n$, a hyperparameter α
Output: Trained network parameter W

Train g by an arbitrary supervised learning method with $D' = \{(x_n, y_n)\}_n$, e.g.:
$g = \arg\min_{g'} \sum (y_n - g'(x_n))^2$
if Method is RMNet-HSIC **then**
  Set weight $\beta_n = 1/\hat{p}(a_n)$ for each instance, where $\hat{p}(a_n)$ is the count $\hat{p}(a_n) \coloneqq \frac{|\{n \in D | a = a_n\}|}{|D|}$
else
  Set $\beta_n = 1$ for all n
end if
while Convergence is not met **do**
  Sample mini-batch $\{n_1, \ldots, n_b\} \subset \{1, \ldots, N\}$
  Calculate the gradient of the supervised loss L in (4.5): $g_1 = \nabla_W \frac{1}{b} \sum L(f(x_{n_i}, a_{n_i}; W), y_{n_i}; g(x_{n_i}), \beta_{n_i})$
  Calculate the gradient of the representation balancing regularizer: $g_2 = \nabla_W D_{\text{bal}}(\{\phi(x_{n_i}, a_{n_i}; W)\})$
  Obtain step size η with an optimizer (e.g., Adam [Kingma and Ba, 2015])
  $W \leftarrow [W - \eta(g_1 + \alpha g_2)]$
end while
return W

from the oracle's performance (regret):

$$\text{mCG}_k(f) \coloneqq \frac{1}{k} \mathop{\mathbb{E}}_x \left[\sum_{a:\text{rank}(f(x,a)) \leq k} y_a \right], \tag{4.6}$$

$$\text{Regret}_k(f) \coloneqq \frac{1}{k} \mathop{\mathbb{E}}_x \left[\sum_{a:\text{rank}(y_a) \leq k} y_a \right] - \text{mCG}_k(f), \tag{4.7}$$

where $\text{rank}(\cdot)$ is the rank among all the feasible actions, i.e., $\text{rank}(f(x,a)) = \text{rank}(f(x,a); \{f(x,a')\}_{a'}) \coloneqq |\{a' \mid f(x,a') \geq f(x,a), a' \in \mathcal{A}\}|$. Here, for binary outcome cases, $(1 - \text{mCG}_{k=1}(f))$ is known as the policy risk [Shalit et al., 2017]. Since the first term in (4.7) is constant with respect to f, the mCG and the regret are two sides of the same coin as the performance metrics of a model.

The relation between the regret and the regression and classification accuracies is the following (full proof and analysis on the tightness can be found in Appendix B.1.1).

Proposition 4.4.1. *The regret in* (4.7) *will be bounded with uniform MSE in* (4.1) *as*

$$\text{Regret}_k(f) \leq \frac{|\mathcal{A}|}{k} \sqrt{\text{ER}_k^u(f) \cdot \text{MSE}^u(f)}, \tag{4.8}$$

where $\mathrm{ER}_k^u(f)$ *is the top-k classification error rate, i.e.,*

$$\mathrm{ER}_k^u(f) \coloneqq \mathop{\mathbb{E}}_x \left[\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} I\left((\mathrm{rank}(y_a) \leq k) \oplus (\mathrm{rank}(f(x,a)) \leq k) \right) \right].$$

Proof Sketch. Let $s(x,a) \coloneqq I(\mathrm{rank}(y_a) \leq k) - I(\mathrm{rank}(f(x,a)) \leq k)$ denote the classification error. Then, we have

$$\begin{aligned}
k \cdot \mathrm{Regret}_k(f) &= |\mathcal{A}| \mathop{\mathbb{E}}_{x,a \sim p^u(x,a)} [s(x,a) y_a] \\
&\leq |\mathcal{A}| \mathop{\mathbb{E}}_{x,a \sim p^u(x,a)} [s(x,a)(y_a - f(x,a))] \qquad (4.9) \\
&\leq |\mathcal{A}| \sqrt{\mathop{\mathbb{E}}_{x,a \sim p^u(x,a)} [s(x,a)^2] \mathop{\mathbb{E}}_{x,a \sim p^u(x,a)} [(y_a - f(x,a))^2]} \qquad (4.10) \\
&= |\mathcal{A}| \sqrt{\mathrm{ER}_k^u(f) \cdot \mathrm{MSE}^u(f)}.
\end{aligned}$$

Equation (4.9) is from the definition of $s(x,a)$ and (4.10) is from the Cauchy-Schwarz inequality. By dividing both sides by k, we get the proposition. □

Since $\mathrm{ER}_k^u(f) \leq 1$ for any f, we see that only minimizing the uniform MSE as in existing causal inference methods leads to minimizing the regret. However, if $|\mathcal{A}|/k$ is large, the bound would be loose, and only unrealistically small MSE^u provides a meaningful guarantee for the regret.

At the same time, we see that the bound can be further improved by minimizing the uniform top-k classification error rate $\mathrm{ER}_k^u(f)$ simultaneously, which leads to our proposed method. Let k' be the past decision-makers' average performance, i.e., $y_{a^*_{k'+1}} \leq \mathbb{E}_{a \sim \mu(a|x)}[y_a|x] \leq y_{a^*_{k'}}$. Then, the proposed method can be interpreted as minimizing the upper-bound of $\mathrm{Regret}_{k'}$. While training a model for a particular k is an interesting direction, the proposed method is not so sensitive to the difference between the decision-making performance of the data k' and the actual k to be evaluated, as we will see in Section 4.5. Another interesting direction is optimizing k or the decision-making policy. The mCG_k can be interpreted as the expected performance (reward) of the following plug-in policy that takes an action uniformly at random from the predicted top-k actions:

$$\pi_k^f(a|x) := \begin{cases} \dfrac{1}{k} & \text{if } \mathrm{rank}(f(x,a); \{f(x,a')\}_{a'}) \leq k \\ 0 & \text{otherwise,} \end{cases}$$

Therefore, choosing k means choosing a policy. If we choose k greater than 1, the oracle's performance (the first term in (4.7)) would be smaller, but the upper bound of the regret (4.8) would be larger. Thus there may exist an optimal $k > 1$

that maximizes the overall performance of the decision-making.

4.5 Experiments

We investigated the effectiveness of our method through numerical experiments on synthetic and semi-synthetic datasets. We newly designed both datasets for the problem setting with a large action space.

4.5.1 Setup

Baseline methods

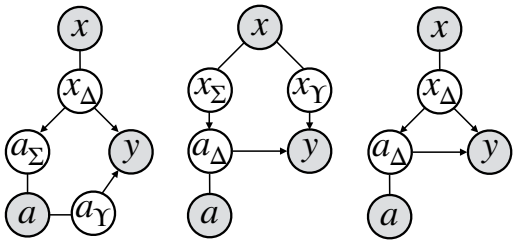
We compared our proposed method (RMNet) with ridge linear regression (OLS), random forests [Breiman, 2001] (RF), k-nearest neighbor (kNN), Bayesian additive regression trees (BART) [Hill, 2011], naive deep neural network (S-DNN), naive DNN with multi-head architecture for each action (M-DNN) (a.k.a. TARNET [Shalit et al., 2017]), RankNet [Burges et al., 2005], and a straightforward extension of the existing action-wise representation balancing method (counterfactual regression network (CFRNet)) [Shalit et al., 2017]. We also made an ablation study to clarify the contributions of each component. The strength of representation balancing regularizer α in CFRNet and the proposed method was selected from $[0.1, 0.3, 1.0, 3.0, 10.0]$. Other specifications of the DNN parameters can be found in Appendix B.2.1.

Evaluation

We used the normalized mean gain (NMG) as the main metric, defined as

$$\text{NMG} := \frac{\sum_x y_{\hat{a}^*}(x)}{\sum_x y_{a^*}(x)},$$

where $\hat{a}^*$ and a^* are the predicted and true best actions for each x, respectively. The NMG is proportional to the mean CG ($k = 1$) (4.6). We can see NMG ≤ 1. Since we have standardized the outcome, the chance rate is NMG $= 0$. In addition to NMG, we have also evaluated with respect to MSE^u and $\text{ER}^u_{k=1}$. The validation and the model selection were based on the NMG. For those cases where the complete validation dataset to compute NMG is not accessible, an alternative validation strategy needs to be considered, e.g., imputing missing values by 1-NN or BART (as in [Hassanpour and Greiner, 2019]) or constructing a special method (such as the counterfactual cross-validation in [Saito and Yasui, 2020]).



(a) Setup A (b) Setup B (c) Setup C

Figure 4.4: Data generation models for synthetic data experiment. Shaded variables denote the accessible variables in training. Non-shaded variables are latent one-dimensional representations of x and a.

Infrastructure

All the experiments were run on a machine with 28 CPUs (Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz), 250GB memory, and 8 GPUs.

4.5.2 Experiment on synthetic data

Dataset

We prepared four biased datasets with sampling bias in total to examine the robustness of the proposed and baseline methods. For a detailed description of the generation process, see Appendix B.2.1. The feature space and the action space are fixed to $\mathbb{R}^5$ and $\{0,1\}^5$, respectively. The true causal models are set as follows. Three settings (called Quadratic) have a relation $y_a(x) = a_\Upsilon^2 - 2x_\Upsilon + \varepsilon$, where $a_\Upsilon = w_a^\top a$ and $x_\Upsilon = w_x^\top x$ are the one-dimensional representations of a and x, respectively, and where $w_a, w_x \sim N(0, 1/5)^5$. The last setting (called Bilinear) has a bilinear relation $y = x^\top W a + \varepsilon$, where $W \sim N(0, 1/25)^{5\times 5}$. For training, only one action and the corresponding outcome for each x are sampled as $p(a|x) \propto \exp(10\,|x_\Sigma - a_\Sigma|)$, where x_Σ and a_Σ are additional representations of x and a. The three settings for the quadratic patterns correspond to the relation between $\cdot_\Sigma$ and $\cdot_\Upsilon$ as illustrated in Fig. 4.4(a)–(c), i.e., $x_\Sigma = x_\Upsilon$ (=: x_Δ) in Setups A and C, and $a_\Sigma = a_\Upsilon$ (=: a_Δ) in Setups B and C. These relations of variables were designed to reproduce spurious correlations, which mislead the decision-making as follows. In Setup A, a_Σ would have dependence on y through its dependence on x_Δ despite a_Σ itself having no causal relation to y. In the same manner, in Setup B, x_Σ would have a dependence on y through a_Δ, and the causal effect of a_Δ may appear discounted. Setup C has both effects. The sample sizes for x were 1,000 for training, 100 for validation, and 200 for testing.

Table 4.1: Synthetic results on NMG (larger is better and the maximum is one) and its standard error in ten data generations. Best and second-best methods are in bold.

Method	Quadratic-A	Quadratic-B	Quadratic-C	Bilinear
OLS	0.35 ± 0.13	0.74 ± 0.10	0.73 ± 0.12	0.02 ± 0.02
RF	0.71 ± 0.08	0.24 ± 0.02	0.91 ± 0.04	0.67 ± 0.03
kNN	0.58 ± 0.05	0.33 ± 0.04	0.53 ± 0.07	0.59 ± 0.03
BART	0.53 ± 0.12	**0.91 ± 0.05**	**0.99 ± 0.00**	0.14 ± 0.07
M-DNN	0.46 ± 0.09	0.42 ± 0.12	0.57 ± 0.12	−0.01 ± 0.04
S-DNN	0.63 ± 0.08	0.43 ± 0.07	0.60 ± 0.08	0.58 ± 0.09
CFRNet	0.46 ± 0.08	0.43 ± 0.12	0.63 ± 0.13	−0.01 ± 0.04
RankNet	0.62 ± 0.09	0.70 ± 0.05	0.68 ± 0.08	**0.74 ± 0.04**
RMNet-IPM	**0.86 ± 0.04**	0.84 ± 0.03	0.82 ± 0.05	**0.77 ± 0.04**
RMNet-HSIC	**0.90 ± 0.02**	**0.88 ± 0.05**	**0.86 ± 0.07**	0.14 ± 0.03

Table 4.2: Semi-synthetic results on NMG with the standard error in ten different samplings of the training data. The MSE^u and $\mathrm{ER}^u_{k=1}$ are also shown. Best and second-best methods are in bold.

	Normalized mean gain				MSE^u				$\mathrm{ER}^u_{k=1}$			
$\lvert\mathcal{A}\rvert$	8	16	32	64	8	16	32	64	8	16	32	64
Method												
OLS	−0.04 ± 0.15	−0.08 ± 0.20	−0.10 ± 0.13	−0.01 ± 0.10	1.12	1.89	1.70	5.86	0.221	0.116	0.061	0.031
RF	0.24 ± 0.08	0.33 ± 0.07	0.33 ± 0.05	0.38 ± 0.05	1.03	0.87	0.93	1.07	0.214	0.114	0.059	0.030
kNN	0.35 ± 0.04	0.39 ± 0.04	0.33 ± 0.04	0.39 ± 0.02	**0.59**	**0.64**	**0.64**	0.63	0.211	0.113	0.059	0.030
BART	−0.05 ± 0.13	0.13 ± 0.13	0.13 ± 0.10	0.04 ± 0.09	1.06	1.05	1.15	1.63	0.222	0.116	0.060	0.031
M-DNN	0.40 ± 0.05	0.48 ± 0.06	0.30 ± 0.07	0.37 ± 0.05	0.78	0.83	0.82	0.84	0.211	0.113	0.059	0.030
S-DNN	0.28 ± 0.09	0.25 ± 0.10	0.32 ± 0.07	0.45 ± 0.05	0.75	**0.64**	0.74	**0.74**	0.212	0.114	0.059	0.029
CFRNet	0.50 ± 0.06	0.39 ± 0.14	0.39 ± 0.10	0.35 ± 0.05	0.78	0.80	0.87	0.86	0.210	0.113	0.058	0.030
RankNet	0.35 ± 0.07	0.29 ± 0.09	0.38 ± 0.06	0.45 ± 0.05	6.08	10.13	8.47	2.42	0.210	0.113	0.058	0.029
RMNet-IPM	**0.68 ± 0.01**	**0.61 ± 0.05**	**0.61 ± 0.04**	**0.51 ± 0.06**	0.76	0.81	0.85	0.75	**0.204**	**0.109**	**0.055**	**0.029**
RMNet-HSIC	**0.59 ± 0.04**	**0.57 ± 0.06**	**0.55 ± 0.06**	**0.69 ± 0.06**	**0.48**	0.66	**0.61**	**0.39**	**0.207**	**0.109**	**0.056**	**0.028**

Results

The results listed in Table 4.1 show that our proposed method achieved the best or comparable performance under all settings, while the other methods varied in performance across settings. We analyze the reason for the poor performance of RMNet-HSIC in Bilinear in the ablation study in Section 4.5.4.

4.5.3 Experiment on semi-synthetic data

Dataset (GPU kernel performance)

We used the SGEMM GPU kernel performance dataset [Nugteren and Codreanu, 2015, Ballester-Ripoll et al., 2019], which has 14 feature attributes of GPU kernel parameters and four target attributes of elapsed times in milliseconds for four independent runs of each combination of parameters. We used the inverse of the mean elapsed times as the outcome, resulting in 241.6k instances in total. By treating some of the feature attributes as action dimensions, we obtained a *complete* dataset, which has all the entries (potential outcomes) in Fig. 4.1 observed. Then we composed our semi-synthetic dataset by biased subsampling of only one action a and the corresponding potential outcome y_a for each x. The details of this preprocess can be found in Appendix B.2.1.

The sampling policy in the training data was

$$p(a|x,y) \propto \exp(-10|y - [x^\top, a^\top]^\top w|),$$

where w is sampled from $\mathcal{N}(0,1)^{d+m}$. This policy reproduces a spurious correlation; that is, a random projection of the feature and the action $[x^\top, a^\top]^\top w$ is likely to have a little causal relation with y but does have a strong correlation due to the sampling policy. This policy also depends on y, which violates the unconfoundedness assumption. However, the dataset we used has a low noise level, i.e., $y \simeq g(x,a)$ for some function g, and thus the violation is limited, i.e., $p(a|x,y) \simeq p(a|x,g(x))$.

We split the feature set $\{x_n\}_n$ into 80% for training, 5% for validation, and 15% for testing. Then, for the training set, only one action a and the corresponding outcome y was taken for each x. The resulting training sample size for each setting of m is listed in Table B.2 in Appendix B.2.1. We repeated the training and evaluation process ten times for different splits and samplings of a.

Results

The results listed in Table 4.2 show that our proposed methods outperformed the others in NMG in all cases. The decision performance (NMG) was more consistent with ER than MSE, indicating that ER, as well as MSE, needs to be considered. The performance of multi-head DNNs (M-DNN and CFRNet) decreased in larger action spaces, while single-head DNNs (S-DNN and the proposed methods) maintained their performance. This demonstrates the importance of sample efficiency by extracting the representation of both the feature and the action.

4.5.4 Ablation study

We examined the effect of each component of the proposed method, i.e., the balancing regularizer (D_{bal}), each component of the risk (MSE and ER), and the representation extraction from the action (ϕ_a) and the reweighting with respect to the marginal distribution of the action (β) for RMNet-HSIC. Table 4.3 shows the results.

The effectiveness of D_{bal} was verified in the setting of $|\mathcal{A}| = 32$. Also, the effectiveness of ER was significant in the Bilinear setting. Extracting representation from the action (ϕ_a) was quite effective in Semi-synthetic settings. The reweighting (β) was also effective in the Semi-synthetic settings, while it decreased the performance in the Bilinear setting. A possible reason is the estimation variance induced by plugging the estimated marginal distribution of the action $\hat{p}(a)$ into

Table 4.3: Ablation study of the proposed methods (indicated by (†)) on semi-synthetic dataset. D_{bal} indicates the type of balancing regularizer. MSE and ER are the used loss. ϕ_a indicates whether or not the representation is also extracted from the action, i.e., if ϕ_a is not checked, identity function is used for ϕ_a (i.e., $\phi_a = a$). β indicates the reweighting with $1/\hat{p}(a)$, which is needed only in the HSIC-based methods (as explained in Section 4.3.2). Best and second-best methods are in bold.

					Normalized mean gain		
					Synthetic	Semi-synthetic	
D_{bal}	MSE	ER	ϕ_a	β	Bilinear	$\lvert\mathcal{A}\rvert = 32$	$\lvert\mathcal{A}\rvert = 64$
† IPM	✓	✓	✓	—	**0.77** ± **0.04**	**0.61** ± **0.04**	0.51 ± 0.06
IPM		✓	✓	—	**0.73** ± **0.03**	**0.61** ± **0.05**	0.58 ± 0.05
IPM	✓		✓	—	0.55 ± 0.10	0.55 ± 0.05	0.49 ± 0.05
None	✓	✓	✓		0.72 ± 0.03	0.39 ± 0.07	0.49 ± 0.06
†HSIC	✓	✓	✓	✓	0.14 ± 0.03	0.55 ± 0.06	**0.69** ± **0.06**
HSIC		✓	✓	✓	0.11 ± 0.02	0.56 ± 0.07	**0.72** ± **0.02**
HSIC	✓		✓	✓	0.16 ± 0.05	0.59 ± 0.05	0.68 ± 0.06
HSIC	✓	✓		✓	0.04 ± 0.03	0.31 ± 0.08	0.23 ± 0.09
HSIC	✓	✓	✓		0.51 ± 0.07	0.38 ± 0.07	0.49 ± 0.06
HSIC	✓	✓			0.63 ± 0.05	0.29 ± 0.07	0.22 ± 0.09

weights as its inverse, which is the same issue as the inverse propensity score weighting approach.

4.6 Summary

In this chapter, we have investigated causal inference on a large action space with a focus on decision-making performance. We analyzed the decision-making performance brought about by a model through a simple prediction-based decision-making policy. We showed that the bound with only the regression accuracy (MSE) gets looser as the action space gets large, which demonstrates the difficulty of utilizing causal inference in decision-making in a large action space. At the same time, however, our bound indicates that minimizing not only the regression loss but also the classification loss leads to better performance. From this viewpoint, our proposed methods minimize both the MSE and the classification loss of whether or not the outcome is better than the average performance of the past decision-makers. Specifically, we adopt the cross-entropy with a teacher label indicating whether an observed outcome is better than the estimated average decision performance of the past decision-makers under a given feature. For the sample efficiency in a large treatment space, we proposed extracting representations from both the feature and the action. To generalize in the distribution of RCTs, we proposed two balancing regularizers that encourage the representation distribution to be similar to that of RCTs as extensions of existing approaches. Experiments on synthetic and semi-synthetic datasets, which were designed to have misleading spurious correlations, demonstrated the superior performance of the proposed methods with respect to the decision performance.

Chapter 5

Causality-aware Modeling for Set-wise Recommendation

5.1 Introduction

Recently, the importance of selecting the combinations of items, i.e., the set-wise (exact-k, slate, combinatorial) modeling, has received attention in the recommendation context [Gong et al., 2019, Wang et al., 2019a, Ie et al., 2019, Jiang et al., 2019]. The set-wise modeling aims to overcome the limitation with the greedy top-k recommendations [Cremonesi et al., 2010], such as the lack of diversity in the recommended items [Ziegler et al., 2005, Wang et al., 2018]. For example, recommending multiple TVs at the same time is unlikely to result in the purchase conversion of both, while recommending a TV and a DVD player may increase the probability that both will be purchased. The former is called the substitute relation and the latter is called the complementary relation [Kök et al., 2008]. We consider learning the set-wise model that captures such relationships to evaluate the whole set of items to recommend.

Here we should note the difference between causation and association. Just because some items are often purchased at the same time does not necessarily mean that the probability of being purchased increases if they are recommended at the same time [Manchanda et al., 1999, Kök et al., 2008]. We want to recommend a set of items that results in a larger expected total outcome (e.g., purchase conversion) through recommending simultaneously. That is, we have to consider the prediction under interventions (a.k.a. actions or treatments) by a recommender system, which is known as the causal effect inference problem [Rubin, 2005]. As shown in Fig. 5.1, the conditional average treatment effect estimation (CATE) in causal inference as well as the recommendation problems can be viewed as learning from biasedly missing dataset (missing not completely at random) [Rubin, 2005, Little and Rubin, 2019].

This chapter is based on Akira Tanimoto, Tomoya Sakai, Takashi Takenouchi, and Hisashi Kashima, Causal combinatorial factorization machines for set-wise recommendation, in Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2021.

x	a	Y_a		y
		$a=0$	$a=1$	
x_1	0	1	–	1
x_2	1	–	3	3
x_3	1	–	2	2
x_4	0	2	–	2

(a) CATE estimation

	Item 1	Item 2	Item 3	Item 4
User 1	1			4
User 2			2	3
User 3	4	2		
User 4		5	2	

(b) Top-k recommendation

a	{Item 1, Item 2}		{Item 1, Item 3}		{Item 1, Item 4}		{Item 2, Item 3}		{Item 2, Item 4}		{Item 3, Item 4}	
i	Item 1	2	1	3	1	4	2	3	2	4	3	4
User 1					1	4						
User 2											2	3
User 3	4	2										
User 4							5	2				

(c) Set-wise recommendation

Figure 5.1: Example data tables for existing and our problem settings. Potential outcome prediction for CATE estimation and matrix completion for recommendation are both missing value completion under biased observations (missing not completely at random). The set-wise recommendation can model the dependence among items, e.g., a customer purchase only an item out of recommended items in the same category.

Recently, in the context of the top-k recommendation, several methods have been proposed that address the missing entries not completely at random using debiasing techniques in causal inference [Schnabel et al., 2016, Wang et al., 2019b]. As shown in Fig. 5.1(c), a dataset for the set-wise recommendation would be severely sparse, and thus it is quite important to take care not to overfit for biased training data. We therefore consider the problem of debiased inference for the set-wise recommendation.

Our approach is training a set-wise evaluation (rate/click prediction) model for the recommendation in a debiased manner. Considering that the final goal is to choose the action that is expected to maximize the outcome, a straightforward approach is learning a policy that outputs the recommended set (as in [Gong et al., 2019, Wang et al., 2019a, Ie et al., 2019, Jiang et al., 2019]) instead of making a prediction of outcomes. Even so, it is reasonable to make a prediction of the outcome when the predicted value itself is needed. In assortment optimization in retail stores, for example, store owners should also determine the ordering quantity on the basis of the demand forecast [Kök et al., 2008]. In such cases, the prediction of the outcome itself is essential for the decision-makers. We discuss how to optimize the set of items to recommend using a prediction model in Section 5.4.3.

5.2 Problem Setting

Our goal is to build an outcome (rate/click) prediction model under the feature x (typically the customer ID) and the action a (the set of recommended items) from biased data. Our training set is $S = \{(x_n, a_n, y_n)\}_{n=1}^N \sim p(y|a,x)\mu(a|x)p(x)$, where $x \in \mathcal{X}$ is the feature of a user (typically the one-hot encoding of the user ID), $a \in \mathcal{A} \subset \{0,1\}^{|\mathcal{I}|}$ is the action (the recommended set) in a combinatorial action space $\mathcal{A}$ with the candidate set of items $\mathcal{I}$, $\mu(a|x)$ is the propensity (the

policy of the past decision-makers or logging policy), and $y_n = (y_{n,t})_{t \in a_n} = (y_{n,t_1}, \ldots, y_{n,t_{|a_n|}}) \in \mathbb{R}^{|a|}$ is the outcome vector that consists of outcomes for each recommended items. Then, we train a model $f(x, a)$ to predict the outcomes $(y_t)_{t \in a}$.

The overall outcome of a set-wise recommendation a for a user can be evaluated by the summation of the rates of the recommended items, i.e., $y_a = \sum_{t \in a} y_{n,t}$. In that case, the difference from the simple outcome prediction on a combinatorial action space [Zou et al., 2020] is that we observe not only the overall outcome $y_a \in \mathbb{R}$ but all the rates for each recommended item $\{y_{s,t}\}_t$. The challenge in this chapter is how to obtain an outcome prediction model f sample-efficiently from observational data collected by a biased and possibly unknown policy (propensity) $\mu(a|x)$. Formally, we pursue the prediction accuracy on unbiased distribution:

$$L^u(f) := \mathop{\mathbb{E}}_{p(y_a|x)p^u(a)p(x)}[\ell(y_a, f(x, a))], \tag{5.1}$$

where y_a denotes the potential outcome (rate or click; each entry of Fig. 5.1(c)), $p^u(a) = \mathrm{Unif}(\mathcal{A})$ is the discrete uniform distribution on the action space $\mathcal{A}$, and ℓ is the instance-wise loss. To evaluate (5.1) unbiasedly, we use unbiased datasets for testing. In addition to the prediction accuracy, we also evaluate the value of recommendation, i.e., the estimated average clicks when we optimize the item set to recommend with the model. We will discuss this metric in Section 5.5.2.

5.3 Related Work

5.3.1 Treatment effect estimation

The goal of conditional average treatment effect (CATE) estimation is to estimate the causal effect τ for each individual specified by the feature x. CATE is defined as $\tau(x) = \mathbb{E}[y^{(1)} - y^{(0)}|x]$, where $y^{(1)}, y^{(0)} \in \mathcal{Y} \subset \mathbb{R}$ are the potential outcomes for each action, namely, if we take an action $a = 1$, then we observe $y^{(1)}$, and if $a = 0$, we observe $y^{(0)}$. The challenges here are the missing values and the selection bias, i.e., true τ is never observed but either $y^{(0)}$ or $y^{(1)}$ is observed, and the logging policy $\mu(a|x)$ is not constant (biased) in x. A typical approach is to train a potential outcome prediction model $f : \mathcal{X} \times \{0, 1\} \to \mathcal{Y}$ and estimate CATE by $\hat{\tau}(x) = f(x, a = 1) - f(x, a = 0)$. A typical performance measure is the expected precision in estimation of heterogeneous effect (PEHE) [Hill, 2011] $\epsilon_{\mathrm{PEHE}}(\tau) := \mathbb{E}_x[(\tau(x) - \hat{\tau}(x))^2]$, or the MSE on the joint distribution with the uniform policy.

A well-known workaround called the inverse probability of treatment weighting using the propensity score (IPW) [Austin, 2011] aims to debias by means of

instance weighting using the propensity $\mu(a|x)$ as

$$L_{\mathrm{IPW}}(f) := \mathop{\mathbb{E}}_{x,a}\left[\frac{1}{2\mu(a|x)}(y_a(x) - f(x,a))^2\right].$$

Since the expected IPW risk matches the expected risk on the randomized controlled trials (RCTs), a good performance can be expected asymptotically. When the true propensity is not recorded, however, we have to estimate the propensity score and plug-in with finite sample size, and then its performance might degrade [Kang et al., 2007]. A recent trend to improve non-asymptotic performance is using a representation balancing regularizer [Shalit et al., 2017, Johansson et al., 2016], which encourages matching the conditional probability of feature representations $p(\phi(x)|a)$ among actions. They utilize an adversarial domain adaptation technique, i.e., training a feature extractor ϕ to deceive a treatment discriminator $g(x)$ and construct hypotheses $\{h_a\}_a$ on the extracted feature representation $z = \phi(x)$ for each action. We utilize the causal inference approach for a large action space discussed in Chapter 4 since our action space is vast, as illustrated in Fig. 5.1(c). We extract representations from both the features and the actions as $z = \phi(x, a)$ and regularize the representation distribution to be similar to that extracted from randomized actions $z^u = \phi(x, a^u)$.

5.3.2 Modeling for recommendation

In real-world recommendation systems, the sampling distribution for items is not uniform because popular items tend to be frequently recommended, among other reasons. In order to reduce such sample selection bias, treatment effects have been actively considered recently. Schnabel et al. [2016] proposed a simple approach of utilizing propensity scores to weigh the error of the matrix factorization method. A similar but different approach in [Bonner and Vasile, 2018] aims at debiasing by means of multi-task learning of a large (biased) observational dataset and a small randomized dataset. At the same time, the importance of selecting the combinations of items, i.e., set-wise modeling, has also received attention recently in the recommendation context [Gong et al., 2019, Wang et al., 2019a, Ie et al., 2019, Jiang et al., 2019]. Gong et al. [2019] considered the exact-k recommendation problem, in which the task is to select k items to show to users in a limited area of a screen. In [Gong et al., 2019], the item interaction is expressed as a graph, and then a neural network with an attention mechanism learns a policy for selecting items one by one. While these methods [Gong et al., 2019, Ie et al., 2019, Jiang et al., 2019] focus on generating the recommended set of items in a computationally efficient manner, the selection bias is not considered, and there is a risk of performance decay under strong selection biases in real-world problems. Therefore, we investigate the debiased modeling of the evaluator for

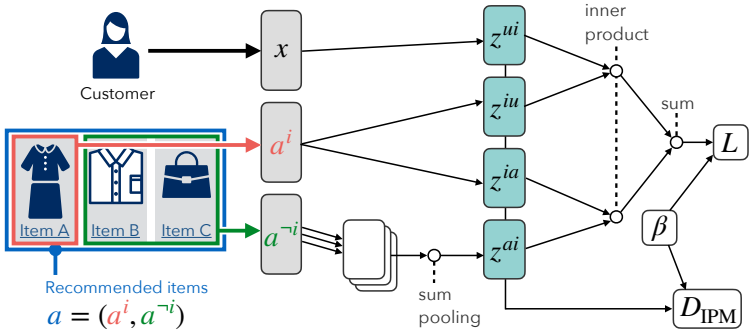


Figure 5.2: Combinatorial FM structure

set-wise modeling utilizing recent techniques in causal inference.

5.4 Causal Combinatorial Factorization Machines for Set-wise Recommendation

5.4.1 Model: combinatorial factorization machines

In recommendation tasks, the outcome, which we aim to maximize, would typically be the sum of the rates of recommended items to users. However, we can observe not only the sum but also the rates for each item. Therefore, we use the rates for each item as the supervision, with consideration of the other items recommended to (or rated by) each customer user. That is, our data consists of

$$D = \{y_n, x_n, a_n^i, a_n^{\neg i}\}_{n=1}^N,$$

where n is the sample index, x_n and a_n^i are the one-hot encoded user ID and the target item ID associated with the rate y_n, respectively, and $a_n^{\neg i} = (0, 1, 1, 0, \ldots)$ is the other items recommended to the user at the same time where 1s correspond to the IDs of other recommended items. The final set-wise outcome for a user identified by x' is the summation of the outcomes of recommended items $\sum_{n:x_n=x'} y_n$ and the corresponding action is denoted as $a = a_n^i + a_n^{\neg i}$.

Factorization machines [Rendle, 2010] enable us to learn the matrix factorization model by means of SGD with one-hot encoding of the user IDs and the item IDs. We extend the factorization machines to take the second-order interactions between the recommended items into account for the set-wise modeling. Specifically, we include the second-order interaction term of the target item and the other recommended items (or other items rated by the same user), as

$$f(x, a) := w_0 + \sum_j w_j^u x_j + \sum_j w_j^a a_j^i + \sum_{j,j'} \langle z_j^{ui}, z_{j'}^{iu} \rangle x_j a_{j'}^i + \sum_{j,j'} \langle z_j^{ia}, z_{j'}^{ai} \rangle a_j^i a_{j'}^{\neg i}, \tag{5.2}$$

where $w_0, w^u, w^a, z^{ui}, z^{iu}, z^{ia}$, and z^{ai} are the model parameters. The resulting network structure is shown in Fig. 5.2. Let 1_j be the one-hot encoding of an integer j. When one recommends the t-th item to the s-th user, and at the same time the other item set recommended is $\mathcal{T}'$, the prediction (5.2) is written as

$$f\left(x = 1_s, a = \left(1_t, \sum_{t' \in \mathcal{T}'} 1_{t'}\right)\right) = w_0 + w^u_s + w^a_t + \langle z^{ui}_s, z^{iu}_t \rangle + \left\langle z^{ia}_t, \sum_{t' \in \mathcal{T}'} z^{ai}_{t'} \right\rangle.$$

The final term handles the interaction between the target item t and other recommended items $\mathcal{T}'$, which represents the substitution or complementary relation between recommended items. A positive inner product value $\langle z^{ia}_j, z^{ai}_{j'} \rangle > 0$ represents that the j-th target item is a complementary relation with respect to the j'-th item, and the rate would be higher when recommended with the j'-th item. Since the interaction is considered to be invariant to the permutation of other recommended items, we utilize the sum-pooling as proposed in deep sets for permutation-invariant functions [Zaheer et al., 2017].

5.4.2 Debiased loss with causal inference techniques

To train our model (5.2) in a debiased manner from the biased observational data, we introduce two debiasing techniques, namely, the weighting technique proposed in the top-k recommendation and the representation balancing technique proposed in causal inference for large treatment spaces.

Although the representation balancing approach in Chapter 4 is scalable to a vast set-wise action space in both statistically and computationally, a limitation is that the balanced representation of inputs cannot capture the difference in the output distributions (i.e., when $p(y) \neq p^u(y)$) as shown in [Johansson et al., 2019, Zhao et al., 2019]. Especially in recommendation datasets with explicit feedbacks, the rate prior shift is often observed because the users are likely to rate their favorite items among others. This difference is exactly what previous IPW-based methods for recommendation address (called naive-Bayes IPW) [Schnabel et al., 2016, Wang et al., 2019b]. Therefore, we combine this weighting with the representation balancing approach.

Let us define the integral probability metric (the representation balancing regularizer) as

$$D_{\mathrm{IPM}}(p_1, p_2) := \sup_{g \in G} \left| \int_{\mathcal{Z}} g(z)(p_1(z) - p_2(z)) \mathrm{d}z \right| \tag{5.3}$$

with a function class G. We utilize the 1-Lipschitz function class for G as in [Shalit et al., 2017, Tanimoto et al., 2021b], after which D_{IPM} would be the Wasserstein distance D_{wass}. With any weighting function $\beta(z)$, assuming that

the representation extractor $z = \phi(x, a)$ is invertible and $\frac{1}{B}\ell(z)$ is in the function class G for some $B > 0$ with respect to z, our target loss on the randomized distribution (5.1) can be bounded as

$$L^u(f) \leq L(f; \beta) + B \cdot D_{\mathrm{IPM}}\left(p(z)\beta(z), p^u(z)\right),$$

where $L(f; \beta)$ is the weighted loss on the observational data. This bound justifies minimizing the empirical estimate of r.h.s. as a proxy of unobservable unbiased loss $L^u(f)$. The proof is given by just replacing the source distribution $p(z)$ in the non-weighted version of the bound in Chapter 4 with the weighted distribution $p(z)\beta(z)$. Note that, the weighted distribution must satisfy $\int p(z)\beta(z)\mathrm{d}z = 1$, otherwise a constant critic $g(z) = c$ for $c > 0$ gives a non-zero IPM value and the supremum in (5.3) does not exist when G is the 1-Lipschitz function class.

For the weights β, we can utilize the information obtained in each problem setting. Assuming that there exists the true rating function $y = h^*(z)$, the naive-Bayes weighting can be reproduced as

$$\beta(z) := \frac{\mathbb{E}_{p^u(z)}[h^*(z)]}{\mathbb{E}_{p(z)}[h^*(z)]} = \frac{p^u(y)}{p(y)} =: \beta(y). \tag{5.4}$$

Thus, when we have no access to the true propensity μ but have access to the rate prior shift $p^u(y)/p(y)$ as assumed in [Schnabel et al., 2016, Wang et al., 2019b], we can utilize this weighting. If we have access to the true propensity, we can utilize it as $\beta(z = \phi(x, a)) = p^u(a)/\mu(a|x)$, after which D_{IPM} would be zero, which recovers the full IPW method.

Our resulting objective function is

$$\min_f \frac{1}{N}\sum_{n=1}^{N} \beta_n \ell(f(x_n, a_n), y_n) + \mathfrak{R}(f) + \alpha \cdot \hat{D}_{\mathrm{wass}}\left(\{z_n, z_n^u, \beta_n\}_{n=1}^N\right), \tag{5.5}$$

where $\beta_n = \beta(y_n) = p^u(y_n)/p(y_n)$ if available, $\ell(y', y)$ is the instance-wise loss, namely the weighted MSE or cross-entropy for rate and click prediction, respectively, $\mathfrak{R}$ is a regularizer, $z_n = \phi(x_n, a_n)$, $z_n^u = \phi(x_n, a_n^u)$, a_n^u is random actions sampled from $\mathrm{Unif}(\mathcal{A})$, $\alpha \geq 0$ is the regularization strength, $\hat{D}_{\mathrm{wass}}$ is the balancing regularizer with weights, as

$$\hat{D}_{\mathrm{wass}}(\{z_n, z_n^u, \beta_n\}_{n=1}^N) := \sup_{g \in G} \left| \frac{1}{\sum_{n=1}^N \beta_n} \sum_{n=1}^{N} \beta_n g(z_n) - \frac{1}{N}\sum_{n=1}^{N} g(z_n^u) \right|, \tag{5.6}$$

where G is the 1-Lipschitz function class.

Algorithm 5 Combinatorial Factorization Machines

Input: $D = \{x_n, a_n, y_n\}_{n=1}^N$, where $a_n = (a_n^i, a_n^{\neg i})$
Output: trained network parameter W

Calculate instance weights $\{\beta_n\}_n$ by, e.g., (5.4), with the training and the validation sets
while Convergence is not met **do**
  Sample mini-batch $\{n_1, \ldots, n_b\} \subset \{1, \ldots, N\}$
  Calculate the gradient of the supervised loss: $g_1 = \nabla_W \frac{1}{b} \sum_i^b L(\phi(x_{n_i}, a_{n_i}; W), y_{n_i})$
  Sample uniformly random action $\{a_1^u, \ldots, a_b^u\} \sim \mathcal{A}^b$.
  Calculate the gradient of the critic with β in (5.6): $g_2 = \nabla_W \hat{D}_{\text{wass}}(\{\phi(x_{n_i}, a_{n_i}; W)\}, \{\phi(x_{n_i}, a_i^u; W)\}; \{\beta_{n_i}\})$
  Obtain step size η with an optimizer, e.g., Adam [Kingma and Ba, 2015]
  $W \leftarrow [W - \eta(g_1 + \alpha g_2)]$
end while
return W

5.4.3 Optimizing the item set to recommend using a model

We here explain how to obtain a set-wise recommendation from our prediction model. Recall that k is the number of items that we present to customers from $|\mathcal{I}|$ candidate items. One approach to finding a promising set-wise recommendation is that we first prepare $|\mathcal{A}| = {}_{|\mathcal{I}|}\mathrm{C}_k$ candidates of set-wise recommendation and then choose the one that achieves the highest estimated outcome. Specifically, for a customer whose feature vector is x, we prepare a set of item-set vectors $\{a_j\}_{j=1}^{{}_{|\mathcal{I}|}\mathrm{C}_k}$ with $|a| = k$ and then choose the combination by $\mathrm{argmax}_j \widehat{f}(x, a_j)$, where $\widehat{f}$ is the learned predictor. This approach is accurate, but it is intractable when ${}_{|\mathcal{I}|}\mathrm{C}_k$ is large. When the number of item-sets is large, one can adopt a greedy approach to a set-wise recommendation. That is, we iteratively select one item to construct a set-wise recommendation. Initialize the selected item-set vector a' with zero vector. For a customer x, we select an item by $j' = \mathrm{argmax}_j \widehat{f}\left(x, (a^i = 1_j, a^{\neg i} = a')\right)$. Let a'' be the vector of current selected item-set as $a'' = a' + 1_{j'}$. We again select an item by $j'' = \mathrm{argmax}_j \widehat{f}\left(x, (a^i = 1_j, a^{\neg i} = a'')\right)$. We repeat the above procedure until the number of selected items becomes k. Since ${}_{|\mathcal{I}|}\mathrm{C}_k$ increase quickly even for a small k, this greedy approach is effective in terms of computation.

5.5 Experiments

We investigated the effectiveness of our method through experiments using three real-world datasets. Two of them were originally for the top-k recommendation. The other one was recorded in a way that follows the set-wise recommendation, where multiple items were shown simultaneously to a user.

5.5.1 Sequential display setting

Datasets

As in [Schnabel et al., 2016], we first evaluated on two real-world datasets with explicit feedbacks, Yahoo! R3 [Marlin et al., 2007] and Coat [Schnabel et al., 2016], to compare with existing causal-aware top-k recommendation methods [Schnabel et al., 2016, Wang et al., 2019b]. Yahoo! R3 had 15,300 user IDs and 1,000 song IDs, and Coat had 290 user IDs and 300 item IDs, both of which contain missing not at random (MNAR) data for training and missing completely at random (MCAR) data for testing. For the combi-FM-based methods, we used the set of rated items for each user as a in each of the training and the testing datasets without overlap, i.e., a in the test data did not contain the rated items in the training. We cannot completely reproduce the situation in which a user examines each item and rates it sequentially due to the lack of the order of items that the user rated, though the set of rated items contains the set of previously exposed items to the user, which can be captured by our set-wise modeling.

Baseline methods

We compared our proposed method with several existing methods and straightforward combinations of our model and existing training methods, namely, factorization machines (FM) [Rendle, 2010], FM with IPW with weights estimated by naive Bayes (FM-IPWnb), combinatorial FM (5.2) without IPM regularization nor IPW (Combi-FM), Combi-FM with naive-Bayes IPW (Combi-FM-IPWnb), Combi-FM with the Wasserstein with weights by naive-Bayes IPW (Combi-FM-Wass) and existing matrix factorization (MF) and its causal-aware extensions based on naive-Bayes IPW reported in [Wang et al., 2019b]. For the FM-based methods, we used the width of 10 for the representation ϕ. For the combi-CFR (proposed), α in (5.5) was fixed to 0.5.

Results

Table 5.1 lists the overall results. The proposed method outperformed all other methods with respect to MSE in the Yahoo! R3 dataset. In contrast, an existing method (MF-DR-JL) performed best in the Coat dataset. The Coat dataset is relatively small, which might be why the SGD-based methods (FM-based and Combi-FM-based) did not achieve a good performance. In Yahoo! R3 dataset, in contrast, the Combi-FM model worked well, which implies that the set of items rated by the user affected the user's rating to the target item and our model effectively extracted that information. Combi-FM-based methods suffered from their model complexity and might overfit the biased observational training

Table 5.1: Test MAE and MSE on Yahoo and Coat datasets. (*) reported in [Wang et al., 2019b]. The top methods for each metric are in bold and the second places are italicized and underlined.

	YAHOO		COAT	
Method	MAE	MSE	MAE	MSE
MF (*)	1.154	1.891	0.920	1.257
MF-IPS (*)	0.810	0.989	*0.860*	*1.093*
MF-DR-JL (*)	*0.747*	*0.966*	**0.778**	**0.990**
FM	0.803	1.170	1.187	2.534
FM-IPWnb	**0.736**	1.031	1.148	2.398
Combi-FM	0.959	1.259	0.930	1.290
Combi-FM-IPWnb	0.821	1.050	0.945	1.281
Combi-FM-Wass (proposed)	0.781	**0.958**	0.966	1.287

data (e.g., vanilla Combi-FM performed worse than FM); however, with a proper regularization, as proposed (IPW and the Wasserstein-based), the generalization improved. This indicates that the combination of set-wise modeling and debiased training is important.

5.5.2 Simultaneous display setting

We investigated a CTR prediction for situations in which more realistic set-wise recommendations are made. A customer sees three items simultaneously in an impression, and clicks for each item are recorded. Here, we evaluated not only the accuracy of the predictions but also the value of the recommendations made.

Dataset

We used Open Bandit Dataset (OBD) [Saito et al., 2020] taken from a fashion e-commerce platform, ZOZOTOWN. This dataset was constructed for evaluating bandit algorithms offline and for evaluating offline policy evaluation methods. OBD contains two datasets taken with two (recorded) logging policies μ, namely, a random policy and a biased policy (Bernoulli Thompson sampling, BTS). We used the dataset with BTS for training and validation, and used the dataset with the random policy for testing. Half of the BTS dataset was used for validation. OBD contains three "campaigns", namely, "men's", "women's", and "all". We used only the dataset of the "all" campaign. The size of the candidate set of items was $|\mathcal{I}| = 80$, and the size of the action space would be $|\mathcal{A}| = {}_{80}C_3 = 82,160$.

We preprocessed datasets as follows. OBD is anonymized, i.e., the customer ID is deleted; thus, we constructed pseudo-user ID (PUID) from four hashed customer features. Only records tied to PUIDs that appeared in both training and test datasets were used, after which we had 397 unique PUIDs in total. The original data was not intended for the set-wise recommendation, and the

three items displayed at the same impression were divided into three (mostly consecutive) records; thus, we processed to combine them. Consecutive records with the same pseudo-user ID and with different display positions were treated as an impression. After these processes, we had $2,549,288$ combined records in the BTS training/validation set and had $293,871$ records in the random test set.

Baseline methods

We compared FM and our Combi-FM models with three losses, namely, the naive loss, weighted loss with the true propensity score, and the Wasserstein regularized loss without weights ($\beta_n = 1$). The regularization strength of the balancing regularizer was chosen from $\{0.1, 0.3, 1., 3.\}$ by validation.

Evaluation

We evaluated our method and baselines with two metrics. The first one is a conventional ranking-based metric for imbalanced classification, average precision (AP), which is the area under the precision-recall curve. While AP (or AUC, discounted cumulative gain, etc.) is a popular metric in recommendation and information retrieval, these global ranking-based metrics do not fit well with the recommendation problem on e-commerce platforms. A platform needs to choose an recommendation action for a customer rather than choosing a customer to recommend, and therefore it is preferable to use metrics based on local ranking of candidate actions for each customer. For this reason, the other metric we evaluated was the value of the policy with predictions $V(\pi_f)$, i.e., the expected clicks when determining the action using the model:

$$V(\pi_f) = \mathbb{E}_{\pi_f, p(x)}\left[y_a\right] = \mathbb{E}_{\mu(a|x), p(x)}\left[\frac{\pi_f(a|x)}{\mu(a|x)} y_a\right], \tag{5.7}$$

where π_f is a plug-in policy distribution with a model f, which is defined below, μ is the propensity (logging policy) of the dataset, and y_a is the summation of clicks for each item shown at the same time. we use a policy of randomly performing an action from among the top k%-predicted actions for evaluation:

$$\pi_f^k(a|x) = \begin{cases} \frac{k}{100} & \left(\mathrm{rank}(\tilde{f}(x,a); \{\tilde{f}(x,a')\}_{a' \in |\mathcal{A}|}) \leq \frac{|\mathcal{A}|k}{100}\right) \\ 0 & (\text{otherwise}), \end{cases} \tag{5.8}$$

where $\mathrm{rank}(v; V)$ denotes the ranking of a value v among a set of values V, $\tilde{f}(x,a) = \sum_{t \in \mathcal{T}_a} f(x, (a^i = 1_t, a^{\neg i} = \sum_{t' \in \mathcal{T}_a \setminus t} 1_{t'}))$ is the total predicted clicks, and where $\mathcal{T}_a$ is the set of recommended items. Since the expectation in (5.7) is taken over the same distribution with the dataset, we can empirically estimate (5.7) with the test set, which is known as the inverse propensity score estimate

Table 5.2: Test policy value $V\left(\pi_f^{k=1\%}\right)$ and average precision (AP) on the ZOZO dataset. The top methods for each metric are in bold. Mean and standard deviation under three runs with different training/validation splits are reported.

	ZOZOTOWN	
Method	Policy value (k=1%) ($\times 10^{-2}$)	AP ($\times 10^{-3}$)
FM	1.18 ± 0.05	4.05 ± 0.06
FM-IPW	0.93 ± 0.02	**4.64 ± 0.09**
FM-Wass	1.23 ± 0.07	3.81 ± 0.16
Combi-FM	1.18 ± 0.05	4.05 ± 0.06
Combi-FM-IPW (proposed)	**1.47 ± 0.02**	4.51 ± 0.08
Combi-FM-Wass (proposed)	*1.43 ± 0.09*	*4.58 ± 0.42*

[Bottou et al., 2013]. This metric (with the plug-in policy (5.8)) is similar to the cumulative gain, where the outcomes of the top-k best-predicted items are counted, but the difference is that the ranking takes place for all candidate actions for each customer. To avoid heavy computation of $\{f(x, a')\}_{a' \in |\mathcal{A}|}$ for each customer, we subsampled $\mathcal{A}' \subset A$ of cardinality $1,000$ to evaluate $\pi_f^{k=1\%}$.

Here, it is difficult in terms of off-line evaluation to evaluate only with respect to the best-predicted action as described in Section 5.4.3. This is because it is very rare that a single action that is chosen among the ${}_{|\mathcal{I}|}\mathrm{C}_k$ candidates matches exactly the recorded action, and the estimation variance of the metric would be too large. Therefore we adopted a stochastic policy rather than the deterministic policy of performing the best-predicted action.

Results

As shown in Table. 5.2, our proposed Combinatorial FM model with debiasing techniques achieved the best performances in policy value and comparable performances in AP. Notably, Combi-FM-Wass (without weights β) performed almost the best in both scores despite not using propensity score information. The chance rate that calculated from the click rate was $V(\mathrm{Unif}(\mathcal{A})) = 1.05 \times 10^{-2}$. Thus the proposed method achieved approximately 1.4 times as many clicks as random, even though the action was not optimized but randomly chosen from the top 1% predicted actions.

5.6 Summary

In this chapter, we have proposed an extended FM model and debiased learning method for the set-wise recommendation. Our model is based on the factorization machines and extended to take into account the second-order interactions between recommended items. We utilize weighting and the representation balancing regularizer to alleviate the bias in observations and to achieve better performance.

Experiments on real-world recommendation datasets demonstrated the superior performance of the proposed methods, especially for large-scale datasets.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

In this thesis, we aimed at more automated modeling for data-driven decision making, started with the machine learning formulation problem, and explored the idea of utility-level modeling, its challenges, and approaches. That is, we have to consider the utility of the decision based on the trained model to fully automate the modeling procedure, including the choice of performance metric and loss function. In Chapter 1, we discussed three challenges when transforming predictive modeling into direct utility modeling:

1-1 Computationally expensive optimization during training with respect to the next action in sequential decision-making problems

1-2 Sample selection bias due to the past decision-makers' policy (propensity)

2 How to incorporate with intermediate results (states) for sample-efficiency

In the following chapters, we discussed approaches for solving each of these challenges.

In Chapter 2, we discussed the first challenge. When the decision-making is sequential, the outcome can only be evaluated in the long run, and thus the immediate utility of action in a time step would be a prediction of the outcome under the conditions of subsequent actions optimized. Even when the state transition phenomenon is independent, the optimization objective is often dependent among components of the whole target. An example we take was the maintenance optimization of infrastructure. Although the deterioration processes of each patch of infrastructure are independent of each other, when it comes to the maintenance action, simultaneous maintenance is economical, which leads to combinatorial optimization of maintenance action. For this problem, we exploited a locality in dependency and designed a decomposed Q-function for efficient optimization by dynamic programming.

In Chapter 3, we discussed (2) the statistical challenge of the scarce outcome, i.e., how to incorporate with intermediate states. This information is not associated with inputs nor output of the utility model to train; thus, how to incorporate this information is not straightforward. Therefore, we discussed the framework of learning using privileged information (LUPI), enabling us to utilize the intermediate information as a part of supervision (privileged information; PI). We examined this idea under a typical situation, i.e., imbalanced classification with numerical labels, and revealed the benefit of utilizing PI theoretically and experimentally. Our theoretical analysis revealed that our proposed method reduces the estimation variance by identifying "near-miss" instances that alleviate the class imbalance. Experiments have shown the versatile performance of our method for various datasets and metrics compared to other approaches, i.e., direct classification without PI, indirect regression, and rank-based modeling of intermediate results. This versatility implies the possibility of integrating these approaches, i.e., direct modeling and predictive modeling of each phenomenon, by modeling directly but utilizing intermediate results as PI. On the other hand, utilizing a predictive model for intermediate state by sampling counterfactual intermediate results is a more popular approach in reinforcement learning [Moerland et al., 2020]. How this differs from the LUPI framework and which approach is appropriate in which cases are matters for future research.

In Chapter 4, we discussed (1-2) the statistical challenge of vast action spaces, i.e., the sampling bias due to the past decision-makers' policy. We have shown that extending the causal inference approach to large action spaces is effective for this problem, i.e., we suppose the potential outcomes of all possible actions, including counterfactual ones, and train a model aiming accuracy over all the potential outcomes. We revealed that the performance of decision-making (regret) is (not directly evaluated but) bounded using two types of accuracies, namely, the regression accuracy (MSE) and a kind of classification accuracy of whether the action is relatively good or not. This is the first work linking the accuracy in causal inference to the decision-making performance to the best of our knowledge. To generalize the large action space, we proposed a single-head architecture and debiased training with representation balancing both feature and action.

In Chapter 5, we extended our causal inference method to combinatorial (set-wise) recommendation problem. To take into account the dependency of recommended items such as substitutional or complementary relationships, we extended the factorization machines to input the simultaneously recommended items other than the prediction target. We integrated the representation balancing method introduced in Chapter 4 with the instance weighting approach, which overcomes the weakness of representation balancing for the rate (outcome) prior shift. While the set-wise model tends to overfit the biased training data due to its complexity,

our debiased objective alleviates the overfit and achieves favorable performances.

In summary, in order to eliminate the manual formalization in predictive analytics, we extended the utility-level modeling to a wide range of previously challenging areas to handle with reinforcement learning. They are (1) the domains where the action space is vast and thus computationally or statistically challenging, and (2) ones where the outcome is scarce and thus the intermediate information is essential. Since many decision-making problems can be expressed as optimization of the utility function, our utility modeling is a promising direction towards automating data-driven decision-making.

6.2 Future Directions

Finally, we discuss possible future directions towards automated data-driven decision-making. First, while we focused on the LUPI framework for the sample-efficiency, a model-agnostic and thus versatile approach, incorporating model knowledge is another possible approach, such as the Markov property utilized in model-based reinforcement learning. Investigating which of these approaches is promising in what situations or integrating them is an exciting direction.

Second, we have focused on the utility modeling part and assumed a greedy plug-in policy for the decision-making. However, the policy also has room for improvement, or even is essential to consider, in some situations. When the action optimization is computationally hard with no desirable properties such as locality to exploit, a possible remedy is modeling the policy and sample from it instead of optimization, which is known as the actor-critic algorithm [Konda and Tsitsiklis, 2000]. Also, the greedy policy might be vulnerable under the existence of rare state-action pairs (i.e., the propensity is nearly zero $\mu(a|x) \simeq 0$, which is usually excluded by the strong ignorability assumption in causal inference). For such situations, inducing the policy to avoid such rare actions by incorporating imitation learning [Hussein et al., 2017] or estimating utility conservatively out-of-distribution [Kumar et al., 2020] are possible approaches.

Combining online settings is the final direction. We have focused on the offline setting in this thesis since the experiment in real situations is often expensive or even infeasible due to political issues. However, offline modeling requires strong assumptions such as ignorability and might converge slowly compared to online methods. For example, when some information utilized by past decision-makers is not recorded (e.g., patient's complexion), then the conditional independency $a \not\perp (y_a)_a \mid x)$ and thus the strong ignorability is no longer satisfied, and the training can be misled. Also, offline evaluation of the trained model for model selection or deployment decision has the same issue. Although there are a number of causal inference methods developed for this situation, all of them have other

assumptions to the best of our knowledge. To make matters worse, it is not possible to determine from data alone whether these conditions are met or not. Therefore, an analyst with a good understanding of these conditions should be involved in offline-only settings.

We hope that further investigation will be made in these directions and that modeling technology will automate decision-making everywhere in society.

Appendix A

Appendix to Chapter 3

A.1 Proofs

A.1.1 Proof of Theorem 3.4.1

First, we prepare a lemma to upper bound using the Lipschitz constant of the instance-wise loss function. In the contraction lemma of the Rademacher complexity [Shalev-Shwartz and Ben-David, 2014], the Lipschitz constant with respect to the scoring function value is constant for all instances. However, in the case of cost-sensitive loss, the Lipschitz constant is large (C_+) only for a small number of instances (positive), and it is small (C_-) for most of the instances (negative). Therefore, to get a tighter upper bound, it is preferable to evaluate the Lipschitz constant, instance by instance.

Lemma A.1.1 (Element-wise contraction)**.** *For each $n \in [N]$, let $\ell_n : \mathbb{R} \to \mathbb{R}$ be a ρ_n-Lipschitz function; namely, for all $\alpha, \beta \in \mathbb{R}$ we have $|\ell_n(\alpha) - \ell_n(\beta)| \leq \rho_n |\alpha - \beta|$. Then, the Rademacher complexity R of the losses is bounded as*

$$R(\{\ell_n(a_n)\}) \leq R(\{\rho_n a_n\}).$$

Proof. First, we set an upper bound for an instance, $n = 1$. Let $p(\epsilon_n = 1) = 1/2$ and $p(\epsilon_n = -1) = 1/2$ for all $n \in [N]$. Then, the Rademacher complexity is

bounded as

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\epsilon}\left[\sup_{\{a_n\}}\left\{\sum \epsilon_n \ell_n(a_n)\right\}\right] \\
=&\frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\}}\left\{\rho_1\ell(a_1)+\sum_{n=2}^{N}\epsilon_n\ell_n(a_n)\right\}+\sup_{\{a_n\}}\left\{-\rho_1\ell(a_1)+\sum_{n=2}^{N}\epsilon_n\ell_n(a_n)\right\}\right] \\
=&\frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\},\{a'_n\}}\left\{\rho_1(\ell(a_1)-\ell(a'_1))+\sum_{n=2}^{N}\epsilon_n\ell_n(a_n)+\sum_{n=2}^{N}\epsilon_n\ell_n(a'_n)\right\}\right] \\
\leq&\frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\},\{a'_n\}}\left\{\rho_1|a_1-a'_1|+\sum_{n=2}^{N}\epsilon_n\ell_n(a_n)+\sum_{n=2}^{N}\epsilon_n\ell_n(a'_n)\right\}\right] \\
=&\frac{1}{2}\mathop{\mathbb{E}}_{\epsilon_2,\ldots,\epsilon_N}\left[\sup_{\{a_n\},\{a'_n\}}\left\{\rho_1(a_1-a'_1)+\sum_{n=2}^{N}\epsilon_n\ell_n(a_n)+\sum_{n=2}^{N}\epsilon_n\ell_n(a'_n)\right\}\right] \\
=&\mathop{\mathbb{E}}_{\epsilon_1,\ldots,\epsilon_N}\left[\sup_{\{a_n\}}\left\{\epsilon_1\rho_1 a_1+\sum_{n=2}^{N}\epsilon_n\ell_n(a_n)\right\}\right].
\end{aligned}
$$

The inequality comes from the definition of the Lipschitz function. By applying this repeatedly for all instances, we get the lemma. □

Next, we provide the proof of the theorem. Let $g = w^\top x$ be the decision function value. Then $\ell(y, g(x))$ is m_n-Lipschitz w.r.t. g, where $m_n := \max\left\{C_+\frac{p_+}{p_{T,+}}\sigma(z/T), C_-\frac{p_-}{p_{T,-}}\sigma(-z/T)\right\}$. Then, we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{S}\left[L_T(\hat{w})\right] - \inf_{w:\|w\|_2\leq B} L_T(w) \\
&\leq 2\mathop{\mathbb{E}}_{S}\mathop{\mathbb{E}}_{\epsilon}\left[\sup_{w:\|w\|_2\leq B}\left\{\frac{1}{N}\sum \epsilon_n \ell(w, x_n, y_n)\right\}\right] \\
&\leq 2\mathop{\mathbb{E}}_{S,\epsilon}\left[\frac{1}{N}\sup_{\|w\|_2\leq 1}\sum_n \epsilon_n m_n w^\top x\right] && \text{(Lemma A.1.1)} \\
&= 2\mathop{\mathbb{E}}_{S,\epsilon}\left[\frac{B}{N}\left\|\sum_n \epsilon_n m_n x_n\right\|_2\right] \\
&\leq 2\mathop{\mathbb{E}}_{S}\left[\frac{B}{N}\sqrt{\mathop{\mathbb{E}}_{\epsilon}\left[\left\|\sum_n^N \epsilon_n m_n x_n\right\|_2^2\right]}\right] && \text{(Jensen's ineq.)} \\
&\leq \frac{2BX}{N}\sqrt{\mathop{\mathbb{E}}_{S}\left[\sum_n^N m_n^2\right]}. && \text{(Jensen's ineq.)}
\end{aligned}
$$

The first inequality comes form Theorem 26.3 in Shalev-Shwartz and Ben-David [2014].

Since $s_n^2 \leq s_n$, $\max(a, b) \leq a + b$ for $a, b > 0$, and $\mathbb{E}_S[\sum_n^N s_n] = Np_{T,+}$, the

r.h.s. is bounded as follows:

$$\text{r.h.s.} \leq \frac{2BX}{\sqrt{N}} \sqrt{C_+^2 \frac{p_+^2}{p_{T,+}} + C_-^2 \frac{p_-^2}{p_{T,-}}},$$

which concludes the proof.

A.1.2 Proof of Proposition 3.4.2

The additional bias can be rewritten as

$$\begin{aligned}(\text{bias1} + \text{bias2}) = \mathop{\mathbb{E}}_{x} \Big[\Delta_+ \mathop{\mathbb{E}}_{S} \left[\ell(g^*(x)) - \ell(\hat{g}(x))\right] \\ + \Delta_- \mathop{\mathbb{E}}_{S} \left[\ell(-g^*(x)) - \ell(-\hat{g}(x))\right] \Big],\end{aligned} \tag{A.1}$$

where Δ_+ and Δ_- are the differences in weighted labels defined as $\Delta_+ := \mathbb{E}_{z|x}\left[C_{T,+}\sigma(z/T) - C_+ I(z \geq 0)\right]$ and $\Delta_- := \mathbb{E}_{z|x}\left[C_{T,-}\sigma(-z/T) - C_- I(z < 0)\right]$.

From Hölder's inequality, we have

$$\begin{aligned}\text{r.h.s. of (A.1)} \leq & \mathop{\mathbb{E}}_{x}\left[|\Delta_+|\right] \max_{x:p(x)>0} \left| \mathop{\mathbb{E}}_{S}\left[\ell(g^*(x)) - \ell(\hat{g}(x))\right] \right| \\ & + \mathop{\mathbb{E}}_{x}\left[|\Delta_-|\right] \max_{x:p(x)>0} \left| \mathop{\mathbb{E}}_{S}\left[\ell(-g^*(x)) - \ell(-\hat{g}(x))\right] \right| \\ \leq & c\left(\mathop{\mathbb{E}}_{x}\left[|\Delta_+|\right] + \mathop{\mathbb{E}}_{x}\left[|\Delta_-|\right] \right).\end{aligned} \tag{A.2}$$

On the other hand, from the definition of η, we have

$$\begin{aligned}p(\eta = 1) &= \mathop{\mathbb{E}}_{z}\left[\sigma(z/T)\right] = p_{T,+}, \\ p(\eta = 1|x) &= \mathop{\mathbb{E}}_{z|x}\left[\sigma(z/T)\right],\end{aligned}$$

and thus

$$\frac{1}{2p_{T,+}} \mathop{\mathbb{E}}_{z|x}\left[\sigma(z/T)\right] p(x) = \frac{1}{2}\frac{p(\eta = 1|x)p(x)}{p(\eta = 1)} = \frac{1}{2}p(x|\eta = 1).$$

Samely, we have

$$\frac{1}{2p_+} \mathop{\mathbb{E}}_{z|x}\left[I(z \geq 0)\right] p(x) = \frac{1}{2}p(x|y = 1).$$

Therefore, in the BER minimization setting, i.e., $C_+ = 1/2p_+$ and $C_{T,+} =$

$1/2p_{T,+}$,

$$\begin{aligned}\mathop{\mathbb{E}}_{x}\left[|\Delta_+|\right] =& \int \left| \mathop{\mathbb{E}}_{z|x}\left[\frac{1}{2p_{T,+}}\sigma(z/T) - \frac{1}{2p_+}I(z \geq 0)\right]\right| p(x)\mathrm{d}x \\ =& \frac{1}{2}\int \left|p(x|\eta=1) - p(x|y=1)\right| \mathrm{d}x \\ =& \mathrm{TV}\left(p(x|\eta=1), p(x|y=1)\right).\end{aligned}$$

Samely, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{x}\left[|\Delta_-|\right] =& \int \left| \mathop{\mathbb{E}}_{z|x}\left[\frac{1}{2p_{T,-}}\sigma(-z/T) - \frac{1}{2p_-}I(z \leq 0)\right]\right| p(x)\mathrm{d}x \\ =& \frac{1}{2}\int \left|p(x|\eta=0) - p(x|y=0)\right| \mathrm{d}x \\ =& \mathrm{TV}\left(p(x|\eta=0), p(x|y=0)\right).\end{aligned}$$

By substituting $\mathbb{E}_x\left[|\Delta_+|\right]$ and $\mathbb{E}_x\left[|\Delta_-|\right]$ in (A.2), we get the proposition.

A.2 Experimental Details

A.2.1 Computing infrastructure

All the experiments were run on a machine with eight CPUs (Intel Xeon E7-8850 2.0GHz, ten cores) and 1.0TB RAM.

A.2.2 Data preprocesses

We here describe the preprocesses for real datasets. First, we describe the common preprocesses for all datasets and then describe preprocesses for each dataset.

Common preprocesses

We applied the following preprocesses for all the datasets.

- The standardization, i.e., scaling and shifting so as to $\mathbb{E}[x] = 0$ and $\mathrm{Var}[x] = 1$ for each feature, was applied.
- The binary expansion was applied to categorical features, i.e., a categorical feature that has k categories are expanded into $k-1$ binary features. The first category in the alphabetical order was not expanded.
- For datasets that has multiple files (wine quality and student datasets) are concatenated, and a categorical feature that represents the source files was added.

- Instances that have missing features were deleted.

Toy

The toy data shown in Fig. 3.2(a), which we used also in the experiments, was generated as follows. The coefficients w and the features x are drawn from 100-dimensional standard normal distribution, and then, positivity z is drawn as

$$z \sim \mathcal{N}(5 \exp(w^\top x/15), 2).$$

Air quality

For the target attribute (CO(GT)), the value -200 means missing and thus removed. Categorical features named Date and Time was removed. In addition, a feature named NMHC(GT) was removed since there exist many missing entries.

Year prediction MSD

We sampled 10k instances at random.

A.2.3 Baseline methods and hyperparameter ranges

The methods compared include conventional classification methods, regression-based methods, and a rank-based method, as listed below. We also describe here the hyperparameter ranges considered.

Hyperparameter ranges

The considered hyperparameter configurations are the following:

- The regularization strength was ranged from 10^{-2} to 10^2.
- T of our proposed method ranged from 10^{-3} to 10^2.
- γ of an RBF kernel $\exp(\gamma \|x - x'\|^2)$ ranged from 10^{-2} to 10^2.
- For the GP, the hyperparameter optimizer was restarted five times.
- For the SMOTE-based methods, the number of neighboring points used to synthesize over-sampled points was optimized from $[3, 5, 8]$.
- For the RUSBoost, the number of estimators was optimized from $[20, 30, 50]$, random state ranged 0–2.

Models with our proposed method

As our method is model-agnostic, we performed experiments on different types of base classification learners. We adopted three models: logistic regression (LR) with L1 and L2 regularizers each, and a support vector machine (SVM) with an RBF kernel. The methods in this setting were as follows:

- proposed (base learner: LR (regularization: L1))
- proposed (base learner: LR (regularization: L2))
- proposed (base learner: SVM (kernel: radial basis function))

Conventional classification methods

These models use only the binary label $y \in \{0, 1\}$, not the numerical mediator $z \in \mathbb{R}$. We adopted the same models as those for the proposed method, namely LR with L1 and L2 regularizers each and SVM for the cost-sensitive classification and SMOTE. In a manner similar to that with the proposed method, the sample weights were rebalanced in the cost-sensitive classification. In other words, we learned from the data set $D = \{d_1, d_2, \ldots, d_N\}$, where $d_n = (\mathbf{x}_n, y_n)$ consists of a feature vector and a class label. This setting was normal classification. The models compared in this setting were as follows:

- LR (regularization: L1)
- LR (regularization: L2)
- SVM (kernel: radial basis function)

Also, we compared the RUSBoost, which utilize the boosting method as the base learner.

Regression-based methods

These methods learn and predict $z \in \mathbb{R}$ and then apply the threshold to the prediction. We adopted Lasso regression, Ridge regression, and a Gaussian process with an RBF kernel. In other words, we learned from the data set $D = \{d_1, d_2, \ldots, d_N\}$, where $d_n = (\mathbf{x}_n, z_n)$ consists of a feature vector and a target variable. This setting was normal regression. The models compared in this setting were as follows:

- Lasso regression
- Ridge regression
- Gaussian process (kernel: radial basis function)

Rank-based method

This model is based on a pair-wise ranking method in which the rank information is extracted from z. It learned a ranking function $r(\cdot)$. In $\{(\mathbf{x}_i, \mathbf{x}_j) \mid z_i > z_j\}$, the model was optimized to satisfy the pair-wise rank constraints: $r(\mathbf{x}_i) > r(\mathbf{x}_j)$ or $r(\mathbf{x}_i) - r(\mathbf{x}_j) = 0$, that is $\mathbf{w}^\top(\mathbf{x}_i - \mathbf{x}_j) = 0$. In general, the linear SVM with slack variables is commonly used for the pair-wise ranking method because of its computational-efficiency. The model employed in this setting was as follows:

- Rank-SVM (kernel: linear)

Appendix B

Appendix to Chapter 4

B.1 Proofs and Additional Analyses

B.1.1 Proof of Proposition 4.4.1

Proposition B.1.1. *The expected regret will be bounded with uniform MSE in (4.1) as*

$$\mathrm{Regret}_k(f) \leq \frac{|\mathcal{A}|}{k}\sqrt{\mathrm{ER}_k^u(f)\cdot \mathrm{MSE}^u(f)},$$

where $\mathrm{ER}_k^u(f)$ *is the top-k classification error rate, i.e.,*

$$\mathrm{ER}_k^u(f) \coloneqq \mathop{\mathbb{E}}_x\left[\frac{1}{|\mathcal{A}|}\sum_{a\in\mathcal{A}} I\left((\mathrm{rank}(y_a)\leq k)\oplus(\mathrm{rank}(f(x,a))\leq k)\right)\right],$$

where $\oplus$ *denotes the logical XOR.*

Proof. We denote the true and the predicted i-th best action by a_i^* and $\hat{a}_i^*$, respectively; i.e., $\mathrm{rank}(y_{a_i^*}) = \mathrm{rank}(f(x,\hat{a}_i^*)) = i$. For all $k \in [|\mathcal{A}|]$, the target-wise regret can be bounded as follows:

$$\begin{aligned}
k\cdot\mathrm{Regret}_k(x) &\coloneqq \sum_{i\leq k}\left(y_{a_i^*} - y_{\hat{a}_i^*}\right)\\
&\leq \sum_{i\leq k}\left(y_{a_i^*} - y_{\hat{a}_i^*}\right) + \sum_{i\leq k}\left(f_{\hat{a}_i^*} - f_{a_i^*}\right) \qquad \text{(B.1)}\\
&= \sum_{i\leq k}\left\{\left(y_{a_i^*} - f_{a_i^*}\right) - \left(y_{\hat{a}_i^*} - f_{\hat{a}_i^*}\right)\right\}\\
&= \sum_a \big\{(I(\mathrm{rank}(y_a)\leq k) - I(\mathrm{rank}(f_a)\leq k))(y_a - f_a)\big\},
\end{aligned}$$

where $f_a = f(x,a)$. Inequality (B.1) is from the definition of $\hat{a}_i^*$; i.e., $\sum_{i\leq k} f_{\hat{a}_i^*}$ is the summation of the top-k f_as out of $\{f_a\}_{a\in\mathcal{A}}$, which must be larger than or equal to the summation of k f_as that are not necessarily top-k, $\sum_{i\leq k} f_{a_i^*}$. Let us

define a classification error s and the regression error e as

$$s(x,a) \coloneqq I(\mathrm{rank}(y_a) \le k) - I(\mathrm{rank}(f_a) \le k),$$
$$e(x,a) \coloneqq y_a - f_a.$$

The r.h.s. is written as

$$\text{r.h.s.} = \sum_a s(x,a)e(x,a).$$

By taking the expectation with respect to x, we have

$$\begin{aligned}
k \cdot \mathrm{Regret}_k(f) &= \mathop{\mathbb{E}}_x \left[\sum_a s(x,a)e(x,a) \right] \\
&= |\mathcal{A}| \mathop{\mathbb{E}}_{(x,a)\sim p^u(x,a)} [s(x,a)e(x,a)] \\
&\le |\mathcal{A}| \sqrt{\mathop{\mathbb{E}}_{(x,a)\sim p^u(x,a)} [|s(x,a)|^2] \cdot \mathop{\mathbb{E}}_{(x,a)\sim p^u(x,a)} [|e(x,a)|^2]} \qquad \text{(B.2)} \\
&= \left\{ \mathop{\mathbb{E}}_x \left[\sum_a I(\mathrm{rank}(y_a) \le k \oplus \mathrm{rank}(f(x,a)) \le k) \right] \right. \qquad \text{(B.3)} \\
&\qquad \left. \cdot \mathop{\mathbb{E}}_x \left[\sum_a (y_a - f(x,a))^2 \right] \right\}^{1/2} \\
&= |\mathcal{A}| \sqrt{\mathrm{ER}_k^u(f) \cdot \mathrm{MSE}_k^u(f)},
\end{aligned}$$

where the inequality (B.2) comes from the Cauchy – Schwarz inequality and the equality (B.3) comes from the definitions of s and e. By dividing both sides by k, we get the proposition. □

Note that our bound cannot be improved without additional assumptions on the true and assumed model classes of the causal mechanism $f(x,a)$ (and thus the true potential outcomes and its predictions). For any $|\mathcal{A}|$, $k \le |\mathcal{A}|/2$, ER_k^u, MSE^u, and $\epsilon > 0$, there exist a joint distribution of potential outcomes and x, and a model f that have the gap (the ratio) between both sides of the proposition is $(1+\epsilon)$.

Let us define a prototype of a potential outcome vector $\boldsymbol{y}^\kappa$ as

$$\begin{aligned}
\boldsymbol{y}^\kappa \coloneqq &(y_1, \ldots, y_{|\mathcal{A}|}) \\
= &(\underbrace{1, \ldots, 1}_{\kappa}, \underbrace{-1, \ldots, -1}_{\kappa}, 0, \ldots, 0),
\end{aligned}$$

that is, the first κ dimensions are 1, the following κ dimensions are -1, and the rest are 0. When the true outcome is $\boldsymbol{y} = t\boldsymbol{y}^\kappa$ for $t > 0$ and $\kappa \le k$, and when the prediction of the model is bad (misleading) as $\hat{y} = -\epsilon\boldsymbol{y}$, the components of the

r.h.s. would be

$$\mathrm{MSE}^u = 2\kappa t^2(1+\epsilon)^2/|\mathcal{A}|,$$
$$\mathrm{ER}_k^u = 2\kappa/|\mathcal{A}|,$$

and thus the r.h.s. would be

$$\text{r.h.s.} = 2t\kappa(1+\epsilon)/k,$$

while the l.h.s. would be

$$\mathrm{Regret}_k = 2t\kappa/k.$$

The gap (the ratio) between them is $(1+\epsilon)$ for any ϵ. Since we have two free parameters κ and t, any MSE^u and ER_k^u can be (almost) achieved. At this point, the constraint $\kappa \in \mathbb{N}$ also causes a constraint on ER, but it can be removed ($\kappa \coloneqq |\mathcal{A}|\mathrm{ER}_k^u/2$ for any ER_k^u) as follows. We consider a domain partition $\mathcal{X}_1 \in \mathcal{X}$ and the potential outcomes as

$$p(\boldsymbol{y} = t\boldsymbol{y}^{\kappa_1}|x) = 1 \quad (x \in \mathcal{X}_1 \subset \mathcal{X}),$$
$$p(\boldsymbol{y} = t\boldsymbol{y}^{\kappa_2}|x) = 1 \quad (x \in \mathcal{X}\backslash\mathcal{X}_1),$$

where $\kappa_1 \coloneqq \lfloor |\mathcal{A}|\mathrm{ER}_k^u/2 \rfloor$, $\kappa_2 \coloneqq \lceil |\mathcal{A}|\mathrm{ER}_k^u/2 \rceil$, and the partition $\mathcal{X}_1$ can be determined to satisfy $\mathbb{E}_x\left[\mathrm{ER}_k^u(f,x)\right] = 2\kappa/|\mathcal{A}|$. Thus, for any $|\mathcal{A}|$, k, MSE^u, and ER_k^u, the bound cannot be improved without any assumption.

Our bound means that, when $\kappa = k \ll |\mathcal{A}|$ holds, despite this prediction $\hat{\boldsymbol{y}}$ being quite "accurate" in terms of MSE^u, the decision is constantly misleading regardless of $|\mathcal{A}|/k$ (and thus MSE^u). It could be improved when the spaces of $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are limited and well-specified, but such specification of the model class is another big issue in real-world applications. We therefore conclude that minimizing only the regression accuracy MSE^u is insufficient in terms of decision performance when the treatment space is large, and minimizing the classification accuracy ER_k^u is also important.

B.1.2 Error analysis for representation balancing regularization

By performing the representation balancing regularization, our method enjoys better generalization through minimizing the upper bound of the error on the test distribution (under the uniform random policy). We briefly show how minimizing the combination of empirical loss on training and the regularization of distribution (4.5) results in minimizing the test error. First, we define the point-wise loss function under a hypothesis h and an invertible extractor $\phi(\cdot,\cdot)$, which defines

the representation $\phi = \phi(x, a)$ with its inverse $(x, a) = \psi(\phi)$, as

$$\ell_h^{x,a}(\phi) \coloneqq \int_{\mathcal{Y}} L(Y_a, h(\phi)) p(Y_a|x) \mathrm{d}Y_a.$$

Then, the expected losses for the training (source) and the test distribution (target) are

$$\epsilon^s(h) \coloneqq \int_{\mathcal{X},\Phi} \sum_{a\in\mathcal{A}} \ell_h^{x,a}(\phi) p(\phi|x,a) p(x,a) \mathrm{d}\phi \mathrm{d}x,$$
$$\epsilon^t(h) \coloneqq \int_{\mathcal{X},\Phi} \sum_{a\in\mathcal{A}} \ell_h^{x,a}(\phi) p(\phi|x,a) p^u(a|x) p(x) \mathrm{d}\phi \mathrm{d}x,$$

where $p(\phi|x, a) = \delta(\phi - \phi(x, a))$. We assume there exists $B > 0$ such that $\frac{1}{B}\ell_h^{x,a}(\phi) \in G$ for the given function space G. Then the integral probability metric IPM_G is defined for $\phi \in \Phi = \{\phi(x, a)|p(x, a) > 0\}$ as

$$\mathrm{IPM}_G(p_1, p_2) \coloneqq \sup_{g\in G} \left| \int_\Phi g(\phi)(p_1(\phi) - p_2(\phi)) \mathrm{d}\phi \right|.$$

The difference between the expected losses under training and test distributions are then bounded as

$$\begin{aligned}
\epsilon^t(h) - \epsilon^s(h) &= \int_\Phi \ell_h^{\psi(\phi)}(\phi) \left(p^u(\phi) - p(\phi)\right) \mathrm{d}\phi \\
&= B \int_\Phi \frac{1}{B} \ell_h^{\psi(\phi)}(\phi) \left(p^u(\phi) - p(\phi)\right) \mathrm{d}\phi \\
&\leq B \sup_{g\in G} \left| \int_\Phi g(\phi) \left(p^u(\phi) - p(\phi)\right) \mathrm{d}\phi \right| \\
&= B \cdot \mathrm{IPM}_G \left(p(\phi), p^u(\phi)\right).
\end{aligned}$$

Although B is unknown, the hyperparameter tuning of the regularization strength α in (4.5) can achieve the tuning of B.

B.1.3 Minimizing IPM while preserving the causal relation

We show that minimizing the discrepancy between $p(\phi_{x,a})$ and $p^u(\phi_{x,a})$ and preserving the causal relationships are not necessarily in conflict with each other.

Let us consider an example of $p(x, a)$ shown in Table B.1 and a representation $\phi(x, a) = x + a$. Then, for any $\epsilon \in (-1/9, 1/9)$, the representation distribution is calculated as, e.g., $p(\phi_{x,a} = 1) = p(x = 1, a = 0) + p(x = 0, a = 1) = 2/9$. In the

same manner, we have

$$
\begin{aligned}
&p(\phi_{x,a}=0) = p(\phi_{x,a}=4) = 1/9,\\
&p(\phi_{x,a}=1) = p(\phi_{x,a}=3) = 2/9,\\
&p(\phi_{x,a}=2) = 1/3.
\end{aligned}
\tag{B.4}
$$

Also, the uniform target distribution is calculated as $p^u(x,a) \coloneqq p(x)p^u(a) = 1/9$ for all $x, a \in \{0,1,2\}$, and the representation distribution is the same as (B.4), meaning $\mathrm{IPM}(p(\phi_{x,a}), p^u(\phi_{x,a})) = 0$. If the true causal relation can be written via $\phi_{x,a}$ as $y = h(\phi(x,a))$ for some h, then this representation extractor can achieve $\mathrm{IPM} = 0$ while preserving the causal relation, and thus it can still achieve $L^u = 0$.

On the other hand, the action-wise representation extraction approach (e.g., CFRNet [Shalit et al., 2017] in Fig. 4.3(a)) cannot achieve both the extraction of fully balanced representation $\sum D_{\mathrm{IPM}} = 0$ and the preservation of the relation in this case with $\epsilon \neq 0$. Only constant representation $\phi(x) = c$ for all $x \in \{0,1,2\}$ can achieve $\sum_{a,a' \in \mathcal{A}} D_{\mathrm{IPM}}(p(\phi_x|a), p(\phi_x|a')) = 0$, and then the true relation $y = h(x+a)$ is not expressible.

When the action-wise representation achieves $\mathrm{IPM} = 0$, our representation $\phi(x,a)$ can also achieve $\mathrm{IPM} = 0$ under an assumption that the marginal action distribution is uniform, i.e., $p(a) = p^u(a)$. By defining the representation of both the feature and action as a concatenation $\phi(x,a) = (\phi_x, a)$, we have

$$
\begin{aligned}
p\left(\phi_{x,a}\right) &= p\left(\phi_x, a\right)\\
&= p\left(\phi_x \mid a\right) p(a)\\
&= p\left(\phi_x\right) p(a)\\
&= p\left(\phi_x\right) p^u(a)\\
&= p^u\left(\phi_x, a\right)
\end{aligned}
$$

under $p(a) = p^u(a)$.

These facts demonstrate that our proposed regularizer encourages a weaker (but sufficient) condition than the action-wise representation balancing approach.

Note that there is a potential issue with the use of a representation balancing regularizer with such a non-invertible representation as $\phi(x,a) = x + a$. That is, an unobservable error term would be induced in the upper-bound [Johansson et al., 2019, Zhao et al., 2019]. Thus, a minimization of only the observable error terms ($L + D_{\mathrm{bal}}$) may not lead to a minimization of the target error in such cases. Some countermeasures have been proposed to address this issue, such as adding a reconstruction loss of inputs to guarantee invertibility of the representation [Zhang et al., 2020], but in some cases, $D_{\mathrm{bal}} = 0$ is achieved only by using a

Table B.1: Example observational distribution $p(x, a)$ and its marginal distributions $p(x)$ and $p(a)$.

$\mu(a\|x)p(x)$	$a=0$	$a=1$	$a=2$	$p(x)$
$x=0$	$1/9$	$1/9+\epsilon$	$1/9-\epsilon$	$1/3$
$x=1$	$1/9-\epsilon$	$1/9$	$1/9+\epsilon$	$1/3$
$x=2$	$1/9+\epsilon$	$1/9-\epsilon$	$1/9$	$1/3$
$p(a)$	$1/3$	$1/3$	$1/3$	

non-invertible representation, as in the example in Table B.1. Therefore, this point remains an area for improvement in future work.

B.1.4 Connection between ER_k^u in Proposition 4.4.1 and ER_μ^u in (4.2)

We explain how we obtain ER^u in Eq. (4.2) from ER_k^u in Proposition 4.4.1. Recall ER_k^u in Proposition 4.4.1:

$$\mathrm{ER}_k^u(f) \coloneqq \mathop{\mathbb{E}}_x \left[\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} I\left((\mathrm{rank}(y_a) \leq k) \oplus (\mathrm{rank}(f(x,a)) \leq k) \right) \right].$$

We can show the following:

$$\begin{aligned}
&I\left((\mathrm{rank}(y_a) \leq k) \oplus (\mathrm{rank}(f(x,a)) \leq k) \right) \\
&= I\left((y_{a_k^*} \leq y_a) \oplus (f(x, \hat{a}_k^*) \leq f(x,a)) \right) \\
&= I\left((y_{a_k^*} \leq y_a) \oplus (y_{a_k^*} \leq f(x,a) - f(x, \hat{a}_k^*) + y_{a_k^*}) \right) \\
&= I\left((y_{a_k^*} \leq y_a) \oplus (y_{a_k^*} \leq f'(x,a)) \right),
\end{aligned}$$

where $f'(x,a) \coloneqq f(x,a) - f(x, \hat{a}_k^*) + y_{a_k^*}$. We then have

$$\mathrm{ER}_k^u(f) = \mathop{\mathbb{E}}_x \left[\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} I((y_a \geq y_{a_k^*}) \oplus (f'(x,a) \geq y_{a_k^*})) \right].$$

Here, the rank with f' ($\mathrm{rank}(f'(x,a))$) is the same as that of f, but the difference is that f' satisfies the condition $f'(x, \hat{a}_k^*) = y_{a_k^*}$. That is, the k-th largest value among $\{f'(x,a)\}_a$ equals to $y_{a_k^*}$. Although, since $y_{a_k^*}$ is unobservable, we relaxed the optimization of f' in the function space that satisfies the condition into the optimization in the general function space. In addition, we used the average performance of the past decision-makers (μ) with respect to the target (x) $\bar{y}$ instead of unobservable $y_{a_k^*}$, as

$$\mathrm{ER}_\mu^u(f) = \mathop{\mathbb{E}}_x \left[\frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} I(y_a \geq \bar{y} \oplus f(x,a) \geq \bar{y}) \right].$$

In the end, for such k' that satisfies $y_{a_{k'+1}^*} \leq \bar{y} \leq y_{a_{k'}^*}$ and for such f' that

satisfies $f'(x, \hat{a}^*_{k'}) = y_{a^*_{k'}}$, we have $\mathrm{ER}^u_\mu(f') = \mathrm{ER}^u_{k=k'}(f')$.

B.2 Experimental Details and Additional Results

B.2.1 Detailed experimental settings

Synthetic data generation process

Our synthetic datasets are built as follows.

1 Sample $x \sim \mathcal{N}(0,1)^d$, where $d = 5$.

2 Sample $a \in \{0,1\}^m$, where $m = 5$, from $p(a|x) \propto \exp(10\,|x_\Sigma - a_\Sigma|)$, where x_Σ and a_Σ are the following.

- 2-1 In settings other than Setup B, $x_\Sigma = x_\Delta = w_x^\top x$, where the projection is sampled as $w_x \sim \mathcal{N}(0, 1/d)^d$.
- 2-2 In Setup B, $x_\Sigma = x_1$, i.e., only the first dimension in x is used to bias a.
- 2-3 $a_\Sigma = w_a^\top a$, where $w_a \sim \mathcal{N}(0, 1/m)^m$.

3 Calculate the expected outcome $y_a = f(x, a)$, where we examine two types of functions f, namely, Quadratic and Bilinear. In the Quadratic setting, $f(x, a) = a_\Upsilon^2 - 2x_\Upsilon$, where x_Υ and a_Υ are one-dimensional representations of x and a, respectively.

- 3-1 In Setup B, $x_\Upsilon = w_{x,2:d}^\top x_{2:d}$, where $x_{2:d}$ denotes all dimensions other than the first one (x_Σ).
- 3-2 In settings other than Setup B, $x_\Upsilon = x_\Sigma (=: x_\Delta)$.
- 3-3 In Setup A, $a_\Upsilon = w_a'^\top a$, where $w'_a \sim \mathcal{N}(0, 1/m)^m$.
- 3-4 In settings other than Setup A, $a_\Upsilon = a_\Sigma (=: a_\Delta)$.
- 3-5 In Bilinear setting, $f(x, a) = x^\top W a$, where $W \sim \mathcal{N}(0, 1/(dm))^{(d,m)}$.

4 Sample the observed outcome $y \sim \mathcal{N}(y_a, 0.1)$.

Details of semi-synthetic data

We transformed the target attributes of elapsed times into the average speed as the outcome, i.e., $y = \frac{4}{\sum z_i}$, where $\{z_i\}_{1:4}$ are the original elapsed times. Then we standardized y and the feature attributes. Each feature attribute can take binary values or up to four different powers of two values. Out of 1,327k total parameter combinations, only 241.6k feasible combinations are recorded. We split these original feature dimensions into a and x as follows. The dimensions of the action space m ranged from three to six, and the 8th, 11th, 12th, 13th, 14th, and 3rd

Table B.2: Training sample size for each setting.

m	$\lvert\mathcal{A}\rvert$	N_{tr}
3	8	24,160
4	16	12,080
5	32	6,040
6	64	3,591

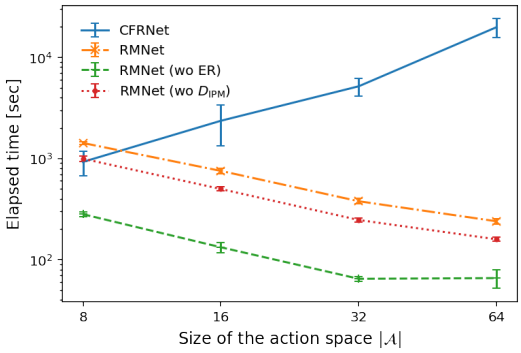


Figure B.1: Elapsed time for training. Error bars indicate standard deviation.

dimensions are regarded as a from the head in order (e.g., for $m = 3$, the 8th, 11th, and 12th dimensions in the original feature attributes are regarded as a). This split was for maximizing the overlap of $\mathcal{A}(x)$ among $\mathcal{X}$.

Other DNN parameters

The detailed parameters we used for DNN-based methods (S-DNN, M-DNN, CFRNet, and the proposed methods RMNet-IPM and RMNet-HSIC) were as follows. The backbone DNN structure had four layers for representation extraction and three layers for the hypothesis with the width of 64 for the middle layers and the width of 10 for the representation $\phi_{x,a}$. In RMNet-HSIC, the representation $\phi_{x,a} = (\phi_x, \phi_a)$ was composed of representations of feature (ϕ_x) and action (ϕ_a), each of which had a width of 5. The batch size was 64 except for CFRNet, where it was 512 due to the need to approximate the distributions for each action. The strength of the L2 regularizer was 10^{-4}. We used Adam [Kingma and Ba, 2015] as the optimizer with the learning rate of 10^{-4}.

B.2.2 Additional experimental results

Elapsed times compared to CFR

Figure B.1 shows the comparison in training time between the proposed method RMNet-IPM and CFRNet. For CFRNet, the elapsed time increased when the size of the action space $|\mathcal{A}|$ became large. The main reason for this is the calculation of distance between the representation distributions for each pair of actions

Table B.3: Semi-synthetic results on normalized mean cumulative gain (NMCG) and other metrics in $k = 2$. Best and second-best methods are in bold.

	Normalized mean cumulative gain @ k=2				MSE^u				$\text{ER}^u_{k=2}$			
$\lvert\mathcal{A}\rvert$	8	16	32	64	8	16	32	64	8	16	32	64
Method												
OLS	0.08 ± 0.15	0.01 ± 0.19	−0.00 ± 0.12	0.03 ± 0.10	1.12	1.89	1.70	5.86	0.374	0.215	0.118	0.061
RF	0.27 ± 0.08	0.34 ± 0.07	0.33 ± 0.05	0.38 ± 0.05	1.03	0.87	0.93	1.07	0.358	0.205	0.111	0.057
kNN	0.27 ± 0.04	0.30 ± 0.06	0.30 ± 0.04	0.37 ± 0.02	**0.59**	**0.64**	**0.64**	**0.63**	0.356	0.206	0.112	0.057
BART	0.13 ± 0.13	0.18 ± 0.12	0.15 ± 0.10	0.09 ± 0.09	1.06	1.05	1.15	1.63	0.371	0.213	0.116	0.060
M-DNN	0.30 ± 0.08	0.43 ± 0.05	0.29 ± 0.06	0.34 ± 0.04	0.81	0.82	0.81	0.85	0.357	0.205	0.114	0.057
S-DNN	0.32 ± 0.08	0.27 ± 0.10	0.32 ± 0.07	0.45 ± 0.05	0.69	0.68	0.74	0.72	0.353	0.208	0.111	0.055
CFRNet	0.44 ± 0.09	0.41 ± 0.08	0.34 ± 0.07	0.35 ± 0.04	0.79	0.78	0.83	0.86	**0.334**	0.204	0.111	0.057
RankNet	0.37 ± 0.07	0.29 ± 0.09	0.37 ± 0.07	0.44 ± 0.05	6.98	11.34	8.26	2.60	0.349	0.205	0.109	0.055
RMNet-IPM	**0.66 ± 0.01**	**0.43 ± 0.06**	**0.49 ± 0.04**	**0.50 ± 0.05**	0.73	0.85	0.73	0.73	**0.304**	**0.197**	**0.106**	**0.053**
RMNet-HSIC	**0.49 ± 0.07**	**0.49 ± 0.09**	**0.52 ± 0.05**	**0.65 ± 0.04**	**0.44**	**0.61**	**0.66**	**0.30**	0.334	**0.194**	**0.104**	**0.050**

Table B.4: Semi-synthetic results on normalized mean cumulative gain (NMCG) and other metrics in $k = 4$. Best and second-best methods are in bold.

	Normalized mean cumulative gain @ k=4				MSE^u				$\text{ER}^u_{k=4}$			
$\lvert\mathcal{A}\rvert$	8	16	32	64	8	16	32	64	8	16	32	64
Method												
OLS	0.18 ± 0.15	−0.01 ± 0.15	−0.00 ± 0.11	0.02 ± 0.07	1.12	1.89	1.70	5.86	0.471	0.373	0.221	0.117
RF	0.24 ± 0.08	0.34 ± 0.07	0.34 ± 0.05	0.36 ± 0.05	1.03	0.87	0.93	1.07	0.459	0.330	0.198	0.104
kNN	0.19 ± 0.05	0.26 ± 0.06	0.28 ± 0.04	0.36 ± 0.02	**0.59**	0.64	**0.64**	**0.63**	0.467	0.339	0.203	0.106
BART	0.11 ± 0.12	0.23 ± 0.11	0.16 ± 0.10	0.13 ± 0.09	1.06	1.05	1.15	1.63	0.485	0.350	0.212	0.114
M-DNN	**0.42 ± 0.05**	0.38 ± 0.06	0.28 ± 0.06	0.26 ± 0.04	0.79	0.82	0.82	0.85	**0.418**	0.334	0.207	0.110
S-DNN	0.28 ± 0.08	0.28 ± 0.10	0.31 ± 0.07	0.44 ± 0.05	0.68	**0.59**	0.79	0.69	0.451	0.339	0.198	0.098
CFRNet	**0.46 ± 0.05**	0.40 ± 0.06	0.30 ± 0.06	0.26 ± 0.03	0.79	0.79	0.86	0.86	**0.408**	0.327	0.204	0.111
RankNet	0.33 ± 0.07	0.28 ± 0.10	0.36 ± 0.06	0.44 ± 0.04	6.20	11.09	7.98	4.48	0.439	0.331	0.192	0.099
RMNet-IPM	0.39 ± 0.06	**0.43 ± 0.07**	**0.43 ± 0.05**	**0.49 ± 0.05**	0.69	0.77	0.67	0.72	0.422	**0.318**	**0.188**	**0.095**
RMNet-HSIC	0.35 ± 0.09	**0.49 ± 0.07**	**0.47 ± 0.05**	**0.62 ± 0.02**	**0.65**	**0.53**	**0.55**	**0.35**	0.438	**0.305**	**0.180**	**0.087**

$\sum_{a \neq a'} D_{\text{IPM}}\left(p_a(\phi), p_{a'}(\phi)\right)$ in Fig. 4.3(a). The decrease of the elapsed time for RMNet is mainly due to the sample sizes shown in Table B.2.

Semi-synthetic results for $k > 1$

Table. 4.2 shows the results for $k = 1$. We also evaluated with respect to $k = 2$ and $k = 4$ as shown in Table B.3 and Table B.4, respectively. The metric for $k > 1$ is defined as the following normalized mean cumulative gain (NMCG):

$$\text{NMCG}_k(f) \coloneqq \frac{\mathbb{E}_x\left[\sum_{a:\text{rank}(f(x,a))\leq k} y_a\right]}{\mathbb{E}_x\left[\sum_{a:\text{rank}(y_a)\leq k} y_a\right]}.$$

The model selection is also performed with respect to NMCG_k. The results were similar to that in $k = 1$, which demonstrates the robustness of the proposed methods with respect to the choice of k (and thus the policy π_k).

List of Publications

1. A. Tanimoto. Combinatorial Q-learning for condition-based infrastructure maintenance. *IEEE Access*, 9:46788–46799, 2021.

2. A. Tanimoto, S. Yamada, T. Takenouchi, M. Sugiyama, and H. Kashima. Improving imbalanced classification using near-miss instances. Submitted to Expert Systems with Applications.

3. A. Tanimoto, T. Sakai, T. Takenouchi, and H. Kashima. Regret minimization for causal inference on large treatment space. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 946–954, 2021b.

4. A. Tanimoto, T. Sakai, T. Takenouchi, and H. Kashima. Causal combinatorial factorization machines for set-wise recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 498–509, 05 2021a.

Other Publications

1. S. Hayashi, A. Tanimoto, and H. Kashima. Long-term prediction of small time-series data using generalized distillation. In *2019 IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

2. 林 勝悟, 谷本 啓, 鹿島 久嗣. “一般化蒸留を用いた少量時系列データの長期予測”. 人工知能学会論文誌, 35(5), B-K33_1-9, 2020.

References

S. S. Adlinge and A. Gupta. Pavement deterioration and its causes. *International Journal of Innovative Research and Development*, 2(4):437–450, 2013.

N. Aissani, B. Beldjilali, and D. Trentesaux. Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. *Engineering Applications of Artificial Intelligence*, 22(7):1089–1103, 2009.

A. Ali-Gombe and E. Elyan. Mfc-gan: class-imbalanced dataset classification using multiple fake class generative adversarial network. *Neurocomputing*, 361: 212–221, 2019.

C. Andriotis and K. Papakonstantinou. Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliability Engineering & System Safety*, 212:107551, 2021.

K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34 (6):26–38, 2017.

P. C. Austin. An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate Behavioral Research*, 46 (3):399–424, 2011.

R. Ballester-Ripoll, E. G. Paredes, and R. Pajarola. Sobol tensor trains for global sensitivity analysis. *arXiv preprint arXiv:1712.00233*, 2017.

R. Ballester-Ripoll, E. G. Paredes, and R. Pajarola. Sobol tensor trains for global sensitivity analysis. *Reliability Engineering & System Safety*, 183:311–322, 2019.

P. Barach and S. D. Small. Reporting and preventing medical mishaps: lessons from non-medical near miss reporting systems. *BMJ: British medical journal*, 320(7237):759, 2000.

S. R. Barde, S. Yacout, and H. Shin. Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. *Journal of Intelligent Manufacturing*, 30(1):147–161, 2019.

S. Barua, M. M. Islam, X. Yao, and K. Murase. MWMOTE–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):405–425, 2012.

S. Bonner and F. Vasile. Causal embeddings for recommendation. In *ACM Conference on Recommender Systems (RecSys)*, pages 104–112, New York, NY, USA, 2018. ACM.

L. Bottou, J. Peters, J. Quiñonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

A. Bousdekis, B. Magoutas, D. Apostolou, and G. Mentzas. Review, analysis and synthesis of prognostic-based decision support methods for condition based maintenance. *Journal of Intelligent Manufacturing*, 29(6):1303–1316, 2018.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, pages 89–96, 2005.

S. Chambon and J.-M. Moliard. Automatic road pavement assessment with image processing: review and comparison. *International Journal of Geophysics*, 2011, 2011.

P.-C. Chang and C.-Y. Lai. A hybrid system combining self-organizing maps with case-based reasoning in wholesaler's new-release book forecasting. *Expert Systems With Applications*, 29(1):183–192, 2005.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006. ISBN 0262033585.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting. In *European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 107–119. Springer, 2003.

C. P. Chen and C.-Y. Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences*, 275:314–347, 2014.

X.-w. Chen and M. Wasikowski. Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 124–132. ACM, 2008.

H. Cloke and F. Pappenberger. Ensemble flood forecasting: A review. *Journal of Hydrology*, 375(3-4):613–626, 2009.

G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7–18, 2006.

M. Compare, L. Bellani, E. Cobelli, and E. Zio. Reinforcement learning-based flow management of gas turbine parts under stochastic failures. *The International Journal of Advanced Manufacturing Technology*, 99(9-12):2981–2992, 2018.

P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *ACM Conference on Recommender Systems (RecSys)*, pages 39–46. ACM, 2010.

M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2292–2300, 2013.

R. Dekker, R. E. Wildeman, and F. A. Van der Duyn Schouten. A review of multi-component maintenance models with economic dependence. *Mathematical Methods of Operations Research*, 45(3):411–435, 1997.

D. Dheeru and E. Karra Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

J. P. Dmochowski, P. Sajda, and L. C. Parra. Maximum likelihood in cost-sensitive learning: Model specification, approximations, and upper bounds. *Journal of Machine Learning Research*, 11(Dec):3313–3332, 2010.

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

C. Elkan. The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 17, pages 973–978. Lawrence Erlbaum Associates Ltd, 2001.

T. Elsken, J. H. Metzen, F. Hutter, et al. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019.

S. M. Famurewa, T. Xin, M. Rantatalo, and U. Kumar. Optimisation of maintenance track possession time: A tamping case study. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 229 (1):12–22, 2015.

A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 2018.

D. Fuqua and T. Razzaghi. A cost-sensitive convolution neural network learning for control chart pattern recognition. *Expert Systems with Applications*, 150: 113275, 2020.

A. Gandomi and M. Haider. Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2):137–144, 2015.

L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. MIT press, 2007.

Y. Gong, Y. Zhu, L. Duan, Q. Liu, Z. Guan, F. Sun, W. Ou, and K. Q. Zhu. Exact-k recommendation via maximal clique optimization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 617–626, New York, NY, USA, 2019. ACM.

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 63–77. Springer, 2005.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems (NIPS)*, pages 585–592, 2008.

S. Gu, T. Lillicrap, I. Sutskever, and S. Levine. Continuous deep Q-learning with model-based acceleration. In *International Conference on Machine Learning (ICML)*, pages 2829–2838, 2016.

G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

N. Hassanpour and R. Greiner. Counterfactual regression with importance sampling weights. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5880–5887, 2019.

S. Hayashi, A. Tanimoto, and H. Kashima. Long-term prediction of small time-series data using generalized distillation. In *2019 IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

H. He, Y. Bai, E. A. Garcia, and S. Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1322–1328. IEEE, 2008.

X. He, K. Zhao, and X. Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021.

H. W. Heinrich, D. C. Petersen, N. R. Roos, and S. Hazlett. *Industrial Accident Prevention: A Safety Management Approach*. McGraw-Hill Companies, 1980.

K. Hendricks and A. Sorensen. Information and the skewness of music sales. *Journal of Political Economy*, 117(2):324–369, 2009.

D. Herremans, D. Martens, and K. Sörensen. Dance hit song prediction. *Journal of New Music Research*, 43(3):291–302, 2014.

J. L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

H. Hong and S. Wang. Stochastic modeling of pavement performance. *International Journal of Pavement Engineering*, 4(4):235–243, 2003.

A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

E. Ie, V. Jain, J. Wang, S. Narvekar, R. Agarwal, R. Wu, H.-T. Cheng, T. D. Chandra, and C. Boutilier. SlateQ: A tractable decomposition for reinforcement learning with recommendation sets. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

G. W. Imbens and J. M. Wooldridge. Recent developments in the econometrics of program evaluation. *Journal of Economic Literature*, 47(1):5–86, 2009.

D. Inaudi and B. Glisic. Long-range pipeline monitoring by distributed fiber optic sensing. *Journal of Pressure Vessel Technology*, 132(1):011701, 2010.

N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

A. K. Jardine and A. H. Tsang. *Maintenance, Replacement, and Reliability: Theory and Applications*. CRC press, 2005.

R. Jiang, S. Gowal, Y. Qian, T. Mann, and D. J. Rezende. Beyond greedy ranking: Slate optimization via list-CVAE. In *International Conference on Learning Representations (ICLR)*, 2019.

T. Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

F. Johansson, U. Shalit, and D. Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning (ICML)*, pages 3020–3029, 2016.

F. D. Johansson, D. Sontag, and R. Ranganath. Support and invertibility in domain-invariant representations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 527–536, 2019.

S. Jones, C. Kirchsteiger, and W. Bjerke. The importance of near miss reporting to further improve safety performance. *Journal of Loss Prevention in the Process Industries*, 12(1):59–67, 1999.

J. D. Kang, J. L. Schafer, et al. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science*, 22(4):523–539, 2007.

K. H. Kim and S. Y. Sohn. Hybrid neural network with cost-sensitive support vector machine for class-imbalanced multimodal data. *Neural Networks*, 130: 176–184, 2020.

S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *2007 6th International Symposium on Information Processing in Sensor Networks*, pages 254–263, Apr. 2007.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *International Conference on Learning Representations (ICLR)*, 2015.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

A. G. Kök, M. L. Fisher, and R. Vaidyanathan. Assortment planning: Review of literature and industry practice. In *Retail Supply Chain Management*, pages 99–153. Springer, 2008.

V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1008–1014, 2000.

B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

A. Kuhnle, J. Jakubik, and G. Lanza. Reinforcement learning for opportunistic maintenance optimization. *Production Engineering*, 13(1):33–41, 2019.

A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1179–1191, 2020.

M. T. Lash and K. Zhao. Early predictions of movie success: Statistical analysis with missing datathe who, what, and when of profitability. *Journal of Management Information Systems*, 33(3):874–903, 2016.

J. Lee, J. Ni, D. Djurdjanovic, H. Qiu, and H. Liao. Intelligent prognostics tools and e-maintenance. *Computers in Industry*, 57(6):476–489, 2006.

K.-c. Lee, B. Orten, A. Dasdan, and W. Li. Estimating conversion rate in display advertising from past erformance data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 768–776, 2012.

J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42, 2018.

S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

H.-N. Li, D.-S. Li, and G.-B. Song. Recent applications of fiber optic sensors to health monitoring in civil engineering. *Engineering Structures*, 26(11):1647–1657, 2004.

Y. Li and P. Nilkitsaranont. Gas turbine performance prognostic for condition-based maintenance. *Applied Energy*, 86(10):2152–2161, 2009.

C. X. Ling and V. S. Sheng. Cost-sensitive learning and the class imbalance problem. *Encyclopedia of Machine Learning*, pages 231–235, 2010.

R. J. Little and D. B. Rubin. *Statistical Analysis with Missing Data*, volume 793. John Wiley & Sons, 2019.

J. Liu, L. Sun, W. Chen, and H. Xiong. Rebalancing bike sharing systems: A multi-source data smart optimization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1005–1014. ACM, 2016.

Y. Liu, Y. Chen, and T. Jiang. Dynamic selective maintenance optimization for multi-state systems over a finite horizon: A deep reinforcement learning approach. *European Journal of Operational Research*, 283(1):166–181, 2020.

R. Lopez, C. Li, X. Yan, J. Xiong, M. Jordan, Y. Qi, and L. Song. Cost-effective incentive allocation via structured counterfactual inference. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 4997–5004, 2020.

D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations (ICLR)*, pages 1–10, 2016.

P. Manchanda, A. Ansari, and S. Gupta. The "shopping basket" : A model for multicategory purchase incidence decisions. *Marketing Science*, 18(2):95–114, 1999.

B. M. Marlin, R. S. Zemel, S. Roweis, and M. Slaney. Collaborative filtering and the missing at random assumption. In *Uncertainty in Artificial Intelligence*, pages 267–275. AUAI Press, 2007.

V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

T. M. Moerland, J. Broekens, and C. M. Jonker. Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*, 2020.

A. Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, pages 429–443, 1997.

K. Napierała, J. Stefanowski, and S. Wilk. Learning from imbalanced data in presence of noisy and borderline examples. In *International Conference on Rough Sets and Current Trends in Computing*, pages 158–167. Springer, 2010.

N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari. Cost-sensitive learning with noisy labels. *The Journal of Machine Learning Research*, 18(1):5666–5698, 2017.

K.-A. Nguyen, P. Do, and A. Grall. Multi-level predictive maintenance for multi-component systems. *Reliability Engineering & System Safety*, 144:83–94, 2015.

Q. Nguyen, H. Valizadegan, and M. Hauskrecht. Learning classification with auxiliary probabilistic information. In *International Conference on Data Mining (ICDM)*, pages 477–486, 2011a.

Q. Nguyen, H. Valizadegan, A. Seybert, and M. Hauskrecht. Sample-efficient learning with auxiliary class-label information. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1004. American Medical Informatics Association, 2011b.

Q. Nguyen, H. Valizadegan, and M. Hauskrecht. Learning classification models with soft-label information. *Journal of the American Medical Informatics Association*, 21(3):501–508, 2014.

R. P. Nicolai and R. Dekker. Optimal maintenance of multi-component systems: a review. In *Complex System Maintenance Handbook*, pages 263–286. Springer, 2008.

C. Nugteren and V. Codreanu. CLTune: A generic auto-tuner for opencl kernels. In *International Symposium on Embedded Multicore/Many-core Systems-on-Chip*, pages 195–202. IEEE, 2015.

I. S. Papadakis and P. R. Kleindorfer. Optimizing infrastructure network maintenance when benefits are interdependent. *OR Spectrum*, 27(1):63–84, 2005.

J. Pearl. *Causality*. Cambridge university press, 2009.

C.-Y. Peng and S.-T. Tseng. Statistical lifetime inference with skew-wiener linear degradation models. *IEEE Transactions on Reliability*, 62(2):338–350, 2013.

P. Peng, R. C.-W. Wong, and P. S. Yu. Learning on probabilistic labels. In *SIAM International Conference on Data Mining (SDM)*, pages 307–315. SIAM, 2014.

Y. Peng, M. Dong, and M. J. Zuo. Current status of machine prognostics in condition-based maintenance: a review. *The International Journal of Advanced Manufacturing Technology*, 50(1-4):297–313, 2010.

S. Rendle. Factorization machines. In *International Conference on Data Mining (ICDM)*, pages 995–1000. IEEE, 2010.

S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

M. Riedmiller. Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning (ECML)*, pages 317–328. Springer, 2005.

D. B. Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

J. A. Sáez, J. Luengo, J. Stefanowski, and F. Herrera. SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203, 2015.

Y. Saito and S. Yasui. Counterfactual cross-validation: Stable model selection procedure for causal inference models. In *International Conference on Machine Learning (ICML)*, 2020.

Y. Saito, S. Aihara, M. Matsutani, and Y. Narita. A large-scale open dataset for bandit algorithms. *arXiv preprint arXiv:2008.07146*, 2020.

T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning (ICML)*, volume 48, pages 1670–1679, 2016.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. In *International Conference on Machine Learning (ICML)*, pages 1255–1262, 2012.

P. Schwab, L. Linhardt, and W. Karlen. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*, 2018.

C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano. RUSBoost: A hybrid approach to alleviating class imbalance. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(1):185–197, 2009.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge university press, 2014.

U. Shalit, F. D. Johansson, and D. Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning (ICML)*, pages 3076–3085, 2017.

H. A. Simon. Spurious correlation: A causal interpretation. *Journal of the American Statistical Association*, 49(267):467–479, 1954.

S. P. Singh and R. S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22(1-3):123–158, 1996.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems (NIPS)*, 25:2951–2959, 2012.

B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, G. R. Lanckriet, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

N. K. Srivastava and S. Mondal. Development of predictive maintenance model for n-component repairable system using nhpp models and system availability concept. *Global Business Review*, 17(1):105–115, 2016.

Z. Su, A. Jamshidi, A. Núñez, S. Baldi, and B. De Schutter. Multi-level condition-based maintenance planning for railway infrastructures–a scenario-based chance-constrained approach. *Transportation Research Part C: Emerging Technologies*, 84:92–123, 2017.

Z. Su, A. Jamshidi, A. Núñez, S. Baldi, and B. De Schutter. Integrated condition-based track maintenance planning and crew scheduling of railway networks. *Transportation Research Part C: Emerging Technologies*, 105:359–384, 2019.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

A. Tanimoto. Combinatorial Q-learning for condition-based infrastructure maintenance. *IEEE Access*, 9:46788–46799, 2021.

A. Tanimoto, S. Yamada, T. Takenouchi, M. Sugiyama, and H. Kashima. Improving imbalanced classification using near-miss instances. Submitted to Expert Systems with Applications.

A. Tanimoto, T. Sakai, T. Takenouchi, and H. Kashima. Causal combinatorial factorization machines for set-wise recommendation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 498–509, 05 2021a.

A. Tanimoto, T. Sakai, T. Takenouchi, and H. Kashima. Regret minimization for causal inference on large treatment space. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 946–954, 2021b.

Z. Tian and H. Liao. Condition based maintenance optimization for multi-component systems using proportional hazards model. *Reliability Engineering & System Safety*, 96(5):581–589, 2011.

L. Torgo. Regression data sets, 2001. *URL http://www.liaad.up.pt/l̃torgo/Regression/DataSets.html*, 2018.

T. W. van der Schaaf. Near miss reporting in the chemical process industry: An overview. *Microelectronics Reliability*, 35(9-10):1233–1243, 1995.

A. Van Horenbeek and L. Pintelon. A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering & System Safety*, 120:39–50, 2013.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.

V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

S. Varma and R. Simon. Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, 7(1):91, 2006.

F. Vasile, D. Lefortier, and O. Chapelle. Cost-sensitive learning for utility optimization in online advertising auctions. In *Proceedings of the ADKDD'17*, page 8. ACM, 2017.

K. Verbert, B. De Schutter, and R. Babuška. Timely condition-based maintenance planning for multi-component systems. *Reliability Engineering & System Safety*, 159:310–321, 2017.

F. Wang, X. Fang, L. Liu, Y. Chen, J. Tao, Z. Peng, C. Jin, and H. Tian. Sequential evaluation and generation framework for combinatorial recommender system. *arXiv preprint arXiv:1902.00245*, 2019a.

X. Wang, J. Qi, K. Ramamohanarao, Y. Sun, B. Li, and R. Zhang. A joint optimization approach for personalized recommendation diversification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, pages 597–609. Springer, 2018.

X. Wang, R. Zhang, Y. Sun, and J. Qi. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning (ICML)*, pages 6638–6647, 2019b.

C. J. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3-4):279–292, 1992.

J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang. NI-MWMOTE: An improving noise-immunity majority weighted minority oversampling technique for imbalanced classification problems. *Expert Systems with Applications*, page 113504, 2020a.

S. Wei, Y. Bao, and H. Li. Optimal policy for structure maintenance: A deep reinforcement learning framework. *Structural Safety*, 83:101906, 2020b.

A. Xanthopoulos, A. Kiatipis, D. E. Koulouriotis, and S. Stieger. Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system. *IEEE Access*, 6:576–588, 2017.

Y. Xue and M. Hauskrecht. Learning of classification models from noisy soft-labels. In *European Conference on Artificial Intelligence (ECAI)*, pages 1618–1619, 2016.

Y. Xue and M. Hauskrecht. Efficient Learning of Classification Models from Soft-label Information by Binning and Ranking. In *International Florida AI Research Society Conference. Florida AI Research Symposium*, page 164, 2017.

L. Yao, Q. Dong, J. Jiang, and F. Ni. Deep reinforcement learning for long-term pavement maintenance planning. *Computer-Aided Civil and Infrastructure Engineering*, 2020.

Q. Yao, M. Wang, Y. Chen, W. Dai, H. Yi-Qi, L. Yu-Feng, T. Wei-Wei, Y. Qiang, and Y. Yang. Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*, 2018.

J. Yoon, J. Jordon, and M. van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations (ICLR)*, 2018.

M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola. Deep sets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3391–3401, 2017.

N. Zhang and W. Si. Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks. *Reliability Engineering & System Safety*, 203:107094, 2020.

Y. Zhang, A. Bellot, and M. van der Schaar. Learning overlapping representations for the estimation of individualized treatment effects. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1005–1014, 2020.

H. Zhao, R. T. D. Combes, K. Zhang, and G. Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning (ICML)*, volume 97, pages 7523–7532, 2019.

S. Zhao, H. Ren, A. Yuan, J. Song, N. Goodman, and S. Ermon. Bias and generalization in deep generative models: an empirical study. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10815–10824, 2018.

H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li. Improving deep neural networks using softplus units. In *2015 IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 1–4. IEEE, 2015.

R. R. Zhou, N. Serban, and N. Gebraeel. Degradation modeling applied to residual lifetime prediction using functional data analysis. *The Annals of Applied Statistics*, pages 1586–1610, 2011.

C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *International Conference on World Wide Web (WWW)*, pages 22–32. ACM, 2005.

H. Zou, P. Cui, B. Li, Z. Shen, J. Ma, H. Yang, and Y. He. Counterfactual prediction for bundle treatment. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 19705–19715. Curran Associates, Inc., 2020.