Towards Automated Semi-Supervised Learning

Yu-Feng Li[1] **Hai Wang**[1,2] **Tong Wei**[1] **Wei-Wei Tu**[2]
[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] 4Paradigm Inc., Beijing, China
{liyf,wanghai,weit}@lamda.nju.edu.cn; tuww.cn@gmail.com

Abstract

Automated Machine Learning (AutoML) aims to build an appropriate machine learning model for any unseen dataset automatically, i.e., without human intervention. Great efforts have been devoted on AutoML while they typically focus on supervised learning. In many applications, however, semi-supervised learning (SSL) are widespread and current AutoML systems could not well address SSL problems. In this paper, we propose to present an automated learning system for SSL (AUTO-SSL). First, meta-learning with enhanced meta-features is employed to quickly suggest some instantiations of the SSL techniques which are likely to perform quite well. Second, a large margin separation method is proposed to fine-tune the hyperparameters and more importantly, alleviate performance deterioration. The basic idea is that, if a certain hyperparameter owns a high quality, its predictive results on unlabeled data may have a large margin separation. Extensive empirical results over 200 cases demonstrate that our proposal on one side achieves highly competitive or better performance compared to the state-of-the-art AutoML system AUTO-SKLEARN and classical SSL techniques, on the other side unlike classical SSL techniques which often significantly degenerate performance, our proposal seldom suffers from such deficiency.

Introduction

In traditional machine learning, given a dataset, a fine-tuned learning model is built by human. Nowadays, however, along with the successfulness of machine learning, more and more communities (governments, industrial companies) require a learning model for their specific data. Existing approaches based on manually fine-tuned learning models consume a large amount of human resources and efforts. To overcome this issue, the development of automated machine learning (AutoML) (Thornton et al. 2013; Feurer et al. 2015), which attempts to build an appropriate machine learning model for unseen dataset in an automatic manner (without human intervention), has received increasing attention recently. For example, many workshops have been hold in recent machine learning conferences,[1] and a number of AutoML competitions have been organized.[2]

[1] http://automl.chalearn.org

[2] https://www.4paradigm.com/competition/nips2018

AutoML is magnificent yet challenging, since absolute AutoML is infeasible (Guyon et al. 2016). Previous work on AutoML typically focuses on supervised learning problems, and addresses the difficulties including feature engineering (Guyon et al. 2016), model selection (Sun 2016) and hyperparameter optimization (Hutter, Hoos, and Leyton-Brown 2011). Recently there are a couple of systematical schemes that achieve promising performance. For example, AUTO-WEKA combines the machine learning framework WEKA (Hall et al. 2009) with a bayesian optimization method (Hutter, Hoos, and Leyton-Brown 2011) to select a good instantiation of WEKA for a new dataset. AUTO-SKLEARN (Feurer et al. 2015) improves AUTO-WEKA and uses meta-learning (Lemke, Budka, and Gabrys 2015) to warmstart the bayesian optimization procedure, and finally includes an automated ensemble construction step. Google Cloud AutoML is a suite of machine learning products that can automatically train high quality models by leveraging google's state-of-the-art transfer learning techniques and neural architecture search techniques.[3] These above studies show that one could automatically select a quite good learning model and hyperparameter for supervised learning problems.

In many applications, except for supervised learning scenario, many other problems such as semi-supervised learning (SSL) (Chapelle, Schölkopf, and Zien 2006; Zhou and Li 2010) are widespread in reality. However, the efforts on Automated SSL remain limited. In this work we study automated SSL and it is notable that existing AutoML techniques could not directly be applied for the automated SSL problem, since SSL introduces some new challenges.

- First, although meta-learning is able to quickly suggest some instantiations of the learning techniques which are likely to perform quite well, the feature engineering is much harder for SSL since many meta-features extracted from a number of labeled examples (Feurer et al. 2015) are no longer available and suitable.
- Second, unlike supervised learning which typically has performance improvement with more labeled examples, SSL with the use of auxiliary unlabeled instances may sometimes even be outperformed by direct supervised learning with only limited labeled examples (Li and Zhou 2015; Krijthe and Loog 2015; Li, Kwok, and Zhou 2016). Such

[3] https://cloud.google.com/automl/

phenomenon is crucial for SSL and needs to be alleviated in automated SSL. Recently, a scheme termed safe SSL (Li and Zhou 2015) has been presented to alleviate the performance deterioration issue in SSL. They are proposed to address the performance unsafeness in terms of data quality (Li, Wang, and Zhou 2016), model uncertainty (Li and Zhou 2015; Krijthe and Loog 2015) and measure flexibility (Li, Kwok, and Zhou 2016). These works typically focus on one aspect of the learning process, which do not finalize a systematical solution and are not automated SSL.

To alleviate these issues, in this paper we propose to present an automated learning system (AUTO-SSL) for SSL. First, inspired by AUTO-SKLEARN (Feurer et al. 2015), we consider meta-learning to quickly suggest some instantiations of the SSL techniques that are likely to perform quite well. By considering that unlabeled data distribution is important to SSL techniques, diverse unsupervised clustering algorithms are performed and meta-features with respect to intra-cluster and inter-cluster statistics are extracted to enhance the performance of meta-learning. Second, we propose a large margin separation method to fine-tune the hyperparameters and meanwhile alleviate the performance deterioration issue in SSL. Our basic idea is that, if a certain hyperparameter owns a high quality, its predictive results on unlabeled data may have a large- margin separation. We should exploit the large margin hyperparameters while keeping the small margin hyperparameters (which might be risky) to be rarely exploited. Extensive empirical results on 40 datasets over 200 cases demonstrate that our proposal achieves highly competitive or better performance compared to the state-of-the-art AutoML system AUTO-SKLEARN and classical SSL techniques, in addition unlike classical SSL techniques often degenerate performance significantly, our proposal seldom suffers from such deficiency.

In the following, we first highlight the differences with related work, and then present our proposal and the extensive empirical justification, finally we conclude this work.

Related Work

This work is mostly related to two branches of studies.

AutoML: AutoML is challenging. Many issues have been raised (Guyon et al. 2016), such as automated feature engineering (Guyon et al. 2016), automated model selection (Sun 2016), automated hyperparameter optimization (Hutter, Hoos, and Leyton-Brown 2011). From the systematical scheme aspect, one of the earliest work for AutoML is AUTO-WEKA (Hall et al. 2009) which combines the machine learning framework WEKA with a bayesian optimization method to select a good configuration for a new dataset. Later on, to further alleviate the high computational cost and derive a more accurate solution, AUTO-SKLEARN (Feurer et al. 2015) improves AUTO-WEKA and uses meta-learning (Lemke, Budka, and Gabrys 2015) step to warmstart the bayesian optimization procedure, and finally includes an automated ensemble construction for robustness. Although many AutoML techniques have been proposed, they typically work on supervised learning, while the efforts on semi-supervised learning (SSL) remain to be limited.

Safe SSL: Safeness is one important aspect to AutoSSL, since it is not desirable to have an automated yet performance-degenerated SSL system. Recently, safe SSL has some studies, conquering the performance unsafeness in terms of data quality (Li, Wang, and Zhou 2016), model uncertainty (Li and Zhou 2015; Krijthe and Loog 2015) and measure flexibility (Li, Kwok, and Zhou 2016). Nevertheless, they are not AutoSSL sine they do not finalize a systematical scheme.

Towards Automated SSL

Generally, an AutoML system consist of four procedures. Given a collection of datasets, an AutoML system performs meta-learning which extracts meta-features of datasets and then uses a supervised learning model to select a learning algorithm which is likely to perform well for unseen dataset. Then, the AutoML system performs hyperparameter optimization/selection, to derive a good candidate hyperparameter for the selected algorithm. Later, model evaluation is conducted to finalize the ultimate model. Finally, in the prediction phrase, given a new dataset, the AutoML system first represents the dataset via meta-features, and then predicts an appropriate algorithm with a good hyperparameter, which finally finalizes the learning model.

Among the above procedures, we need to tackle with two major difficulties. a) How to design appropriate SSL meta-features to facilitate a better meta-learning? b) How to choose a good quality parameter and alleviate the performance degeneration issue in SSL? In the following, we first present preliminaries and problem description, then we present a scheme for the above two difficulties respectively.

Preliminaries and Problem Definition

Let $\mathcal{D} = \{\{\mathbf{x}_i, y_i\}_{i=1}^{l}, \{\mathbf{x}_j\}_{j=l+1}^{l+u}\}$ be a SSL dataset with limited labeled data and a large amount of unlabeled data, where $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{l}$ corresponds to the labeled instances and $\mathcal{U} = \{\mathbf{x}_j\}_{j=l+1}^{l+u}$ corresponds to the unlabeled instances. $y_i \in \{+1, -1\}$ corresponds to the label of instance $\mathbf{x}_i$, $i = 1, \ldots, l$. Inspired by (Feurer et al. 2015), we have the goal of AutoSSL as following.

Definition 1 (AutoSSL) *Let $\mathcal{S} = \{S^1, \cdots, S^N\}$ be a set of SSL algorithms, and the hyper-parameters of each algorithm S^j have a domain Θ^j. Let A be a baseline supervised learning algorithm, and the hyper-parameters of algorithm A have domain Λ. Suppose that M^{auto} is the output model of the automated SSL system on data set $\mathcal{D}$ and A_Λ the model of supervised learning algorithm trained on labeled data set $\mathcal{L}$. The goal of the automated SSL system is that $Per(M^{auto})$ is always significantly better than $Per(A_\Lambda)$, and rarely worse than $Per(A_\Lambda)$, where $Per(M)$ denotes the performance of model M on testing data.*

Compared to the definition of AutoML and CASH (Combined Algorithm Selection and Hyperparameter optimization) in (Feurer et al. 2015), the definition of AutoSSL does not involve cross-validation to evaluate/optimize the performance, since the labeled examples are limited to afford a reliable model selection. Moreover, unlike AutoML

which does not suffer from performance degeneration problem, the reliability of AutoSSL is crucial and stated. Compared to the description of Safe SSL (Li and Zhou 2015; Li, Zha, and Zhou 2017) where the automated model evaluation is not mentioned, AutoSSL explicitly highlights it.

Meta-Learning with Enhanced Meta-Features

Meta-learning (Brazdil et al. 2008) aims to reason about the performance of learning algorithms across different datasets. Specifically, in AutoML, we collect the performance data and a set of meta-features for a large number of datasets, where meta-features are characteristics of the dataset that help determine which algorithm to use for a new dataset and can be computed in an efficient manner (Feurer et al. 2015).

Meta-feature is central to meta-learning. Rather than the reasoning based on meta-features, meta-features themselves play a more important role to the final performance (Feurer et al. 2015). However, there is a lack of a principle way to design appropriate meta-features for meta-learning. Multiple kinds of meta-features from different aspects have been tried and presented in previous AutoML studies, including a) simple meta-features: describe the basic dataset structure (Michie et al. 1994; Feurer, Springenberg, and Hutter 2015), such as the number of instances, the number of features, etc. b) statistical meta-features (Michie et al. 1994): characterize the data via descriptive statistics such as the kurtosis and skewness; c) PCA-based meta-features (Bardenet et al. 2013): compute various statistics of the dataset principal components.

For SSL, it is important to include characteristics that related to SSL techniques. Nevertheless, current general AutoML meta-features have little to do with SSL methods. To alleviate it, we propose to characterize the distribution of unlabeled data or data distribution assumption (Chapelle, Schölkopf, and Zien 2006; Zhou and Li 2010), which is known as an important factor for SSL, by unsupervised clustering algorithms. Specifically, different SSL techniques prefer to different data distributions, e.g.,

- Graph-based SSL techniques (Zhu, Ghahramani, and Lafferty 2003; Zhou et al. 2004) prefer to manifold assumption or smoothness assumption, where similar instances are likely to have similar labels.
- large margin SSL techniques (Joachims 1999; Li and Zhou 2015) prefer to low-density assumption or cluster assumption, where the decision boundary goes across a low-density region of data distribution.

To our best knowledge, meta-features for the above data distribution assumptions have not been throughly studied before. In this paper, we employ classical clustering algorithms to realize different data assumptions of unlabeled data. The first one is the k-means algorithm (Jain 2010), which describes the mixture-gaussian data distribution and each cluster simulates a density function of data distribution. Therefore, k-means algorithm is a choice to realize the low-density assumption. The second one is the spectral clustering algorithm (Von Luxburg 2007), which partitions the data with respect to data similarity matrix and each cluster simulates a manifold structure of data distribution. Consequently, spectral clustering algorithm is a choice to realize the manifold assumption. Finally, the last one is the hierarchical clustering algorithm (Jain and Dubes 1988) which provides a flexible way to hybrid multiple local clusters. For each clustering algorithm, four popular and prominent meta-features (Maulik and Bandyopadhyay 2002; Davies and Bouldin 1979), namely, intra-cluster cohesion, inter-cluster separation, Davies-Bouldin index and Dumn Validity index are employed.

Table 1: List of Meta-Features in AUTO-SSL

Traditional Meta-Features	
Simple Meta-Features:	**Statistic:**
number of instances	kurtosis min
log number of instances	kurtosis max
number of features	kurtosis mean
log number of features	kurtosis std
dataset dimensionality	skewness min
log dataset dimensionality	skewness max
inverse dataset dimensionality	skewness mean
log inverse dataset dimensionality	skewness std
class probability min	**PCA Statistic:**
class probability max	pca 95%
class probability mean	pca skewness first pc
class probability std	pca kurtosis first pc
Meta-Features with *Unsupervised Clustering*	
Algorithms:	**Meta-Features:**
K-Means	Intra-cluster cohesion
SpectralClustering	Inter-cluster separation
AgglomerativeClustering	Davies-Bouldin Index
	Dunn Validity Index

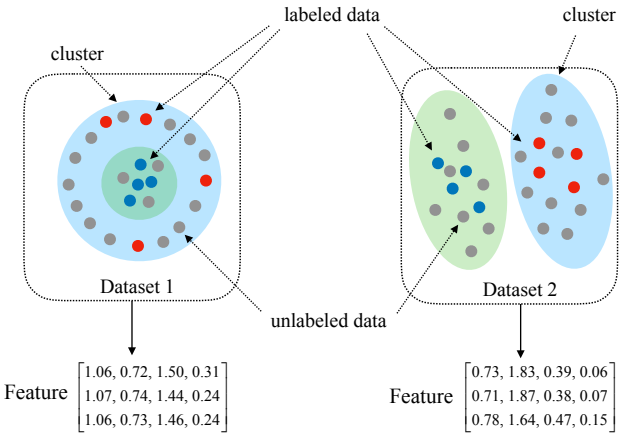

Figure 1: Meta-features via unsupervised clustering

Figure 1 shows an illustrative example for the meta-features extracted from unsupervised clusterings. In Figure 1, there are two datasets from two very different data distributions. As can be seen, the first one coincides manifold-assumption and fits graph-based SSL techniques, and the second one coincides cluster-assumption and large margin SSL techniques would be more preferable. Such observations, however, could not be explicitly caught by traditional meta-features, especially when the two data sets share with similar numbers of features, instances and label proportions. In con-

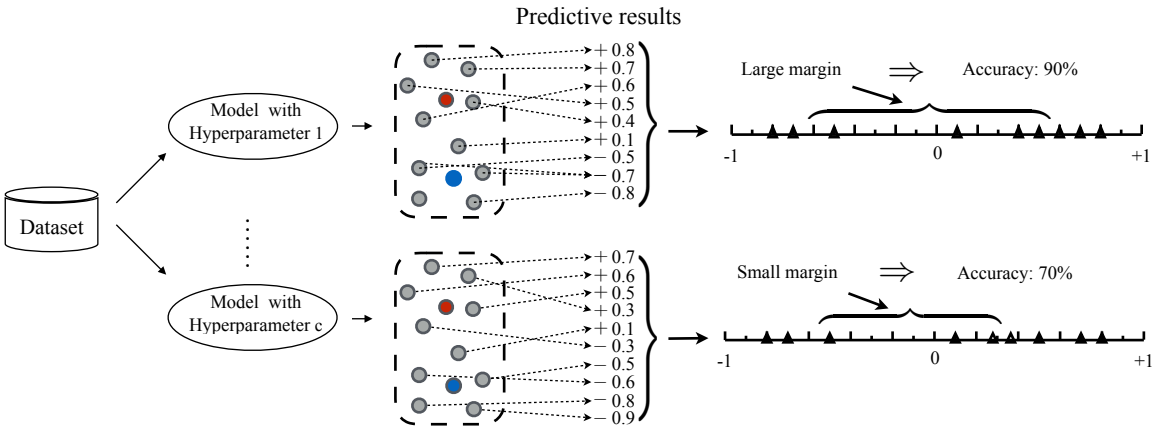

Figure 2: Illustration for hyperparameter selection by large margin separation

trast, the meta-features extracted by unsupervised clustering algorithms turn out to be more appropriate. More specifically, the meta-features highlight that spectral clustering is more coincident with the first dataset while k-means is more coincident with the second one. Table 1 summarizes the list of meta-features used in AUTO-SSL.

Large Margin Hyperparameter Selection

Meta-learning is good at suggesting some instantiations of the SSL techniques which are likely to perform well in a quick manner, whereas it could not provide fine-grained performance. In contrast, hyperparameter optimization is good at fine-tuning performance over time, but it is much slower. The above two steps are complementary and help each other.

For automated supervised learning, it is now widely accepted that tree-based Bayesian optimization approach, e.g., SMAC (Hutter, Hoos, and Leyton-Brown 2011) achieves promising performance, which fine tunes the hyperparameter based on its accelerated process of k-fold cross-validation.

However, it is not available to fine tune hyperparameters in SSL since the labeled examples are often too few to afford a reliable model selection. What is worse, the performance may be even degenerated compared to direct supervised learning with only labeled examples, as reported in many SSL studies (Li and Zhou 2015). It is desired to choose a good-quality parameter and thus alleviate performance degeneration.

In this paper, we propose to present a large margin separation method for hyperparameter optimization in AutoSSL. The basic idea is that once a certain hyperparameter owns a high quality, its predictive results on the unlabeled data may have a large margin separation and vice versa (Vapnik 1999). Intuitively, Figure 2 illustrates the idea of large margin hyperparameter selection. It can be seen that when one hyperparameter owns a better accuracy on the unlabeled data, it may also have a larger margin separation (Vapnik 1999). In other words, large margin separation may help judge the quality of hyperparameter. Based on this recognition, we should exploit the large margin hyperparameters while keeping the small margin hyperparameter (which might be risky) to be rarely exploited.

Formally, suppose we perform a classical SSL method S with a set of hyperparameters $\Theta = \{\theta_1, \cdots, \theta_r\}$, and collect the corresponding predictive values $\boldsymbol{F} = \{\boldsymbol{f}^{\theta_1}, \cdots, \boldsymbol{f}^{\theta_r}\}$ where $\boldsymbol{f}^{\theta_i} = [f^{\theta_i}(\boldsymbol{x}_{l+1}); \ldots; f^{\theta_i}(\boldsymbol{x}_{l+u})], i = 1, \cdots, r$ refers to the prediction on unlabeled data based on the hyperparameter θ_i. Let $\mathcal{P} = \{j | f^{\theta_i}(\boldsymbol{x_j}) > 0, j = l+1, \cdots, l+u\}$ and $\mathcal{N} = \{j | f^{\theta_i}(\boldsymbol{x_j}) \leq 0, j = l+1, \cdots, l+u\}$ denote the sets of predictive positive an negative instances. large margin separation is to select the high-quality hyperparameter of SSL model, such that the margin of the predictive results is maximized,

$$\theta^\star \in \arg\max_{\theta_k \in \Theta} |\frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} f^{\theta_k}(\boldsymbol{x_i}) - \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}} f^{\theta_k}(\boldsymbol{x_j})| \quad (1)$$

The large margin separation principle is simple, and rather effective to exclude poor quality parameters. Since it only runs the model once, it is much faster to compute than the expensive standard cross-validation procedure.

Finally, once the candidate best hyperparameter $\theta^\star$ is selected, model evaluation is conduct to finalize the best model for unseen data. In order to select a robust model that will not be outperformed by direct supervised learning with only labeled data, both the hyperparameters of SSL techniques and baseline supervised learning algorithm are considered for comparison, and we employ K-fold cross-validation (Kohavi 1995) to be an estimate of the model performance. Formally, let $D = \{\{\boldsymbol{x}_i, y_i\}_{i=1}^{l}, \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}\}$ be a training set which is split into K cross-validation folds $\{D_{train}^1, \cdots, D_{train}^K\}$ and $\{D_{valid}^1, \cdots, D_{valid}^K\}$ such that $D_{train}^i = D_{train} \backslash D_{valid}^i$ for $i = 1, \cdots, K$. AUTO-SSL selects model that minimizes the empirical loss (Feurer et al. 2015) on the dataset:

$$M^\star \in \arg\min_{M \in S_\theta^* \cup A_\lambda, \lambda \in \Lambda} \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}(M, D_{train}^i, D_{valid}^i) \quad (2)$$

As can be realized in Eq. (2), when SSL techniques do not perform as well as baseline supervised algorithm, AUTO-SSL still performs robustly, since in Eq. (2), both the hyperparameters

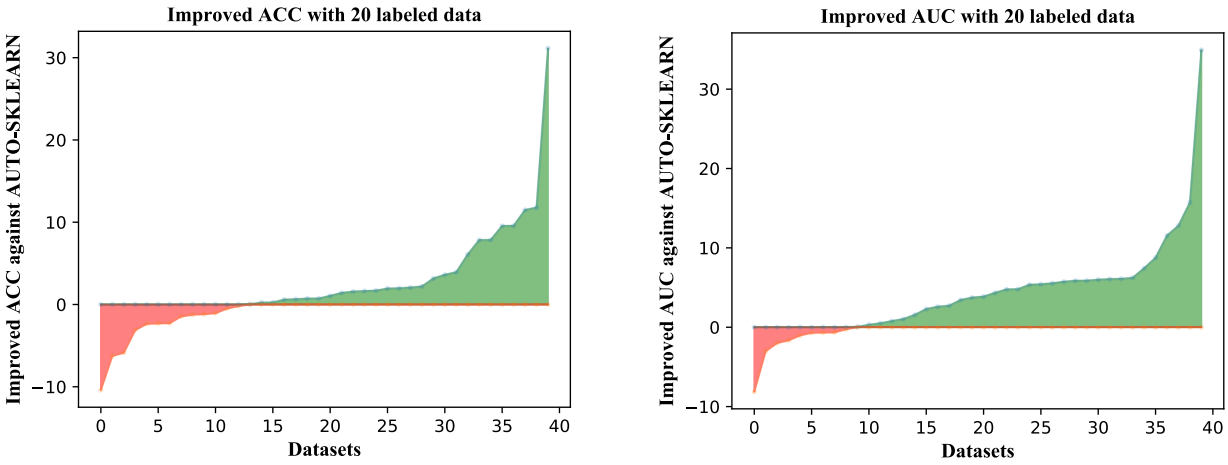

Figure 3: Improved performance of AUTO-SSL against AUTO-SKLEARN

of SSL techniques and baseline supervised learning algorithm are involved and considered.

Experiments

To explore the ability of AUTO-SSL in realistic settings without domain-specific parameter tuning, extensive experiments are conducted on a broad range of 40 datasets that cover diverse domains including business (blood), life (echocardiogram), physical (cylinder-bands), social (house-votes), finance (credit-approval), computer (spambase), etc. Detail information of datasets please refer to the supplementary file.

AUTO-SSL is compared with the two state-of-the-art supervised counterparts and two classic SSL techniques.

- SVM: The baseline supervised learning method SVM (Vapnik 1999) on only labeled examples, a discriminative classifier formally defined by a separating hyperplane. Its hyperparameter, the penalty factor C_{svm} is selected from 7 configurations $\{2^{-3}, 2^{-2}, 2^{-1}, 2^{0}, 2^{1}, 2^{2}, 2^{3}\}$.
- AUTO-SKLEARN: The state-of-the-art automated machine learning system (Feurer et al. 2015), which shows promising performance on supervised learning problems. The running time is set to one minute which is sufficient to ensure AUTO-SKLEARN system to finish successfully.
- CMN: The classic graph-based SSL technique (Zhu, Ghahramani, and Lafferty 2003). Without sufficient domain knowledge of graph construction, k-nearest neighbor graph is recognized as a good candidate graph (Zhu 2008) and the value of k is the hyperparameter selected from 3 configurations $\{5, 7, 9\}$.
- TSVM: The classic large margin SSL technique (Joachims 1999) seeks the largest separation between labeled and unlabeled data. The penalty factor C_{TSVM} is its hyperparameter selected from the same configurations of SVM.

For each data set, a series of limited labeled instances $(20, 40, 60, 80, 100)$ are considered, where labeled data are randomly chosen, and the remaining data are used as unlabeled data. Each dataset is split for 20 times and average performance in terms of accuracy and area Under the ROC Curve (AUC) is reported. In the meta-learning of AUTO-SSL, performance data of two classical SSL techniques are collected on a large amount of testing data, and we use paired t-tests at 95% significance level to select the best learning algorithm for one dataset. In testing phrase, each time one empirical dataset is treated as target dataset, while the other ones are used as the training datasets of automated SSL. For target dataset, 6NN is employed to the supervised learner in meta-learning. We have tried many other supervised learners in AUTO-SKLEARN (such as linear model, random forest, SVM) and k-nearest neighbor works the best.

AUTO-SSL vs. AUTO-SKLEARN

In comparison to AUTO-SKLEARN, Tables 2-3 show the detail comparison results and Figure 3 shows improved accuracy and AUC of AUTO-SSL against AUTO-SKLEARN with 20 labeled data. More detail results with similar observation please refer to our supplementary file. In Figure 3, the green part indicates the amount of performance improvement, and the red part indicates the amount of performance degradation. Specifically, let $Per(\text{AUTO-SSL})$ and $Per(\text{AUTO-SKLEARN})$ be the performance of AUTO-SSL and AUTO-SKLEARN, respectively, and the values $\big(Per(\text{AUTO-SSL}) - Per(\text{AUTO-SKLEARN})\big)$ on 40 datasets are plotted in increasing order. As can be seen, in the SSL setting, particularly when labeled data are few, AUTO-SSL works clearly better than direct supervised AutoML. These above results demonstrate the effectiveness of AUTO-SSL, i.e., AUTO-SSL with the use of unlabeled instances work better than the automated supervised learning solution.

AUTO-SSL vs. Classical SSL Techniques

In comparison to classical SSL techniques, Tables 2-3 show the comparison results and Figures 4 show the results of improved ACC and AUC against SVM on 20 labeled data, respectively. More detail results behaved similarly refer to our supplementary file. Moreover, Figure 5 shows the average ACC and AUC of the comparison experiment on different

Table 2: Accuracy % (mean ± std) on 20 labeled instances for the compared methods. For the compared methods, if the performance is significantly better/worse than the baseline SVM, the corresponding entries are then bolded/underlined. The average performance is listed for comparison. ASK is short for AUTO-SKLEARN and ASSL is short for AUTO-SSL.

Data	SVM	TSVM	CMN	ASK	ASSL
1	74.9±3.2	67.6±2.6	70.9±5.0	72.5±4.5	73.1±4.1
2	94.8±2.3	**96.5±0.2**	95.8±0.7	93.5±2.2	95.1±2.2
3	62.4±5.0	55.0±5.4	59.9±5.8	58.7±4.1	60.7±5.8
4	58.6±4.7	59.4±3.2	**63.8±2.0**	59.8±4.0	**63.0±2.9**
5	80.5±3.1	75.6±4.5	79.0±3.6	80.6±3.0	78.3±3.6
6	77.5±2.6	**81.9±3.1**	79.9±13.4	70.5±6.0	**80.0±2.5**
7	76.6±5.8	**80.6±1.4**	78.0±3.4	78.4±5.9	76.1±5.4
8	70.6±4.3	68.6±3.9	70.5±5.2	71.6±6.3	71.1±4.5
9	91.3±3.9	91.6±1.1	89.6±1.8	89.7±6.2	91.9±1.7
10	66.2±3.7	64.2±3.6	**69.9±1.4**	**69.2±3.3**	**69.9±1.4**
11	80.9±4.3	**85.9±5.2**	78.0±6.8	78.8±7.0	82.8±5.4
12	96.9±1.5	**99.7±0.1**	96.9±2.0	87.9±5.8	**99.4±0.8**
13	57.2±2.3	50.2±3.7	56.0±2.3	56.3±4.0	56.6±2.7
14	77.8±3.3	78.1±0.4	72.2±4.2	76.4±6.7	75.2±5.9
15	60.5±3.7	58.4±6.3	52.9±2.4	61.2±10.5	58.1±5.0
16	56.9±2.9	52.9±2.3	59.3±5.1	**61.3±5.3**	55.4±3.0
17	68.8±5.0	63.9±8.8	56.6±3.9	**74.4±9.3**	64.0±8.0
18	84.7±3.7	85.7±3.6	86.8±4.6	86.0±4.4	85.0±3.8
19	64.0±4.8	59.6±5.5	65.0±3.0	63.0±4.1	64.6±3.7
20	61.6±5.9	52.7±3.9	56.4±2.7	58.6±4.9	57.1±6.0
21	97.6±1.2	95.1±0.3	**99.7±0.1**	90.1±4.3	**99.7±0.1**
22	99.5±0.2	**99.9±0.1**	**99.9±0.0**	92.0±5.2	**99.8±0.2**
23	80.0±4.0	77.1±6.3	65.1±4.1	76.2±7.6	79.8±4.2
24	61.8±5.6	64.1±6.8	63.2±4.1	61.1±6.2	63.1±4.2
25	71.4±2.4	73.8±7.1	**72.9±0.4**	72.7±3.7	**72.8±0.5**
26	99.5±0.8	96.0±0.3	**100.0±0.0**	99.3±0.9	98.1±1.9
27	70.7±7.9	64.0±5.0	**99.5±0.8**	68.4±11.2	**99.5±0.8**
28	74.1±5.7	71.7±3.8	64.3±12.7	72.3±3.7	73.7±5.9
29	78.6±3.3	74.3±3.6	75.8±3.5	76.4±5.1	77.5±4.0
30	77.4±4.1	**80.9±4.8**	77.4±4.5	79.8±4.7	79.6±5.1
31	77.0±3.8	77.3±3.3	73.3±4.8	75.1±4.8	75.8±4.4
32	90.2±2.5	88.6±2.6	88.4±1.9	89.2±4.4	89.5±3.1
33	73.5±4.5	57.1±6.7	60.0±4.0	68.0±7.2	69.9±7.4
34	69.2±3.7	70.2±3.2	66.2±2.0	66.4±4.6	68.1±3.6
35	54.9±3.1	**57.0±1.8**	**65.0±3.4**	**64.1±4.2**	**61.8±5.8**
36	62.5±3.9	61.2±2.4	**68.7±2.1**	**68.0±3.2**	61.7±3.5
37	74.4±5.6	**81.5±3.1**	77.1±5.0	72.1±7.6	**79.9±5.7**
38	92.9±1.9	**97.8±0.1**	69.3±21.3	87.6±5.8	93.7±2.5
39	85.3±4.4	**93.0±3.7**	**92.3±4.1**	78.2±6.8	**90.0±4.9**
40	85.9±3.4	**93.1±1.8**	79.8±6.6	85.0±5.0	85.8±4.0
Ave	76.0±12.5	75.0±14.8	74.9±13.6	74.8±10.9	76.9±13.4

Table 3: AUC % (mean ± std) on 20 labeled instances for the compared methods. For the compared methods, if the performance is significantly better/worse than the baseline SVM, the corresponding entries are then bolded/underlined. The average performance is listed for comparison. ASK is short for AUTO-SKLEARN and ASSL is short for AUTO-SSL.

Data	SVM	TSVM	CMN	ASK	ASSL
1	70.9±4.1	63.2±4.7	64.7±5.2	59.5±10.0	65.4±5.5
2	98.7±1.3	99.1±0.1	99.1±0.1	98.9±0.2	98.6±1.3
3	67.5±6.4	55.7±6.7	65.3±5.8	58.4±7.7	64.6±7.6
4	59.4±5.4	58.2±3.8	60.9±4.7	57.2±5.4	59.9±4.7
5	68.3±11.2	69.8±10.1	67.4±7.5	56.3±8.6	67.9±8.2
6	86.2±2.2	84.2±3.3	**96.3±1.9**	78.8±7.5	**94.6±4.4**
7	82.6±5.2	**86.3±1.8**	82.9±3.7	83.1±4.2	83.8±4.6
8	75.5±4.1	70.9±4.1	71.7±4.6	73.1±8.6	73.2±5.6
9	97.2±2.6	97.8±0.6	94.7±0.8	94.9±3.8	97.2±2.4
10	66.9±5.6	59.9±6.2	62.9±5.7	55.8±9.1	61.6±6.6
11	81.1±6.7	**90.8±4.9**	83.7±4.9	80.3±10.9	**86.4±5.4**
12	99.5±0.3	**99.9±0.0**	**99.7±0.1**	93.7±5.7	99.7±0.2
13	53.0±5.8	48.8±4.5	51.9±3.7	55.7±5.8	52.7±4.9
14	84.3±3.8	84.6±1.1	80.2±3.1	81.4±9.4	82.4±4.1
15	64.5±5.5	63.0±8.5	54.9±2.7	61.5±10.5	59.8±7.1
16	49.8±3.2	50.6±4.1	**53.3±3.9**	50.2±3.2	50.6±4.1
17	75.3±5.6	68.5±9.3	59.3±4.4	78.6±10.0	70.4±10.3
18	89.5±3.2	87.6±3.6	86.7±4.8	90.3±5.8	88.3±4.3
19	64.9±5.3	55.3±8.3	60.5±6.4	55.2±5.8	59.9±6.4
20	65.5±7.1	50.5±4.6	58.6±4.4	58.3±7.0	57.6±7.1
21	99.8±0.1	97.7±0.7	**99.9±0.0**	96.0±4.8	**99.9±0.0**
22	99.9±0.0	**99.9±0.0**	**99.9±6.1**	96.2±4.3	**99.9±6.1**
23	86.4±4.1	82.9±7.2	87.7±2.7	81.3±12.1	86.7±5.5
24	64.2±6.4	67.9±7.6	67.6±4.0	61.7±8.4	67.5±4.9
25	66.9±5.4	**78.0±14.5**	**75.7±3.9**	69.9±11.7	**77.3±5.2**
26	99.9±0.0	98.5±0.4	100.0±0.0	99.9±0.0	99.2±0.7
27	72.4±10.5	60.7±5.9	**99.2±1.2**	64.3±13.2	**99.2±1.2**
28	67.0±5.5	69.4±3.5	64.8±7.6	63.6±9.0	66.1±7.4
29	86.5±3.4	80.3±3.9	83.0±3.4	81.5±6.8	83.1±4.5
30	83.8±3.5	85.6±5.5	82.9±2.4	85.5±3.7	84.4±4.5
31	85.2±4.0	85.1±3.2	81.8±4.3	78.5±6.0	83.9±4.4
32	96.5±0.6	95.5±1.8	95.5±0.8	95.0±3.1	95.5±1.4
33	81.7±5.1	61.4±10.9	67.2±5.2	72.1±9.3	77.6±8.6
34	71.5±6.1	**75.4±4.8**	67.1±4.3	64.2±11.6	73.0±5.9
35	50.4±3.6	50.2±2.7	50.9±3.0	50.7±2.3	50.1±2.7
36	62.0±6.2	56.9±4.4	56.4±4.1	56.1±8.2	59.5±6.0
37	81.7±6.1	**88.7±2.4**	84.8±3.6	77.9±9.8	84.0±6.1
38	98.2±0.8	**99.7±0.0**	**99.2±0.1**	95.0±6.4	**99.3±1.0**
39	93.9±2.7	**95.9±2.6**	**99.0±0.9**	86.2±8.1	**99.0±0.9**
40	97.2±1.0	**97.9±0.7**	96.5±0.9	92.1±4.9	96.9±0.9
Ave	78.6±14.8	76.8±16.8	77.8±16.5	74.7±15.4	78.9±16.0

numbers of labeled data, respectively. As can be seen, AUTO-SSL achieves better performance than classical SSL techniques in most cases, which further justifies the effectiveness of automated SSL.

AUTO-SSL vs. Direct Supervised Learning

Finally, we extensively discuss the comparison with direct supervised learning, i.e., baseline SVM with only labeled data. According to results in Tables 2-3 and Figure 4, SVM obtains highly competitive performance with few labeled examples. Classic SSL techniques are not easy to beat SVM, and often significantly degenerate the performance, while AUTO-SSL obtains more reliable performance.

Table 4 further gives the counts of Win/Tie/Loss against SVM with different number of labeled instances for all compared methods and AUTO-SSL. The results further justify that AUTO-SSL obtains quite good performance. Specifically, in terms of ACC, AUTO-SSL rarely degenerates the performance

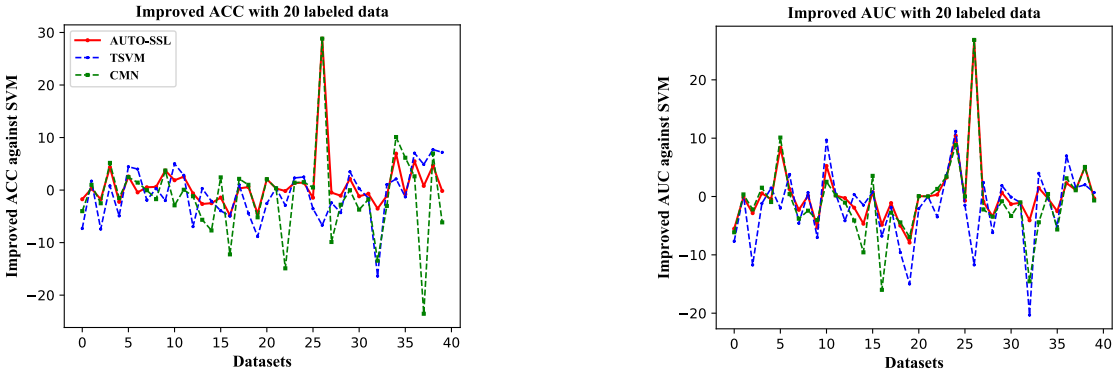

Figure 4: Improved performance against SVM on 40 datasets

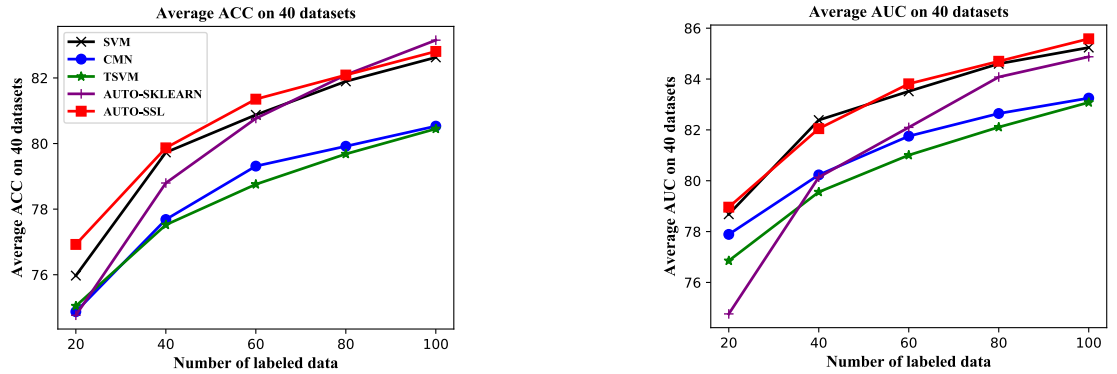

Figure 5: Average performance with different numbers of labeled insatnces on 40 datasets

Table 4: The counts of Win/Tie/Loss against SVM with respect to all compared methods. 'Win/Tie/Loss' counts the datasets for which the compared methods is statistically significantly better/comparable/significantly worse than SVM (paired t-tests at 95% significance level). The method with the smallest number of losses against SVM is bolded.

	Method	Number of labeled instances 20	40	60	80	100
ACC	TSVM	12/15/13	13/11/16	13/10/17	12/11/17	13/11/16
	CMN	10/16/14	13/9/18	13/9/18	11/10/19	12/9/19
	ASSL	**11/25/4**	**13/22/5**	**13/24/3**	**10/29/1**	**11/26/3**
AUC	TSVM	10/14/16	11/13/16	9/19/12	12/13/15	12/16/12
	CMN	9/14/17	9/11/20	9/11/20	8/12/20	10/10/20
	ASSL	**8/24/8**	**8/27/5**	**9/28/3**	**10/26/4**	**9/28/3**

and is the most robust among all the compared methods. Similar conclusion can also be reached in terms of AUC.

Conclusion

Automated Machine Learning (AutoML) attempts to build an appropriate machine learning model for unseen data set in an automatic manner (without human intervention). Previous work on AutoML typically focuses on supervised learning problems. In many applications, however, other problems such as semi-supervised learning (SSL) problems are widespread. Existing AutoML techniques could not well address such difficulties. In this paper, we present an automated SSL system (AUTO-SSL). We first consider meta-learning that transforms automated SSL as a supervised learning and then exact appropriate features for data sets by not only traditional meta-features but also unsupervised learning. To alleviate performance deterioration, which is crucial for SSL, we design a large margin principle to avoid low-quality hyper-parameters, and save considerable computation overhead compared to direct cross-validation. Extensive empirical results show that our proposal outperforms classical SSL techniques and state-of-the-art AutoML system AUTO-SKLEARN, in addition clearly improves the reliability of SSL.

Our system could be improved by removing some shortcomings. For instance, we have not yet tackled SSL with multi-class problems or very large-scale datasets. Moreover, applying our system to deep learning models and deriving a good representation (meta-features) of datasets would be a worth-studying future work.

Acknowledgements

The authors want to thank the reviewers for helpful comments and suggestions. This research was supported by the National

Key R&D Program of China (2018YFB1004301) and the National Natural Science Foundation of China (61772262).

References

Bardenet, R.; Brendel, M.; Kégl, B.; and Sebag, M. 2013. Collaborative hyperparameter tuning. In *Proceedings of the 30th International Conference on Machine Learning*, 199–207.

Brazdil, P.; Carrier, C. G.; Soares, C.; and Vilalta, R. 2008. *Metalearning: Applications to data mining*. Springer Science & Business Media.

Chapelle, O.; Schölkopf, B.; and Zien, A. 2006. *Semi-Supervised Learning*. Cambridge, MA: MIT Press.

Davies, D. L., and Bouldin, D. W. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2):224–227.

Feurer, M.; Klein, A.; Eggensperger, K.; Springenberg, J. T.; Blum, M.; and Hutter, F. 2015. Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems 28*. 2962–2970.

Feurer, M.; Springenberg, J. T.; and Hutter, F. 2015. Initializing bayesian hyperparameter optimization via meta-learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 1128–1135.

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*.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter* 11(1):10–18.

Hutter, F.; Hoos, H. H.; and Leyton-Brown, K. 2011. Sequential model-based optimization for general algorithm configuration. In *Proceedings of International Conference on Learning and Intelligent Optimization*, 507–523.

Jain, A. K., and Dubes, R. C. 1988. *Algorithms for clustering data*. Prentice-Hall, Inc.

Jain, A. K. 2010. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters* 31(8):651–666.

Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, 200–209.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 9th AAAI Conference on Artificial Intelligence*, volume 14, 1137–1145.

Krijthe, J. H., and Loog, M. 2015. Implicitly constrained semi-supervised least squares classification. In *Advances in 14th International Symposium on Intelligent Data Analysis*, 158–169.

Lemke, C.; Budka, M.; and Gabrys, B. 2015. Metalearning: a survey of trends and technologies. *Artificial Intelligence Review* 44(1):117–130.

Li, Y.-F., and Zhou, Z.-H. 2015. Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(1):175–188.

Li, Y.-F.; Kwok, J.; and Zhou, Z.-H. 2016. Towards safe semi-supervised learning for multivariate performance measures. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 1816–1822.

Li, Y.-F.; Wang, S.-B.; and Zhou, Z.-H. 2016. Graph quality judgement: a large margin expedition. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 1725–1731.

Li, Y.-F.; Zha, H.-W.; and Zhou, Z.-H. 2017. Learning safe prediction for semi-supervised regression. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2217–2223.

Maulik, U., and Bandyopadhyay, S. 2002. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(12):1650–1654.

Michie, D.; Spiegelhalter, D. J.; Taylor, C. C.; and Campbell, J. 1994. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood.

Sun, L. 2016. Automl challenge: System description. In *Proceedings of AutoML workshop on the 33rd International Conference on Machine Learning*.

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.

Vapnik, V. N. 1999. An overview of statistical learning theory. *IEEE Transactions on Neural Networks* 10(5):988–999.

Von Luxburg, U. 2007. A tutorial on spectral clustering. *Statistics and Computing* 17(4):395–416.

Zhou, Z.-H., and Li, M. 2010. Semi-supervised learning by disagreement. *Knowledge and Information Systems* 24(3):415–439.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2004. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*. 321–328.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning*, 912–919.

Zhu, X. 2008. Semi-supervised learning literature survey. Technical report, University of Wisconsin–Madison, Madison, WI.