Is Unsupervised Performance Estimation Impossible When Both Covariates and Labels shift?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Accurately estimating and explaining an ML model's performance on new datasets is increasingly critical in reliable ML model deployment. With no labels on the new datasets, performance estimation paradigms often assume either covariate shift or label shift, and thus lead to poor estimation accuracy when the assumptions are broken. *Is unsupervised performance monitoring really impossible when both covariates and labels shift?* In this paper, we give a negative answer. To do so, we introduce Sparse Joint Shift (SJS), a new distribution shift model considering the shift of labels and a few features. We characterize the mathematical conditions under which SJS is identifiable. This shows that unsupervised performance monitoring is indeed feasible when a few features and labels shift. In addition, we propose SEES, an algorithmic framework for performance estimation under SJS. Preliminary experiments show the superior estimation performance of SEES over existing paradigms. This opens the door to tackling joint shift of both covaraites and labels without observing new datasets' labels.

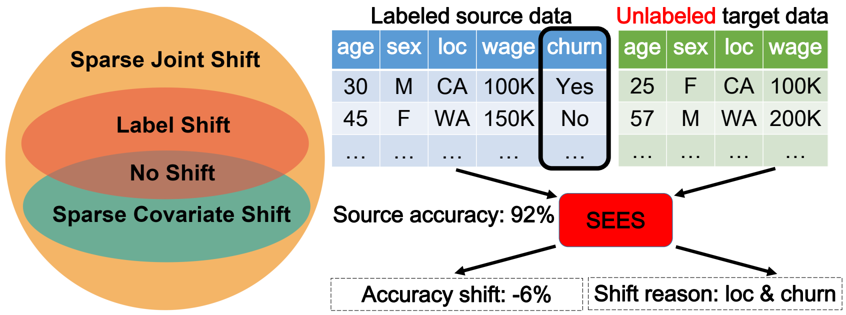
1 Introduction

Encountering new data different from training data is increasingly common during machine learning (ML) deployments. For example, geographical locations [4], demographic features [2], and label balance [3] are observed to shift between model development and deployment and thus affect the model performance. For safe ML applications, it is an important step to estimate and explain how a model's performance changes.

Estimating and explaining performance shift is challenging for several reasons, however. One major challenge is that the data distribution might shift in flexible ways. Another obstacle is that we often do not have labels on the new data, especially in ML monitoring applications. Without any assumption on the distribution shift, it's impossible to estimate how well the model would perform on the unlabeled new data. Previous work often assumes (i) label shift [5], where feature distributions conditional on the labels are fixed, or (ii) covariate shift [10], where label distributions conditional on features stay the same. However, we often do not know whether the real data shift is limited to label or covariate shift, and naively applying estimation methods designed for one shift may produce inaccurate assessments [7]. Moreover, labels and features may shift simultaneously in practice, invalidating these common assumptions. Thus, we ask: *Is unsupervised performance estimation really impossible when both covariates and labels shifts?*

Our contributions: In this paper, we give a negative answer by proposing a new distribution shift model, Sparse Joint Shift (SJS), to consider the joint shift of both labels and a few features. SJS assumes labels and a few features shift, but the remaining features' distribution conditional on the shifted features and labels is fixed. This unifies and generalizes sparse covariate shift and label shift:

Submitted to Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Sparse joint shift (SJS) (b) Shift estimation and explanation under SJS

Figure 1: Overview of sparse joint shift (SJS). (a) Both label shift and sparse covariate shift are SJS, but SJS contains additional shifts as well. (b) illustrates SEES, a framework for performance shift estimation and explanation under SJS. Given labeled source and unlabeled target data, SEES exploits the joint shift modeled by SJS to estimate the model performance change and explain which factors drive the shift. In this example, the goal is to predict *churn*.

both of them are SJS, but some SJS is not label or sparse covariate shift (Figure 1). Then we provide mathematical conditions under which SJS is provably identifiable: if the non-shifted features are weakly correlated, then the marginal feature distribution uniquely determines the joint distribution under SJS. This makes it possible to quantify the shift and estimate model performance on new data without any labels. This makes it possible to quantify the shift and estimate model performance on new data without any labels. Furthermore, we propose SEES, an algorithmic paradigm for performance shift estimation and explanation under SJS. Preliminary experiments show that SEES significantly reduces the performance estimation error compared to existing methods.

2 Problem Statement: Unsupervised Performance Estimation

We start by defining unsupervised performance estimation. Suppose we are given a labeled dataset $D_s \triangleq \{(\boldsymbol{x}^{s,i}, y^{s,i})\}_{i=1}^{n_s}$ from some source distribution $\mathbb{P}_s$, an unlabeled dataset $D_t \triangleq \{(\boldsymbol{x}^{t,i})\}_{i=1}^{n_t}$ from some target distribution $\mathbb{P}_t$, and an ML model $f(\cdot)$ predicting the associated label $\in [L]$ given any feature vector $\boldsymbol{x} \in \mathbb{R}^d$. Our goal is to estimate the performance on the target domain. Let $\ell(\cdot,\cdot)$ denote some performance metric (e.g., the 0-1 loss). Then formally we aim at estimating $\Delta \triangleq \mathbb{E}_{(\boldsymbol{x},y)\sim\mathbb{P}_t}[\ell(f(\boldsymbol{x}), y)]$. This is challenging as we do not observe labels on the target domain.

3 SJS: A Tractable Unification of Label Shift and Sparse Covariate Shift

Without labels on the target domains, the joint distribution of target labels and features is not identifiable, rendering unsupervised performance estimation arbitrarily unreliable in the worse case. To mitigate nonidentifiability, it's necessary to make additional assumptions. The most popular assumptions in literature are label shift [5] and covariate shift [9]. Label shift assumes that only label distribution may change, but the feature distribution given a label remains, i.e., $p_s(\boldsymbol{x}|y) = p_t(\boldsymbol{x}|y)$. On the other hand, covariate shift assumes that feature distribution can shift, but the label distribution given the features is fixed, i.e., $p_s(y|\boldsymbol{x}) = p_t(y|\boldsymbol{x})$. However, those assumptions disallow simultaneous changes of both features and labels, which often happen in real-world data [4, 8, 11]. To enable joint feature and label estimation which is tractable, we introduce a subclass of joint distribution shift, *Sparse Joint Shift* (SJS), as follows.

Definition 1 (Sparse Joint Shift (SJS))**.** *Suppose for an integer $m \le d$ and an index set $\mathcal{I} \subset [d]$ with size at most m (i.e., $|I| \le m$), $p_s(\boldsymbol{x}_{I^c}|\boldsymbol{x}_I, y) = p_t(\boldsymbol{x}_{I^c}|\boldsymbol{x}_I, y)$. Then we say the source and target pair (p_s, p_t) is under m-Sparse Joint Shift, or m-*SJS*. Here, $I^c \triangleq [d] - I$. We call I the shift index set.*

Roughly speaking, SJS allows both labels and a few features to shift, but assumes the remaining features' conditional distribution to stay the same. Next, we will study when this assumption allows tractable performance shift estimation.

3.1 When is sparse joint shift identifiable?

Note that when $m = d$, m-SJS simply becomes general joint distribution shift, which is unidentifiable. Thus, it is worthy understanding when m-SJS resolves the identifiability issue. To do so, let us first formally introduce the notation of identifiability.

Definition 2 (Identifiable)**.** *Suppose the source-target tuple (p_s, p_t) is under m-SJS. (p_s, p_t) is identifiable if and only if for any alternative distribution $p_a(\boldsymbol{x}, y)$, if $p_a(\boldsymbol{x}) = p_t(\boldsymbol{x})$ and $\exists \mathcal{J} \subset [d], |\mathcal{J}| \leq m$, such that $p_a(\boldsymbol{x}_{\mathcal{J}^c}|\boldsymbol{x}_{\mathcal{J}}, y) = p_s(\boldsymbol{x}_{\mathcal{J}^c}|\boldsymbol{x}_{\mathcal{J}}, y)$, then $p_a(\boldsymbol{x}, y) = p_t(\boldsymbol{x}, y)$.*

The following statement shows when (p_s, p_t) is identifiable.

Theorem 1. *Suppose (p_s, p_t) is under m-SJS. Assume for any set $\mathcal{J} \subset [d], |\mathcal{J}| \leq m$ and any fixed $\bar{\boldsymbol{x}} \in \mathcal{X}$, the probability density (or mass) functions $\{p_s(\boldsymbol{x}_{J^c \cap I^c}, \boldsymbol{x}_{\mathcal{J} \cup I} = \bar{\boldsymbol{x}}_{\mathcal{J} \cup I}, y = i)\}_{i=1}^{L}$ are linearly independent. Then (p_s, p_t) is identifiable.*

This statement sheds light on why uniquely identifying the target distribution without target label is feasible under sparse joint shift. Roughly speaking, m-SJS requires that given the shifted features and labels, the remaining features' distribution remains the same on both domains. If those remaining features are different enough (linear independence), they can uniquely determine the distribution of the shifted features and labels. We stress that the linear independence is necessary: if it does not hold, then for any m, we can always find some source-target pair (p_s, p_t) which is not identifiable. Linear independence implicitly requires sparsity: if $m > d/2$, then $J^c \cap I^c$ can be empty and the linear independence does not hold. In other words, the sparsity is necessary for the shift to be identifiable.

3.2 How does SJS relate to label shift and covariate shift?

A natural question is how does SJS relates to standard label shift and covariate shift. To answer this, let us first introduce label and sparse covariate shift formally.

Definition 3. *The source and target (p_s, p_t) is under Label Shift iff $p_s(\boldsymbol{x}|y) = p_t(\boldsymbol{x}|y)$, and under m-Sparse Covariate Shift iff $p_s(\boldsymbol{x}_{I^c}, y|\boldsymbol{x}_I) = p_t(\boldsymbol{x}_{I^c}, y|\boldsymbol{x}_I)$ for some index set I with size $m < d$.*

Now we are ready to answer the above question.

Theorem 2. *If (p_s, p_t) is under label shift, then it is also under* 0-SJS. *If (p_s, p_t) is under m-sparse covariate shift, then it is also under* m-SJS. *In addition, there exists (p_s, p_t) under* m-SJS *such that it is under neither label shift or covariate shift.*

There are several takeaways. First, label shift implies SJS without additional requirements. In fact, as certain distribution pairs are under SJS but not label shift, SJS is strictly more general than label shift. Second, SJS also includes sparse covariate shift. When $m = d$, SJS completely unifies both label shift and covariate shift, though it is not identifiable. Identifiable SJS, on the other hand, unifies label shift and sparse covariate shift. Finally, SJS also allows shifts not covered by label shift and covaraite shift: the correlation between label and (a set of) features can be shifted.

3.3 How to estimate an ML model's performance under identifiable SJS?

Now we are ready to present SEES (sparsity-aware performance estimation), an algorithmic framework for performance estimation under SJS. It consists of two steps. First, it learns an importance weight function $\hat{w}(\boldsymbol{x}, y)$ to approximate the density ratio $w(\boldsymbol{x}, y) \triangleq p_t(\boldsymbol{x}, y)/p_s(\boldsymbol{x}, y)$. Next, the performance is estimated by reweighting the accuracy on the source domain by the importance weights, i.e., $\frac{1}{n_s}\sum_{i=1}^{n_s} \hat{w}(\boldsymbol{x}^{s,i}, y^{s,i})\ell(\boldsymbol{x}^{s,i}, y^{s,i})$. Note that if $\hat{w}(\boldsymbol{x}, y)$ matches the true importance weight $w(\boldsymbol{x}, y)$ exactly, the proposed estimation is an unbiased estimation of the true performance. The estimated $\hat{w}(\boldsymbol{x}, y)$ is the solution to the following sparsity-aware optimization framework

$$\begin{aligned} &\min_{w(\boldsymbol{x},y) \in \mathcal{W}} D(p_t(\boldsymbol{x}), \hat{p}_t(\boldsymbol{x})) \\ &\quad \text{s.t. } \hat{p}_t(\boldsymbol{x}) = \sum_{y=1}^{L} w(\boldsymbol{x}, y) \cdot p_s(\boldsymbol{x}, y), \text{ and } w(\boldsymbol{x}, y) \text{ depends on at most } m \text{ features of } \boldsymbol{x}. \end{aligned} \tag{3.1}$$

Here, $D(\cdot, \cdot)$ is some distance metric that measures the difference between two density functions. We minimize the distance between the induced feature density $\hat{p}_t(\boldsymbol{x})$ and the target feature density $p_t(\boldsymbol{x})$.

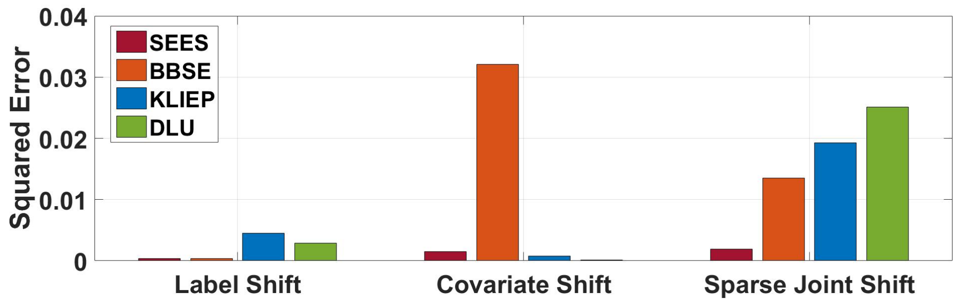


Figure 2: Squared ℓ_2 estimation error of various methods on the COVID-19 dataset under different data shifts. Overall, SEES is the only method that consistently produces accurate estimation across all shifts and significantly improves estimation performance over existing methods under SJS.

The minimization is not over joint label and feature distributions since target labels are not available. The induced feature density function can be easily derived from source density function and the weight function, encoded in the first constraint. m-SJS is enforced by the second constraint: m-SJS means given m features and labels, the distributions of remaining features are fixed across source and the induced domain, which holds if and only if their density ratio $w(\boldsymbol{x}, y)$ only depends on those m features. $\mathcal{W}$ represents the set of all feasible weight functions. Different parameterization can be easily realized by adopting different $\mathcal{W}$. Assume access to density functions $p_s(\boldsymbol{x}, y)$ and $p_t(\boldsymbol{x})$, and a weight function set $\mathcal{W}$ containing the true weight $w^*(\boldsymbol{x}, y) \triangleq \frac{p_t(\boldsymbol{x},y)}{p_s(\boldsymbol{x},y)}$. One can easily show the above optimization returns the true weight function $w^*(\boldsymbol{x}, y)$ for identifiable m-SJS. In practice, one can replace $p_s(\boldsymbol{x}, y)$ and $p_t(\boldsymbol{x})$ with their empirical estimation, and use standard distance metrics (such as KL-divergence or ℓ_2 norm) to instantiate $D(\cdot, \cdot)$.

4 Preliminary Experiments

In this section, we provide preliminary experiments to study the performance of SEES.Our goal is to (i) justify whether SEES estimates model performance accurately when SJS occurs, and (ii) understand how robust the performance of SEES is given various performance shifts.

ML models, Datasets and baselines. We use a gradient boosting tree model as the ML model, and focus on a case study on the COVID-19 dataset [1]. This dataset contains demographic features (such as age and gender) and symptom features (for example, fever, cough, and sore throat) of patients collected by the Israel government. The goal is to predict if a patient test positive or negative for COVID-19. We then evaluate performance of SEES when label shift, covariate shift (by varying the feature age), and sparse joint shift (by varying both label and feature age) occur. Compared baselines includes BBSE [5] for label shift, KLIEP [10], and DLU [6] for covariate shift.

Analysis. As shown in Figure 2, estimation error achieved by SEES is significantly smaller than all compared baselines when both feature age and label shift (i.e., the sparse joint shift). In addition, SEES is the only approach robust to different shifts. In fact, when labels shift, KLIEP and DLU lead to large estimation errors. When covariates shift, a poor estimation performance is induced by BBSE. This is because all existing baselines require that either labels or covariates shift. On the other hand, SEES is able to produce reliable performance estimation under different data shift models.

5 Conclusion

In this paper, we propose Sparse Joint Shift (SJS), a new distribution shift model that considers both label and covariate shifts. We show how SJS unifies and generalizes existing distribution shift models and remains identifiable under reasonable assumptions. We develop SEES, an algorithmic framework for unsupervised model performance estimation under SJS. Many problems remain open. A natural next step is how to improve estimation performance under SJS when a small number of target labels can be queried. Developing ML models robust to different SJS is also an open question.

References

[1] The COVID-19 government dataset. `https://data.gov.il/dataset/covid-19`. [Accessed 2022].

[2] Patrick Grother, Mei Ngan, and Kayee Hanaoka. Face recognition vendor test part 3: Demographic effects, 2019-12-19 2019.

[3] Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Comput. Surv.*, 52(4):79:1–79:36, 2019.

[4] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[5] Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pages 3122–3130. PMLR, 2018.

[6] Sangdon Park, Osbert Bastani, James Weimer, and Insup Lee. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 3219–3229. PMLR, 2020.

[7] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32, 2019.

[8] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

[9] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

[10] Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. *Advances in neural information processing systems*, 20, 2007.

[11] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020.