AutoDAL: Distributed Active Learning with Automatic Hyperparameter Selection

Xu Chen,[1] **Brett Wujek**[1]
[1]SAS Inc
steven.xu.chen@gmail.com, Brett.Wujek@sas.com

Abstract

Automated machine learning (AutoML) strives to establish an appropriate machine learning model for any dataset automatically with minimal human intervention. Although extensive research has been conducted on AutoML, most of it has focused on supervised learning. Research of automated semi-supervised learning and active learning algorithms is still limited. Implementation becomes more challenging when the algorithm is designed for a distributed computing environment. With this as motivation, we propose a novel automated learning system for distributed active learning (AutoDAL) to address these challenges. First, automated graph-based semi-supervised learning is conducted by aggregating the proposed cost functions from different compute nodes in a distributed manner. Subsequently, automated active learning is addressed by jointly optimizing hyperparameters in both the classification and query selection stages leveraging the graph loss minimization and entropy regularization. Moreover, we propose an efficient distributed active learning algorithm which is scalable for big data by first partitioning the unlabeled data and replicating the labeled data to different worker nodes in the classification stage, and then aggregating the data in the controller in the query selection stage. The proposed AutoDAL algorithm is applied to multiple benchmark datasets and a real-world electrocardiogram (ECG) dataset for classification. We demonstrate that the proposed AutoDAL algorithm is capable of achieving significantly better performance compared to several state-of-the-art AutoML approaches and active learning algorithms.

Introduction

The development of automated machine learning (AutoML) (Thornton et al. 2013)(Guyon et al. 2016) has become popular in data science discussions, publications, applications, and systems, as an important tool to build better machine learning models. Typically, existing methods rely on manually fine-tuned machine learning models requiring a significant amount of human resources, time and effort. To address this, AutoML (Guyon et al. 2016) techniques have been widely investigated and applied in applications such as autonomous vehicles, sales forecasting, lead prioritization systems, and many other systems. In general, AutoML is used to generate and optimize machine learning pipelines that can transform data, select features, select the best model type, and optimize hyperparameter settings to discover the "best" model. While automated machine learning for supervised learning has been extensively studied, in many applications such as medical image analysis, fraud detection, and mechanical system monitoring and maintenance, the available dataset contains a limited number of observations with labels assigned (e.g. malignant, fraudulent, abnormal, failed, etc.). In these situations, active learning (AL) (Wang and Ye 2015)(Maystre and Grossglause 2017), which incorporates dynamic interactive querying for labels, is preferred due to the power of combining the limited labeled data and large amount of unlabeled data, accounting for the selection of the most informative data. Like supervised learning, good hyperparameter tuning schemes also play a crucial role in active learning. For instance, some theoretical analysis (C.S.Ong, Smola, and Willamson 2003) have identified that the machine learning performance is sensitive to the kernel width selection in similarity measurement. There has also been research (Beatty, Kochis, and Bloodgood 2018) studying the behavior of batch size in query selection for active learning. Therefore, given a limited number of labeled samples, it is critical to automatically determine a good set of hyperparameters for active learning in order to efficiently maximize the classification performance.

However, research on automated active learning remains limited. Traditional automated supervised learning techniques cannot be directly applied to automated active learning for two main reasons. First, whereas supervised learning relies on more labeled examples for model selection and performance improvement, automated active learning has limited labeled data and therefore must resort to exploitation of unlabeled observations for performance improvement in the process of automatic hyperparameter selection. The techniques provided in automated supervised learning such as cross-validation (Koch et al. 2018) are not directly applicable for automated active learning. Second, in order to capture the synergy between the classification and the query selection in active learning (Sener and Savarese 2018) and determine the hyperparameters in an automated fashion, it is very desirable to formulate a joint optimization problem to

solve them in a unified procedure.

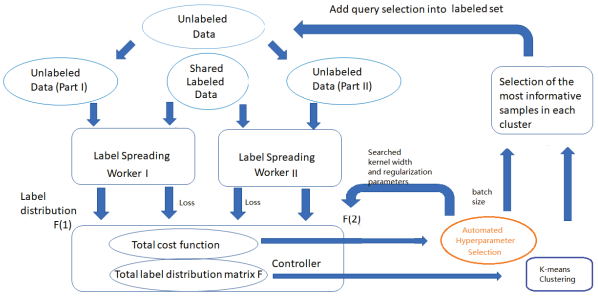
The computational expense of training machine learning models is another important consideration, often relying on the use of distributed machine learning algorithms due to their advantages in handling big data (Liu, J.Wang, and S.Chang 2012). Applying automated active learning in such a distributed computing environment (Chang, Lin, and Zhou 2017)(Chakraborty 2018) is an even more challenging task due to the level of coordination required among the compute nodes. Thus, it is very desirable to develop a new algorithm for automated active learning in a distributed setting for processing large scale data. In this work, we propose a novel automated distributed active learning framework by first integrating the entropy regularization (Li et al. 2014) into the loss function from the distributed graph-based learning. Subsequently, the hybrid search algorithm consisting of a genetic algorithm and a local generating set search is designed to solve the optimization problem using a two-step optimization for automation. The overall algorithm consists of a distributed classification model based on semi-supervised learning and a centralized sample selection strategy as described in Fig.1. By decentralization in the classification stage, the proposed algorithm is capable of handling big data classification in a distributed fashion. In addition, by aggregating the information from different worker nodes in the controller, the selection process enables nodes to cooperatively select data based on uncertainty, diversity, and representativeness of data. This achieves diversity of the selected data for active learning with automatic hyperparameter selection. The main contributions of this paper can be summarized as follows:

- This is the first time that a framework for automated distributed active learning algorithm has been proposed. By jointly optimizing multiple hyperparameters in the classification stage and query selection stage based on the distributed graph loss and the entropy regularization, AutoDAL enables automatic selection of hyperparameters in a unified framework and therefore achieves promising classification performance gains.
- To efficiently exploit distributed computing resources, in the classification stage, we randomly partition the unlabeled data and replicate the labeled data, which provides scalable and superior performance for big data classification. In the query selection stage, the most informative and representative samples are collected in a centralized manner.
- Application of AutoDAL on multiple benchmark datasets and large scale electrocardiogram (ECG) signal classification on a real-world dataset has demonstrated significant performance gain over state-of-the-art approaches, including two popular active learning methods and three existing autotuning methods.

Related Work

Entropy regularization (Li et al. 2014)(Li et al. 2016) has been shown to be successful as a means to benefit from unlabeled data in the framework of maximum a posteriori estimation for semi-supervised learning (SSL) and active learning (AL). Different from (Li et al. 2014)(Li et al. 2016), here we propose to utilize cluster-specific maximum entropy regularization for joint optimization and apply it for an automated active learning framework. In (Liu, J.Wang, and S.Chang 2012) it was reported that by utilizing graph transduction via alternating minimization (GTAM), joint optimization of both the classification function and the initial label matrix using alternating minimization is feasible and effective. In comparison, our work resolves the new problem for automatic active learning with entropy regularization in a distributed manner and optimizes for a larger set of hyperparameters and functions. Recently, a scheme called safe SSL (Li 2015) has been introduced to alleviate the performance degradation issue in SSL. In (Koch et al. 2018), a combination of search methods has been proposed for automated machine learning and achieved successful results on multiple datasets. However, that work primarily focused on supervised methods and did not explore the interesting and common problems of automated semi-supervised learning and active learning. (Li et al. 2019) has proposed to use meta-learning and a large margin separation method for automated semi-supervised learning. However, that work did not consider the issue of applying the algorithm in an active learning framework and did not explore a distributed solution for big data.

Figure 1: A block diagram of the proposed automated distributed active learning (AutoDAL) system with two worker machines.



The Proposed Algorithm

Preliminaries and Problem Definition

The proposed algorithm is initialized by relying on the automated label spreading algorithm for semi-supervised learning. As described in (Zhou et al. 2003), assume that we have a point set $Z = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_n\}$, and the label set $L = \{l_1, \ldots, l_c\}$, where c is the number of classes. The first l data points, $\{x_1, x_2, \ldots, x_l\}$, are labeled by $\{y(x_1), y(x_2), \ldots, y(x_l)\}$. The aim is to predict the labels of the unlabeled data points using the information from both the labeled data and the unlabeled data. Let F denote the set of $n \times c$ matrices where the entries in the matrices are nonnegative. A matrix $F = [F_1^T, \ldots, F_n^T]^T$ indicates a classification on the dataset χ by labeling each point x_i as a label $y_i = \arg\max_{j \leq c} F_{i,j}$. Define a $n \times c$ matrix Y with $Y_{ij} = 1$ if x_i is labeled as $y_i = j$ and $Y_{ij} = 0$ otherwise. Convention-

ally, the edge weight between point x_i and x_j, W_{ij}, is calculated by a Gaussian kernel $W_{ij} = \exp(-\|x_i - x_j\|^2/2\sigma^2)$ if $i \neq j$ and $W_{ii} = 0$ Subsequently, the normalized similarity matrix is constructed as $S = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with its diagonal element (i, i) equal to the sum of the ith row of W (Zhou et al. 2003).

$$F^{t+1} = \alpha SF^t + (1 - \alpha)Y \ , \tag{1}$$

where t indicates the current iteration and α is the weight to control the confidence in initial labels. α is between 0 and 1, where greater α indicates stronger confidence and it can be tuned in the proposed automation algorithm. $F^0 = Y$.

procedure AUTOMATED DISTRIBUTED ACTIVE LEARNING ALGORITHM(Observations x, initial label matrix Y)
  while $\Omega - L > 0$ **do** ▷ Ω is the budget for the total number of labeled data, L is the number of data already labeled
    Distribute the unlabeled data randomly and replicate the labeled data in different worker nodes. Solve (8) using the hybrid search strategy of GA with GSS:
    Evaluate initial parent points P asynchronously in parallel. Populate reference cache-tree, R, with unique points from P. Associate each point $p \in P$ with step $\triangle_p$ initialized to $\triangle$.
    while $(|R| \leq n_b)$ where n_b is evaluation budget **do**
      Select $\Lambda \subset P$ for local search based on the optimization problem formulated in (8).
      for $p \in P$, search $\zeta_p = \zeta_p \cup \{p + \triangle_p\} \cup \{p - \triangle_p\}$;
      if $\min_{\chi \in \zeta_p} J(F, \chi) < J(F, p) - \triangle_p^2$, then set p=$\chi$ ▷ pattern search success
      else $\triangle_p$=$\triangle_p/2$ ▷ pattern search failure
    end while
    Conduct K-means clustering on F^*. Given F^* and h^* , solve (9) to determine $\hat{Y}^*$ with the top h selections.
    Add the selected samples $x_{j_1}, \ldots, x_{j_h}$ and estimated labels $\hat{y}_{j_1}, \ldots, \hat{y}_{j_h}$ to labeled dataset and output the optimal selections including label probability distribution matrix F^*, hyperparameter set χ^*, updated label matrix $\hat{Y}^*$ and batch size h^*.
  end while
end procedure

Search Methods

The proposed automated active learning uses a combination of a genetic algorithm (GA) and a generating set search (GSS)(Griffin and Kolda 2010) technique for searching the optimal hyperparameter values as described in (Koch et al. 2018). A GA is a global search algorithms that calculates optimal solutions to problems by applying the principles of natural selection and evolution. GAs can be applied to various types of optimization problems and are particularly effective for situations when gradient-based optimization techniques do not work. Since a GA does not rely on gradient information, it is effective for problems where objective function has multiple local optima, when the objective function is not differentiable or continuous, or when solution elements are constrained to be integers or sequences, all of which are common cases for algorithm hyperparameter spaces. On the other hand, GSS is a local search technique designed for problems that have continuous variables, exploiting gradient information to fine-tune optimal points found by the GA. As explained in (Koch et al. 2018), the automated algorithm begins with a Latin hypercube sample (LHS) of the hyperparameter space to ensure coverage across the range of each hyperparameter. The best configurations from the LHS are then used to generate the initial population for the GA, which iteratively searches for the best model configurations. The automated algorithm adds an additional incremental step to each iteration of GA, invoking the GSS algorithm to perform local search in a neighborhood of the current GA best solutions and improving the convergence to the optimal solutions once the GA is getting close to the convergence region.

Automated Semi-supervised Learning

Definition: Define χ to be the parameter sets that the automated algorithm intends to optimize (in this case the algorithm hyperparameters). Suppose M^{auto} is the selected optimal set of parameters from the algorithm, and $Per(M^{auto})$ is the classification performance of the semi-supervised learning algorithm. The goal of the automated algorithm is to determine M^{auto} such that of $Per(M^{auto})$ is always better than $Per(M^{random})$. For RBF kernels, an optimal parameter set $\chi = \{\sigma, \alpha\}$ including the variance of the Gaussian kernels (kernel width) σ and the weight parameter α (from Equation (1)) are determined in the automated algorithm. For the k nearest neighbor kernel, the automated algorithm selects and optimizes $\chi = \{\sigma, \alpha, k\}$ which consists of three parameters including the number of nearest neighbors k, σ and α.The cost function that the proposed automated algorithm is optimizing is calculated as follows:

$$Q(F, \chi) = \ \tfrac{1}{2}(\textstyle\sum_{i,j=1}^{n} W_{ij} \| \frac{1}{\sqrt{D_{ii}}} F_i - \frac{1}{\sqrt{D_{jj}}} F_j \|^2 + \mu \sum_{i=1}^{n} \|F_i - Y_i\|^2) \ , \tag{2}$$

The first term of the cost function is the smoothness constraint, characterizing the similarity of the nearby points. The second term is the fitting constraint, which emphasizes the closeness of the classification results compared to the initial label assignment, where the positive regularization parameter μ captures the trade-off between these two competing constraints with $\mu = \frac{1}{\alpha} - 1$ as in (Zhou et al. 2003). The automated semi-supervised learning algorithm is minimizing the cost function in order to determine the classification function F and the optimal set of the hyperparameters χ.

$$F^*, \chi^* = \arg\min_{F, \chi} Q(F, \chi) \tag{3}$$

The equation (3) is a combinatorial optimization problem, which can be efficiently addressed with the proposed combined GA+GSS based search method.

Automated Distributed Semi-supervised Learning

In order to alleviate the computational burden for big data and leverage parallel processing, we extend the automated semi-supervised learning algorithm to execute within a distributed computing environment. The major challenge for the distributed algorithm is to compute the $n \times n$ distance matrix (where n is the number of observations) in a parallel fashion. Here we propose a novel and effective solution to address this issue: given the distributed setting with M multiple worker nodes, we randomly partition the unlabeled data into M disjoint subsets and allocate each subset to one worker node. In order to fully utilize the labeled data and maximize the classification performance, the labeled data is replicated and distributed to every worker node. Since the unlabeled data is partitioned, the original large distance matrix in the label spreading algorithm is approximated by computation of multiple sub-matrices with much lower dimensions. This scheme has shown to significantly reduce the running time and memory requirement for big data applications. Denote m as the index of the mth worker machine, $m = 1, 2, \ldots, M$, the total cost function for the distributed solution is represented as:

$$J(F,\chi) = \sum_{m=1}^{M} \frac{1}{2}\Big(\sum_{i,j=1}^{n} W_{ij}^{m} \Big|\Big| \frac{1}{\sqrt{D_{ii}^{m}}} F_i^m - \frac{1}{\sqrt{D_{jj}^{m}}} F_j^m \Big|\Big|^2 + \mu \sum_{i=1}^{n} ||F_i^m - Y_i^m||^2\Big) \tag{4}$$

Here F^m, D^m, W^m and Y^m represent the probability distribution function, the diagonal weight matrix, the full weight matrix and the initial label matrix calculated in the mth node respectively. Note that the top l rows are identical for Y^m since these are observations with known labels.

$$F^*, \chi^* = \arg\min_{F,\chi} J(F,\chi) \tag{5}$$

Since the data partitions are distinct, non-overlapping subsets, the optimization of the cost function (5) can be further decomposed into m independent sub-problems for optimizing the optimal hyperparameters efficiently. Given the output probability distribution matrix F^m from (5), the controller collects all the F^m from worker machines to formulate the estimated total probability distribution matrix F^*.

Automated Distributed Active Learning

Active learning usually selects the most informative samples as labeled data in order to maximize the classification performance. For active learning, denote $\hat{Y}$ as the updated label matrix after the selected samples are merged into the labeled dataset and h as the batch size. Assume the sample $x_{j_{h_i}}$ is selected in the query selection with the label $c_{j_{h_i}}$, the value in $\hat{Y}$ will be updated to be 1 (from 0) at the row j_{h_i} and the column $c_{j_{h_i}}$. In order to automatically determine the batch size and updated label matrix, the joint optimization tends to simultaneously minimize the graph loss in the distributed classification and maximize the uncertainty of the selected labeled data in the query selection. Therefore, the total parameter sets to optimize in active learning include the probability distribution function F, the parameter set χ in semi-supervised learning and updated label matrix $\hat{Y}$ and the batch size h. The formulation of the joint optimization problem is cast as follows:

$$\begin{aligned} S(F,\chi,\hat{Y},h) &= C_1 + C_2 + C_3 \\ C_1 &= \sum_{m=1}^{M} \frac{1}{2} \sum_{i,j=1}^{n} W_{ij}^{m} \Big|\Big| \frac{1}{\sqrt{D_{ii}^{m}}} F_i^m - \frac{1}{\sqrt{D_{jj}^{m}}} F_j^m \Big|\Big|^2 \\ C_2 &= \sum_{m=1}^{M} \frac{1}{2} (\mu \sum_{i=1}^{n} ||F_i^m - \hat{Y}_i^m||^2) \\ C_3 &= -\lambda \sum_{k=1}^{c} \sum_{i=1}^{[h/c]} H(y_i|x_i)\ , \end{aligned} \tag{6}$$

where $H(y_i|x_i) = -\sum_{j=1}^{c} P(y_i = j|x_i) \log(P(y_i = j|x_i))$ and $P(y_i = j|x_i)$ is retrieved from F^*. C_1 and C_2 calculate the loss from distributed semi-supervised learning. C_3 in the equation (6) tends to maximize the conditional entropy for the top $[h/c]$ samples in each cluster where the clusters are generated by first applying K-means clustering to the probability distribution matrix F. Here $[h/c]$ stands for the largest integer not bigger than h/c. The conditional entropy is an effective measure of class overlap, which characterizes the usefulness of the unlabeled data where labeling is ambiguous and uncertain. Moreover, as the selection of samples with maximum entropy is conducted within local clusters, it expects to select samples from different classes instead of choosing most of the samples from the majority class. The proposed cluster specific entropy regularization ensures the best trade-off between diversity and uncertainty in AL.

$$F^*, \chi^*, \hat{Y}^*, h^* = \arg\min_{F,\chi,\hat{Y},h^*} S(F,\chi,\hat{Y},h) \tag{7}$$

As solving the optimization problem (7) directly is intractable, we propose a two-step solution. In the first step, we fix $\hat{Y}$ in order to optimize F^* , χ^* and h^*, and the optimization problem can be simplified as:

$$F^*, \chi^*, h^* = \arg\min_{F,\chi,h^*} (C_1 + C_3)\ , \tag{8}$$

In the second step, the optimized F^*, χ^* and h^* are utiized to estimate $\hat{Y}^*$. Thus, the minimization problem can be reduced to

$$\hat{Y}^* = \arg\min_{\hat{Y}} \sum_{m=1}^{M} \sum_{i=1}^{n} ||F_i^m - \hat{Y}_i||^2 \tag{9}$$

By iteratively solving the optimization equations (8) and (9) until all the estimated hyperparameters are stable, the joint optimization problem can be resolved for AutoDAL. Once the query selection is completed, the labeled dataset is augmented with the selected queries and the models are updated for the next iteration of learning as shown in Fig.1.

Figure 2: Comparison of the accuracy using USDM (Yang et al. 2015), AER (Fu et al. 2018), Auto-WEKA (Thornton et al. 2013), Auto-sklearn(JFeurer et al. 2015), ASSL+US (Li et al. 2019), DAL and the proposed AutoDAL algorithms for five benchmark datasets with different percentages of labeled data varying from 0.1% to 20%.

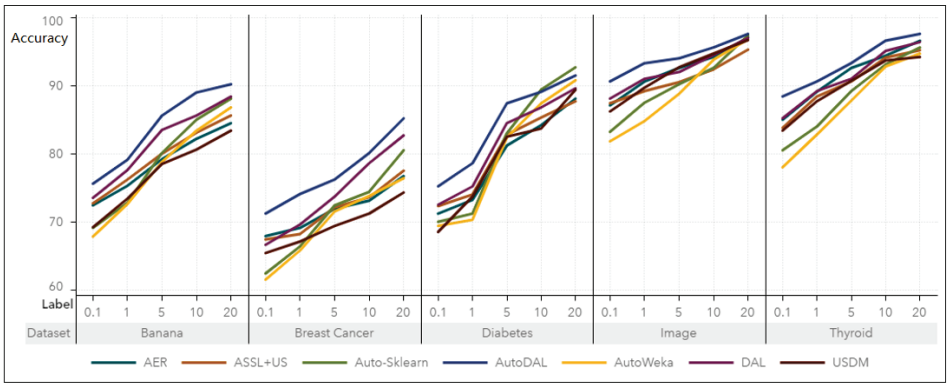
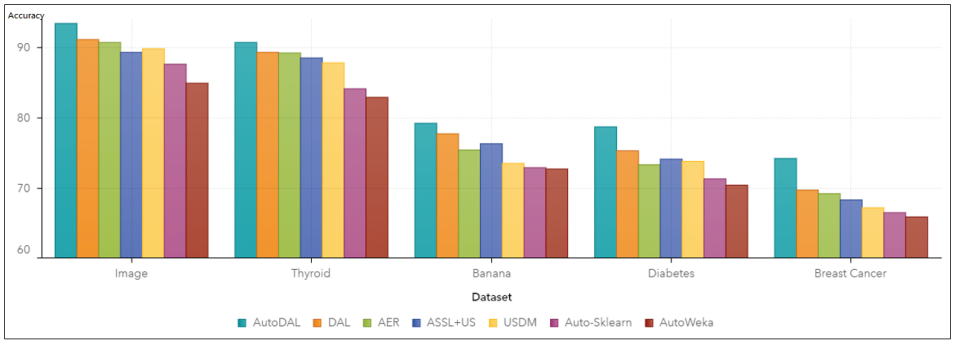


Figure 3: Comparison of the accuracy using USDM, AER, Auto-WEKA, Auto-sklearn, ASSL+US, DAL and the proposed AutoDAL algorithms for five benchmark datasets when the percentage of labeled data is 1%.



Experiments

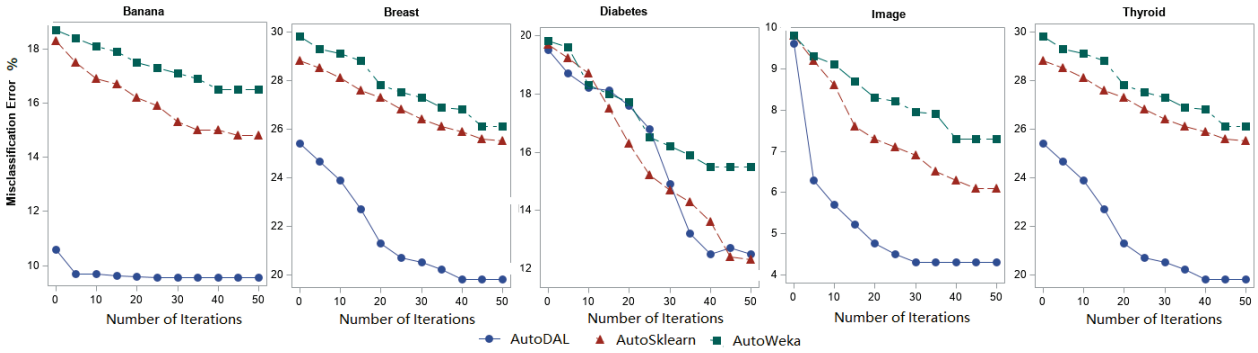
Datasets: We evaluate the classification performance of the proposed method over five benchmark datasets taken from mldata.org ($http://mldata.org/repository/tags/data$) including banana, breast cancer, diabetes, image and thyroid. AutoDAL is also applied to a real-world ECG heartbeat categorization dataset from Kaggle for classification ($https://www.kaggle.com/shayanfazeli/heartbeat$). Specifically, the Arrhythmia Dataset includes 109446 samples with 5 categories. The five classes are ['N': 0, 'S': 1, 'V': 2, 'F': 3, 'Q': 4] with the sampling 125Hz. Each heartbeat observation is 188 dimesions. Here the class "N" is the majority class representing normal heart beats, and the rest of four classes represent various types of abnormal heartbeats as minority classes.

Methods for Comparison: The AutoDAL algorithm is compared with five different state-of-the-art approaches: The three competing automated methods are Auto-WEKA (Thornton et al. 2013), Auto-sklearn (JFeurer et al. 2015) and ASSL+US (Li et al. 2019).

- Auto-WEKA considers the problem of simultaneously selecting a learning algorithm and setting its hyperparameters automatically.
- Auto-sklearn adds Bayeisan optimization on top of Auto-WEKA, which demonstrates good performance on supervised learning problems. The running time is set to one minute which is sufficient to ensure the automated system finishes successfully.
- ASSL+US applies uncertainty sampling using maximum entropy on automated semi-supervised learning.
- AL by uncertainty sampling with diversity maximization (USDM) (Yang et al. 2015) exploits the entire active pool to evaluate the uncertainty of the data across multiple classes and an efficient algorithm is used to optimize the objective function.
- AL by approximated error reduction (AER) (Fu et al. 2018). AER (Fu et al. 2018) estimates the error reduction of each candidate based on an expected impact over all data points and an approximated ratio between the error reduction and the impact over its nearby datapoints relying on hierarchical anchor graphs.

In order to further demonstrate the effectiveness of automation, we also compare AutoDAL with distributed active learning (DAL) where we randomly select the hyperparameter values and report the performance as an average of 20 executions. The distributed computing environment is comprised of 139 machines where each machine is running with

Figure 4: The comparison of misclassification error versus the number of iterations for the proposed AutoDAL algorithm, Auto-WEKA and Auto-sklearn for benchmark datasets, where the horizontal axis represents the number of iterations and the vertical axis represents the misclassification error(%).



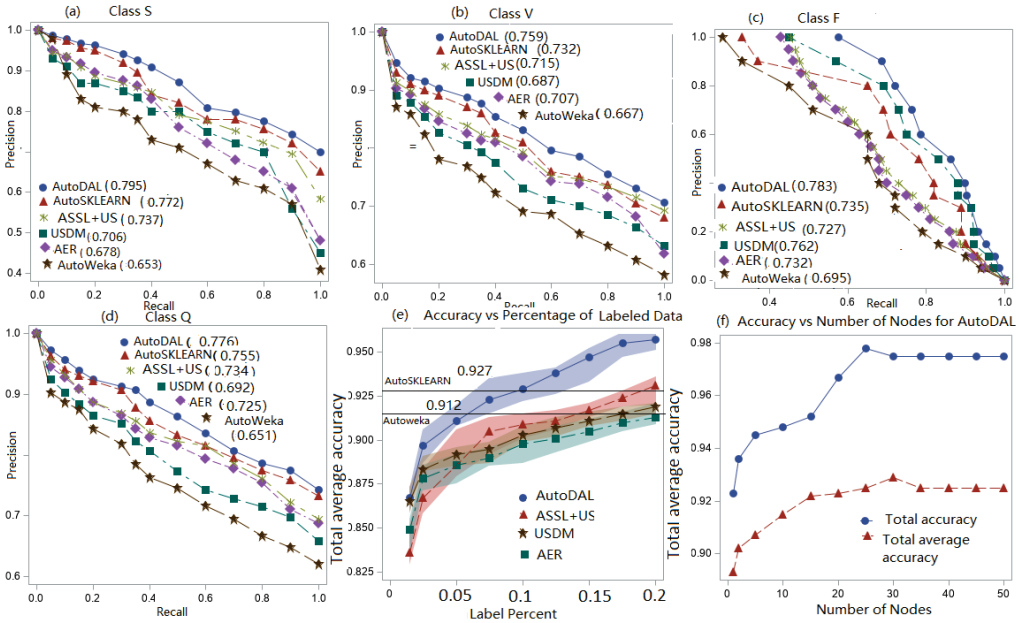
32 threads. We performed 10 runs of each method and report the average performance. For supervised methods, we performed a leave-one-dataset-out validation.

Experiments on benchmark datasets: In the evaluation of benchmark datasets with AutoDAL, both the kNN kernel and RBF kernel are applied for semi-supervised learning and the one with better performance is reported. The hyperparameters for the automated algorithm to estimate are initially provided as ranges where $k = [1, 100]$, $\sigma = [0.001, 10000]$, $\mu = [0, 10]$, and the batch size in active learning $h = [1, 30]$. The step sizes for searching on k, σ, μ, h are set to be $2, 5, 0.1, 2$ respectively. For each dataset, a small percentage of data is randomly selected as labeled instances including $\{0.1\%, 1\%, 5\%, 10\%, 20\%\}$ and the remaining are chosen to be unlabeled data. λ is 0.5.

As demonstrated in Fig.2, when the percentage of labeled data is less than 1%, active learning based methods usually outperform supervised learning methods such as Auto-WEKA and Auto-sklearn with a large margin (greater than 5%). The performance gain over Auto-WEKA and Auto-sklearn can be mainly attributed to the fact that AutoDAL is leveraging the information from the large amount of unlabeled data and selecting the most informative and representative labeled data. The superiority over the method of AutoSSL with uncertainty sampling is probably due to the utilization of the clustering-based uncertainty sampling in joint optimization. When the percentage of labeled data is greater than 5 percent, the performance of supervised learning methods including Auto-WEKA and Auto-sklearn outperform that of the three active learning methods like AuotSSL+US(Li et al. 2019), USDM(Yang et al. 2015) and AER(Fu et al. 2018). AutoDAL achieves the best performance for 23 of the 25 cases, only surpassed by auto-sklearn for the case of the diabetes dataset where the labeled data are greater than 10%. This is probably because the features in diabetes dataset are very discriminative, so with relatively little labeled data a good classifier trained with gradient boosting in Auto-sklearn achieves slightly better performance. We also noticed that the greatest performance gain from AutoDAL compared to other methods is from the breast cancer dataset when the percentage of labeled data is 0.1%, and that AutoDAL improves the classification accuracy compared to DAL with randomly selected hyperparameters by at least 2%. This is due to the effectiveness of the proposed optimization framework which jointly searches and optimizes the kernel width, regularization parameter, and batch size in distributed active learning. In order to better demonstrate the relative performance of accuracy for different methods, Fig.3 visualizes the histogram comparisons for each method when the percentage of labeled data is 1%. The variance of the accuracy achieved by different methods are all within 1.5% on the 1% labeled dataset. We also plot and compare the iteration history of the misclassification error for the proposed AutoDAL and the comparative AutoML methods in Fig.4 for the five benchmark datasets when there are 10% labeled data. As shown in Fig.4 all the autotune methods for the five datasets converge well as the number of iterations increases. AutoDAL performs the best for four out of five datasets; AutoSklearn performs slights better than AutoDAL for the diabetes dataset. For the banana, breast cancer and image datasets, AutoDAL converges faster than other methods due to its selection of the most informative and representative instances as labeled data to use in the classification tasks.

We also studied the effect of each hyperparameter by automatically tuning one hyperparameter at a time and comparing with the performance without tuning that parameter. For three out of five datasets, the kernel width contributes the most to performance gain in the proposed AutoDAL algorithm. The batch size plays the most important role in classification of the other two datasets. Typically, for higher dimensional datasets such as the breast cancer dataset with 30 features and thyroid dataset with 29 features, the kernel width serves as the most important tuning hyperparameter. For instance, for the breast cancer dataset with 0.1 labeled observations, by tuning only the kernel width σ, the classification performance of AutoDAL is improved by 3% compared to the total improvement of 4.6% when tuning all the parameters. For the diabetes dataset, batch size is the most important parameter, individually contributing around 2% performance gain out of the average total gain of 2.4% from all hyperparameters.

Figure 5: The comparison of precision and recall curves for ECG signal classification on different classes including class S, V, F and Q ((a)-(d)), where AutoDAL is compared to USDM, AER, Auto-WEKA, Auto-sklearn and ASSL+US. For each method, the area under the curve (AUC) is reported as a measure for the imbalanced dataset. (e) evaluates the total average accuracy for different active learning methods with 95 percent confidence intervals and the supervised learning methods are implemented with half of data for training and the rest half for cross-validation. (f) demonstates the avarage accuracy versus number of machines for the proposed AutoDAL.



Case study for ECG signal classification: In order to further evaluate the performance of AutoDAL with real-world data, 20 variations of each ECG signal were generated by adding Gaussian noise with zero mean and variance ranging from 0.1 to 0.5 at intervals of 0.1. For each level of variance, four samples were generated. The augmented data set contains 2185920 samples in total. We demonstrate the precision-recall curves for the proposed AutoDAL algorithm and compare with other methods including USDM(Yang et al. 2015), AER(Fu et al. 2018), Auto-WEKA (Thornton et al. 2013), Auto-sklearn (JFeurer et al. 2015) and ASSL+US (Li et al. 2019) on the classes S, V, F and Q in Fig.5 (a)-(d). It can be seen that for all four classes, AutoDAL achieves the top performance in terms of area under the curve, demonstrating the superiority of the proposed algorithm for classification of big data with an imbalanced dataset. In Fig.5(e), the average accuracy of different methods is compared with respect to the percentage of labeled data. Again, AutoDAL consistently achieves the highest avarage accuracy with more than 3% performance margin. When the percentage of labeled data is limited, ASSL+US is the second best method. With increasing number of labeled data, AutoSklearn achieves the second best classfication performance.

To gain further insight on the impact of the hyperparameters governing AutoDAL, a detailed study on the performance gain with various components was conducted, with the following observations. Total accuracy and the average accuracy with AutoDAL increase with computing nodes first and remain stable when the number of machines is greater than 25 as shown in Fig.5(f). This is probably because the percentage of labeled data is higher with increasing nodes which results in better classification for big data. By tuning batch size, a 1.8% performance gain is achieved compared to random selections. The typical range for batch size to achieve the best performances is between 10 and 30. Tuning kernel width boosts the total accuracy by 0.7% . A range of 20-50 for σ results in good classification performance. Tuning the regularization parameter contributes 0.5% gain in terms of the total classification accuracy and the best range is 0.7 to 0.8.

Conclusions

In this paper, we have presented a novel framework for automated distributed active learning (AutoDAL). The proposed AutoDAL algorithm is capable of automatically selecting optimal values of important hyperparameters including kernel width, regularization parameters, and batch size, and efficiently solving the combinatorial optimization problems to achieve good classification accuracy. The distributed framework is scalable to big data and achieves a good tradeoff between the classification accuracy and the computational time. The proposed algorithm has demonstrated promising classification performance for automated hyperparameter tuning in distributed active learning.

References

Beatty, G.; Kochis, E.; and Bloodgood, M. 2018. Impact of Batch Size on Stopping Active Learning for Text Classification. *International Conference on Semantic Computing*.

Chakraborty, S. 2018. Distributed Active Learning for Image Recognition. *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Chang, X.; Lin, S.; and Zhou, D. 2017. Distributed Semi-supervised Learning with Kernel Ridge Regression. *Journal of Machine Learning Research*.

C.S.Ong; Smola, A.; and Willamson, R. 2003. Learning the kernel with hyperkernels. *Journal of Machine Learning Research (JMLR)*.

Fu, W.; Wang, M.; Hao, S.; and Wu, X. 2018. Scalable Active Learning by Approximated Error Reduction. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining* 15.

Griffin, J. D., and Kolda, T. G. 2010. Nonlinearly-constrained Optimization Using Heuristic Penalty Methods and Asynchronous Parallel Generating Set Search. *Applied Mathematics Research*.

Guyon, I.; Chaabane, I.; Escalante, H.; and et al. 2016. A brief review of the chalearn automl challenge. *In Proceedings of AutoML workshop on the 33rd International Conference on Machine Learning*.

JFeurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J.; Blum, M.; and F.Hutter. 2015. Efficient and Robust Automated Machine Learning. *Advances in Neural Information Processing Systems*.

Koch, P.; Golovidov, O.; Gardner, S.; Wujek, B.; Griffin, J.; and Y, X. 2018. Autotune: A Derivative-free Optimization Framework for Hyperparameter Tuning. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Li, Y.; Wang, H.; Wei, T.; and Tu, W. 2014. Semi-supervised learning by entropy minimization. *Neural Information Processing Systems (NIPS)*.

Li, Y.; Kwok, J.; Wei, T.; and Tu, W. 2016. Towards safe semisupervised learning for multivariate performance measures. *In Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Li, Y.; Wang, H.; Wei, T.; and Tu, W. 2019. Towards Automated Semi-supervised Learning. *Association for the Advancement of Artificial Intelligence Conference*.

Li, Y.-F.and Zhou, Z.-H. 2015. Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Liu, W.; J.Wang; and S.Chang. 2012. Robust and Scalable Graph-based Semi-supervised Learning. *Proceeding of IEEE*.

Maystre, L., and Grossglause, M. 2017. Just sort it ! A simple and effective approach to active preference learning . *International Conference on Machine Learning*.

Sener, O., and Savarese, S. 2018. Actively learning for convolutional neural network. *International Conference on Learning Representations*.

Thornton, C.; Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2013. Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. *In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data mining* 847–855.

Wang, Z., and Ye, J. 2015. Querying discriminative and representative samples for batch mode active learning. *ACM Transactions on Knowledge Discovery from Data (TKDD)*.

Yang, Y.; Ma, Z.; Nie, F.; Chang, X.; and Hauptmann, A. G. 2015. Multi-class active learning by uncertainty sampling with diversity maximization.

Zhou, D.; Noble, O.; Lal, T.; Weston, J.; and Scholkopf, B. 2003. Learning with Local and Global Consistency. *Advances in Neural Information Processing*.