Learning to Diversify for Ensembling of An Amount of Classifiers

Chun Yang

University of Science and Technology Beijing

Xu-Cheng Yin (✉ xuchengyin@ustb.edu.cn)

University of Science and Technology Beijing

Article

Keywords: learned classifiers, classifier ensemble, learning to diversify

Posted Date: August 23rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-799035/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Learning to Diversify for Ensembling An Amount of Classifiers

Chun Yang*

CHUNYANG@USTB.EDU.CN

*Department of Computer Science and Technology, University of Science and Technology Beijing
Beijing 100083, China*

Xu-Cheng Yin*

XUCHENGYIN@USTB.EDU.CN

*Department of Computer Science and Technology, University of Science and Technology Beijing
Institute of Artificial Intelligence, University of Science and Technology Beijing
Beijing 100083, China*

Editor: ???

Abstract

Not only common man but also intelligent machine always merge all available decisions to solve problems. However, given an amount of learned classifiers, how to select and combine diverse classifiers for machine is still a grand challenge in the literature for decades of history. In this paper, we introduce a novel approach for classifier ensemble, learning to diversify, which learns to adaptively combine classifiers by considering both accuracy and diversity. Specifically, our approach, learning to diversify via weighted kernels, performs classifier combination by optimizing a direct but simple criterion: maximizing ensemble accuracy and adaptive diversity simultaneously by minimizing a loss function. Given a measure formulation, the diversity is calculated with weighted kernels, i.e., the diversity is measured on the component classifiers' outputs which are kernelized and weighted. Moreover, we propose an iterative training algorithm for weights optimization, where this loss function is iteratively minimized by estimating the kernel weights in conjunction with the classifier weights. Extensive experiments on a variety of classification benchmark datasets show that the proposed approach consistently outperforms state-of-the-art ensembles.

1. Introduction

Combining available decision experts to solve a specific problem is a natural ability of man. Similarly, in most classification systems, one application solution for machine is to merge all learned classifiers. For example, an ensemble machine learning model is employed to estimate anti-SARS-CoV-2 activities (KC et al., 2021). There are many famous ensembles (e.g., Bagging (Breiman, 1996), Boosting (Freund and Schapire, 1996, 1997), Stacking (Wolpert, 1992), Random Forests (Breiman, 2001) and neural network ensembles (Hansen and Salamon, 1990; Zhou et al., 2002)) and also a variety of recent ensemble methods (Jaffe et al., 2016; Sun and Zhou, 2018; Zhou and Feng, 2019; Fernandes et al., 2020), many of which have been widely applied in numerous real-world applications. Some previous researches show that the performance of the classifier ensemble relies on not only the accuracy but also the diversity of component classifiers (Hansen and Salamon, 1990; Opitz and Shavlik, 1996; Krogh and Sollich, 1997; Zhou, 2012). Generally speaking, ensemble of diverse classifiers can allow us to get higher accuracy, which is often not achievable by a single model.

Consequently, how to diversely combine classifiers plays an important role and becomes a main topic in ensemble learning. In this paper, we focus on choosing a subset of classifiers that are diverse. Namely, given an amount of pre-constructed component classifiers, our method can select and combine a subset of diverse classifiers to construct a more accurate classification system.

Given a diversity measure formulation, most conventional diversity-based ensemble methods combine classifiers by calculating this fixed measure directly (Martínez-Muñoz and Suárez, 2006; Martinez-Munoz et al., 2009). Several diversity measures are extensively acknowledged, e.g., *disagreement*, *Q-statistics*, *double fault*, and *Kappa* (Skalak, 1996; Shipp and Kuncheva, 2002; Kuncheva and Whitaker, 2003). Obviously, the diversity is useful, and an ensemble should combine diverse component classifiers for improving accuracy; in an extreme case, ensembling all same classifiers has no any improvement. However, how to adaptively use diversity and suitably optimize combination is still unclear and challenging in the literature. Up to now, many research evidences showed that accuracy did not monotonically increase with diversity using various diversity measures, and seemingly verified that there was no clear correlation between diversity and accuracy in ensembles (Ruta and Gabrys, 2001; Kuncheva and Whitaker, 2003; Saitta, 2006). Some researchers even argued that in practice it was not possible to define and use a diversity measure that is clearly linked to the ensemble accuracy (Saitta, 2006). The point is “how to use the useful diversity”. Since Krogh and Vedelsby presented the error-ambiguity decomposition for ensembles (Krogh and Vedelsby, 1995), “how to use the useful diversity” seems to become a “fast knot” in ensemble learning for decades of history.

From a different view to conventional diversity-based ensembles, firstly, we argue that the diversity (called as **adaptive diversity with data**) should be correlated to the accuracy, not directly calculated on the training data but adaptively measured on the data. Currently, given data samples, all existing diversity measures are calculated on the validation set without considering the data distribution, i.e., most datasets contain an overwhelming number of easy examples and a small number of hard examples, although there are a few related descriptions. For example, Ruta and Gabrys observed that certain diversity measures (e.g., Q statistics) performed well on the artificial data, but was inadequate for realistic datasets (Ruta and Gabrys, 2001). Because of hard samples, the diversity uniformly calculated on the original data is always not suitable for diversity-based classifier ensembles, and the diversity should be measured on another space derived from the original data. In our method, the diversity is computed on the data via weighted kernels and emphasized on hard samples. This diversity is called as “adaptive diversity with data” in this paper. Secondly, we also believe that diversity-based ensembles, i.e., ensembles with both accuracy and (adaptive) diversity, can improve performance of the final classification system ¹.

Consequently, we introduce a new term in classifier ensemble, **learning to diversify** ², which should learn to adaptively select and combine a subset of classifiers by considering

-
1. In our paper, for simplifying, combining classifiers with both accuracy and (adaptive) diversity is also called as “adaptively using diversity for classifier ensemble” or “adaptively combining classifiers”.
 2. In information retrieval, given a list of documents or web pages, learning to diversify is to learn to rank and present a diverse set of results by considering both relevance and diversity (Radlinski et al., 2008, 2009). Similarly, here given a list of available component classifiers, learning to diversify in classifier ensemble is to learn to select and combine a diverse set of classifiers by considering both accuracy and diversity.

both accuracy and diversity, given a list of available (pre-constructed) component classifiers. There are two main essential tasks of this **learning to diversify** strategy: (1) First and obviously, same as diversity-based ensembles, the finally selected classifiers in an ensemble should be more diverse than the general combination strategy, e.g., averaging; (2) second but more importantly, according to different data under various situations, **learning to diversify** should adaptively measure the diversity with a given formulation, i.e., adaptive diversity with data via focusing on hard samples.

Specifically, within a linear combination framework for classifier ensemble, we propose a unified learning-to-diversify approach, Learning TO Diversify via Weighted Kernels (**L2DWK**). For the first task, we propose to learn **classifier weights** by optimizing a direct but simple criterion: maximizing the accuracy and the diversity simultaneously. For the second task, different from the conventional methods where the diversity is directly calculated on the component classifiers’ outputs from the validation samples, we seek the adaptive diversity with data in the view of the data space (with kernelizing) and the data distribution (with weighting). The diversity in our approach is adaptively evaluated with a set of weights (called as **kernel weights**) on the samples’ outputs within a kernel, i.e., the classifiers’ outputs are converted into another space with weighted kernels, and the **kernel weights** are learned automatically according to the contribution of validation samples. Furthermore, we combine these two tasks into a unified framework with an iterative training algorithm. In each iteration of this algorithm, first, given the kernel and weights (**kernel weights**), the **classifier weights** for combination are learned by minimizing a convex loss function which maximizes accuracy and diversity of the ensemble simultaneously. Then, given the current classifier weights, the **kernel weights** are automatically updated with a dynamically damped learning trick. This two-step optimization is then repeated until convergence or the maximum number of iterations is exceeded. We argue that this diversity via weighted kernels is adaptive and robust to data distributions in different situations, and thereby classifier ensemble based on **learning to diversify** can produce a subset of important and diverse classifiers. To some extent, we hopefully arrive the above point: “**to use the useful diversify**” with this learning strategy.

2. Notations

Let the validation dataset be $\text{Val} = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$, where y_i is the (ground truth) label of sample x_i and all the outputs are in C classes $\{\omega_1, \omega_2, \dots, \omega_C\}$ for a generic multi-class classification task. The base classifiers $H = \{h_1, h_2, \dots, h_L\}$ of an ensemble are pre-constructed on the original training set, and an output of a base classifier h_j on sample x_i for the k^{th} class is $h_j^k(x_i)$. The decision of the base classifier is conducted with

$$h_j(x_i) = \arg \max_{k=1, \dots, C} h_j^k(x_i). \quad (1)$$

Given each base classifier h_j , together with its weight w_j , we define the vector of **classifier weights** as $\mathbf{w} = [w_1, w_2, \dots, w_L]$, where $\sum_{j=1}^L w_j = 1$ and $w_j \geq 0$. In this paper, we focus on linear combination of classifiers. By taking a weighted vote among the base classifiers, for the ensemble h_{ens} , there is

$$h_{ens}^k(x_i) = \frac{1}{L} \sum_{j=1}^L w_j h_j^k(x_i). \quad (2)$$

The final classification decision for the ensemble is to choose the class label receiving the largest weighted vote, i.e.,

$$h_{ens}(x_i) = \arg \max_{k=1,\dots,C} h_{ens}^k(x_i). \quad (3)$$

Instead of the original output, the oracle output $\mathbf{O}$ (Tang et al., 2006) of the ensemble is used for optimization in ensembling classifiers. The oracle output is a $N \times L$ matrix, the element of which is

$$O_{ij} = \begin{cases} 1 & h_j(x_i) = y_i \\ -1 & h_j(x_i) \neq y_i \end{cases} \quad (4)$$

The main notations used in this paper are summarized as follows:

- 1) N : the number of total samples.
- 2) L : the number of total base classifiers (pre-constructed component classifiers).
- 3) O_{ij} : the oracle output of the base classifier h_j on sample x_i .
- 4) $\mathbf{O}_j$: the oracle output of the base classifier h_j on the whole sample set, i.e., $\mathbf{O}_j = [O_{1j}, O_{2j}, \dots, O_{Nj}]^T$.
- 5) m_i : the margin of sample x_i , and there is

$$m_i = \sum_{h_j(x_i)=y_i} w_j - \sum_{h_j(x_i) \neq y_i} w_j = \sum_{j=1}^L w_j O_{ij}. \quad (5)$$

- 6) $\mathbf{P}$: the average ‘‘accuracy’’ of classifiers on the training set. $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_L]^T$, where $\mathbf{P}_j$ is the accuracy of the j^{th} classifier. Here, in order to simplify the formula derivation, the accuracy is defined as

$$\mathbf{P} = \frac{1}{N} \mathbf{1}_{N \times 1}^T \mathbf{O}, \quad (6)$$

instead of the traditional one, i.e.,

$$\mathbf{P} = \frac{1}{2N} \mathbf{1}_{N \times 1}^T (\mathbf{O} + \mathbf{1}_{N \times L}). \quad (7)$$

- 7) $div(\mathbf{w})$: the diversity of the ensemble with classifier weights $\mathbf{w}$. If we use the pairwise diversity, $div()$ can be calculated by averaging with

$$\begin{aligned} div(\mathbf{w}) &= \mathbf{w}^T \mathbf{D} \mathbf{w}, \\ \mathbf{D} &= f_D(\mathbf{O}^T \mathbf{O}, \mathbf{1}_{N \times 1}^T \mathbf{O}) \end{aligned} \quad (8)$$

where $\mathbf{D}$ is the diversity matrix of base classifiers, and $\mathbf{D}$ has a functional relationship $f_D()$ with $\mathbf{O}^T \mathbf{O}$ and $\mathbf{1}^T \mathbf{O}$. For example, the diversity matrixes with a common pairwise diversity measure (*disagreement* (Skalak, 1996)) is shown below. The diversity matrix with *disagreement* (**dis**) is

$$\mathbf{D}_{dis} = \frac{1}{2N} (N \mathbf{1}_{L \times L} - \mathbf{O}^T \mathbf{O}). \quad (9)$$

This disagreement measure is defined based on the intuition that two component classifiers perform differently on the same validation data. That is to say, the disagreement (diversity) between two component classifiers is first measured on the evaluation samples, and diversity within the whole set of component classifiers is then calculated by averaging over all pairs of component classifiers. According to the formulation (Equation (9)), we can easily get that the diversity of the ensemble increases with the value of the disagreement measure.

3. Learning to Diversify

In diversity-based ensembles, the diversity is measured on the classifiers' outputs. We argue that by introducing kernels, i.e., classifiers' outputs are kernelized, the diversity measure via kernelizing can be adaptive and more useful in a variety of situations for different data. Moreover, we are also sure that by introducing weights, i.e., classifier's outputs are weighted, the diversity measure via weighting can be more representative for various data (especially for hard samples) with different distributions in diversity-based ensembles. Consequently, we expand the ensemble diversity via weighted kernels, and introduce a novel ensemble method, learning to diversity via weighted kernels. We hope that this new method can learn to select and combine a subset of classifiers by considering both accuracy and (adaptive) diversity.

3.1 Learning to Diversify via Weighted Kernels

We first summarize our Learning TO Diversify via Weighted Kernels (**L2DWK**) approach by giving a direct but simple optimization criterion. Given the ground truth labels of the classifiers' outputs $Y = \{y_1, y_2, \dots, y_N\}$ and classifier weights $\mathbf{w}$, the optimization criterion of one generic diversity-based classifier ensemble strategy (with accuracy-diversity trade-off) is (Yin et al., 2014)

$$\begin{aligned} \min_{\mathbf{w}, \alpha} & \text{loss}(Y, \mathbf{w}) - \lambda \text{div}(Y, \mathbf{w}), \\ \text{s.t. } & \mathbf{w} \succeq 0, \mathbf{1}^T \mathbf{w} = 1. \end{aligned} \quad (10)$$

where $\text{loss}()$ is the loss function of the classification error, $\text{div}()$ is the diversity of the ensemble, and λ is the diversity regularization control parameter.

Additionally given the kernel $\mathbb{k}$ ³ for the classifier' output and the kernel weights α , the new criterion in our **L2DWK** method becomes

$$\begin{aligned} \min_{\mathbf{w}, \alpha} & \text{loss}(Y, \mathbb{k}_\alpha, \mathbf{w}) - \lambda \text{div}(Y, \mathbb{k}_\alpha, \mathbf{w}), \\ \text{s.t. } & \mathbf{w} \succeq 0, \mathbf{1}^T \mathbf{w} = 1. \end{aligned} \quad (11)$$

Obviously, this criterion maximizes the accuracy and diversity of the ensemble simultaneously, and our method can learn to diversify via weighted kernels for classifier ensemble. Thus, learning to diversify is integrated into the original classifier combination problem (Equation (10)) through optimizing the accuracy and diversity loss function with weighted kernels. Selecting component classifiers and measuring the diversity are achieved automatically by optimizing this criterion (Equation (11)).

3. Kernelling is to map the data into a high dimensional feature space, where Linear, Gaussian and Polynomial kernels are typically used. For example, the **Linear Kernel** is given by the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ of variable (vector) pairs plus an optional constant $c \in \mathbb{R}$, $\mathbb{k}^{\text{linear}}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{y} + c$. The **Gaussian Kernel** is an example of the radial basis function kernel ($\sigma \in \mathbb{R}$), $\mathbb{k}^{\text{Gauss}}(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{2\sigma^2})$. The **Polynomial Kernel** is a non-stationary kernel, $\mathbb{k}^{\text{poly}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T \mathbf{y} + c)^m$, where $c \in \mathbb{R}$ and $m \in \mathbb{N}$. Detailed descriptions can be referred to the Supplementary Material of this paper.

Next, we present analysis and formulations of **L2DWK** in detail. For classifiers h_i and h_j , the products of $\mathbf{1}_{N \times 1}^T$ and $\mathbf{O}$, and $\mathbf{O}^T$ and $\mathbf{O}$ are kernelized, which are marked as ⁴

$$\begin{aligned}\mathbb{k}(\mathbf{1}_{N \times 1}^T \mathbf{O})_j &= \mathbb{k}(\mathbf{1}_{N \times 1}, \mathbf{O}_j), \\ \mathbb{k}(\mathbf{O}^T \mathbf{O})_{i,j} &= \mathbb{k}(\mathbf{O}_i, \mathbf{O}_j),\end{aligned}\tag{12}$$

where $\mathbf{O}_i$ and $\mathbf{O}_j$ are the i^{th} and j^{th} columns in the oracle output $\mathbf{O}$ of the ensemble, respectively. Consequently, $\mathbb{k}(\mathbf{1}_{N \times 1}^T \mathbf{O})$ is a vector of kernelized results, and $\mathbb{k}(\mathbf{O}^T \mathbf{O})$ is a matrix. These kernels are then weighted, and we get

$$\begin{aligned}\mathbb{k}_\alpha(\mathbf{1}_{N \times 1}^T \mathbf{O})_j &= \sum_k \alpha_k \mathbb{k}(1, \mathbf{O}_{kj}), \\ \mathbb{k}_\alpha(\mathbf{O}^T \mathbf{O})_{i,j} &= \sum_k \alpha_k \mathbb{k}(\mathbf{O}_{ki}, \mathbf{O}_{kj}),\end{aligned}\tag{13}$$

where $k = 1, 2, \dots, N$, and N is the number of data samples.

In our method, the average accuracy $\mathbf{P}$ (Equation (6)) is used as the measure for the loss from the classification error, i.e., the classification error loss can be simply defined as the minus of $\mathbf{P}$ with multiplying the classifier weights, i.e.,

$$loss(Y, \mathbf{w}) = -\mathbf{1}_{N \times 1}^T \mathbf{O} \mathbf{w}.\tag{14}$$

Via kernelling with the weighted kernel $\mathbb{k}_\alpha$, there is

$$loss(Y, \mathbb{k}_\alpha, \mathbf{w}) = -\mathbb{k}_\alpha(\mathbf{1}_{N \times 1}^T \mathbf{O}) \mathbf{w}.\tag{15}$$

Using the pairwise diversity for $div()$ (Equation (8)), we have

$$div(Y, \mathbf{w}) = \mathbf{w}^T f_D(\mathbf{O}^T \mathbf{O}, \mathbf{1}^T \mathbf{O}) \mathbf{w}.\tag{16}$$

Embedding the weighted kernel $\mathbb{k}_\alpha$, we will get

$$div(Y, \mathbb{k}_\alpha, \mathbf{w}) = \mathbf{w}^T f_D(\mathbb{k}_\alpha(\mathbf{O}^T \mathbf{O}), \mathbb{k}_\alpha(\mathbf{1}^T \mathbf{O})) \mathbf{w}.\tag{17}$$

Consequently, by combining the above two formulations (Equation (15) and (17)), the criterion (in Equation (11)) changes to

$$\begin{aligned}min_{\mathbf{w}, \alpha} loss(Y, \mathbb{k}_\alpha, \mathbf{w}) - \lambda div(Y, \mathbb{k}_\alpha, \mathbf{w}) &= \\ min_{\mathbf{w}, \alpha} -\mathbb{k}_\alpha(\mathbf{1}^T \mathbf{O}) \mathbf{w} - \lambda \mathbf{w}^T f_D(\mathbb{k}_\alpha(\mathbf{O}^T \mathbf{O}), \mathbb{k}_\alpha(\mathbf{1}^T \mathbf{O})) \mathbf{w}.\end{aligned}\tag{18}$$

If we use the *disagreement* diversity (Equation (9)), the element of $\mathbf{D}_{\mathbb{k}_\alpha}$ will equal to

$$\mathbf{D}_{\text{dis}, \mathbb{k}_\alpha} = f_{D_{\text{dis}}}(\mathbb{k}_\alpha(\mathbf{O}^T \mathbf{O}), \mathbb{k}_\alpha(\mathbf{1}^T \mathbf{O})) = \frac{1}{2}(\mathbf{1}_{L \times L} - \mathbb{k}_\alpha(\mathbf{O}^T \mathbf{O})).\tag{19}$$

Replacing Equation (19) in Equation (18), the optimization function is converted to

$$min_{\mathbf{w}, \alpha} -\mathbb{k}_\alpha(\mathbf{1}^T \mathbf{O}) \mathbf{w} - \lambda \mathbf{w}^T \frac{1}{2}(\mathbf{1}_{L \times L} - \mathbb{k}_\alpha(\mathbf{O}^T \mathbf{O})) \mathbf{w}.\tag{20}$$

As a result, given one generic kernel $\mathbb{k}$ (linear, Gaussian or polynomial) and the kernel weights α , the objective optimization (Equation (11)) in our **L2DWK** approach can be conveniently converted to a convex quadratic programming problem (Equation (20)).

4. In fact, according to the kernel definitions (e.g., linear, Gaussian and polynomial kernels), we can't represent a kernel function with only one variable, e.g., $\mathbb{k}(\mathbf{1}_{N \times 1}^T \mathbf{O})$ and $\mathbb{k}(\mathbf{O}^T \mathbf{O})$ in Equation (12). Here, for simpler and lighter descriptions of the related formulations in our method, kernelizing of the product of two variables $\mathbf{x}$ and $\mathbf{y}$, i.e., $\mathbb{k}(\mathbf{x}, \mathbf{y})$, is specifically marked as $\mathbb{k}(\mathbf{x}^T \mathbf{y})$.

Algorithm I: Iterative training algorithm for **Learning TO Diversify via Weighted Kernels (L2DWK)**
Input:

Val: the validation set, $|\text{Val}| = N$.
 $H = \{h_1, h_2, \dots, h_L\}$: the base classifier set, $|H| = L$.
 $\mathbf{D}$: pairwise diversity.
 $\mathbb{k}$: the kernel function.

Output:

$\mathbf{w}$: the classifier weights.

Parameters:

T : the max iteration.
 α^t : a $1 \times N$ vector (the kernel weights), and α_i^t is the kernel weight of sample x_i at the t^{th} iteration.
 α^{t*} : a $1 \times N$ vector (the new kernel weights for updating) at the t^{th} iteration.
 ϵ_t : the ensemble error rate at the t^{th} iteration.
 β_t : a parameter that $\beta_t \in [0, 1]$, and $\beta_t \leq \beta_{t+1}$.

Procedure:

- 1: Set $\alpha_i^1 = 1/N$.
- 2: **For** $t = 1, 2, \dots, T$
- 3: Learn classifier weights $\mathbf{w}$ (Equation (18)) with weighted kernels ($\mathbb{k}$ and α^t).
- 4: Calculate the ensemble classification error ϵ_t by $\mathbf{w}$ and Val.
- 5: Compute new kernel weights α^{t*} with ϵ_t (Equation (23) or (??)).
- 6: Update kernel weights α^{t+1} with α^t and α^{t*} (Equation (22)).
- 7: **End**

3.2 Iterative Training Algorithm

In general, it is difficult to find the solution for the optimization in Equation (18) without knowing both $\mathbf{w}$ (classifier weights) and $(\alpha/\mathbb{k})$ (kernel weights). That is to say, the generic joint optimization of $\mathbf{w}$ and $(\alpha/\mathbb{k})$ in Equation (18) is non-convex. However, with known $\mathbb{k}$ and α , the optimization is simplified to a convex quadratic programming problem for learning classifier weights. Consequently, we propose an iterative training algorithm for **L2DWK**.⁵

There are three important steps in this learning algorithm: initialization, classifier weight calculation, and kernel weight updating, which are sequentially presented in the following subsections in detail. Note that in our iterative training algorithm, the final learned results will be selected as the kernel weights and the classifier weights at the t^{th} iteration where the learned classifier ensemble has the minimal classification error on the validation set. It may be more rational to use a new validation set in this training algorithm if there are additional data samples available.

3.2.1 INITIALIZATION

In the iterative training algorithm, firstly given the validation set (Val) and base classifiers H for optimization, assign the pairwise diversity $\mathbf{D}$. Then, set the max epoch T as a stop constraint, and $\alpha_i = 1/N$ for each sample x_i , where N is the number of samples in Val. In our method, we can use general kernels, e.g., linear, polynomial, and Gaussian kernels.

5. To some extent, our algorithm (shown in Algorithm I) can be seen as a variation on similar iterative training algorithms and follows several typical steps in known ones (e.g. in (Wolpert, 1992; Yang and Yin, 2019)). In each iteration, given the known kernel and the known kernel weights, the optimization objective is expressed as a convex formulation.

3.2.2 CLASSIFIER WEIGHT CALCULATION

In each iteration, given a kernel $\mathbb{k}$ with known weights α , we first use Equation (18) and (13) to calculate the base classifier weights vector $\mathbf{w}$. As described above, this optimization can be solved as a typical convex quadratic programming problem. Then, samples of the validation set Val are classified with $\mathbf{w}$, and the ensemble classification error rate ϵ_t is correspondingly calculated.

3.2.3 KERNEL WEIGHT UPDATING

Updating the kernel weights α^t is a key process in **L2DWK**. In the iterative training algorithm, we assume the kernel weights α^{t+1} have a relationship with α^t , and use a dynamically damped trick, i.e., the damped factor $\beta_t \in [0, 1]$ and $\beta_t \leq \beta_{t+1}$. We empirically set β_t at the t^{th} iteration as

$$\beta_t = \frac{1}{t}. \quad (21)$$

Then, we use the following equation to update kernel weights for the $(t+1)^{\text{th}}$ iteration,

$$\alpha^{t+1} = \beta_t \alpha^{t*} + (1 - \beta_t) \alpha^t, \quad (22)$$

where α^t and α^{t*} are the original and new kernel weights at the t^{th} iteration, respectively.

Here, for increasing diversity while keeping fair classification accuracy, the new weight vector α^{t*} for kernel weight updating should increase the weights of corrected-classified samples. Our proposed technique, called as **L2DWK-Hinge**, gets the idea from the hinge loss (Rosasco et al., 2004). The hinge loss is a loss function used to train classifiers in conventional machine learning techniques. It is usually used for “maximum-margin” classification, e.g., Support Vector Machines. **L2DWK-Hinge** computes α^{t*} as

$$\alpha_i^{t*} = \begin{cases} \frac{1}{N_{\epsilon_t}} & m_i \leq 0 \\ 0 & \text{otherwise} \end{cases}, \quad (23)$$

where m_i is the margin of sample x_i (Equation (5)), N_{ϵ_t} is the number of samples which are correctly classified by the ensemble at iteration t .

In addition, for both above methods, as $\sum_{i=1}^N \alpha_i^1 = 1$, the sum of vector α^{t+1} is

$$\begin{aligned} \sum_{i=1}^N \alpha^{t+1} &= \sum_{i=1}^N (\beta_t \alpha_i^{t*} + (1 - \beta_t) \alpha_i^t) \\ &= \beta_t \sum_{i=1}^N (\alpha_i^{t*}) + (1 - \beta_t) \sum_{i=1}^N (\alpha_i^t) \\ &= \beta_t + (1 - \beta_t) \sum_{i=1}^N (\alpha_i^t) \\ &= \beta_t + (1 - \beta_t) \beta_{t-1} + \dots + \prod_{p=1}^t (1 - \beta_p) \sum_{i=1}^N (\alpha_i^1) \\ &= \beta_t + (1 - \beta_t) \beta_{t-1} + \dots + \prod_{p=1}^t (1 - \beta_p) \\ &= 1. \end{aligned} \quad (24)$$

Obviously, there are $\sum_{i=1}^N \alpha_i^t = 1$ ($\forall t$) and $\alpha_i^t \in [0, 1]$ ($\forall t, i$) in the training procedure of this iterative algorithm.

4. Results

4.1 Experimental Setting and Datasets

In our L2DWK method, base classifiers are generated from the generic baseline ensemble, Bagging. Please note that without specification, in the following experiments the Classifi-

cation and Regression Tree (CART) classifier is used as the base learner in all ensembles, where CART is derived from the standard Matlab CART function with the default setting. The *disagreement* (Equation (9)) is chosen to measure the ensemble diversity. As most of the experimental UCI datasets are with rather small-scale, the linear kernel is used in our experiments. Moreover, 10-fold cross validation is conducted on each dataset, and the validation set is bootstrapped from the training set ⁶. A plenty number of base classifiers (here, 101 as generally used in Bagging systems) are trained in Bagging for all experimental ensembles.

Table 1: Information for 30 typical UCI datasets.

Data Set	Instances	Attributes	Classes
allbp	3,772	29	3
artificial	5,109	7	10
audiology	226	69	24
auto-mpg	399	7	4
autos	205	25	7
breast-w	699	9	2
clean1	476	166	2
colic	368	22	2
german	1,000	24	2
glass	214	9	7
heart-c	303	13	5
heart-h	294	13	5
imageseg	2,310	19	7
iris	153	4	3
labor	57	6	2
led24	3,200	24	10
led7	3,200	7	10
lymph	148	18	4
machine	209	7	8
page-blocks	5,473	10	5
sickeuthyroid	3,263	24	2
sonar	208	60	2
splice	3,190	61	3
tic-tac-toe	958	9	2
vehicle	848	18	4
vote	435	16	2
vowel	990	13	11
wave21	5,000	21	3
wave40	5,000	40	3
zoo	104	17	7

To verify the effect of the proposed method compared to a number of other ensembles, 30 typical datasets from UCI machine learning repository (Frank and Asuncion, 2010) are used in our experiments. Most of these datasets are rather challenging with a low accuracy for general classification methods (e.g., CART). More information for the datasets is presented in Table 1.

6. In our experiments, this validation set is bootstrapped from the initial training set, and is used as the new training set for learning kernel weights and classifier weights in the iterative training algorithm for **L2DWK**. Here, because of the rather small size of the original training set, 50% samples are bootstrapped (with repeated sampling) for each dataset.

4.2 Experimental Results

Our L2DWK approach is extensively compared with several representative ensemble methods, i.e., Bagging (Bag) (Breiman, 1996), Gasen (Zhou et al., 2002), Forward Ordering (FO) (Martinez-Munoz et al., 2009), Backward Ordering (BO) (Martinez-Munoz et al., 2009), Genetic Algorithm with diversity (GA-div) (Trawinski et al., 2009), and Ensemble Pruning with Semi-Definite Programming (SDP) (Zhang et al., 2006). The Ordering algorithms (FO, BO and GA-div) select classifiers directly using the diversity measure on the validation set, and we call them as pure-diversity ensembles. In the Ordering algorithms and SDP, about 20% classifiers are selected according to the suggestion in (Zhang et al., 2006; Martinez-Munoz et al., 2009). In the GA-based ensembles (Gasen and GA-div), the selecting threshold for GA is set to the default value $1/L$, where L is the number of component classifiers. In L2DWK, related parameters are empirically set on the validation set, i.e., $\lambda = 1$ for L2DWK in Equation (18).

We also present two important statistical comparison measures (Demsar, 2006) for comparing different ensembles over multiple datasets. Firstly, we assign ranks (one typical measure in the Friedman test) to evaluate the methods’ performance on each dataset. Mark the best method rank 1, and the worse the larger. Then, we calculate the average rank for each method. Secondly, we use the Nemenyi test to evaluate and compare different methods.

The experimental results with base classifiers (CART) from Bagging are shown in Table 2. We also present results from the Nemenyi test measure approach (Demsar, 2006) in Figure 1 ($CD = 1.92$ for 9 ensemble methods on all 30 datasets), where methods whose average ranks are not significantly different from each other, according to the Nemenyi test ($p\text{-value} \leq 0.05$), are connected with a horizontal line.

Figure 1: Comparison of the different ensembles with Bagging base classifiers using the Nemenyi test, where ensembles not significantly different ($p\text{-value} \leq 0.05$) are connected in the diagram.

As shown in Table 2, we can find that the results of pure-diversity learning methods (Ranks 4.20 – 6.70) and our L2DWK method (2.27) show the importance of diversity with weighted kernels (also see details in Section 5). Our L2DWK method has the smallest rank and the larger average recognition rate, and outperforms all other representative ensembles. More detailed observations can be found as follows.

Among all 8 ensembles, our L2DWK method have the excellent rank (2.27), followed by SDP (2.67). Pure-diversity learning methods (PO (5.97), BO (6.70) and GA-div (4.20)) perform much worse ranks among all experimental ensembles. The pure-diversity learning has some limitations because of the weak relation between the direct diversity on samples

Table 2: Comparison of the recognition rate (%) (Average $\pm$ Standard Deviation) with Bagging base classifiers.

Dataset	Bag	Gasen	GA-div	FO	BO	QDP	SDP	L2DWK
allbp	97.59 ± 0.88	97.64 ± 0.95	97.61 ± 0.83	97.61 ± 0.72	97.67 ± 0.78	97.80 ± 0.86	97.75 ± 0.72	97.72 ± 0.80
artificial	61.03 ± 1.96	61.01 ± 1.55	61.01 ± 1.78	60.32 ± 1.92	60.83 ± 1.86	61.46 ± 1.68	61.05 ± 1.28	61.38 ± 1.75
audiology	59.02 ± 20.04	69.45 ± 20.50	65.01 ± 18.91	70.76 ± 19.13	49.34 ± 21.46	74.14 ± 20.65	68.15 ± 19.64	75.54 ± 21.46
auto-mpg	86.68 ± 5.14	85.93 ± 5.15	87.18 ± 4.92	85.42 ± 4.40	85.17 ± 3.64	84.66 ± 4.22	86.43 ± 4.53	86.43 ± 4.43
autos	82.74 ± 10.67	84.17 ± 9.66	82.74 ± 10.67	76.73 ± 12.31	78.45 ± 12.38	83.07 ± 9.73	80.83 ± 10.32	83.54 ± 12.26
breast-w	95.71 ± 2.23	96.13 ± 2.35	95.85 ± 2.38	96.14 ± 1.79	95.28 ± 2.24	96.57 ± 2.45	96.42 ± 1.81	96.71 ± 2.14
clean1	86.74 ± 2.54	87.14 ± 3.55	86.33 ± 3.22	85.08 ± 3.34	86.93 ± 3.97	87.54 ± 3.96	88.01 ± 1.88	87.97 ± 3.67
colic	85.54 ± 5.71	85.01 ± 5.17	85.54 ± 5.71	84.46 ± 6.22	85.00 ± 6.00	85.00 ± 6.39	86.08 ± 5.49	85.82 ± 5.86
german	76.40 ± 3.17	76.50 ± 2.92	76.10 ± 3.18	75.30 ± 4.08	74.90 ± 3.81	74.80 ± 3.82	77.10 ± 2.91	75.30 ± 3.23
glass	77.20 ± 6.61	75.85 ± 6.33	77.20 ± 6.61	76.32 ± 7.60	76.32 ± 7.27	75.37 ± 7.22	79.10 ± 5.88	79.03 ± 5.13
heart-c	79.52 ± 6.49	79.18 ± 5.52	79.52 ± 6.49	78.15 ± 9.03	78.82 ± 7.52	78.55 ± 7.22	80.18 ± 5.34	78.85 ± 6.77
heart-h	79.64 ± 7.88	78.61 ± 8.57	79.30 ± 7.28	79.60 ± 5.26	78.61 ± 8.26	79.26 ± 7.27	79.60 ± 7.32	79.30 ± 7.80
imageseg	97.45 ± 1.41	97.58 ± 1.43	97.53 ± 1.44	97.27 ± 1.44	97.36 ± 1.30	97.66 ± 1.42	97.53 ± 1.33	97.75 ± 1.44
iris	94.67 ± 7.57	94.00 ± 7.34	94.67 ± 7.57	96.00 ± 5.62	94.67 ± 6.13	95.33 ± 7.06	96.67 ± 4.71	95.33 ± 6.32
labor	91.67 ± 11.79	95.00 ± 8.05	91.67 ± 11.79	95.00 ± 8.05	88.33 ± 13.72	91.67 ± 8.78	95.00 ± 8.05	91.67 ± 8.78
led24	72.22 ± 0.80	71.88 ± 1.43	72.34 ± 0.85	71.56 ± 1.00	72.03 ± 1.01	72.06 ± 1.23	72.44 ± 0.96	72.31 ± 0.89
led7	73.41 ± 2.54	73.38 ± 2.49	73.38 ± 2.49	73.25 ± 2.83	73.34 ± 2.49	73.06 ± 2.97	73.53 ± 2.36	73.41 ± 2.52
lymph	81.13 ± 10.23	83.23 ± 7.72	81.13 ± 10.23	81.69 ± 8.48	67.08 ± 22.11	83.79 ± 7.77	83.79 ± 8.96	81.79 ± 7.62
machine	84.29 ± 10.78	85.69 ± 8.37	83.79 ± 10.06	86.64 ± 6.23	82.38 ± 11.01	89.05 ± 7.79	87.14 ± 8.46	90.95 ± 6.53
page-blocks	97.06 ± 0.62	97.11 ± 0.61	97.06 ± 0.62	97.04 ± 0.56	97.06 ± 0.63	96.97 ± 0.63	97.19 ± 0.58	97.19 ± 0.68
sickeuthyroid	97.66 ± 0.97	97.76 ± 0.81	97.66 ± 0.97	97.66 ± 0.90	97.66 ± 0.97	97.79 ± 0.83	97.72 ± 1.08	97.82 ± 0.57
sonar	83.03 ± 8.40	80.55 ± 10.93	83.46 ± 9.55	83.18 ± 4.57	80.18 ± 6.67	80.60 ± 6.53	82.08 ± 7.78	82.61 ± 7.47
splice	94.80 ± 1.12	94.76 ± 1.31	94.70 ± 1.14	94.64 ± 1.66	94.55 ± 1.24	95.02 ± 1.44	94.86 ± 1.14	94.98 ± 1.20
tic-tac-toe	93.21 ± 2.53	93.11 ± 2.65	93.21 ± 2.76	91.33 ± 2.70	92.18 ± 3.29	95.51 ± 2.20	93.21 ± 2.79	96.76 ± 2.05
vehicle	74.34 ± 2.98	74.70 ± 3.73	74.23 ± 3.35	73.74 ± 4.13	73.54 ± 2.96	74.80 ± 4.27	74.69 ± 2.98	76.09 ± 4.02
vote	95.91 ± 2.99	96.14 ± 2.84	95.91 ± 2.99	95.91 ± 3.35	94.55 ± 4.31	95.00 ± 4.12	95.68 ± 2.84	96.59 ± 2.68
vowel	89.70 ± 4.92	89.70 ± 4.66	90.20 ± 4.67	88.08 ± 4.31	87.07 ± 4.85	90.20 ± 4.13	89.80 ± 4.19	91.01 ± 4.82
wave21	84.12 ± 1.43	83.74 ± 1.41	84.08 ± 1.18	83.50 ± 1.37	83.26 ± 1.42	84.06 ± 1.10	84.20 ± 1.29	84.22 ± 1.34
wave40	84.00 ± 1.52	83.76 ± 1.44	83.86 ± 1.51	82.28 ± 1.54	82.74 ± 1.47	83.38 ± 1.63	83.66 ± 1.18	83.68 ± 1.45
zoo	71.18 ± 12.14	73.18 ± 10.74	71.18 ± 12.14	70.18 ± 12.65	71.18 ± 12.14	73.18 ± 10.74	73.18 ± 10.74	73.18 ± 10.74
win-tie-loss (L2DWK)	22/2/6	23/1/6	22/2/6	25/1/4	30/0/0	23/3/4	15/3/12	0/30/0
Rank	4.20	4.33	4.20	5.97	6.70	4.10	2.67	2.27

and the ensemble accuracy (see Section 5). In contrast, our L2DWK method has more impressive performance considering the adaptive diversity with data via weighted kernels. For most experimental datasets, there are a plenty of samples or/and a number of features (attributes) per each set, which can be used to train fairly diverse base classifiers. In such cases, our L2DWK methods can adaptively select and combine diverse classifiers by learning suitable kernel weights and classifier weights.

Specifically, L2DWK ranks 2.27. It is significantly better, in terms of the average rank, than Bag, PO, BO, GA-div and Gasen. It achieves the top results on most datasets, e.g., “audiology”, “breast-w”, “imageseg”, “machine”, “page-blocks”, “sickeuthyroid”, “tic-tac-toe”, “vechicle”, “vote”, “vowel”, “wave21”, and “zoo”.

Note that GA-based methods (GA-div and Gasen) rank 4.20–4.33, which are just a little better than Bagging (4.20). Generally speaking, by utilizing a better fitness function or choosing more appropriate parameter values, the performance of GA-based ensembles might be improved.

4.3 Failed Case Analysis

As shown in Table 2, there are a small part of datasets on which L2DWK’s performance is slightly worse than SDP. Most of these datasets always have very few samples and attributes, e.g., “iris” with 153 samples and 4 attributes, “labor” with 57 samples and 6 attributes, and “lymph” with 148 samples and 18 attributes (see Table 1). It is essentially difficult to train diverse base classifiers on such datasets where the existed data itself can’t reflect some diversity, and it is also challenging to learn and select diverse classifiers on the validation set with few samples.

5. Discussion

To investigate the relation between accuracy and diversity, we present empirical results of experimental ensembles on all 30 datasets. We compare Bag, Gasen, GA-div, FO, BO, QDP, SDP, and L2DWK ensembles on the testing set of each dataset in our experiments (Section 4.2). Note that the diversity of our L2DWK method is measured with kernel weights learned by Algorithm I (the classifiers’ outputs are kernelized and weighted), while for all other ensembles, the diversity is measured with the uniform distribution (an averaging probability on the samples). The results between accuracy and diversity in ensembles for 5 typical datasets are shown in Figure 2, where the first 3 and the last 2 sets represent common cases and cases with few samples and/or attributes, respectively.

Here, instead of the whole dataset, the diversity is respectively calculated on two different sets divided from the testing set, i.e., the one containing all samples correctly classified by Bagging (easy samples) and the other for those wrongly classified by Bagging (hard samples). Because easy samples are easily recognized by component classifiers, there will always be no obvious relation between the accuracy of the ensemble and the diversity on these easy samples (shown in Figure 2(a)). Generally speaking, it needs more diverse classifiers to effectively recognize the hard samples in the ensemble system.

As shown in Figure 2, obviously conventional diversity-based ensembles (e.g., GA-div, FO and BO) and other representative ensembles (e.g., Bag, Gasen and SDP) don’t have a clear relation between ensemble accuracy and its diversity, which has already been announced by several researchers in the literature (Ruta and Gabrys, 2001; Kuncheva and Whitaker, 2003; Saitta, 2006). On the contrary, for our L2DWK method, the ensemble diversity (on hard samples) is strongly correlated to the accuracy. On most datasets (23 of 30), L2DWK archives both the largest diversity (on hard samples) and a higher accuracy. That is to say, our L2DWK method seeks adaptive diversity with data (via weighted ker-

(a) Diversity measured on correctly classified samples by Bagging (easy samples), where there is not a clear relation between accuracy and diversity (on easy samples) for all ensembles.

(b) Diversity measured on wrongly classified samples by Bagging (hard samples), where there is a strong relation between ensemble accuracy and (adaptive) diversity (on hard samples) in L2DWK on most datasets (the first row). The last two datasets (the second row) are failed cases because of few samples and/or attributes for each dataset.

Figure 2: Diversity-accuracy maps of ensembles on several typical datasets.

nels) in the view of data distributions in different situations, and its diversity is strongly correlated to the accuracy in most cases.

There are also a few cases (7 of 30) datasets (e.g., “labor” and “lymph” of the last two examples in Figure 2), where L2DWK doesn’t have a clear relation between accuracy and diversity. These datasets have very few samples and/or attributes, e.g., “labor” with only 57 samples and 6 attributes, and “lymph” with 148 samples and 18 attributes. It is essentially difficult to train and select diverse base classifiers on such datasets.

6. Conclusions

Classifier ensemble is widely considered as an effective method to improve accuracy of base classifiers, which has a variety of applications in pattern recognition, machine learning and data mining. Generally speaking, ensemble of diverse classifiers should allow us to get higher accuracy. However, there is not a definitive connection between ensemble diversity and accuracy. In this paper, we introduce learning to diversify in classifier ensemble, and construct a clear relation between diversity and accuracy. Specifically, within a linear classifier ensemble framework, we propose Learning TO Diversify via Weighted Kernels (**L2DWK**) which learns classifier weights by maximizing the accuracy and the (adaptive) diversity simultaneously, where the diversity is measured with a set of weights on the samples’ outputs within a kernel. Moreover, we propose an iterative training algorithm to adaptively learn classifier weights and kernel weights.

Our proposed approach has a very impressive performance on 30 typical UCI datasets most of which are with small-scale samples. Consequently, one future issue is to examine suitable situations for the Linear, Gaussian and Polynomial kernels, and improve L2DWK for complicated real systems where large-scale datasets are always imbalanced or affected by different challenging sources.

References

- L. Breiman. Bagging predictors. *Machine Learning*, 24(1):123–140, 1996.
- L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.
- Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- Everlandio R. Q. Fernandes, André C. P. L. F. de Carvalho, and Xin Yao. Ensemble of classifiers based on multiobjective genetic sampling for imbalanced data. *IEEE Trans. Knowledge and Data Engineering*, 32(6):1104–1115, 2020.
- A. Frank and A. Asuncion. *UCI machine learning repository*. University of California, Irvine, School of Information and Computer Sciences, 2010. URL <http://archive.ics.uci.edu/ml>.
- Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156, 1996.

- Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.
- Ariel Jaffe, Ethan Fetaya, Boaz Nadler, Tingting Jiang, and Yuval Kluger. Unsupervised ensemble learning with dependent classifiers. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 351–360. JMLR.org, 2016.
- Govinda B. KC, Giovanni Bocci, Srijan Verma, Md Mahmudulla Hassan, Jayme Holmes, Jeremy J. Yang, Suman Sirimulla, and Tudor I. Oprea. A machine learning platform to estimate anti-SARS-CoV-2 activities. *Nature Machine Intelligence*, 3:527–535, 2021.
- A. Krogh and P. Sollich. Statistical mechanics of ensemble learning. *Physical Review E*, 55(1):811–825, 1997.
- A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems 7*, pages 231–238, 1995.
- L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles. *Machine Learning*, 51(2):181–207, 2003.
- G. Martinez-Munoz, D. Hernandez-Lobato, and A. Suarez. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(2):245–259, 2009.
- Gonzalo Martínez-Muñoz and Alberto Suárez. Pruning in ordered bagging ensembles. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pages 609–616. ACM, 2006.
- David W. Opitz and Jude W. Shavlik. Generating accurate and diverse members of a neural network ensemble. In *Advances in Neural Information Processing Systems (NIPS’96)*, pages 535–541. MIT Press, 1996.
- Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 784–791. ACM, 2008.
- Filip Radlinski, Paul N. Bennett, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance. *SIGIR Forum*, 43(2):46–52, 2009.
- L. Rosasco, D. V. Caponnetto, M. Piana, and A. Verri. Are loss functions all the same? *Neural Computation*, 16:1063–1076, 2004.

- Dymitr Ruta and Bogdan Gabrys. Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting. In *Multiple Classifier Systems, Second International Workshop, MCS 2001 Cambridge, UK, July 2-4, 2001, Proceedings*, pages 399–408, 2001.
- Lorenza Saitta. Hypothesis diversity in ensemble classification. In *Foundations of Intelligent Systems, 16th International Symposium, ISMIS 2006, Bari, Italy, September 27-29, 2006, Proceedings*, pages 662–670, 2006.
- Catherine A. Shipp and Ludmila Kuncheva. Relationships between combination methods and measures of diversity in combining classifiers. *Information Fusion*, 3(2):135–148, 2002.
- D. B. Skalak. The sources of increased accuracy for two proposed boosting algorithms. In *Proceeding of the 11th American Association for Artificial Intelligence*, 1996.
- Tao Sun and Zhi-Hua Zhou. Structural diversity for decision tree ensemble learning. *Frontiers of Computer Sciences*, 12(3):560–570, 2018. doi: 10.1007/s11704-018-7151-8. URL <https://doi.org/10.1007/s11704-018-7151-8>.
- E. K. Tang, P. N. Suganthan, and X. Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, 2006.
- K. Trawinski, A. Quirin, and O. Cordon. On the combination of accuracy and diversity measures for genetic selection of bagging fuzzy rule-based multiclassification systems. In *Proceedings of the 9th Intelligent Systems Design and Applications*, pages 121–127, 2009.
- D.H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–260, 1992.
- Chun Yang and Xu-Cheng Yin. Diversity-based random forests with sample weight learning. *Cognitive Computation*, 11(5):685–696, 2019.
- Xu-Cheng Yin, Kaizhu Huang, Chun Yang, and Hong-Wei Hao. Convex ensemble learning with sparsity and diversity. *Information Fusion*, 20:49–59, 2014.
- Y. Zhang, A. Burer, W. N. Street, K. Bennett, and E. Parrado-hern. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research*, 7:1315–1338, 2006.
- Z. H. Zhou, J. Wu, and W. Tang. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, 137:239–263, 2002.
- Zhi-Hua Zhou. *Ensemble Methods: Foundations and Algorithms*. Chamman & Hall/CRC, Boca Raton, FL, 2012.
- Zhi-Hua Zhou and Ji Feng. Deep forest. *National Science Review*, 6(1):74–86, 2019.

Author Contributions

X.-C. Yin conceived the study. C. Yang and X.-C. Yin developed the methodology, and designed the experiments. C. Yang implemented the experiments. All authors discussed the results, wrote the manuscript, and approved the manuscript.

Competing Interests

The authors declare no competing interests.

Additional Information

The experimental datasets and the (Matlab) source codes of our L2DWK method are available at <https://github.com/onionych/L2DWK/>. Related work and supplementary experiments are also provided in the Supplementary Materials.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [LearningtoDiversifysupp20210810.pdf](#)