Structural versus Evaluation Based Solutions Similarity in Genetic Programming Based System Identification

Stephan M. Winkler

Abstract. Estimating the similarity of solution candidates represented as structure trees is an important point in the context of many genetic programming (GP) applications. For example, when it comes to observing population diversity dynamics, solutions have to be compared to each other. In the context of GP based system identification, i.e., when mathematical expressions are evolved, solutions can be compared to each other with respect to their structure as well as to their evaluation. Obviously, structural similarity estimation of formula trees is not equivalent to evaluation based similarity estimation; we here want to see whether there is a significant correlation between the results calculated using these two approaches. In order to get an overview regarding this issue, we have analyzed a series of GP tests including both similarity estimation strategies; in this paper we describe the similarity estimation methods as well as the test data sets used in these tests, and we document the results of these tests. We see that in most cases there is a significant positive linear correlation for the results returned by the evaluation based and structural methods. Especially in some cases showing very low structural similarity there can be significantly different results when using the evaluation based similarity methods.

1 Solutions Similarity Estimation in GP Based System Identification

1.1 Related Work

Genetic diversity and population dynamics are very interesting aspects in the analysis of genetic programming (GP, [7, 8]) processes; several methods for measuring

Stephan M. Winkler
Department for Medical and Bioinformatics, Upper Austria University of Applied Sciences, Heuristic and Evolutionary Algorithms Laboratory
e-mail: stephan.winkler@fh-hagenberg.at

J.R. González et al. (Eds.): NICSO 2010, SCI 284, pp. 269–282, 2010.
springerlink.com

the diversity of population using some kind of similarity measure can be found in the literature.

The entropy of a population of trees can be measured for example by considering the programs' scores (as is explained for example in [13]); in [10] the traditional fitness sharing concept from the work described in [4] is applied to test its feasibility in GP.

Several other approaches consider the programs' genotypes, i.e., their genetic make-up instead of their fitness values, the most common type of diversity measure being that of structural differences between programs. Koza [7] used the term variety to indicate the number of different programs in populations by comparing programs structurally and looking for exact matches. The Levenshtein distance [9] can be used for calculating the distance between trees, but it is considered rather far from ideal ([6], [12], [8]); in [5] an edit distance specific to genetic programming parse trees was presented which considered the cost of substituting between different node types.

A comprehensive overview of program tree similarity and diversity measures has been given for instance in [3]. The standard tree structures representation in GP makes it possible to use more fine grain structural measures that consider nodes, subtrees, and other graph theoretic properties (rather than just entire trees). In [6], for example, subtree variety is measured as the ratio of unique subtrees over total subtrees and program variety as a ratio of the number of unique individuals over the size of the population; [11] investigated diversity at the genetic level by assigning numerical tags to each node in the population.

1.2 Solutions Similarity Estimation Measures Used in This Work

In this section we describe measures which we have used for estimating the genetic diversity in GP populations as well as among populations of multi-population GP applications. What we use as basic measures for this are the following two functions that calculate the similarity of GP solution candidates or, a bit more specific, in our case formulas represented as structure trees:

- *Evaluation based* similarity estimation compares the subtrees of two GP formulas with respect to their evaluation on the given training or validation data. The more similar these evaluations are with respect to the squared errors or linear correlation, the higher is the similarity for these two formulas.
- *Structural* similarity estimation compares the genetic material of two solution candidates; we can so determine how similar the genetic make-up of formulas is without considering their evaluation.

As documented for example in [15] and [2], these similarity estimation measures can be used for monitoring population diversity in GP populations. We have analyzed the effects of the use of several different selection schemes as well as multi-population approaches. Please note that in these applications we use similarity estimation in the following way: The similarity measures used here are asymmetric, so when comparing structure trees T_1 and T_2 there might be a difference between

the similarities $sim(T_1,T_2)$ and $sim(T_2,T_1)$. This is why we mostly use a symmetric variant of the measures described here: We calculate both similarity values and calculate their average as $sim_{avg}(T_1,T_2) = \frac{sim(T_1,T_2)+sim(T_2,T_1)}{2}$. This average similarity function (sim_{avg}) is used for estimating the similarities of GP individuals and monitoring the progress of genetic diversity in GP populations.

1.3 Evaluation Based Solutions Similarity Estimation

The main idea of our evaluation based similarity measures is that the building blocks of GP formulas are subtrees that are exchanged by crossover and so form new formulas. So, the evaluation of these branches of all individuals in a GP population can be used for measuring the similarity of two models m_1 and m_2:

For all sub-trees in the structure-tree of model m, collected in t, we collect the evaluation results by applying these sub-formulas to the given data collection *data* as

$$\forall(st_i \in t)\forall(j \in [1;N]) : e_i[j] = eval(st_i, data[j]) \tag{1}$$

where N is the number of samples included in the data collection, no matter if training or validation data are considered.

The evaluation based similarity of models m_1 and m_2, $es(m_1,m_2)$, is calculated by iterating over all subtrees of m_1 (collected in t_1) and, for each branch, picking that subtree of t_2 (containing all sub-trees of m_2) whose evaluation is most "similar" to the evaluation of that respective branch. So, for each branch b_a in t_1 we compare its evaluation e_a with the evaluation e_b of all branches b_b in t_2, and the "similarity" can be calculated using the sum of squared errors or the linear correlation coefficient:

- When using the sum of squared errors (*sse*) function, the sample-wise differences of the evaluations of the two given branches are calculated and their sum of squared differences is divided by the total sum of squares *tss* of the first branch's evaluation. This results in the similarity measure s for the given branches.

$$\overline{e_a} = \frac{1}{N}\sum_{j=1}^{N} e_a[j];\ \overline{e_b} = \frac{1}{N}\sum_{j=1}^{N} e_b[j] \tag{2}$$

$$sse = \sum_{j=1}^{N}(e_a[j]-e_b[j])^2;\ tss = \sum_{j=1}^{N}(e_a[j]-\overline{e_a})^2;\ s_{sse}(b_a,b_b) = 1 - \frac{sse}{tss} \tag{3}$$

- Alternatively the linear correlation coefficient can be used:

$$s_{lc}(b_a,b_b) = \left|\frac{\frac{1}{n-1}\sum_{j=1}^{N}(e_a[j]-\overline{e_a})(e_b[j]-\overline{e_b})}{\sqrt{\frac{1}{n-1}\sum_{j=1}^{N}(e_a[j]-\overline{e_a})^2}\sqrt{\frac{1}{n-1}\sum_{j=1}^{N}(e_b[j]-\overline{e_b})^2}}\right| \tag{4}$$

No matter which approach is chosen, the calculated similarity measure for the branches b_a and b_b, $s(b_a,b_b)$, will always be in the interval $[0;1]$; the higher this value becomes, the smaller is the difference between the evaluation results.

As we can now quantify the similarity of evaluations of two given subtrees, for each branch b_a in t_a we can elicit that branch b_x in t_b with the highest similarity to b_a; the similarity values **s** are collected for all branches in t_a and their mean value finally gives us a measure for the evaluation based similarity of the models m_a and m_b, $es(m_a, m_b)$.

Optionally we can force the algorithm to select each branch in t_b not more than once as best match for a branch in t_a for preventing multiple contributions of certain parts of the models.

Finally, this similarity function can be parameterized by giving minimum and maximum bounds for the height and / or the level of the branches investigated. This is important since we can so control which branches are to be compared, be it the rather small ones, rather big ones or all of them.

Further details about this similarity measure can be found in [15].

1.4 Structural Solutions Similarity Estimation

Structural similarity estimation is, unlike the evaluation based method described before, independent of data; it is calculated on the basis of the genetic make-up of the models which are to be compared. When analyzing the structure of models we have to be aware of the fact that often structurally different models can be equivalent. This is why we have designed and implemented a method that systematically collects all pairs of ancestor and descendant nodes and information about the properties of these nodes. Additionally, for each pair we also document the distance (with respect to the level in the model tree) and the index of the ancestor's child tree containing the descendant node. The similarity of two models is then, in analogy to the method described in the previous section, calculated by comparing all pairs of ancestors and descendants in one model to all pairs of the other model and averaging the similarity of the respective best matches.

Figure 1 shows a simple formula and all pairs of ancestors and descendants included in the structure tree representing it; the input indices as well as the level differences ("level delta") are also given. Please note: The pairs given on the right side of Figure 1 are shown intentionally as they symbolize the pairs of nodes with level difference 0, i.e., nodes combined with themselves.

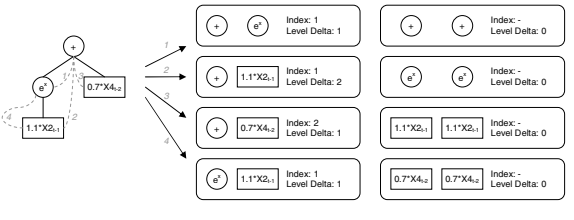


Fig. 1 Simple formula structure and all included pairs of ancestors and descendants (genetic information items)

We define a *genetic item* as a 6-tuple storing the following information about the ancestor node a and descendant node d: $type_a$ (the type of the ancestor a), $type_d$ (the type of the descendant d), δl (the level delta), $index$ (the index of the child branch of a that includes d), np_a (the node parameters characterizing a), and np_d (the node parameters characterizing d); the parameters characterizing nodes are represented by tuples containing the following information: var (the variant (of functions)), $coeff$ (the coefficient (of terminals)), to (the time offset (of terminals)), and vi (the variable index (of terminals)).

Now we can define the similarity of two *genetic items* gi_1 and gi_2, $s(gi_1, gi_2)$, as follows: Most important are the types of the definitions referenced by the nodes; if these are not equal, then the similarity is 0 regardless of all other parameters. If the types of the nodes correspond correctly, then the similarity of gi_1 and gi_2 is calculated using the similarity contributors $s_1 \ldots s_{10}$ of the parameters of gi_1 and gi_2 weighted with coefficients $c_1 \ldots c_{10}$. The similarity contributors $s_1 \ldots s_{10}$, all ranging from 0.0 to 1.0, are calculated with respect to input indices, variants, variable indices, level differences, coefficients, and time offsets; details can be found in [15] and [2].

Finally, there are two possibilities how to calculate the structural similarity of gi_1 and gi_2, $sim(gi_1, gi_2)$: On the one hand this can be done in an *additive* way, on the other hand in a *multiplicative* way.

- When using the *additive* calculation, which is the obviously more simple way, $sim(gi_1, gi_2)$ is calculated as the sum of these similarity contributions $s_{1\ldots10}$ weighted using the factors $c_{1\ldots10}$ and, for the sake of normalization of results, divided by the sum of the weighting factors:

$$sim_{add}(gi_1, gi_2) = \frac{\sum_{i=1}^{10} s_i \cdot c_i}{\sum_{i=1}^{10} c_i}. \tag{5}$$

- Otherwise, when using the *multiplicative* calculation method, we first calculate a punishment factor p_i for each s_i (again using weighting factors c_i, $0 \leq c_i \leq$ for all $i \in [1;10]$) as $\forall (i \in [1;10]) : p_i = (1 - s_i) \cdot c_i$ and then get the temporary similarity result as $sim_{tmp}(gi_1, gi_2) = \prod_{i=1}^{10}(1 - p_i)$.

  In the worst case scenario we get $s_i = 0$ for all $i \in [1;10]$ and therefore the worst possible sim_{tmp} is $sim_{worst} = \prod_{i=1}^{10}(1 - ((1 - s_i) \cdot c_i)) = \prod_{i=1}^{10}(1 - c_i)$. As sim_{worst} is surely greater than 0 we linearly scale the results to the interval $[0;1]$:

$$sim_{mult}(gi_1, gi_2) = \frac{sim_{tmp}(gi_1, gi_2) - sim_{worst}}{1 - sim_{worst}}. \tag{6}$$

  In fact, we prefer this *multiplicative* similarity calculation method since it allows more specific analysis: By setting a weighting coefficient c_j to a rather high value (i.e., near or even equal to 1.0) the total similarity will become very small for pairs of genetic items that do not correspond with respect to this specific aspect j, even if all other aspects would lead to a high similarity result.

Based on this similarity measure it is easy to formulate a similarity function that measures the structural similarity of two model structures. In analogy to the approach presented in the previous section, for comparing models m_1 and m_2 we collect all pairs of ancestors and descendants (up to a given maximum level difference) in m_1 and m_2 and look for the best matches in the respective opposite model's pool of genetic items, i.e., pairs of ancestor and descendant nodes. As we are able to quantify the similarity of genetic items, for each genetic item gi_1 in the structure tree of m_1 we can elicit exactly that genetic item gi_x in the model structure m_2 with the highest similarity to gi_1; the similarity values $\mathbf{s}$ are collected for all genetic items contained in m_1 and their mean value finally gives us a measure for the structure-based similarity of the models m_1 and m_2, $sim(m_1, m_2)$.

2 Test Setup

For comparing structural and evaluation based similarity values we executed GP based system identification experiments using the following two data sets:

- The NO_x data set contains the measurements taken from a 2 liter 4 cylinder BMW diesel engine at a dynamical test bench (simulated vehicle: BMW 320d Sedan). Several emissions (including NO_x, CO and CO_2) as well as several other engine parameters were recorded; for identifying formulas for the NO_x emissions we have only used parameters which are directly measured from the engine's control unit and not in any sense connected to emissions. We cordially thank members of the Institute for Design and Control of Mechatronical Systems at JKU, Linz[1] who provided and helped us with these data.
- The *Thyroid* data set is a widely used machine learning benchmark data set containing 21 attributes and 7200 samples representing the results of medical measurements which were recorded while investigating patients potentially suffering from hypotiroidism[2]. In short, the task is to determine whether a patient is hypothyroid or not; three classes are formed: normal (not hypothyroid), hyperfunction and subnormal functioning.

Detailed information about these two data collections can also be found in [15] as well as in [2].

For the target variables of both data collections we trained nonlinear models using a functional basis containing standard functions (such as for example addition, subtraction, multiplication, trigonometrics, conditionals, and others) as described in [16]; the maximum formula tree height was set to 6, the maximum number of nodes was set to 50. We have used the GP implementation for HeuristicLab [14] and applied two different training methods for training models for both data sets: Standard GP as well as GP using strict offspring selection (OS, [1]). In both cases the

[1] The homepage of the Institute for Design and Control of Mechatronical Systems at the Johannes Kepler University, Linz can be found at `http://desreg.jku.at/`

[2] Further information about the data set used can be found on the UCI homepage `http://www.ics.uci.edu/~{}mlearn/`

population size was set to 1000, we used single point crossover and 15% structural as well as parametric mutation as described in [15], e.g; in standard GP we applied tournament selection ($k = 3$), in GP with OS we applied gender specific parents selection combining random and proportional selection. For standard GP processes the number of iterations was set to 2000, GP runs with offspring selection were terminated as soon as the selection pressure reached 100.

All test cases were executed three times independently; the maximum tree height was set to 6, the maximum tree size to 50 (for NO_x as well as *Thyroid* tests). The similarity values among individuals were calculated in the context of population diversity estimation analysis executed after every 100^{th} generation in standard GP runs and after each 5^{th} generation in GP runs with offspring selection. We have thus collected the results of all similarity calculations; as this is done for 1,000 models we get 1,000,000 for each similarity function each time the population is analyzed. For each standard GP test we therefore eventually get 21 million similarity values for each function (because we also analyze after initializing the population), and for each GP test with OS we get a comparable amount of similarity values[3]. We will in the following not care whether standard or extended GP produced pairs of solutions are compared; in total we will use data of approximately 120 million solution comparisons for each function and each data set.

The following similarity estimation functions are used:

- Evaluation based similarity estimation: As described in Section 1.3, all subtrees are evaluated on training and validation data, and we can analyze the similarity of the values calculated by evaluating the subtrees of the formula trees which are to be compared. We here use validation data for this similarity estimation and the squared differences based approach.
- Additive structural similarity estimation: Structural components of structure trees are analyzed as described in Section 1.4 using the additive approach; we here weight all possible contributing aspects equally, i.e. the contributions' weighting factors $c_{1...10}$ are all set to 1.0, only the level difference is weighted stronger with factor 4.0.
- Multiplicative structural similarity estimation: Again, structural components of structure trees are analyzed as described in Section 1.4 using the multiplicative approach; again, we set all weighting factors equally, namely to 0.2, only the level difference is weighted stronger with factor 0.8.

3 Test Results

The NO_x test series are hereafter referred to as series *(n)*, the *Thyroid* runs as *(t)*. The similarity values calculated for the *(n)* series using evaluation based, additive structural and multiplicative structural comparison are hereafter denoted as $\mathbf{n_e}$, $\mathbf{n_{s1}}$ and $\mathbf{n_{s2}}$, respectively; in analogy to this, the similarity values for the *(t)* series are denoted as $\mathbf{t_e}$, $\mathbf{t_{s1}}$ and $\mathbf{t_{s2}}$, respectively.

[3] This number is not constant for extended GP with OS due to the fact that the selection pressure reaches its limit not at the same time in each test case execution.

Please note that for each index i the values $n_e(i)$, $n_{s1}(i)$ and $n_{s2}(i)$ belong to the same pair of models (structure trees) that have been compared; in analogy to this, for each index i also the corresponding comparison results $t_e(i)$, $t_{s1}(i)$ and $t_{s2}(i)$ are associated to the same pair of formulas.

All test runs were executed on Pentium© 4 computers with 3.00 GHz CPU speed and 2 GB RAM.

First, several statistics are calculated for the similarity values collected in $\mathbf{n_e}$, $\mathbf{n_{s1}}$, $\mathbf{n_{s2}}$, $\mathbf{t_e}$, $\mathbf{t_{s1}}$ and $\mathbf{t_{s2}}$; N_n stands for the number of values in $\mathbf{n_e}$, $\mathbf{n_{s1}}$ and $\mathbf{n_{s2}}$, N_t for the number of values in $\mathbf{t_e}$, $\mathbf{t_{s1}}$ and $\mathbf{t_{s2}}$. The results are summarized in Table 1; *std* here stands for standard deviation ($std(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_{i\in[1;N]}(x_i-\bar{x})^2}$, $\bar{x} = \frac{1}{N}\sum_{i\in[1;N]} x$, $N = |x|$), and *corr* again for the linear correlation (please see for example Section 1.3 for details about this function).

Table 1 Comparing similarity estimation results: Basic statistics

$mean(\mathbf{n_e}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(n_e(i))$	0.3444	$std(\mathbf{n_e}-\mathbf{n_{s1}})$	0.1625
$mean(\mathbf{n_{s1}}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(n_{s1}(i))$	0.6467	$std(\mathbf{n_e}-\mathbf{n_{s2}})$	0.1500
$mean(\mathbf{n_{s2}}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(n_{s2}(i))$	0.6061	$std(\mathbf{n_{s1}}-\mathbf{n_{s2}})$	0.0268
$mean(\mathbf{t_e}) = \frac{1}{N_n}\sum_{i\in[1;N_t]}(t_e(i))$	0.4224	$std(\mathbf{t_e}-\mathbf{t_{s1}})$	0.2159
$mean(\mathbf{t_{s1}}) = \frac{1}{N_n}\sum_{i\in[1;N_t]}(t_{s1}(i))$	0.6595	$std(\mathbf{t_e}-\mathbf{t_{s2}})$	0.1992
$mean(\mathbf{t_{s2}}) = \frac{1}{N_n}\sum_{i\in[1;N_t]}(t_{s2}(i))$	0.6327	$std(\mathbf{t_{s1}}-\mathbf{t_{s2}})$	0.0305
$mse(\mathbf{n_e},\mathbf{n_{s1}}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(n_e(i)-n_{s1}(i))^2$	0.1178	$corr(\mathbf{n_e},\mathbf{n_{s1}})$	0.8179
$mse(\mathbf{n_e},\mathbf{n_{s2}}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(n_e(i)-n_{s2}(i))^2$	0.0910	$corr(\mathbf{n_e},\mathbf{n_{s2}})$	0.8455
$mse(\mathbf{n_{s1}},\mathbf{n_{s2}}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(n_{s1}(i)-n_{s2}(i))^2$	0.0024	$corr(\mathbf{n_{s1}},\mathbf{n_{s2}})$	0.9954
$mse(\mathbf{t_e},\mathbf{t_{s1}}) = \frac{1}{N_n}\sum_{i\in[1;N_t]}(t_e(i)-t_{s1}(i))^2$	0.1028	$corr(\mathbf{t_e},\mathbf{t_{s1}})$	0.7634
$mse(\mathbf{t_e},\mathbf{t_{s2}}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(t_e(i)-t_{s2}(i))^2$	0.0839	$corr(\mathbf{t_e},\mathbf{t_{s2}})$	0.7998
$mse(\mathbf{t_{s1}},\mathbf{t_{s2}}) = \frac{1}{N_n}\sum_{i\in[1;N_n]}(t_{s1}(i)-t_{s2}(i))^2$	0.0016	$corr(\mathbf{t_{s1}},\mathbf{t_{s2}})$	0.9947
Runtime consumption per generation (evaluation based similarity)			2h08'30"
Runtime consumption per generation (structural similarity, per method)			38'02"

Obviously, the structural similarity values tend to be a lot higher than the evaluation based ones – which is not really surprising as even small changes in the formula's structure can affect its evaluation significantly. The mean squared difference between structural and evaluation based similarity values ranges from ∼0.08 to ∼0.12; the respective standard deviations of the similarity differences range from 0.15 to ∼0.216. The much more informative statistic feature is the linear correlation coefficient: Analyzing NO_x tests we see that the correlation between structural and evaluation based similarities is between ∼0.82 (for the additive structural calculation) and ∼0.8455 (for multiplicative structural approach); for the *Thyroid* tests, these are not quite as high, namely ∼0.76 and ∼0.8, respectively.

As we had expected, the correlation between the results calculated using the additive structural model comparison method and the multiplicative one is very high, namely approximately 0.995 for NO_x as well as *Thyroid* tests.

The runtime consumption of the evaluation based similarity estimation method is, of course, a lot higher than the runtime consumption caused by structural population diversity analysis: Although only 400 validation samples are evaluated for

evaluation based similarity estimation, structural similarity calculation consumes only approximately a fourth as much runtime.

Even more detailed results discussion becomes possible by partitioning all pairs of corresponding similarity values into five groups with equal range. This means that we collect all structural similarity results in the intervals [0.0 ... 0.2], [0.2 ... 0.4], ..., [0.8 ... 1.0]; of course, we also collect all evaluation based similarity values in the same intervals. Thus, what we get is a number of partitions of data sets which are defined and summarized in Table 2.

Table 2 Partitions formed for detailed comparison of similarity estimation results

Partition Index	*Index and Data Set Definitions*
$a0$	$I_{a0} = \{i : (0.0 \leq n_e(i) \leq 0.2)\}; \mathbf{n_e^{a0}} = \mathbf{n_e}(I_{a0}), \mathbf{n_{s1}^{a0}} = \mathbf{n_{s1}}(I_{a0}), \mathbf{n_{s2}^{a0}} = \mathbf{n_{s2}}(I_{a0})$
$a1$	$I_{a1} = \{i : (0.2 < n_e(i) \leq 0.4)\}; \mathbf{n_e^{a1}} = \mathbf{n_e}(I_{a1}), \mathbf{n_{s1}^{a1}} = \mathbf{n_{s1}}(I_{a1}), \mathbf{n_{s2}^{a1}} = \mathbf{n_{s2}}(I_{a1})$
$a2$	$I_{a1} = \{i : (0.4 < n_e(i) \leq 0.6)\}; \mathbf{n_e^{a2}} = \mathbf{n_e}(I_{a2}), \mathbf{n_{s1}^{a2}} = \mathbf{n_{s1}}(I_{a2}), \mathbf{n_{s2}^{a2}} = \mathbf{n_{s2}}(I_{a2})$
$a3$	$I_{a1} = \{i : (0.6 < n_e(i) \leq 0.8)\}; \mathbf{n_e^{a3}} = \mathbf{n_e}(I_{a3}), \mathbf{n_{s1}^{a3}} = \mathbf{n_{s1}}(I_{a3}), \mathbf{n_{s2}^{a3}} = \mathbf{n_{s2}}(I_{a3})$
$a4$	$I_{a1} = \{i : (0.8 < n_e(i) \leq 1.0)\}; \mathbf{n_e^{a4}} = \mathbf{n_e}(I_{a4}), \mathbf{n_{s1}^{a4}} = \mathbf{n_{s1}}(I_{a4}), \mathbf{n_{s2}^{a4}} = \mathbf{n_{s2}}(I_{a4})$
$b0$	$I_{b0} = \{i : (0.0 \leq n_{s1}(i) \leq 0.2)\}; \mathbf{n_e^{b0}} = \mathbf{n_e}(I_{b0}), \mathbf{n_{s1}^{b0}} = \mathbf{n_{s1}}(I_{b0}), \mathbf{n_{s2}^{b0}} = \mathbf{n_{s2}}(I_{b0})$
$b1$	$I_{b1} = \{i : (0.2 < n_{s1}(i) \leq 0.4)\}; \mathbf{n_e^{b1}} = \mathbf{n_e}(I_{b1}), \mathbf{n_{s1}^{b1}} = \mathbf{n_{s1}}(I_{b1}), \mathbf{n_{s2}^{b1}} = \mathbf{n_{s2}}(I_{b1})$
$b2$	$I_{b2} = \{i : (0.4 < n_{s1}(i) \leq 0.6)\}; \mathbf{n_e^{b2}} = \mathbf{n_e}(I_{b2}), \mathbf{n_{s1}^{b2}} = \mathbf{n_{s1}}(I_{b2}), \mathbf{n_{s2}^{b2}} = \mathbf{n_{s2}}(I_{b2})$
$b3$	$I_{b3} = \{i : (0.6 < n_{s1}(i) \leq 0.8)\}; \mathbf{n_e^{b3}} = \mathbf{n_e}(I_{b3}), \mathbf{n_{s1}^{b3}} = \mathbf{n_{s1}}(I_{b3}), \mathbf{n_{s2}^{b3}} = \mathbf{n_{s2}}(I_{b3})$
$b4$	$I_{b4} = \{i : (0.8 < n_{s1}(i) \leq 1.0)\}; \mathbf{n_e^{b4}} = \mathbf{n_e}(I_{b4}), \mathbf{n_{s1}^{b4}} = \mathbf{n_{s1}}(I_{b4}), \mathbf{n_{s2}^{b4}} = \mathbf{n_{s2}}(I_{b4})$
$c0$	$I_{c0} = \{i : (0.0 \leq n_{s2}(i) \leq 0.2)\}; \mathbf{n_e^{c0}} = \mathbf{n_e}(I_{c0}), \mathbf{n_{s1}^{c0}} = \mathbf{n_{s1}}(I_{c0}), \mathbf{n_{s2}^{c0}} = \mathbf{n_{s2}}(I_{c0})$
$c1$	$I_{c1} = \{i : (0.2 < n_{s2}(i) \leq 0.4)\}; \mathbf{n_e^{c1}} = \mathbf{n_e}(I_{c1}), \mathbf{n_{s1}^{c1}} = \mathbf{n_{s1}}(I_{c1}), \mathbf{n_{s2}^{c1}} = \mathbf{n_{s2}}(I_{c1})$
$c2$	$I_{c2} = \{i : (0.4 < n_{s2}(i) \leq 0.6)\}; \mathbf{n_e^{c2}} = \mathbf{n_e}(I_{c2}), \mathbf{n_{s1}^{c2}} = \mathbf{n_{s1}}(I_{c2}), \mathbf{n_{s2}^{c2}} = \mathbf{n_{s2}}(I_{c2})$
$c3$	$I_{c3} = \{i : (0.6 < n_{s2}(i) \leq 0.8)\}; \mathbf{n_e^{c3}} = \mathbf{n_e}(I_{c3}), \mathbf{n_{s1}^{c3}} = \mathbf{n_{s1}}(I_{c3}), \mathbf{n_{s2}^{c3}} = \mathbf{n_{s2}}(I_{c3})$
$c4$	$I_{c4} = \{i : (0.8 < n_{s2}(i) \leq 1.0)\}; \mathbf{n_e^{c4}} = \mathbf{n_e}(I_{c4}), \mathbf{n_{s1}^{c4}} = \mathbf{n_{s1}}(I_{c4}), \mathbf{n_{s2}^{c4}} = \mathbf{n_{s2}}(I_{c4})$
$d0$	$I_{d0} = \{i : (0.0 \leq t_e(i) \leq 0.2)\}; \mathbf{t_e^{d0}} = \mathbf{t_e}(I_{d0}), \mathbf{t_{s1}^{d0}} = \mathbf{t_{s1}}(I_{d0}), \mathbf{t_{s2}^{d0}} = \mathbf{t_{s2}}(I_{d0})$
$d1$	$I_{d1} = \{i : (0.2 < t_e(i) \leq 0.4)\}; \mathbf{t_e^{d1}} = \mathbf{t_e}(I_{d1}), \mathbf{t_{s1}^{d1}} = \mathbf{t_{s1}}(I_{d1}), \mathbf{t_{s2}^{d1}} = \mathbf{t_{s2}}(I_{d1})$
$d2$	$I_{d1} = \{i : (0.4 < t_e(i) \leq 0.6)\}; \mathbf{t_e^{d2}} = \mathbf{t_e}(I_{d2}), \mathbf{t_{s1}^{d2}} = \mathbf{t_{s1}}(I_{d2}), \mathbf{t_{s2}^{d2}} = \mathbf{t_{s2}}(I_{d2})$
$d3$	$I_{d1} = \{i : (0.6 < t_e(i) \leq 0.8)\}; \mathbf{t_e^{d3}} = \mathbf{t_e}(I_{d3}), \mathbf{t_{s1}^{d3}} = \mathbf{t_{s1}}(I_{d3}), \mathbf{t_{s2}^{d3}} = \mathbf{t_{s2}}(I_{d3})$
$d4$	$I_{d1} = \{i : (0.8 < t_e(i) \leq 1.0)\}; \mathbf{t_e^{d4}} = \mathbf{t_e}(I_{d4}), \mathbf{t_{s1}^{d4}} = \mathbf{t_{s1}}(I_{d4}), \mathbf{t_{s2}^{d4}} = \mathbf{t_{s2}}(I_{d4})$
$e0$	$I_{e0} = \{i : (0.0 \leq t_{s1}(i) \leq 0.2)\}; \mathbf{t_e^{e0}} = \mathbf{t_e}(I_{e0}), \mathbf{t_{s1}^{e0}} = \mathbf{t_{s1}}(I_{e0}), \mathbf{t_{s2}^{e0}} = \mathbf{t_{s2}}(I_{e0})$
$e1$	$I_{e1} = \{i : (0.2 < t_{s1}(i) \leq 0.4)\}; \mathbf{t_e^{e1}} = \mathbf{t_e}(I_{e1}), \mathbf{t_{s1}^{e1}} = \mathbf{t_{s1}}(I_{e1}), \mathbf{t_{s2}^{e1}} = \mathbf{t_{s2}}(I_{e1})$
$e2$	$I_{e2} = \{i : (0.4 < t_{s1}(i) \leq 0.6)\}; \mathbf{t_e^{e2}} = \mathbf{t_e}(I_{e2}), \mathbf{t_{s1}^{e2}} = \mathbf{t_{s1}}(I_{e2}), \mathbf{t_{s2}^{e2}} = \mathbf{t_{s2}}(I_{e2})$
$e3$	$I_{e3} = \{i : (0.6 < t_{s1}(i) \leq 0.8)\}; \mathbf{t_e^{e3}} = \mathbf{t_e}(I_{e3}), \mathbf{t_{s1}^{e3}} = \mathbf{t_{s1}}(I_{e3}), \mathbf{t_{s2}^{e3}} = \mathbf{t_{s2}}(I_{e3})$
$e4$	$I_{e4} = \{i : (0.8 < t_{s1}(i) \leq 1.0)\}; \mathbf{t_e^{e4}} = \mathbf{t_e}(I_{e4}), \mathbf{t_{s1}^{e4}} = \mathbf{t_{s1}}(I_{e4}), \mathbf{t_{s2}^{e4}} = \mathbf{t_{s2}}(I_{e4})$
$f0$	$I_{f0} = \{i : (0.0 \leq t_{s2}(i) \leq 0.2)\}; \mathbf{t_e^{f0}} = \mathbf{t_e}(I_{f0}), \mathbf{t_{s1}^{f0}} = \mathbf{t_{s1}}(I_{f0}), \mathbf{t_{s2}^{f0}} = \mathbf{t_{s2}}(I_{f0})$
$f1$	$I_{f1} = \{i : (0.2 < t_{s2}(i) \leq 0.4)\}; \mathbf{t_e^{f1}} = \mathbf{t_e}(I_{f1}), \mathbf{t_{s1}^{f1}} = \mathbf{t_{s1}}(I_{f1}), \mathbf{t_{s2}^{f1}} = \mathbf{t_{s2}}(I_{f1})$
$f2$	$I_{f2} = \{i : (0.4 < t_{s2}(i) \leq 0.6)\}; \mathbf{t_e^{f2}} = \mathbf{t_e}(I_{f2}), \mathbf{t_{s1}^{f2}} = \mathbf{t_{s1}}(I_{f2}), \mathbf{t_{s2}^{f2}} = \mathbf{t_{s2}}(I_{f2})$
$f3$	$I_{f3} = \{i : (0.6 < t_{s2}(i) \leq 0.8)\}; \mathbf{t_e^{f3}} = \mathbf{t_e}(I_{f3}), \mathbf{t_{s1}^{f3}} = \mathbf{t_{s1}}(I_{f3}), \mathbf{t_{s2}^{f3}} = \mathbf{t_{s2}}(I_{f3})$
$f4$	$I_{f4} = \{i : (0.8 < t_{s2}(i) \leq 1.0)\}; \mathbf{t_e^{f4}} = \mathbf{t_e}(I_{f4}), \mathbf{t_{s1}^{f4}} = \mathbf{t_{s1}}(I_{f4}), \mathbf{t_{s2}^{f4}} = \mathbf{t_{s2}}(I_{f4})$

Now we can analyze these partitions separately: For each partition we have calculated the linear correlation between evaluation based, additive structural and multiplicative structural similarities as well as the mean squared difference between these respective values; Table 3 summarizes these partition-wise statistics. Additionally, the frequency of each partition is also given: The frequency of a partition is hereby given by the number of pairs of values included divided by the number of all pairs of values available, $frequ(I_{ki}) = \frac{|I_{ki}|}{\sum_{j \in [0;4]} I_{kj}}$ for $k \in \{a,b,c,d,e,f\}$ and $i \in [0;4]$.

Table 3 Comparing similarity estimation results: Detailed partition-wise statistics

$freq(l_{a0}) = 0.3172$	$corr(\mathbf{n}_e^{a0}, \mathbf{n}_{s1}^{a0}) = 0.6294$ $mse(\mathbf{n}_e^{a0}, \mathbf{n}_{s1}^{a0}) = 0.1061$	$corr(\mathbf{n}_e^{a0}, \mathbf{n}_{s2}^{a0}) = 0.6772$ $mse(\mathbf{n}_e^{a0}, \mathbf{n}_{s2}^{a0}) = 0.0751$	$freq(l_{a1}) = 0.2609$	$corr(\mathbf{n}_e^{a1}, \mathbf{n}_{s1}^{a1}) = 0.8407$ $mse(\mathbf{n}_e^{a1}, \mathbf{n}_{s1}^{a1}) = 0.1083$	$corr(\mathbf{n}_e^{a1}, \mathbf{n}_{s2}^{a1}) = 0.8574$ $mse(\mathbf{n}_e^{a1}, \mathbf{n}_{s2}^{a1}) = 0.0818$
$freq(l_{a2}) = 0.2595$	$corr(\mathbf{n}_e^{a2}, \mathbf{n}_{s1}^{a2}) = 0.7886$ $mse(\mathbf{n}_e^{a2}, \mathbf{n}_{s1}^{a2}) = 0.1364$	$corr(\mathbf{n}_e^{a2}, \mathbf{n}_{s2}^{a2}) = 0.8047$ $mse(\mathbf{n}_e^{a2}, \mathbf{n}_{s2}^{a2}) = 0.1106$	$freq(l_{a3}) = 0.1272$	$corr(\mathbf{n}_e^{a3}, \mathbf{n}_{s1}^{a3}) = 0.6963$ $mse(\mathbf{n}_e^{a3}, \mathbf{n}_{s1}^{a3}) = 0.1279$	$corr(\mathbf{n}_e^{a3}, \mathbf{n}_{s2}^{a3}) = 0.7376$ $mse(\mathbf{n}_e^{a3}, \mathbf{n}_{s2}^{a3}) = 0.1077$
$freq(l_{a4}) = 0.0352$	$corr(\mathbf{n}_e^{a4}, \mathbf{n}_{s1}^{a4}) = 0.7174$ $mse(\mathbf{n}_e^{a4}, \mathbf{n}_{s1}^{a4}) = 0.1184$	$corr(\mathbf{n}_e^{a4}, \mathbf{n}_{s2}^{a4}) = 0.7559$ $mse(\mathbf{n}_e^{a4}, \mathbf{n}_{s2}^{a4}) = 0.0983$			
$freq(l_{b0}) = 0.0974$	$corr(\mathbf{n}_{s1}^{b0}, \mathbf{n}_e^{b0}) = 0.3815$ $mse(\mathbf{n}_{s1}^{b0}, \mathbf{n}_e^{b0}) = 0.1407$	$corr(\mathbf{n}_{s1}^{b0}, \mathbf{n}_{s2}^{b0}) = 0.9890$ $mse(\mathbf{n}_{s1}^{b0}, \mathbf{n}_{s2}^{b0}) = 0.0057$	$freq(l_{b1}) = 0.1222$	$corr(\mathbf{n}_{s1}^{b1}, \mathbf{n}_e^{b1}) = 0.6744$ $mse(\mathbf{n}_{s1}^{b1}, \mathbf{n}_e^{b1}) = 0.0884$	$corr(\mathbf{n}_{s1}^{b1}, \mathbf{n}_{s2}^{b1}) = 0.9931$ $mse(\mathbf{n}_{s1}^{b1}, \mathbf{n}_{s2}^{b0}) = 0.0028$
$freq(l_{b2}) = 0.1363$	$corr(\mathbf{n}_{s1}^{b2}, \mathbf{n}_e^{b2}) = 0.7591$ $mse(\mathbf{n}_{s1}^{b2}, \mathbf{n}_e^{b2}) = 0.0985$	$corr(\mathbf{n}_{s1}^{b2}, \mathbf{n}_{s2}^{b2}) = 0.9962$ $mse(\mathbf{n}_{s1}^{b2}, \mathbf{n}_{s2}^{b0}) = 0.0026$	$freq(l_{b3}) = 0.2451$	$corr(\mathbf{n}_{s1}^{b3}, \mathbf{n}_e^{b3}) = 0.8350$ $mse(\mathbf{n}_{s1}^{b3}, \mathbf{n}_e^{b3}) = 0.1080$	$corr(\mathbf{n}_{s1}^{b3}, \mathbf{n}_{s2}^{b3}) = 0.9963$ $mse(\mathbf{n}_{s1}^{b3}, \mathbf{n}_{s2}^{b0}) = 0.0024$
$freq(l_{b4}) = 0.3990$	$corr(\mathbf{n}_{s1}^{b4}, \mathbf{n}_e^{b4}) = 0.7677$ $mse(\mathbf{n}_{s1}^{b4}, \mathbf{n}_e^{b4}) = 0.1337$	$corr(\mathbf{n}_{s1}^{b4}, \mathbf{n}_{s2}^{b4}) = 0.9975$ $mse(\mathbf{n}_{s1}^{b4}, \mathbf{n}_{s2}^{b0}) = 0.0013$			
$freq(l_{c0}) = 0.1160$	$corr(\mathbf{n}_{s2}^{c0}, \mathbf{n}_e^{c0}) = 0.4119$ $mse(\mathbf{n}_{s2}^{c0}, \mathbf{n}_e^{c0}) = 0.0997$	$corr(\mathbf{n}_{s2}^{c0}, \mathbf{n}_{s1}^{c0}) = 0.9888$ $mse(\mathbf{n}_{s2}^{c0}, \mathbf{n}_{s1}^{c0}) = 0.0059$	$freq(l_{c1}) = 0.1335$	$corr(\mathbf{n}_{s2}^{c1}, \mathbf{n}_e^{c1}) = 0.7667$ $mse(\mathbf{n}_{s2}^{c1}, \mathbf{n}_e^{c1}) = 0.0580$	$corr(\mathbf{n}_{s2}^{c1}, \mathbf{n}_{s1}^{c1}) = 0.9961$ $mse(\mathbf{n}_{s2}^{c1}, \mathbf{n}_{s1}^{c0}) = 0.0023$
$freq(l_{c2}) = 0.1584$	$corr(\mathbf{n}_{s2}^{c2}, \mathbf{n}_e^{c2}) = 0.8229$ $mse(\mathbf{n}_{s2}^{c2}, \mathbf{n}_e^{c2}) = 0.0730$	$corr(\mathbf{n}_{s2}^{c2}, \mathbf{n}_{s1}^{c2}) = 0.9963$ $mse(\mathbf{n}_{s2}^{c2}, \mathbf{n}_{s1}^{c0}) = 0.0027$	$freq(l_{c3}) = 0.2728$	$corr(\mathbf{n}_{s2}^{c3}, \mathbf{n}_e^{c3}) = 0.8764$ $mse(\mathbf{n}_{s2}^{c3}, \mathbf{n}_e^{c3}) = 0.0794$	$corr(\mathbf{n}_{s2}^{c3}, \mathbf{n}_{s1}^{c3}) = 0.9967$ $mse(\mathbf{n}_{s2}^{c3}, \mathbf{n}_{s1}^{c0}) = 0.0021$
$freq(l_{c4}) = 0.3193$	$corr(\mathbf{n}_{s2}^{c4}, \mathbf{n}_e^{c4}) = 0.7528$ $mse(\mathbf{n}_{s2}^{c4}, \mathbf{n}_e^{c4}) = 0.1205$	$corr(\mathbf{n}_{s2}^{c4}, \mathbf{n}_{s1}^{c4}) = 0.9969$ $mse(\mathbf{n}_{s2}^{c4}, \mathbf{n}_{s1}^{c0}) = 0.0011$			
$freq(l_{d0}) = 0.3241$	$corr(\mathbf{t}_e^{d0}, \mathbf{t}_{s1}^{d0}) = 0.4233$ $mse(\mathbf{t}_e^{d0}, \mathbf{t}_{s1}^{d0}) = 0.1964$	$corr(\mathbf{t}_e^{d0}, \mathbf{t}_{s2}^{d0}) = 0.4777$ $mse(\mathbf{t}_e^{d0}, \mathbf{t}_{s2}^{d0}) = 0.1572$	$freq(l_{d1}) = 0.1239$	$corr(\mathbf{t}_e^{d1}, \mathbf{t}_{s1}^{d1}) = 0.8323$ $mse(\mathbf{t}_e^{d1}, \mathbf{t}_{s1}^{d1}) = 0.0336$	$corr(\mathbf{t}_e^{d1}, \mathbf{t}_{s2}^{d1}) = 0.8409$ $mse(\mathbf{t}_e^{d1}, \mathbf{t}_{s2}^{d1}) = 0.0295$
$freq(l_{d2}) = 0.2216$	$corr(\mathbf{t}_e^{d2}, \mathbf{t}_{s1}^{d2}) = 0.8455$ $mse(\mathbf{t}_e^{d2}, \mathbf{t}_{s1}^{d2}) = 0.0703$	$corr(\mathbf{t}_e^{d2}, \mathbf{t}_{s2}^{d2}) = 0.8606$ $mse(\mathbf{t}_e^{d2}, \mathbf{t}_{s2}^{d2}) = 0.0587$	$freq(l_{d3}) = 0.1919$	$corr(\mathbf{t}_e^{d3}, \mathbf{t}_{s1}^{d3}) = 0.8471$ $mse(\mathbf{t}_e^{d3}, \mathbf{t}_{s1}^{d3}) = 0.0688$	$corr(\mathbf{t}_e^{d3}, \mathbf{t}_{s2}^{d3}) = 0.8607$ $mse(\mathbf{t}_e^{d3}, \mathbf{t}_{s2}^{d3}) = 0.0566$
$freq(l_{d4}) = 0.1385$	$corr(\mathbf{t}_e^{d4}, \mathbf{t}_{s1}^{d4}) = 0.7956$ $mse(\mathbf{t}_e^{d4}, \mathbf{t}_{s1}^{d4}) = 0.0433$	$corr(\mathbf{t}_e^{d4}, \mathbf{t}_{s2}^{d4}) = 0.8109$ $mse(\mathbf{t}_e^{d4}, \mathbf{t}_{s2}^{d4}) = 0.0375$			
$freq(l_{e0}) = 0.1079$	$corr(\mathbf{t}_{s1}^{e0}, \mathbf{t}_e^{e0}) = 0.2693$ $mse(\mathbf{t}_{s1}^{e0}, \mathbf{t}_e^{e0}) = 0.1435$	$corr(\mathbf{t}_{s1}^{e0}, \mathbf{t}_{s2}^{e0}) = 0.9853$ $mse(\mathbf{t}_{s1}^{e0}, \mathbf{t}_{s2}^{e0}) = 0.0077$	$freq(l_{e1}) = 0.1043$	$corr(\mathbf{t}_{s1}^{e1}, \mathbf{t}_e^{e1}) = 0.3053$ $mse(\mathbf{t}_{s1}^{e1}, \mathbf{t}_e^{e1}) = 0.2216$	$corr(\mathbf{t}_{s1}^{e1}, \mathbf{t}_{s2}^{e1}) = 0.9854$ $mse(\mathbf{t}_{s1}^{e1}, \mathbf{t}_{s2}^{e0}) = 0.0024$
$freq(l_{e2}) = 0.1193$	$corr(\mathbf{t}_{s1}^{e2}, \mathbf{t}_e^{e2}) = 0.4652$ $mse(\mathbf{t}_{s1}^{e2}, \mathbf{t}_e^{e2}) = 0.2120$	$corr(\mathbf{t}_{s1}^{e2}, \mathbf{t}_{s2}^{e2}) = 0.9954$ $mse(\mathbf{t}_{s1}^{e2}, \mathbf{t}_{s2}^{e0}) = 0.0015$	$freq(l_{e3}) = 0.2412$	$corr(\mathbf{t}_{s1}^{e3}, \mathbf{t}_e^{e3}) = 0.8559$ $mse(\mathbf{t}_{s1}^{e3}, \mathbf{t}_e^{e3}) = 0.0479$	$corr(\mathbf{t}_{s1}^{e3}, \mathbf{t}_{s2}^{e3}) = 0.9986$ $mse(\mathbf{t}_{s1}^{e3}, \mathbf{t}_{s2}^{e0}) = 0.0006$
$freq(l_{e4}) = 0.4274$	$corr(\mathbf{t}_{s1}^{e4}, \mathbf{t}_e^{e4}) = 0.8376$ $mse(\mathbf{t}_{s1}^{e4}, \mathbf{t}_e^{e4}) = 0.0641$	$corr(\mathbf{t}_{s1}^{e4}, \mathbf{t}_{s2}^{e4}) = 0.9985$ $mse(\mathbf{t}_{s1}^{e4}, \mathbf{t}_{s2}^{e0}) = 0.0006$			
$freq(l_{e0}) = 0.1305$	$corr(\mathbf{t}_{s2}^{e0}, \mathbf{t}_e^{e0}) = 0.3430$ $mse(\mathbf{t}_{s2}^{e0}, \mathbf{t}_e^{e0}) = 0.0837$	$corr(\mathbf{t}_{s2}^{e0}, \mathbf{t}_{s1}^{e0}) = 0.9860$ $mse(\mathbf{t}_{s2}^{e0}, \mathbf{t}_{s1}^{e0}) = 0.0069$	$freq(l_{e1}) = 0.0978$	$corr(\mathbf{t}_{s2}^{e1}, \mathbf{t}_e^{e1}) = 0.3380$ $mse(\mathbf{t}_{s2}^{e1}, \mathbf{t}_e^{e1}) = 0.2230$	$corr(\mathbf{t}_{s2}^{e1}, \mathbf{t}_{s1}^{e1}) = 0.9964$ $mse(\mathbf{t}_{s2}^{e1}, \mathbf{t}_{s1}^{e0}) = 0.0021$
$freq(l_{e2}) = 0.1456$	$corr(\mathbf{t}_{s2}^{e2}, \mathbf{t}_e^{e2}) = 0.6722$ $mse(\mathbf{t}_{s2}^{e2}, \mathbf{t}_e^{e2}) = 0.1307$	$corr(\mathbf{t}_{s2}^{e2}, \mathbf{t}_{s1}^{e2}) = 0.9960$ $mse(\mathbf{t}_{s2}^{e2}, \mathbf{t}_{s1}^{e0}) = 0.0012$	$freq(l_{e3}) = 0.2435$	$corr(\mathbf{t}_{s2}^{e3}, \mathbf{t}_e^{e3}) = 0.8513$ $mse(\mathbf{t}_{s2}^{e3}, \mathbf{t}_e^{e3}) = 0.0505$	$corr(\mathbf{t}_{s2}^{e3}, \mathbf{t}_{s1}^{e3}) = 0.9986$ $mse(\mathbf{t}_{s2}^{e3}, \mathbf{t}_{s1}^{e0}) = 0.0007$
$freq(l_{e4}) = 0.3826$	$corr(\mathbf{t}_{s2}^{e4}, \mathbf{t}_e^{e4}) = 0.8501$ $mse(\mathbf{t}_{s2}^{e4}, \mathbf{t}_e^{e4}) = 0.0518$	$corr(\mathbf{t}_{s2}^{e4}, \mathbf{t}_{s1}^{e4}) = 0.9985$ $mse(\mathbf{t}_{s2}^{e4}, \mathbf{t}_{s1}^{e0}) = 0.0005$			

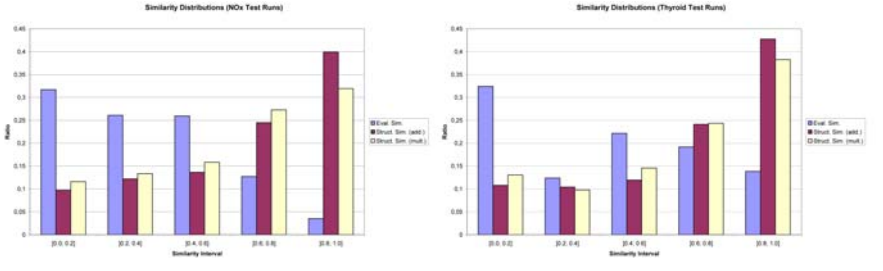
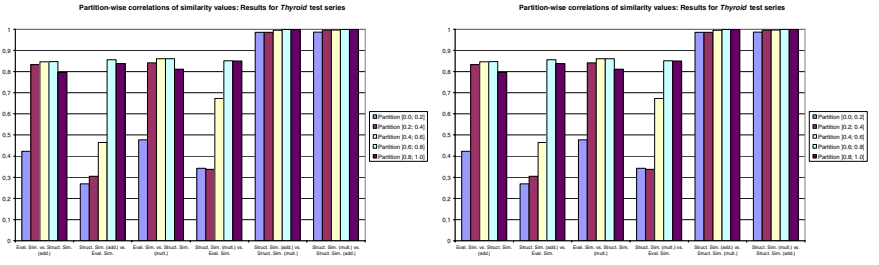


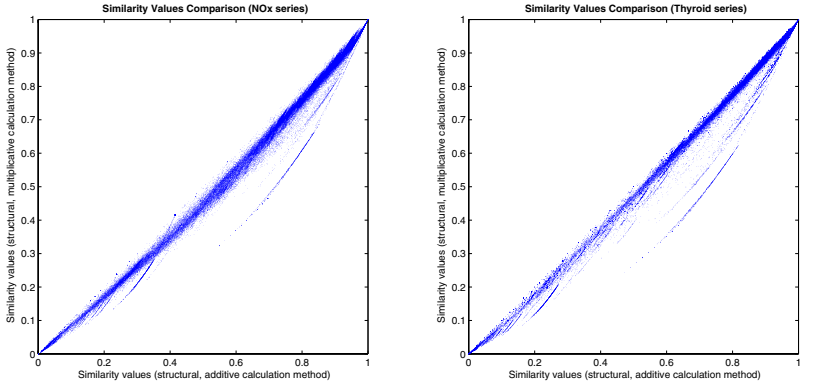
Fig. 2 Distribution of similarity values calculated using structural and evaluation based similarity functions

Figure 2 shows the distributions of structural and evaluation based similarity estimation for the NO_x and *Thyroid* tests separately. As we see in both charts the structural similarity values are significantly higher than the evaluation based ones.



(a) NO_x test series (b) *Thyroid* test series

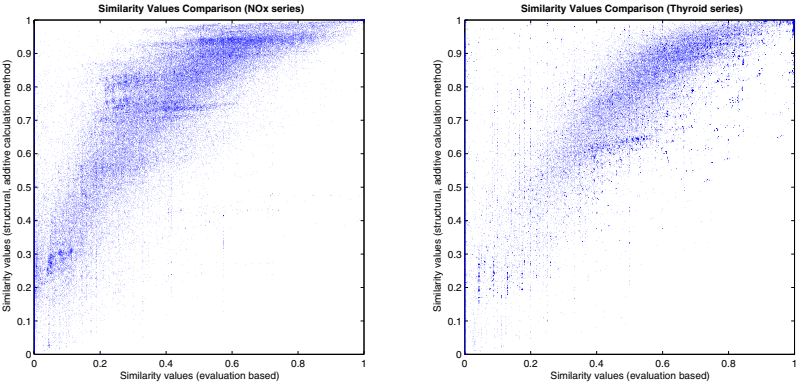
Fig. 3 Partition-wise correlations of similarity values for NO_x and *Thyroid* test series



(a) NO_x *test series* (b) *Thyroid* test series

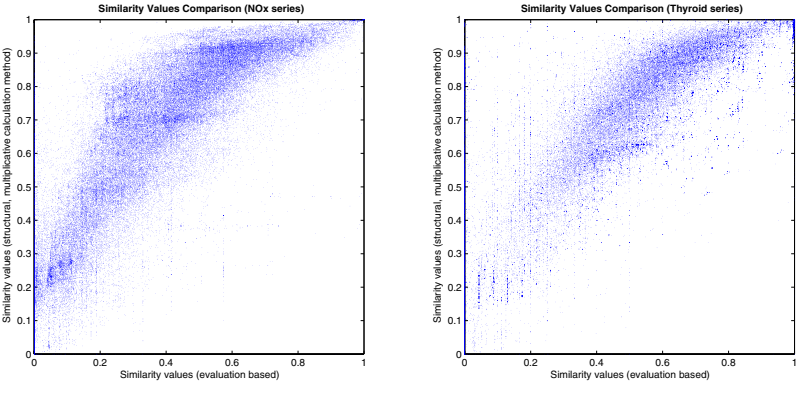
Fig. 4 Similarity values comparison: Structural (additive calculation) vs. structural (multiplicative calculation)

Regarding results correlations, the figures documented in Table 3 can be summarized in the following way: The correlations between structural and evaluation based similarity are approximately in the range between 0.3 and 0.85. Especially low correlation coefficients are calculated for the comparison of structural and evaluation based similarities, especially when the structural similarity is considered very low ($<$0.4). This impression becomes even more clear when we analyze Figures 3(a) and 3(b) which give the partition wise correlations of similarity values. In each of the 6 series shown in each of these figures we show the correlations of similarity values calculated by each possible pair of methods; in each case those partitions of value pairs are selected that correspond to the values calculated by the first method mentioned in the respective label. So, for example, in the first series we see the partition-wise correlations of similarity values calculated by the evaluation based and the additive structural method; the values are classified in partitions with respect to the evaluation specific similarities.



(a) NO_x test series (b) *Thyroid* test series

Fig. 5 Similarity values comparison: Evaluation based vs. structural (additive calculation)



(a) NO_x test series (b) *Thyroid* test series

Fig. 6 Similarity values comparison: Evaluation based vs. structural (multiplicative calculation)

The Figures 3(a) and 3(b) show clearly that the structural similarity estimation methods calculate very similar values (with high correlations for trees that are very different as well as for those which are considered rather similar). Furthermore, the correlation of structural and evaluation based similarity values is rather low in the case of low structural similarities (<0.4).

Finally, for graphically illustrating the direct comparison of similarity values calculated by the three estimation methods chosen we have randomly chosen 100,000 structure tree comparison cases both from the NO_x and the *Thyroid* tests. The respectively correspondent similarity values are drawn against each other in the

Figures 4 – 6. On the one hand there is no high correlation which can be seen when comparing structural and evaluation based similarity values, but on the other hand the high correlation between the similarities calculated by the structural similarity estimation methods becomes obvious.

4 Conclusion

In this paper we have summarized a series of GP test runs incorporating evaluation based as well as structural similarity estimation for measuring the genetic diversity in GP populations.

In general, evaluation based similarity calculation consumes a lot more runtime than structural comparison, and on average it also tends to produce lower similarity values. The results show that in most cases there is a linear correlation of approximately 0.4 – 0.9 for the results returned by the evaluation based and structural methods; not very surprisingly, this correlation is positive, but not very high. Especially in some cases showing very low structural similarity there can be significantly different results when using the evaluation based similarity methods.

Furthermore, we have also compared additive and multiplicative structural similarity estimation. These two variants tend to produce rather similar results with high correlations for pairs of structure trees with low as well as rather high similarities; the results retrieved by the multiplicative structural method show a higher correlation with those calculated using the evaluation based similarity function.

Thus, analyzing these correlations, we see that structural and evaluation based similarity measures give non-redundant information about the similarity of structure trees used in GP; both types of similarity measures should therefore be used for analyzing GP populations and algorithms.

Acknowledgments

The work described in this paper was done within the Translational Research Project L284-N04 “GP-Based Techniques for the Design of Virtual Sensors” funded by the Austrian Science Fund (FWF).

References

[1] Affenzeller, M., Wagner, S.: Offspring selection: A new self-adaptive selection scheme for genetic algorithms. In: Adaptive and Natural Computing Algorithms, pp. 218–221. Springer, Springer Computer Science, Heidelberg (2005)

[2] Affenzeller, M., Winkler, S., Wagner, S., Beham, A.: Genetic Algorithms and Genetic Programming - Modern Concepts and Practical Applications. Chapman & Hall / CRC (2009)

[3] Burke, E.K., Gustafson, S., Kendall, G.: Diversity in genetic programming: An analysis of measures and correlation with fitness. IEEE Transactions on Evolutionary Computation 8(1), 47–62 (2004)

[4] Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: Proceedings of the Third International Conference on Genetic Algorithms, pp. 42–50. Morgan Kaufmann, San Francisco (1989)
[5] Ekart, A., Nemeth, S.Z.: A metric for genetic programs and fitness sharing. In: Poli, R., Banzhaf, W., Langdon, W.B., Miller, J.F., Nordin, P., Fogarty, T.C. (eds.) EuroGP 2000. LNCS, vol. 1802, pp. 259–270. Springer, Heidelberg (2000)
[6] Keijzer, M.: Efficiently representing populations in genetic programming. In: Angeline, P.J., Kinnear Jr., K.E. (eds.) Advances in Genetic Programming, vol. 2, ch.13, pp. 259–278. MIT Press, Cambridge (1996)
[7] Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press, Cambridge (1992)
[8] Langdon, W.B., Poli, R.: Foundations of Genetic Programming. Springer, Heidelberg (2002)
[9] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
[10] McKay, R.I.B.: Fitness sharing in genetic programming. In: Whitley, D., Goldberg, D., Cantu-Paz, E., Spector, L., Parmee, I., Beyer, H.G. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000), pp. 435–442. Morgan Kaufmann, Las Vegas (2000)
[11] McPhee, N.F., Hopper, N.J.: Analysis of genetic diversity through population history. In: Banzhaf, W., et al. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, pp. 1112–1120. Morgan Kaufmann, Orlando (1999)
[12] O'Reilly, U.M.: Using a distance metric on genetic programs to understand genetic operators. In: IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, Florida, USA, vol. 5, pp. 4092–4097 (1997)
[13] Rosca, J.P.: Entropy-driven adaptive representation. In: Rosca, J.P. (ed.) Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications, Tahoe City, California, USA, pp. 23–32 (1995)
[14] Wagner, S.: Heuristic optimization software systems – modeling of heuristic optimization algorithms in the heuristiclab software environment. PhD thesis, Johannes Kepler University Linz (2009)
[15] Winkler, S.: Evolutionary system identification - modern concepts and practical applications. PhD thesis, Institute for Formal Models and Verification, Johannes Kepler University Linz (2008)
[16] Winkler, S., Affenzeller, M., Wagner, S.: Using enhanced genetic programming techniques for evolving classifiers in the context of medical diagnosis - an empirical study. Genetic Programming and Evolvable Machines 10(2), 111–140 (2009)