P-Mixup: Improving Generalization Performance of Evolutionary Feature Construction with Pessimistic Vicinal Risk Minimization

Hengzhe Zhang, Qi Chen, Bing Xue, Wolfgang Banzhaf, Mengjie Zhang

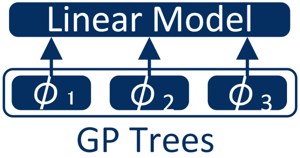
Victoria University of Wellington

06/09/2024

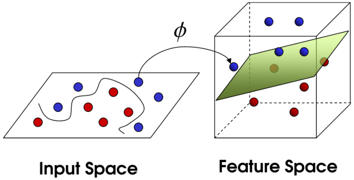
Table of Contents

Background

- **Objective:** Construct a set of new features, $\{\phi_1, \ldots, \phi_m\}$, to *enhance learning* on the dataset $\{\{x_1, y_1\}, \ldots, \{x_n, y_n\}\}$ compared to learning on the original features $\{x^1, \ldots, x^p\}$.
- **Approaches:**
  - Kernel Methods: Black-box, non-parametric.
  - Neural Networks: Black-box, gradient-based.
  - Genetic Programming: Interpretable, gradient-free.



(a) Feature Construction on Linear Regression



(b) New Feature Space

Overfitting

- **Challenge:** Overfitting is a significant issue in evolutionary feature construction.
- **Phenomenon:** Overfitted models perform well on training data but poorly on unseen data.
- **Cause:** Models may become too complex and fit noise in the training data, especially when:
  - Sample size is limited.
  - Data contains noise.

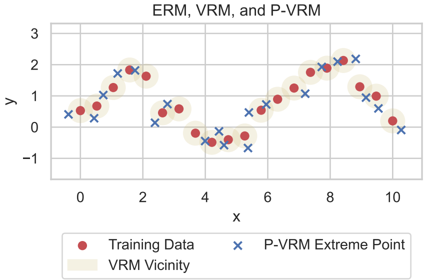
Objective

Mitigate overfitting to enhance generalization and robustness of constructed features.

- **ERM (Empirical Risk Minimization):**

$$\mathcal{L}(f) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) \tag{1}$$

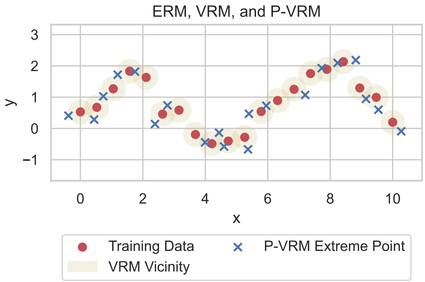
- **Concept:**
  - ERM minimizes the average loss over the training data.
  - Focuses only on given data points (x_i, y_i) without considering neighbors or unseen data.



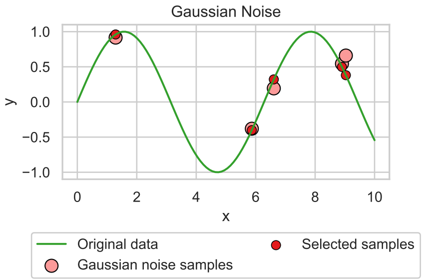
- **VRM (Vicinal Risk Minimization):**

$$\mathcal{VIC}(f) = \frac{1}{n}\sum_{i=1}^{n}\int \mathcal{L}(f(\mathbf{x}), y_i)dP_{x_i}(\mathbf{x}) \tag{2}$$

- **Concept:**
  - VRM incorporates vicinal samples (neighbors of training points).
  - It minimizes the expected loss over a distribution of neighboring samples P_{x_i} around each training point.



- **Concept:**
  - Gaussian Synthesis generates vicinal samples by adding Gaussian noise to the original training data.
  - The noise is sampled from a normal distribution $\mathcal{N}(0, \sigma^2)$, where σ represents the standard deviation.
- **Limitation:**
  - Gaussian noise may create synthetic samples that do not lie on the true data manifold.



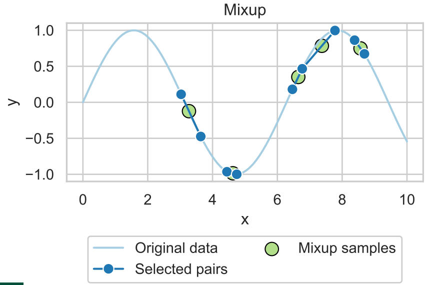
Method

- **Concept:**
  - MixUp Synthesis generates new vicinal samples by linearly combining two training samples x_a and x_b.
  - The formula for the synthesized sample is:

$$x_{\text{mixup}} = \lambda \cdot x_a + (1 - \lambda) \cdot x_b \quad (3)$$

where $\lambda \in [0, 1]$ is a randomly sampled mixing ratio.

- **Objective:** Create synthetic samples that lie between two real data points, helping the model generalize by learning from intermediate data.

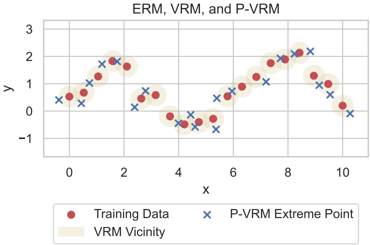


P-VRM (Pessimistic Vicinal Risk Minimization):

$$\mathcal{V}(f) = \frac{1}{n} \sum_{i=1}^{n} \max_{x \in \mathbb{N}(x_i)} \mathcal{L}(f(\mathbf{x}), y_i) \tag{4}$$

Concept:

- P-VRM minimizes the worst-case loss among the vicinal samples in the neighborhood $\mathbb{N}(x_i)$ of each training point.
- Focusing on the worst-case scenario, the model becomes more robust and stable, improving generalization to unseen samples.



Key Algorithm Components

- **Population Initialization:** GP trees are initialized using the ramped-half-and-half method.
- **Parent Selection:** Lexicase selection is used to select parents by iteratively eliminating poorly performing individuals on some instances.
- **Offspring Generation:** Random subtree crossover, mutation, and dynamic tree addition/deletion operators.
- **Objective Evaluation:**
  - Vicinal data is synthesized using the mixup technique.
  - Pessimistic vicinal risk and cross-validation loss are evaluated.
- **Final Model Selection:** Based on the lowest vicinal risk.

VICTORIA UNIVERSITY OF WELLINGTON TE HERENGA WAKA NEW ZEALAND

- **Theorem:** Pessimistic MixUp encourages local linearity around each sample $x_a \in X$ by minimizing the objective:

$$\max_{\lambda,(x_b,y_b)\in\mathbb{N}(x_a)} (0.5 - |\lambda - 0.5|)^2 (y_b - y_a - \nabla f(x_a)^\top (x_b - x_a))^2$$

  where $\lambda \sim \text{Beta}(\alpha, \beta)$ is the MixUp ratio.
- The optimal gradient is:

$$\nabla f(x_a) = \frac{(y_b^* - y_a)}{\|\Delta x\|^2} \Delta x$$

  where $\Delta x = (x_b^* - x_a)$.

Optimal Gradient Interpretation

- **The optimal gradient is:**

$$\nabla f(x_a) = \frac{(y_b^* - y_a)}{\|\Delta x\|^2} \Delta x$$

  where $\Delta x = (x_b^* - x_a)$.
- This is not $\nabla f(x_a) = x_a^2$ or $\nabla f(x_a) = x_a^3$. Instead, it is a constant vector.
- A **constant gradient** implies that the function $f(x)$ behaves linearly in the local region around x_a.

Experimental Settings

Datasets

- **Source:** 58 real-world datasets from the Penn Machine Learning Benchmark (PMLB).
- **Criteria:** Excluded synthetic datasets.
- **Focus:** Emphasis on real-world applicability.

Parameter Settings

Parameter Settings for GP

Parameter	Value
Maximum Population Size	200
Number of Generations	100
Crossover Rate	0.9
Mutation Rate	0.1
Tree Addition Rate	0.5
Tree Deletion Rate	0.5
Initial Tree Depth	0-3
Maximum Tree Depth	10
Initial Number of Trees	1

Parameter	Value
Maximum Number of Trees	10
Elitism (Number of Individuals)	1
Bandwidth of Gaussian Kernel	0.5
β of Beta Distribution	1
Iterations of Risk Estimation (K)	10
Functions	+, -, *, AQ, Abs, Sqrt, Neg, Log, Max, Min, Sin, Cos, Square

Compared Methods:

- Standard GP without regularization
- Parsimonious Pressure (PP)
- Tikhonov Regularization (TK)
- Grand Complexity (GC)
- Rademacher Complexity (RC)
- Weighted MIC between Residuals and Variables (WCRV)
- Correlation between Input and Output Distances (IODC)

Results

- Training R^2: Standard GP is the best.

Statistical comparison of training R^2 scores.

	VRM	PP	RC	GC
P-VRM	0(+)/14(∼)/44(-)	28(+)/17(∼)/13(-)	56(+)/2(∼)/0(-)	46(+)/10(∼)/2(-)
VRM	—	48(+)/10(∼)/0(-)	58(+)/0(∼)/0(-)	56(+)/2(∼)/0(-)
PP	—	—	58(+)/0(∼)/0(-)	46(+)/12(∼)/0(-)
RC	—	—	—	1(+)/4(∼)/53(-)

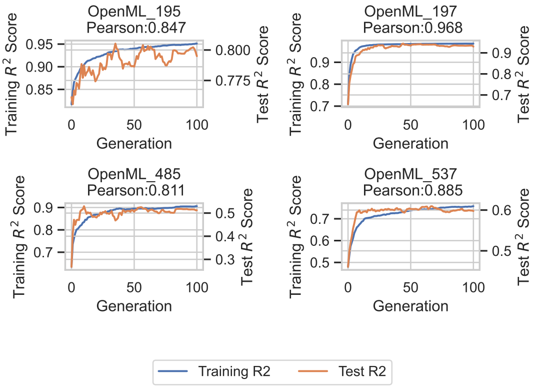
	IODC	TK	WCRV	Standard GP
P-VRM	37(+)/15(∼)/6(-)	36(+)/15(∼)/7(-)	39(+)/13(∼)/6(-)	1(+)/13(∼)/44(-)
VRM	53(+)/5(∼)/0(-)	53(+)/4(∼)/1(-)	48(+)/8(∼)/2(-)	2(+)/38(∼)/18(-)
PP	29(+)/23(∼)/6(-)	32(+)/20(∼)/6(-)	38(+)/15(∼)/5(-)	0(+)/3(∼)/55(-)
RC	1(+)/4(∼)/53(-)	0(+)/1(∼)/57(-)	2(+)/9(∼)/47(-)	0(+)/0(∼)/58(-)
GC	12(+)/27(∼)/19(-)	11(+)/26(∼)/21(-)	23(+)/25(∼)/10(-)	0(+)/1(∼)/57(-)
IODC	—	16(+)/19(∼)/23(-)	27(+)/19(∼)/12(-)	0(+)/0(∼)/58(-)
TK	—	—	31(+)/12(∼)/15(-)	0(+)/4(∼)/54(-)
WCRV	—	—	—	0(+)/2(∼)/56(-)

Test Performance

- Test R^2: P-VRM significantly improves generalization compared to standard GP and other overfitting control methods.
- P-VRM outperforms VRM, indicating the effectiveness of pessimistic vicinal risk minimization.

	VRM	**PP**	**RC**	**GC**
P-VRM	22(+)/29(∼)/7(-)	22(+)/31(∼)/5(-)	46(+)/10(∼)/2(-)	23(+)/32(∼)/3(-)
VRM	—	11(+)/38(∼)/9(-)	38(+)/10(∼)/10(-)	16(+)/29(∼)/13(-)
PP	—	—	38(+)/13(∼)/7(-)	15(+)/34(∼)/9(-)
RC	—	—	—	5(+)/10(∼)/43(-)
	IODC	**TK**	**WCRV**	**Standard GP**
P-VRM	31(+)/24(∼)/3(-)	44(+)/12(∼)/2(-)	38(+)/18(∼)/2(-)	31(+)/21(∼)/6(-)
VRM	24(+)/25(∼)/9(-)	28(+)/29(∼)/1(-)	27(+)/23(∼)/8(-)	22(+)/34(∼)/2(-)
PP	23(+)/29(∼)/6(-)	25(+)/31(∼)/2(-)	26(+)/26(∼)/6(-)	25(+)/25(∼)/8(-)
RC	5(+)/21(∼)/32(-)	5(+)/20(∼)/33(-)	11(+)/17(∼)/30(-)	13(+)/14(∼)/31(-)
GC	25(+)/26(∼)/7(-)	23(+)/33(∼)/2(-)	25(+)/30(∼)/3(-)	24(+)/21(∼)/13(-)
IODC	—	19(+)/24(∼)/15(-)	15(+)/30(∼)/13(-)	21(+)/16(∼)/21(-)
TK	—	—	11(+)/33(∼)/14(-)	10(+)/27(∼)/21(-)
WCRV	—	—	—	17(+)/22(∼)/19(-)

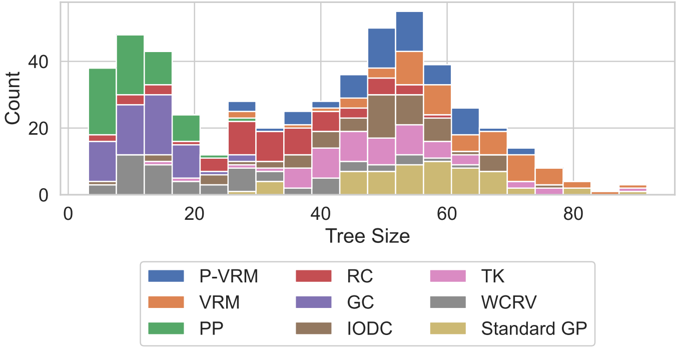
- **Correlation between training and test R^2 scores:** P-VRM method effectively controls overfitting.



Evolutionary plots of the *training and test R^2 scores* for VR.

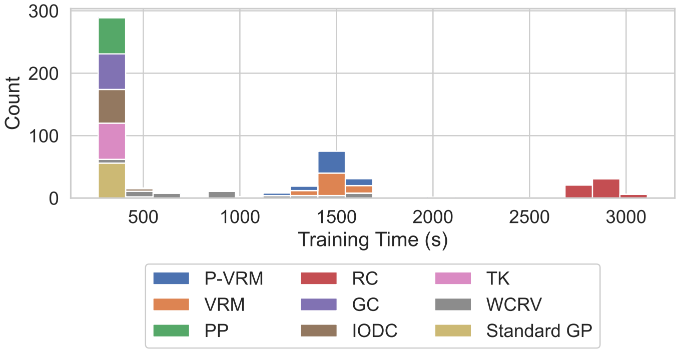
Tree Size

- **Tree sizes:** P-VRM does not significantly reduce tree size compared to standard GP.



Distribution of tree sizes across methods.

- **Training Time:** P-VRM method is computationally more intensive.



Distribution of training time (seconds) across methods.

- **Key Questions:**
  - Is MixUp better than Gaussian perturbation for generating vicinal samples?
- **MixUp vs. Gaussian Noise:**
  - P-VRM outperforms P-GVRM on 23 datasets, worse on only 8.
  - MixUp better aligns with the data manifold than Gaussian noise.

R^2 Score Comparison

	P-GVRM	**GVRM**
P-VRM	23(+)/27(∼)/8(-)	30(+)/21(∼)/7(-)
P-GVRM	—	22(+)/34(∼)/2(-)

Conclusions

VICTORIA UNIVERSITY OF WELLINGTON TE HERENGA WAKA NEW ZEALAND

Key Takeaways

- **P-VRM** minimizes worst-case vicinal risks for improved robustness.
- **MixUp-based vicinal samples** help the model generalize between real data points.
- **Empirical results:** P-VRM reduces overfitting and outperforms traditional methods.

Thanks for listening!

Email: Hengzhe.zhang@ecs.vuw.ac.nz

GitHub Project: https://github.com/hengzhe-zhang/EvolutionaryForest