MAP-Elites with Cosine-Similarity for Evolutionary Ensemble Learning

Hengzhe Zhang[1], Qi Chen[1], Alberto Tonda[2], Bing Xue[1], and Wolfgang Banzhaf[3] and Mengjie Zhang[1]

[1] Victoria University of Wellington, Wellington, New Zealand
{hengzhe.zhang, qi.chen, bing.xue, mengjie.zhang}@ecs.vuw.ac.nz
[2] UMR 518 MIA-Paris, INRAE, Paris, France
alberto.tonda@inrae.fr
[3] Michigan State University, East Lansing, MI, USA
banzhafw@msu.edu

Abstract. Evolutionary ensemble learning methods with Genetic Programming have achieved remarkable results on regression and classification tasks by employing quality-diversity optimization techniques like MAP-Elites and Neuro-MAP-Elites. The MAP-Elites algorithm uses dimensionality reduction methods, such as variational auto-encoders, to reduce the high-dimensional semantic space of genetic programming to a two-dimensional behavioral space. Then, it constructs a grid of high-quality and diverse models to form an ensemble model. In MAP-Elites, however, variational auto-encoders rely on Euclidean space topology, which is not effective at preserving high-quality individuals. To solve this problem, this paper proposes a principal component analysis method based on a cosine-kernel for dimensionality reduction. In order to deal with unbalanced distributions of good individuals, we propose a zero-cost reference points synthesizing method. Experimental results on 108 datasets show that combining principal component analysis using a cosine kernel with reference points significantly improves the performance of the MAP-Elites evolutionary ensemble learning algorithm.

Keywords: Evolutionary ensemble learning · Quality diversity optimization · Multi-dimensional Archive of Phenotypic Elites.

1 Introduction

Ensemble learning methods have gained popularity in recent years due to their ability to reduce the variance of unstable machine learning algorithms without increasing bias. Typically, the generalisation loss of an ensemble model $\mathbb{E}_F$ for a given dataset $\{X, Y\}$ can be decomposed into two terms, as shown in Eq. (1):

$$\mathbb{E}_F = \underbrace{\mathbb{E}_{f \in F}\left[(f(X) - Y)^2\right]}_{\text{average loss}} - \underbrace{\mathbb{E}_{f \in F}\left[(f(X) - \mathbb{E}_{f' \in F}\left[f'(X)\right])^2\right]}_{\text{ambiguity}} \tag{1}$$

On the right side of this equation, the first term represents the average loss between the prediction of each model $f(X)$ and the target Y, and the second

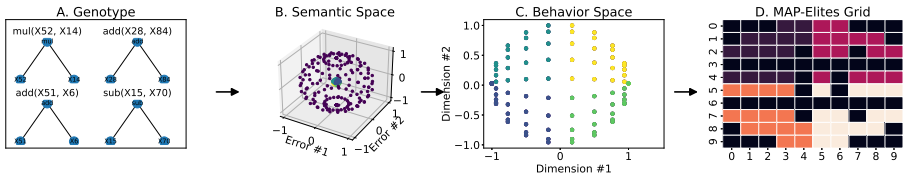

Fig. 1. The workflow of MAP-Elites

term ambiguity represents the difference between the prediction of each model $f(X)$ and the average prediction among models in the ensemble $\mathbb{E}_{f'}[f'(X)]$. For an evolutionary ensemble learning method, having two groups of base learners with the same average fitness values often means they have the same average loss. However, they may have different predictive accuracy, due to the difference in their ambiguity. Typically, a more diverse group of base learners has larger ambiguity and thus provides a more accurate prediction. In theory, we can optimize E_f with an evolutionary algorithm. However, evaluating the fitness value of an ensemble model may be computationally expensive in practice, and a more practical way is to implicitly optimize E_f by maintaining a set of high-quality and diverse individuals during the evolutionary process.

In this work, we focus on using genetic programming (GP) [2] to evolve a set of high-quality and diverse regressors for ensemble learning. GP has been widely used in regression tasks due to its flexible representation. However, the traditional GP framework mainly focuses on finding regressors minimizing the training error during the evolutionary process, making it ineffective at obtaining a diverse set of regressors in a single run. In order to obtain a set of complementary regressors, one idea is to take the semantics of regressors into consideration. The semantics of each GP individual represents the predictions for a set of samples. The target semantics is a point in the semantic space representing the target labels $\{y_1, \ldots, y_n\}$. In semantic GP for ensemble learning for regression, a desired ensemble model is a set of regressors with complementary semantics, thus the combined prediction of this kind of regressor can be approximately equal to the target semantics. To generate a desired ensemble model, it is important to develop novel selection operators that highlight both quality and diversity.

In the field of evolutionary computation (EC), there are a variety of techniques for finding diverse individuals with high quality. The Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [22] is a representative example. As shown in Fig. 1, MAP-Elites defines a behavioral space for a given problem that describes the desired property of high-quality solutions. In this example, a cosine-kernel-based principal component analysis (KPCA) method that only considers the angle distance between individuals is used to define the behavioral space. The general concept behind MAP-Elites is to divide the behavioral space into multiple cells and retain the best individual in each cell to maintain population diversity. Based on this idea, the MAP-Elites algorithm has been used to evolve an ensemble of classifiers [24], where the MAP-Elites algorithm employs a grid to record a diverse

set of well-performing classifiers with different semantics from an ensemble model in a single run.

Despite the many benefits MAP-Elites can bring, it is still not widely adopted in evolutionary ensemble learning due to the difficulty in defining the behavioral space. Initially, defining behavioral descriptors requires domain knowledge, such as a handcrafted descriptor named the entropy of instructions in linear genetic programming (LGP) [11], which is a GP variant with a sequence of instructions to represent GP programs. Recent research demonstrates that behavioral descriptors for each GP individual can be automatically obtained based on its semantic vector [25]. For a regression problem with n training samples, its semantic vector is n-dimensional. When n is large, the curse of dimensionality causes exponential growth of the number of cells in a MAP-Elites grid. Recently, auto-encoders (AE) have been used to automatically discover behavioral descriptors on robot control tasks [7] and classification tasks (Neuro-MAP-Elites) [25]. AE is a deep-learning-based dimensionality reduction method that uses a bottleneck architecture to compress high-dimensional data into low-dimensional representations. For evolutionary machine learning tasks, the optimal behavioral space should be able to describe the distribution of high-quality individuals. This means that AE for generating the behavioral space should be trained on high-quality individuals. To achieve this goal, Neuro-MAP-Elites trains a variational auto-encoder (VAE) [14] on good individuals from the final population of a GP run. Then, the pre-trained VAE can define a good MAP-Elites grid for evolving diverse and high-performing individuals in another GP run.

There are two potential limitations with Neuro-MAP-Elites. First, the behavioral descriptor generated by VAE may not be effective to find complementary learners. Considering a case where letting the semantic vectors of three individuals A, B, C be $A = \{y_1 - 100, \ldots, y_n - 100\}, B = \{y_1 - 500, \ldots, y_n - 500\}, C = \{y_1 + 100, \ldots, y_n + 100\}$. If selecting two individuals with the largest Euclidean distance to form an ensemble model, they will be $\{B, C\}$. However, the optimal set is $\{A, C\}$ because the average prediction results of these two individuals match the semantic target $\{y_1 \ldots y_n\}$. Unfortunately, Euclidean-space-based VAE may prefer $\{B, C\}$ and thus does not perform well for evolutionary ensemble learning. The second issue is that training a VAE on good individuals obtained from the final population of a GP run is inefficient and may misguide the evolutionary process. Compared with using good individuals in a single GP run, it is more efficient to use the target semantics in supervised learning tasks to generate reference points to train a dimensionality reduction model. Moreover, due to the mismatched distributions of the initial and the final populations in GP, a VAE trained on well-performing individuals in the previous GP run might not be helpful to the initial population.

In this paper, we propose a new ensemble learning method based on MAP-Elites and GP, named MEGP, with the following objectives:

– Considering that it is difficult for Euclidean-space-based VAE to find complementary individuals, we propose using cosine-kernel-based PCA for dimensionality reduction in MAP-Elites to better find complementary base learners.

KPCA with cosine-kernel focuses on the relative angle to the target semantics, thereby encouraging GP to find diverse and complementary regressors to create an ensemble model.
- We propose a zero-cost method for generating reference points representing good solutions in the semantic space for training a dimensionality reduction model. A dimensionality reduction model trained on reference points can be viewed as a good behavioral descriptor and can be used in MAP-Elites.

2 Related Work

2.1 Semantic GP

In recent years, semantic GP has attracted considerable attention. The key idea of semantic GP is to use semantic information in genetic operators or selection operators to generate offspring with high behavioral correlation with their parents. In terms of genetic operators, a considerable number of semantic-based crossover and mutation operators have been developed to fulfill the semantics for the new generation [21, 30]. As for selection operators, there are some works that consider selecting parent individuals based on semantic vectors instead of fitness values [16, 8], which improves population diversity and thus results in better performance.

2.2 GP-Based Ensemble Learning

The idea of using multiple GP models to form an ensemble model can be traced back to BagGP [15], where multiple runs of GP are performed within the bagging framework. However, it is possible to maintain a diverse set of models in a single GP run since it is a population-based method. In spatial structure with bootstrapped elitism (SS+BE) [10], the niching method [13] and a bootstrapping strategy are used to form an ensemble of GP models in a single GP run. A similar idea of using niching in GP to form an ensemble has been applied to vehicle routing problems [35]. Recently, an algorithm named 2SEGP [34] shows that purely relying on the bootstrapping strategy can also yield satisfactory results. In a GP-based feature construction scenario, it is also possible to rely on the randomness of base learners to produce an ensemble model that outperforms XGBoost [37]. When the base learner is not random enough and the bootstrapping strategy is not allowed, a diverse set of base learners can still be produced by using the quality-diversity optimization framework [25].

2.3 Quality Diversity Optimization

In recent years, quality diversity (QD) optimization has been widely used to tackle the problem of deceptive landscapes [36] and produce diverse solutions [9]. QD algorithms can be classified into grid-based and archive-based methods, based on whether they rely on a discretized behavioral space to maintain population diversity or not. Grid-based QD optimization methods discretize the behavioral space to preserve diversity, with MAP-Elites being a typical example. As shown in Fig. 1, MAP-Elites first maps an individual from a high-dimensional semantic

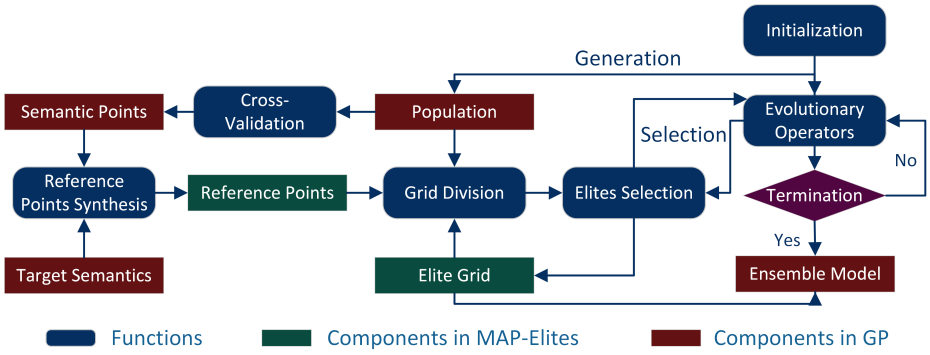

Fig. 2. All components in MEGP.

space to a low-dimensional behavioral space. Then, it divides the behavioral space into multiple grids and retains only the best individual in each cell, where all individuals in the MAP-Elites grid can be used to form an ensemble model. MAP-Elites was initially developed for robot design [22], but it has been applied to a variety of other problems, including agent control [27], airfoil optimization [12], workforce scheduling [32], and the traveling thief problem [26]. In the GP domain, MAP-Elites was initially applied to program synthesis and symbolic regression tasks [11, 4]. Subsequently, it has been extended to classification tasks for evolving an ensemble of classifiers [24].

As for archive-based methods, a typical example is Novelty Search with Local Competition (NSLC) [18]. The key idea of NSLC is to use an external archive to keep diverse individuals and use a multi-objective optimization algorithm to breed individuals based on diversity and local ranking. There are a lot of differences between MAP-Elites and NSLC. One key difference is that the MAP-Elites algorithm uses a grid to explicitly keep the structure, while NSLC implicitly keeps the structure based on a distance measure. In the evolutionary ensemble learning domain, both grid-based and archive-based QD methods have been studied [3, 25], but a comparison between them is still lacking.

3 The Proposed Ensemble Learning Algorithm

This work presents a MAP-Elites-based ensemble GP method, named MEGP. First, we introduce the algorithmic framework. Then we describe dimensionality reduction methods in MEGP and a method for generating reference semantic points that can be used in training a dimensionality reduction model.

3.1 The Overall Framework

MEGP introduces MAP-Elites into the GP-based ensemble learning scenario. The pseudocode for MEGP is presented in Algorithm 1, and all components of MEGP are shown in Fig. 2. MEGP follows the conventional framework of GP, but differs from it in the following ways:

Algorithm 1 MEGP

Input: Population Size N, Number of Generations max_gen, Dimensionality of the MAP-Elites Grid G, Training Data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$
Output: MAP-Elites Grid E

Randomly initialize a population of GP individuals $P = \{\Phi_1 \ldots \Phi_N\}$
$E \leftarrow$ MAP-Elites grid initialization with P ▷ MAP-Elites Grid
$gen \leftarrow 0$
while $gen \leq max\_gen$ **do** ▷ Main loop
    $P \leftarrow$ mutation and crossover(P)
    for $\Phi \in P$ **do** ▷ Evaluation
        $\{\hat{y_1}, \ldots, \hat{y_n}\} \leftarrow$ cross-validation($\Phi, \{\hat{x_1}, \ldots, \hat{x_n}\}$)
        $\hat{Y_\Phi} \leftarrow \{\hat{y_1}, \ldots, \hat{y_n}\}$
    $PE \leftarrow$ selecting top-50% individuals from $P \cup E$
    $\{R_1, \ldots, R_{2|PE|}\} \leftarrow$ reference point synthesis ($\{y_1, \ldots, y_n\}, \{\hat{Y_1}, \ldots, \hat{Y_{|PE|}}\}$)
    $\{Z_1, \ldots, Z_{|PE|}\} \leftarrow$ dimensionality reduction ($PE, \{R_1, \ldots, R_{2|PE|}\}$)
    $E \leftarrow$ grid division ($PE, \{Z_1, \ldots, Z_{|PE|}\}, G$)
    $E \leftarrow$ elites selection(E)
    $P \leftarrow$ random selection(E) ▷ Selection
    $gen \leftarrow gen + 1$
return E

- Multi-tree Representation: MEGP uses multiple GP trees to represent a single individual, and a linear model is used to combine these GP trees to make a prediction. The multi-tree GP is used due to its more expressive and flexible representation ability [17].
- Cross-validation Loss: MEGP uses an efficient leave-one-out cross-validation method [6] in the fitness function to evaluate each GP individual Φ based on a ridge regressor, which allows mitigating the over-fitting issue.
- Ensemble Learning: MEGP uses all individuals $e \in E$ in the final MAP-Elites grid to form an ensemble model. For an unseen data point x', the prediction result is the average of all prediction results, i.e., $\frac{\sum_{e \in E} e(x')}{|E|}$.

3.2 Angle-Based Dimensionality Reduction

The mapping of individuals from a high-dimensional semantic space to a low-dimensional behavior space is a key step in MAP-Elites. We propose to employ cosine-kernel principal component analysis (KPCA) for dimensionality reduction in MEGP. A dimensionality reduction algorithm maps the semantics $\{\hat{y}_1^i, \ldots, \hat{y}_n^i\}$ of an individual i in the semantic space to a low-dimensional point $\{z_1^i, z_2^i\}$ in the behavior space, where n is the number of data points/instances in the training dataset. PCA is a simple and efficient algorithm for dimensionality reduction tasks. However, the standard PCA algorithm focuses on capturing the variance in Euclidean space. Sometimes, individuals with bad fitness values can have large Euclidean distances from others, but they are not good candidates for an ensemble model and should not be included in the behavior space. In order to solve this problem, a cosine kernel function defined as $cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}^\top}{\|\vec{i}\| \|\vec{j}\|}$ for

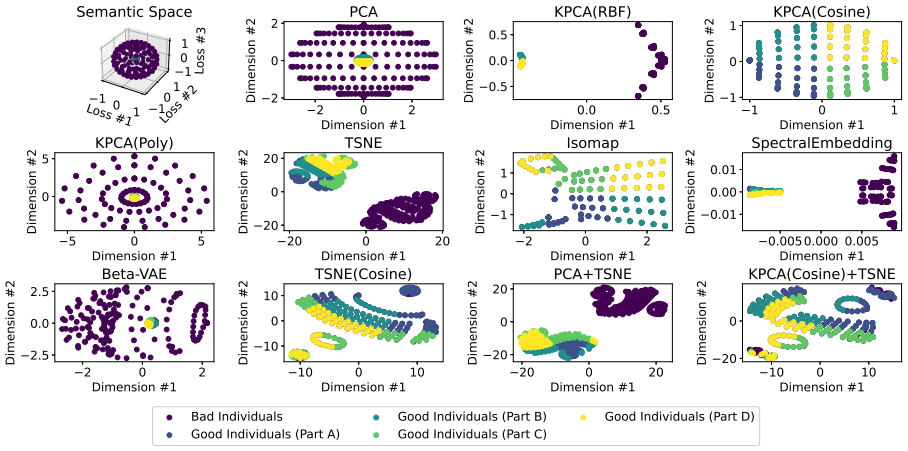

Fig. 3. An illustrative example of different dimensionality reduction techniques for inducing a behavioral space.

any two points $\vec{i}, \vec{j}$ in the semantic space is used to transform points from the n-dimensional semantic space to another n-dimensional implicit feature space. In the implicit feature space, each dimension represents the cosine similarity between a data point and the others. Next, PCA is applied to the implicit feature space to generate a two-dimensional behavior space. Because cosine similarity ignores scale, the implicit feature space only preserves the angle distance between points. Thus, individuals are only considered novel if they approach the target semantics from a different angle, i.e., from a different direction.

To illustrate why the cosine-kernel-based PCA method is suitable for MAP-Elites, Fig. 3 provides an example of dimensionality reduction results for a three-dimensional semantic space, where the central point represents the optimal predictive result. The purple data points in Fig. 3 represent a group of bad individuals located far from the target semantics. The remaining data points represent a group of good individuals. A perfect behavioral space should keep the best individual in each cell and remove all inferior individuals. However, Fig. 3 shows that many conventional dimensionality reduction methods fail to achieve this goal. For example, with PCA, large parts of behavioral space are filled with purple data points, indicating that several bad individuals will be retained due to their excellent diversity. In this example, only KPCA with a cosine kernel and Isomap place good individuals in the entire space. However, Isomap focuses to preserve the local structure in a low-dimensional space. Thus, it may fail to perform well if good and bad points are connected and distributed on a single manifold, such as the "Swiss roll" data [1]. In contrast, KPCA with a cosine kernel only considers the angles between points during the dimensionality reduction procedure. Thus, individuals with different fitness values will fall within the same region if the cosine similarity between predicted values is high.

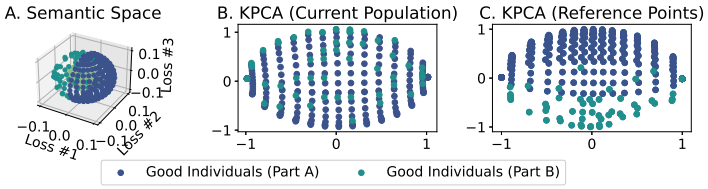

Fig. 4. An illustrative example to show the effect of constructing a dimensionality reduction model on the current population and symmetrical reference semantic points.

3.3 Reference Semantic Points

To train a dimensionality reduction model to generate a behavior space capturing the distribution of high-quality individuals only, previous research [24] used the semantics of good individuals in the final population of a GP run. These semantic points are used to construct a behavior space and are referred to as reference points. However, the target semantics $\{y_1, y_2, \ldots, y_n\}$ is available for supervised learning tasks. Consequently, for each semantic point $\{\hat{y_1}, \hat{y_2}, \ldots, \hat{y_n}\}$ in the current population P and the current MAP-Elites grid E, a reference point can be generated with $\{(1-\alpha)*y_1+\alpha*\hat{y_1}, (1-\alpha)*y_2+\alpha*\hat{y_2}, \ldots, (1-\alpha)*y_n+\alpha*\hat{y_n}\}$, where α is a hyperparameter indicating how close a synthetic reference point is to the target semantics. α is empirically set as 0.1 in this paper. Notably, for each individual, we not only synthesize a reference point based on α but also generate a symmetric reference point with $-\alpha$. Generating a symmetric reference point guarantees that the average of all reference points equals the target semantics. After obtaining reference points, we can train a dimensionality reduction model using these points and then apply the trained model to the semantic points of the current population to construct a MAP-Elites grid.

Fig. 4 shows the behavioral space of two imbalanced sets of data points using reference points or not. As shown in Fig. 4B, if constructing a KPCA model on the current population, the blue points representing individuals over-estimating the value of the first sample and the green points representing individuals under-estimating the value of the first sample will be mixed. This mix makes it very hard to obtain complementary base learners by selecting the best one from each cell. Conversely, if KPCA is constructed with symmetrical reference points, complementary points will be dispersed across distinct regions of a behavioral space. MAP-Elites can easily obtain a collection of complementary base learners.

Nevertheless, it is important to note that pre-training a dimensionality reduction model based on reference points is risky. Fig. 5 provides an example of dimensionality reduction results based on online and offline modes. The online mode means training a dimensionality reduction model on the current population, whereas offline means training a model on reference points. Both kinds of models will be applied to the current population for dimensionality reduction. In Fig. 5, the colored points on the outer circle represent models in the current population, while the red points on the inner circle represent synthetic reference

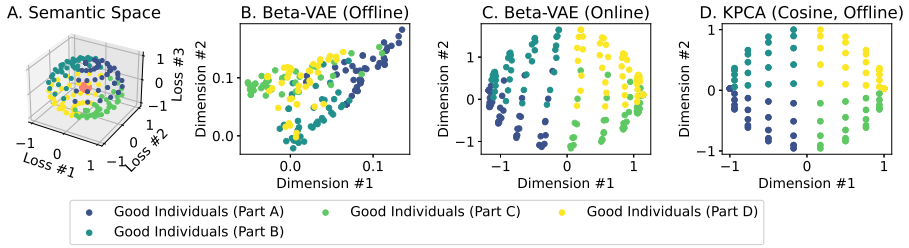

Fig. 5. Online versus offline dimensionality reduction paradigms.

points. Fig. 5B provides dimension reduction results in the offline mode, while Fig. 5C provides a comparative example of dimension reduction results in the online mode. Comparing these two plots reveals that VAE fails to generate an appropriate behavioral space for mismatched population distribution, as colored points are highly concentrated in the center. In contrast, if we train a VAE on the current population, the behavioral space can maintain the current population's structure, proving that the problem is the mismatched distribution, not using a VAE. Furthermore, Fig. 5D shows that KPCA with a cosine kernel is not significantly affected by this issue, illustrating another advantage of using KPCA. To sum up, synthesizing reference points is helpful to generate a good behavioral space, but it should be paired with an appropriate dimensionality reduction method to alleviate the negative impact of the mismatched distribution.

4 Experiment Settings

In this section, several experiments are conducted to answer the following questions:

- Does the cosine kernel PCA-based dimensionality reduction method result in a better ensemble model in MEGP when compared to the commonly used dimensionality methods?
- Does a behavioral space generated by reference points improve the predictive performance of the ensemble model?

4.1 Datasets

In this paper, we conduct experiments on the Penn Machine Learning Benchmark (PMLB) [28], a curated list of datasets derived from OpenML datasets [33]. For comparison of dimensionality reduction algorithms and reference points, experiments are carried out on 108 datasets in PMLB with less than 10000 instances due to limited computational resources. For comparison with other algorithms, experiments are performed on standard PMLB with 122 datasets. Among these datasets, 63 were synthesized by the Friedman function and are synthetic datasets, while the remaining 59 are real-world datasets.

Table 1. Parameter settings for MEGP.

Parameter	Value
Population Size	1000
Maximal Number of Generations	50
Crossover and Mutation Rates	0.9 and 0.1
Maximum Tree Depth	8
Maximum Initial Tree Depth	2
Number of Trees in An Individual	10
Dimensionality of the MAP-Elites Grid	10
Functions	+, -, *, AQ, Sin, Cos, Abs, Max, Min, Negative

4.2 Experimental Protocol

For the following experiments, we follow a conventional experimental protocol in the evolutionary computation domain, i.e., each algorithm is tested on each dataset with 30 independent runs. In each run, 80% of the data is used as the training data, and the remaining is used as the test data. After runs are finished, a Wilcoxon rank sum test with a significance level of 0.05 is used to verify the effectiveness of the proposed method. As for the comparison with other machine learning algorithms, we follow the convention of SRBench [5], i.e., each algorithm is tested on each dataset with 10 independent runs. The hyper-parameters of benchmark algorithms are tuned using the halving-grid search method [19] to ensure that the prediction performance of benchmark algorithms is fully exploited.

4.3 Parameter Settings

Table 1 presents the parameter settings of MEGP. The population size and crossover rate are conventional settings for GP [8]. Analytical quotient (AQ) is used in MEGP to replace the division operator in order to avoid division by zero. AQ is defined as $AQ(a, b) = \frac{a}{\sqrt{1+b^2}}$, where a and b represent two input variables.

4.4 Benchmark Dimensionality Reduction Methods

Here, we select 9 popular dimensionality reduction methods for comparisons because these methods are widely used in the machine learning field [29]. A brief introduction of investigated methods is as follows:

- Principal Component Analysis (PCA) [31]: PCA is a linear dimensionality reduction method that finds new dimensions to maximize variance in the data.
- Kernel PCA with RBF/Polynomial/Cosine Kernel [31]: These three methods are based on PCA, with the difference of using RBF/Polynomial/Cosine kernels to calculate the similarity between points instead of the covariance.
- T-distributed Stochastic Neighbor Embedding (t-SNE-Euclidean/Cosine) [20]: t-SNE is a non-linear dimensionality reduction method that keeps both the local and the global structure. The key idea is to minimize the Kullback–Leibler divergence between high-dimensionality representation and low-dimensionality representation through gradient descent.

Table 2. Experimental results of nine dimensionality reduction methods in MEGP ("+", "∼" or "-" mean that a method in a row is significantly better than, similar to, and worse than the method in the column).

	t-SNE(COSINE)	**PCA**	**KPCA(RBF)**
KPCA(COSINE)	12(+)/96(∼)/0(-)	45(+)/63(∼)/0(-)	62(+)/45(∼)/1(-)
TSNE(Cosine)	—	30(+)/78(∼)/0(-)	54(+)/54(∼)/0(-)
PCA	—	—	46(+)/60(∼)/2(-)
	KPCA(POLY)	**TSNE**	**Beta-VAE**
KPCA(COSINE)	74(+)/34(∼)/0(-)	58(+)/50(∼)/0(-)	71(+)/37(∼)/0(-)
TSNE(Cosine)	75(+)/33(∼)/0(-)	53(+)/55(∼)/0(-)	67(+)/41(∼)/0(-)
PCA	70(+)/38(∼)/0(-)	12(+)/94(∼)/2(-)	47(+)/61(∼)/0(-)
KPCA(RBF)	70(+)/37(∼)/1(-)	0(+)/69(∼)/39(-)	8(+)/81(∼)/19(-)
KPCA(POLY)	—	0(+)/39(∼)/69(-)	5(+)/39(∼)/64(-)
TSNE	—	—	32(+)/76(∼)/0(-)
	Isomap	**SpectralEmbedding**	
KPCA(COSINE)	63(+)/45(∼)/0(-)	64(+)/44(∼)/0(-)	
TSNE(Cosine)	58(+)/50(∼)/0(-)	55(+)/53(∼)/0(-)	
PCA	31(+)/76(∼)/1(-)	30(+)/77(∼)/1(-)	
KPCA(RBF)	7(+)/64(∼)/37(-)	1(+)/82(∼)/25(-)	
KPCA(POLY)	0(+)/41(∼)/67(-)	0(+)/42(∼)/66(-)	
TSNE	20(+)/87(∼)/1(-)	17(+)/91(∼)/0(-)	
Beta-VAE	2(+)/87(∼)/19(-)	1(+)/94(∼)/13(-)	
Isomap	—	8(+)/97(∼)/3(-)	

– Beta-VAE [14]: Beta-VAE is a deep-learning-based dimensionality reduction method. It maps input variables into a multivariate latent distribution. Unlike AE, it optimizes the reconstruction error and Kullback–Leibler divergence simultaneously to make the latent distribution approximate the expected distribution. A hyperparameter β is used to control the tradeoff between minimizing the reconstruction error and Kullback–Leibler divergence.
– Isomap [1]: Isomap is a manifold learning method to keep the local structure. It tries to keep the geodesic distance between points the same in the high-dimension and the low-dimension space.
– SpectralEmbedding [23]: Spectral embedding is similar to KPCA, but with the difference in that the eigen-decomposition is performed on a Laplacian matrix rather than on a kernel-matrix.

5 Experimental Results

5.1 Comparisons of MAP-Elites using Different Dimensionality Reduction Methods

In this section, we present the experimental results of using 9 dimensionality reduction methods in MAP-Elites. MAP-Elites with cosine-kernel-based PCA significantly outperform beta-VAE on 71 out of the 108 datasets, see Table 2. On the other 37 datasets, the two methods have comparable performance. To examine the results in more detail, we plot curves of the test score of the ensemble model, average fitness of individuals in the MAP-Elites grid, and mean negative cosine similarity of individuals in the MAP-Elites grid against the number of

generations in Fig. 6, Fig. 7, and Fig. 8, respectively. Fig. 6 demonstrates that using KPCA with a cosine kernel is superior to using other methods in terms of the test R^2 score. To find out the reasons, Fig. 7 shows the average fitness of all base learners in the MAP-Elites grid. It indicates that some dimensionality reduction methods such as KPCA (POLY) and KPCA (RBF), make MAP-Elites select individuals with an average fitness lower than 0.8 and this may impair the accuracy of the ensemble model. Moreover, the superior performance of MAP-Elites with KPCA not only comes from selecting good fitness individuals but also from selecting individuals with a high level of diversity. To validate whether cosine-kernel-based KPCA is useful for keeping archive diversity, the average negative cosine similarity of base learners is presented in Fig. 8. Here, we use cosine similarity as opposed to Euclidean distance because a large negative cosine similarity indicates good complementarity between base learners, whereas a large Euclidean distance may be caused by base learners with very low accuracy. As shown in Fig. 8, the negative cosine similarity of base learners consistently decreases as evolution goes on when using PCA as the dimensionality reduction method. In contrast, the average negative cosine similarity of cosine-kernel-based KPCA stays at a stable level after 30 iterations, and it is higher than the results of PCA, providing evidence that using cosine-kernel PCA as a dimensionality reduction method is beneficial. It is worth noting that cosine KPCA does not have the best negative cosine similarity because half of the individuals with poor performance will be filtered out as shown in Algorithm 1, and such an elimination process may reduce the negative cosine similarity. Other methods, like KPCA (POLY) and KPCA (RBF), are less affected by this process because they select a large number of bad individuals.

5.2 Impact of Using Reference Points

In this section, we investigate whether inducing a behavioral space from reference points is beneficial. We compare the prediction performance on the test set with and without reference points. Several dimensionality reduction techniques, such as t-SNE and spectral embedding, are omitted because they cannot predict unseen data points. For the remaining methods, the results with and without reference points are shown in Fig. 9. As shown, reference points improve the predictive performance of cosine-kernel-based KPCA on 39 datasets and do not degrade it on any other dataset. However, reference points do not work well with PCA and Beta-VAE, and even worsen performance on 46 and 7 datasets, respectively. These results validate our assumptions in Section 3. Consequently, we can conclude that using reference points to develop a dimensionality reduction model is useful, but it should be paired with suitable dimensionality reduction techniques.

5.3 Comparison with Other Machine Learning and Symbolic Regression Methods

To validate the efficacy of the proposed method, we compare MEGP to 14 symbolic regression methods and 8 machine learning methods on 122 datasets

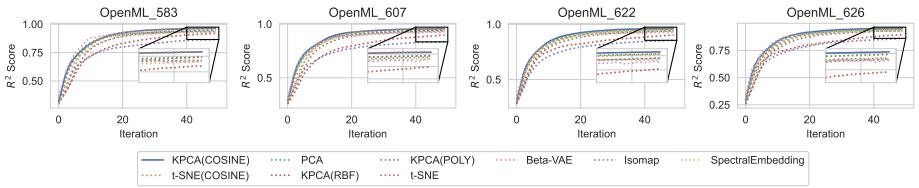

Fig. 6. Test R^2 score with respect to the number of generations

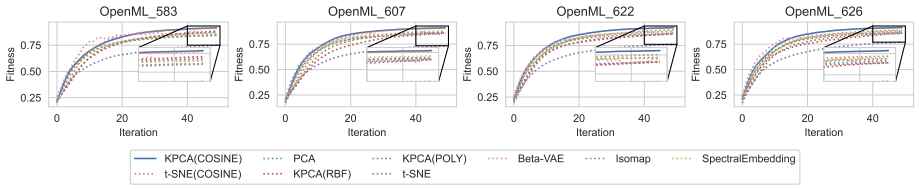

Fig. 7. Average fitness of individuals in an archive with respect to the number of generations

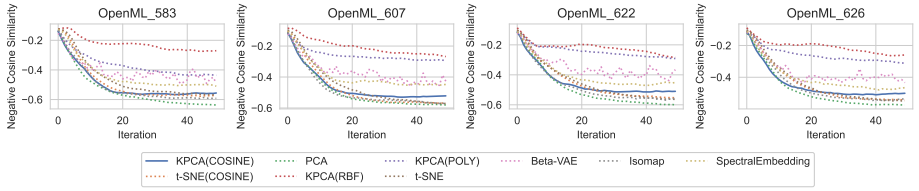

Fig. 8. Average negative cosine similarity of semantic vectors in the archive with respect to the number of generations

from SRBench. Fig. 10 demonstrates the distribution of test R^2 scores for various algorithms. The red dot denotes the mean values of the median R^2 scores for all datasets. This figure depicts that MEGP outperforms other SR and ML methods on average for both synthetic and real-world datasets. For example, on real-world datasets, MEGP has an average R^2 score of 0.704, outperforming a representative GP-based ensemble learning method 2SEGP [34], which has an average R^2 score of only 0.692. The advantage is significant with a p-value of $5 * 10^{-5}$.

6 Conclusions

In this paper, a new GP-based ensemble learning method named MEGP is proposed. First, MEGP uses an angle-based dimensionality reduction method in MAP-Elites to preserve good and complementary individuals. Meanwhile, MEGP synthesizes reference points to deal with an imbalanced distribution of good individuals. Experimental results show that MEGP with cosine-kernel KPCA outperforms MEGP with PCA on 45 datasets and is comparable to PCA on 63 datasets. Also, reference points improve its performance on 39 datasets and do not hurt it on others. Experimental results on SRBench demonstrate that MEGP outperforms 22 ML and SR algorithms across 122 datasets. This paper only examines the performance of MEGP in the regression scenario, it

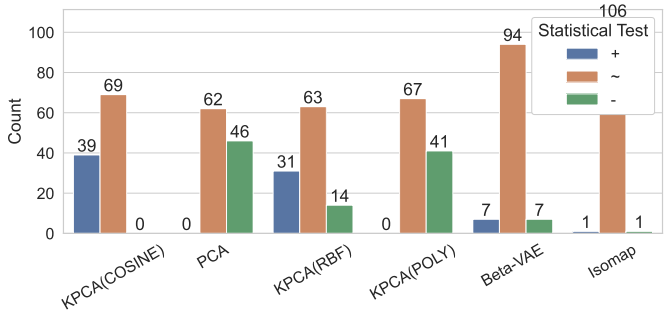

Fig. 9. Impact of reference semantic points on different dimensionality reduction techniques ("+", "∼" or "-" mean that using reference points is significantly better than, similar to, and worse than not using reference points on the specific dimensionality reduction method).

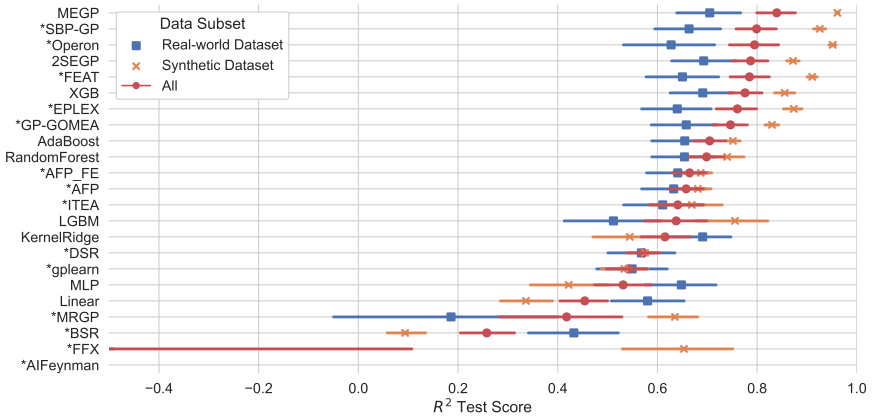

Fig. 10. Experimental results on 122 PMLB datasets (Results for AIFeynman are out of bounds and are therefore not shown).

would be intriguing to study MEGP in the classification scenario in the future. Furthermore, while this paper focuses on improving MAP-Elites, the findings may also be applicable to archive-based quality-diversity optimization methods, which merit further investigation in the future. Last, finding new ways to aggregate GP models is also a promising direction to investigate in the future.

References

1. Balasubramanian, M., Schwartz, E.L.: The isomap algorithm and topological stability. Science **295**(5552), 7–7 (2002)
2. Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D.: Genetic programming: an introduction: on the automatic evolution of computer programs and its applications. Morgan Kaufmann Publishers Inc. (1998)
3. Boisvert, S., Sheppard, J.W.: Quality diversity genetic programming for learning decision tree ensembles. In: European Conference on Genetic Programming (Part of EvoStar). pp. 3–18. Springer (2021)

4. Bruneton, J.P., Cazenille, L., Douin, A., Reverdy, V.: Exploration and exploitation in symbolic regression using quality-diversity and evolutionary strategies algorithms. arXiv preprint arXiv:1906.03959 (2019)
5. Cava, W.L., Orzechowski, P., Burlacu, B., de Franca, F.O., Virgolin, M., Jin, Y., Kommenda, M., Moore, J.H.: Contemporary symbolic regression methods and their relative performance. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1) (2021)
6. Cawley, G.C., Talbot, N.L.: Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. Neural Networks **17**(10), 1467–1475 (2004)
7. Cazenille, L.: Ensemble feature extraction for multi-container quality-diversity algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 75–83 (2021)
8. Chen, Q., Xue, B., Zhang, M.: Preserving population diversity based on transformed semantics in genetic programming for symbolic regression. IEEE Transactions on Evolutionary Computation **25**(3), 433–447 (2020)
9. Cully, A., Clune, J., Tarapore, D., Mouret, J.B.: Robots that can adapt like animals. Nature **521**(7553), 503–507 (2015)
10. Dick, G., Owen, C.A., Whigham, P.A.: Evolving bagging ensembles using a spatially-structured niching method. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 418–425 (2018)
11. Dolson, E., Lalejini, A., Ofria, C.: Exploring genetic programming systems with map-elites. In: Genetic Programming Theory and Practice XVI, pp. 1–16. Springer (2019)
12. Gaier, A., Asteroth, A., Mouret, J.B.: Aerodynamic design exploration through surrogate-assisted illumination. In: 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. p. 3330 (2017)
13. Goldberg, D.E., Richardson, J., et al.: Genetic algorithms with sharing for multimodal function optimization. In: Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms. vol. 4149 (1987)
14. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. In: International Conference on Learning Representations (2017)
15. Iba, H.: Bagging, boosting, and bloating in genetic programming. In: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2. pp. 1053–1060 (1999)
16. La Cava, W., Helmuth, T., Spector, L., Moore, J.H.: A probabilistic and multi-objective analysis of lexicase selection and ε-lexicase selection. Evolutionary Computation **27**(3), 377–402 (2019)
17. La Cava, W., Singh, T.R., Taggart, J., Suri, S., Moore, J.H.: Learning concise representations for regression by evolving networks of trees. In: International Conference on Learning Representations (2018)
18. Lehman, J., Stanley, K.O.: Evolving a diversity of virtual creatures through novelty search and local competition. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation. pp. 211–218 (2011)
19. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., Talwalkar, A.: Hyperband: A novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research **18**(1), 6765–6816 (2017)
20. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(11) (2008)

21. Moraglio, A., Krawiec, K., Johnson, C.G.: Geometric semantic genetic programming. In: International Conference on Parallel Problem Solving from Nature. pp. 21–31. Springer (2012)
22. Mouret, J.B., Clune, J.: Illuminating search spaces by mapping elites. arXiv preprint arXiv:1504.04909 (2015)
23. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems **14** (2001)
24. Nickerson, K., Hu, T.: Principled quality diversity for ensemble classifiers using map-elites. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 259–260 (2021)
25. Nickerson, K., Kolokolova, A., Hu, T.: Creating diverse ensembles for classification with genetic programming and neuro-map-elites. In: European Conference on Genetic Programming (Part of EvoStar). pp. 212–227. Springer (2022)
26. Nikfarjam, A., Neumann, A., Neumann, F.: On the use of quality diversity algorithms for the traveling thief problem. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 260–268 (2022)
27. Nilsson, O., Cully, A.: Policy gradient assisted map-elites. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 866–875 (2021)
28. Olson, R.S., La Cava, W., Orzechowski, P., Urbanowicz, R.J., Moore, J.H.: Pmlb: a large benchmark suite for machine learning evaluation and comparison. BioData mining **10**(1), 1–13 (2017)
29. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. The Journal of Machine Learning Research **12**, 2825–2830 (2011)
30. Pietropolli, G., Manzoni, L., Paoletti, A., Castelli, M.: Combining geometric semantic gp with gradient-descent optimization. In: European Conference on Genetic Programming (Part of EvoStar). pp. 19–33. Springer (2022)
31. Schölkopf, B., Smola, A., Müller, K.R.: Kernel principal component analysis. In: International Conference on Artificial Neural Networks. pp. 583–588. Springer (1997)
32. Urquhart, N., Hart, E.: Optimisation and illumination of a real-world workforce scheduling and routing application (wsrp) via map-elites. In: International Conference on Parallel Problem Solving from Nature. pp. 488–499. Springer (2018)
33. Vanschoren, J., Van Rijn, J.N., Bischl, B., Torgo, L.: Openml: networked science in machine learning. ACM SIGKDD Explorations Newsletter **15**(2), 49–60 (2014)
34. Virgolin, M.: Genetic programming is naturally suited to evolve bagging ensembles. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 830–839 (2021)
35. Wang, S., Mei, Y., Zhang, M.: Novel ensemble genetic programming hyper-heuristics for uncertain capacitated arc routing problem. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1093–1101 (2019)
36. Wang, Y., Xue, K., Qian, C.: Evolutionary diversity optimization with clustering-based selection for reinforcement learning. In: International Conference on Learning Representations (2021)
37. Zhang, H., Zhou, A., Zhang, H.: An evolutionary forest for regression. IEEE Transactions on Evolutionary Computation **26**(4), 735–749 (2022)