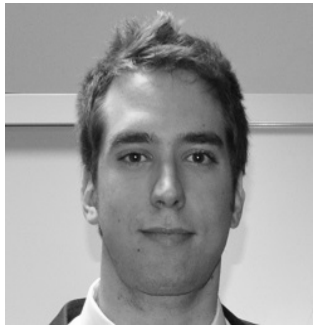
SGP-DT: Semantic Genetic Programming Based on Dynamic Targets

Stefano Ruberto



Valerio Terragni

Jason H. Moore

Penn
University of Pennsylvania

evo* 2020

Università della Svizzera italiana

Symbolic Regression

Training Cases

$$x_1\ x_2\ x_3\ \dots\ x_n \qquad \hat{y}_1$$

$$x_1\ x_2\ x_3\ \dots\ x_n \qquad \hat{y}_2$$

$$\dots\dots \qquad \dots$$

$$x_1\ x_2\ x_3\ \dots\ x_n \qquad \hat{y}_m$$

Symbolic Regression

Training Cases

$$F(\, x_1\ x_2\ x_3 \ldots x_n\,) = \hat{y}_1$$

$$F(\, x_1\ x_2\ x_3 \ldots x_n\,) = \hat{y}_2$$

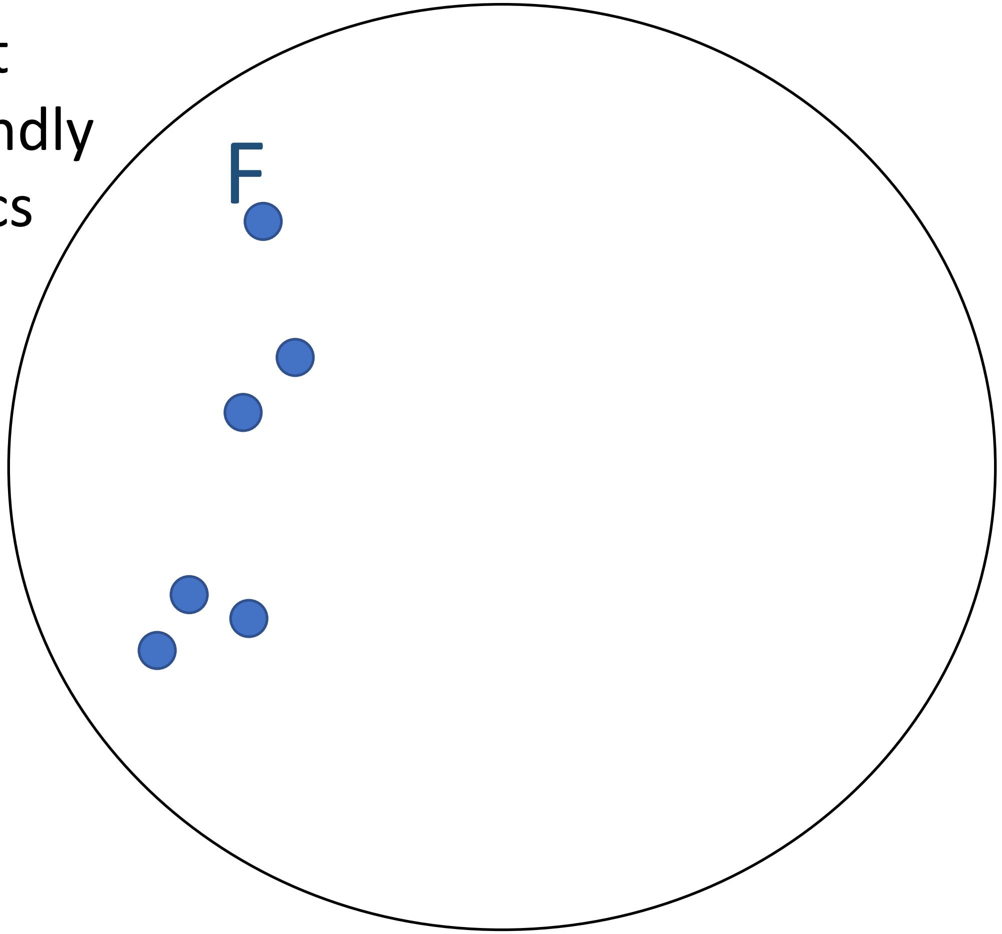
$$\ldots\ldots \qquad \ldots$$

$$F(\, x_1\ x_2\ x_3 \ldots x_n\,) = \hat{y}_m$$

Genetic Programming (GP)

(Semantic) Search Space

GP operates at syntactic level blindly to the semantics



Training Cases

$$F(\ x_1\ x_2\ x_3\ \dots\ x_n\ ) = \quad y_1$$

$$F(\ x_1\ x_2\ x_3\ \dots\ x_n\ ) = \quad y_2$$

$$\dots\dots \qquad\qquad \dots$$

$$F(\ x_1\ x_2\ x_3\ \dots\ x_n\ ) = \quad y_m$$

Semantic Genetic Programming (SGP)

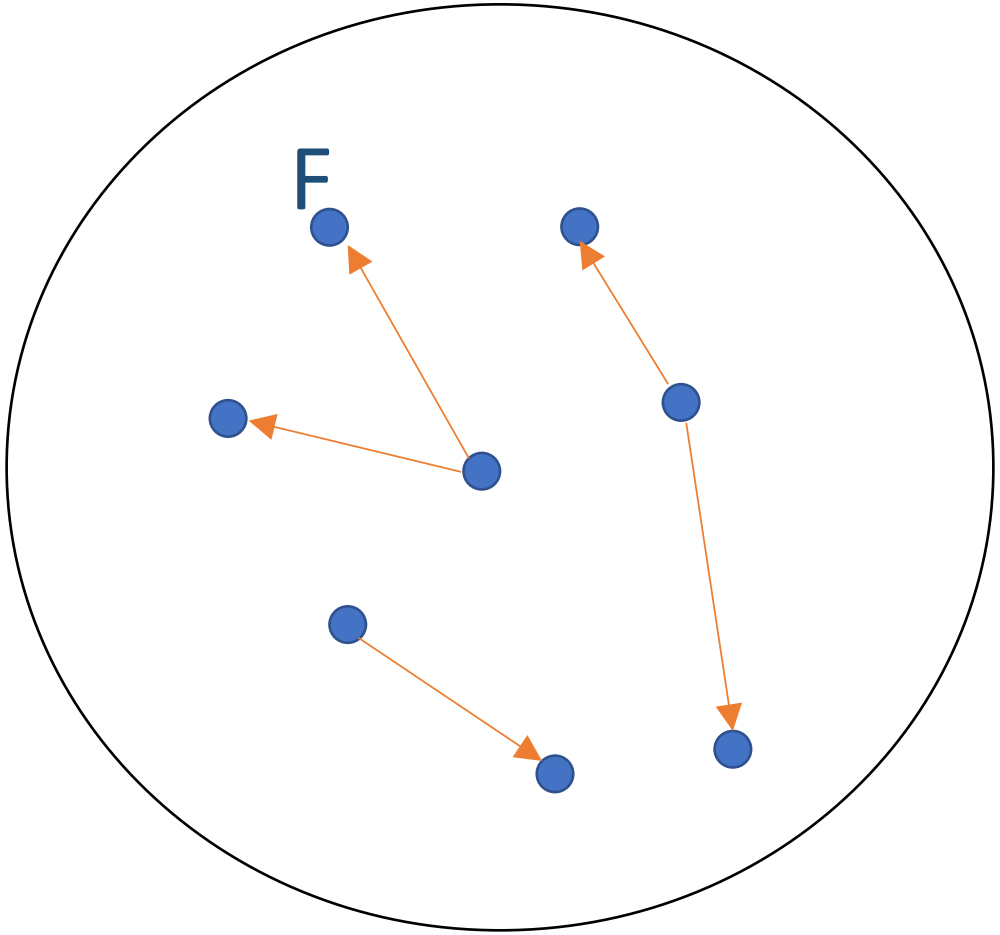
Training Cases

$$F(x_1\ x_2\ x_3\ \dots\ x_n) = y_1$$
$$F(x_1\ x_2\ x_3\ \dots\ x_n) = y_2$$
$$\dots\dots \qquad \dots$$
$$F(\underbrace{x_1\ x_2\ x_3\ \dots\ x_n}) = \underbrace{y_m}$$

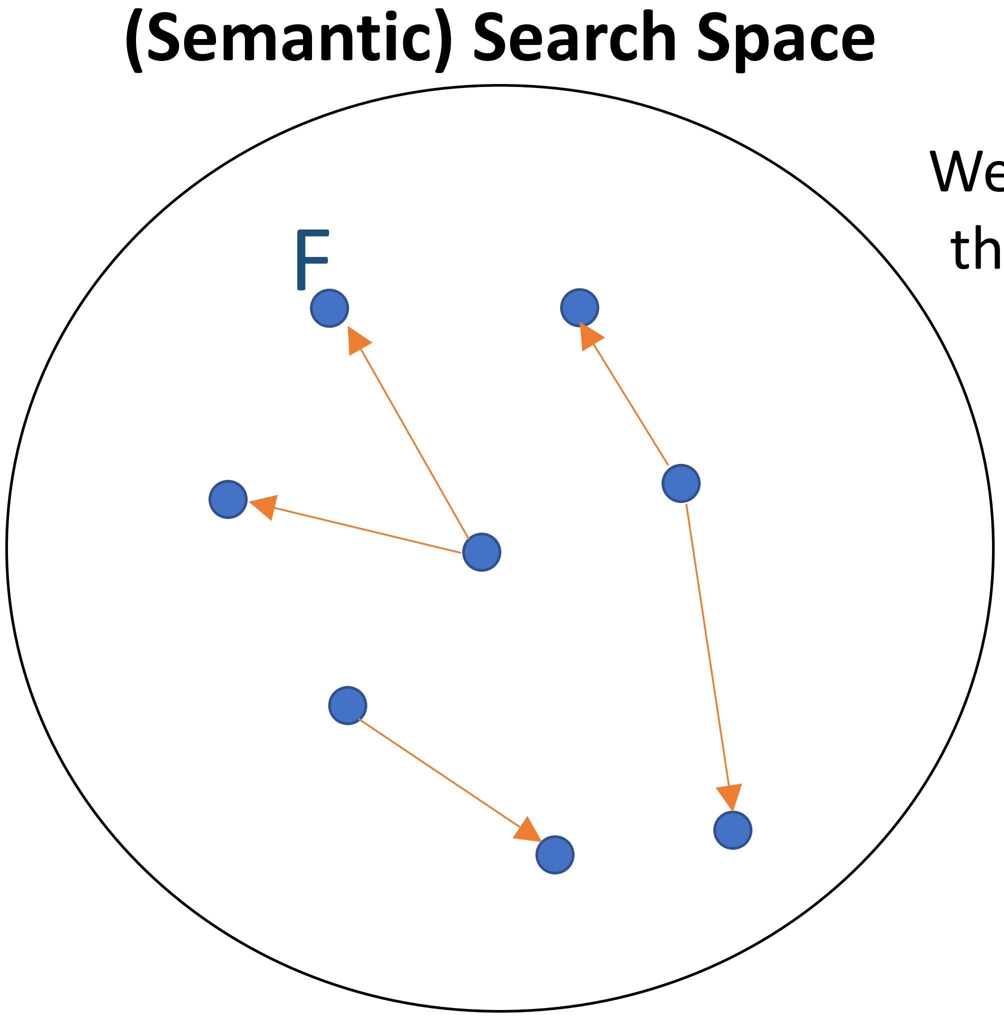
$$sem(F) = (y_1, y_2, \dots y_m)$$

Semantic Genetic Programming (SGP)

(Semantic) Search Space



Semantic Genetic Programming (SGP)



We need SGP approaches that effectively navigate the semantic space

The Goals of SGP Approaches

- computational efficient
- alleviate the bloat problem
- guarantee semantic properties of the offspring

The Goals of SGP Approaches

- computational efficient
- alleviate the bloat problem
- guarantee semantic properties of the offspring

It is difficult to achieve all of
these goals at the same time!

SGP-DT: SGP Based on Dynamic Targets

SGP-DT: SGP Based on Dynamic Targets

external iterations	**dynamic targets (dt)**	**GP runs**	**partial models (PM)**	**final model**
1		GP_1		
2		GP_2		
….		….		
n		GP_n		

SGP-DT: SGP Based on Dynamic Targets

external iterations	dynamic targets (dt)		GP runs		partial models (PM)	final model
1	$dt_1 = \hat{y}$	$\longrightarrow$	GP_1	$\longrightarrow$	$a_1 + b_1 \bullet PM_1$	
2			GP_2			
….			….			
n			GP_n			

SGP-DT: SGP Based on Dynamic Targets

external iterations	**dynamic targets (dt)**		**GP runs**		**partial models (PM)**	**final model**
1	$dt_1 = \hat{y}$	$\longrightarrow$	$\boxed{GP_1}$	$\longrightarrow$	$a_1 + b_1 \cdot PM_1$	
2	$dt_1 - sem(a_1 + b_1 \cdot PM_1)$	$\longrightarrow$	$\boxed{GP_2}$	$\longrightarrow$	$a_2 + b_2 \cdot PM_2$	
....			....			
n			$\boxed{GP_n}$			

SGP-DT: SGP Based on Dynamic Targets

external iterations	dynamic targets (dt)		GP runs		partial models (PM)	final model
1	$dt_1 = \hat{y}$	$\longrightarrow$	$\boxed{GP_1}$	$\longrightarrow$	$a_1 + b_1 \cdot PM_1$	
2	$dt_1 - sem(a_1 + b_1 \cdot PM_1)$	$\longrightarrow$	$\boxed{GP_2}$	$\longrightarrow$	$a_2 + b_2 \cdot PM_2$	
….	….		….		….	
n	$dt_{n-1} - sem(a_{n-1} + b_{n-1} \cdot PM_{n-1})$	$\longrightarrow$	$\boxed{GP_n}$	$\longrightarrow$	$a_n + b_n \cdot PM_n$	

SGP-DT: SGP Based on Dynamic Targets

external iterations	dynamic targets (dt)	GP runs	partial models (PM)	final model
1	$dt_1 = \hat{y}$	$\boxed{GP_1}$	$a_1 + b_1 \cdot PM_1$	
2	$dt_1 - sem(a_1 + b_1 \cdot PM_1)$	$\boxed{GP_2}$	$a_2 + b_2 \cdot PM_2$	$\sum_{i=1}^{n} a_i + bi \cdot PM_i$
….	….	….	….	
n	$dt_{n-1} - sem(a_{n-1} + b_{n-1} \cdot PM_{n-1})$	$\boxed{GP_n}$	$a_n + b_n \cdot PM_n$	

SGP-DT: SGP Based on Dynamic Targets

external iterations	dynamic targets (dt)	GP runs	partial models (PM)	final model
1	$dt_1 = \hat{y}$	$\rightarrow$ $\boxed{GP_1}$ $\rightarrow$	$a_1 + b_1 \cdot PM_1$	
2	$dt_1 - sem(a_1 + b_1 \cdot PM_1)$	$\rightarrow$ $\boxed{GP_2}$ $\rightarrow$	$a_2 + b_2 \cdot PM_2$	$\sum_{i=1}^{n} a_i + b_i \cdot PM_i$
....	....	....	....	
n	$dt_{n-1} - sem(a_{n-1} + b_{n-1} \cdot PM_{n-1})$	$\rightarrow$ $\boxed{GP_n}$ $\rightarrow$	$a_n + b_n \cdot PM_n$	

linear scaling

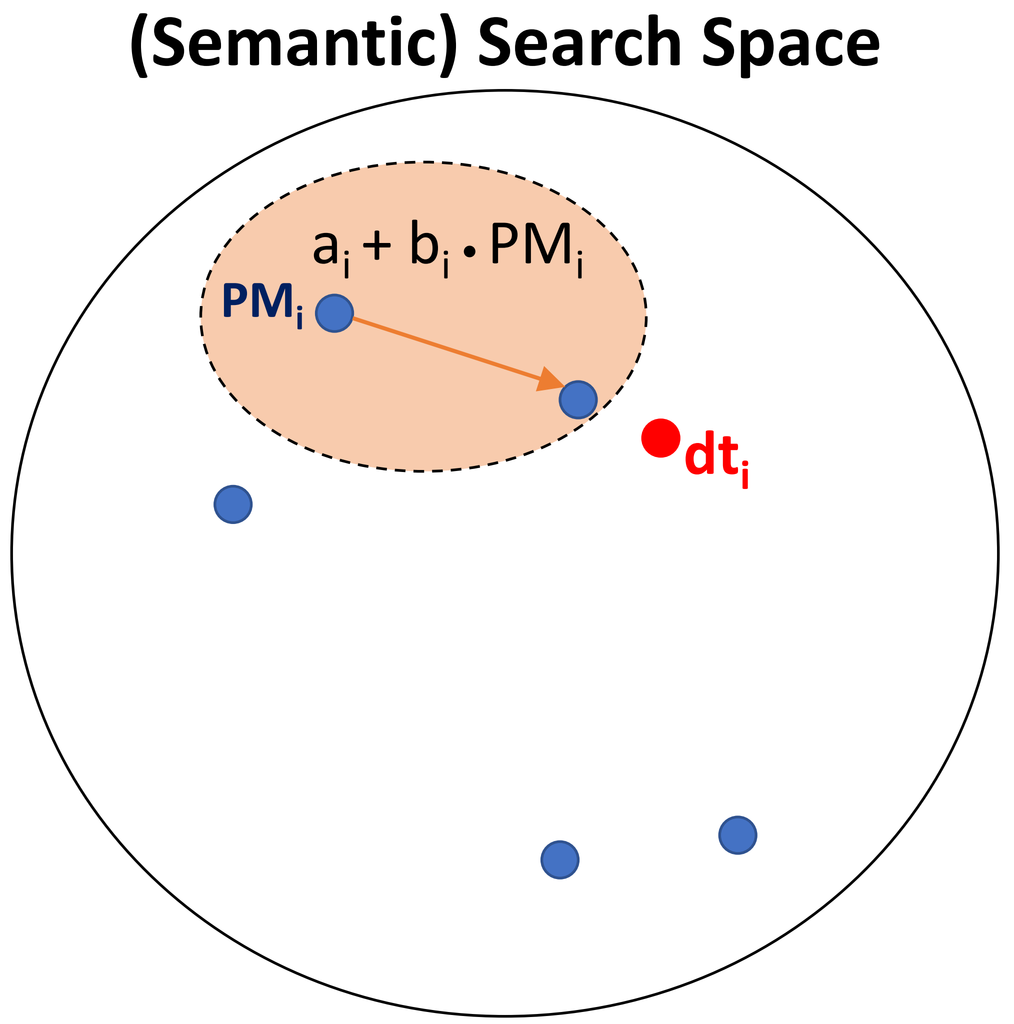
Linear Scaling (Keijzer 2003)

$$\mathcal{I}_{ls} = a + b \cdot \mathcal{I}$$

$$\text{where} \quad \underset{\textit{intersect}}{a = \overline{\hat{y}} - b \cdot \overline{y}} \quad \text{and} \quad \underset{\textit{slope}}{b =} \frac{\sum_{i=1}^{n}[(\hat{y}_i - \overline{\hat{y}}) \cdot (y_i - \overline{y})]}{\sum_{i=1}^{n}[(y_i - \overline{y})^2]}$$

- it guarantees a bound on the error
- directly optimizes the model without waiting that GP spontaneously evolve it

Linear Scaling (Keijzer 2003)



SGP-DT: SGP Based on Dynamic Targets

external iterations	dynamic targets (dt)	GP runs	partial models (PM)	final model
1	$dt_1 = \hat{y}$	**GP_1**	$a_1 + b_1 \cdot \mathbf{PM_1}$	
2	$dt_1 - sem(a_1 + b_1 \cdot PM_1)$	**GP_2**	$a_2 + b_2 \cdot \mathbf{PM_2}$	$\sum_{i=1}^{n} a_i + bi \cdot \mathbf{PM_i}$
....	....	....	....	
n	$dt_{n-1} - sem(a_{n-1} + b_{n-1} \cdot PM_{n-1})$	**GP_n**	$a_n + b_n \cdot \mathbf{PM_n}$	

Individuals

Individual:

Tree-like expression

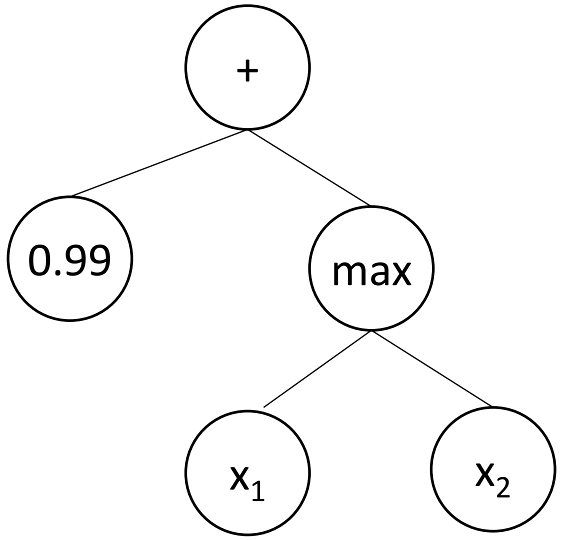
Symbols:

+, -, x, /, ERC [-1 ; +1], Min and Max

Genetic operators:

Mutation: subtree substitution

Elite: one individual



Fitness Function

$$error = sem(\mathrm{a} + \mathrm{b} \cdot \mathrm{I}) - \hat{y}$$

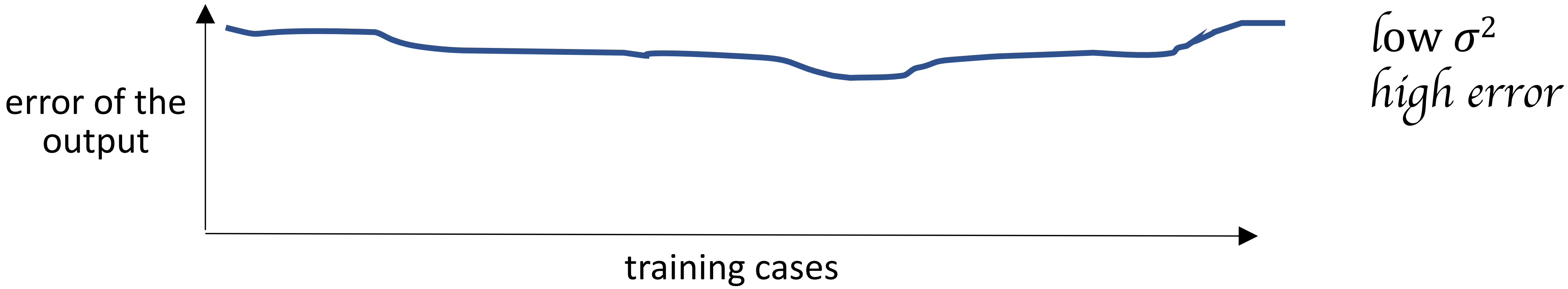
$$Fitness\text{-}function\ (\mathrm{I}) = \sigma^2(error)$$

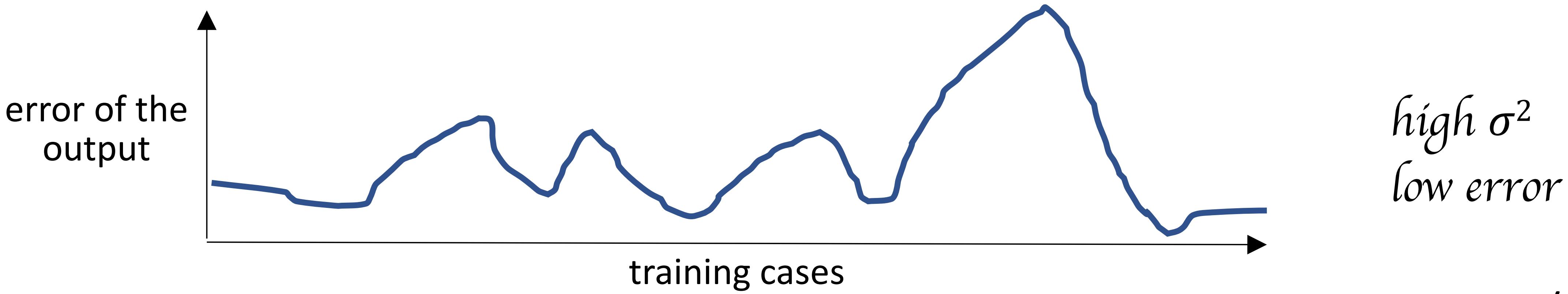
$$\mathrm{MSE} = \frac{\sum_{i=0}^{m} (y_i - \widehat{y_i})^2}{m} \leq \sigma^2(\hat{y})$$

due to linear scaling

$\sigma^2(\hat{y})$ is an upper bound on the MSE

Fitness Function





GP Runs

Tournament Selection

We reduce the disruptiveness of GP operators by:

- Avoiding Crossover
- Only Mutation (biased towards the leaves)

GP Runs

Tournament Selection

We reduce the disruptiveness of GP operators by:

Avoiding Crossover

Only Mutation (biased towards the leaves)

Without Crossover how SGP-DT exchanges genetic materials and functionalities?

SGP-DT: SGP Based on Dynamic Targets

external iterations	dynamic targets (dt)	GP runs	partial models (PM)	final model
1	$dt_1 = \hat{y}$	GP_1	$a_1 + b_1 \cdot \mathbf{PM_1}$	
2	$dt_1 - sem(a_1 + b_1 \cdot PM_1)$	GP_2	$a_2 + b_2 \cdot \mathbf{PM_2}$	$\sum_{i=1}^{n} a_i + bi \cdot PM_i$
....	....	....	....	
n	$dt_{n-1} - sem(a_{n-1} + b_{n-1} \cdot PM_{n-1})$	GP_n	$a_n + b_n \cdot \mathbf{PM_n}$	

Final Model

$$\sum_{i=1}^{1} a_i + bi \bullet PM_i$$

$a_1 + b_1 \bullet PM_1$ $a_2 + b_2 \bullet PM_2$ $a_3 + b_3 \bullet PM_3$ ..... $a_K + b_K \bullet PM_K$ ..... $a_n + b_n \bullet PM_n$

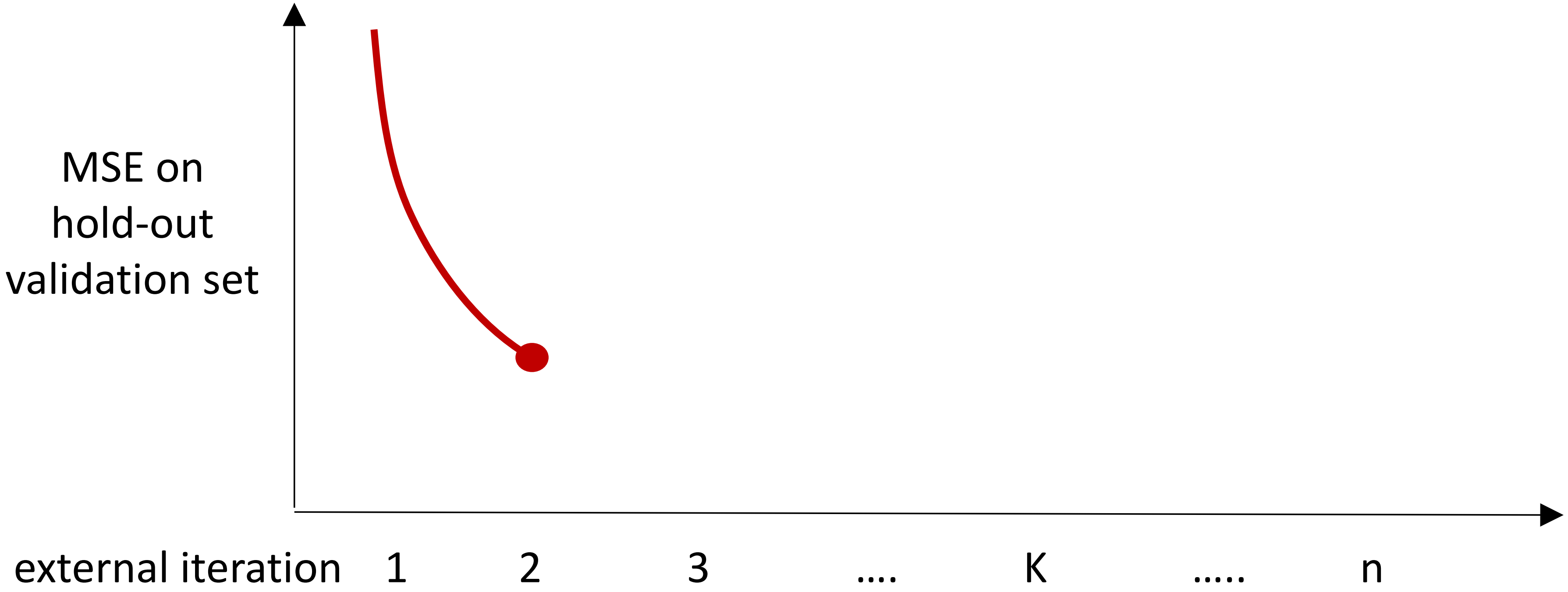
MSE on
hold-out
validation set

external iteration 1 2 3 .... K ..... n

Final Model

$$\sum_{i=1}^{2} a_i + bi \cdot PM_i$$

$a_1 + b_1 \cdot PM_1$ $a_2 + b_2 \cdot PM_2$ $a_3 + b_3 \cdot PM_3$ ..... $a_K + b_K \cdot PM_K$ ..... $a_n + b_n \cdot PM_n$

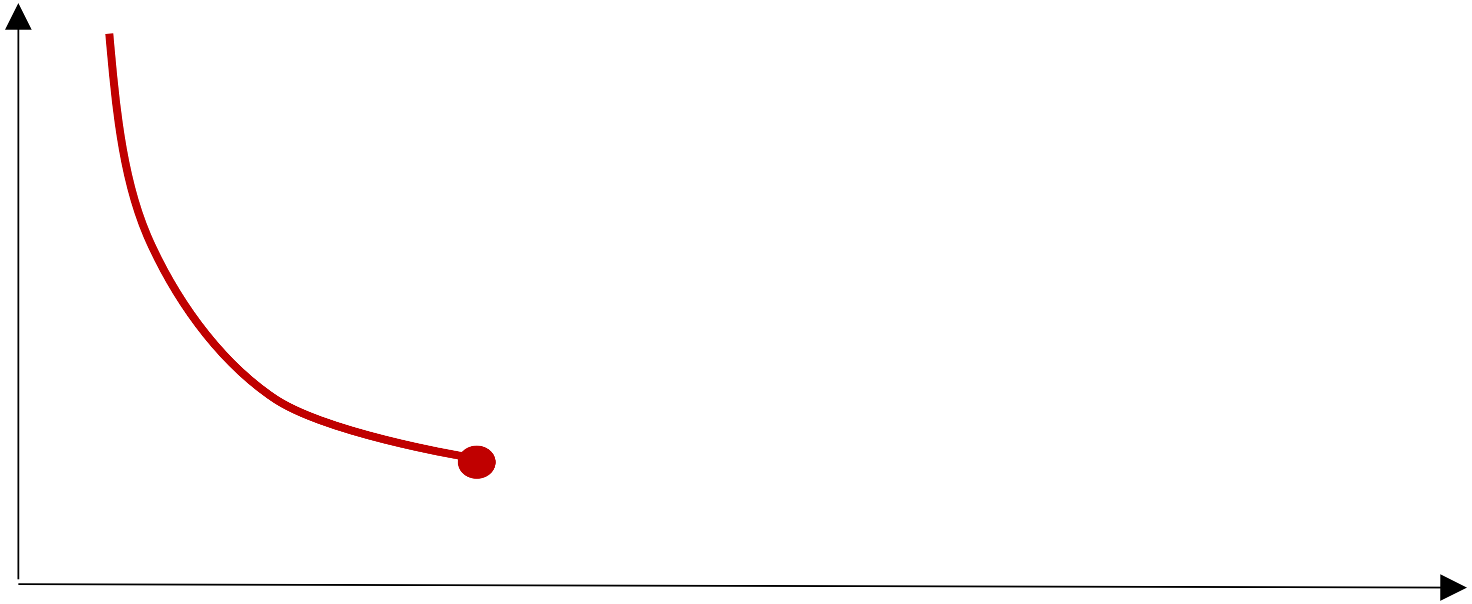


Final Model

$$\sum_{i=1}^{3} a_i + bi \cdot PM_i$$

$a_1 + b_1 \cdot PM_1$ $a_2 + b_2 \cdot PM_2$ $a_3 + b_3 \cdot PM_3$ ..... $a_K + b_K \cdot PM_K$ ..... $a_n + b_n \cdot PM_n$

MSE on hold-out validation set

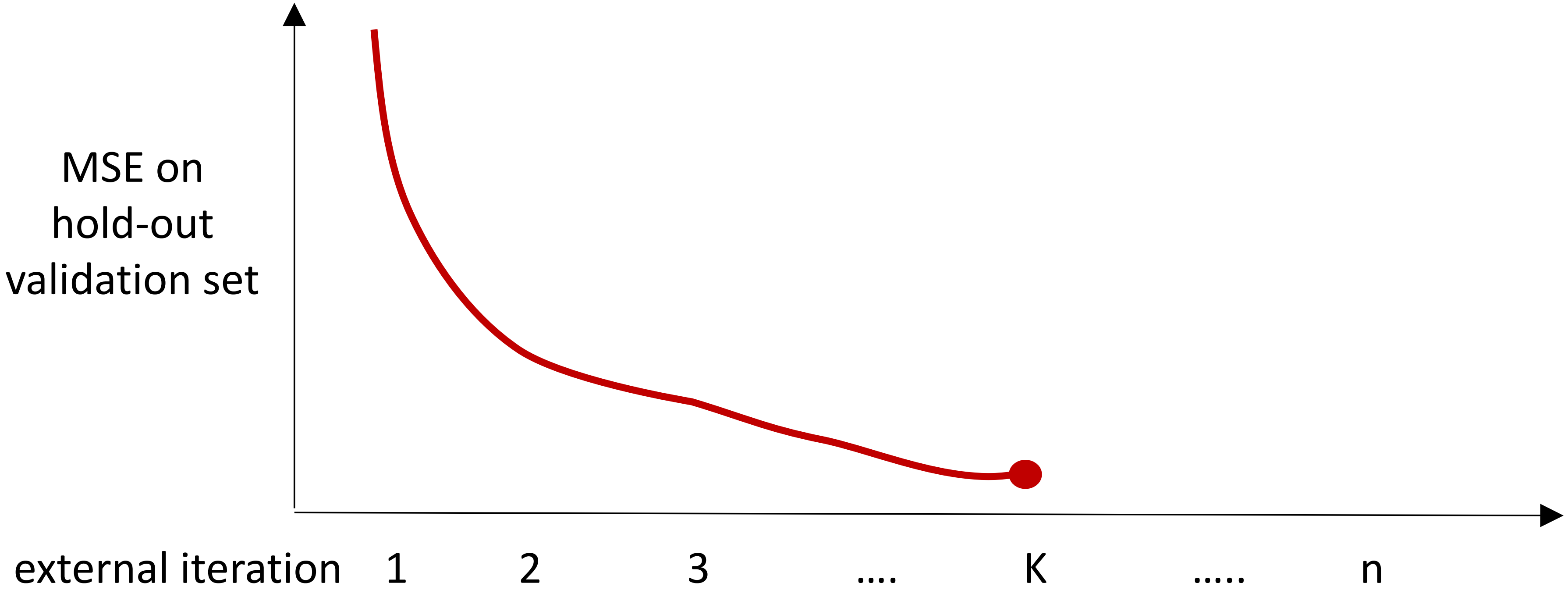


external iteration 1 2 3 .... K ..... n

Final Model

$$\sum_{i=1}^{K} a_i + bi \cdot PM_i$$

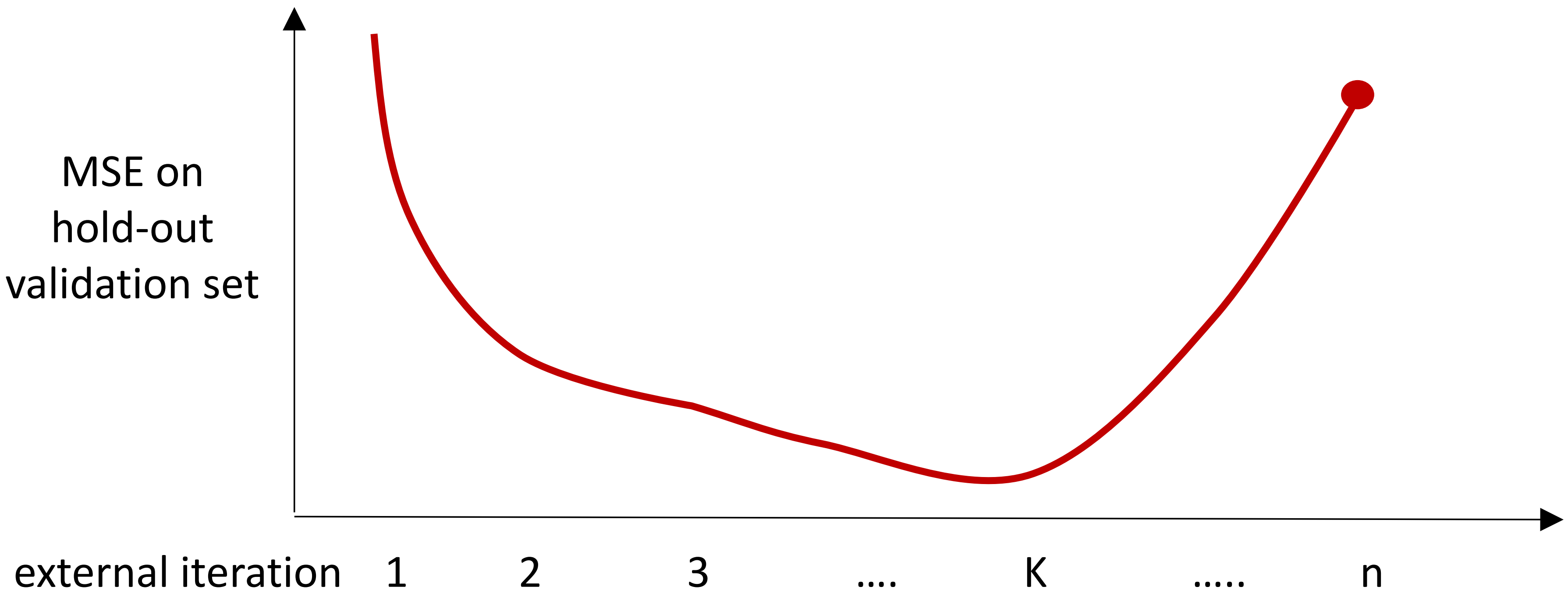
$a_1 + b_1 \cdot PM_1$ $a_2 + b_2 \cdot PM_2$ $a_3 + b_3 \cdot PM_3$ ..... $a_K + b_K \cdot PM_K$ ..... $a_n + b_n \cdot PM_n$



Final Model

$$\sum_{i=1}^{K} a_i + bi \cdot PM_i$$

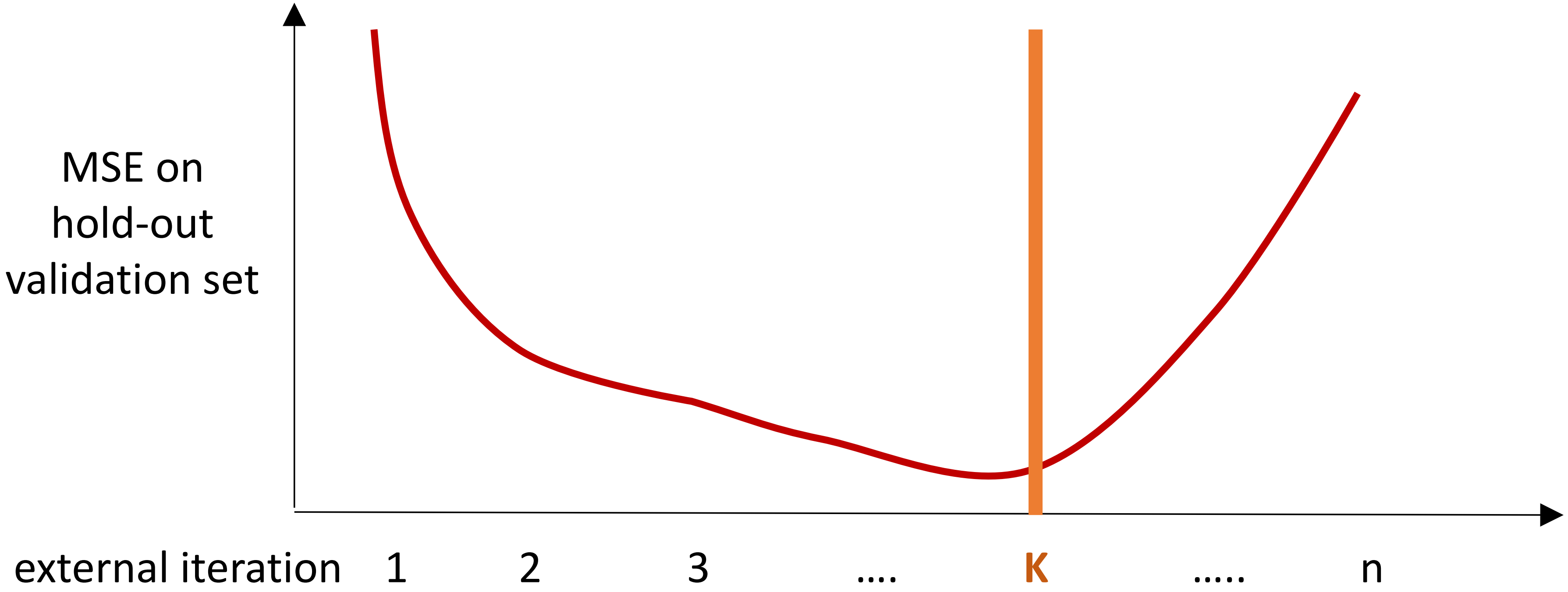
$a_1 + b_1 \cdot PM_1$ $\quad a_2 + b_2 \cdot PM_2$ $\quad a_3 + b_3 \cdot PM_3$ $\quad .....$ $\quad a_K + b_K \cdot PM_K$ $\quad .....$ $\quad a_n + b_n \cdot PM_n$



Final Model

$$\sum_{i=1}^{K} a_i + bi \cdot PM_i$$

$a_1 + b_1 \cdot PM_1$ $\quad$ $a_2 + b_2 \cdot PM_2$ $\quad$ $a_3 + b_3 \cdot PM_3$ $\quad$ ..... $\quad$ $a_K + b_K \cdot PM_K$ $\quad$ ..... $\quad$ $a_n + b_n \cdot PM_n$



Evaluation

Subjects

name	# attributes	# instances
airfoil	5	1,503
concrete	8	1,030
enc	8	768
enh	8	768
housing	14	506
tower	25	3,135
yatch	6	309
uball5d	5	6,024

Subjects

name	# attributes	# instances
airfoil	5	1,503
concrete	8	1,030
enc	8	768
enh	8	768
housing	14	506
tower	25	3,135
yatch	6	309
uball5d	5	6,024

50 times

training set (75%) | test set (15%)

Subjects

name	# attributes	# instances
airfoil	5	1,503
concrete	8	1,030
enc	8	768
enh	8	768
housing	14	506
tower	25	3,135
yatch	6	309
uball5d	5	6,024

UCI Machine Learning Repository

50 times

7.5%

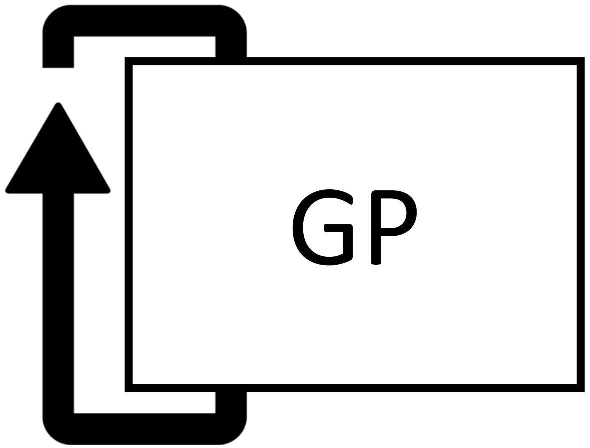
val set | training set (67.5%) | test set (15%)

SGP-DT Evaluation Setup

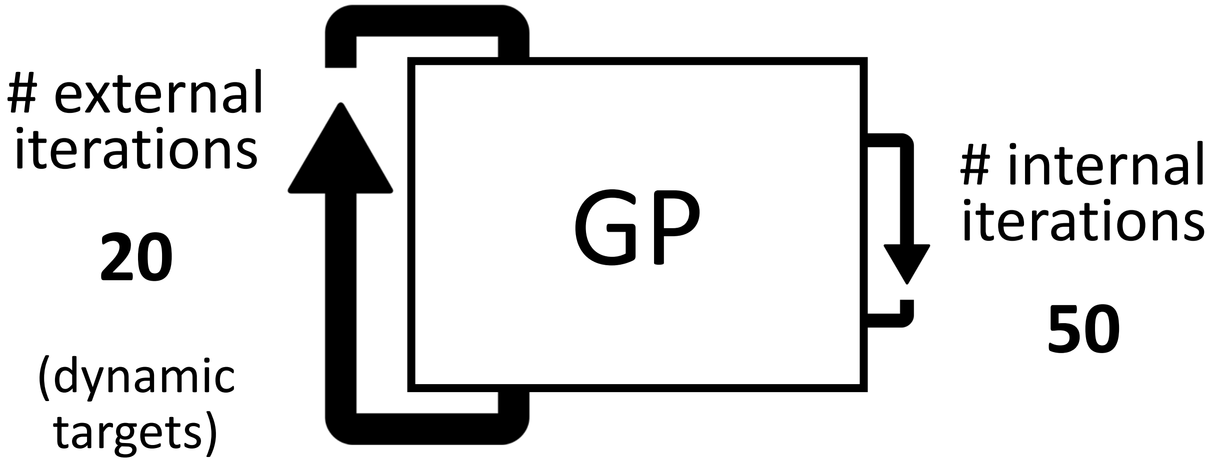
external iterations

20

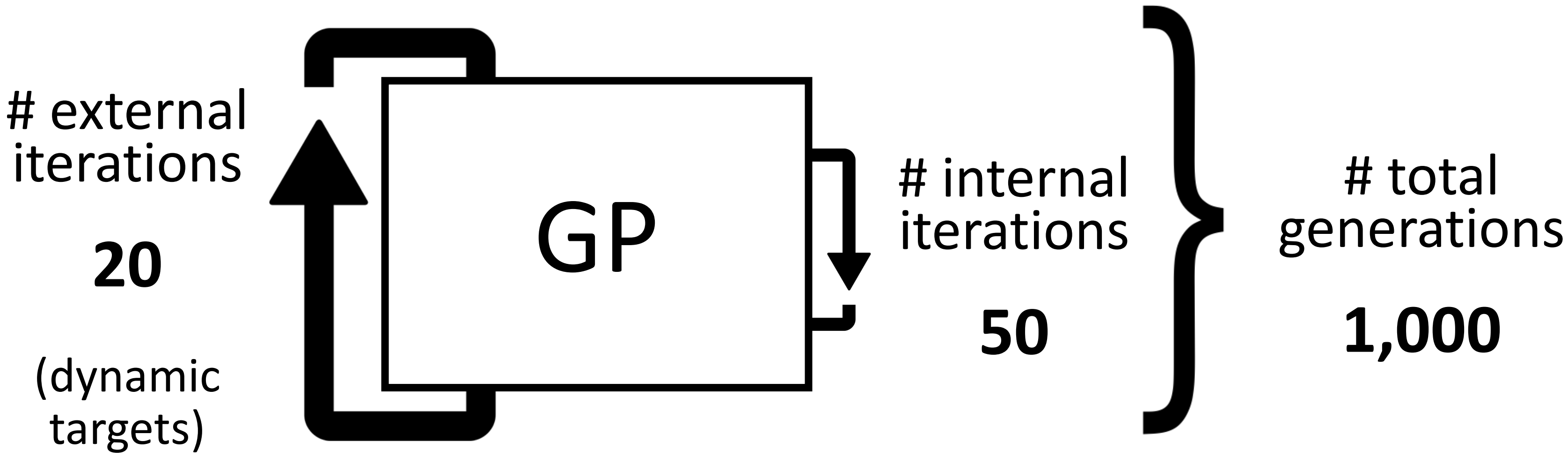
(dynamic targets)



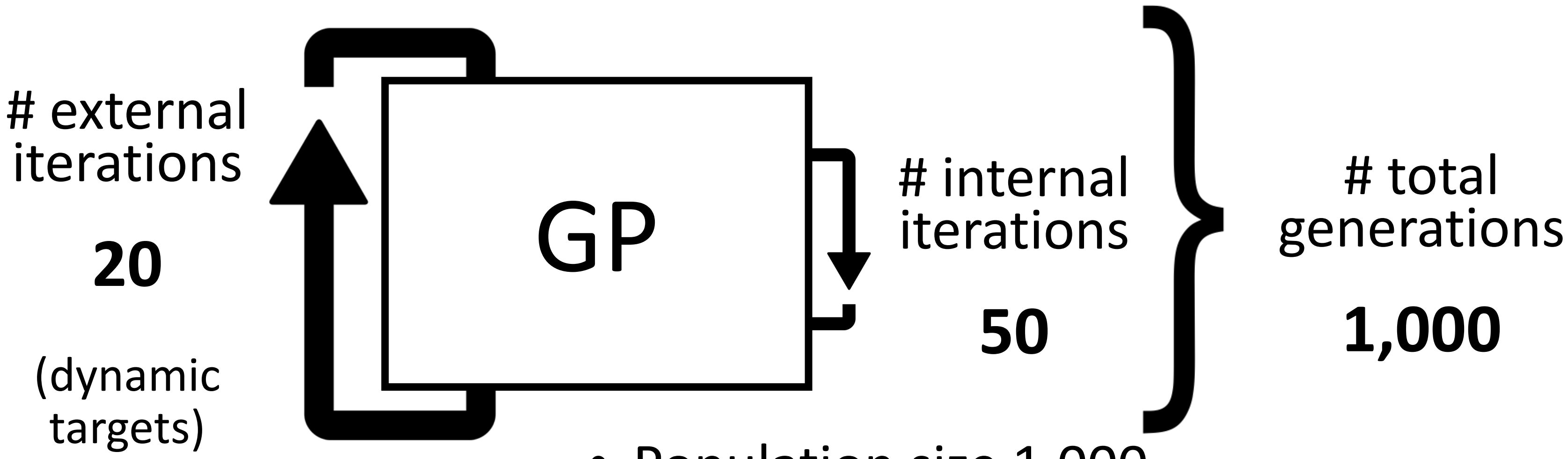
SGP-DT Evaluation Setup



SGP-DT Evaluation Setup



SGP-DT Evaluation Setup



- Population size 1,000
- Probability mutation 100%
- Size elitism 1
- No bound on the size of the individuals

Results (median RMSE 50 trials)

	Median Root Mean Square Error (RMSE)
subject	**SGP-DT**
airfoil	2.46
concrete	6.51
enc	1.48
enh	.56
housing	4.47
tower	.26
yatch	1.02
uball5d	.04

Competitors

Lasso (Efron 2004)

Least square regression method

ε-lexicase (La Cava 2016)

Evolutionary technique based on lexicase (Starosta 70s)

Outperforms many GP algorithms

We ran 1,000 generations (same as SGP-DT)

Validation set (10% of the training set)

Results (median RMSE 50 trials)

	Median Root Mean Square Error (RMSE)		
subject	**SGP-DT**	**Lasso**	**ε-lexicase**
airfoil	2.46	4.85	3.65
concrete	6.51	10.54	7.07
enc	1.48	3.25	1.86
enh	.56	2.96	1.29
housing	4.47	4.91	4.28
tower	.26	.29	.30
yatch	1.02	9.02	1.36
uball5d	.04	.19	.06

Results (median RMSE 50 trials)

	Median Root Mean Square Error (RMSE)		
subject	**SGP-DT**	**Lasso**	**ε-lexicase**
airfoil	2.46	4.85	3.65
concrete	6.51	10.54	7.07
enc	1.48	3.25	1.86
enh	.56	2.96	1.29
housing	4.47	4.91	4.28
tower	.26	.29	.30
yatch	1.02	9.02	1.36
uball5d	.04	.19	.06

Results (median RMSE 50 trials)

	Median Root Mean Square Error (RMSE)		
subject	**SGP-DT**	**Lasso**	**ε-lexicase**
airfoil	2.46	+49%	+32%
concrete	6.51	+38%	+8%
enc	1.48	+54%	+20%
enh	.56	+81%	+57%
housing	4.47	+9%	-4%
tower	.26	+12%	+12%
yatch	1.02	+89%	+25%
uball5d	.04	+79%	+35%

$$\frac{\text{Lasso - SGP-DT}}{\text{Lasso}}100 \qquad \frac{\varepsilon\text{-lexicase - SGP-DT}}{\varepsilon\text{-lexicase}}100$$

SGP-DT Variants

DT-EM Directly minimize the error

$$\mathcal{Fitness\text{–}function}\,(\mathrm{I}) = \mathrm{MSE} = \frac{\sum_{i=0}^{m} (y_i - \widehat{y_i})^2}{m}$$

 # training cases

DT-NM without Min and Max

+, -, x, /

ERC [-1; +1]

~~Min Max~~

Results (median RMSE 50 trials)

	Median Root Mean Square Error (RMSE)		
subject	**SGP-DT**	**DT-EM**	**DT-NM**
airfoil	2.46	+4%	+16%
concrete	6.51	-1%	-2%
enc	1.48	+1%	-2%
enh	.56	+3%	-3%
housing	4.47	-1%	+1%
tower	.26	+10%	+10%
yatch	1.02	+20%	+13%
uball5d	.04	+7%	-8%

$$\frac{\text{DT-EM} - \text{SGP-DT}}{\text{DT-EM}}100 \qquad \frac{\text{DT-NM} - \text{SGP-DT}}{\text{DT-NM}}100$$

Results (number of evaluated nodes 50 trials)

	Median Number of Evaluated Nodes
subject	**SGP-DT**
airfoil	1.00E+10
concrete	1.14E+10
enc	1.18E+10
enh	1.18E+10
housing	7.70E+09
tower	7.21E+10
yatch	4.62E+09
uball5d	9.83E+10

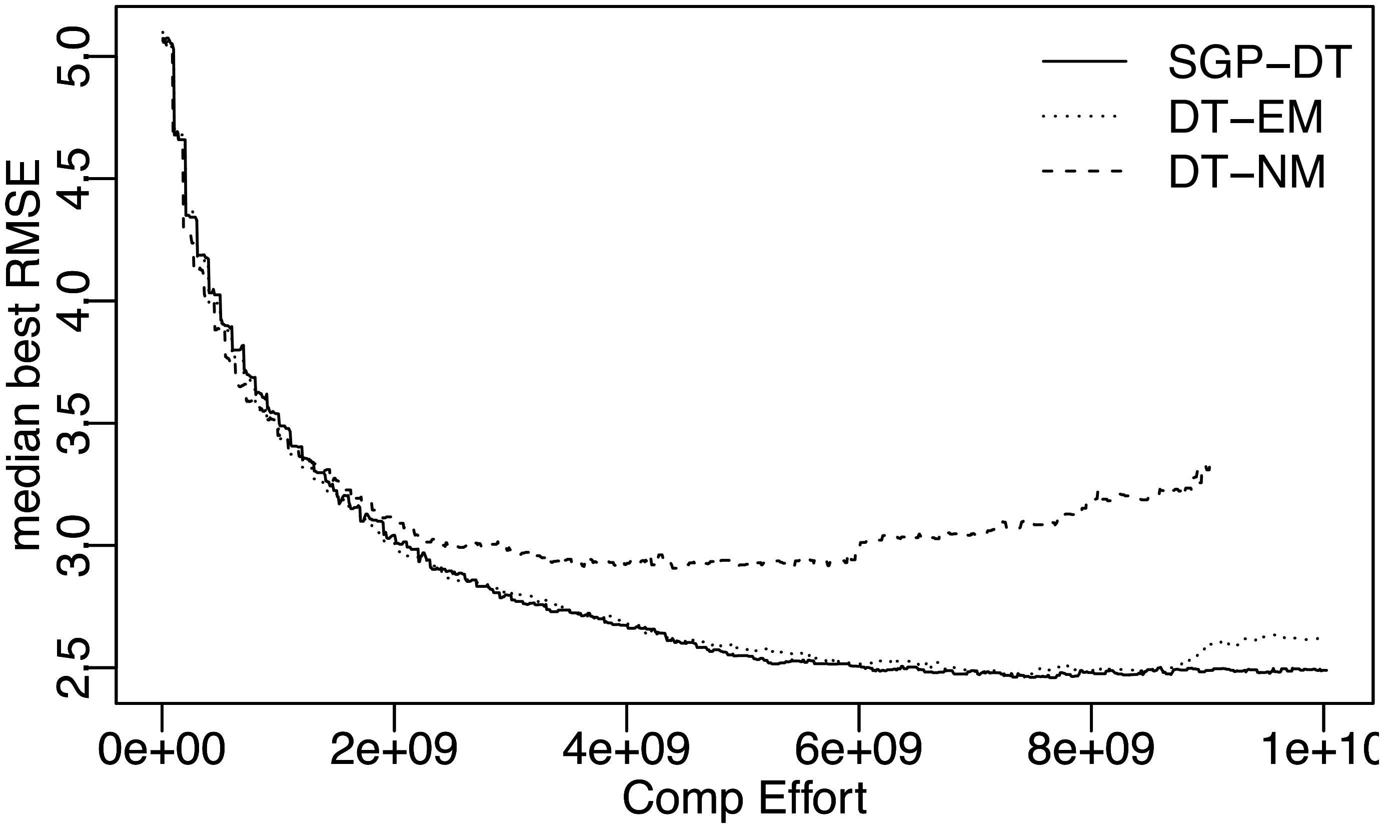
Results (number of evaluated nodes 50 trials)

	Median Number of Evaluated Nodes			
subject	**SGP-DT**	**ε-lexicase**	**DT-EM**	**DT-NM**
airfoil	1.00E+10	9.26×	1.00x	.90x
concrete	1.14E+10	5.64×	1.00x	.77x
enc	1.18E+10	4.25x	.99x	.80x
enh	1.18E+10	4.30x	.99x	.78x
housing	7.70E+09	4.02x	.99x	.78x
tower	7.21E+10	2.69x	.99x	.62x
yatch	4.62E+09	4.34x	.99x	.75x
uball5d	9.83E+10	4.01x	.99x	.76x

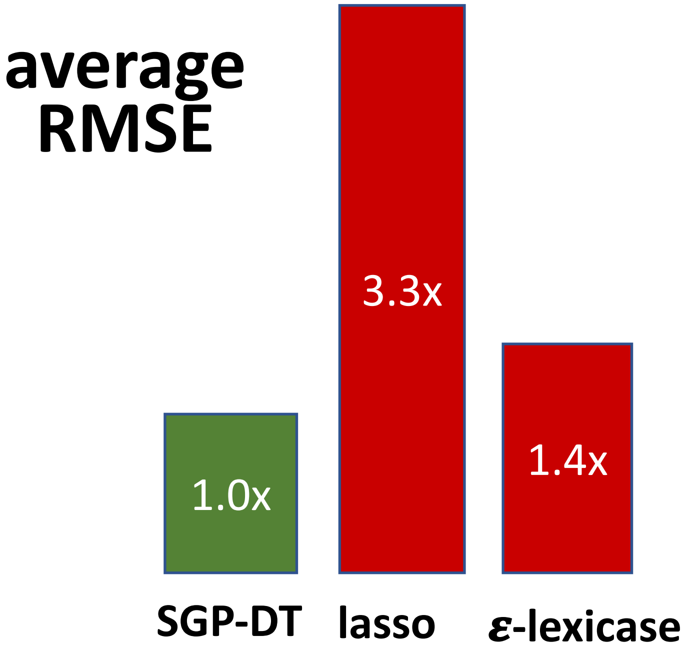
$\frac{\varepsilon\text{-lexicase}}{\text{SGP-DT}}$ $\frac{\text{DT-EM}}{\text{SGP-DT}}$ $\frac{\text{DT-NM}}{\text{SGP-DT}}$

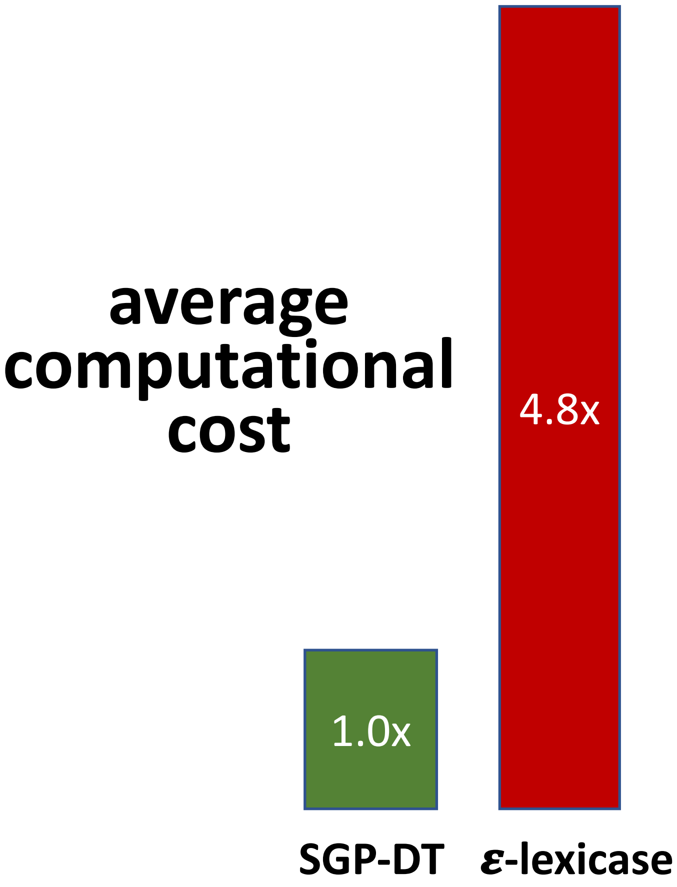
Overfitting

airfoil dataset

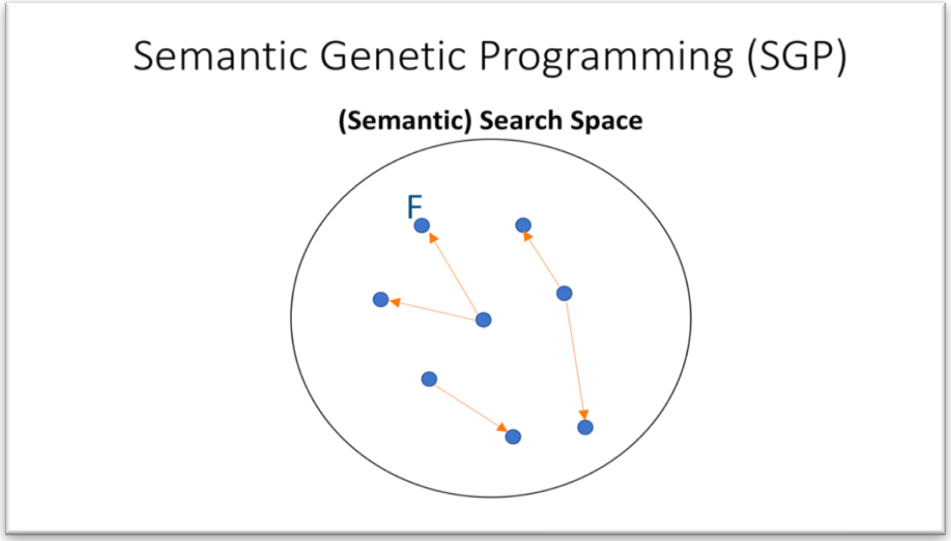


Results Summary

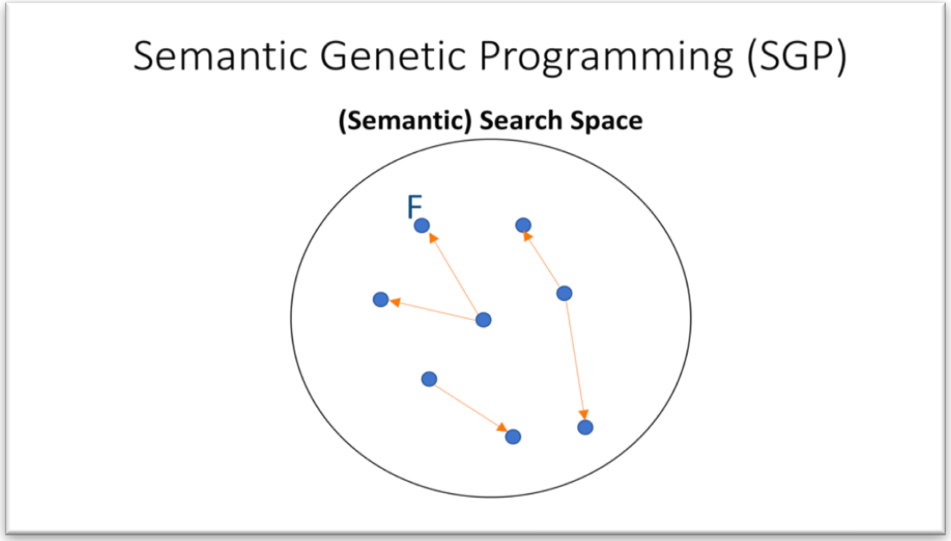


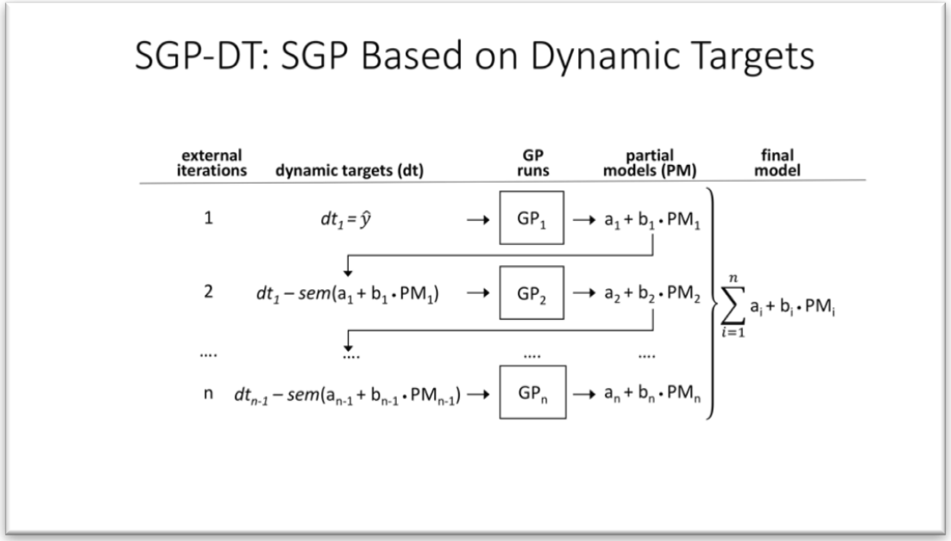


Conclusion



Conclusion

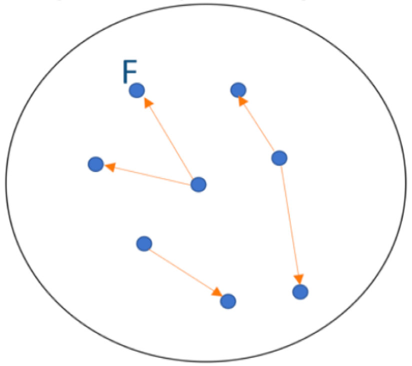

Semantic Genetic Programming (SGP)
(Semantic) Search Space
F


SGP-DT: SGP Based on Dynamic Targets
external iterations
dynamic targets (dt)
GP runs
partial models (PM)
final model
1
$dt_1 = \hat{y}$
GP_1
$a_1 + b_1 \cdot PM_1$
2
$dt_1 - sem(a_1 + b_1 \cdot PM_1)$
GP_2
$a_2 + b_2 \cdot PM_2$
....
n
$dt_{n-1} - sem(a_{n-1} + b_{n-1} \cdot PM_{n-1})$
GP_n
$a_n + b_n \cdot PM_n$
$\sum_{i=1}^{n} a_i + b_i \cdot PM_i$

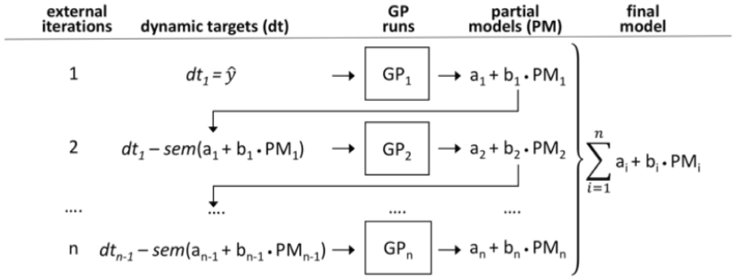
Conclusion

Semantic Genetic Programming (SGP)

(Semantic) Search Space



SGP-DT: SGP Based on Dynamic Targets



Subjects

name	# attributes	# instances
airfoil	5	1,503
concrete	8	1,030
enc	8	768
enh	8	768
housing	14	506
tower	25	3,135
yacth	6	309
uball5d	5	6,024

UCI Machine Learning Repository

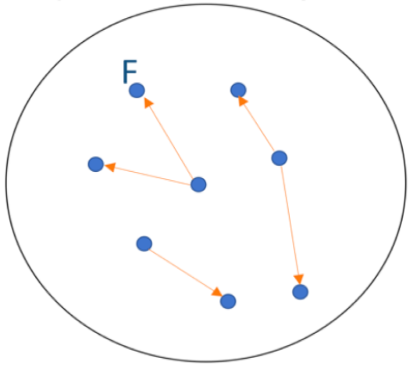
50 times

training set (75%) | test set (15%)

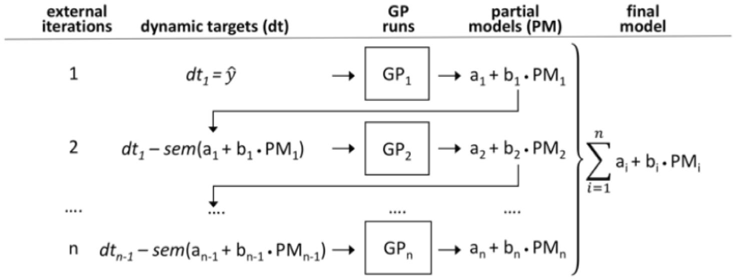
Conclusion

Semantic Genetic Programming (SGP)

(Semantic) Search Space



SGP-DT: SGP Based on Dynamic Targets



Subjects

name	# attributes	# instances
airfoil	5	1,503
concrete	8	1,030
enc	8	768
enh	8	768
housing	14	506
tower	25	3,135
yacth	6	309
uball5d	5	6,024

UCI Machine Learning Repository

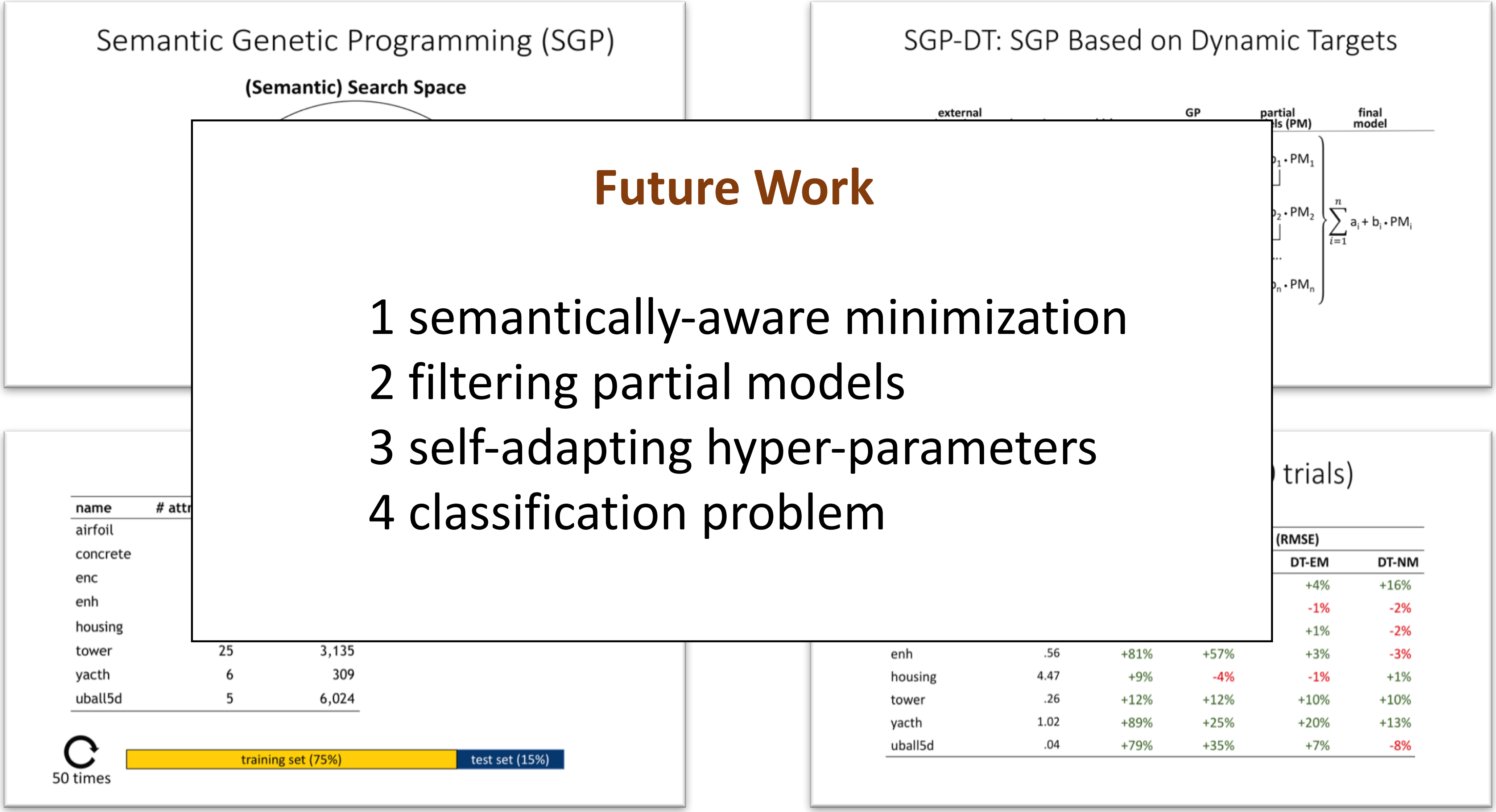
50 times

training set (75%) | test set (15%)

Results (median RMSE 50 trials)

	Median Root Mean Square Error (RMSE)				
subject	**SGP-DT**	**Lasso**	**ε-lexicase**	**DT-EM**	**DT-NM**
airfoil	2.46	+49%	+32%	+4%	+16%
concrete	6.51	+38%	+8%	-1%	-2%
enc	1.48	+54%	+20%	+1%	-2%
enh	.56	+81%	+57%	+3%	-3%
housing	4.47	+9%	-4%	-1%	+1%
tower	.26	+12%	+12%	+10%	+10%
yacth	1.02	+89%	+25%	+20%	+13%
uball5d	.04	+79%	+35%	+7%	-8%

Conclusion



Future Work

1 semantically-aware minimization
2 filtering partial models
3 self-adapting hyper-parameters
4 classification problem

Results (median RMSE)

	Median Root Mean Square Error (RMSE)				
subject	**SGP-DT**	**Lasso**	**ε-lexicase**	**DT-EM**	**DT-NM**
airfoil	2.4634	4.8484	3.6505	2.5643	2.9237
concrete	6.5123	10.5383	7.0707	6.4476	6.4132
enc	1.4838	3.2498	1.8647	1.4993	1.4584
enh	0.5560	2.9645	1.2952	0.5714	0.5410
housing	4.4700	4.9155	4.2785	4.4377	4.5273
tower	0.2606	0.2953	0.2975	0.2900	0.2900
yacth	1.0221	9.0237	1.3577	1.2849	0.0372
uball5d	0.0402	0.1939	0.0618	0.0430	1.1786