ϵ-Lexicase selection: a probabilistic and multi-objective analysis of lexicase selection and with continuous valued problems

William La Cava, Lee Spector, Jason Moore

December 2016

Abstract

Lexicase selection is a parent selection method that considers test cases separately, rather than in aggregate, when performing parent selection. As opposed to previous work that has demonstrated the ability of lexicase selection to solve difficult problems, the goal of this paper is to develop the theoretical underpinnings that explain its performance. To this end, we derive an analytical formula that gives the expected probabilities of selection under lexicase selection, given a population and its behavior. In addition, we expand upon the relation of lexicase selection to many-objective optimization methods to show the effect of lexicase, which is to select individuals on the boundaries of Pareto fronts in high-dimensional space. We show analytically why lexicase selection performs more poorly for certain sizes of population and training cases, and why it has been shown to perform more poorly in continuous error spaces. To addres this last concern, we introduce ϵ-lexicase selection, which modifies the pass condition defined in lexicase selection to allow near-elite individuals to pass cases, thereby improving selection performance. We show that ϵ-lexicase outperforms several diversity-maintenance strategies for regression problems.

1 INTRODUCTION

2 Lexicase Selection

Lexicase selection is a parent selection technique based on lexicographic ordering of test (i.e. fitness) cases. Each parent selection event proceeds as follows:

1. The entire population is added to the selection pool.
2. The fitness cases are shuffled.
3. Individuals in the pool with a fitness worse than the best fitness on this case among the pool are removed.

4. If more than one individual remains in the pool, the first case is removed and 3 is repeated with the next case. If only one individual remains, it is the chosen parent. If no more fitness cases are left, a parent is chosen randomly from the remaining individuals.

The lexicase selection algorithm for a single selection event is presented below:

Lexicase Selection

`GetLexicaseParent`$(\mathcal{N}, \mathcal{T})$:	
$T' \leftarrow$ `shuffle`$(\mathcal{T})$	training cases
$S \leftarrow \mathcal{N}$	initial selection pool is the population
while $\lvert T' \rvert > 0$ and $\lvert \mathcal{S} \rvert > 1$:	main loop
`case` $\leftarrow$ random choice from $\mathcal{T}'$	choose a random case
`elite` $\leftarrow$ best fitness in $\mathcal{S}$ on `case`	determine elite fitness
$\mathcal{S} \leftarrow n \in \mathcal{S}$ if fitness$(n) =$ `elite`	reduce selection pool to elites
$\mathcal{T}' \leftarrow \mathcal{T}' -$ `case`	remove top case
return random choice from $\mathcal{S}$	return parent

3 ϵ-Lexicase Selection

ϵ-Lexicase Selection

`Get_`ϵ`-Lexicase_Parent`$(\mathcal{N}, \mathcal{T})$:	
$T' \leftarrow$ `shuffle`$(\mathcal{T})$	training cases
$S \leftarrow \mathcal{N}$	initial selection pool is the population
$\epsilon \leftarrow MAD(\text{fitness}(\mathcal{N}))$ for $t \in \mathcal{T}$	get ϵ for each case
while $\lvert T' \rvert > 0$ and $\lvert \mathcal{S} \rvert > 1$:	main loop
`case` $\leftarrow T'[0]$	consider the top case
`elite` $\leftarrow$ best fitness in $\mathcal{S}$ on `case`	determine elite fitness
$\mathcal{S} \leftarrow n \in \mathcal{S}$ if fitness$(n) <=$ `elite`$+\epsilon_{\texttt{case}}$	reduce selection pool to elites
$\mathcal{T}' \leftarrow \mathcal{T}' -$ `case`	remove top case
return random choice from $\mathcal{S}$	return parent

3.1 Expected Probabilities of Selection

What is the probability of an individual being selected, given its performance in a given population on a set of training cases?
$\mathcal{N} = \{n_i\}_{i=1}^{N}$: population
$\mathcal{T} = \{t_i\}_{i=1}^{T}$: training cases
$\mathcal{K}_n = \{k_i\}_{i=1}^{K} \subseteq \mathcal{T}$: training cases from $\mathcal{T}$ for which individual n is elite

To put it in words, the probability of n being selected is the probability that a case n passes ($t \in \mathcal{K}_n$) is selected and:

1. no more cases remain and n is selected among the set of individuals that pass the selected case; or
2. n is the only individual that passes the case; or
3. n is selected via the selection of another case that n passes (repeating the process).

Table 1: Example population from original lexicase paper (Spector 2013).

Program	Test				$\mathcal{K}(\mathcal{T})$	MAE	P_{sel}	P_t
	t_1	t_2	t_3	t_4				
n_1	2	2	4	2	$\{t_2, t_4\}$	2.5	0.25	0.28
n_2	1	2	4	3	$\{t_2\}$	2.5	0.00	0.28
n_3	2	2	3	4	$\{t_2, t_3\}$	2.75	0.33	0.12
n_4	0	2	5	5	$\{t_1, t_2\}$	3.0	0.208	0.04
n_5	0	3	5	2	$\{t_1, t_4\}$	2.5	0.208	0.28

Formally, let $P_{sel}(n|\mathcal{N}, \mathcal{T}, \mathcal{K}_n)$ be the probability of n being selected in a population $\mathcal{N}$ with training cases $\mathcal{T}$. Let $\mathcal{K}_n(\mathcal{T})$ be the subset of cases in $\mathcal{T}$ for which n is elite. Then lexicase probability can be represented as a piecewise recursive function:

$$P_{sel}(n|\mathcal{N}, \mathcal{T}) = \tag{1}$$
$$\begin{cases} 1 & : \quad |\mathcal{T}| > 0, |\mathcal{N}| = 1; \\ 1/|\mathcal{N}| & : \quad |\mathcal{T}| = 0; \\ \frac{1}{|\mathcal{T}|} \sum_{k_s \in K_n(\mathcal{T})} P_{sel}\left(n|\mathcal{N}(m|k_s \in K_m(\mathcal{T})), \mathcal{T}(t|t \neq k_s)\right) & : \quad \text{otherwise} \end{cases}$$

The first two elements of P_{sel} follow from the lexicase algorithm: if there is one individual in $\mathcal{N}$, then it is selected; if there no more cases in in $\mathcal{T}$, then n has a probability of selection split among the individuals in $\mathcal{N}$, i.e., $1/|\mathcal{N}|$. If neither of these conditions are met, the remaining probability of selection is $1/|\mathcal{T}|$ times the summation of P_{sel} over n's elite cases. Each case for which n is elite (represented by $k_s \in K_n(\mathcal{T})$) has a probability of $1/|\mathcal{T}|$ of being selected. For each potential selection k_s, the probability of n being selected as a result of this case being chosen is dependent on the number of individuals that are also elite on these cases, represented by $\mathcal{N}(m|k_s \in K_m(\mathcal{T}))$, and the cases that are left to be chosen, represented by $\mathcal{T}(t|t \neq k_s))$.

Example As an example of calculating absolute probabilities, we consider the illustrative problem from the original lexicase selection paper [**?**], shown in Table 1. Using Eqn. 1, the probabilities for each individual can be calculated as follows:

$$\begin{aligned} P_{sel}(n_1) &= 1/4 * (1/3 * (1) + 1/3 * (1 + 1)) = 0.25 \\ P_{sel}(n_2) &= 1/4 * (0) = 0 \\ P_{sel}(n_3) &= 1/4 * (1/3 * (1 + 1/2 * (1)) + 1/3 * (1)) = 0.20833 \\ P_{sel}(n_4) &= 1/4 * (1/3 * (1/2 * (1) + 1) + 1/3 * (1)) = 0.20833 \end{aligned}$$

We compare the probability of selection under lexicase selection to that using tournament selection with an indentical population and fitness structure. To do so we must first formulate the probability of selection for an individual

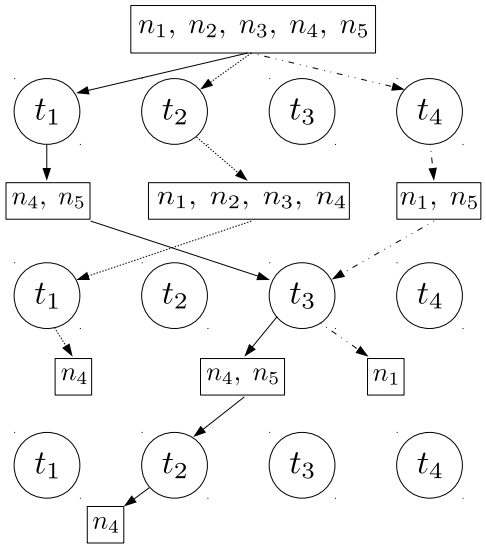

Figure 1: A graphical representation of 3 example parent selections using lexicase selection on the population in Table 1. The bold, dashed and dot-dashed lines indicate different selection paths through the test cases in circles. The boxes indicate the selection pool at each step in the process.

undergoing tournament selection with r-size tournaments. Consider that the mean absolute error is used to aggregate the fitness cases. Then the fitness ranks of $\mathcal{N}$ can be calculated, with lower rank indicating better fitness. Let S_i be the individuals in $\mathcal{N}$ with a fitness rank of i, and let Q be the number of unique fitness ranks. Then Xie et. al. (cite) showed that the probability of selecting an individual with rank j in a single tournament is

$$P_t = \frac{1}{|S_j|}\left(\left(\frac{\sum_{i=j}^{Q}|S_i|}{N}\right)^r - \left(\frac{\sum_{i=j+1}^{Q}|S_i|}{N}\right)^r\right) \tag{2}$$

Complexity Can we analytically calculate the probability of selection an individual with less complexity than executing the lexicase selection algorithm? It appears not. Eqn. 1 has a worst-case complexity of $O(T^N)$ when all individuals are elite on $\mathcal{T}$, which discourages its use as a selection method. The lexicase selection algorithm samples the expected probabilities of each individual by recursing on random orders of cases in $\mathcal{T}$, considering one at a time rather than branching to consider all combinations of cases that could result in selection for each individual in question. This gives lexicase selection a complexity of $O(TN)$ for selecting a single parent, and therefore a complexity of $O(TN^2)$ per generation.

3.2 Effect of N and T

What does the an analysis of the probability of selection for lexicase selection tell us about how lexicase behaves for cases in which $|\mathcal{N}| << |\mathcal{T}|$? $|\mathcal{T}| << |\mathcal{N}|$?

The sampling of P_{sel} done by lexicase is tied to the population size because lexicase selection conducts N depth-first searches of the case orderings to choose N parents. This implies that the value of N determines the fidelity with which P_{sel} is approximated via the sampling. Smaller populations will therefore produce poorer approximations of P_{sel}.

The effectiveness of lexicase selection has also been tied to the number of fitness cases, T. When T is small, there are very few ways in which individuals can be selected. For example, if $T = 2$, an individual must be elite on one of these two cases to be selected. For continuous errors in which few individuals are elite, this means that only two individuals are likely to produce all of the children for the subsequent generation.

3.3 Effect of different population structures

1. compare probability of lexicase selection to probability of selection with tournament / roulette
2. compare population structures: maintain correlation structure and vary population size/ number of test cases
3. see how many iterations of lexicase selection are required to converge on the probabilities of selection for the example problem in Table 1
4. population where there is one individual that sucks at everything but is good at other things - how does it compare to tournament selection probabilities?
5. do not assume that N rounds of lexicase selection are conducted!

4 Multi-objective Interpretation of Lexicase Selection

Lexicase selection is guarantees the return of individuals that are on the Pareto front with respect to the fitness cases. This is a necessary but not sufficient condition. In fact, lexicase selection only selects those individuals in the "corners" of the Pareto front, meaning they are on the front *and* elite, globally, with respect to at least one fitness case. Put another way, no individual can be selected via lexicase selection unless it is elite with respect to at least one objective among the entire population, regardless of its performance on other objectives.

Interestingly, the worst-case complexity of NSGA-II is the best-case complexity for lexicase selection. Add notes from discourse

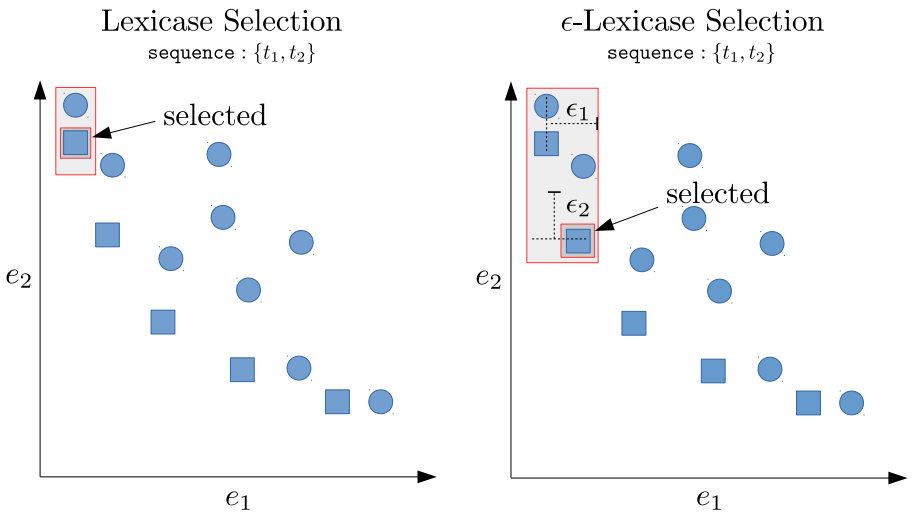

Figure 2: An illstration of the performance of lexicase selection and ϵ-lexicase selection in a scenario involving two cases. Each point represents and individual in the population. The squares are individuals on the Pareto front. In each case, the selection ordering shown is $\{t_1, t_2\}$.

Farina and Amato [1]:

> the Pareto definition of optimality in a multi-criteria decision making problem can be unsastisfactory due to essentially two reasons: the number of improved or equal objective values is not taken into account, the (normalized) size of improvements is not taken into account

Here we define Pareto dominance relations with respect to the training cases.

Definition: n_1 dominates n_2, i.e., $n_1 \prec n_2$, if $e_j(n_1) \leq e_j(n_2)\ \forall j \in \{1, \ldots, N\}$ and $\exists j \in \{1, \ldots, N\}$ for which $e_j(n_1) < e_j(n_2)$.

Theorem 1: Individuals selected by lexicase selection are non-dominated in $\mathcal{N}$ with respect to the training cases $\mathcal{T}$.

Proof: Let $n_1, n_2 \in \mathcal{N}$ be individuals in a population selected by lexicase selection. Suppose $n_1 \prec n_2$. Then $e_j(n_1) \leq e_j(n_2)\ \forall j \in \{1, \ldots, N\}$ and $\exists j \in \{1, \ldots, N\}$ for which $e_j(n_1) < e_j(n_2)$. Therefore n_1 is selected for every case that n_2 is selected, and $\exists t \in \mathcal{T}$ for which n_2 is removed from selection due to n_1. Therefore n_2 cannot be selected by lexicase selection, the supposition is false, and the theorem is true.

Table 2: Symbolic regression problem settings.

Setting	Value
Population size	1000
Crossover / mutation	80/20%
Program length limits	[3, 50]
ERC range	[-1,1]
Generation limit	1000
Trials	30
Terminal Set	{$\mathbf{x}$, ERC, $+$, $-$, $*$, /, sin, cos, exp, log}
Elitism	keep best

We can extend Theorem 1 to ϵ-lexicase selection for conditions in which ϵ is pre-defined for each fitness case, i.e. in static and dynamic cases, but not when ϵ is recalculated for each selection pool. The theorem is extended by defining the dominance relation with respect to ϵ, as follows:

Definition: n_1 ϵ-dominates n_2, i.e., $n_1 + \epsilon_j \prec_\epsilon n_2$, if $e_j(n_1) \leq e_j(n_2)$ $\forall j \in \{1, \ldots, N\}$ and $\exists j \in \{1, \ldots, N\}$ for which $e_j(n_1) + \epsilon_j < e_j(n_2)$.

In this case, the analagous condition holds for individuals selected with ϵ-lexicase selection.

Theorem 2: Individuals selected by ϵ-lexicase selection are non-ϵ-dominated in $\mathcal{N}$ with respect to the training cases $\mathcal{T}$.

Proof: Let $n_1, n_2 \in \mathcal{N}$ be individuals in a population selected by ϵ-lexicase selection. Suppose $n_1 \prec_\epsilon n_2$. Then $e_j(n_1) \leq e_j(n_2) + \epsilon_j$ $\forall j \in \{1, \ldots, N\}$ and $\exists j \in \{1, \ldots, N\}$ for which $e_j(n_1) < e_j(n_2) + \epsilon_j$. Therefore n_1 is selected for every case that n_2 is selected, and $\exists t \in \mathcal{T}$ for which n_2 is removed from selection due to n_1. Therefore n_2 cannot be selected by lexicase selection, the supposition is false, and the theorem is true.

Table 3: Problems used for method comparisons.

Regression			
Problem	Dimension	Training Cases	Test Cases
Housing	14	354	152
Tower	25	2195	940
Wind	6	4200	1800
ENH	8	538	230
ENC	8	538	230
UBall5D	5	1024	5000

Dynamical Systems				
Problem	Equations	Initial Conditions	Training Cases	Test Cases

Program Synthesis				
Problem	Input	Output	Training Cases	Test Cases
Number IO	integer in [-100,100], float in [-100.0, 100.0]	printed float	25	1000
Wallis PI	integer in [1, 200]	float	150	50
Vector Average	vector of float of length [1,50] with each float in [-1000.0, 1000.0]	float	100	1000

5 Related Work

6 Experimental Analysis

6.1 Effect of population structure on probabilities of selection

6.2 Results

7 Discussion

8 Conclusions

9 Acknowledgments

The authors would like to thank Thomas Helmuth, Nic McPhee and Bill Tozier for their feedback as well as members of the Computational Intelligence Laboratory at Hampshire College. This work is partially supported by NSF Grant Nos. 1068864, 1129139 and 1331283. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant number ACI-1053575 [2].

References

[1] M. Farina and P. Amato. On the optimal solution definition for many-criteria optimization problems. In *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*, pages 233–238. IEEE, 2002.

[2] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkens-Diehr. XSEDE: Accelerating Scientific Discovery. *Computing in Science and Engineering*, 16(5):62–74, 2014.