This CVPR 2020 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Learning to Select Base Classes for Few-shot Classification

Linjun Zhou[1,2] Peng Cui[1,*] Xu Jia[2,*] Shiqiang Yang[1] Qi Tian[2]
[1]Tsinghua University [2]Noah's Ark Lab, Huawei Technologies
zhoulj16@mails.tsinghua.edu.cn, cuip@tsinghua.edu.cn
jiayushenyang@gmail.com, yangshq@tsinghua.edu.cn, tian.qi1@huawei.com

Abstract

Few-shot learning has attracted intensive research attention in recent years. Many methods have been proposed to generalize a model learned from provided base classes to novel classes, but no previous work studies how to select base classes, or even whether different base classes will result in different generalization performance of the learned model. In this paper, we utilize a simple yet effective measure, the Similarity Ratio, as an indicator for the generalization performance of a few-shot model. We then formulate the base class selection problem as a submodular optimization problem over Similarity Ratio. We further provide theoretical analysis on the optimization lower bound of different optimization methods, which could be used to identify the most appropriate algorithm for different experimental settings. The extensive experiments on ImageNet [4], Caltech256 [8] and CUB-200-2011 [27] demonstrate that our proposed method is effective in selecting a better base dataset.

1. Introduction

Few-shot Learning [6, 13] is a branch of Transfer Learning, its basic setting is to train a base model on the base dataset consisting of base classes with ample labeled samples, then adapt the model to a novel support set consisting of novel classes with few samples, and finally evaluate the model on the novel testing set consisting of the same novel classes as the novel support set.

Traditionally, many works focus on how to learn meta-knowledge from a fixed base dataset. The generation process of the base datasets generally depends on random selection or human experience, which is not necessarily perfect for few-shot learning. Due to the fact that the fine-tuning mechanism on the novel support set is not as effective as learning with large-scaled training samples on novel classes [25], the base dataset plays a critical role for the performance of few shot learning. Till now, however, we have little knowledge on how to measure the quality of a base dataset, and not to mention how to optimize the its selection process.

The targeting problem described above is somewhat related to Curriculum Learning [1, 24] and data selection in transfer learning [19–21]. Different from Curriculum Learning aiming to speed up learning of provided classes, we focus on learning to select base classes in a transfer learning manner, where the selected base classes are used for classification on novel classes. With respect to the data selection methods in transfer learning, first, our problem is a class-based selection instead of sample-based selection problem, which significantly decreases the search space for selection. Second, we consider the problem in a few-shot learning scenario, where there is no validation dataset on novel classes, and modern methods with feedback mechanism on validation performance (*e.g.* Bayesian Optimization in [21], Reinforcement Learning in [19]) are not applicable.

Here we consider a realistic and practical setting that M base classes are to be selected from N candidate classes, and each candidate class contains only a small number of labeled samples before selection. Once the M classes are selected, one could expand the samples of these selected classes to a sufficient size by manually labeling, which are further used to construct the base dataset and train the base model. The selection process could be conducted either in an one-time or incremental manner.

To solve the problem, we confront two challenges. First, the problem is a discrete optimization problem. The complexity of naive enumeration method is $O(N^M)$, which is intractable in real cases. Second, there is no touchable way to optimize the classification performance of novel classes directly, hence we need to find a proxy indicator that is both easy to optimize and highly correlated with the classification performance on novel classes.

In this paper, we find a simple yet effective indicator Similarity Ratio, first proposed by our previous work [30]. For a candidate class, the Similarity Ratio considers both its similarities with novel classes and diversity in base classes. We demonstrate that this indicator is highly and positively correlated with the performance of few-shot learning on the

* Co-corresponding authors.

novel testing set. We theoretically prove that this indicator satisfies submodular property, which pledges us to obtain a sub-optimal solution in polynomial time complexity. Thus, the base class selection problem could be surrogated by optimizing a variant of Similarity Ratio. We carry out extensive experiments on three different cases: the Pre-trained Selection, the Cold Start Selection, and the General Selection on ImageNet, Caltech256, and CUB-200-2011 datasets. Results show that our method could significantly improve the performance of few-shot learning in both general image classification and fine-grained image classification. The performance improvement margin is rather stable regardless of the distribution transfer from the support set to the query set, change of few-shot model, or change of few-shot experimental settings.

2. Related Work

Few-shot Learning The concept of One-shot Learning is proposed by [6], and a more general concept is Few-shot Learning. Three mainstreams of approaches are identified in the literature. The first group is based on a meta-learning manner, including Matching Network [25], MAML [7], Prototypical Network [22], Relation Network [23], SNAIL [14] *etc*, which learn an end-to-end task-related model on the base dataset that could generalize across all tasks. The second group of methods is learning to learn image classifiers for unseen categories via some transfer mechanism while keeping the representation space unchanged. The advantage of these methods is to avoid drastically re-training the model and more friendly to extremely large base datasets and model, *e.g.* classification on ImageNet. Common methods are MRN [29], CLEAR [11], Weight Imprinting [18], VAGER [30] *etc.* The third group of methods is to apply data generation. The core idea is to use a pre-defined form of generation function to expand the training data of unseen categories. Typical work includes [9] and [28].

Data Selection The underlying assumption of data selection is that not all training data is helpful to the learning process; some training data may even perform negative effects. Thus, it's important to distinguish *good* data points from *bad* data points to improve both the convergence speed and the performance of the model. Roughly there are two branches of work: one is to assume training data and testing data are sampled from the same distribution, a common way to deal with this problem is to reweight the training samples [5, 12, 24], which is out of the scope and will not be covered in this paper. The other branch is data selection in a transfer learning manner. Mainstream approaches include that [20] proposes a method based on heuristically defined distance metric to find most related data points in the source domain to the target domain; [21] views the effect of data selection process to final performance of the classification on target domain as a black box model and uses Bayesian Optimization to iteratively adjust the selection through performance on validation dataset and further [19] substitutes Bayesian Optimization to Reinforcement Learning, which is more suitable to introduce deep model to encourage more flexibility in designing selection algorithms.

3. Preliminary Study

3.1. Similarity Ratio

[30] first proposes a concept called Similarity Ratio (SR) defined for each novel class as:

$$\text{SR} = \frac{\text{Average Top-K Similarity with Base Classes}}{\text{Average Similarity with Base Classes}}. \quad (1)$$

Here the similarity of two classes is determined by a specific metric on the representation space, *e.g.* the cosine distance of two class centroids. Among all base classes, we sort the similarity of each base class with the corresponding novel class in a descent order. The numerator is calculated by averaging the similarity of the top-K similar base classes and the denominator is calculated by averaging the similarity of all base classes. To improve SR, the numerator indicates there should be some similar base classes with the corresponding novel class and the denominator indicates the base classes should be diversified conditioned on each novel class. [30] further points out that the few-shot performance is positively correlated with this indicator.

3.2. The Relationship Between SR and Few-shot Learning Performance

In this part, we will show more evidence from a statistical perspective of the relationship between SR and few-shot learning performance.

Specifically, a preliminary experiment is conducted as follows: we randomly choose 500 classes from ImageNet dataset, and further split them into 400 base classes and 100 novel classes. For each few-shot classification setting, we randomly select 100 base classes over 400 as the base dataset, and using all 100 novel classes to perform a 100-way 5-shot classification. A ResNet-18 [10] is trained on the base dataset, and we extract the high-level image features (512-dimensional features after conv5_x layer) for novel support set and novel testing set. We calculate the average feature for each novel class in the novel support set as the class centroid and directly use 1-nearest neighbor based on the cosine distance metric defined on the representation space to obtain the Top-1 accuracy for each novel class of the testing set. The base dataset selection, training and evaluating process is repeated for 100 times and for each novel class, we run the regression model:

$$Acc = \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + \alpha + \epsilon \quad (2)$$

$$\begin{cases} x_1 = \text{Average Top-K Similarity with Base Classes} \\ x_2 = \text{Average Similarity with Base Classes} \end{cases}$$

where Acc represents for the Top-1 accuracy for the corresponding novel class, α represents for the residual term and ϵ represents for noise. The similarity of two classes in this regression model is calculated by the cosine distance of two centroids defined on the representation space of ResNet-18 trained by all 400 candidate base classes. Hence, totally we could obtain 100 regression models, each for a novel class, and each model is learned under 100 data points related to 100 different choices of base dataset.

With a different choice of K, the regression model may show different properties. We conclude our findings from Figure 1, 2, 3.

We calculate the average of β_1 and β_2 for all novel classes, denoted as $\bar{\beta}_1$ and $\bar{\beta}_2$. $\bar{\beta}_1$ is constantly positive in all choices of K, demonstrating the positive effect of Average Top-K Similarity to accuracy. Figure 1 shows the change of coefficient $\bar{\beta}_2/\bar{\beta}_1$ with K. The result shows that $K = 5$ is a demarcation point in this specific setting. The positive effect of Average Similarity (*i.e.* x_2) will become negative after $K = 5$. The reason is that when K is small, the positive classes are insufficient, there is need to add more positive classes to improve the performance, and with the increase of K, the positive classes tend to saturate and there is an increasing need of negative classes to enhance diversity. In later main experiments, we set K to be a hyper-parameter.

Figure 2 is a snapshot for the two settings with $K = 3$ and $K = 10$, which further proves the viewpoint above. Moreover, Figure 2 gives more information about the distribution of β_1 and β_2.

Figure 3 shows that the two components of the SR are relatively good proxy of the performance for few-shot learning when K is a small number (*i.e.* The average R^2 reaches above 0.3 when $K \leq 10$). When $K = 1$ the two components of SR explain about 45% of the dependent variable.

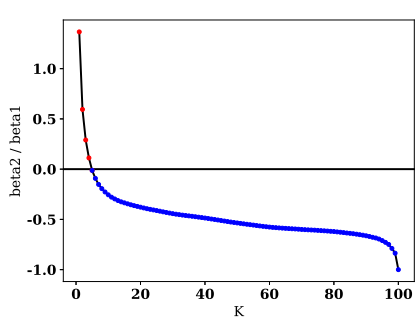

Figure 1. The coefficient $\bar{\beta}_2/\bar{\beta}_1$ changed with K.

Based on our findings, an optimization process could be designed to select core base classes.

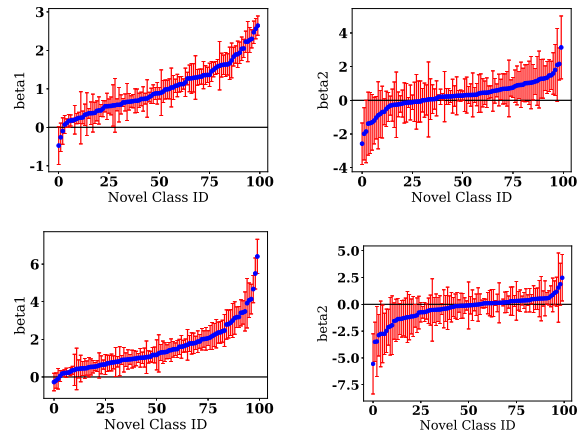

Figure 2. We plot the coefficients β_1, β_2 of each novel class after sorting increasingly. The red bar represents for the 95% confidence interval and the blue dot shows the exact coefficients. Top: result for Regression with $K = 3$, $\bar{\beta}_1 = 0.99$, $\bar{\beta}_2 = 0.29$; Bottom: result for Regression with $K = 10$, $\bar{\beta}_1 = 1.52$, $\bar{\beta}_2 = -0.39$.

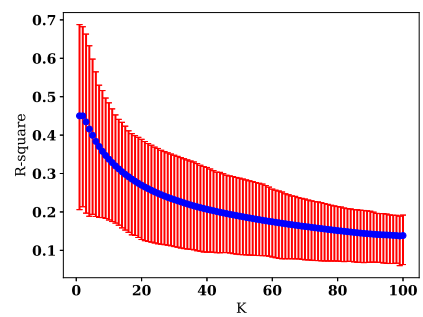

Figure 3. R^2 with the change of K for 100 regression models, the red bar represents for the interval from 25-quantile to 75-quantile, and the blue dot represents for the average R^2.

4. Algorithm

4.1. A Brief Introduction to Submodularity

Definition 1. *Given a finite set $V = \{1, 2, \cdots, n\}$, a set function $f : 2^V \to \mathbb{R}$ is submodular if for every $A, B \in V$: $f(A \cap B) + f(A \cup B) \leq f(A) + f(B)$.*

A better way to understand submodularity property is that of diminishing returns: denote $f(u|A)$ as $f(A \cup u) - f(A)$, then we have $f(u|A) \geq f(u|B)$ for every $A \subseteq B \subseteq V$ and $u \notin B$. These two definitions are proved to be equivalent [15]. It has been proved that maximizing a submodular objective function $f(\cdot)$ is an NP-hard problem. However, with polynominal time complexity, several algorithms have been proposed to obtain a sub-optimal solution.

A function is monotone non-decreasing if $\forall A \subseteq B, f(A) \leq f(B)$. $f(\cdot)$ is called normalized if $f(\emptyset) = 0$.

In this paper we mainly introduce a submodular optimization setting with cardinality constraint. The problem is formulated as: $max_{S \subseteq V, |S|=k} f(S)$, where $f(\cdot)$ is a sub-

modular function. [15] shows that a simple greedy algorithm could be used to maximize a normalized monotone non-decreasing submodular fuction with cardinality constraints, with a worst-case approximation factor of $1 - 1/e \approx 0.632$. [2] shows that a normalized submodular function (may not be monotone non-decreasing) with an exact cardinality constraint $|S| = k$ could reach an approximation of $max\{\frac{1-k/en}{e} - \epsilon, (1 + \frac{n}{2\sqrt{(n-k)k}})^{-1} - o(1)\}$ with a combination of random greedy algorithm and continuous double greedy algorithm, where k is the exact number of chosen elements and n is the total number of elements. The proposed algorithm guarantees a 0.356-approximation, which is smaller than 0.632.

4.2. Formulation

Let B_u represent for collection of unselected base classes, B_s for selected base classes and N for novel classes. The selection process is to select a subset U with m elements from B_u and the base dataset is composed of U and B_s. For each class l, we denote c_l as certain class feature (*e.g. its centroid of high-level feature*), and for each class set A, we denote $c_A = [c_{l_1}, c_{l_2}, \cdots c_{l_{|A|}}], l_1, l_2 \cdots l_{|A|} \in A$ as a collection of class features.

Next, we define an operator *max-k-sum* as follows:

$$M^k(y) := \max_{|K|=k} \sum_{i \in K} y_i = \sum_{j=1}^{k} y_{[j]},$$

where y is a numerical vector, $y_{[1]}, \cdots, y_{[n]}$ are the y_i's listed in nonincreasing order. Based on our findings that SR is highly and positively correlated to the performance on novel classes in Section 3, the base class selection problem could be formulated as an optimization process on SR as a proxy. Concretely we have:

$$\begin{aligned} \max_{\substack{U \subset B_u \\ |U|=m}} & \frac{1}{|N|} \sum_{n \in N} \frac{1}{K} \cdot M^K(f(c_n, \{c_{B_s}, c_U\})) \\ & - \frac{\lambda}{|N|} \cdot \sum_{n \in N} \frac{1}{|B_s| + m} \sum_{u \in B_s \cup U} f(c_n, c_u), \end{aligned} \tag{3}$$

where $f(c_a, \{c_{b1}, \cdots, c_{bn}\}) = [f(c_a, c_{b1}), \cdots, f(c_a, c_{bn})]$ is a similarity function (*e.g.* Cosine Distance). The optimization function is the same form of Equation 2, where the first term is the numerator of SR and the second term is the denominator). λ is seen as a hyper-parameter, whose meaning is equivalent to $-\bar{\beta}_2/\bar{\beta}_1$ in Section 3.2. K is also a hyper-parameter. For simplicity we may assume $\lambda \geq 0$, as when $\lambda < 0$ the two terms of optimization function 3 has a strong positive correlation, experiment results show there is not much improvement compared with directly setting $\lambda = 0$. $|U| = m$ is the cardinality constraint that exact m base classes are needed to be selected.

The next corollary shows that Problem 3 is equivalent to a submodular optimization.

Corollary 4.1. *Considering optimization problem 3, when $\lambda = 0$, Problem 3 is equivalent to a submodular monotone non-decreasing optimization with exact cardinality constraint and when $\lambda > 0$, Problem 3 is equivalent to a submodular optimization with exact cardinality constraint.*

4.3. Optimization

4.3.1 Case 1: $\lambda = 0$

The case $\lambda = 0$ could be seen as a standard submodular monotone non-decreasing optimization, hence we could directly use a greedy method on the value of target function, as Algorithm 2 shows. However, for this specific target function, a trivial setting with $m \geq K \cdot |N|$ needs further consideration. For this setting, a greedy algorithm on novel class (Algorithm 1) could be proved to reach an optimal solution, while Algorithm 2 could just reach sub-optimal. Thus, the two different greedy algorithms are proposed to deal with the trivial and non-trivial case separately. For our description of the algorithms below, $f(\cdot, \cdot)$ denotes for the similarity function and $h(\cdot)$ denotes for the optimization function of Problem 3 with $\lambda = 0$.

Algorithm 1 Greedy Algorithm on Novel Class (f, m)

Let $U_0 \leftarrow \emptyset$, $S \leftarrow N$
for $i = 1$ to m **do**
  Let $u \in B_u \backslash U_{i-1}$, $n \in S$ be the samples maximizing $f(c_u, c_n)$.
  Let $U_i \leftarrow U_{i-1} + u$, $S \leftarrow S - n$.
  if $S = \emptyset$ **then**
    $S \leftarrow N$.
  end if
end for
return U_m

Algorithm 2 Greedy Algorithm on Target Function (h, m)

Let $U_0 \leftarrow \emptyset$
for $i = 1$ to m **do**
  Let $u_i \in B_u \backslash U_{i-1}$ maximizing $h(u_i|U_{i-1})$.
  Let $U_i \leftarrow U_{i-1} + u_i$.
end for
return U_m

We further give Thm. 1, 2 to show the optimization bound of the two algorithms. For this specific problem, the bounds are much tighter than the generic version in [15].

Theorem 1. *For $B_s = \emptyset$ and $\lambda = 0$, when $m \geq K \cdot |N|$, using Algorithm 1 to solve for optimization problem 3, the solution will be optimal.*

Theorem 2. *For $B_s = \emptyset$ and $\lambda = 0$, using Algorithm 2 to solve for optimization problem 3, let $h(\cdot)$ be the optimization function, and let Q be*

$$Q = \mathbb{E}_{u\sim Uniform(B), v\sim Uniform(N)}(f(c_u, c_v))$$

representing for the average similarity between base classes and novel classes, we have $h(U) \geq (1 - 1/e) \cdot h(OPT) + 1/e \cdot Q$, where $h(OPT)$ is the global optimal value of the optimization problem.

4.3.2 Case 2: $\lambda > 0$

The case $\lambda > 0$ could be seen as a non-monotone submodular optimization, with the technique in [2], we combine both Random Greedy Algorithm (Algorithm 3) and Continuous Double Greedy Algorithm (Algorithm 4) for better optimization. The Random Greedy Algorithm is an extension of the standard Greedy Algorithm (Algorithm 2), which is fit for settings with extremely low m. Details of the algorithm are given in Algorithm 3.

Algorithm 3 Random Greedy Algorithm (h, m)

Let $U_0 \leftarrow \emptyset$
for $i = 1$ to m **do**
  Let $M_i \subset B_u \backslash U_{i-1}$ be a subset of size m maximizing $\sum_{u\in M_i} h(u|U_{i-1})$.
  Let u_i be a uniformly random sample from M_i.
  Let $U_i \leftarrow U_{i-1} + u_i$.
end for
return U_m

For much larger m, we will introduce the Continuous Double Greedy Algorithm. The core idea is to convert the discrete optimization of Problem 3 to a continuous version.

Let $F(x)$ be the multilinear extension of the optimization function $h(\cdot)$ as:

$$F(x) = \sum_{S\subseteq B_u} h(S) \prod_{u\in S} x_u \prod_{u\notin S}(1 - x_u) \tag{4}$$

where $x \in [0, 1]^{|B_u|}$. Given a vector x, $F(x)$ represents for the expectation of function h given a random subset of B_u with every element $u \in B_u$ *i.i.d.* sampled with probability x_u . For two vectors x and y, define $x \vee y$ and $x \wedge y$ to be coordinate-wise maximum and minimum separately, *i.e.* $(x \vee y)_u = \max(x_u, y_u)$ and $(x \wedge y)_u = \min(x_u, y_u)$. An important property for multilinear form function F is:

$$\frac{\partial F(x)}{\partial x_u} = F(x \vee u) - F(x \wedge (B_u - u)) \tag{5}$$

For simplicity, in this part, notation for a subset could also be represented as a 0-1 vector where the corresponding elements belonging to the subset are 1 and otherwise 0, consistent with [2]. In the double continuous greedy algorithm, we don't need to calculate the exact value for $F(x)$, the only difficulty is to calculate $F(x\vee u) - F(x\wedge(B_u - u))$. Theorem 3 gives a dynamic programming for fast calculation.

Theorem 3. *Let $S \subseteq B_u$ be a random set, with each element v in B_u i.i.d. sampled with probability $(x \wedge (B_u - u))_v$. For each novel class $n \in N$, sort the similarity function $f(c_n, c_b)$ for each base class $b \in B = B_u \cup B_s$ in descent order, denoting as $q_{n,[1]}, q_{n,[2]}, \cdots q_{n,[|B|]}$, also, sort the similarity function for every base class in $S \cup B_s$ in descent order, denoting as $s_{n,[1]}, s_{n,[2]}, \cdots s_{n,[|S|+|B_s|]}$, then we have:*

$$\begin{aligned} &F(x \vee u) - F(x \wedge (B_u - u)) \\ &= \frac{1}{|N|\cdot K} \sum_{n\in N} \sum_{i=1}^{|B|} P(s_{n,[K]} = q_{n,[i]}) \max(f(c_n, c_u) - q_{n,[i]}, 0) \\ &- \lambda \cdot \frac{1}{|N| \cdot m} \sum_{n\in N} f(c_n, c_u) \end{aligned} \tag{6}$$

The probability term $P(s_{n,[K]} = q_{n,[i]})$ for $n \in N$ is defined over all random subsets S, where $s_{n,[K]}$ could be seen as a random variable. This probability term could be solved using dynamic programming in $O(K \cdot |B| \cdot |N|)$ time complexity by the following recursion equations:

$$\begin{cases} P(s_{n,[j]} \geq q_{n,[i]}) = (1 - x_{[i]}) \cdot P(s_{n,[j]} \geq q_{n,[i-1]}) \\ \qquad + x_{[i]} \cdot P(s_{n,[j-1]} \geq q_{n,[i-1]}) \ for\ [i] \in B_u \\ P(s_{n,[j]} \geq q_{n,[i]}) = P(s_{n,[j-1]} \geq q_{n,[i-1]}) \ for\ [i] \in B_s \end{cases} \tag{7}$$

$$P(s_{n,[j]} = q_{n,[i]}) = P(s_{n,[j]} \geq q_{n,[i]}) - P(s_{n,[j]} \geq q_{n,[i-1]})$$

where j runs for $1 \cdots K$ and i runs for $1 \cdots |B|$. [1]

Algorithm 4 shows the complete process of the Continuous Double Greedy Algorithm. The algorithm first uses a gradient-based method to optimize the surrogate multilinear extension of the submodular target function and returns a sub-optimal continuous vector x, which represents for the probability each element is selected. Then, certain rounding technique such as Pipage Rounding [3, 26] is used to transform the resulting fractional solution into an integral solution. [2]

A similar optimization bound analysis of Algorithm 3 and Algorithm 4 is given in Theorem 4.

Theorem 4. *For $B_s = \emptyset$ and $\lambda > 0$, using a combination of Algorithm 3 and 4 to solve for optimization problem 3 with $\lambda > 0$, h and Q are defined same as Theorem 2, we have*

$$\begin{aligned} \mathbb{E}(h(U)) \geq max\ (&\frac{1 - m/er}{e} \cdot h(OPT) + C_1 \cdot Q, \\ &(1 + \frac{r}{2\sqrt{(r-m)m}})^{-1} \cdot h(OPT) + C_2 \cdot Q) \end{aligned}$$

For $0 < \lambda < \frac{1}{e-1}$, we have $C_1 = \frac{1}{e} + (1 - \frac{1}{e})\frac{m}{r} - (1 - \frac{1}{e}) \cdot \lambda > 0$ and $C_2 = \frac{(1-\lambda)r}{2\sqrt{(r-m)m}+r} - \epsilon \geq \frac{1}{2}(1 - \lambda) > 0$, where $r = |B_u|$. The first term is the lower bound for Algorithm 3 and the second term for Algorithm 4.

[1] Details are shown in Appendix 2.2 and 2.3.
[2] See Appendix 2.4.

Algorithm 4 Continuous Double Greedy Algorithm (F, m)

Initialize: $x^0 \leftarrow \emptyset$, $y^0 \leftarrow B_u$
for time step $t \in [1, T]$ **do**
  for every $u \in B_u$ **do**
    Let $a_u \leftarrow \frac{\partial F(x^{t-1})}{\partial x_u}$, $b_u \leftarrow \frac{\partial F(y^{t-1})}{\partial y_u}$ by Eq. 6, 7.
    Let $a'_u(l) \leftarrow max(a_u - l, 0)$,$b'_u(l) \leftarrow max(b_u + l, 0)$
    Let $\frac{dx_u}{dt}(l, t-1) \leftarrow \frac{a'_u}{a'_u + b'_u}$, $\frac{dy_u}{dt}(l, t-1) \leftarrow -\frac{b'_u}{a'_u + b'_u}$.
  end for
  Find l^* satisfying $\sum_{u \in B_u} \frac{dx_u}{dt}(l^*, t-1) = m$.
  Do a step of Gradient Ascent for x and Gradient Descent for y: $x_u^t = x_u^{t-1} + \frac{1}{T} \cdot \frac{dx_u}{dt}(l^*, t-1)$, $y_u^t = y_u^{t-1} - \frac{1}{T} \cdot \frac{dy_u}{dt}(l^*, t-1)$.
end for
Process certain rounding technique using x^T to get U.
return U

Theorem 4 indicates that if neglecting the term with Q, when $m < 0.08r$ or $m > 0.92r$, we should use Algorithm 3 and otherwise Algorithm 4 by comparing two bounds.

As a conclusion of this section, we list the applicability of different algorithms for this specific problem in Table 1.

5. Experiments

5.1. Experimental Settings

Basically, we design three different settings to show the superiority of our proposed algorithm:

Pre-trained Selection A pre-trained model is given, and the base classes selection could be conducted with the help of the pre-trained model. Generally we could use the pre-trained model to extract image representations. The setting also supposes that we know about the novel support set. In this paper, we evaluate the generalized performance only via the base model trained on the selected base classes, while in practice we could also use these selected base classes to further fine-tune the given pre-trained model.

Cold Start Selection No pre-trained model is given, hence the base classes selection is conducted in an incremental manner. For each turn, the selection of the incremental base classes is based on the trained base model from the previous turn. The novel support set is also given. Note that the setting is somewhat like a curriculum learning [1].

General Selection The novel support set is not known beforehand (*i.e.* Select a general base dataset that performs well on **any** composition of novel classes). In this paper for simplicity, we also suppose a pre-trained model is given as in the Pre-trained Selection setting.

In our experiments, we use two datasets for validating general classification: ImageNet and Caltech256, and one for fine-grained classification: CUB-200-2011. For ImageNet, we use the other 500 classes in addition to those used in the preliminary experiment in Section 3, which are further split into 400 candidate base classes and 100 novel classes. For all three tasks, the base dataset is selected from these 400 candidate base classes, and further evaluate the generalization performance on the 100 novel ImageNet classes, Caltech256 and CUB-200-2011.

For all experiments, we train a standard ResNet-18 [10] backbone as the base model on the selected base classes. For few-shot learning task on novel classes, we use two different heads: one is the cosine similarity on the representation space (512-dimensional features after conv5_x layer), which is a simplified version of Matching Network [25] without meta training step, representing the branch of metric-based approaches in few-shot learning. The other is the softmax regression on the representation space, which is a simple method from the branch of learning-based approaches. [3] We use different heads to show our proposed selection method is model-agnostic.

As for the details of the experiment, we use an active learning manner as mentioned in Section 1. Each candidate base class only contains 50 images before selected. We utilize these images to calculate class representation. When a base class is selected, the number of training images for this class could be expanded to a relatively abundant number (For this experiment all training images of this class in ImageNet are used, which locates at the interval from about 800 to 1,300). We allow for a slight difference in the number of images per class to simulate a practical scenario. For a p-way k-shot setting, we randomly select p novel classes and then choose k samples per novel class as the novel support set; another 100, 50, 40 samples disjoint with the support set per novel class as the novel testing set for ImageNet, Caltech and CUB-200-2011. The flow of the experiment is to run selection algorithms, expand the selected classes, train a base model on the expanded base dataset and evaluate performance on testing set. The process is repeated for 10 times with different randomization, and we report the average Top-1 accuracy for each experiment setting. For settings containing pre-trained model, in this paper we use ResNet-18 trained on full training images from randomly selected 100 classes extracted from the candidate base classes in Section 3, which is disjoint with the base and novel classes used in this section. We also emphasize that when comparing with different methods within the same setting, the same novel support set and novel testing set are used for each turn of the experiment for a fair comparison.

We consider three baselines in our experiments: the first is the Random Selection, which draws the base classes uniformly, which is a rather simple baseline but common in the real scenario, the second is using the Domain Similarity metric which is generally used in [17, 20, 21]. The idea is to maximize a pre-defined domain similarity between rep-

[3] The result of softmax regression head is shown in 4.

Table 1. Conclusion of Applicability of Different Algorithms

Parameter	Algorithm	Applicability	Complexity
$\lambda = 0$	Greedy on Novel Class	$m > \gamma \cdot K \cdot \lvert N \rvert$, with γ slightly larger than 1	$O(\lvert B \rvert \cdot log \lvert B \rvert \cdot \lvert N \rvert)$
$\lambda = 0$	Greedy on Target Function	$m < \gamma \cdot K \cdot \lvert N \rvert$, with γ slightly larger than 1	$O(m \cdot (\lvert B \rvert + \lvert N \rvert \cdot logK))$
$\lambda > 0$	Random Greedy	$m < 0.08 \cdot \lvert B_u \rvert$ or $m > 0.92 \cdot \lvert B_u \rvert$	$O(m \cdot (\lvert B \rvert \cdot logm + \lvert N \rvert \cdot logK))$
$\lambda > 0$	Continuous Double Greedy	$0.08 \cdot \lvert B_u \rvert < m < 0.92 \cdot \lvert B_u \rvert$	$O(T \cdot K \cdot \lvert B \rvert^2 \cdot \lvert N \rvert)$

resentation for each selected element in the source domain and the representation for the target domain. The method is first proposed for sample selection, and in this paper we extend to the class selection by viewing the centroid of features for a class as a sample and viewing the centroid of the novel support set as representation for the target domain. The baseline will be used in Pre-trained Selection and Cold Start Selection. The third is the K-medoids algorithm [16], which is a clustering algorithm as a baseline of the General Selection setting. For all baselines and our algorithm, cosine similarity on representation space is used for calculating the similarity of two representations.

5.2. Results

5.2.1 Pre-trained Selection

Table 2, 3, 4 show the results of the Pre-trained Selection. When setting $K = 1$, the algorithm reaches the best performance in all cases. For the ImageNet dataset in Table 2, we show that Algorithm 1 and Algorithm 2 are fit for different cases, depending on the number of selected classes, as Table 1 describes. For $m = 100$ and $m = 20$ case, our algorithm obtains a superior accuracy of about 4% and 2% separately compared with random selection, which is a relatively huge promotion in few-shot image classification. Besides, the promotion is rather stable concerning the shot number. The Domain Similarity algorithm performs worse because of the cluster effect, where the selected base classes are concentrated around the centroid of the target domain, in contrast with the idea of enhancing diversity we show in Section 3. For Caltech256 as novel classes in Table 3, a transfer distribution on dataset is introduced. It shows that in such case, the improved margin compared to random selection is much larger, reaching about 10% when $m = 100$. This is because our algorithm enjoys the double advantages of transfer effect and class selection effect; the former also promotes the Domain Similarity algorithm. For the CUB-200-2011 dataset in Table 4, we further show that our algorithm improves the margin much more significantly in a fine-grained manner, reaching about 11.2% for 5-shot setting and 13.6% for 20-shot setting.

5.2.2 Cold Start Selection

The Cold Start Selection is more difficult than the Pre-trained Selection in that there is no pre-trained model at the early stage, leading to an unknown or imprecise image representation space. Hence the representation space needs to be learned incrementally. For each turn, the selection of the incremental base classes is based on the trained base model from the previous turn. Noticing that in this incremental learning manner both the complexity and the effectiveness of selection should be considered. To limit the complexity we increasingly select the same number of classes in each turn as the total number of selected base classes in the previous turn (*i.e.* doubling the number of selected classes in each turn). This double-increasing mechanism could guarantee a linear time complexity of m in training the base model. For example, in Table 5 a 6-12-25-50-100 mechanism represents for selecting 6 classes randomly in Turn 1, and continue selecting another 6 classes based on the model trained by classes from Turn 1 to form a selection of 12 classes in Turn 2 and so on. As the representation space is not so stable as the Pre-trained Selection, a larger K with $K = 3$, $\lambda = 0$ is much better. Table 5 shows the result of the algorithms. Our proposed method exhibits a 2.8% promotion compared to random selection. We also highlight that the upper bound of the algorithm is limited by the Pre-trained selection (with a pre-trained model on 100 classes with $K = 3$), which is 42.89%. By using the double-increasing mechanism, the performance is just slightly lower than this upper bound in linear time complexity.

We also show some ablation studies by changing the selection of K and the selection mechanism. As for the selection mechanism, comparing 6-12-25-50-100 and 50-100, we draw a conclusion that the incremental learning of the representation space is much more effective, and compared to 10-20-40-80-100 it shows that the selection in the early stage of Cold Start Selection is more important than the later stage.

5.2.3 General Selection

General Selection is the most difficult setting in this paper, as we do not know the novel classes previously. The goal is to select a base dataset that could perform well on any composition of novel classes. In dealing with this problem, we make a slight change to our optimization framework that **we take all candidate base classes as the novel classes**. The implicit assumption is that the candidate base classes represent for a subsample of the global world categories. In this setting, we should choose a much larger K and λ for this setting, especially for fine-grained classification, to enhance representativeness and diversity for each selected class.

Table 2. ImageNet: Pre-trained Selection, 100-way novel classes

Algorithm	m=100, 5-shot	m=100, 20-shot	m=20, 5-shot	m=20, 20-shot
Random	$39.39\% \pm 0.82\%$	$49.47\% \pm 0.67\%$	$23.89\% \pm 0.56\%$	$33.06\% \pm 0.47\%$
DomSim	$38.00\% \pm 0.36\%$	$48.80\% \pm 0.79\%$	$23.15\% \pm 0.43\%$	$31.81\% \pm 0.58\%$
Alg. 1, $K = 1, \lambda = 0$	$\mathbf{43.42\% \pm 0.78\%}$	$\mathbf{53.79\% \pm 0.37\%}$	$25.71\% \pm 0.43\%$	$34.67\% \pm 0.36\%$
Alg. 2, $K = 1, \lambda = 0$	$43.20\% \pm 0.76\%$	$53.61\% \pm 0.27\%$	$\mathbf{26.13\% \pm 0.44\%}$	$\mathbf{34.97\% \pm 0.45\%}$
Alg. 2, $K = 3, \lambda = 0$	$42.89\% \pm 0.43\%$	$53.13\% \pm 0.27\%$	$25.10\% \pm 0.48\%$	$34.52\% \pm 0.51\%$

Table 3. Caltech256: Pre-trained Selection, 100-way

Algorithm	m=100, 5-shot	m=100, 20-shot
Random	$45.31\% \pm 1.32\%$	$54.97\% \pm 1.23\%$
DomSim	$49.55\% \pm 1.28\%$	$58.84\% \pm 1.01\%$
Alg. 1, $K = 1, \lambda = 0$	$\mathbf{55.41\% \pm 1.25\%}$	$\mathbf{64.46\% \pm 0.99\%}$
Alg. 2, $K = 3, \lambda = 0$	$54.94\% \pm 1.14\%$	$63.58\% \pm 0.98\%$

Table 4. CUB-200-2011: Pre-trained Selection, 100-way

Algorithm	m=100, 5-shot	m=100, 20-shot
Random	$18.46\% \pm 1.19\%$	$26.14\% \pm 1.44\%$
DomSim	$28.11\% \pm 0.44\%$	$38.26\% \pm 0.45\%$
Alg. 1, $K = 1, \lambda = 0$	$\mathbf{29.65\% \pm 0.82\%}$	$\mathbf{39.77\% \pm 0.41\%}$
Alg. 2, $K = 3, \lambda = 0$	$28.04\% \pm 1.82\%$	$37.22\% \pm 0.69\%$

Table 5. ImageNet: Cold Start Selection, 100-way

Algorithm	Mechanism	Top-1 Accuracy
Random	-	$39.39\% \pm 0.82\%$
DomSim	6-12-25-50-100	$39.30\% \pm 0.40\%$
Alg. 1, $K = 1, \lambda = 0$	6-12-25-50-100	$40.96\% \pm 0.50\%$
Alg. 2, $K = 1, \lambda = 0$	6-12-25-50-100	$41.75\% \pm 0.59\%$
Alg. 2, $K = 3, \lambda = 0$	6-12-25-50-100	$\mathbf{42.17\% \pm 0.67\%}$
Alg. 2, $K = 5, \lambda = 0$	6-12-25-50-100	$41.33\% \pm 0.36\%$
Alg. 2, $K = 3, \lambda = 0$	10-20-40-80-100	$41.61\% \pm 0.76\%$
Alg. 2, $K = 3, \lambda = 0$	50-100	$40.88\% \pm 0.66\%$
Pre-trained (Upperbound)	[100]-100	$42.89\% \pm 0.43\%$

Results of ImageNet and Caltech256 (Table 6, 7) show that our algorithms perform much better when the number of selected classes is larger. Specifically, in $m = 100$ case we promote 0.9% and 4.5% in two datasets separately compared with random selection, however in $m = 20$ case the promotion is not so obvious, only 0.3% and 0.9%, which shows that a larger base dataset may contain more general image information. As for the result of CUB-200-2011 (Table 8), our proposed algorithm performs much better due to the effect of diversity, reaching an increase of 6.4% in $m = 100$ case. Besides, the result also shows that the performance reaches the best with a positive λ in fine-grained classification, illustrating the necessity of diversity (According to Table 1, we choose Algorithm 3 for $m = 20$ and Algorithm 4 for $m = 100$). The results also show that the baseline K-Medoids is rather unstable in different cases. It may reach the state-of-the-art in some cases but may perform even worse than random in other cases.

Table 6. ImageNet: General Selection, 100-way

Algorithm	m=20, 20-shot	m=100, 20-shot
Random	$33.06\% \pm 0.47\%$	$49.47\% \pm 0.67\%$
K-Medoids	$\mathbf{33.50\% \pm 0.28\%}$	$49.17\% \pm 0.38\%$
Alg. 2, $K = 3, \lambda = 0$	$33.38\% \pm 0.25\%$	$50.00\% \pm 0.38\%$
Alg. 2, $K = 5, \lambda = 0$	$33.32\% \pm 0.30\%$	$\mathbf{50.35\% \pm 0.29\%}$
Alg. 2, $K = 10, \lambda = 0$	$33.01\% \pm 0.38\%$	$50.21\% \pm 0.26\%$
Alg. 3/4, $K = 5, \lambda = 0.2$	$32.82\% \pm 0.35\%$	$49.19\% \pm 0.34\%$

Table 7. Caltech256: General Selection, 100-way

Algorithm	m=20, 20-shot	m=100, 20-shot
Random	$40.26\% \pm 0.90\%$	$54.97\% \pm 1.23\%$
K-Medoids	$40.16\% \pm 0.83\%$	$59.27\% \pm 1.01\%$
Alg. 2, $K = 3, \lambda = 0$	$40.72\% \pm 0.92\%$	$59.23\% \pm 0.94\%$
Alg. 2, $K = 5, \lambda = 0$	$40.98\% \pm 0.84\%$	$58.68\% \pm 0.94\%$
Alg. 2, $K = 10, \lambda = 0$	$\mathbf{41.18\% \pm 0.88\%}$	$\mathbf{59.52\% \pm 0.91\%}$
Alg. 3/4, $K = 5, \lambda = 0.2$	$40.31\% \pm 1.24\%$	$57.79\% \pm 0.94\%$

Table 8. CUB-200-2011: General Selection, 100-way

Algorithm	m=20, 20-shot	m=100, 20-shot
Random	$15.25\% \pm 0.91\%$	$26.14\% \pm 1.44\%$
K-Medoids	$14.96\% \pm 0.47\%$	$24.38\% \pm 0.59\%$
Alg. 2, $K = 3, \lambda = 0$	$14.74\% \pm 0.48\%$	$27.08\% \pm 0.52\%$
Alg. 2, $K = 5, \lambda = 0$	$16.06\% \pm 0.59\%$	$28.33\% \pm 0.57\%$
Alg. 2, $K = 10, \lambda = 0$	$16.21\% \pm 0.33\%$	$27.63\% \pm 0.66\%$
Alg. 3/4, $K = 5, \lambda = 0.2$	$16.61\% \pm 0.36\%$	$\mathbf{32.50\% \pm 0.58\%}$
Alg. 3/4, $K = 5, \lambda = 0.5$	$\mathbf{17.09\% \pm 0.33\%}$	$31.01\% \pm 0.58\%$

6. Conclusions

This paper focuses on how to construct a high-quality base dataset with limited number of classes from a wide broad of candidates. We propose the Similarity Ratio as a proxy of the performance of few-shot learning and further formulate the base class selection problem as an optimization process over Similarity Ratio. Further experiments in different scenarios show that the proposed algorithm is superior to random selection and some typical baselines in selecting a better base dataset, which shows that, besides advanced few-shot algorithms, a reasonable selection of base dataset is also highly desired in few-shot learning.

7. Acknowledgement

This work was supported in part by National Key R&D Program of China (No. 2018AAA0102004), National Natural Science Foundation of China (No. U1936219, No. 61772304, No. U1611461), Beijing Academy of Artificial Intelligence (BAAI).

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, 2009.

[2] Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1433–1452. Society for Industrial and Applied Mathematics, 2014.

[3] Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[5] Yang Fan, Fei Tian, Tao Qin, Jiang Bian, and Tie-Yan Liu. Learning what data to learn. *arXiv preprint arXiv:1702.08635*, 2017.

[6] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[8] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

[9] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] Jedrzej Kozerawski and Matthew Turk. Clear: Cumulative learning for one-shot one-class image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3446–3455, 2018.

[12] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

[13] Erik G Miller, Nicholas E Matsakis, and Paul A Viola. Learning from one example through shared densities on transforms. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 464–471. IEEE, 2000.

[14] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

[15] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[16] Hae-Sang Park and Chi-Hyuck Jun. A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2-part-P2):3336–3341.

[17] Barbara Plank and Gertjan Van Noord. Effective measures of domain similarity for parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1566–1576. Association for Computational Linguistics, 2011.

[18] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5822–5830, 2018.

[19] Chen Qu, Feng Ji, Minghui Qiu, Liu Yang, Zhiyu Min, Haiqing Chen, Jun Huang, and W Bruce Croft. Learning to selectively transfer: Reinforced transfer learning for deep text matching. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 699–707. ACM, 2019.

[20] Robert Remus. Domain adaptation using domain similarity- and domain complexity-based instance selection for cross-domain sentiment analysis. In *2012 IEEE 12th international conference on data mining workshops*, pages 717–723. IEEE, 2012.

[21] Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with bayesian optimization. *arXiv preprint arXiv:1707.05246*, 2017.

[22] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pages 4077–4087, 2017.

[23] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[24] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. Learning the curriculum with bayesian optimization for task-specific word representation learning. *arXiv preprint arXiv:1605.03852*, 2016.

[25] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[26] Jan Vondrák. Symmetry and approximability of submodular maximization problems. *SIAM Journal on Computing*, 42(1):265–304, 2013.

[27] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[28] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7278–7286, 2018.

[29] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning.

In *European Conference on Computer Vision*, pages 616–634. Springer, 2016.

[30] Linjun Zhou, Peng Cui, Shiqiang Yang, Wenwu Zhu, and Qi Tian. Learning to learn image classifiers with visual analogy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11497–11506, 2019.