K-means Clustering Based Feature Consistency Alignment for Label-free Model Evaluation

Shuyu Miao, Lin Zheng, Jingjing Liu, and Hong Jin
Tiansuan Lab, Ant Group
{miaoshuyu.msy, zhenglin.zhenglin, jingjing.lqq, jinhong.jh}@antgroup.com

Abstract

The label-free model evaluation aims to predict the model performance on various test sets without relying on ground truths. The main challenge of this task is the absence of labels in the test data, unlike in classical supervised model evaluation. This paper presents our solutions for the 1st DataCV Challenge of the Visual Dataset Understanding workshop at CVPR 2023. Firstly, we propose a novel method called K-means Clustering Based Feature Consistency Alignment (KCFCA), which is tailored to handle the distribution shifts of various datasets. KCFCA utilizes the K-means algorithm to cluster labeled training sets and unlabeled test sets, and then aligns the cluster centers with feature consistency. Secondly, we develop a dynamic regression model to capture the relationship between the shifts in distribution and model accuracy. Thirdly, we design an algorithm to discover the outlier model factors, eliminate the outlier models, and combine the strengths of multiple autoeval models. On the DataCV Challenge leaderboard, our approach secured 2nd place with an RMSE of 6.8526. Our method significantly improved over the best baseline method by 36% (6.8526 vs. 10.7378). Furthermore, our method achieves a relatively more robust and optimal single model performance on the validation dataset.

1. Introduction

Label-free model evaluation task, also known as AutoEval [9], requires models to evaluate the performance of datasets autonomously without explicit labels or categories. The models must identify inherent patterns and structures within the data without relying on pre-defined labels. Unlike supervised model evaluation [7, 13, 23, 30, 32, 33], AutoEval does not require a vast amount of labeled data, as shown in Figure 1, saving time and expensive costs. Furthermore, it can reveal potential data patterns and relationships that may not be discovered by supervised evaluation.

However, this task is challenging due to the lack of explicit labels. Additionally, a test set comprises numerous images, and each image has varied and rich visual content [7]. In the 1st DataCV Challenge of the Visual Dataset Understanding workshop held at CVPR 2023 [1], participants are required to design a model that can estimate the accuracy of a given model on test sets without ground truths.

In our daily lives, AutoEval mirrors real-world scenarios more closely. Evaluating the performance of an online model on out-of-time or out-of-distribution datasets typically requires data annotation, which can be prohibitively expensive and time-consuming. For instance, various risk data are often encountered in financial risk control scenarios. In order to detect various risky transactions, it is essential to evaluate the model’s performance in real-time. Hence, determining how to evaluate the model’s performance with unlabeled test datasets is crucial.

Recently, several studies have demonstrated promising
performance in this task [5, 8, 9, 14, 16, 22, 26]. Calibration generated on the unseen distribution (target domain) yields consistent estimates and further helps infer the model’s performance [14, 26]. However, methods that require calibration in the target domain frequently produce poor estimates because deep learning models trained and calibrated on seen data (source domain) may not be calibrated in the previously unseen target domain. Some proposed methods [8, 9, 16] introduce additional labeled data from several target domains to learn a regression function of a distributional distance, which then predicts model performance. This method assumes that a strong correlation exists between the invisible test set and the visible train/val set in the fundamental distance measurement. The challenge baselines follow this paradigm, making it crucial to identify an appropriate linear correlation between the seen train sets and the unseen test sets and design an appropriate regression model. We address this challenge from three aspects: (1) designing excellent autoeval methods; (2) selecting the appropriate regressor; and (3) constructing the best integration strategy for multiple autoeval models.

To address the challenges stated above, we suggest three corresponding solutions. Firstly, we propose a novel model, K-means Clustering Based Feature Consistency Alignment (KCFCA), capable of representing the distribution shifts in various datasets. KCFCA utilizes the k-means clustering algorithm [17] to cluster the seen training set and unseen test set into clusters with a known number of categories. If the task at hand is N-classified, the centers of training samples and test samples that are clustered into N clusters should show close-to-distribution consistency. The distribution shifts between the two clustered centers can be used to fit a model regression. Secondly, experimental evidence has proved that different regression models will have a significant impact on the final result [9]. Therefore, we create a dynamic regression model that takes advantage of different regression models to fit the relationship between the shifts and the model accuracy. Thirdly, we design an outlier model factor discovery algorithm to eliminate outlier models and integrate the advantages of multiple autoeval models. In the course of this, we discover an interesting phenomenon: autoeval models based on various pre-trained models exhibit remarkable performance gaps. Lastly, our experiments validate the effectiveness of our solutions, and our model achieves second place on the DataCV Challenge leaderboard with an RMSE of 6.8526.

To summarize, this paper’s main contributions are as follows:

- We propose a novel method, K-means Clustering Based Feature Consistency Alignment (KCFCA), which can represent the distribution shifts in various datasets.
- We construct a dynamic regression model that fits the relationship between the distribution shifts and model accuracy.
- We design an outlier model factor discovery algorithm to eliminate outlier models and integrate the advantages of multiple autoeval models.

2. Related Work

Our work intersects with multiple related lines of research that have seen significant progress in recent years. Therefore, in this section, we provide a summary of the most closely related works.

2.1. Label-free Model Evaluation

Label-free model evaluation aims to predict the accuracy of an unseen test set when the ground truth is not accessible [5, 8, 9, 14, 16, 22, 26, 44]. This area has recently garnered widespread attention in the research community. Deng et al. [9] constructed a meta-dataset by transforming original images into various forms, adopted feature statistics to capture the distribution of a sample dataset, and trained a regression model to predict model performance. The difference of confidence was proposed [16] to yield successful estimates of a classifier’s performance across different shifts and model architectures. Such models rely on additional labeled data from several target domains to learn a linear regression function. The Average Thresholded Confidence (ATC) [14] method trained a threshold on the model’s confidence to predict accuracy as the fraction of unlabeled examples for which model confidence exceeds the threshold.

2.2. Out-Of-Distribution Detection

Out-of-distribution (OOD) detection is a critical task in machine learning that aims to identify examples outside the realm of the training distribution [10, 15, 21, 24, 28, 31, 36, 39, 44]. Model confidence outputs are commonly used as indicators to identify out-of-distribution samples [15, 21]. Liang et al. [28] proposed using temperature scaling and input perturbations to enhance OOD detection with model confidence. Devries and Taylor [10] introduced a method to learn confidence estimates for neural networks to produce intuitive and interpretable outputs. Sun et al. [41] designed ReAct - a straightforward and effective technique to reduce model overconfidence on OOD data, motivated by novel analysis on the internal activations of neural networks. Ren et al. [39] explored deep generative model-based approaches for OOD detection and observed that the likelihood score is heavily influenced by population-level background statistics. Learning the prediction uncertainty on OOD data remains a fundamental challenge in this task [24, 36].
2.3. Model Generalization Prediction

Predicting the generalization capabilities of models [2–4, 6, 25, 35, 40, 43] on unseen data has been a topic of interest in research for a long time. Complexity measurements on trained models and training sets have been explored as a means of predicting the generalization gap [2, 6, 25, 35]. Yang et al. [43] provided a simple explanation of this by measuring the bias and variance of neural networks to redefine how models generalize. Schiff et al. [40] used perturbation response (PR) curves to evaluate the accuracy change of a given network as a function of varying levels of training sample perturbation. Our work focuses on the cluster difference concerning the prediction of unseen test sets.

2.4. K-means Clustering

The k-means clustering algorithm is a popular unsupervised machine learning algorithm that serves to partition a given dataset into k clusters [17, 18, 27, 29, 34, 37]. In this algorithm, each data point is assigned to the cluster whose centroid is closest to it. The algorithm iteratively updates cluster centroids until convergence, which is achieved when the assignment of data points to clusters no longer changes. The basic principle of the k-means algorithm is to minimize the sum of squared distances between each data point and its assigned cluster centroid. The algorithm randomly initializes k centroids and then assigns each data point to the nearest centroid. In the next step, the centroids are updated by computing the mean of all the data points assigned to each cluster. This process is repeated until convergence. K-means is recognized as being both simple and efficient for partitioning datasets into k clusters. It has several advantages, such as computational efficiency and scalability.

3. Methods

3.1. Problem Formulation

We define this task by the source meta dataset $D^m(D_{train}, D_{val})$ (i.e., seen training dataset), which consists of the labeled training data $D_{train}$ and validation data $D_{val}$. Following the challenge and approach in [9], the source sample datasets $D^i = \{D_i(S_{xi}, S_{yi})\}_{i=1}^n$ are transformed from the original $D^m$, where $S_{xi}$ is the $i$-th training sample dataset, $S_{yi}$ are its corresponding labels, $n$ is the total number of the sample datasets, and $D_i$ is the $i$-th sample dataset. In addition, the target unlabeled test set is denoted as $D_t$, and does not contain any ground truths. We assume that the model $\mathcal{M}(\theta_m)$ is pretrained on $D^m$, where $\theta_m$ refers to the learned parameters that are fixed. If we have access to the label of $D_i$, we can easily obtain the accuracy using $acc = \mathcal{M}(D_i|\theta_m)$. However, in the absence of labeled ground truth, the objective of this task is to predict the accuracy of the unlabeled test set under the a priori conditions of $D^m$ and $\mathcal{M}(\theta_m)$, which is expressed as Equation (1).

$$acc = f_\theta(\omega, D_t|D^m, \mathcal{M}(\theta_m))$$
where \( f_\theta(\cdot) \) represents the regression model that needs to be learned and \( \theta_\omega \) are the parameters of this model.

Our work comprises three core components - K-means Clustering Based Feature Consistency Alignment (KCFCA), a novel AutoEval model that learns feature shifts between seen training data and unseen test data; Dynamic Regression Model (DRM), which strives to best fit the relationship between these shifts and model performance; and Outlier Model Factor Discovery (OMFD), which eliminates outlier autoeval models and integrates the advantages of multiple autoeval models. KCFCA will be discussed in Section 3.2, DRM in Section 3.3, and OMFD in Section 3.4.

### 3.2. K-means Clustering Based Feature Consistency Alignment

If we assume the meta task is a \( \mathcal{N} \) classification task, then \( \mathcal{M}(\theta_m, X_i) \in \{1, 2, \ldots, \mathcal{N}\} \), where \( X_i \) is \( i \)-th input image. By using the k-means clustering algorithm to cluster sample features, we can theoretically divide them into \( \mathcal{N} \) clusters with the best silhouette coefficient [17]. Additionally, we propose K-means Clustering-based Feature Consistency Alignment, abbreviated as KCFCA and shown in Figure 2. We first review and summarize the K-means:

1. **Initialize**: Choose the number of clusters as \( K \) and select \( K \) random points (centroids) from the dataset as the initial centroids.
2. **Assign**: Assign each data point to the nearest centroid based on the euclidean distance between the data point and the centroids.
3. **Update**: Recalculate the centroids of each cluster by taking the mean of all the data points in that cluster.
4. **Iterate**: Repeat steps 2 and 3 until convergence, which occurs when the centroids no longer change or a maximum number of iterations is reached.
5. **Output**: The algorithm outputs the \( K \) clusters and \( \mathcal{K} \) cluster centers, where each cluster contains a set of data points that are similar to each other and dissimilar to data points in other clusters.

As previously discussed, a dataset for a \( \mathcal{N} \) classification task can be clustered into the most suitable \( \mathcal{N} \) clusters. Given the sample datasets \( \{\mathcal{D}_i, \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}\} \), and pretrained model \( \mathcal{M}(\theta_m) \), we construct dataset pairs as \( \mathcal{D}_{\text{sam}} = \{\mathcal{D}_{\text{sam}}(\mathcal{D}_i, \mathcal{D}_{\text{val}}\}^{n=1} \) that can calculate the accuracy \( \text{acc}_i \). For each pair \( \mathcal{D}_{\text{sam}}(\mathcal{D}_i, \mathcal{D}_{\text{val}}) \), we first feed the \( \mathcal{D}_i \) and \( \mathcal{D}_{\text{val}} \) into the pretrained model \( \mathcal{M}(\theta_m) \) to extract the feature map \( \mathcal{F}_t \) and \( \mathcal{F}_{\text{val}} \). The feature \( \mathcal{F}_t \) and \( \mathcal{F}_{\text{val}} \) are then clustered into \( \mathcal{N} \) clusters by K-means, with the cluster centers \( \{\mathcal{F}_i\}_{i=1}^{\mathcal{N}} \) and \( \{\mathcal{F}_{\text{val}}\}_{i=1}^{\mathcal{N}} \). Ideally, the distribution of each dataset pair should exhibit feature consistency. However, due to the uncertainty of unseen test data, distribution shifts often occur. Thus, we model the feature distance using frechet distance to fit these distribution shifts by [9, 12]. This distance is denoted as \( D_i = D(\mathcal{F}_{ci}, \mathcal{F}_{\text{val}}) \). Finally, a regression model \( f_\theta(\cdot) \) is designed to regress the relations between \( D_i \) and \( \text{acc}_i \).

KCFCA can be formulated as:

\[
\text{acc} = f_\theta(\theta_\omega, D(\mathcal{K}(\mathcal{M}(\theta_m, \mathcal{D}_{\text{val}})), \mathcal{K}(\mathcal{M}(\theta_m, \mathcal{D}_i)))
\]

where \( \mathcal{K}(\cdot) \) is the k-means clustering algorithm.

The training and testing process of the model can be outlined as follows.

- **Training**: the regression model \( f_\theta(\cdot) \) is adopted to learn the relation of \( \{\mathcal{D}_i, \text{acc}_i\} \).
- **Testing**: calculate the feature distance between \( \mathcal{D}_{\text{val}} \) and \( \mathcal{D}_t \), put it into the regression model \( f_\theta(\cdot) \), and obtain the dataset accuracy.

Specifically, the KCFCA algorithm can be represented as the following Algorithm 1.

\[
\begin{align*}
\text{Algorithm 1} & \quad \text{K-means Clustering Based Feature Consistency Alignment} \\
\text{Require:} & \quad \text{The sample datasets \{\mathcal{D}_i, \mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}\}, and pretrained model} \ \mathcal{M}(\theta_m) \\
\text{Ensure:} & \quad \text{The accuracy of the unlabeled test set} \ \mathcal{D}_t. \\
1: & \quad \text{Training:} \\
2: & \quad \text{for number of source samples datasets training: do} \\
3: & \quad \quad \text{a. Feature extraction} \\
4: & \quad \quad \mathcal{F}_t = \mathcal{M}(\theta_m, \mathcal{D}_i), \mathcal{F}_{\text{val}} = \mathcal{M}(\theta_m, \mathcal{D}_{\text{val}}) \\
5: & \quad \quad \text{b. K-means clustering} \\
6: & \quad \quad \{\mathcal{F}_i\}_{i=1}^{\mathcal{N}} = \mathcal{K}(\mathcal{F}_t), \{\mathcal{F}_{\text{val}}\}_{i=1}^{\mathcal{N}} = \mathcal{K}(\mathcal{F}_{\text{val}}) \\
7: & \quad \quad \text{c. Feature distance} \\
8: & \quad \quad \mathcal{D}_i = D(\mathcal{F}_{ci}, \mathcal{F}_{\text{val}}) \\
9: & \quad \quad \text{d. Learn regression model} \\
10: & \quad \quad \text{acc} = f_\theta(\theta_\omega, D(\mathcal{K}(\mathcal{M}(\theta_m, \mathcal{D}_{\text{val}})), \mathcal{K}(\mathcal{M}(\theta_m, \mathcal{D}_i))) \\
11: & \quad \text{end for} \\
12: & \quad \text{Testing:} \\
13: & \quad \text{Calculate the accuracy of} \ \mathcal{D}_{\text{test}}: \\
14: & \quad \text{acc} = f_\theta(\theta_\omega, D(\mathcal{K}(\mathcal{M}(\theta_m, \mathcal{D}_{\text{val}})), \mathcal{K}(\mathcal{M}(\theta_m, \mathcal{D}_i))) \\
16: & \quad \text{END}
\end{align*}
\]

### 3.3. Dynamic Regression Model

In Section 1 and in our experiments of Section 4.2.2, we observed that different regression models have distinct advantages when using the same feature input. This highlights the importance of designing a suitable regression model. To address this issue, we propose a Dynamic Regression Model, named DRM, that incorporates the advantages of multiple regression models. DRM comprises several base regressors \( f_{\theta_b} = \{f_{\theta_i}\}_{i=1}^{\mathcal{I}} \) and a meta-regressor.
As shown in Section 4.2.3, we observed that different autoeval algorithms have varying performance across distinct unlabeled test sets. In this challenge, combining different autoeval algorithms can improve the final prediction performance significantly. However, we found that simply fusing the results of all the algorithms cannot achieve optimal outcomes because outlier models may appear on different datasets or pre-trained models. Hence, we propose an Outlier Model Factor Discovery (OMFD) method to eliminate autoeval algorithms with lower performance stability.

Intuitively, most models predict that the consistency of results is more likely to be the correct result. Conversely, there is a possibility of the results being wrong. We denote the various performances of autoeval algorithms as \( \mathcal{A} = \{ A_i \}_{i=1}^m \), where \( A_i \) is the \( i \)-th autoeval model. We define a threshold \( \tau \) for the anomaly factor that measures whether the model is an anomalous outlier. The flow of OMFD can be illustrated as:

1. Initialize: Visualize the performance of the autoeval model and manually select the appropriate centroid.
2. Calculate: Calculate the distance between the other autoeval models and this center.
3. Mark: If the maximum distance is greater than threshold \( \tau \), the corresponding model is marked as an outlier.
4. Iterate: Repeat steps 1, 2 and 3 until convergence, which occurs when the maximum distance is no longer greater than the threshold \( \tau \).
5. Output: Autoeval models marked as outliers.

We blend all the autoeval models except outlier models to achieve the best model performance. Note that the threshold \( \tau \) can be debugged based on the validation set or set empirically.

4. Experiments

In all of our experiments, we follow the same dataset and settings as the DataCV Challenge [1].

4.1. Experimental Settings

4.1.1 Datasets

- **Training dataset:** The training dataset consists of 1,000 transformed datasets from the original CIFAR-10 test set, using the transformation strategy proposed by Deng et al. [9].
- **Validation dataset:** The validation set was composed of CIFAR-10.1 [38, 42], CIFAR-10.1-C [20] (add corrup-
Table 1. The validation dataset results are based on the same pre-trained model. The colored cells are sub-optimal and optimal performance. “Dataset A” indicates the training dataset is provided by the challenge; “Dataset B” denotes the training dataset is regenerated using the same transformed strategy as the challenge and [9]. Note that we just generate such a training dataset in this part to validate the robustness of various autoeval methods, and we just use the training dataset provided by the challenge to summit in the challenge leaderboard.

- Test dataset: The test set comprises 100 datasets provided by the challenge [1].
- • Pretrained classifier models

In our experiments, we follow this challenge and evaluate the classifiers ResNet-56 [19] and RepVGG-A0 [11]. Both implementations can be accessed in the public repository at the website 3. To benefit from the models and load their pre-trained weights, use the code provided on the website.

4.1.3 Evaluation metrics

The evaluation metric used in our experiments is the root-mean-square error (RMSE), which can be formulated as:

\[
RMSE = \sqrt{\frac{\sum_{t=1}^{T} (\hat{y}_t - y_t)^2}{T}}
\]

4.2. Experiments and Findings

To verify the effectiveness of our proposed KCFCA, DRM, and OMFD, we conducted detailed experiments on the same validation datasets.

4.2.1 Experiments on various autoeval methods

In our study, we conduct comprehensive experiments on various autoeval methods, including ConfScore [21], Entropy [16], Rotation [8], ATC [14], FID [9], and our proposed KCFCA. All of the experiments are conducted based on the pre-trained ResNet56 provided by the challenge. Moreover, to validate the robustness of the various models, we perform experiments on 1,000 transformed datasets (denoted as “Dataset A”) provided by the challenge and an additional 1,000 transformed datasets (denoted as “Dataset B”) generated using the same transformation strategy as the challenge and [9]. Note that we only use the training dataset provided by the challenge to submit the challenge results.

As shown in Table 2, it is rare to find a regression model that guarantees to outperform under all methods. However, it is evident that MLP has the least satisfactory outcome. It is encouraging to find that our proposed DRM can achieve relatively stable and excellent performance across different methods. For the final challenge submission, we experimentally use the LR regressor on ResNet-56 and the DRM.
4.2.3 Experiments on multiple autoeval methods

To investigate the impact of different model fusion methods on the final challenge results, we conducted a series of experiments. The methods include ConfScore [22], Entropy [16], Rotation [8], ATC [14], FID [9], the average of all methods (AVG), and our proposed Outlier Model Factor Discovery (OMFD). We present all the results in Table 3.

The table indicates that averaging the results of all methods leads to decent results, surpassing any single method, with an average score of 3.870. However, leveraging our OMFD method to eliminate anomalous outlier methods leads to surprising optimal results of 2.873, a 34.7% improvement over the average score. Thus, our findings suggest that the inclusion of anomalous outlier methods is detrimental to the fusion process and adversely affects the final model output.

4.3. Results on DataCV Challenge

In the challenge, there are two models ResNet-56 and RepVGG-A0 to be evaluated on the unlabeled test set in total by RMSE. The results of the challenge are shown in Table 4.

For our final challenge submission, we combined the K-means Clustering Based Feature Consistency Alignment (KCFCA), Dynamic Regression Model (DRM), and Outlier Model Factor Discovery (OMFD) methods. Our team secured second place in the challenge, as shown in the table. Additionally, our proposed approach outperformed the optimal model results [9] provided in the challenge, achieving a 36% improvement with a RMSE score of 6.8526 compared to 10.7378.

5. Conclusion

This paper highlights the various strategies we adopted in the challenge. Specifically, we propose the K-means Clustering Based Feature Consistency Alignment method to represent distribution shifts in different datasets, Dynamic Regression Model to analyze the relationship between shifts and model performance, and Outlier Model Factor Discovery to remove anomalous outlier autoeval models. Our approach secured second place in the challenge ranking. Furthermore, our KCFCA method achieved the most robust and optimal single model performance on the validation dataset.
References


3307