Detecting Distribution Shifts in FashionMNIST

Department of Artificial Intelligence
2277006 Solbin Kim

# GitHub repository link : https://github.com/2277006/Detecting-Distribution-Shifts

1. Introduction

One of the most common challenges faced when deploying machine learning models in real-world environments is distribution shift. Typically, engineers train models using carefully curated datasets with high-quality, well-distributed samples to maximize accuracy. However, in actual deployment scenarios, the input data may significantly differ from the training data. In applications like image classification for e-commerce platforms, for example, user-uploaded product images may have lower quality or resolution, and the distribution of product categories may vary substantially compared to the training phase.

To simulate such real-world conditions, this study utilizes the FashionMNIST dataset to explore two major types of distribution shifts. The first is a covariate shift, implemented by degrading image quality through resolution reduction and brightness enhancement. The second is a label shift, where the frequency of a specific class (sneakers) is reduced by 90%, realistically modeling changes in user behavior. Through statistical techniques, a domain classifier, and an evaluation of a classification model (ResNet-18), this research aims to analyze the distribution shift and interpret its impact on model performance.

2. Dataset Description

In this study, we utilized the FashionMNIST dataset, which is widely used in the field of image classification. FashionMNIST consists of 70,000 grayscale images, each sized at 28×28 pixels. The dataset is evenly divided into 10 fashion item categories: T-shirt/top, trousers, pullover, dress, coat, sandal, shirt, sneaker, bag, and ankle boot. With approximately 7,000 samples per class, the dataset is considered well-balanced and suitable for classification tasks.

For the purpose of this study, we designed a realistic scenario: A machine learning engineer develops an image classification model for use in an online shopping mall using a clean and balanced set of 60,000 training images. To simulate deployment conditions, the training set was randomly split into 70% (Domain A) and 30% (Domain B). Domain A represents the training data used during model development, while Domain B is treated as incoming data from actual users. To reflect real-world usage—such as users uploading product photos via mobile devices—Domain B was transformed to introduce distribution shifts. This setup allows us to more realistically analyze how distribution shifts can degrade model performance in actual deployment environments.

3. Method for Dataset Division and Transformation

3.1 Basic Dataset Splitting Method

In this study, the 60,000 training images from the FashionMNIST dataset were divided into two domains: Domain A, representing the training environment used by engineers, and Domain B, simulating the deployment environment with real user input data. To maintain randomness while ensuring reproducibility, the random seed was fixed at 42 during the split. The dataset was divided into 70% (Domain A, 42,000 images) and 30% (Domain B, 18,000 images). Immediately after splitting, the distributions of the two domains were verified to be statistically similar, establishing a reliable baseline for further comparison.

3.2 Realistic Data Transformation Method (Domain B_shift Creation)

In this study, we constructed **Domain B_shift** by applying realistic data transformations to Domain B, simulating common distribution changes that may occur in real-world service environments. The transformations applied to Domain B_shift are primarily divided into two categories

(1) Covariate Shift

To simulate the potential image quality degradation that may occur when real users upload product images through an online shopping website or mobile app, we intentionally downsampled the original 28×28 pixel images to 7×7 pixels and then upscaled them back to 28×28 pixels. While images captured by modern smartphones are typically of much higher quality and resolution, this transformation—though not a perfect replica of real-world conditions—serves as a controlled method to simulate a loss of detail. It effectively enables us to analyze how such a degradation in input quality could impact model performance. Additionally, to reflect user-uploaded images taken under bright lighting conditions, we uniformly increased the brightness of all pixel values by 0.1.

(2) Label Shift

In real-world online shopping platforms, it is common for the frequency of specific product categories to fluctuate due to seasonal demand or trends. To reflect this, we introduced an intentional class imbalance in the dataset by reducing the number of samples from the sneaker class (class 7) by 90%. Specifically, only 10% of the original sneaker images were retained, chosen randomly. This simulates a realistic scenario where user interest or availability of a certain product drops significantly, allowing us to examine the model's ability to generalize under such label distribution shifts.

4. Methods for Detecting Data Distribution Shifts

In this study, we selected three key statistical methods to effectively detect data distribution shifts. These methods allow us to quantitatively assess how much the data distribution has changed from various perspectives, providing objective and reliable insights into the presence and severity of distribution shifts.

4.1 KS-test

The KS-test is a non-parametric statistical test used to evaluate whether two datasets follow the same distribution. In this study, the KS-test was selected for the following reasons: it measures the maximum difference between the CDFs of two datasets, making it highly sensitive to even subtle distribution differences. Moreover, because it does not require any assumptions about the underlying distribution, it is broadly applicable to real-world data scenarios such as online shopping platforms. In this study, we applied the KS-test to compare the distributions of average pixel intensities across images, allowing us to effectively quantify covariate shifts caused by image quality degradation.

4.2 KL Divergence

KL Divergence is a quantitative measure of the difference between two probability distributions. From an information-theoretic perspective, it expresses how much one distribution diverges from another. In this study, KL Divergence was chosen to numerically characterize the distributional shift between two datasets, enabling a clearer understanding of the magnitude and severity of the change. By calculating the KL Divergence between the histograms of mean pixel values from the two datasets, we were able to quantify the strength of the shift and use it as a basis for predicting potential performance degradation.

4.3 Domain Classifier

A domain classifier is a machine learning model used to evaluate how clearly two domains can be distinguished based on their data distributions. In this study, we used a logistic regression model as the domain classifier to measure the extent to which the two domains could be separated. The classifier's performance is evaluated using the Area Under the ROC Curve (AUC); a value close to 0.5 indicates that the two domains are similar in distribution, while a value significantly above 0.5 indicates a clear distributional difference.

4.4 Visual analysis

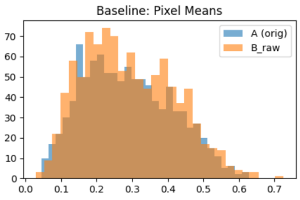
In addition to these quantitative methods, we also employed visual analysis techniques. Specifically, we visualized the distribution of average pixel values in both domains using histograms for intuitive comparison. Furthermore, we compared sample images from each domain side by side to better understand the real impact of the data transformations.

5. Results & Interpretation

5.1 Dataset A vs. Dataset B_raw

This comparison simulates a real-world scenario where the distribution of incoming data matches that of the training data. The random split ensures similar distributions between domains.

	KS-test	p-value	KL	AUC
B_raw	0.0290	0.7947	1.4617	0.5436

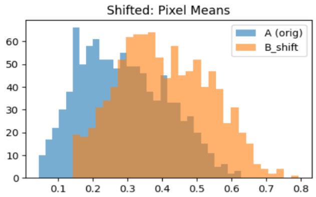


- The KS-test shows minimal difference between the cumulative distribution functions of the two domains, with a high p-value.
- The KL divergence is very small, indicating little to no informational difference between the distributions.
- The domain classifier yields an AUC close to 0.5, meaning it cannot distinguish between the two domains.

5.2 Dataset A vs. Dataset B_shift

This comparison evaluates the distributional difference between Dataset A and the transformed B_shift, simulating a potential distribution shift that may occur after deployment.

	KS-test	p-value	KL	AUC
B_shift	0.3440	0.0000	1.0000	112.8645



- p-value indicating the two distributions are not the same.
- The KL divergence shows a significant increase compared to the baseline, reflecting a larger shift in data distribution.
- The domain classifier's AUC is much higher than 0.5, suggesting that it can easily distinguish between A and B_shift.

6. Model Evaluation: Single Dataset Training and Deployment Impact

This section analyzes how the performance of an image classification model trained on a single domain (Domain A) changes when applied to real-world data with a different distribution (Domain B_shift). This reflects the potential issues that can arise when a machine learning model is deployed in actual user environments.

6.1 Model Description

We used ResNet-18 for its strong performance and real-world applicability. Its residual connections help prevent gradient vanishing in deep networks. Using pre-trained weights, we adapted the model by modifying the first layer for grayscale input and adjusting the output layer to classify the 10 FashionMNIST classes.

6.2 Experimental Setup and Training Environment

- **Training Dataset**: Domain A (randomly selected 42,000 FashionMNIST images)
- **Test Datasets**
    - (1) Same domain: Domain A (In-domain evaluation)
    - (2) Shifted domain: Domain B_shift (Out-of-domain evaluation)

The model was trained for 3 epochs using the Adam optimizer with a learning rate of 1e-4.

6.3 Result

	Data A	Data B_shift
Accuracy	0.9075	0.3398

The model achieved high accuracy on the training-distribution dataset (Domain A), but experienced a performance drop of approximately 56% when evaluated on the shifted-distribution dataset (Domain B_shift).

	Precision	Recall	F1-Score
Class 7	0.0000	0.0000	0.0000

In particular, the Sneakers class (label 7) showed precision, recall, and F1-score of 0.0000, indicating that the model completely failed to detect this class under distribution shift.

6.4 Problems that can occur in a physical deployment

This failure resulted from the intentional 90% reduction of sneaker samples in the B_shift set. In a real-world shopping platform, this would lead to the complete omission of sneakers in automatic tagging or search results, potentially damaging user trust.

7. Discussion on Monitoring & Mitigation Methods

To mitigate such degradation, real-time monitoring of distribution shift is crucial in deployed systems. Statistical metrics such as KS-test, KL Divergence, and domain classifier AUC—as used in this study—can help detect shifts early. These metrics should be tracked continuously, and exceeding predefined thresholds should trigger alerts or automated responses.

In addition, a regular model re-training (fine-tuning) strategy is recommended. By collecting real user-uploaded images periodically and merging them with existing training data, the model can be updated to maintain performance. Special attention should be paid to underrepresented classes like sneakers, where oversampling or synthetic data generation may be necessary.

Finally, uncertainty-based prediction management can be employed. For example, predictions with low confidence or close probabilities between multiple classes can be flagged for manual review, preventing critical misclassifications and improving user experience.

8. Conclusion

This study experimentally demonstrates how sensitive a model can be to distribution shifts between training and deployment data. A model trained on a single domain suffers significant performance degradation in real-world settings—especially for specific classes—highlighting the need for robust monitoring and adaptive strategies to ensure stable deployment.