Developing a noise-aware AI system for change risk assessment with minimal human intervention

Subhadip Paul, Anirban Chatterjee, Binay Gupta and Kunal Banerjee*

Walmart Global Tech. Bengaluru, Karnataka, India

Abstract

Introducing changes to a system in production may sometimes result in failures, and eventual revenue loss, for any industry. Therefore, it is important to monitor the "risk" that each such change request may present. Change risk assessment is a sub-field in operations management that deals with this problem in a systematic manner. However, a manual or even a human-centered AI system may find it challenging to meet the scaling demands for a big industry. Accordingly, an automated system for change risk assessment is highly desired. There are a few commercial solutions available to address this problem but those solutions lack the ability to deal with highly noisy data, which is quite a possibility for such systems. There are literature which proposed methods to integrate the feedback of domain experts into the training process of a machine learning model to deal with noisy data. Even though some of these methods produced decent risk prediction accuracy of the model but such an arrangement to collect feedback from the domain experts continuously has practical challenges due to the limitation in bandwidth and availability of the domain experts at times. Therefore, as part of this work, we explore a way to take the transition from a human-centered AI system to a near-autonomous AI system, which minimizes the need of intervention of domain experts without compromising with the prediction accuracy of the model. Initial experiments with the proposed AI system exhibit 10% improvement in risk prediction accuracy in comparison with the baseline which was trained by integrating the feedback of domain experts in the training process.

Keywords

change management, risk assessment, human-centered decision making,

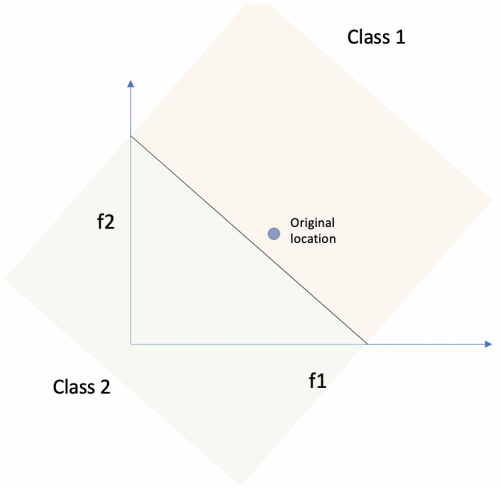
1. Introduction

Launching a new business or expanding the repertoire of features for an existing business is a common phenomenon in the modern technology-driven industries. All such upgrades require a series of software changes to a base system that is already in production. However, one needs to be cautious prior to pushing in these changes because each one of these potentially can cause a failure in the system. In the current era of agile development, often a large volume of requests come right before the sprint deadlines. At times, a tight delivery schedule severely restricts the scope for thorough inspection and review before the deployment. Moreover, from our experience, in case of manual change risk assessment, when the risk associated with a change is marked as "low" by the change requester (which, in reality, need not be so – this may happen if the developer is new or less skilled, and hence may have applied poor judgement), that request is often completely disregarded by the domain experts while reviewing, which eventually may manifest as a critical issue later in the pipeline. Reducing the number of failures in a production system is one of the key challenges for an industry to provide seamless service to its customers.

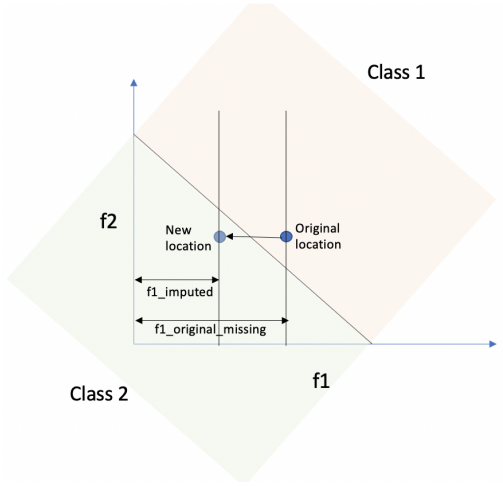
There are a few commercial solutions, such as the one provided by [1], which address the problem of automated change risk assessment. In [2], the authors addressed few of the limitations of the currently available commercial solutions such as concept drift in data and seeking feedback from domain experts depending on the estimated uncertainty of the model and few others. However, in practice, the problem of predicting risk associated with a change request can be further exacerbated by the presence of label noise in the data. Such label noise can be primarily attributed to inaccuracies crept in during imputation of missing values and some remedial intervention by the change management team which prevents some of the change requests from failing in production. We need experts' frequent and elaborate feedback on several data samples to ensure high reliability and generalization accuracy of the model which is trained with change data with high degree of label noise. However, frequent and elaborate feedback from the domain experts may not be always practically possible due to the limitation of bandwidth and availability of the domain experts. That motivates us into a transition from human-centered AI system to a near-autonomous AI system to predict risk of change requests in order to minimize the requirement of intervention by the domain experts.

In this paper, we present our experience of exploring the following questions while building an automated change risk assessment system:

Proceedings of the CIKM 2022 Workshops
*Corresponding author.
subhadip.paul0@walmart.com (S. Paul);
anirban.chatterjee@walmart.com (A. Chatterjee);
binay.gupta@walmart.com (B. Gupta);
kunal.banerjee1@walmart.com (K. Banerjee)
0000-0002-0605-630X (K. Banerjee)



(a) Data without missing values



(b) Data with imputed missing values

Figure 1: Data with and without imputed missing values.

- How can the label noise in the data affect the generalization accuracy of risk prediction model ?
- Can we have an automated process to remove the label noise in the data and train a model simultaneously?

The remainder of the paper is organized as follows. Section 2 covers the background and motivation of our work. Section 3 briefly explains our methodology. Section 4 provides the dataset description and the experimental results. Lastly, Section 5 describes some future work along with the concluding remarks.

2. Background & Motivation

In course of explaining our motivation into the transition from human-centered AI system to an autonomous AI system, we revolve our discussion around the following question one by one:

- ***Question 1.*** *How can label noise get introduced into the change data?*
- ***Question 2.*** *How can label noise impact the generalization error of the risk prediction model?*

2.1. Analysis of Question 1

There are multiple ways in which label noise may get introduced into the data. In the context of our change data, let us introduce two primary reasons for label noise:

2.1.1. Feature Sparsity in Data.

Some of the features of our data exhibit high degree of sparsity. We impute the missing values but some error always gets introduced by the process of imputation. Let us try to understand why the error originating from the process of missing value imputation leads to label noise.

Consider a toy example where a data instance has two features (refer to Figure **??**) and originally it belongs to *'class 1'*. Now consider a situation where the same data point as depicted in Figure **??** has the value of feature f_1 missing and it is eventually imputed (refer to Figure **??**). Notice that, after imputation the data instance has moved leftward and got located in the region of *'class 2'*. However, in spite of the new location of the data instance in the region of *'class 2'* after imputation, it is still labelled as *'class 1'* as that was the original label of the data instance. It eventually introduces label noise in the dataset. Notice also that the data instances located close to the class boundary are more prone to produce label noise in case the missing values of some of their features are imputed.

2.1.2. Change management process of the organization.

Another major source of label noise lies in the change management process itself of the organization. Consider a situation when a change request (CRQ) is raised which has high likelihood of causing failure in production and change manager along with the change requesting team took some mitigatory action against this CRQ to prevent it from causing failure in production. Due to such manual

intervention by the change management team, this CRQ may not end up causing failure in production. When this CRQ will be part of historical dataset for training risk prediction model, it will lead to the illusion that this CRQ belongs to 'normal' or 'non-risky' class as it didn't lead to any failure in production but ideally it should have been considered otherwise as this CRQ had high potential to cause failure in production. Therefore, such manual intervention in the change management process, which does not reflect in the change data, causes label noise in change data.

2.2. Analysis of Question 2

In our work, we model the change risk predictor as a binary classifier. Therefore, to understand the impact of label noise on the accuracy of change risk prediction model, we need to understand how label noise impacts a classifier in a generalized setting of supervised learning. Consider the following notations for the generalized setting of supervised classification: A training dataset $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ is available. In each pair (x_i, y_i), x_i represents the feature vector and y_i represents the associated label. X and Y denote the space of x and y respectively. Jointly (x, y) are drawn from an unknown distribution $\mathscr{P}$ over $X \times Y$. In other words, x is drawn from a distribution $\mathscr{D}$, and the true label y for x is given by a function $f : X \rightarrow Y$ drawn from a distribution $\mathscr{F}$. The learner's algorithm $\mathscr{A}$ represents a function which takes in the training data S as input parameters and returns a distribution of classifiers $h : X \rightarrow Y$. We define $err_{\mathscr{P}(\mathscr{A},S)} := E_{h\sim\mathscr{A}(S)}[err_{\mathscr{P}(h)}]$ to represent the generalization error function, where $err_{\mathscr{P}}(h) := \mathbb{E}_{\mathscr{P}}[\mathbb{1}(h(x)\neq y)]$ and $\mathbb{1}(\cdot)$ is the indicator function. We also assume $|X| = n$ and $|Y| = m$. We follow the notation below to characterize the training dataset S:

- Consider $\pi = \pi_1, \cdots, \pi_n$ to represent the priors for each $x \in X$.
- For each $x \in X$, sample a quantity p_x independently and uniformly from the set π.
- The probability mass function of x is given by $D(x) = \frac{p_x}{\sum_{x\in X} p_x}$.

When a model becomes sufficiently complex, many of the times it ends up memorizing the labels of some of the instances in the training dataset. Theorem 6 of [3] shows how memorizing noisy labels for data instances of frequency l leads to a sharper decline in the generalization power of a supervised classifier.

Theorem 1. (Theorem 6 of [3]) *For $x \in X_{S=l}$ with true label y, h memorizing its l noisy labels leads to the following order of individual excessive generalization error*: $\Omega\left(\frac{l^2}{n^2} \cdot weight\left(\pi, \left[\frac{2}{3}\frac{l-1}{n-1}, \frac{3}{4}\frac{n}{l}\right]\right)\right) \cdot \sum_{k\neq y} \mathbb{P}[\tilde{y} = k|x]$, where $weight(\pi, [\beta_1, \beta_2]) = \mathbb{E}\left[\sum_{x\in X} D(x) \cdot \mathbb{1}(D(x) \in [\beta_1, \beta_2])\right]$ and
n is the total number data instances in the training dataset.

Note that higher is the value of l, higher is the lower bound of the generalization error of the model. When it comes to dealing with tabular dataset with moderate to high dimension such as the dataset of ours, repetition of data instances in the dataset may apparently seem unlikely but still it is approximately possible. An intuitive explanation could be that in the context of supervised learning, a data instance is approximately represented by the set of its significant features with respect to the classification model even though there can be high number of insignificant or redundant features of that data instance. In that way, two data instances are seen by the model as repetition if the values of their significant features are the same.

Therefore, our primary motivation to take up this problem is to do away with the label noise due to some inherent noise generation process and its adverse impact on the model's accuracy.

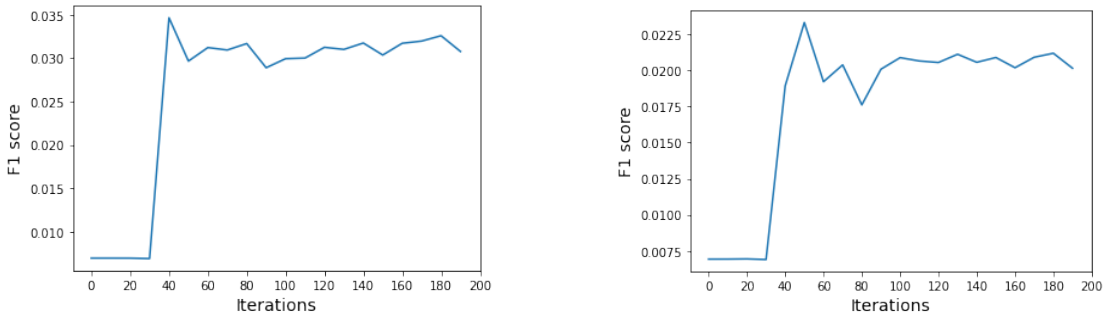
3. Methodology

As proposed by [4], we employ Progressive Label Correction (PLC) method to iteratively correct the labels and train the binary classification model. We first train the XGBoost model with the original noisy data for first few iterations and we call it warm-up period. We start correcting the labels once the warm-up period is over. We only correct those labels on which the classifier f exhibits high confidence. The idea is based on the intuition that there exists a region in the data in which noisy classifier f produces highly confident prediction and exhibit consistency with the clean Bayes optimal classifier. Thus within the specified data region as mentioned above, the algorithm produces clean labels. More formally, within the specified data region, if f predicts a different label than the observed label, $\tilde{y}$, with confidence above the threshold, θ, i.e. $|f(x) - 1/2| > \theta$, we flip the label $\tilde{y}$ to the prediction of f. We continue this process until we reach a stage where no label can be corrected. We choose the value of θ empirically.

4. Experimental Setup & Results

4.1. Dataset Description

We have collected change data for 3 months which comprises of ~27K data samples that are labelled as "risky" (class 1), i.e., potentially may lead to a failure in the production system, or "normal" (class 0). Out of the ~27K



(a) Experiment 1.

(b) Experiment 2.

Figure 2: Change in F1 score with iterations while training the model with PLC method (with a warm-up period of 30 iterations).

data samples there are only 65 instances which belong to the class "risky" or the positive class.

- **Feature Description:** Each instance in the data consists of 20 features; out of these, 2 features are continuous while the rest are categorical.
- **Sparsity:** There are many features that have missing values; some of the features even have almost 30% values missing.

4.2. Experimental Results

Table 1
Comparison of results of two models for Experiment 1

Experiment 1	Baseline	After PLC
True Positive Rate	0.62	**0.89**
False Positive Rate	0.19	**0.05**
True Negative Rate	0.81	**0.95**
False Negative Rate	0.38	**0.11**
Precision	0.06	**0.26**
Positive Likelihood Ratio	3.35	**17.57**
F1 Score	0.11	**0.40**
Balanced Accuracy	0.72	**0.92**

Table 2
Comparison of results of two models for Experiment 2

Experiment 2	Baseline	After PLC
True Positive Rate	0.51	**0.74**
False Positive Rate	0.06	**0.02**
True Negative Rate	0.94	**0.98**
False Negative Rate	0.49	**0.26**
Precision	0.05	**0.20**
Positive Likelihood Ratio	8.19	**42.44**
F1 Score	0.09	**0.32**
Balanced Accuracy	0.72	**0.86**

We create 3 separate datasets for each month from the overall data and perform two experiments:

- **Experiment 1**: The model is trained with the change data of Month 1 and the change data of Month 2 is used for validation.
- **Experiment 2**: The model is trained with the change data of Month 2 and the change data of Month 3 is used for validation.

We use a gradient-boosted decision tree (XGBoost) [5] to generate the probability with which a new change request may cause failure in production. We consider this probability as the estimation of the risk for a change. This is our baseline model. Note that we had explored other models as well; however, the XGBoost model produced the best results as recorded in our prior work [2].

Next we use the PLC algorithm [4] to remove the label noise in the dataset. Then we re-train the XGBoost model with the label corrected dataset iteratively as described in the previous section. Detailed comparisons between the baseline model (trained on original data) and the model trained following the PLC method for Experiment 1 and Experiment 2 are shown in Table 1 and Table 2, respectively. Note that the metric *balanced accuracy* is useful when the classes are imbalanced and is defined as *(sensitivity + specificity)/2*; we believe that the rest of the metrics used in these tables are standard and need no definition. As can be seen from Table 1 and Table 2, the model trained with PLC method outperformed the baseline across all the metrics. Figure 2 shows the plot of how F1 score varies with iterations during training the model with PLC method. Note that we had used a warm-up period of 30 iterations, which is why a sharp jump is noticed upon applying label correction 31st iteration onward.

5. Conclusion & Future Work

In this paper we have shown how we made transition from a human-centred AI system to a near-autonomous AI system by employing progressive label correction method in order to get rid of inherent label noise in the data. We now seek labels for a handful of samples from the domain experts only when the model is extremely uncertain about their class. Experimental results exhibit significant improvement in the model's performance with respect to various metrics.

As part of the future work, we aim to build not just a *more accurate* model but a *more accurate and trustworthy* model as earning the trust of the end users for the ML model is the key to success in driving business values by ML especially in '*operations*' in a large-scale organization. Therefore, we are in the process to build an enhanced label-noise removal method which is based on the intuition that *in noisy data, there exists a 'data region' in which the noisy classifier f produces highly confident and trustworthy prediction which is consistent with the clean 'Bayes optimal classifier'*. A standard approach to quantify a classifier's trustworthiness is to use its own estimated confidence or score such as probabilities from the softmax layer of a neural network, distance to the separating hyper-plane in support vector classification or mean class probabilities for the trees in a random forest. However, latest research shows that a higher confidence score from the model does not necessarily assure higher probability of correctness of the classifier. Therefore, the fact that, a classifier's own confidence score may not be the best judge of its own trustworthiness, makes our on-going work all the more challenging but interesting.

References

[1] Digital.ai, Change risk prediction (2019). URL: https://digital.ai/change-risk-prediction.

[2] B. Gupta, A. Chatterjee, S. Paul, H. Matha, L. Parsai, K. Banerjee, V. Agneeswaran, Look before you leap! designing a human-centered AI system for change risk assessment, in: ICAART, 2022, pp. 655–662.

[3] Y. Liu, Understanding instance-level label noise: Disparate impacts and treatments, in: ICML, volume 139 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 6725–6735.

[4] Y. Zhang, S. Zheng, P. Wu, M. Goswami, C. Chen, Learning with feature-dependent label noise: A progressive approach, in: ICLR, 2021. URL: https://openreview.net/forum?id=ZPa2SyGcbwh.

[5] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: KDD, 2016, pp. 785–794.