REPRODUCIBILITY REPORT - FREE LUNCH FOR FEW-SHOT LEARNING: DISTRIBUTION CALIBRATION

Pok Man Kwan
3323 4132
pmk1u21@soton.ac.uk

Brijee Rana
3338 5874
br1e21@soton.ac.uk

Sanrakshak Adhikari
3002 9155
sa2e21@soton.ac.uk

ABSTRACT

Few-shot learning is a hot research topic that focuses on learning from very few training data. In the paper *Free Lunch for Few-Shot Learning: Distribution Calibration* Yang et al. (2021), Yang et al. proposed to use the distribution of base classes where sufficient data is available to calibrate the distribution of the novel classes. Results in the paper show that by using this technique, a simple classifier can easily achieve a higher accuracy than the state-of-the-art techniques. The main objective of this report is to reproduce and validate the results shown in the paper. We show that most results in the paper are reproducible and further investigated the property of the algorithm with some extra experiments.

1 INTRODUCTION

In the past decade, advancements in machine learning algorithms, especially deep neural networks, have led to great success in many different fields such as computer vision and natural language processing. However, these models usually require a huge amount of high-quality data for training. Learning from only a few samples is a challenging task for these models due to the highly biased samples. To overcome the difficulties in few-shot learning, Yang et al. (2021) calibrates the distribution of novel classes in the feature space by using statistics from base classes, so that a more accurate data distribution can be used as the input of a classification model for better generalization.

There are two main reasons for choosing this paper for the reproducibility challenge. We believe that few-shot learning is very important for real-world applications, where data is often insufficient for models to make accurate predictions. If machines can learn from a small amount of data, we can tackle problems where collecting and labelling a large dataset requires high costs or is impossible. Another reason is that the method proposed in this paper can be applied on top of other methods, which makes it a general technique that can improve the performance of many few-shot learning models. In the sections below, the method is explained briefly to support the experiments, and then the implementation details and experiments to reproduce the results are described, followed by the results and discussions.

2 DISTRIBUTION CALIBRATION

The few-shot learning problem is defined as training a model on base classes (with a lot of samples) and then use the model to learn from novel classes with only a few samples (support set) and make predictions on the query set of the novel classes. To perform distribution calibration, all data is mapped into a latent feature space using a feature extractor. The authors assume the base class features are Gaussian distributions with mean $\boldsymbol{\mu_i}$ and covariance $\mathbf{\Sigma_i}$, where i is the index of the base class. Tukey's transformation is then applied to the support and query sets to make their distribution more Gaussian. Tukey's Transformation is defined as follows:

$$\tilde{\mathbf{x}} = \begin{cases} \mathbf{x}^{\lambda} & \text{if } \lambda \neq 0 \\ log(\mathbf{x}) & \text{if } \lambda = 0 \end{cases} \tag{1}$$

The top-k closest base classes are selected for each sample $\tilde{\mathbf{x}}$ in the support set according to the Euclidean distance between $\tilde{\mathbf{x}}$ and $\boldsymbol{\mu}_i$. The calibrated mean of sample $\tilde{\mathbf{x}}$ is then defined as the average of $\boldsymbol{\mu}_i$ of the k base classes with $\tilde{\mathbf{x}}$, while the calibrated covariance is the average of the k covariance matrices $\mathbf{\Sigma_i}$ plus a hyperparameter α.

$$\boldsymbol{\mu}' = \frac{\sum_{i \in topk} \boldsymbol{\mu}_i + \tilde{\mathbf{x}}}{k+1}, \mathbf{\Sigma}' = \frac{\sum_{i \in topk} \mathbf{\Sigma}_i}{k} + \alpha \tag{2}$$

With the calibrated mean and covariance, more samples can be sampled from this calibrated distribution and act as the input to a classifier such as support vector machine (SVM). Calibration is done in the feature space, so this method can be applied to any feature extractor and classifier, which makes it a general technique to improve the performance of many other few-shot learning models.

3 EXPERIMENTS

3.1 IMPLEMENTATION

The goal of this project is to reproduce the results from the paper and to validate the algorithm, so we decided to use the authors' data generation codes to make sure we are comparing our implementations on the same set of data. We reimplemented the proposed algorithm ourselves mainly using PyTorch and added codes for running experiments and plotting results. The codes can be found in `https://github.com/COMP6248-Reproducability-Challenge/Free-Lunch-for-Few-Shot-Learning`. Compare to the author's implementation, our PyTorch version is at least 2 times faster. However, because most of the experiments are repeated many times and the algorithm cannot fully utilize GPU acceleration, a lot of time is required to finish the experiments. As an example, a 5way5shot experiment repeated for 10000 tasks takes about 4-5 hours on the ECS GPU compute service with a single NVidia RTX2070. In the sections below, we will first show and compare the results we reproduced with the results in the paper, then show some extra results we produced with the distribution calibration technique.

3.2 COMPARISON TO STATE-OF-THE-ART

Table 1: Distribution Calibration with Simple Classifiers

Methods	*mini*ImageNet		*CUB*	
	5way1shot	5way5shot	5way1shot	5way5shot
SVM with DC	67.98 ± 0.20	83.12 ± 0.14	80.28 ± 0.20	90.78 ± 0.11
Logistic Regression with DC	68.09 ± 0.20	83.38 ± 0.14	80.24 ± 0.20	90.79 ± 0.11

First, we reproduced the results of using distribution calibration with SVM and logistic regression on both *mini*ImageNet and CUB. *mini*ImageNet and CUB are two commonly used image dataset for few-shot learning. These results are used in the paper to compare the performance of the proposed method with other state-of-the-art methods. For this experiment, we used the default hyperparameters mentioned in the paper: $k = 2$, $\lambda = 0.5$, number of generated features is 750, and $\alpha = 0.21$ for *mini*ImageNet and $\alpha = 0.3$ for CUB. SVM and logistic regression from scikit-learn are used here with default settings. The results are generated by averaging the top-1 accuracy over 10000 few-shot learning tasks. Table 1 shows the result of the experiments. It can be seen that the accuracy of all the experiments are similar to the results in the paper.

3.3 ABLATION STUDY

Table 2: Ablation Study

Tukey Transformation	Training with generated features	*mini*ImageNet	
		5way1shot	5way5shot
No	No	60.06 ± 0.20	81.42 ± 0.14
Yes	No	64.66 ± 0.20	83.43 ± 0.13
No	Yes	63.53 ± 0.21	80.59 ± 0.14
Yes	Yes	68.09 ± 0.20	83.38 ± 0.14

Table 2 shows the results of the model's performance during ablation study; without Tukey's transformation for the features or generated features. The result from the original paper is reproducible as in a 5way1shot setting, the performance is almost 8% worse when neither options were used and around 5% if either of the options weren't used; which was similar performance as the original paper. However, in the 5way5shot setting the accuracy drops slightly when features are generated without Tukey's transformation.

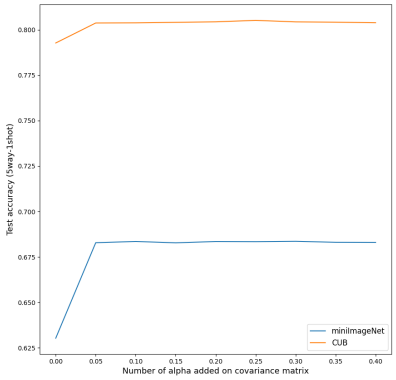

Figure 1: Effect of different values of α

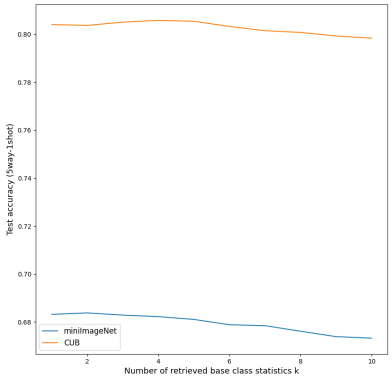

Figure 2: Effect of different values of k

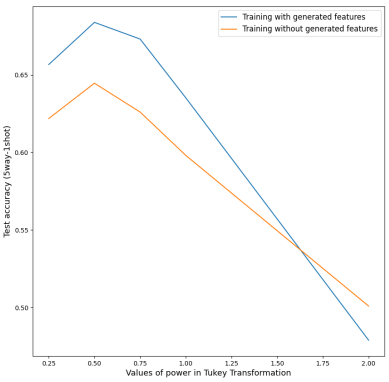

Figure 3: Effect of different values of λ

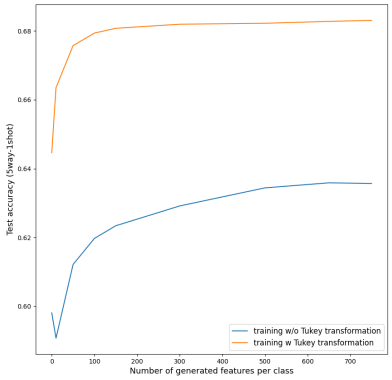

Figure 4: Effect of different number of generated features

3.4 Hyperparameter Tuning

The paper includes the testing of 4 hyper-parameters so that the best hyper-parameter value is chosen for the rest of their experiments. We reproduced these results as shown in Figures 1, 2, 3 and 4 which shows the accuracy of different values of α, k, λ and number of generated features. Our results are similar to the paper. However, for Figure 3, Tukey's transformation is undefined when $\lambda \leq 0$ if features contain zero values. Thus we could not reproduce those values and the paper and authors' code did not mention how to handle this problem.

3.5 t-SNE

We reproduced the t-SNE visualization for the calibrated distributions (Figure 5). It is not clear which dataset and what hyperparemeters the authors used to create the t-SNE plots, so we used the *mini*ImageNet with default hyperparameters except for the number of generated features and α. We generated 300 features from the calibrated distribuion and set $\alpha = 0$, which is crucial for generating a reasonable t-SNE plot. The generated features are shown on the left, the query set is shown on the right, and the support set is labelled as a star in both plots. The plots are not exactly the same with that in the paper because it depends on the specific support set and the unknown settings, but it is still clear that features in different classes are separated and are similar to the query set distribution (except for the blue class). This shows how using distribution calibration can reduce bias from the support set and improve the prediction accuracy.

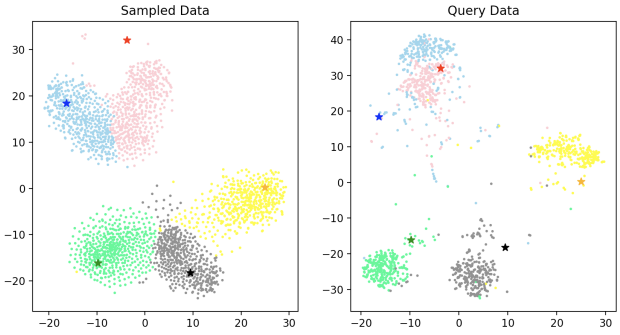

Figure 5: t-SNE visualization of the calibrated samples

3.6 Alternative Classifiers

Table 3: Using different classifiers on the same dataset and hyper-parameters

Methods	*mini*ImageNet		*CUB*	
	5way1shot	5way5shot	5way1shot	5way5shot
Naive Bayes	55.30 ± 0.21	57.77 ± 0.18	69.65 ± 0.23	66.59 ± 0.18
Naive Bayes with DC	30.26 ± 0.20	30.83 ± 0.22	32.87 ± 0.25	33.04 ± 0.26
Random Forest	24.93 ± 0.14	47.27 ± 0.18	24.81 ± 0.13	57.30 ± 0.20
Random Forest with DC	46.93 ± 0.19	57.01 ± 0.17	62.08 ± 0.23	76.61 ± 0.16
K-nearest neighbours	20.00 ± 0.00	67.93 ± 0.18	20.00 ± 0.00	82.30 ± 0.16
K-nearest neighbours with DC	64.47 ± 0.21	76.35 ± 0.16	78.73 ± 0.21	88.09 ± 0.13

The paper claims that the calibration can be paired with any classifier. In their experiments, they used a logistic classifier and SVM (as shown in Table 1). We decided to further investigate with the classifiers: Naive Bayes, random forest and KNN, using the default parameters from scikit-learn and applying them on the same dataset with the same hyper-parameters they have used. The results of our experiments are shown on Table 3. Our results show that other classifiers can be used, and the algorithm significantly increases the accuracy of random forest and KNN. However, Naive Bayes' accuracy decreases with the calibrated distributions. We suspect that this behaviour is due to the strong independence assumption of Naive Bayes, which is not satisfied by the generated features.

4 Conclusion

In this project, we reimplemented the algorithm proposed by Yang et al. (2021) and reproduced most of the results in the paper. We noticed that our implementation is faster than the author's implementation, but a few hours is still required to run repeated experiments. In general, the results from the paper are reproducible, which shows that the algorithm can really boost the accuracy of few-shot learning by generating more samples in the feature space using the calibrated distribution. Some results cannot be reproduced perfectly due to invalid feature values, unknown parameters, and unknown settings, but the performance and correctness of the algorithm is not really affected by these imperfectness. We also showed that the algorithm can boost the accuracy of other classifiers, but it could also harm the performance if it is used blindly without considering the properties of the classifier.

References

Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. In *International Conference on Learning Representations (ICLR)*, 2021.