Leveraging Uncertainty Quantification for Data Labeling Expansion

Leo Jenkins
Department of Computer Science
Caltech
Pasadena, CA 91125
lbjenkin@caltech.edu

Damon Lin
Department of Computer Science
Caltech
Pasadena, CA 91125
dllin@caltech.edu

Evan Wang
Department of Computer Science
Caltech
Pasadena, CA 91125
ezwang@caltech.edu

Abstract

Unlabeled data is abundant and easy to obtain thanks to web scraping, but in most situations, the data is useless without accurate labels. Manual annotation of these data points is often expensive and time-consuming. In this work, we propose an automated approach to expand the labeled dataset by utilizing uncertainty quantification. By training a model on a small labeled dataset, we can then leverage it to predict labels for unlabeled data, prioritizing those with the highest confidence. Our method employs uncertainty quantification techniques to estimate the confidence of model predictions using a Bayesian neural network. By repeatedly adding the unlabeled data points to the labeled dataset with the highest predicted confidence, we incrementally expand the labeled dataset. This automated process eliminates the need for manual annotation and significantly reduces the costs associated with data labeling. We show the results of our method on the CIFAR10 dataset with ResNet as the backbone model.

1 Introduction

The scarcity of labeled data poses a significant challenge in many machine learning applications. Insufficiently labeled data can limit the performance and generalization of models. Moreover, labeling data points by hand can be expensive and time-consuming. As a result, many promising avenues of research cannot get off the ground. In this paper, we propose a method that utilizes uncertainty quantification to automatically generate more labeled data, thereby alleviating the burden of manual data annotation.

The idea is to train a model initially using the available labeled data. Then, we utilize this trained model to evaluate the uncertainty or confidence level of the unlabeled data. By estimating the model's uncertainty, we can identify instances where the model is highly confident in its predictions. If the model assigns a high confidence level to a particular unlabeled sample, we can consider it as if it were labeled with that prediction and add it to the training set. We then train the model again on the increased training set. This process can then be repeated, gradually expanding the labeled dataset size until all of the data has been labeled. This process of increasing labeled dataset size will be referred to as bootstrapping.

Preprint. Under review.

One of the key advantages of this approach is the potential to save valuable time and human resources associated with manual data labeling. Instead of relying solely on labor-intensive manual annotation, the model itself becomes an active participant in the labeling process. By leveraging the model's uncertainty estimates, we can effectively generate labeled data, reducing the need for manual annotation efforts and accelerating the training process.

However, there are certain considerations and potential limitations to be aware of when employing uncertainty quantification for data labeling expansion. First, the reliability of uncertainty estimates depends on the model's architecture and training. Different models may yield different uncertainty estimates, and it is crucial to select a model that is well-suited for capturing uncertainty effectively. Moreover, even with the best techniques of uncertainty quantification, machine learning models frequently overestimate their confidence level.

Another challenge is determining an appropriate threshold for confidence levels. Setting the threshold too low may result in the inclusion of incorrectly labeled samples, introducing noise into the training set. Conversely, setting the threshold too high may overlook valuable but slightly uncertain samples, limiting the benefits of data labeling expansion. Moreover, the higher the confidence level, the more computing power is necessary to train and sample the model over and over again. Careful experimentation and validation are necessary to find the optimal threshold that balances precision and recall.

2 Background and Related Work

2.1 Bayesian Neural Network

Bayesian Neural Networks (BNNs) have gained popularity in the field of machine learning thanks to their ability to provide probabilistic outputs and uncertainty quantification. Unlike traditional neural networks that produce point estimates, BNNs leverage Bayesian inference to model distributions over network weights, enabling them to capture and express uncertainties in predictions 2.

One popular approach using BNNs for confidence estimation is Variational Inference (VI), which formulates the Bayesian learning problem as an optimization task. VI optimizes a variational lower bound on the true posterior distribution of the weights. This framework allows for efficient uncertainty estimation by modeling weight distributions and providing posterior approximations.

2.2 Pseudo-Labels

Some commonly used methods for generating pseudo labels involve both supervised 4 and unsupervised [3] training. We will focus on the supervised setting. Initially, a model is trained on a small labeled dataset. Then, the model is used to make predictions on the unlabeled data, and the predicted labels with high confidence are assigned as pseudo labels. These pseudo-labeled samples are combined with the original labeled data to create an expanded training set. The model is retrained using this augmented dataset, and the process is iterated to refine the model's performance.

Researchers have explored different strategies to improve the quality of pseudo labels and mitigate their inherent noise. Techniques like calibration are used to select pseudo-labeled samples with high confidence, reducing the risk of mislabeled instances affecting the model's performance. Adversarial training is another technique used to enhance the robustness of the model to noisy pseudo labels.

3 Research Question and Approach

The main question we are trying to solve is how to create an automated, reliable labeling process to convert unlabeled data to labeled data using deep learning models, thereby expanding the labeled dataset size for supervised learning. We approach this problem using uncertainty quantification to get an estimate of the confidence level of the model on an input. We would expect that if the model predicts a label with high confidence, the data is more likely to be correctly labeled, so it is reasonable to put it into the labeled dataset. Ideally, we would implement this approach on a variety of vision and language tasks and a wide variety of models to determine the effectiveness and generalization of the method. Due to time and resource constraints, we will only discuss the experiments and results of our method on one study.

4 Experiments and Results

4.1 Data

We test the performance of our model on an image classification task. The CIFAR10 dataset will be used as it is a standard benchmark in computer vision. The labels are the ten classes for which the images belong to. Out of the 60000 32×32 images, 5000 will be used to train the initial model, 45000 will be used for label prediction to expand the labeled dataset, and 10000 will be used for evaluation only. These three subsets will be referred to as the training set, expansion set, and test set respectively. Although we have access to the labels of all 60000 images, the model will not have access to the expansion set labels. The model will predict labels for the expansion set, and the predicted labels will be treated as the true labels for training. The goal is to have as much of the expansion set correctly labeled to ensure high classification accuracy on the test set.

4.2 Model

The residual network (ResNet) has emerged as a popular choice in deep learning for image classification tasks, primarily due to its ability to address the issue of vanishing gradients that can occur in deep architectures [1]. Unlike traditional deep learning models that predict a fixed label for a given input, we introduce modifications to the ResNet architecture to incorporate a weight distribution at each layer. This adaptation transforms the ResNet into a BNN framework.

In our approach, we train the BNN to learn the parameters of the weight distribution at each layer. Instead of determining a single set of weights, the model learns a posterior distribution over the weights using VI. Specifically, we choose the multivariate normal distribution as the parametric family to represent the weight distributions. During the training process, the BNN aims to learn the mean and covariance of the weight distribution that best captures the true posterior distribution.

With this trained BNN model, we can leverage its probabilistic nature to estimate the uncertainty or confidence level associated with the predictions. By repeatedly inputting a fixed data point into the model, we obtain a distribution of predicted labels for that input. This distribution reflects the uncertainty inherent in the model's predictions. The spread or concentration of the distribution provides us with a measure of confidence in the model's classification for the given data point.

4.3 Data Expansion

The outline of our bootstrapping algorithm consists of several key steps to leverage uncertainty quantification and expand the labeled dataset. We begin by splitting the CIFAR10 dataset into three distinct subsets: the training set, the expansion set, and the test set, as described in Section 4.1.

The training set serves as the initial dataset for training the model, containing labeled data points that the model has access to during the training process. The expansion set comprises the unlabeled data points that we aim to label and subsequently add to the training set. Finally, the test set is reserved solely for evaluation purposes, allowing us to monitor the test accuracy of our model as the algorithm progresses.

Once the three datasets are established, we proceed with training the model using the labeled data from the training set. This initial training phase equips the model with the knowledge necessary to make predictions on unseen data. This initial training phase is important because it is the only time the model trains on data we know is 100% labeled correctly.

Subsequently, we enter the bootstrapping stage of our algorithm. We pass the unlabeled data points from the expansion set through the trained model to obtain predicted labels. To gauge the model's confidence in these predictions, we repeat this process multiple times and record the distribution of predicted labels for each data point. By capturing the variability in predictions, we can estimate the uncertainty associated with each data point's label.

To determine which labels to assign to the unlabeled data points, we compare the distribution of predicted labels to a given confidence threshold. If the percentage of times a particular label is predicted exceeds the threshold, we assign that label to the corresponding data point. These labeled data points are then added to the training set, augmenting its size and diversity. While the model is not given the correct label, we compare the predicted and actual labels to see whether the model made the correct decision to include the data point in its training set.

Following the expansion of the training set, we proceed to retrain the model using the updated dataset. This iterative process allows the model to incorporate the newly labeled data and refine its predictions accordingly. We repeat the steps of passing the unlabeled data through the model, estimating the uncertainty, assigning labels based on confidence, and expanding the training set until no further predictions surpass the confidence threshold. Any remaining unlabeled data points would then need to be manually classified.

4.4 Bootstrapping Results

Now, we present the results of our experiment. First, we define a baseline model, which is the ResNet BNN trained on the training set of 5000 images for 40 epochs. For the bootstrapped experiment, we train the model on the training set for 20 epochs before expanding the training set with newly labeled data points.

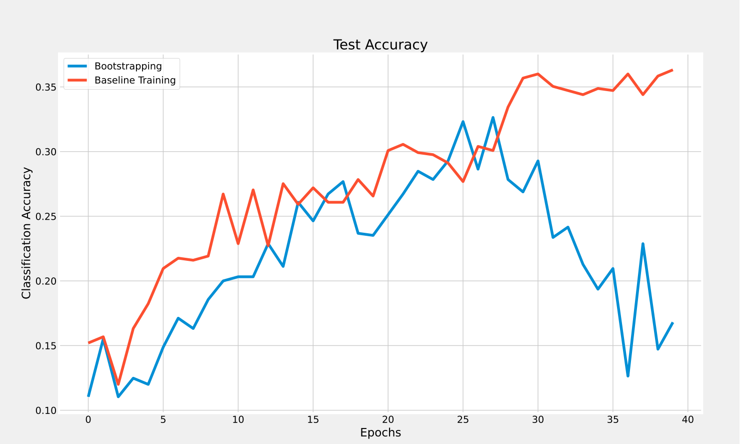

Figure 1: Comparison of accuracy on the test set of baseline model versus the bootstrapped model over 40 epochs of training

The results are summarized in 1 In this setting, the baseline model achieves 36.3% accuracy on the test set. We can also see that the bootstrapped model performs similarly to the baseline model for the first 20 epochs since the training procedure is identical. Since the accuracy of the bootstrapped model is slightly lower after these first 20 epochs due to random deviations, the bootstrapping procedure faces an uphill battle.

Past 20 epochs, the test accuracy of the bootstrapped model sees a large increase, to a level slightly above the baseline model. This increase can be contributed to the increase in the training set from bootstrapping. A larger dataset allows the model to learn new attributes of the datapoints, potentially improving its ability to classify the remaining unlabeled datapoints.

However, after 27 epochs, we see a large drop in the test accuracy of the bootstrapped model, dropping to as low as 13.1%. We believe this is due to the addition of mislabeled data into the training set which then throws off the entire bootstrapping process. This phenomenon can be seen in the two figures shown below.

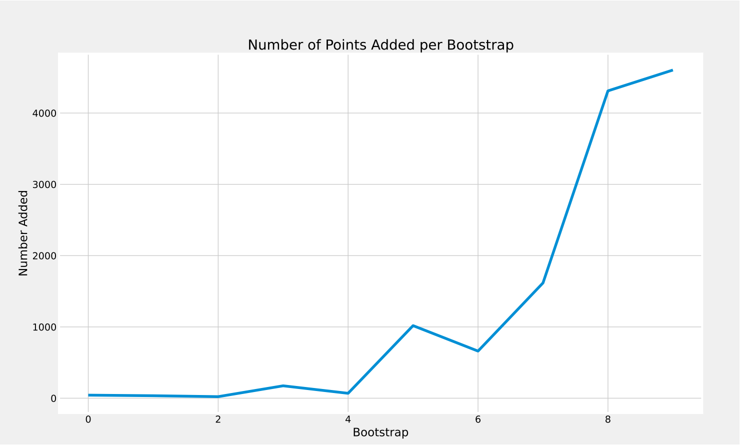

Figure 2: Number of new data points added to the training set after each iteration, with very little added initially and a lot more at the end

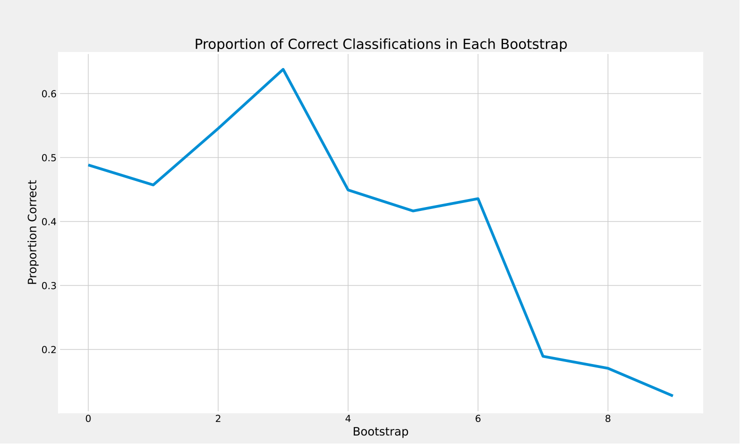

Figure 3: The proportion of correctly classified data points added after each iteration, with a significant degradation in accuracy at the end

From 2 and 3, we see that the data points added initially were few but accurately labeled. We can tell that our confidence bound makes a positive impact as the points that are transferred are classified correctly at a much higher rate than the overall testing set (for either the bootstrapped or the baseline model).

This initial expansion of the training set led to the bootstrapped model to have a boost in performance. Rising from below 30% to above 35%, surpassing the baseline model along the way. However, in the later rounds, the model becomes overconfident in the predictions, leading to a large injection of mislabeled data points into the training set. A decrease in classification accuracy compounded with an increase in the number of data points added significantly degraded the quality of the training set. The result is a positive feedback loop where the more noise that is added, the worse the predictions are, and the more likely additional noise will be added. This was reflected in the large and sudden drop in the bootstrap model test accuracy as it was not learning as intended.

5 Discussion and Conclusion

As demonstrated in this work, utilizing uncertainty quantification for data labeling expansion offers a promising avenue to address the problem of insufficiently labeled data in machine learning. By leveraging the model's uncertainty estimates, we can automatically generate labeled data, reducing the need for extensive manual annotation efforts and accelerating the model training process. This approach holds great potential for improving model performance and scalability in real-world applications.

However, the results also demonstrate how tricky this is to get right. For every mislabeled data point transferred to the training set will bring more mislabeled data points with it. Thus it is extremely important for our model to be disciplined in its confidence estimations. We believe we can improve our model by testing it in friendlier conditions. By only letting it train for a small number of epochs on a 10-class dataset, it was unlikely to do well. Afterall, the baseline only achieves 40% confidence over 40 epochs. Such a challenging context makes it more likely for the model to introduce mislabeled predictions.

Moreover, merely increasing the accuracy of the pre-databoosted model will not fix our issues. Several challenges and areas of future investigation exist within the field of uncertainty quantification for data labeling expansion. One such challenge is determining appropriate thresholds for uncertainty measures to determine the confidence level required for labeling. We hypothesize that setting a static threshold could be an issue. Since more noise is being introduced into the training set as we progress, we think we would want to tighten the confidence threshold as the size of the training set increases.

While there is not one answer to these questions, setting these thresholds effectively is crucial to ensure the quality and reliability of the generated labels. Further research is needed to develop robust and adaptive thresholding techniques that can accommodate different application domains and data characteristics.

Improving uncertainty estimation techniques is another important research direction. Enhancing the model's ability to accurately capture and quantify uncertainty is essential for reliable data labeling expansion. This includes exploring advanced Bayesian modeling approaches, ensemble methods, and deep learning architectures tailored specifically for uncertainty quantification. With additional computing power, we would also want to explore an increase in the number of samples taken when calculating the confidence of a prediction.

Future research should also focus on synergies between uncertainty quantification and active learning strategies. Active learning can complement uncertainty quantification by selecting the most informative and uncertain data points for manual labeling, further optimizing the data labeling process. Combining uncertainty quantification with active learning techniques, such as uncertainty sampling or query-by-committee, can lead to more efficient and effective data labeling expansion.

One specific avenue for future investigation is the exploration of soft labels that incorporate the model's confidence level. By indicating the model's certainty in the label itself, such as including a confidence probability, the amount of noise entering the training dataset could potentially be reduced. This would allow the model to assign higher weights to more reliable and confidently labeled data points, further improving its learning capacity.

Another promising direction is to calibrate the model's uncertainty estimates by using a calibration set. Models often exhibit overconfidence in their predictions, leading to miscalibrated uncertainty estimates. By incorporating a separate calibration set and recalibrating the model's uncertainty measures, we can obtain a more accurate assessment of which unlabeled data points should be added to the labeled dataset. Calibrated uncertainty quantification can enhance the reliability of the data labeling expansion process and contribute to better model performance.

Incorporating these proposed procedures, such as using soft labels and calibration techniques, would significantly boost the performance of machine learning models by reducing the number of incorrectly labeled data points. This would result in improved model generalization, increased robustness to uncertainty, and better decision-making capabilities in real-world scenarios.

In real-world scenarios, you would also have a differing ratio of labeled to unlabeled data. In our experiment, we only tested a ratio of 1:9 labeled to unlabeled data. Therefore, more research would be necessary to figure out what are acceptable and ideal ratios for this data expansion problem. This research would dictate when this approach could be used on real-world unlabeled datasets.

In summary, uncertainty quantification for data labeling expansion holds tremendous potential for overcoming the challenges posed by insufficiently labeled data. Further research should focus on advancing uncertainty estimation methods, investigating synergies with active learning strategies, exploring the use of soft labels and calibration techniques, and developing practical guidelines and benchmarks for uncertainty-guided data labeling expansion. By addressing these research directions, we can unlock the full potential of uncertainty quantification in expanding the labeled dataset and advancing the field of machine learning.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[2] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization, 2022.

[3] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training, 2020.

[4] Haoning Zhang, Junwei Bao, Haipeng Sun, Huaishao Luo, Wenye Li, and Shuguang Cui. Css: Combining self-training and self-supervised learning for few-shot dialogue state tracking, 2022.