Enhancing Counterfactual Classification Performance via Self-Training

Ruijiang Gao[*1]**, Max Biggs**[2]**, Wei Sun**[3]**, Ligong Han**[4]

[1] University of Texas at Austin
[2] University of Virginia
[3] IBM Research
[4] Rutgers University
ruijiang@utexas.edu, biggsm@darden.virginia.edu, sunw@us.ibm.com, lh599@rutgers.edu

Abstract

Unlike traditional supervised learning, in many settings only partial feedback is available. We may only observe outcomes for the chosen actions, but not the counterfactual outcomes associated with other alternatives. Such settings encompass a wide variety of applications including pricing, online marketing and precision medicine. A key challenge is that observational data are influenced by historical policies deployed in the system, yielding a biased data distribution. We approach this task as a domain adaptation problem and propose a self-training algorithm which imputes outcomes with *categorical* values for *finite* unseen actions in the observational data to simulate a randomized trial through pseudolabeling, which we refer to as Counterfactual Self-Training (CST). CST iteratively imputes pseudolabels and retrains the model. In addition, we show input consistency loss can further improve CST performance which is shown in recent theoretical analysis of pseudolabeling. We demonstrate the effectiveness of the proposed algorithms on both synthetic and real datasets.

Introduction

Counterfactual inference (Pearl et al. 2000) attempts to address a question central to many applications - *What would be the outcome had an alternative action was chosen?* This includes selecting relevant ads to engage with users in online marketing (Li et al. 2010), determining prices that maximize profit in revenue management (Bertsimas and Kallus 2016), or designing the most effective personalized treatment for a patient in precision medicine (Xu, Xu, and Saria 2016). With observational data, we have access to past actions, their outcomes, and possibly some context, but in many cases not the complete knowledge of the historical policy which gave rise to the action (Shalit, Johansson, and Sontag 2017). Consider a pricing setting in the form of targeted promotions. We might record information about a customer (context), the promotion offered (action) and whether an item was purchased (outcome), but we do not know why a particular promotion was selected.

Unlike traditional supervised learning with full feedback, we only observe partial feedback for the chosen action in observational data, but not the outcomes associated with other

---

alternatives (i.e., in the pricing example, we do not observe what would occur if a different promotion was offered). In contrast to the gold standard of a randomized controlled trial, observational data are influenced by the historical policy deployed in the system which may over or underrepresent certain actions, yielding a biased data distribution. A naive but widely used approach is to learn a machine learning algorithm directly from observational data and use it for prediction. This is often referred to as the direct method (DM) (Dudík et al. 2014). Failure to account for the bias introduced by the historical policy often results in an algorithm which has high accuracy on the data it was trained on, but performs considerably worse under different policies (Shalit, Johansson, and Sontag 2017). For example, in the pricing setting, if historically most customers who received high promotion offers have a certain profile, then a model based on direct method may fail to produce reliable predictions on these customers when low offers are given. In this paper, we focus on ***Counterfactual Classification (CC)***, where the outcome of each action is categorical. For instance, one may want to estimate whether a customer will purchase an item in a pricing application or a user clicks on an ad in online marketing. This setting is in contrast to the body of literature on treatment effect estimation which assumes a continuous outcome (Yoon, Jordon, and van der Schaar 2018; Shalit, Johansson, and Sontag 2017; Shi, Blei, and Veitch 2019; Alaa and van der Schaar 2017).

To overcome the limitations of the direct method, Shalit, Johansson, and Sontag (2017); Johansson, Shalit, and Sontag (2016); Lopez et al. (2020) cast counterfactual learning as a *domain adaptation* problem, where the source domain is observational data and the target domain is a *randomized trial* whose assignment of actions follows a uniform distribution for a given context. The key idea is to map contextual features to an embedding space and jointly learn a representation that encourages similarity between these two domains, leading to better counterfactual inference. The embedding is generally learned by a neural network and the estimation of the domain gap is usually slow to compute. Such representation learning approaches also pose challenges for interpretable models since the learned representation often loses the semantic meaning of origin features. In this paper, while we also view counterfactual classification as a domain adaptation problem between observational data and an ideal

randomized trial, we take a different approach - instead of estimating the domain gap between the two distributions via an embedding, we explicitly simulate a randomized trial by imputing pseudolabels for the unobserved actions in the observational data. The optimization process is done by iteratively updating the pseudolabels and a model that is trained on both the factual and the counterfactual data, as illustrated in Figure 1. As this method works in a self-supervised fashion (Zou et al. 2018; Amini and Gallinari 2002), we refer to our proposed framework as Counterfactual Self-Training (CST).

The contributions of our paper are as follows. First, we propose a novel pseudolabeling self-training algorithm for counterfactual inference. To the best of our knowledge, this is the first application of a self-training algorithm using pseudolabeling for the counterfactual classification problem. In contrast to the existing methods from domain adaption on counterfactual inference, CST shares the advantage of PL methods which are data-modality-agnostic and work for a wide range of models. Second, we show incorporating a counterfactual input consistency loss can further improve CST's performance, which is consistent with recent theoretical analysis of pseudolabeling when the underlying data distribution and pseudolabeler satisfies certain assumptions such as sufficiently high labeling accuracy and the underlying data is well-separated so that the wrong pseudolabels can be denoised by close correct labels. This finding suggests promising results for pseudolabeling based approaches in counterfactual classification problems. Third, we present comprehensive experiments on toy data, several synthetic datasets and real datasets converted from multi-label classification tasks to evaluate our method against state-of-the-art baselines. In all experiments, CST shows competitive or superior performance against all the baselines.

Related Work

Counterfactual policy optimization and evaluation has received a lot of attention in the machine learning community in recent years (Swaminathan and Joachims 2015a; Joachims, Swaminathan, and de Rijke 2018; Shalit, Johansson, and Sontag 2017; Lopez et al. 2020; Kallus 2019; Kallus and Zhou 2018; Wang et al. 2019; Gao et al. 2021; Biggs, Gao, and Sun 2021). Most of the proposed algorithms can be divided into two categories: counterfactual risk minimization (CRM) and direct method (DM). Both can be used together to construct doubly robust estimators (Dudík et al. 2014) to further improve efficiency. CRM, also known as off-policy learning or batch learning from bandit feedback (BLBF), typically utilizes inverse propensity weighting (IPW) (Rosenbaum 1987; Rosenbaum and Rubin 1983) to account for the bias in the data. Swaminathan and Joachims (2015a) introduces the CRM principle with a variance regularization term derived from an empirical Bernstein bound (Maurer and Pontil 2009) for finite samples. In order to reduce the variance of the IPW estimator, Swaminathan and Joachims (2015b) proposes a self-normalized estimator, while BanditNet (Joachims, Swaminathan, and de Rijke 2018) utilizes the baseline technique (Greensmith, Bartlett, and Baxter 2004) in deep nets. As pointed out

by Lefortier et al. (2016), CRM-based methods tend to struggle with medium to large action spaces in practice. Moreover, CRM-based methods generally require a known (or accurately estimated) and stochastic logging policy, along with full support on the action space. When either one of the requirements is violated, Sachdeva, Su, and Joachims (2020); Kang, Schafer et al. (2007) observe that the direct method often demonstrates a more robust performance. In CRM, a continuous reward is often assumed and the objective is to either maximize policy reward or evaluate the policy reward accurately. In the counterfactual classification problem, such rewards are not well-defined and may be later defined differently by various possible downstream tasks.

Another line of research in learning from observational data is often referred to as estimating Individualized Treatment Effects (ITE) (Shpitser and Pearl 2012; Forney and Bareinboim 2019; Yin et al. 2021) or conditional average treatment effect (CATE), which is defined as the difference of expected outcomes between two actions, with respect to a given context. The main challenge of identifying ITE is that unlike an ideal randomized trial, observational data is biased and we do not have the access to the counterfactuals. Hill (2011) uses a bayesian nonparametric algorithm to address this issue. Yoon, Jordon, and van der Schaar (2018) propose using generative adversarial nets to capture the uncertainty in the counterfactual distributions to facilitate ITE estimation. Johansson, Shalit, and Sontag (2016); Shalit, Johansson, and Sontag (2017) approach counterfactual inference with representation learning and domain adaptation. Lopez et al. (2020) further extends this framework to multiple treatments using Hilbert-Schmidt Independence Criterion (HSIC) (Gretton et al. 2008) and achieves state-of-the-art performance. The HSIC proposed in Lopez et al. (2020) has a computation time of at least $\mathcal{O}(N^2)$, making its training process slow. While the aforementioned methods and our approach can be viewed as extensions to the direct method, similarly to CRM methods, most of them require a specific reward function a priori and are either designed for binary treatment or cannot directly be applied in our Counterfactual Classification problem which treats outcome as a categorical variable. We tackle the domain adaptation problem differently by explicitly augmenting the observational data to create a simulated randomized trial via self-training. Some works develop meta-learners classified as X-, T-, S-learners (Künzel et al. 2019), for example, Hill (2011) is an instance of S-learner. Our approach is similar to X-learner in the sense that both methods use pseudolabels to generate counterfactuals. However, X-learner only considers binary actions with a continuous outcome, while CST allows multiple actions with categorical outcomes. Moreover, X-learner is one-shot while CST is trained in an iterative fashion.

Self-training algorithms have been widely studied in semi-supervised learning and domain adaptation problems (Nigam et al. 2000; Amini and Gallinari 2002; Grandvalet and Bengio 2005; Zou et al. 2019; Han et al. 2019) in various forms. Grandvalet and Bengio (2005) propose using entropy regularization for semi-supervised learning as a class-overlapping measure to utilize unlabeled data. Nigam et al. (2000); Amini and Gallinari (2002); Zou et al. (2019) for-
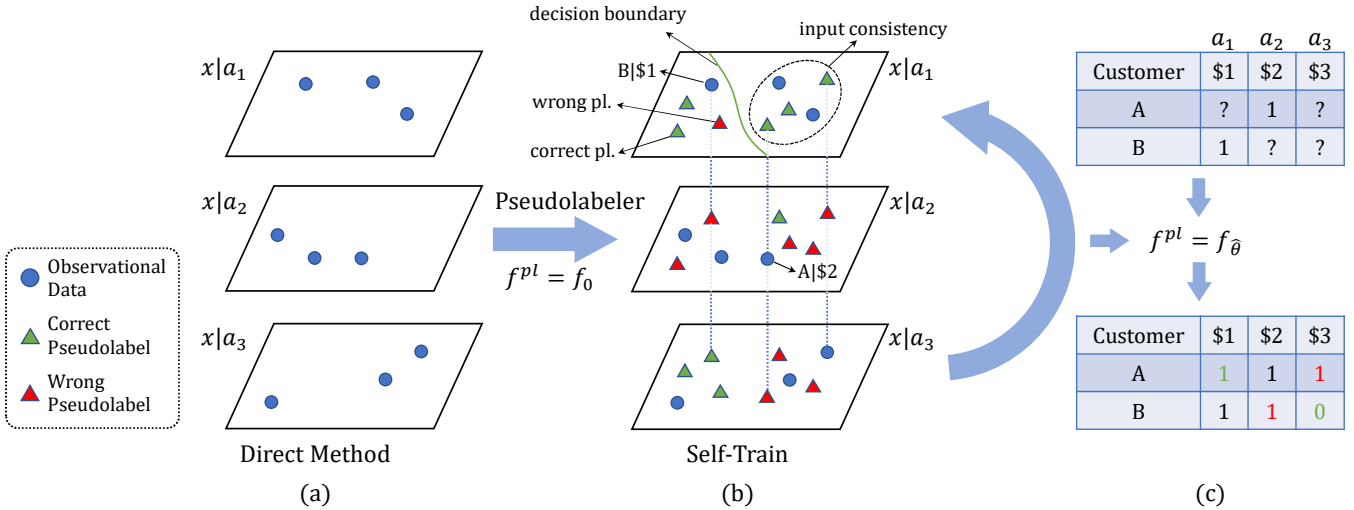
Figure 1: Illustration of the proposed Counterfactual Self-Training (CST) framework. There are two sales records (observational data) shown in the table, *i.e.*, Customer A was offered $2 and bought an item; Customer B was offered $1 and also bought. The question marks in the tables represent the counterfactual outcomes which we do not observe. For all these unseen counterfactual outcomes, pseudolabels which are colored in the tables are imputed by a model and are used to augment the observational data. The model is subsequently updated by training on both the imputed counterfactual data and the factual data. This iterative training procedure continues until it converges. Blue points represent factual observations. Triangles represent imputed pseudolabels. For pseudolabels, red means wrong labels and green means correct labels. The input consistency regularization correct the wrong pseudolabel with its neighbors' correct labels, resulting in a refined decision boundary.

mulate the pseudolabel imputation as classification EM algorithm and show its convergence under proper initialization. Recent theoretical analysis (Wei et al. 2020) and empirical evidence shows input consistency loss such as VAT loss (Miyato et al. 2018) can further improve pseudolabeling in semi-supervised learning. Han et al. (2019) points out that pseudolabel imputation can be viewed as minimizing min-entropy as a type of Rényi entropy $\frac{1}{1-\alpha} \log(\sum_{i=1}^{n} p_i^{\alpha})$ when $\alpha \to \infty$, and Shannon entropy in (Grandvalet and Bengio 2005) is the case when $\alpha \to 1$. Self-training is also shown to be effective in semi-supervised learning for many other machine learning models besides neural networks (Tanha, van Someren, and Afsarmanesh 2017; Li et al. 2008). Unlike traditional self-training where the target domain is given by the problem, we propose to construct a target domain by imputing pseudolabels on all unseen actions to simulate a pseudo-randomized trial.

Problem Statement

Following the notation in Lopez et al. (2020), we use $\mathcal{X}$ to represent an abstract space and $\mathbb{P}(x)$ is a probability distribution on $\mathcal{X}$. Each sample $x = x_1, \cdots, x_n \in \mathcal{X}^n$ is drawn independently from $\mathbb{P}(x)$. $\mathcal{A}$ is the *discrete* action space that a central agent can select for each sample, after applying action a, a *categorical* outcome $r_a \in \{1, \cdots, m\}$ is revealed to the agent. In precision medicine, $\mathcal{X}$ may represent a patient cohort, $\mathcal{A}$ refers to feasible treatments for a disease, and r can be the indicator of whether a patient recovers after the treatment. Similarly, $\mathcal{X}, \mathcal{A}, r$ can represent visitors, ads shown and whether the visitor clicks in online marketing.

We focus on a pricing example to illustrate our method. We use $x \in \mathcal{X}^n \sim \mathbb{P}(x)$ to denote a customer. Let $\mathcal{A}$ represent finite price options a central agent can offer to customers. After offering price $a \in \mathcal{A}$, the agent observes the response from the customer $r_a \in \{0, 1\}$, *i.e.*, either a 1 (buy) or a 0 (no-buy). Our objective is to learn a function $f(x, a)$ by maximizing the log likelihood

$$\mathbb{E}_{x \sim \mathbb{P}(x)} \sum_{a \in \mathcal{A}} \sum_{r} \mathbb{P}(r|x, a) \log f(r|x, a), \quad (1)$$

where $\mathbb{P}(r|x, a)$ is the true probability of r given x, a and the historical data is generated by $\pi_0(a|x)$, a stochastic assignment policy (Shalit, Johansson, and Sontag 2017; Lopez et al. 2020). The estimation task is often referred to as demand estimation (Wales and Woodland 1983), which is critical for many downstream decisions such as inventory optimization and revenue management (Kök and Fisher 2007; McGill and Van Ryzin 1999). This is in contrast to CRM-based methods which use the final reward as its objective to learn a policy $\pi(a|x)$ that maximizes $\mathbb{E}_{x \sim \mathbb{P}(x), p \sim \pi(a|x)} \mathcal{R}(r, x)$ (Swaminathan and Joachims 2015a), where $\mathcal{R}(\cdot)$ is a pre-defined reward function.

With observational data, the counterfactual outcome is not always identifiable. We use Rubin's potential outcome framework and assume consistency, overlap and ignorability which is a sufficient condition for identifying potential outcomes from historical data (Imbens and Wooldridge 2009; Pearl 2017). Here we formally present the ignorability assumption (Rubin 2005; Shalit, Johansson, and Sontag 2017):

Assumption 1 (Ignorability)**.** *Let $\mathcal{A}$ be action set, x is context (feature), $r_a|x$ is observed outcome for action $a \in \mathcal{A}$*

given context x, $r_a \perp\!\!\!\perp a | x, \forall a \in \mathcal{A}$.

In other words, we assume there is no unobserved confounders. This condition generally cannot be made purely based on data and requires some domain knowledge.

Algorithm

In this section, we introduce the Counterfactual Self-Training (CST) algorithm, which can be viewed as an extension of the direct method via domain adaptation. Unlike existing methods using representation learning, we propose a novel self-training style algorithm to account for the bias inherent in the observational data. The self-training algorithm works in an iterative fashion: First, after training a classifier $f(x, a)$ on a source dataset, pseudolabels are created by the best guess of f. Next, the model is trained on a target dataset, and the trained model is used to generate new pseudolabels. This idea is illustrated in Figure 1.

To formulate the counterfactual learning problem as a domain adaptation problem, the observational data are viewed as data sampled from a source distribution $\mathcal{D}_S = \mathbb{P}(x)\pi(a|x)$. The target domain is a controlled randomized trial on the same feature distribution to ensure a uniformly good approximation on all actions. Our goal is to transfer observational data from the source domain to a simulated pseudo-randomized trial via self-training. More specifically, CST tries to optimize the following objective:

$$
\min_{\theta, \hat{r}} \mathcal{L}_{\text{CST}} = \sum_{i=1}^{N} \Big(\underbrace{\sum_{k=1}^{m} -r_{i,a_i,k} \log f_\theta(r_{i,a_i,k}|x_i, a_i)}_{\mathcal{L}_{src}} +
$$
$$
\sum_{a \in \mathcal{A} \setminus a_i} \sum_{k=1}^{m} -\hat{r}_{i,a,k} \log f_\theta(\hat{r}_{i,a,k}|x_i, a_i) \Big) \quad (2)
$$
$$
s.t. \quad \hat{r}_{i,a} \in \Delta^{(m-1)}
$$

We use $\hat{r}$ to represent imputed pseudolabels, $r_{i,a,k}$ is the k-th entry of $r_{i,a}$, $\hat{r}_{i,a}$ is defined on a simplex and $r_{i,a}$ is one hot encoded. Note when counterfactual outcomes are available, this objective corresponds to the true negative log likelihood function for multi-label classification without partial feedback. The first term $\mathcal{L}_{src}$ in Equation 2 corresponds to the loss used in direct method, defined over the factual data alone. Meanwhile, the second term refers to the loss defined over the imputed counterfactual data. In other words, in order to get a good model across all actions, we jointly optimize $\hat{r}, \theta$ which represent the true negative log likelihood when all counterfactuals are observed. Such objectives can be minimized by the EM algorithm via iteratively updating model and imputing pseudolabels (the pseudolabel is the minimum solution for $\hat{r}$ defined on the simplex (Zou et al. 2019)). To accomplish this, we first train an initial classifier $f_0(x, p)$ on observational data, then impute pseudolabels on all unseen actions from the observation data with $\hat{r}_{i,a} = \bar{f}_0(x_i, a)$, where the k-th entry is imputed by

$$
\bar{f}(r_{i,a,k}|x_i, a) = 1 \text{ if } k = \underset{c}{\arg\max} f(r_{i,a,c}|x_i, a) \text{ else } 0.
$$

However, it has been shown that such a procedure may suffer from local minima (Grandvalet and Bengio 2005) or over-confident wrong pseudolabels (Zou et al. 2019). Wei et al. (2020) shows when the underlying data distribution and pseudolabeler satisfies expansion assumption (See Definition 3.1 and Assumption 4.1, 3.3 in Wei et al. (2020)), self-training algorithms with input-consistency are able to achieve improvement from pseudolabeling (Theorem 4.3 in Wei et al. (2020)). Intuitively, the condition states that there needs to be many correct neighbors around the errors made by pseudolabeler so that correct labels can refine the decision boundary. Note that when imposing the same assumptions 4.1 and 3.3 on the underlying data distribution and error set of our pseudolabeler, we have the same improvement guarantee via self-training. We formalize this statement in Theorem 1, the formal definitions, assumptions and proof are included in Appendix (https://arxiv.org/abs/2112.04461) due to the space limit.

Theorem 1. *If the pseudolabeler has sufficiently good classification performance on all classes across actions, and the underlying data distribution satisfies expansion assumption with good separation (close samples have a high likelihood of having same labels), then minimizing a weighted sum of input consistency loss and pseudolabeling loss can improve pseudolabeler's classification performance.*

Motivated by the theoretical analysis that better input consistency can help further improve the pseudolabeling algorithm, we adapt Virtual Adversarial Training loss (VAT) (Miyato et al. 2018) in our CST algorithm, which we refer to as Counterfactual VAT (CVAT). A key difference between CVAT and VAT is that CVAT considers the joint data-action space (x, a) while VAT only regularizes on the input feature. Intuitively, VAT regularizes the output distribution to be isotropically smooth around each data point by selectively smoothing the model in its most anisotropic direction in an adversarial fashion. This direction is an adversarial perturbation z_i^{adv} of input data x_i. Similar to VAT, CVAT minimizes the distance between $f_{\hat{\theta}}(x_i, a)$ and $f_\theta(x_i + z_i^{\text{adv}}, a)$ over all counterfactual actions for x_i,

$$
\mathcal{L}_{\text{CVAT}} = \sum_i \sum_{a \in \mathcal{A} \setminus a_i} D\left[f_{\hat{\theta}}(x_i, a), f_\theta(x_i + z_i^{\text{adv}}, a)\right]
$$
$$
z_i^{\text{adv}} = \underset{z; \|z\|_2 \leq \epsilon}{\arg\max} \sum_{a \in \mathcal{A} \setminus a_i} D\left[f_{\hat{\theta}}(x_i, a), f_\theta(x_i + z, a)\right]
$$

where $D[\cdot, \cdot]$ is a divergence or distance measure between distributions and $\hat{\theta}$ stands for the current model parameters ($D(\cdot)$ denotes the actual function with substituted values). Like the original VAT, $D(z, x_i, a, \hat{\theta})$ takes the minimal value at $z = 0$ for all a, the differentiability assumption dictates that the first derivative $\nabla_z \sum_{a \in \mathcal{A} \setminus a_i} D(z, x_i, a, \hat{\theta})|_{z=0}$ is zero. Therefore, for each x_i, the loss term can be approximated by its second-order Taylor approximation (around $z = 0$) and z^{adv} can still be effectively approximated by the first dominant eigenvector of the Hessian matrix via power iteration.

The final CVAT regularized self-training objective can be

written as

$$\mathcal{L}_{\text{CST-CVAT}} = \mathcal{L}_{\text{CST}} + \lambda\mathcal{L}_{\text{CVAT}}. \tag{3}$$

We iteratively train the model and impute pseudolabels with a fixed number of iterations. The algorithm is stated in Algorithm 1. CST training with both loss $\mathcal{L}_{\text{CST-CVAT}}$ and $\mathcal{L}_{\text{CST}}$ is convergent, which we show in Proposition 1. The proof is included in the Appendix.

Proposition 1. *CST is convergent with a proper learning rate.*

Algorithm 1: Counterfactual Self-Training

1: Select loss function $\mathcal{L}_{\text{ST}} \leftarrow \mathcal{L}_{\text{CST}}$ in Equation 2 or $\mathcal{L}_{\text{ST}} \leftarrow \mathcal{L}_{\text{CST-CVAT}}$ in Equation 3.
2: Train a base learner f_0 on observational data.
3: **for** each iteration **do**
4:     **for** each $a \in \mathcal{A} \setminus a_i$ **do**
5:         Impute pseudolabel $\hat{r}_{i,a} = \bar{f}_0(x_i, a)$.
6:     **end for**
7:     **while** not converged **do**
8:         $\theta \leftarrow \text{Optim}(\nabla_\theta\mathcal{L}_{\text{ST}}, \theta)$ ▷ Self-training
9:         **for** each $i \in \{1 \dots N\}$ and $a \in \mathcal{A} \setminus a_i$ **do**
10:           $\hat{r}_{i,a} = \bar{f}_0(x_i, a)$.   ▷ Imputation
11:         **end for**
12:     **end while**
13:     $f_0 = f_\theta$
14: **end for**

Experiments

We begin with a toy example of binary actions to visualize and illustrate the benefit brought by CST. Next we construct synthetic datasets for a pricing example and utilize two real datasets to demonstrate the efficacy of our proposed algorithm. For the implementation, we use a three layer neural network with 128 nodes as our model with dropout of $p = 0.2$ and Leaky ReLU activation unless specified otherwise, and cross entropy loss as the loss function. We adapt the following backbone models which are originally designed for counterfactual regression tasks to counterfactual classification in our experiments to warm start CST, the implementation details are included in the Appendix:

- Direct Method (DM): This baseline directly trains a model on observational data.

- HSIC (Lopez et al. 2020): We use the last layer as an embedding and calculate HSIC between the embedding and the actions, HSIC is then used as regularization in addition to cross entropy loss.

- Uniform DM (UDM) (Wang et al. 2019): This method estimates the logging policy using historical data and uses importance sampling to simulate a randomize trial.

Once we finish training the backbone model, we run CST using pseudolabels generated by the backbone model according to Equation 2 (PL) or with additional CVAT loss defined in Equation 3 (PL + CVAT). The distance measure used in CVAT loss is chosen to be KL divergence. The stepsize in

finite-difference approximation, number of power iteration steps and perturbation size are set to 10, 3 and 1 respectively. The number of iteration of CST is set to 2. The hyperparameter of CVAT loss is chosen using a grid search of $[0.01, 0.1, 1, 10]$ on a validation set. All experiments are conducted using one NVidia GTX 1080-Ti GPU with five repetitions and Adam optimizer (Kingma and Ba 2014). Mean and standard error are reported for each metric. For synthetic and real data experiments, we choose one downstream task to examine how each method performs by predicting which action is most likely to cause a positive outcome ($r = 1$).

Toy Data We present a toy example to illustrate CST procedure with PL + CVAT and offer some intuition on how this method works[1]. In our examples, we assume there are two available offers/actions A0 and A1, and two types of customers P0 and P1. P0 reacts favorably to offer A0 (i.e., resulted in purchases) and has no reaction to A1. P1, on the other hand, has exactly the opposite reactions towards the offers. The underlying data distribution for P0 and P1 is a simple two moon distribution as shown in Figure 2 with shaded points. The training (factual) data points are generated through a biased logging policy which has a higher propensity $\exp(-(x_0 - min(x_0)))$ to assign action A1 for features with small x_0, which is the first dimension of feature $\mathcal{X}$. The direct method is chosen to be a three-layer MLP with 16 hidden nodes and dropout layer with $p = 0.5$.

Figure 2a, 2b show the data samples of 50 observed data points. The curve represents the decision boundary of the current model, initially trained with the direct method in Figure 2a-2d. It fails to converge to the optimal solution for action A1, despite achieving a good performance for A0 on the training data. The CST algorithm imputes the pseudolabels and then repeats its training in an iterative fashion. Here we use $\lambda = 1$ for CVAT loss to enforce input consistency. Figure 2e and 2f show the output after the first imputation, while Figure 2g and 2h depict the result after 10 iterations. We observe that CST with PL + CVAT converges to the correct solution for both actions and the margin between the two classes becomes much larger with training. Compared to the direct method, our counterfactual self-training algorithm better utilizes the underlying data structure and refines the decision boundary through extrapolation across actions.

Synthetic Datasets In synthetic experiments, we use a pricing example similar to the experiment in Lopez et al. (2020). Let $U(\cdot, \cdot)$ be a uniform distribution. Assume customer features are a 50-dimensional vector X drawn from $U(0, 1)^{50}$ and there are 5 price options from \$1 to \$5. The logging policy is set as $\pi(p = i|x) = \frac{x_i}{\sum_{i=1}^{10} x_i}$. Five types of demand functions are simulated, and the complete data generation process is detailed in the Appendix. For each action, there is a binary outcome: purchase (1) or not purchase (0).

We generate 1000 samples for each demand function and report the true negative log likelihood (NLL) to examine probability estimation of each method in Table 1. Hamming loss which relies on the hard labels generated by the algorithm is reported in Table 2. Since we want to fit a classi-

---

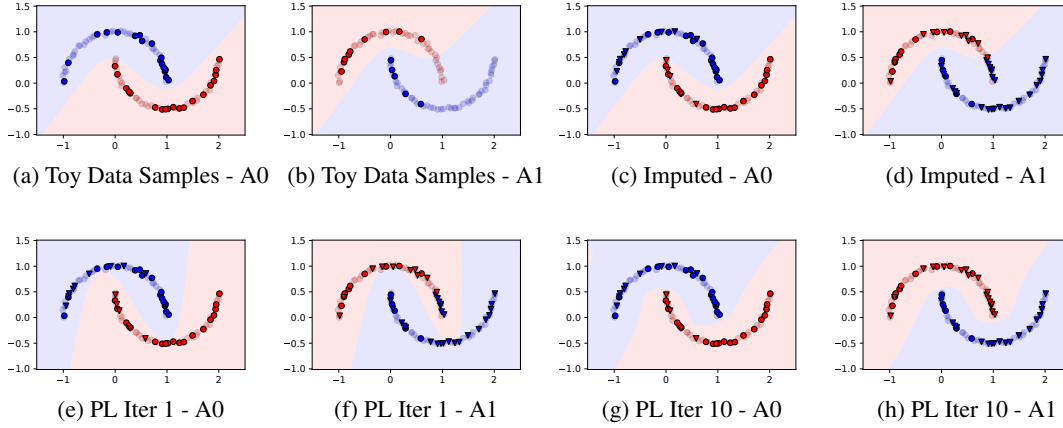[1]Code is available at https://github.com/ruijiang81/CST

Figure 2: CST with PL + CVAT on Toy Data. Points and Triangles represent factual data and imputed pseudolabels respectively. The underlying data distribution is shown using shaded points. The curve represents the decision boundary of the current model. (c), (d): Direct Method Solution; (g), (h): CST with PL + CVAT Solution.

Model	Method	D1	D2	D3	D4	D5
	Backbone	2.3738±0.0747	2.9439±0.2395	3.2444±0.0856	3.3443±0.1210	2.4220±0.1275
DM	PL	0.5744±0.0165	0.6997±0.0317	0.7263±0.0205	0.8043±0.0348	0.6425±0.0253
	PL + CVAT	**0.5422**±0.0080	**0.5863**±0.0145	**0.6825**±0.0144	**0.6807**±0.0132	**0.5754**±0.0211
	Backbone	1.4748±0.0183	1.6062±0.0645	1.8629±0.0190	1.9334±0.0564	1.5240±0.0601
HSIC	PL	0.6028±0.0199	0.6840±0.0361	0.7177±0.0199	0.7423±0.0262	0.6220±0.0197
	PL + CVAT	**0.5418**±0.0147	**0.5759**±0.0074	**0.6553**±0.0086	**0.6593**±0.0108	**0.5591**±0.0135
	Backbone	2.5589±0.2602	2.7615±0.6944	1.6277±0.4775	3.0926±0.5732	2.5924±0.6082
UDM	PL	0.5841±0.0161	0.6315±0.0229	0.7090±0.0195	0.7258±0.0118	0.5936±0.0137
	PL + CVAT	**0.5553**±0.0238	**0.5891**±0.0221	**0.6502**±0.0087	**0.6446**±0.0039	**0.5616**±0.0145

Table 1: Log Loss on Synthetic Dataset.

Model	Method	D1	D2	D3	D4	D5
	Backbone	0.2922±0.0100	0.2775±0.0030	0.3588±0.0046	0.3560±0.0056	0.2816±0.0090
DM	PL	**0.2561**±0.0057	0.2564±0.0046	0.3363±0.0066	0.3244±0.0039	0.2545±0.0060
	PL + CVAT	0.2566±0.0039	**0.2544**±0.0029	**0.3299**±0.0050	**0.3219**±0.0037	**0.2494**±0.0047
	Backbone	0.2804±0.0093	0.2767±0.0040	0.3573±0.0054	0.3551±0.0083	0.2785±0.0076
HSIC	PL	**0.2623**±0.0097	0.2554±0.0030	0.3328±0.0059	0.3308±0.0097	0.2554±0.0053
	PL + CVAT	0.2673±0.0082	**0.2553**±0.0030	**0.3285**±0.0045	**0.3214**±0.0040	**0.2544**±0.0062
	Backbone	0.2718±0.0077	0.2710±0.0023	0.3645±0.0043	0.3639±0.0085	0.2799±0.0055
UDM	PL	**0.2602**±0.0086	0.2603±0.0029	**0.3390**±0.0075	0.3380±0.0090	0.2620±0.0044
	PL + CVAT	0.2608±0.0082	**0.2580**±0.0027	0.3496±0.0144	**0.3274**±0.0019	**0.2564**±0.0029

Table 2: Hamming Loss on Synthetic Dataset.

fier that may work for many possible downstream tasks, the main metric we focus is NLL given current classifiers. For example, in revenue management, estimating demand accurately is critical for price optimization, inventory control and promotion planning. More specifically, NLL is defined as

$$\sum_{i=1}^{N} \Big(\sum_{k=1}^{m} -r_{i,a_i,k} \log f_\theta(r_{i,a_i,k}|x_i, a_i) + \quad (4)$$
$$\sum_{a \in \mathcal{A} \backslash a_i} \sum_{k=1}^{m} -r_{i,a,k} \log f_\theta(r_{i,a,k}|x_i, a_i) \Big)/N|\mathcal{A}|$$

and hamming loss is defined as $\sum_{i=1}^{N} \sum_{a \in \mathcal{A}} \mathbb{I}(y_{i,a} = \hat{y}_{i,a})/N|\mathcal{A}|$.

With respect to different backbone models, we find UDM and HSIC in general improve over Direct Method with a better NLL, with HSIC being the best model. In all experiments, we find PL improves NLL from backbone model and incorporating CVAT loss further improves the performance, highlighting the value of our proposed CST algorithm. Similarly, pseudolabeling and CVAT loss can provide improvement for prediction accuracy measured by hamming loss. In some experiments, PL achieves the best hamming loss, this discrepancy between NLL and hamming loss may be

Model + Metric	Method	$o = 1$	$o = 2$	$o = 3$
	Backbone	0.2886±0.0065	0.2850±0.0085	0.2901±0.0093
DM + Hamming Loss	PL	0.2611±0.0045	**0.2645**±0.0097	**0.2606**±0.0048
	PL + CVAT	**0.2605**±0.0059	0.2677±0.0145	0.2696±0.0092
	Backbone	2.4871±0.1057	2.4993±0.1100	2.3416±0.0093
DM + Log Loss	PL	0.6200±0.0316	0.6053±0.0237	0.6132±0.0213
	PL + CVAT	**0.5643**±0.0118	**0.5672**±0.0170	**0.5664**±0.0126

Table 3: Experiments with Varying Overlap

caused by over-confident wrong predictions made by pseudolabels (Zou et al. 2019) and including CVAT loss provides a viable way to smooth output predictions to avoid over-confident predictions.

Model	Method	Scene	Yeast
	Backbone	3.2289±0.1678	5.5001±0.1218
DM	PL	0.3357±0.0132	0.6956±0.0318
	PL + CVAT	**0.3272**±0.0124	**0.5234**±0.0053
	Backbone	1.0385±0.0320	1.9315±0.0681
HSIC	PL	0.3667±0.0061	0.7372±0.0980
	PL + CVAT	**0.3503**±0.0056	**0.5199**±0.0048
	Backbone	6.2181±0.4192	7.5073±0.9486
UDM	PL	0.5526±0.0160	0.7785±0.0924
	PL + CVAT	**0.4373**±0.0152	**0.5441**±0.0119

Table 4: Log Loss on Real Dataset.

Model	Method	Scene	Yeast
	Backbone	0.1301±0.0036	0.2853±0.0080
DM	PL	0.1304±0.0034	0.2539±0.0066
	PL + CVAT	**0.1258**±0.0024	**0.2434**±0.0054
	Backbone	0.1821±0.0031	0.2821±0.0073
HSIC	PL	0.1546±0.0041	0.2471±0.0052
	PL + CVAT	**0.1473**±0.0033	**0.2366**±0.0043
	Backbone	0.2542±0.0079	0.3252±0.0040
UDM	PL	0.2274±0.0129	0.2910±0.0100
	PL + CVAT	**0.1852**±0.0081	**0.2868**±0.0119

Table 5: Hamming Loss on Real Dataset.

Multi-Label Datasets We use two multi-label datasets from LIBSVM repository (Elisseeff and Weston 2002; Boutell et al. 2004; Chang and Lin 2011), which are used for semantic scene and text classification. We convert the supervised learning datasets to bandit feedback by creating a logging policy using 5% of the data following Swaminathan and Joachims (2015a); Lopez et al. (2020). More specifically, each feature x has a label $y \in \{0,1\}^p$ where p is the number of labels. After the logging policy selects a label (action) i, a reward y_i is revealed as bandit feedback (x, i, y_i), *i.e.*, for each data point, if the policy selects one of the correct labels of that data point, it gets a reward of 1 and 0 otherwise. By doing so, we have the full knowledge of counterfactual outcomes for evaluation. Data statistics are summarized in the Appendix. True negative log loss and hamming loss are reported in Table 4 and 5 respectively.

On both datasets, we observe that pseudolabeling and CVAT loss improve backbone model performance in terms of hamming loss and negative log loss and CVAT loss brings additional improvement. The improvement of pseudolabeling and CVAT loss is also dependent on backbone model performance. In particular, we observe a better base model leads to better CST performance.

Varying Overlap In this section, we vary the level of overlap to control the randomness of observational data. We use the synthetic data setup using D1 and set logging policy as $\pi(p = i|x) = \frac{e^{o*x_i}}{\sum_{i=1}^{10} e^{o*x_i}}$. The results are summarized in Table 3. We find both DM and our proposed method is robust to different level of randomness in observational data. Similar conclusion is also observed in Sachdeva, Su, and Joachims (2020).

Conclusion and Future Work

In this paper, we propose a novel counterfactual self-training algorithm for counterfactual classification problem. CST relies on pseudolabeling to estimate the true log likelihood on observational data and works in an iterative fashion. With expansion and separation assumptions, CST experiences the same theoretical guarantees for performance improvement as shown in existing pseudolabeling literature. Our results hold promises for many counterfactual classification models that may work beyond neural network models due to the model agnostic nature of pseudolabeling. However, our CST framework has several limitations. First, CST requires finite *categorical* action set, which is also a crucial requirement for CRM methods. In order to augment observation data, CST will augment every action not observed. For continuous action, discretization or joint kernel embedding might be used as an extension to CST, which we leave for future work. Second, while our experimental results show encouraging results that CST can improve even state-of-the-art models, the expansion and base-model performance assumption required in current theoretical understanding generally cannot be verified, which means CST may deteriorate model performance when the underlying data is noisy or base model has a high error rate. In practice, practitioners can use model selection methods such as counterfactual cross validation (Saito and Yasui 2020) to avoid this, and we argue CST still provides a promising means for enhancing counterfactual classification performance.

Acknowledgements

References

Alaa, A. M.; and van der Schaar, M. 2017. Bayesian inference of individualized treatment effects using multi-task gaussian processes. *arXiv preprint arXiv:1704.02801*.

Amini, M.-R.; and Gallinari, P. 2002. Semi-supervised logistic regression. In *ECAI*, volume 2, 11.

Bertsimas, D.; and Kallus, N. 2016. The power and limits of predictive approaches to observational-data-driven optimization. *arXiv preprint arXiv:1605.02347*.

Biggs, M.; Gao, R.; and Sun, W. 2021. Loss Functions for Discrete Contextual Pricing with Observational Data. *arXiv preprint arXiv:2111.09933*.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern recognition*, 37(9): 1757–1771.

Chang, C.-C.; and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3): 1–27.

Dudík, M.; Erhan, D.; Langford, J.; Li, L.; et al. 2014. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4): 485–511.

Elisseeff, A.; and Weston, J. 2002. A kernel method for multi-labelled classification. In *Advances in neural information processing systems*, 681–687.

Forney, A.; and Bareinboim, E. 2019. Counterfactual Randomization: Rescuing Experimental Studies from Obscured Confounding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2454–2461.

Gao, R.; Saar-Tschansky, M.; De-Arteaga, M.; Han, L.; Lee, M. K.; and Lease, M. 2021. Human-AI Collaboration with Bandit Feedback. *arXiv preprint arXiv:2105.10614*.

Grandvalet, Y.; and Bengio, Y. 2005. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, 529–536.

Greensmith, E.; Bartlett, P. L.; and Baxter, J. 2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov): 1471–1530.

Gretton, A.; Fukumizu, K.; Teo, C. H.; Song, L.; Schölkopf, B.; and Smola, A. J. 2008. A kernel statistical test of independence. In *Advances in neural information processing systems*, 585–592.

Han, L.; Zou, Y.; Gao, R.; Wang, L.; and Metaxas, D. 2019. Unsupervised domain adaptation via calibrating uncertainties. In *CVPR Workshops*, volume 9.

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

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

Joachims, T.; Swaminathan, A.; and de Rijke, M. 2018. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*.

Johansson, F.; Shalit, U.; and Sontag, D. 2016. Learning representations for counterfactual inference. In *International conference on machine learning*, 3020–3029.

Kallus, N. 2019. Classifying treatment responders under causal effect monotonicity. *arXiv preprint arXiv:1902.05482*.

Kallus, N.; and Zhou, A. 2018. Confounding-robust policy improvement. In *Advances in neural information processing systems*, 9269–9279.

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

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kök, A. G.; and Fisher, M. L. 2007. Demand estimation and assortment optimization under substitution: Methodology and application. *Operations Research*, 55(6): 1001–1021.

Künzel, S. R.; Sekhon, J. S.; Bickel, P. J.; and Yu, B. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10): 4156–4165.

Lefortier, D.; Swaminathan, A.; Gu, X.; Joachims, T.; and de Rijke, M. 2016. Large-scale validation of counterfactual learning methods: A test-bed. *arXiv preprint arXiv:1612.00367*.

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

Li, Y.; Guan, C.; Li, H.; and Chin, Z. 2008. A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system. *Pattern Recognition Letters*, 29(9): 1285–1294.

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

Maurer, A.; and Pontil, M. 2009. Empirical Bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*.

McGill, J. I.; and Van Ryzin, G. J. 1999. Revenue management: Research overview and prospects. *Transportation science*, 33(2): 233–256.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993.

Nigam, K.; McCallum, A. K.; Thrun, S.; and Mitchell, T. 2000. Text classification from labeled and unlabeled documents using EM. *Machine learning*, 39(2-3): 103–134.

Pearl, J. 2017. Detecting latent heterogeneity. *Sociological Methods & Research*, 46(3): 370–389.

Pearl, J.; et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*.

Rosenbaum, P. R. 1987. Model-based direct adjustment. *Journal of the American Statistical Association*, 82(398): 387–394.

Rosenbaum, P. R.; and Rubin, D. B. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1): 41–55.

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

Sachdeva, N.; Su, Y.; and Joachims, T. 2020. Off-policy bandits with deficient support. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 965–975.

Saito, Y.; and Yasui, S. 2020. Counterfactual cross-validation: Stable model selection procedure for causal inference models. In *International Conference on Machine Learning*, 8398–8407. PMLR.

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

Shi, C.; Blei, D. M.; and Veitch, V. 2019. Adapting neural networks for the estimation of treatment effects. *arXiv preprint arXiv:1906.02120*.

Shpitser, I.; and Pearl, J. 2012. Identification of conditional interventional distributions. *arXiv preprint arXiv:1206.6876*.

Swaminathan, A.; and Joachims, T. 2015a. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, 814–823.

Swaminathan, A.; and Joachims, T. 2015b. The self-normalized estimator for counterfactual learning. In *advances in neural information processing systems*, 3231–3239.

Tanha, J.; van Someren, M.; and Afsarmanesh, H. 2017. Semi-supervised self-training for decision tree classifiers. *International Journal of Machine Learning and Cybernetics*, 8(1): 355–370.

Wales, T. J.; and Woodland, A. D. 1983. Estimation of consumer demand systems with binding non-negativity constraints. *Journal of Econometrics*, 21(3): 263–285.

Wang, L.; Bai, Y.; Bhalla, A.; and Joachims, T. 2019. Batch Learning from Bandit Feedback through Bias Corrected Reward Imputation.

Wei, C.; Shen, K.; Chen, Y.; and Ma, T. 2020. Theoretical analysis of self-training with deep networks on unlabeled data. *arXiv preprint arXiv:2010.03622*.

Xu, Y.; Xu, Y.; and Saria, S. 2016. A Bayesian nonparametric approach for estimating individualized treatment-response curves. In *Machine Learning for Healthcare Conference*, 282–300.

Yin, M.; Shi, C.; Wang, Y.; and Blei, D. M. 2021. Conformal Sensitivity Analysis for Individual Treatment Effects. *arXiv preprint arXiv:2112.03493*.

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

Zou, Y.; Yu, Z.; Kumar, B.; and Wang, J. 2018. Domain adaptation for semantic segmentation via class-balanced self-training. *arXiv preprint arXiv:1810.07911*.

Zou, Y.; Yu, Z.; Liu, X.; Kumar, B.; and Wang, J. 2019. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, 5982–5991.