Multi-Model Voting is All You Need

Hiu Pang, Lui
Hong Kong University of Science and Technology
hplui@connect.ust.hk

Abstract

Semi-supervised learning can reduce the cost of manual data labeling by leveraging unlabeled samples. In this work, we propose a novel semi-supervised learning workflow, ***voted pseudo-labeling method***, in image classification tasks with ANN and evaluate its performance against fully-supervised baselines. Our experiments on the CIFAR-10 [1] dataset show that our proposed semi-supervised workflow outperforms a model trained solely on the labeled data by 12.88% in overall accuracy, achieving 0.638 compared to 0.753 for a model trained on the full labeled dataset. This approach can be particularly beneficial for applications where high model accuracy is not a primary concern, but minimizing the labeling costs. Code is open-sourced for verification at:

`https://github.com/VYPang/CV-Ensemble-Learning`

1 Introduction

Semi-supervised learning has emerged as a powerful paradigm in machine learning, offering the potential to leverage labeled and unlabeled data to improve model performance [2]. The learner has both labeled training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{l} \sim^{iid} p(\mathbf{x}, y)$ and unlabeled training data $\{\mathbf{x}_i\}_{i=l+1}^{l+u} \sim^{iid} p(\mathbf{x})$, and learns a predictor $f : X \mapsto Y$. The fundamental premise of semi-supervised learning is that unlabeled data, when used in conjunction with a relatively small amount of labeled data, can provide valuable insights about the underlying data distribution and lead to more accurate and robust models.

[3] proposed the pseudo-labeling method in semi-supervised learning for deep neural networks. In the proposed approach, a base model using labeled data is trained. Pseudo-label is predicted from the base model for unlabeled data. The data with pseudo-label is combined into the labeled data to perform subsequent training. The pseudo-label updating process could be repeated until convergence. Consider a dataset, subset A and subset B represents the labeled data and unlabeled data respectively, while B' is the unlabeled subset with pseudo-labels updated. The workflow is presented in Figure 1.

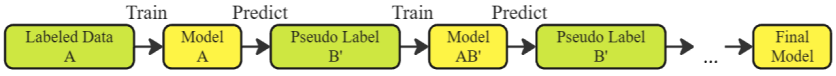

Figure 1: pseudo-labeling method proposed in [3]

In this work, we propose the ***voted pseudo-labeling method***, where *Vote Module*, shown in Figure 2, is introduced to improve the existing pseudo-labeling method. The *Vote Module* made use of the idea of ensemble learning in algorithms such as random forest. It could achieve a more satisfactory accuracy compared to the vanilla pseudo-labeling method.

Figure 2: proposed voted pseudo-labeling method

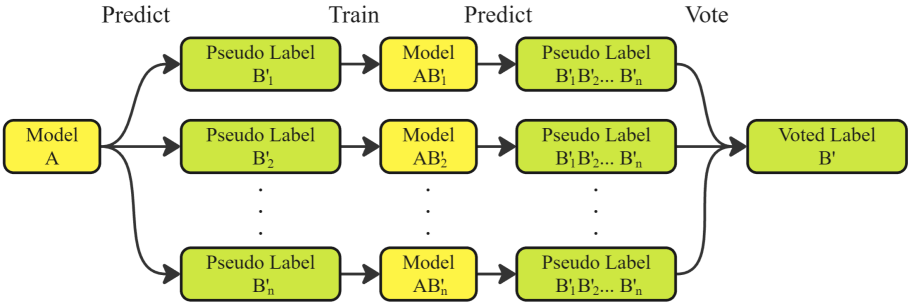

Figure 3: vote module

2 Vote Module

In *Vote Module*, the unlabeled subset B is further divided into multiple subsets determined by the parameter n-*value*.

2.1 Multi-Model Training & Prediction

As shown in Figure 3, the model is trained with labeled subset A. Prediction is done on unlabeled subset $B_1, ..., B_n$ to obtain pseudo-labels in n separated branches. **Note that each unlabeled subset $B_1, ..., B_n$, would have the same number of data if divisible.** In each branch, a new model is trained with labeled subset A and corresponding unlabeled subset B'_i that has been assigned pseudo-labels. This allows the method to capture the unique patterns and characteristics present in each of the unlabeled subsets. Finally, for each branch, pseudo-labels is obtained for all unlabeled subsets $B'_1, ..., B'_n$ through prediction using the model trained on that branch. This results in n sets of pseudo-labels for the entire unlabeled dataset.

By training separate models on the different unlabeled subsets $AB'_1, ..., AB'_n$, the method encourages each model to learn unique representations and capture the distinct characteristics present in each subset. This is analogous to the concept of model diversity in ensemble learning where having models with different inductive biases and learning different aspects of the data can lead to better overall performance.

Compared to the existing method in Figure 1, a single model is trained on the combined subset of AB', which would have a lower expected accuracy as pseudo-labels all unlabeled data solely depend on Model A. In contrast, the proposed method in Figure 3 trains n specialized models, each on a different subset AB'_i, that would have a higher expected accuracy. The ensemble of these models is then used to predict the unlabeled subset. Due to the diversity of representations learned by the n specialized models, we can expect that the Model AB' in our proposed workflow would outperform the Model AB' in Figure 1.

Nevertheless, the *Vote Module* involves several branches which could be computed in parallel. This could reduce the time required to train the final model, meanwhile obtaining a final model with better performance.

2.2 Voting

For the classification task on the CIFAR-10 dataset, the voting process is initiated by obtaining log-softmax value vectors of each data in subset B, $\mathbf{p} = (\vec{p}_1, ..., \vec{p}_n)$, where $\vec{p}_i$is a log-softmax value vector with length equals to number of class in the classification task. Pseudo-label of each data is then obtained by,

$$\hat{y}_{\text{pseudo}} = \arg\max \frac{1}{n} \sum_{i=1}^{n} w_i \vec{p}_i$$

where w_i would be the weighting for each specialized model, which is not used in our experiment. However, one could determine the weighting by normalizing a parameter that can evaluate the accuracy or faithfulness of each specialized model.

For other tasks such as segmentation tasks in computer vision, pseudo-label can also be obtained by voting in a similar method shown above. Let $\mathbf{Y}_{m \times n}$ be the output binary image, the pseudo-label could be achieved by,

$$\hat{y}_{\text{pseudo}} = \frac{1}{n} \sum_{i=1}^{n} w_i \mathbf{Y}_{m \times n}$$

3 Experiment

We study three datasets,

1. **MNIST** [4]
   A dataset of 28x28 grayscale images of handwritten digits (0-9).
2. **MNIST-Fashion** [5]
   A dataset of 28x28 grayscale images of fashion items (e.g. shirts, pants, bags).
3. **CIFAR-10** [1]
   A dataset of 32x32 color images from 10 object classes (e.g. airplanes, cars, birds). Consists 50000 training data and 10000 testing data.

These datasets are fully labeled. To conduct the experiment, 80% of the ground truth label would be ignored to mimic the unlabeled subset, while the rest with ground truth labels would be the labeled subset.

For each dataset, we conduct the following experiments,

1. **Labeled-only**: Train a model using only the labeled subset of the dataset.
2. **Pseudo-labeling**: Train a model using the workflow proposed in [3], which includes an initial model trained on the labeled subset, followed by iterative pseudo-labeling and re-training.
3. **Voted Pseudo-labeling**: Train models with our proposed workflow, using $n = 4$ specialized models.
4. **Full Dataset with Ground-truth**: Train a model using the full dataset with the ground-truth labels.

We compare the performance of these three approaches on the respective test sets (Section 3.1). Additionally, we investigate the effect of the n-value (number of specialized models on the performance of our voted pseudo-labeling method (Section 3.2). We have selected 5-iterations for the pseudo-labeling method to compete with our voted pseudo-labeling method in terms of time consumption and pseudo-label updated times.

The models used in these experiments are standard convolutional neural networks (CNNs) suitable for image classification tasks. To provide a fair comparison, we ensure that the model architectures (Figure 4) and hyperparameters are consistent across the different approaches. The experiments are implemented using PyTorch and the results are reported in terms of the classification of the test sets. We would also ensure that the labeled subset is identical in all methods.

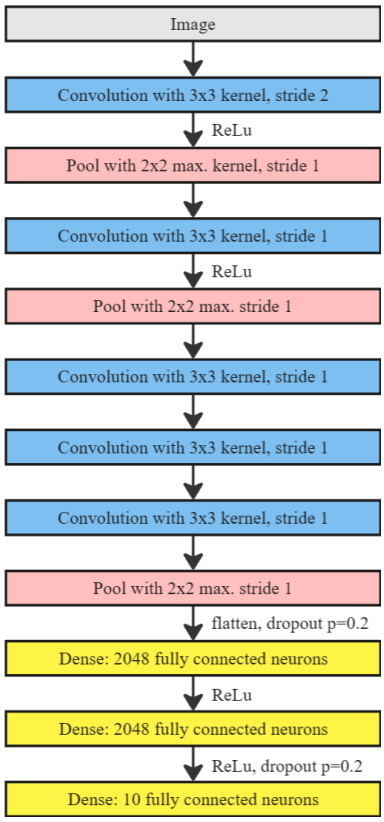

Figure 4: model architecture used in experiment

Cross-entropy loss is used in the experiment, while stochastic gradient descent is used as an optimizer with 0.5 momentum and 0.01 of learning rate.

3.1 Results

Table 1 reports the average and worst-group accuracy of all approaches. *Voted pseudo-labeling method* achieves satisfactory accuracy growth compared to the model trained solely on the labeled subset. Achieving at most 17.2% and 36.6% increase in average and worst-group accuracy respectively. Additionally, compared to the vanilla pseudo-labeling method [3] trained with 5 iterations of pseudo-label update, *voted pseudo-labeling method* consistently achieves higher average accuracy and worst-group accuracy on all 3 datasets.

Method	MNIST (5 epochs)		MNIST-Fashion (5 epochs)		CIFAR-10 (20 epochs)	
	Avg Acc.	Worst-group Acc.	Avg Acc.	Worst-group Acc.	Avg Acc.	Worst-group Acc.
Labeled-only	95.7%	96.5%	82.0%	54.4%	57.4%	31.7%
Pseudo-labeling [3]	96.6%	94.2%	82.6%	34.7%	60.4%	19.2%
Voted Pseudo-labeling (Ours)	**97.4%**	**95.1%**	**82.7%**	**62.3%**	**67.3%**	**43.3%**
Full dataset with ground truth (Baseline)	98.7%	98.0%	87.2%	64.0%	76.8%	43.3%

Table 1: Average and worst-group test accuracy of models trained via different methods and baseline. Voted pseudo-labeling method substantially improves average and worst-group test accuracy relative to pseudo-labeling method [3] and outperforms model solely trained on labeled subset.

It is worth noting that while the voted pseudo-labeling method outperforms the labeled-only and pseudo-labeling [3] approaches, there is still a gap between its performance and the fully-supervised baseline model trained on the complete labeled dataset. The *voted pseudo-labeling method* is able to significantly narrow the performance gap compared to the vanilla pseudo0labeling approach, demonstrating its ability to effectively leverage unlabeled data and approach the upper bound set by the baseline model.

3.2 n-value

The n-value determines number of specialized models used in the *Vote Module* of voted pseudo-labeling method. It also determines the number of subsets $B_1, ..., B_n$ split from unlabeled subset B.

n-value	CIFAR-10 (20 epochs)	
	Avg Acc.	Worst-group Acc.
2	63.3%	32.8%
3	65.9%	37.0%
4 (Table 1)	**67.3**%	**43.3**%
5	66.4%	43.3%
6	66.3%	44.8%
7	65.1%	43.8&

Table 2: Average and worst-group test accuracy of models trained via voted pseudo-labeling method with different n-value.

Table 2 reports the average and worst-group accuracy of models trained via *voted pseudo-labeling method*. It was suggested that the model performance achieve a peak at $n = 4$, while the subsequent increase of n would cause fluctuation in model performance in the cost of longer training time. Therefore we suggest that $n = 4$ would be a thumbs-up selection of n-value.

What is special with $n = 4$?
In the experiment, we randomly select 80% of data from the CIFAR-10 dataset as the unlabeled subset, leaving 10000 data in the labeled subset. Coincidentally, within the *Vote Module*, B_1, B_2, B_3, B_4 have the same data size of 10000. Therefore, it was suspected that the *Vote Module* would achieve good pseudo-label voting results as the data size of unlabeled subset B_i is greater or equal to labeled subset A.

3.3 Class balancing

Class Balanced	CIFAR-10 (20 epochs)	
	Avg Acc.	Worst-group Acc.
No (Table 1)	60.4%	19.2%
Yes	60.8%	44.1%

Table 3: Average and worst-group test accuracy of models trained via pseudo-labeling method [3] with and without class balancing.

Table 1 reveals models trained via Pseudo-labeling[3] could have a lower worst-group test accuracy compared to the Labeled-only. It was suspected that class imbalance in the labeled subset leads to bias in the Model A in Figure 1. The labeled subset should have balanced classes to reduce bias introduced by the dataset. This can be proved by Table 3, where worst-group test accuracy increased by 129% is class balancing is adopted before training. It can be concluded that an imbalanced labeled subset would inherit bias to the subsequent models due to a relatively inaccurate pseudo-label.

Class Balanced	CIFAR-10 (20 epochs)	
	Avg Acc.	Worst-group Acc.
No (Table 1)	67.3%	43.3%
Yes	67.5%	37.4%

Table 4: Average and worst-group test accuracy of models trained via voted pseudo-labeling method with and without class balancing.

The same labeled subset with and without class balancing is used to test on model trained via the *voted pseudo-labeling method*. Table 4 reports a similar average and worst-group test accuracy between models trained via the voted pseudo-labeling method with and without balancing.The difference is within an acceptable range due to training randomness. The experimental results support that *Vote Module* has the ability to mitigate the issue of class imbalance compared to the vanilla pseudo-labeling method [3]. However, class balancing before training should be a good practice to reduce the variance of the resultant model.

4 Model Interpreting

This section reveals an innovative approach to achieving deep learning model decision boundary visualization. We have extracted the output of 2nd dense layer shown in Figure 4 as a feature vector v_i for each input data, where $||v_i|| = 2048$. All feature vectors are collected from the testing data to perform Principle Component Analysis (PCA).

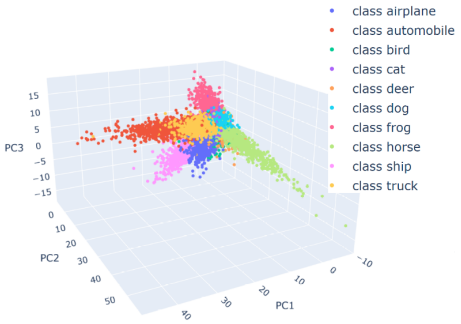

Figure 5: PCA of feature vector obtained from model trained via full dataset CIFAR-10 with ground truth. Interactive 3D scatter plot is available on GitHub.

The visualized feature space in Figure 5 is obtained from the model trained via the full dataset CIFAR-10 with ground truth. It provides valuable insights into the model's decision-making process. The PCA-transformed feature vector exhibits distinct clusters corresponding to different classes, indicating that the model has learned meaningful representation that effectively captures the underlying class structure in the data. An interesting observation is that the clusters are not perfectly separated, suggesting the presence of some degree of overlap or ambiguity in the feature space. This could be attributed to the inherent complexity of the CIFAR-10 dataset, where certain classes may share similar visual characteristics, making it challenging for the model to distinguish them unambiguously.

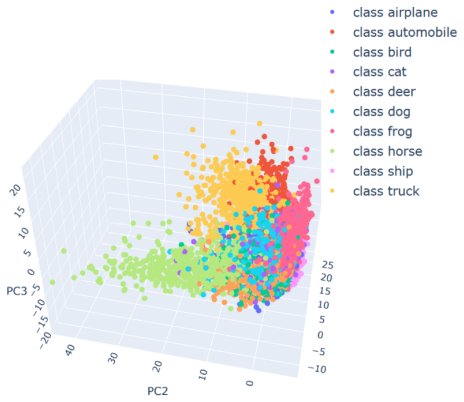

Figure 6: PCA of feature vector obtained from model trained via pseudo-labeling method [3].

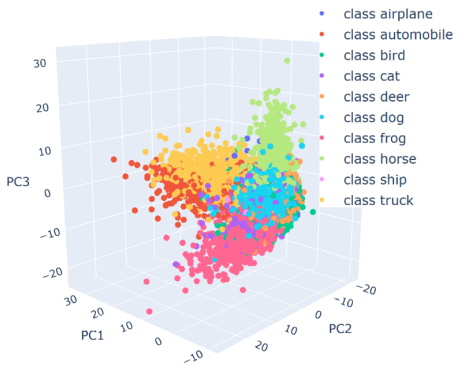

Figure 7: PCA of feature vector obtained from model trained via voted pseudo-labeling method.

In contrast, the PCA visualizations of the model trained using the pseudo-labeling method [3] and the *voted pseudo-labeling* method (Figure 6 & 7) show that the formed clusters are not as distinct as those in Figure 5. This suggests that the semi-supervised learning methods may not have learned representations as effective as the model trained on the full dataset with ground truth labels, which could potentially limit the model's generalization and performance.

5 Conclusion

The proposed *voted pseudo-labeling method* introduces a *Vote Module* that utilizes ensemble learning to improve the existing pseudo-labeling approach for semi-supervised learning. The experiments conducted on MNIST [4], MNIST-Fashion [5], and CIFAR-10 [1] datasets demonstrate that the voted pseudo-labeling method out-

performs the labeled-only and vanilla pseudo-labeling approaches [3]. The key advantages of the proposed workflow are the ability to reduce manual labeling costs while still achieving high model performance, as well as the potential to parallelized the training of the specialized models to reduce the overall computation time. The results suggest that the voted pseudo-labeling method can be a valuable semi-supervised learning technique, particularly for applications where high model accuracy is not the primary concern but minimizing labeling costs is desired.

Future research could explore ways to further enhance the clarity of decision boundaries learned by the voted pseudo-labeling method. Additionally, it would be essential to evaluate the performance of the voted pseudo-labeling method on other computer vision tasks beyond image classification, such as image segmentation, to assess its broader applicability in the field of semi-supervised learning.

References

[1] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. (Technical report, University of Toronto).

[2] Zhu, Xiaojin, and Andrew B. Goldberg. Introduction to semi-supervised learning. Springer Nature, 2022.

[3] Lee, Dong-Hyun. "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks." Workshop on challenges in representation learning, ICML. Vol. 3. No. 2. 2013.

[4] LeCun, Y., Cortes, C., & Burges, C. J. (1998). The MNIST database of handwritten digits.

[5] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.