A Bag-of-Prototypes Representation for Dataset-Level Applications

Weijie Tu\textsuperscript{1} \quad Weijian Deng\textsuperscript{1} \quad Tom Gedeon\textsuperscript{2} \quad Liang Zheng\textsuperscript{1}
\textsuperscript{1}Australian National University \quad \textsuperscript{2}Curtin University

Abstract

This work investigates dataset vectorization for two dataset-level tasks: assessing training set suitability and test set difficulty. The former measures how suitable a training set is for a target domain, while the latter studies how challenging a test set is for a learned model. Central to the two tasks is measuring the underlying relationship between datasets. This needs a desirable dataset vectorization scheme, which should preserve as much discriminative dataset information as possible so that the distance between the resulting dataset vectors can reflect dataset-to-dataset similarity. To this end, we propose a bag-of-prototypes (BoP) dataset representation that extends the image-level bag consisting of patch descriptors to dataset-level bag consisting of semantic prototypes. Specifically, we develop a codebook consisting of K prototypes clustered from a reference dataset. Given a dataset to be encoded, we quantize each of its image features to a certain prototype in the codebook and obtain a K-dimensional histogram. Without assuming access to dataset labels, the BoP representation provides rich characterization of the dataset semantic distribution. Furthermore, BoP representations cooperate well with Jensen-Shannon divergence for measuring dataset-to-dataset similarity. Although very simple, BoP consistently shows its advantage over existing representations on a series of benchmarks for two dataset-level tasks.

1. Introduction

Datasets are fundamental in machine learning research, forming the basis of model training and testing [18, 51, 52, 61]. While large-scale datasets bring opportunities in algorithm design, there lack proper tools to analyze and make the best use of them [6, 51, 56]. Therefore, as opposed to traditional algorithm-centric research where improving models is of primary interest, the community has seen a growing interest in understanding and analyzing the data used for developing models [51, 56]. Recent examples of such goal include data synthesis [29], data sculpting [25, 51], and data valuation [6, 32, 56]. These tasks typically focus on individual sample of a dataset. In this work, we aim to understand nature of datasets from a dataset-level perspective.

This work considers two dataset-level tasks: suitability in training and difficulty in testing. First, training set suitability denotes whether a training set is suitable for training models for a target dataset. In real-world applications, we are often provided with multiple training sets from various data distributions (e.g., universities and hospitals). Due to distribution shift, their trained models have different performance on the target dataset. Then, it is of high practical value to select the most suitable training set for the target dataset. Second, test set difficulty means how challenging a test set is for a learned model. In practice, test sets are usually unlabeled and often come from different distributions than that of the training set. Measuring the test set difficulty for a learned model helps us understand the model reliability, thereby ensuring safe model deployment.

The core of the two dataset-level tasks is to measure the relationship between datasets. For example, a training set is more suitable for learning a model if it is more similar to the target dataset. To this end, we propose a vectorization scheme to represent a dataset. Then, the relationship between a pair of datasets can be simply reflected by the distance between their representations. Yet, it is challenging to encode a dataset as a representative vector, because (i) a dataset has a different cardinality (number of images) and (ii) each image has its own semantic content (e.g., category). It is thus critical to find an effective way to aggregate all image features to uncover dataset semantic distributions.

In the literature, some researchers use the first few moments of distributions such as feature mean and co-variance to represent datasets [20, 62, 74, 75, 82]. While being computationally friendly, these methods do not offer sufficiently strong descriptive ability of a dataset, such as class distributions, and thus have limited effectiveness in assessing attributes related to semantics. There are also some methods learn task-specific dataset representations [1, 63]. For example, given a dataset with labels and a task loss function, Task2Vec [1] computes an embedding based on estimates of the Fisher information matrix associated with a probe network’s parameters. While these task-specific representations are able to predict task similarities, they are not suitable for characterizing dataset properties of interest. They
require training a network on the specific task [1] or on multiple datasets [63], so they are not effective in assessing the training set suitability. Additionally, they require image labels for the specific task, so they cannot be used to measure the difficulty of unlabeled test sets.

In this work, we propose a simple and effective bag-of-prototypes (BoP) dataset representation. Its computation starts with partitioning the image feature space into semantic regions through clustering, where the region centers, or prototypes, form a codebook. Given a new dataset, we quantize its features to their corresponding prototypes and compute an assignment histogram, which, after normalization, gives the BoP representation. The dimensionality of BoP equals the codebook size, which is usually a few hundred and is considered memory-efficient. Meanwhile, the histogram computed on the prototypes is descriptive of the dataset semantic distribution.

Apart from being low dimensional and semantically rich, BoP has a few other advantages. First, while recent works in task-specific dataset representation usually require full image annotations and additional learning procedure [1,63], the computation of BoP does not rely on any. It is relatively efficient and allows for unsupervised assessment of dataset attributes. Second, BoP supports dataset-to-dataset similarity measurement through Jensen-Shannon divergence. We show in our experiment that this similarity is superior to commonly used metrics such as Fréchet distance [27] and maximum mean discrepancy [33] in two dataset-level tasks.

2. Related Work

Dataset representations. A common practice is to use simple and generic statistics as dataset representations [20,62,74,75,82]. For example, Peng et al. [62] use the first moment to represent a dataset. Deng et al. [20] use global feature mean and co-variance as dataset representations. Vanschoren et al. [82] find dataset cardinality (the number of images/classes) useful to encode a dataset. These methods have limited descriptive ability, whereas BoP is more semantically descriptive. Moreover, it is feasible to learn a task-specific dataset representation [1,63,84,87]. For example, Ying et al. [84] learn transfer skills from previous transfer learning experiences for future target tasks. Achille et al. [1] propose to learn a task embedding based on the estimate of Fisher information matrix associated with a task loss. Compared with these task-specific representations, BoP is hand-crafted, avoiding computation overheads incurred by end-to-end learning. It is thus efficient in measuring training set suitability without training any models. Moreover, BoP require no image labels, making it more suitable for assessing the difficulty of unlabeled test sets.

Dataset-to-dataset similarity. We briefly review three strategies. First, some dataset similarity measures are developed in the context of domain adaptation [2,9,10,85]. They typically depend on a loss function and hypothesis class, and use a supremum of that function class to quantify the similarity of datasets. (e.g., $\mathcal{H}\Delta\mathcal{H}$-divergence [9], $f$-divergence [2], and $A$-distance [10]). Second, dataset distance can be computed based on optimal transport [5,17,79]. For example, the squared Wasserstein metric Fréchet distance [27] is widely used in comparing the distribution discrepancy of generated images with the distribution of real images [39]. To better leverage the geometric relationship between datasets, Alvarez et al. [5] use labels to guide optimal transport towards class-coherent matches. Third, existing dataset representations can be used to compute dataset distance [33,62,75,81]. For example, maximum mean discrepancy (MMD) [33] computes the distance between mean elements of distributions on the probability space. Peng et al. [62] eliminate dataset discrepancy by matching datasets moments. CORAL [75] uses the second-order statistics of datasets to measure distance. This work is in the third category and uses JS divergence between BoP representations to calculate dataset-to-dataset similarity.

Assessment of training dataset suitability. Recent works have focused on understanding the importance of individual training instances in training of neural networks [6,32,45,56]. For example, Data Shapley [32] and Consistency Score [45] are proposed to evaluate the value of each data instance. Some methods identify “difficult” instances based on the information of training dynamics [7,76,80].

Different from the above approaches, this work studies the suitability of an entire training set. Given multiple training datasets from different data distributions, the focus is to choose the most appropriate training dataset for the target domain. Dataset-to-dataset similarity can be used for this goal. Intuitively, if a training dataset has high similarity with a target dataset, the model trained on it is expected to be more performant and vice versa. In this work, we use BoP representation coupled with simple JS divergence to calculate dataset-to-dataset similarity and demonstrate its effectiveness in accessing training set suitability.

Assessing test set difficulty without ground truths. The goal of this task (also known as unsupervised accuracy estimation) is to predict the accuracy of a given model on various unlabeled test sets. Existing methods usually use a representation of the test set for accuracy prediction [13,19,20,30,34]. Normally this representation is derived from classifier outputs, such as image features [20], prediction logits [30], average softmax scores [34]. Then, regression is used to establish the relationship between this representation and model test accuracy under various testing environments. Compared with existing dataset features, the BoP representation better characterizes the semantic distribution of training and test sets and thus can be effectively used for model accuracy prediction.
3. Methodology

3.1. Bag-of-Words Model Across Communities

In natural language processing (NLP) and information retrieval, the Bag-of-Words (BoW) model [46, 47, 50, 57] vectorizes textual data as a word histogram. Specifically, for each word in the dictionary, its occurrences in a document are counted, which fills in the corresponding entry of the BoW feature. This word frequency vector is thus used to represent a document. Numerous improvements of the BoW feature were made in NLP, such as n-grams [47, 50] and term frequency-inverse document frequency [68].

In the early 2000s, the BoW representation was introduced to the computer vision (CV) community to encode hundreds or thousands of local image descriptors [8, 53] into a compact vector [73]. As there is no semantic codebook like in NLP, a visual codebook is constructed by performing clustering (e.g., k-means) on a collection of local image features, where the resulting clustering centers are called “visual words”. Local image descriptors are quantized to their nearest cluster center so that a visual word histogram can be computed. This BoW histogram also has undergone extensive improvements in later years, such as Fisher vector [64, 65], vector of locally aggregated descriptors (VLAD) [43], and the use of principal component analysis and whitening [42].

Contribution statement. This paper contributes a baseline method in adopting the BoW idea study the two basic properties of a dataset. To this end, we propose to represent a dataset using its histogram over a series of prototypes. A comparison between the usage of BoW model in NLP, CV and our dataset-level research is shown in Table 1. Specifically, the BoP representation relies on clustering for codebook formation, has a relatively small codebook (depending on the richness of dataset semantics), and has semantically sensible codewords.

3.2. Bag-of-Prototypes Dataset Representation

Given a dataset \( D = \{x_i\}_{i=1}^N \) where \( N \) is the number of images and a feature extractor \( F(\cdot) \) that maps an input image into a \( d \)-dimensional feature \( f \in \mathbb{R}^d \), we extract a set of image features \( \mathcal{F} := \{F(x_i)\}_{i=1}^N \). While it is possible to directly use the dataset images (or features) as model input under small \( N \), it becomes prohibitively expensive when \( N \) is large. We therefore focus on extracting useful semantic features of \( \mathcal{F} \) by encoding its image features into a compact representation. Below we detail the necessary steps for computing the proposed BoP representation (refer Fig. 1).

Step I: Codebook generation. Given a reference dataset \( D_r = \{x'_i\}_{i=1}^{N_r} \), we extract all of its image features \( \mathcal{F}_r := \{F(x'_i)\}_{i=1}^{N_r} \) using a pretrained network, from which a codebook is constructed. Specifically, we adopt standard k-means clustering [54] to partition the feature space \( \mathbb{R}^d \) into \( K \) clusters. Each of the \( K \) cluster centers is called a “prototype”, because oftentimes each center mainly represents a certain semantic content. See Fig. 1 right for exemplar image of each prototype. The prototypes, or centers, constitute the codebook, denoted as \( C = \{c_i\}_{i=1}^K \), where \( c_i \) is the \( i \)-th prototype. Note that, the order of the prototypes is fixed in the codebook.

Step II: Histogram computation. For a dataset to be encoded \( D_e = \{x'_i\}_{i=1}^{N_e} \) where \( N_e \) is the number of images, we project it onto codebook \( C \) of size \( K \) to compute its BoP representation. Specifically, after extracting image features \( \mathcal{F}_e := \{F(x'_i)\}_{i=1}^{N_e} \) from \( D_e \), for each image feature, we compute its distance with all the \( K \) prototypes in the codebook, yielding \( K \) distances \( d_1, \ldots, d_k \), where \( d_i \) is the distance between an image feature and the \( i \)-th prototype. An image feature is quantized to prototype \( c_i \) if \( d_i \) is the lowest among \( d_1, \ldots, d_k \). Following the quantization, we generate a \( K \)-dimensional one-hot encoding where the \( i \)-th entry is 1 and all the others are 0. Having computed the one-hot vectors for all the image features, we sum them which is then normalized by \( N_e \), the number of images in \( D_e \). This gives the histogram representation \( h_e \), or BoP representation, for \( D_e \) where the \( i \)-th entry indicates the density of features in \( D_e \) belonging to prototype \( c_i \).

3.3. Measuring Dataset-to-Dataset Similarity

Similar to image / document retrieval where BoW vectors of instances are used for similarity comparison [14, 26, 59, 66, 73], this work uses the BoP representation to calculate dataset-to-dataset similarity. Specifically, given BoP representations \( h_x \) and \( h_y \) of two datasets \( D_x \) and \( D_y \), we simply define their similarity \( S_{xy} \) using Jensen-Shannon divergence (JS divergence), which is designed for histogram-based similarity measurement [16, 55].

Task-oriented similarity measure. We can build a universal codebook on a large-scale dataset following BoW model [14, 86]. By doing so, the resulting BoP representations are generic. We can also build a task-oriented code-
Figure 1. Workflow of BoP representation computation using CIFAR-10 [49] and one CIFAR-10 out-of-distribution (OOD) test set as an example. **Top:** We group image features of the reference dataset CIFAR-10 into 10 clusters, and the centers are called prototypes. The prototypes constitute the codebook of size 10. **Bottom left:** To encode the OOD test set, we project it onto the codebook by quantizing each image feature to its corresponding prototype. Lastly, we compute the histogram, i.e., BoP representation, of CIFAR-10 OOD test set. **Bottom right:** We regard dataset-to-dataset similarity as the Jensen-Shannon divergence between BoP histograms of CIFAR-10 OOD test set and reference dataset. With such similarity, we can measure the test set difficulty for the model trained on reference dataset.

book on a reference dataset from a specific task to consider more task-oriented information. The latter is more suitable for the two dataset-level tasks considered in this work. For the task of training set suitability assessment, we use the target dataset as the reference for codebook generation to fully consider its the semantic information. As a result, the JS divergence between BoP representations of the training set and the target dataset can well capture how a training set is similar to the target set. Similarly, for the task of test set difficulty assessment, we build codebook on the training set. This practice can effectively measure how an unlabeled test is similar to a given training set.

3.4. Discussion

**Working mechanism of BoP.** Codebook generation of BoP can be viewed as Centroidal Voronoi Tessellations [24]. Specifically, the prototypes (cluster centers) of codebook tessellate the feature space into Voronoi cells. Then, histogram computation approximates a probability distribution function in the same way as the non-parametric histogram [12, 28, 67]. That is, the BoP representation reflects the distribution of a dataset in the feature space.

As shown in Fig. 1, the prototypes of reference dataset tessellate feature space into Voronoi cells. Based on this, we quantify the histogram of the reference dataset to represent its distribution. Given a new dataset, we conduct the same histogram calculation procedure and correspondingly capture its dataset distribution with the histogram. Then, we measure discrepancy of the two datasets by calculating JS divergence between their histograms. Compared with common measures of dataset distance (e.g., FD [27], KID [11] and MMD [33]) that only reflect global structure (e.g., first few moments) of dataset distributions, BoP, collaborated with JS divergence, considers more local structures.

**Training set suitability vs. transferability estimation.** Two tasks relate but differ significantly: 1) Given an unlabeled target dataset and a pool of training datasets, the former aims to select the most suitable training set for the target. The latter assumes a labeled target dataset and a pool of models pretrained on a source dataset, with the goal of selecting the most suitable source model for the target without fine-tuning them all [3, 4, 60]; 2) Datasets in training set suitability are used for the same classification problem. In contrast, in transferability estimation, the problem in the target dataset (e.g., CIFAR-10 classification) is different from that of the source dataset (e.g., ImageNet classification).

**Analysis of the number of prototypes in a codebook.** The codebook size is a critical factor influencing the usefulness of the BoP. A small codebook means a coarser partition of feature space, where similar features will likely be in the same cluster, but dissimilar features may also be in the same cluster. Moreover, a large codebook provides a finer description of the space, where dissimilar features are quantized to different prototypes and more semantics are explored. According to our experiment results in Fig. 2 and Fig. 5, we find, reassuringly, BoP is robust against the codebook size: prototype number can deviate within a wide range around the true classes number (e.g., 345 for Domain-
Application scope and future directions. BoP is proposed to study the two dataset-level tasks, and the datasets considered in each task share the same label space. We may encounter some situations where we need to compare datasets with different label spaces (e.g., pre-training datasets selection [1]). In this case, one potential way is to build a universal codebook on a large-scale and representative dataset similar to BoW models [14, 86]. By doing so, the resulting BoP representations can encode diverse and sufficient semantics for comparing datasets across various label spaces. We view our BoP as a starting point to encode datasets. It would be interesting to study other dataset vectorization schemes and dataset-level tasks.

4. Comparing Training Suitability of Datasets

This task studies dataset valuation where multiple training sets are provided by different data contributors. The goal is to select the most suitable training set (ideally without training) whose trained model performs the best on a target test set. In this section, we first validate that BoP, collaborated with JS divergence (BoP + JS), is predictive of dataset suitability for the target test set. Then, we show that BoP is robust when using a wide range of codebook sizes and different networks.

4.1. Experimental Settings

Correlation study under DomainNet setup. We use domain generalization benchmark DomainNet [62], which consists of 6 domains: Painting, Real, Infograph, Quick-draw, Sketch and ClipArt, where the tasks are 345-way object classification. Each domain has its training and test splits. We conduct the correlation study in an leave-one-out manner, leading to 6 groups of correlation studies, with each group using the test split of one domain as the target test set. Additionally, we apply image transformations to the training split of six original domains. Specifically, we employ ‘Cartoon’ [48], ‘Zoom Blur’ and ‘JPEG Compression’ [36] to convert domains’ style to be one specific type. We also use ‘AugMix’ [38] and ‘AutoAugment’ [15], which transform images with various operations to generate domains with mixed styles. This process synthesizes 30 new datasets, so we have 36 training sets in total.

We follow the training scheme provided by TLlib [44] to train ResNet-101 model [35], whose weights are pretrained on ImageNet [18], yielding 36 models. Moreover, penultimate outputs of pretrained ResNet-101 is used as image feature. On the test set, we generate a codebook of size 1000. Then, for each training set, we compute its BoP histogram, BoP + JS from the test set, and the accuracy of its trained model on the test set. After this, we calculate correlation strength between BoP + JS and model accuracy to evaluate whether BoP is indicative of datasets training suitability.

<table>
<thead>
<tr>
<th>Method</th>
<th>ResNet-34</th>
<th>ResNet-101</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r$</td>
<td>$\rho$</td>
</tr>
<tr>
<td>FD [27]</td>
<td>-0.860</td>
<td>-0.926</td>
</tr>
<tr>
<td>MMD [33]</td>
<td>-0.817</td>
<td>-0.801</td>
</tr>
<tr>
<td>KID [11]</td>
<td>-0.773</td>
<td>-0.904</td>
</tr>
<tr>
<td>BoP + JS</td>
<td>-0.960</td>
<td>-0.927</td>
</tr>
</tbody>
</table>

Table 2. Compare averaged Pearson’s correlation ($r$), Spearman’s correlation ($\rho$) and weighted Kendall’s correlation ($\tau_w$) of Fréchet distance (FD), maximum mean discrepancy (MMD), kernel inception distance (KID) and BoP + JS (codebook size 1000) on six test sets in DomainNet. We report two groups of results using ResNet-34 (Left) and ResNet-101 (Right). We show BoP + JS is more effective in assessing training set suitability than others.

Evaluation metric. We use Pearson’s correlation $r$ and Spearman’s rank correlation $\rho$ to show linearity and monotonicity between BoP-based dataset distance and model accuracy, respectively. Both metrics range in $[-1, 1]$. If $|r|$ or $|\rho|$ is close to 1, the linearity or monotonicity is strong, and vice versa. In addition to these two metrics, we also use weighted variant of Kendall’s correlation ($\tau_w$) [83]. It is shown to be useful when selecting the best ranked item is of interest [71], while a major application of BoP + JS is to select the training dataset leading to the best performance on a test set. This metric has the same range where a number closer to $-1$ or 1 indicates stronger negative or positive correlation, respectively, and 0 means no correlation.

4.2. Evaluation

Strong correlation: A training set is more suitable for a given test set if it has small BoP + JS. Fig. 2 shows correlation study on ClipArt, Painting, Real and Sketch. We notice that there are strong Pearson’s correlations ($|r| > 0.95$), Spearman’s rank correlations ($|\rho| > 0.93$) and relatively high weighted Kendall’s correlations ($|\tau_w| > 0.84$) on four test sets. This suggests that BoP + JS is stable and useful across test sets. Table 2 compares average correlation strength of BoP + JS with Fréchet distance (FD) [27], maximum mean discrepancy (MMD) [33] and kernel inception distance (KID) [11]. They use that same image features as BoP. According to their formulae, mean and covariance of these features are used for distance computation. We see that BoP + JS has the highest average correlation scores on six test sets ($|r| = 0.961$, $|\rho| = 0.929$ and $|\tau_w| = 0.840$). On average, BoP + JS is superior in depicting training sets suitability for a test set without any training procedure.

Impact of codebook size is shown in the Fig. 3. We construct codebooks with different size within approximately one order of magnitude around 345. We find that the three correlation scores increase and then become stable when codebook size becomes larger. This indicates that the per-
Correlation strength

This suggests that BoP + JS is predictive of the suitability of a training set.

5. Assessing Test Set Difficulty without Labels

In the task of test set difficulty assessment, we are provided with a labeled training set and a set of unlabeled datasets for testing. Given a classifier trained on the training set, the goal is to estimate the model accuracy on these test sets without any data annotations. In this section, we first show dataset distance measured by BoP + JS exhibits strong negative correlation with classifier accuracy. We then demonstrate an accuracy predictor based on the BoP representation gives accurate performance estimates compared to state-of-the-art methods.

5.1. Experimental Settings

Correlation study under CIFAR-10 setup. We conduct a correlation study by comparing BoP + JS with classifier accuracy. Following the same setup in [21], we use a series of datasets sharing the same label space (but usually with distribution shift) with CIFAR-10 [49]. Specifically, we train ResNet-44 classifier [35] on the training set of CIFAR-10, which consists of 50,000 images from 10 classes. Here, we use the CIFAR-10-C benchmark [37] for correlation study, which contains different types of corruptions with 5 levels of severity including per-pixel noise, blurring, synthetic weather effects, and digital transforms. Then, for each dataset, we compute its BoP vector, its BoP + JS divergence from CIFAR-10 training set and the classifier accuracy. Following the same setup in [21], we use the RepVGG-A1 [22], VGG-16-BN [72] and MobileNet-V2 [70].

Predicting classification accuracy under CIFAR-10 setup. We train a regressor that takes as input the BoP representation and outputs classification accuracy. The regressor is a neural network with 3 fully connected layers and trained on CIFAR-10-C (regression training set). We evaluate accuracy prediction on CIFAR-10.1 [69], CIFAR-10.2 [69] and CIFAR-10.2-C [58]. The former two are real-world datasets with natural shift, while the latter one is manually corrupted by the same synthetic shift as [58]. Specifically, we add 10 types of unseen and unique corruptions such as warps, blurs, color distortions and noise additions, with 5 severity levels to CIFAR-10.2. Note that, these corruptions have no overlap with those in CIFAR-10-C [58].
Figure 4. Correlation between train-test distance measured by BoP + JS and model accuracy. Top: Correlation study under CIFAR-10 setup using ResNet-44, RepVGG-A1, VGG-16-BN and MobileNet-V2. Each data point denotes a dataset from CIFAR-10-C. Bottom: Correlation study under ImageNet setup using EfficientNet-B1, DenseNet-121, Inception-V4 and ViT-Base-16. ImageNet-C datasets are used as test sets. The straight lines are fit with robust linear regression [41]. Under both setups, we observe the strong Spearman’s rank correlation ($|\rho| > 0.98$) between BoP + JS and model accuracy.

For the above, we extract image features (output of penultimate layer of ResNet-44) from CIFAR-10 training set. We construct a codebook by dividing the features into 80 clusters with k-means.

Correlation study under ImageNet setup. We use DenseNet-121 [40] classifier trained on ImageNet training set. We employ a series of datasets from the ImageNet-C benchmark [36] where the classifier is tested. ImageNet-C uses the same types of corruptions as CIFAR-10-C. We construct a codebook of size 1000 on the ImageNet training set from which images features are extracted by the penultimate layer of DenseNet-121. We project each dataset in ImageNet-C onto the codebook and obtain their BoP representations. When exhibiting linear correlations, we calculate BoP + JS between each ImageNet-C dataset and the training set, and compute classification accuracy. We also use EfficientNet-B1 [78], Inception-V4 [77] and ViT-Base-16 [23] to repeat above procedure for correlation study.

Evaluation metric. Same as Section 4.1, we use Pearson’s correlation $r$ and Spearman’s rank correlation $\rho$ to show linearity and monotonicity between BoP based dataset distance and model accuracy, respectively. To evaluate the effectiveness of accuracy estimation, we use root mean squared error (RMSE) by calculating the difference between estimated accuracy and ground truth before taking the mean across all the test sets. A larger RMSE means a less accurate prediction, and vice versa.

Compared methods. We compare our system with four existing ones. 1) Prediction score: it estimates model accuracy using the maximum of Softmax output (i.e., confidence score). An image with a confidence score greater than a predefined threshold $\tau \in [0, 1]$ is considered correctly predicted. We select two thresholds ($\tau = 0.8$ and $0.9$). 2) Difference of confidence (DoC) [34] trains a linear regressor mapping average confidence to classifier accuracy; 3) Average thresholded confidence with maximum confidence score function (ATC-MC) [31] calculates a threshold on CIFAR-10 validation set and regards an image with a confidence score higher than the threshold as correctly classified; 4) Network regression ($\mu + \sigma + FD$) [21] trains a neural network that takes as input the feature mean, covariance and Fréchet distance between a set of interest and training set and outputs model accuracy. All methods, if applicable, are compared under the same conditions as our system, e.g., classification training set and regression training set.

5.2. Evaluation

Strong correlation: A test set is difficult (low accuracy) if it is dissimilar to the training set using BoP + JS. Fig. 4 presents the correlation study of two setups and various classifiers. We observe a very high Spearman’s rank correlation ($|\rho| > 0.99$) in all the scenarios. It indicates that classification accuracy is highly correlated with JS divergence between BoPs of training and test sets. That is, test accuracy drops proportionally to the distance between the given training set and a test set. The results demonstrate BoP + JS between training and test sets is an effective indicator of classification accuracy. More studies are presented in the supplementary materials.

Effectiveness of the BoP representation in predict-
Table 3. Method comparison in predicting classifier accuracy under CIFAR-10 setup. We compare four methods: predicted score-based method with hard threshold \( \tau \), neural network regression based on feature statistics \((\mu + \sigma + FD)\), average thresholded confidence with maximum confidence score function (ATC-MC) and difference of confidences (DoC). We use CIFAR-10.1 and CIFAR-10.2 (both real-world) and CIFAR-10.2-C (manually corrupted) as unseen test sets for accuracy prediction. We use RMSE (%) to indicate precision of estimates. In each column, we compare our method with the best of the competing ones. We report results by average of five runs.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-10.1</th>
<th>CIFAR-10.2</th>
<th>CIFAR-10.2-C (50)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Severity 1</td>
<td>Severity 2</td>
<td>Severity 3</td>
</tr>
<tr>
<td>Prediction score ((\tau = 0.9))</td>
<td>0.297</td>
<td>0.550</td>
<td>3.638</td>
</tr>
<tr>
<td>DoC [34]</td>
<td>0.490</td>
<td>0.263</td>
<td>2.247</td>
</tr>
<tr>
<td>(\mu + \sigma + FD) [20]</td>
<td>0.455</td>
<td>0.561</td>
<td>5.875</td>
</tr>
<tr>
<td>BoP ((K = 80))</td>
<td>0.218</td>
<td>0.122</td>
<td>2.458</td>
</tr>
<tr>
<td>BoP ((K = 100))</td>
<td>0.186</td>
<td>0.124</td>
<td>2.849</td>
</tr>
</tbody>
</table>

Figure 5. Impact of codebook size on correlation strength on CIFAR-10-C. Correlation scores \(|r|\) and \(|\rho|\) are relatively low under a small size and become stably high when the size is greater than 20 for all four model architectures.

...ing classification accuracy on various unseen test sets. After performing correlation study, we train a neural network regressor on CIFAR-10-C and test it on a series of test sets. Results are summarized in Table 3. We have the following observations. First and foremost, BoP representation achieves the best accuracy prediction performance, evidenced by the lowest RMSE across all the four test scenarios. For example, on the test sets of CIFAR-10.2-C, the RMSE of our method is 4.551, which is 1.051 lower than the second best method [21]. This clearly validates the effectiveness of the BoP representation. Second, we observe that the “Prediction score” method is unstable. While it has good results under \(\tau = 0.9\) on CIFAR-10.1 and CIFAR-10.2 datasets, it is generally inferior to the competing methods in other test scenarios. Our observation is similar to [21], suggesting that a more robust threshold selection method is needed for this method. Third, although BoP has slightly higher RMSE than DoC on Severity 1 of CIFAR-10.2-C (2.458 v.s. 2.247), we stress that BoP is overall more stable and effective on real world datasets and other severity levels of synthetic datasets.

Impact of codebook size is summarized in Fig. 5 under CIFAR-10 setup. We conduct the study using different sizes on four classifiers. We observe correlation scores first increase and then become stable when codebook size is larger than 20. These results are considered validation and help us use competitive and stable codebook sizes in Table 3.

6. Conclusion

This work introduces a bag-of-prototypes (BoP) dataset representation to vectorize visual datasets. It first computes a codebook composed of clustering prototypes and then a prototype histogram for a dataset. The BoP vector considers the underlying local semantic distribution of a dataset and is thus more discriminative than global dataset statistics. Specifically, when used in conjunction with JS divergence, the proposed descriptor better captures the underlying relationship across datasets. This advantage is validated by its promising results in two dataset-level tasks: assessing training set suitability and test set difficulty. This work establishes the baseline usage of the BoP scheme, and more investigations and applications will be made in future work.

Acknowledgements

We thank all anonymous reviewers for their constructive comments in improving this paper. This work was supported by the ARC Discovery Project (DP210102801).
References


[56] Mark Mazumder, Colby Banbury, Xiaozehe Yao, Bojan Karlaš, William Gaviña Rojas, Sudhia Diamos, Greg Diamos, Lynn He, Douwe Kiela, David Jurado, et al. Dataperf:


[61] Amandalynne Paullada, Inioluwa Deborah Raji, Emily Bender, Emily Denton, and Alex Hanna. Data and its (dis)contents: A survey of dataset development and use in machine learning research. Patterns, 2021. 1


