The handle http://hdl.handle.net/1887/22875 holds various files of this Leiden University dissertation

**Author:** Reehuis, Edgar  
**Title:** Guiding evolutionary search towards innovative solutions  
**Issue Date:** 2013-12-17
Criteria for Guiding Exploration

For the realm of real-valued design spaces, this chapter defines criteria for design-space exploration, to be used in iterated adaptive search methods. By iterated adaptive search we mean that after each iteration in the search, the search distribution from which solutions are sampled is adapted to better meet the search objective. Adaptive search potentially allows for quickly moving through regions that do not require closer investigation as they do not contain a lot of information, provided that it is guided by an exploration criterion that properly reflects this. Furthermore, adaptive search has more flexibility in setting the initial search distribution than an approach with a static search distribution, as it is able to self-adapt to appropriate settings. Using adaptive search, however, can cause unintended adaptation behavior in applying a dynamic exploration criterion: If the search distribution starts decreasing in magnitude, a scenario is possible where it keeps doing so, as, at these lower scales, the search landscape keeps changing as well, effectively causing exploration to stagnate.

We aim for design-space exploration to occur in a sequential, path-based fashion, in which exploration draws the search to a certain location and then repels it again, away from all earlier visited areas. This is a requirement arising from certain optimization problems in which the search needs to be started from a single, fixed solution (see Chapter 6). Right after the attraction phase, before being repelled, the search can turn to exploitation of the quality function. To be drawn to a certain location and then repelled from it, the exploration criterion needs to keep track of the history of where the search has been before, and is thereby inherently dynamic. Exploration needs a global view of the search space, whereas exploitation uses a greedy, local view, just comparing to the previous state.

This chapter provides an overview of dynamic exploration measures that are
Chapter 4 - Criteria for Guiding Exploration

based on ideas of novelty of solutions with respect to the history of visited solutions. Different views on novelty are presented, based, roughly, on a distance measure and on fitting a surrogate model. The model-based approach can be further used to derive interestingness of solutions, a more concise expression of the amount of useful knowledge that a solution introduces in the model.

In novelty based on a distance measure, a comparison is made to points in a reference set of solutions, whereas in novelty based on the prediction of a model, the surrogate model is generated from the reference set. It is the reference set that makes the novelty expression dynamic: During exploration, the reference set is updated with new solutions that were encountered. A static formulation of both variants of novelty is possible though with respect to an external reference set of solutions that were not generated while exploring. In using novelty as exploration criterion, one has the choice between

- comparing to a dynamic set of solutions generated earlier during the run;
- comparing to a static set of a number of known established solutions, to which the difference is to be maximized;
- comparing to a single solution only, from which maximally-different solutions are to be generated.

In moving away from a single, e.g., patented solution, incremental steps (i.e., successive exploration phases) may be necessary for the search to successfully traverse the search space, advocating the use of a dynamic set of sampled solutions instead of a single established solution as reference set: In using a single solution as reference, the necessary stepping stones for putting distance to this to-be-avoided solution may be obscured, as a sequence of seemingly counter-productive steps can be required. We therefore limit ourselves to the dynamic formulation of the exploration measures, derived from solutions generated during the search.

From Action-space Exploration to Design-space Exploration

Typically, in developmental robotics, large state spaces are to be searched efficiently through reinforcement learning. Whereas supervised learning is a passive approach in which an agent does not have influence on the data presented to it for training, in reinforcement learning, the agent itself actively chooses the data for which it
wants to receive feedback to build its understanding of the environment [Sutton and Barto, 1998, Barto and Dietterich, 2004]. Reinforcement learning generally tries to optimize an agent’s explorative behavior in its environment [Rückstieß et al., 2010]. The assumption underlying this active learning is that if the learning algorithm is allowed to choose the data from which it learns itself, it can achieve high accuracy using as few training instances as possible [Settles, 2010].

The idea of using novelty and interestingness of possible actions or solutions as an indication of what to try or where to explore next is widespread in neuro-control and reinforcement learning [Schmidhuber, 1991, Schmidhuber, 1997, Cuccu et al., 2011b, Luciw et al., 2011, Graziano et al., 2011, Oudeyer et al., 2007, Oudeyer and Kaplan, 2007, Baranes and Oudeyer, 2013, Hester and Stone, 2012, Lehman and Stanley, 2011]. To a lesser extent, it is being proposed for design-space exploration in black-box optimization [Graening et al., 2010, Schaul et al., 2011, Cuccu et al., 2011a]. A division is made between similarity-derived, distance-based novelty, and learning-based novelty related to fitting a surrogate model (in [Graziano et al., 2011] referred to by coherence and compression, respectively).

Learning-based novelty infers information from a reference set of performed actions through generating an internal model. Situations, based on actions taken, that could not have been predicted by the agent from previous experience are defined as novel, expanding its understanding of the environment. Situations that could have been predicted from previous experiences, despite never actually having been experienced, are not novel under this definition [Gero and Saunders, 2000]. The agent should choose actions for which the expectation of improvement of the model is high [Schmidhuber, 1991]. This interestingness of an action is based on its novelty, but is also related to how well an agent can adopt the information gained from novel experiences. New experiences similar but different from what was experienced before provide the best opportunity for rapid learning [Saunders and Gero, 2004]. Interestingness is the steepness of the learning curve [Schmidhuber, 2009], and by selecting on it we can maximize the knowledge gain.

This chapter provides an overview of novelty and interestingness expressions from the perspective of design-space exploration, by unifying the expressions in notation and proposing adjustments with this application in mind where necessary.
Interestingness in Data Mining

Interestingness measures in *data mining* are based on the same underlying ideas as discussed above. These measures express interestingness of found *association rules* to determine the most-relevant rules [Hébert and Crémilleux, 2007]. Several pattern discovery methods produce many obvious or irrelevant patterns, and do not fully account for valuable, prior domain knowledge that decision makers have [Padmanabhan and Tuzhilin, 1998]. For dealing with this, there are *objective* interestingness measures that depend on the structure of the found pattern and underlying data, and *subjective* interestingness measures that compare properties of a pattern with user beliefs or biases regarding relationships in the data [Silberschatz and Tuzhilin, 1996, Saunders, 2002, Hilderman and Hamilton, 2003]. An overview of objective and subjective interestingness measures in data mining is given in [Geng and Hamilton, 2006].

4.1 · Exploration Measures in Adaptive Search

In the following sections, we present different measures for inducing sequential exploration within an adaptive search method. On a two-dimensional artificial test function, visual analysis is performed of the sampling dynamics of selecting solutions using the exploration measure only: In the selection, no quality criterion is involved. Promising measures with respect to sampling dynamics, which is further specified below, will be selected for performance assessment when integrated as additional criterion into quality-based search in Chapter 5.

Test Setup

As test function a two-dimensional instance of a composite function termed *SineUnlearnable* is used, see Figure 4.1, with search domain $[-15, 15]^n \subseteq \mathbb{R}^n$, $n = 2$,

$$f_{\text{SineUnlearnable}}(x) = \begin{cases} \mathcal{U}(-5,5) & \text{if } -2 < x_i < 2 \text{ for } 1 \leq i \leq n \\ -\frac{1}{n} \sum_{i=1}^{n} \sin(x_i) & \text{otherwise} \end{cases}. \quad (4.1)$$

Test function SineUnlearnable has a cross-shaped region with extreme *noise* (i.e., uncertainty with an intrinsic stochastic nature [Kruisselbrink, 2012]) within which $f_{\text{SineUnlearnable}}$ takes on a different random value between $-5$ and $5$ for each evaluation.
Figure 4.1 Test Function SineUnlearnable. Composite test function that consists of a multivariate sine function on the one hand and a cross-shaped area with extreme noise (i.e., function values in this region differ for each evaluation of the test function) on the other hand.

of $f_{\text{SineUnlearnable}}$. In the rest of the search domain, the composite function behaves as a multivariate sine function.

The search is initialized in point $(0,0)$ in the middle of the unlearnable area, meaning that the region features too excessive noise to be concisely approximated using a surrogate model. Some of the exploration measures that are presented rely on approximation models (see Chapter 2), and we initialize in this region to determine whether these measures are able to make the search escape from it. When integrated in quality-based search, the exploration measures are to assist in finding a diverse set of high-quality solutions. We are thus interested in the pattern of exploration that is induced after the noisy region has been left, which will be judged by the ability of leading to diverse solutions.

As underlying search method, a $(3,6)$-CMA-ES is used with comma-selection and populations of 3 parents and 6 offspring (population sizes prescribed for two-dimensional search spaces, see Chapter 2), and an initial stepsize of 0.3. For constraint handling, we add a variable penalty to the criterion that is selected on, here the exploration measure, that increases linearly with the extent of violation of the lower boundary of $-15$ or the upper boundary of 15 per dimension.

For the exploration measures that employ approximation models, the used modeling technique and accompanying settings are described in the section defining the measure. All methods and techniques used have been implemented using the
Chapter 4 · Criteria for Guiding Exploration

Figure 4.2 Distance-based Novelty. A candidate solution is scored on novelty, based on its distance to the closest solutions in an archive of earlier encountered solutions during the search. A domain-specific distance measure that reflects the distinctiveness of solutions is required for the comparison.

Shark Machine Learning Library v2.3.43.

The remainder of this chapter is organized as follows: Section 4.2 presents exploration measures that directly express novelty of solutions, which will be categorized in distance-based and learning-based novelty. Section 4.3 then describes exploration measures that process learning-based novelty values further to express the interestingness of solutions. To be able to concisely describe and compare the different interestingness measures, a notational scheme is given as well as a means of determining a regional error value from observed learning-based novelty values. In Section 4.4, a summary of the chapter is given, presenting two promising exploration measures for integration in quality-based search.

4.2 · Novelty

To determine a candidate solution’s distinctiveness or novelty from the knowledge gathered on the design space, an appropriate representation of the available knowledge is required such that new solutions can be compared to it. Possible representations of the available knowledge are an archive containing all sampled solutions, and a surrogate model that learns a mapping between different characteristics of a solution, for instance from the solution vector in the search space to the quality value that becomes available after evaluation.

Novelty as it is introduced in Chapter 3 uses the distance to solutions in a reference set with respect to a domain-specific distance measure as novelty value. We will refer

\(^1\text{http://shark-project.sourceforge.net}\)
4.2 · Novelty

Figure 4.3 Learning-based Novelty. Based on a reference set of solutions encountered during the search, a reference model is generated that approximates the mapping between solution vectors and quality values. For a candidate solution, the error that the reference model makes in predicting its quality value serves as novelty value.

to this first type of novelty as distance-based novelty, see Figure 4.2, using an archive of sampled solutions as reference set. Alternatively, using the archive as reference set, one generates an approximation model and the error that the model makes in its approximation for a certain solution serves as its novelty value. We will refer to this type of novelty as learning-based novelty, see Figure 4.3. Chapter 3 associates the usage of a reference model with calculating interestingness but more accurately, in determining interestingness, the learning-based novelty based on two reference models is compared, see Section 4.3.

Selecting on distance-based novelty stepwise promotes domain-specific distance between solutions found and thus the diversity of the entire set of solutions found, with respect to the same domain-specific distance measure. In contrast, in selecting on learning-based novelty, the idea is that solutions with maximum error improve the surrogate model most after including them in training, maximizing learning progress. A scheme aimed at maximizing learning progress realizes this through estimating a pattern of exploration that is optimal for efficiently obtaining an accurate model of the search space. Potentially, in doing so, it implicitly samples solutions that are distant from each other with respect to the domain-specific distance measure, thereby constituting a diverse set.

4.2.1 · Distance-based Novelty

A straightforward way of representing knowledge gathered on the search space is through an archive of the solutions sampled during the search process. New solutions get a novelty score based on their distance to the closest point(s) in this archive. This
requires, however, an appropriate measure to indicate the distance between solutions. This work deals with search spaces consisting of real-valued vectors, therefore a generally-applicable distance formulation is the Euclidean distance between the solution vectors.

Certain optimization problems involve additional representations of solutions that better reflect their distinctiveness. Consider for instance a controller that is to steer a robot in navigating a maze, described in [Lehman and Stanley, 2008]. Here, the solution vector is a parameterization of the artificial neural network that controls the robot. However, the controller can be argued to be better represented by the end position in the maze that it gives rise to. Hence, [Lehman and Stanley, 2008] adopt the Euclidean distance between end positions as distance measure, instead of the Euclidean distance between the solution vectors. Thus, to come to a domain-specific distance measure, choose a characterizing representation first, and then a way of calculating the distance between two instances of that representation.

Distance-based novelty was put forward by [Lehman and Stanley, 2008] as a replacement objective to drive optimization, thereby ignoring the quality of solutions. There are no pre-defined fitness functions in nature; rather, there is ad-hoc fitness determination that is subject to change during the evolutionary process. Therefore, to artificially reproduce open-ended evolution that leads to increasing complexity in generated solutions, one can consider abandoning the a-priori quality objective and search for novel solutions instead. Not all optimization problems are suitable, however, for this pure novelty-search approach [Cuccu and Gomez, 2011]: Novelty search becomes computationally feasible in a scenario in which an appropriate distance measure reduces the effective size of the search space [Lehman and Stanley, 2008].

Alternatively, novelty and quality are jointly optimized on, which was shown to produce better results than pure novelty search in [Mouret, 2011, Cuccu and Gomez, 2011, Lehman et al., 2013]. In our used test setup, the optimization is on exploration measure only like in pure novelty search, but, as stated before, after the initial analysis the most-promising exploration criteria will be integrated into quality-based search in Chapter 5.

Interestingly, [Hester and Stone, 2012] employ another variant of pure novelty search: They optimize on distance-based novelty and learning-based novelty in parallel. A model is generated by using a learning-based novelty expression (similar to dispersion in predictions, see Section 4.2.2.3) that favors solutions with high modeling
error. This allows for efficient modeling, as the solutions for which the model can be improved greatly are selected. On the other hand, a distance-based novelty expression (uniqueness, see Section 4.2.1.2) is used to drive the exploration towards unvisited areas, to improve the global covering of the search space by the model.

### 4.2.1.1 · Sparseness

In [Lehman and Stanley, 2008], sparseness (Sp) is proposed as distance-based novelty measure. A candidate solution \( x \) is compared to solutions \( q \) from the union \( Q \) of a partial archive \( S \) of earlier sampled solutions and the set \( X \setminus \{x\} \) of all current candidate solutions excluding \( x \).

Sampling a solution means evaluating it on quality, and the archive \( S \) consists of tuples of solutions with accompanying quality value \((q, f(q))\). We use this definition of sampling for the sake of generality, although distance-based novelty does not inherently require evaluating solutions on quality, this depends on the domain-specific distance measure that is used.

The closest \( k \) neighbors of \( x \) are determined based on a domain-specific distance measure \( d(x, q) \),

\[
\text{argmin}_{\{q_1, \ldots, q_k\} \subseteq Q} \sum_{i=1}^{k} d(x, q_i), \tag{4.2}
\]

\[
Q = \{ q \mid (q, f(q)) \in S \} \cup X \setminus \{x\}. \tag{4.3}
\]

The sparseness of \( x \) is then taken as the average distance to these closest \( k \) solutions,

\[
\text{Sp}(x) = \frac{1}{k} \sum_{i=1}^{k} d(x, q_i). \tag{4.4}
\]

Settings for the following parameters are required, the listed settings are from [Lehman and Stanley, 2008, Lehman and Stanley, 2011]:

- Number of closest solutions \( k \): A value of 15 is used by Lehman and Stanley for varying problems;

- Dynamic archive-inclusion threshold \( \text{Sp}_{\text{min}} \): Only solutions with sparseness that exceeds this threshold are added to the archive. After some initial setting (we use 1.0), \( \text{Sp}_{\text{min}} \) is lowered by 5% if \( \frac{1}{100} \)-part of the sampling budget was used
Figure 4.4 Sparseness. The bottom-row plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on sparseness, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The top-row plots display the development of the sparseness scores, where the red dots represent the lastly-selected parent population. Selection on sparseness stepwise generates an evenly-spaced sampling that gets finer-grained as the run continues.

and no solutions were added; it is increased by 20% if \( \frac{1}{100} \)-part of the sampling budget was used and more than 4 solutions were added;

- Each sampled solution has an unconditional 0.1% chance to be added to the archive.

In Figure 4.4, the solutions generated in selecting on sparseness are displayed as well as the development of the sparseness scores, using the Euclidean distance between solution vectors as distance measure. In using this distance measure, the sampling behavior is unrelated to the quality function and leads to an evenly-spaced covering that is generated in segments, and that becomes finer-grained as selection on sparseness is continued.

Employing sparseness requires settings for a number of parameters, most of them involved in the update scheme for the dynamic archive-inclusion threshold. Lehman and Stanley do not provide clear guidelines on initializing the parameters, but simply
re-using their reported settings results in the aimed-for ongoing oscillating sampling behavior that generates diverse solutions.

4.2.1.2 · Uniqueness

Seeking to reduce the complexity of the scheme and the number of parameters involved, we reformulate sparseness as the uniqueness (Un) of a solution. The uniqueness is taken as the minimum distance that occurs to a solution $q$ in an archive $S$ containing all sampled solutions,

$$
\text{Un}(x) = \min_{(q, f(q)) \in S} d(x, q).
$$

This definition of distance-based novelty is also used in [Hester and Stone, 2012, Hester, 2012].

In Figure 4.5, the solutions generated in selecting on uniqueness and the development of the uniqueness scores are displayed, using the Euclidean distance between

Figure 4.5 Uniqueness. The bottom-row plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on uniqueness, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The top-row plots display the development of the uniqueness scores, where the red dots represent the lastly-selected parent population. Selecting on uniqueness, like selecting on sparseness, stepwise generates an evenly-spaced sampling that gets finer-grained as the run continues.
solution vectors as distance measure. Like in selecting on sparseness, the sampling behavior leads to an evenly-spaced covering that gets finer-grained sequentially.

In expressing uniqueness, there is no difference in novelty score in case the closest solution $q$ has no similar counterparts in the archive, compared to when the closest solution is part of a group of similar archived solutions. However, as all sampled solutions are added to the archive, this effect is of less consequence than it would be in a sparseness scheme with $k = 1$, as the chance for a solution of being included in the archive there depends heavily on the moment during the search that it is generated. Furthermore, the unconditional archive leads to a higher computational load than the partial archive in sparseness, but the archive size is of course bounded by the number of allowed to-be-sampled solutions (i.e., the quality-function evaluation budget).

4.2.2 · Learning-based Novelty

Maximizing distance-based novelty leads to diverse solutions with respect to the same distance measure, when this is the Euclidean distance between solution vectors. Selection on distance-based novelty is then likely to move through the search space at an optimal speed when applied in methods with adaptive search distributions, as simply the magnitude of the search distribution is increased. For a different domain-specific distance measure that for instance is not determined over the solution vectors, this is not necessarily the case. In such a scenario, we could possibly move through the search space more efficiently in maximizing learning progress.

The naive approach of maximizing learning progress is through maximizing learning-based novelty [Thrun, 1995]. Learning-based novelty requires no explicit distance measure: The distinctiveness from available knowledge is represented as the generalization performance that a surrogate model trained on the available knowledge has in approximating a to-be-learned mapping for an unseen solution. The error that the surrogate model makes in its prediction for a candidate solution serves as novelty expression [Thrun, 1995, Oudeyer and Kaplan, 2007, Graening et al., 2010, Cuccu et al., 2011b]. Typically, we will approximate the mapping from solution vectors to quality values. By including a solution in model training for which the predicted quality value showed high discrepancy with the actual quality value, the assumption is that we are able to greatly improve the model, thereby realizing high learning progress. Question is whether this approach is sufficient for implicitly finding diverse solutions, or that a more elaborate approach of estimating learning progress is required.
4.2 - Novelty

Figure 4.6 Ensemble of Local Models. A global model is emulated using an ensemble of local models. Each local model uses data from a different area in the search space for training. The global modeling error $\text{ME}(x, M)$ is obtained by evaluating on all local models and adopting the lowest modeling error.

Ensemble of Local Models

To be prepared for modeling irregular quality landscapes, we extend the learning-based novelty approach with a global model $M$ emulated through an ensemble of local models $M_i$ [Reehuis et al., 2011], see Figure 4.6. A global model approximates the entire search space, whereas a local model is aimed at only accurately approximating a certain part of the search space. The different local models are trained on disjoint or partly shared data. The global modeling error for a candidate solution $x$ is obtained by evaluating it on all local models and adopting the lowest resulting modeling error,

$$\text{ME}(x, M) = \min_i \text{ME}(x, M_i),$$

originating from the local model that provides the most-accurate approximation for $x$.

Each generation, we train a new local model on the last $\gamma$ generations of sampled solutions. Every $\gamma$-th local model is permanently stored. Together with these stored models, the last local model generated is used within the ensemble. The question remains how many training points to use for generating the local models. A sufficient number of points is required and these should not be too distant from each other. In [Loshchilov et al., 2012], some guidelines are provided depending on the dimensionality of the input space, the used test function, and the used modeling technique. We use $\gamma = 5$ generations of sample data for training, which in using the $(3,6)$-CMA-ES corresponds to maximally 30 training points per local model. This is deemed sufficient for generating local models of the two-dimensional test function. On a higher-dimensional problem, one can use a lower value of $\gamma$ to make sure that the points are not spread out too far. This is possible when in higher dimensions, more solutions are sampled per generation: For the CMA-ES, larger population sizes are prescribed in higher dimensions (see Chapter 2). Only solutions that lie within the lower and upper boundaries, set for the search space, are included in training. Note
that as we are using an adaptive search method, the distance between training points and thereby the locality of the derived model may show high variance between local models.

The proposed horizontal ensemble is different from the vertical model ensembles discussed in, e.g., [Jin and Sendhoff, 2004] and [Lim et al., 2010]. There, an average prediction is calculated using multiple vertical (local) models, trained on the same data, in order to increase prediction fidelity. We do use a similar approach of a vertical stack of local models in one of the learning-based novelty expressions, see Section 4.2.2.3.

4.2.2.1 · Prediction Error

For a surrogate model that approximates a mapping from solution vectors \( \mathbf{x} \) to quality-function values \( f(\mathbf{x}) \), the prediction error (PE) is calculated as the absolute difference between the model’s predicted quality value \( \hat{f}(\mathbf{x}) \) and the actual outcome \( f(\mathbf{x}) \) of evaluating \( \mathbf{x} \) by the quality function [Oudeyer and Kaplan, 2007, Graening et al., 2010],

\[
\text{PE}(\mathbf{x}) = |\hat{f}(\mathbf{x}) - f(\mathbf{x})|.
\]  

Given a model \( \mathcal{M} \), \( \text{PE}(\mathbf{x}) \) is the exact formulation of the modeling error \( \text{ME}(\mathbf{x}, \mathcal{M}) \).

Within the horizontal ensemble of local models, we employ feed-forward neural networks (FFNNs, see Chapter 2) as local modeling technique. The FFNNs have linear outputs and a hidden layer of 10 sigmoidally-activated nodes, a bias node, and are trained using improved Rprop [Igel and Hüsken, 2003] for a maximum of 10 epochs from random initial connection weights in \([-0.1, 0.1]\). They are fully connected, including direct connections from input to output nodes. All data is used for training, no validation sets are used. The local models should jointly represent the current knowledge of the quality landscape best, hence there is no risk of overfitting.

In Figure 4.7, the solutions generated in selecting on PE are displayed, as well as the development of the PE scores. Most notably, selecting on PE the search is not able to leave the unlearnable area featuring extreme noise. The prediction error remains constantly high despite of sampling, but this is in line with the definition and calls for a mechanism to cope with areas that cannot be modeled, which PE does not include. Furthermore, an unintended artifact of using the horizontal ensemble can be observed: As the generations pass, the prediction error decreases in areas where no points have been sampled. As more local models are added to the ensemble, the global modeling
Estimating the Prediction Error

Ranking on PE requires evaluating all candidate solutions on quality. Ideally, one only evaluates the most-promising candidates. Therefore, [Thrun, 1995] and [Oudeyer et al., 2007] describe an approach of using an additional error model, next to the quality model, that estimates the prediction error. Instead of on PE, solutions are ranked on the estimated prediction error $\hat{PE}(\mathbf{x})$ and only selected solutions get evaluated on quality. For these sampled solutions, the actual prediction error is then calculated with the sole purpose of training the error model, so it can predict the PE better for the next round of candidates.

In the following subsections, we present other approaches for estimating the prediction error. In these schemes only selected solutions are to be evaluated on quality as well, but they do not require training an additional error model.
4.2.2.2 · Predictive Variance

In using Gaussian Process modeling (GP) and Kriging approaches to surrogate modeling (see Chapter 2), next to the predicted value \( \hat{f}(x) \) (predictive mean), an estimate of the associated prediction error is readily available as the predictive variance (PV) [Rasmussen and Williams, 2006, Jones et al., 1998],

\[
PV(x) = \sigma^2(x) \ [GP] \doteq s^2(x) = \text{MSE}(x) \ [\text{Kriging}],
\]

assuming the actual quality value \( f(x) \) to be a realization of \( \mathcal{N}(\hat{f}(x), PV(x)) \).

PV is presented by [Schaul et al., 2011] as expressing interestingness through showing a direct relationship from PV to what they term expected information gain, i.e., estimated learning progress,

\[
\Psi(x) = \frac{1}{2} \log PV(x) - \frac{1}{2} \log \sigma^2_{\text{noise}},
\]

where the noise variance \( \sigma^2_{\text{noise}} \) [Rasmussen and Williams, 2006] is constant, and thus it holds that \( PV(x_1) < PV(x_2) \iff \Psi(x_1) < \Psi(x_2) \) for any two solutions \( x_1, x_2 \). We view PV as an expression of learning-based novelty, as PV, like PE, does not include a mechanism for dealing with regions of the search space that effectively cannot be modeled.

Furthermore, although PV provides an estimate of the modeling error and is thus grouped under learning-based novelty, ranking solutions on PV is governed by the spatial distribution of the sample points only, that is, ranking on PV is actually distance-based. We define ruggedness in a certain region of the quality landscape as the variation there in the outcome of the quality function. Although PV depends on the ruggedness of the quality landscape, relatively, PV depends only on the distribution of the sample points used in training, see Figure 4.8, because of the assumption that the ruggedness is uniform in each region of the input space (stationarity assumption). Local predictive variance that takes local ruggedness into account could be obtained by using a non-stationary variant of GP or Kriging [Xiong et al., 2007], or by re-training the surrogate for each candidate solution using its closest neighbors only from the available training data [Emmerich et al., 2006].

In our test setup, we follow [Schaul et al., 2011] who do not explicitly consider the stationarity assumption and rank multiple solutions using a single model, although we do use multiple local models within the horizontal ensemble (discussed in the
beginning of Section 4.2.2). There is non-stationarity between the local models in the horizontal ensemble, but within each local model, the stationarity assumption holds. Ranking multiple solutions by taking local irregularity into account, instead of density only, is therefore not possible using the same local model.

Ordinary Kriging (see Chapter 2) is employed\(^2\) with a power-exponential multi-parameter kernel, trained by tuning the \(\theta_i\) in \([10^{-10}, 10^2]\) and the \(p_i\) in \([1, 2]\) using Differential Evolution [Storn and Price, 1997] with population size 50, for 50 generations, followed by a hill-climbing phase. All data is used for training, no validation set is used.

In Figure 4.9, the solutions generated in selecting on PV are displayed, as well as the development of the PV scores. In our density-based setup, the search is able to leave the noisy region in selecting on PV. However, in doing so, it is probably hindered by the horizontal-ensemble approach leading to lower PV values from local models trained on samples adjacent to, but outside of the noisy region, as compared to local models trained on samples from within the noisy region. The non-stationarity between the local models within the ensemble leads to PV scores of different magnitude, clearly visible from generation 100 on. In generation 495, the search appears to be stuck, zooming in on an area of relatively low PV compared to other regions in the search space. This is likely to be due to the interplay of a dynamic exploration measure on the one hand and an adaptive search distribution on the other hand, in which the search

---

\(^2\) The used Kriging implementation was provided by Giles Endicott.
Figure 4.9 Predictive Variance. The bottom-row plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on PV, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The top-row plots display the development of the PV scores, where the red dots represent the lastly-selected parent population. While in selecting on PV, the search is able to leave the area with extreme noise, in a later phase it zooms in on an area with relatively low PV with other higher-PV regions around.

horizon, i.e., the solutions reachable by the search in a single step, can decrease too far to move out again.

4.2.2.3 · Dispersion in Predictions

The modeling error \( \text{ME}(\mathbf{x}, \mathcal{M}) \) of a given model \( \mathcal{M} \) in point \( \mathbf{x} \) depends on the local variation in the outcome of the quality function around \( \mathbf{x} \) and on the spatial distribution of the solutions used in training the surrogate model. As such, we propose to estimate \( \text{ME}(\mathbf{x}, \mathcal{M}) \) using a vertical ensemble of multiple surrogate models \( \mathcal{M}_j \), trained on the same data [Jin and Sendhoff, 2004], and measuring the variation in their predictions for \( \mathbf{x} \) [Reehuis et al., 2011] (independently proposed in [Hester and Stone, 2012, Hester, 2012]). In presence of sufficient data, the predictions should largely align, while otherwise be more diverse.

For expressing the variation between the predictions, we determine their interquartile range (IQR), which is the difference between the third and first quartile of the
Figure 4.10 Ensemble of Local Model Stacks. A global model stack is obtained using an ensemble of local model stacks. Within local stack \( i \), all surrogate models \( M_{i,j} \) are trained on the same training data, while different local stacks use data from different areas in the search space. The lowest DP score is selected after evaluating a solution \( x \) on all stacks.

Values and is chosen as statistic because it is not prone to outliers. We term this estimation of the modeling error dispersion in predictions (DP), calculated over the quality predictions \( \hat{f}_j(x) \) by the \( \dim(\hat{f}(x)) \) models in the vertical ensemble, or model stack [Reehuis et al., 2011, Reehuis et al., 2013a, Reehuis et al., 2013c],

\[
DP(x) = \text{IQR}(\hat{f}_1(x), \ldots, \hat{f}_\dim(\hat{f}(x))(x))
\]

(4.10)

DP is assumed to provide a similar ordering between candidate solutions as ranking them on their actual PE would, without the requirement of evaluating candidate solutions on quality beforehand; formally, for any two solutions \( x_1, x_2 \) it is assumed that

\[
PE(x_1) < PE(x_2) \iff DP(x_1) < DP(x_2).
\]

(4.11)

DP requires surrogate models with potentially differing outcomes, either a combination of different modeling techniques or multiple instances of a technique that may vary in prediction outcome when differently initialized. In [Jin and Sendhoff, 2004], neural networks of varying structure are used, whereas [Hester and Stone, 2012] use a collection of decision trees (i.e., a random forest) that are each trained on a different, partly-overlapping subset of the available data. We use 10 stacked instances of the same FFNNs that was used for PE, all sharing the same architecture but initialized from different random weights and with the training points supplied in varying random order.
Figure 4.11 Dispersion in Predictions. The bottom-row plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on DP, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The top-row plots display the development of the DP scores, where the red dots represent the lastly-selected parent population. Selection on DP is able to leave the noisy area, but then gets stuck in a region with relatively low DP scores.

**Ensemble of Local Model Stacks**

Thus, instead of using a single local model, DP is calculated using multiple overlapping local models, a *local model stack*. To obtain a *global model stack*, we extend on the ensemble of local models used for PE and PV, and generate an *ensemble of local model stacks*, see Figure 4.10.

For a local model stack with index \( i \), the vector

\[
\hat{f}_i(x) = \left( \hat{f}_{i,1}(x), \ldots, \hat{f}_{i,\text{dim}}(\hat{f}_{i}(x)) \right)
\]

contains the predictions for \( x \) by its dim \( \hat{f}_i(x) = 10 \) models. DP is reformulated as the minimum IQR score resulting from evaluating a solution \( x \) on all local model stacks,

\[
\text{DP}(x) = \min_i \text{IQR} \left( \hat{f}_i(x) \right),
\]

as the models in the originating stack are likely to have the most-accurate prediction for that solution.
In Figure 4.11, the solutions generated in selecting on DP are displayed, as well as the development of the DP scores. In selecting on DP, the search is able to leave the noisy area but, like with PV, gets stuck in a region with relatively low DP scores. Again, this is likely due to self-adaptation in a dynamic scenario that decreases the search horizon to a point where other areas featuring higher DP are obscured.

### 4.3 · Interestingness

In selecting on learning-based novelty, solutions with maximum modeling error are promoted as predictor of their to-be-induced learning progress after inclusion in model training. Such an estimator of learning progress, however, does not account for detecting and avoiding notoriously hard regions in the search space that are effectively impossible to model given a certain used modeling approach, for instance because of extreme non-stationary noise [Kruisselbrink, 2012].

From the results obtained by the learning-based novelty expressions, it can be seen that a better estimator of the learning progress is required. On the one hand, these show attraction to regions with noise, on the other hand, the search is seen to stall in seemingly random locations because of a decreasing search horizon combined with ranking solutions based on dynamic values. Instead of directly selecting on the learning-based novelty that the latest state of the surrogate model gave rise to, we can compare the modeling error at different moments in time during the learning process, see Figure 4.12.

In selecting on interestingness, those solutions are picked that improve or are expected to improve the surrogate model’s approximation of the to-be-learned mapping most by including them in model training, after sampling them. This is equivalent to the aim of learning-based novelty, but interestingness includes an additional scheme for detecting or estimating whether a certain region can effectively be modeled.

In maximizing learning progress, the goal is to invoke an optimal sampling pattern for building an accurate model of the search space. Interestingness should help skipping unlearnable areas, but does not express the chance of finding high-quality solutions somewhere. An area of the quality landscape featuring higher variation in quality than average should in principle remain interesting longer as it requires a more dense sampling for accurate modeling than a more regular landscape. However, this does not mean that this area is a good candidate for exploitation on quality! The area can be rugged but feature far-from-optimal quality values only. Furthermore, that it
Figure 4.12 Interestingness. Based on the development of the learning-based novelty in the region that a candidate solution lies in, (for instance) determined before and after including it in model training, interestingness expresses the learning progress induced by the candidate solution (or in its region).

remained interesting for a relatively long time could also be due to the fact that it is unlearnable. Our assumption is that exploration based on interestingness assists in efficiently uncovering diverse solutions, as it should move quickly through areas with low information density or with properties making them unlearnable, and thus with high likeliness unexploitable via optimization on quality.

4.3.1 · Notational Scheme

The interestingness measures that are presented in Section 4.3.3 determine or derive the learning progress that a candidate solution induces after inclusion in model training. For clearly formulating the different interestingness expressions, we present the following general notational scheme:

- $t$ is the current position in time; time is viewed as passing discretely, each iteration of the iterative search method defining a single time step; $t$ is initialized at 0;

- $x_{i,t}$ is a candidate solution from the set $X_t$ of all candidate solutions at time $t$, with $1 \leq i \leq |X_t|$; notably, $X_t$ contains the solutions that are candidate at time $t$, not all previous candidates up to time $t$;
4.3 · Interestingness

- \( \mathcal{R}(x_{i,t}) \) denotes the region in the design space in which \( x_{i,t} \) lies; a definition of what constitutes a region is given in Section 4.3.2;

- \( s_{j,t} = (q, f(q)) \) is a solution \( q \) sampled at time \( t \), that is, for which quality value \( f(q) \) was determined; possibly, multiple solutions are sampled at a single time step \( t \), hence the index \( j \);

- \( S_t = \bigcup_j \{s_{j,t}\} \) is the set of all solutions sampled at time \( t \); not necessarily all candidate solutions at time \( t \) are sampled: If the modeling errors are estimated, only a subset of \( X_t \) needs to be sampled;

- \( \mathcal{M}(T) \) is a surrogate model that approximates the mapping from solution vectors \( x \) to quality values \( f(x) \), and that is trained on tuples of solutions with quality values in the supplied training set \( T \);

- \( T_t = \bigcup_{k=0}^{t-1} S_k \) is the training data available for \( \mathcal{M} \) at time \( t \), containing tuples of the solutions sampled at all earlier time steps with accompanying quality values;

- \( V_t \) is the test set for determining the performance of \( \mathcal{M} \) at time \( t \); in case of PE, \( V_t \) needs to contain tuples of solution vectors with accompanying quality values \( (v, f(v)) \), in case of DP and PV the quality component is not used;

- \( \text{ME}(v, \mathcal{M}) \) is the modeling error that \( \mathcal{M} \) gives rise to for a solution \( (v, f(v)) \in V_t \) at time \( t \); it is either determined as \( \text{PE}(v) \), or approximated as \( \text{DP}(v) \) or \( \text{PV}(v) \);

- \( \text{Err}(\mathcal{R}, \mathcal{M}, V) \) is the regional error value for region \( \mathcal{R} \) that is derived from the modeling errors \( \text{ME}(v, \mathcal{M}) \) for solutions \( v \in V \).

4.3.2 · Regional Error Value

Central to all interestingness measures is the need for determining regional error values from the observed modeling errors. The interestingness measures compare the regional error at two time steps; as such, it needs to be determined which of the tested points at a certain time step belong to a certain region, and multiple modeling errors are to be aggregated into a regional error value.

To come to a definition for the regional error, we start from the idea that the influence of an observed modeling error on neighboring points can be expressed using a Gaussian function. Therefore, instead of exclusively assigning test points \( v \)
Figure 4.13 Regional Error Value. The regional error $\text{Err}(\mathcal{R}(x), \mathcal{M}, V)$ in the region $\mathcal{R}(x)$ that is defined by the position of $x$, is determined by multiplying the modeling errors in the test points $v$ with an expression of their influence in region $\mathcal{R}(x)$. The influence of a test point in region $\mathcal{R}(x)$ is calculated as $\text{pdf}_{N(v, I)}(x)$, expressing its influence using a standard normal distribution around the test point $v$. The maximum of these modeling errors times their influence, at position $x$, is adopted as the regional error.

to regions, the influence of a test point’s modeling error $\text{ME}(v, \mathcal{M})$ is calculated using a specific Gaussian function, namely, the probability density function of the standard normal distribution $N(v, I)$ around $v$, $\text{pdf}_{N(v, I)}(x)$. Hence, perhaps a bit contradictory, a region $\mathcal{R}(x)$ is defined merely by the position of $x$ that indicates the influence of the test points’ modeling errors. The regional error value is calculated as the maximum of the modeling errors $\text{ME}(v, \mathcal{M})$ in the points $v$, from the test set $V$, multiplied by each point’s influence [Reehuis et al., 2013a, Reehuis et al., 2013c], see Figure 4.13. Thus, we determine the test point $v$ with the highest resulting modeling error after decreasing it relatively to the distance between $x$ and $v$, based on the probability density function of a normal distribution around $v$,

$$\text{Err}(\mathcal{R}(x), \mathcal{M}, V) = \max_{v \in V} \left( \frac{\text{ME}(v, \mathcal{M}) \cdot (2\pi)^{-\frac{1}{2}|v|} \exp \left( -\frac{1}{2} (x - v)'I(x - v) \right)}{\text{influence of } v \text{ on } x, \text{calculated as } \text{pdf}_{N(v, I)}(x)} \right). \quad (4.14)$$

The result is that all regions get an error value at each time step for which training data is available, anywhere in the search space, and if no training data is available at time $u$, the regional error $\text{Err}(\mathcal{R}, \mathcal{M}(T_u = \emptyset), V_u)$ is set to 0.

Most of the interestingness measures in Section 4.3.3 are intended to be calculated by comparing regional error averages $\langle \text{Err}(\mathcal{R}, \mathcal{M}, V) \rangle$ over two time periods [Oudeyer et al., 2007, Baranes and Oudeyer, 2013, Luciw et al., 2011], instead of regional error
values $\text{Err}(\mathcal{R}, \mathcal{M}, V)$ at two singular time steps. Next, we therefore give an overview of different ways of obtaining two distinct regional error averages.

**Singular Errors**

Resuming, the first option for obtaining regional errors at time steps $u$ and $u-\tau$ is thus simply adopting the non-averaged, singular error values at those time steps,

$$\langle \text{Err}(\mathcal{R}, \mathcal{M}(T_u), V_u) \rangle \leftarrow \text{Err}(\mathcal{R}, \mathcal{M}(T_u), V_u),$$  \hspace{1cm} (4.15)

where the test sets $V_u$ and $V_{u-\tau}$ are defined by the used interestingness measure.

**Separated Averages**

[Oudeyer et al., 2007] propose calculating the mean error over two time periods with window size $w$ that lie $\tau$ time steps apart. The mean error over the period ending in time step $u$ and over the period ending in $u-\tau$ is calculated as follows,

$$\langle \text{Err}(\mathcal{R}, \mathcal{M}(T_u), V_u) \rangle \leftarrow \frac{1}{w} \sum_{k=0}^{w-1} \text{Err}(\mathcal{R}, \mathcal{M}(T_{u-k}), V_{u-k}).$$  \hspace{1cm} (4.16)

Based on [Oudeyer et al., 2007], we use $w = \min \left(15, \left\lfloor \frac{1}{2} \cdot u \right\rfloor \right)$ and $\tau = \min(25, u-w)$; $w$ and $\tau$ are increased until its listed value has become possible.

**Adjacent Averages**

In [Baranes and Oudeyer, 2013], a simplification of this scheme is proposed through assigning the $2 \cdot w$ most-recent time steps to the two error averages, taking $w = \left\lfloor \frac{1}{2} \cdot \min(40, u) \right\rfloor$ and $\tau = w$.

**Short-term/Long-term Memory**

[Luciw et al., 2011] use exponential smoothing to obtain error averages in the form of two error memories, one with a slower forgetting rate ($\text{Mem}_{\text{long}}$) than the other ($\text{Mem}_{\text{short}}$),

$$\text{Mem}_{\text{long}}(\mathcal{R}, u) = (1 - \eta_{\text{long}}) \cdot \text{Mem}_{\text{long}}(\mathcal{R}, u - 1) + \eta_{\text{long}} \cdot \text{Err}(\mathcal{R}, \mathcal{M}(T_u), V_u),$$

$$\text{Mem}_{\text{short}}(\mathcal{R}, u) = (1 - \eta_{\text{short}}) \cdot \text{Mem}_{\text{short}}(\mathcal{R}, u - 1) + \eta_{\text{short}} \cdot \text{Err}(\mathcal{R}, \mathcal{M}(T_u), V_u),$$  \hspace{1cm} (4.17)
using $\eta_{\text{long}} = 0.1$ for forgetting slowly, and $\eta_{\text{short}} = 0.2$ for forgetting quicker (settings from [Luciw et al., 2011]). Mem$_{\text{long}}(R,u)$ is used as the regional error average at time $u - \tau$, and Mem$_{\text{short}}(R,u)$ is used as the regional error average at time $u$.

### 4.3.3 · Interestingness Measures

Before we can describe the interestingness measures, a definition of the observed learning progress in a region $R$ at time $u$ is required. This is calculated as the discrete-time derivative of the regional error $\text{Err}(R,M,V)$, taken as the decrease of the error from time $u - \tau$ to time $u$ [Oudeyer and Kaplan, 2007],

$$\text{LP}(R,u) = \text{Err}(R,M(T_{u-\tau}),V_{u-\tau}) - \text{Err}(R,M(T_u),V_u). \quad (4.18)$$

To determine the performance of surrogate model $M(T_u)$, at each time step $u$, the set $X_u$ of candidate solutions at time $u$ serves as test set $V_u$, as these are the solutions for which the modeling error was determined,

$$\frac{\text{Err}(R,M(T_u),V_u = X_u)}{\text{model performance in } R \text{ at time } u}. \quad (4.19)$$

Now, importantly, Equation 4.18 provides the learning progress observed in region $R$ at time $u$, depending on the modeling errors observed in the candidate solutions at time $u$. However, in order to determine the learning progress that a solution that is candidate at time $u$ itself induces, we have to be one time step in the future, at time $u + 1$. The model has to be re-trained, now including the candidate solution, and then tested again in the same region.

Therefore, to determine the learning progress that a solution that is candidate at time $t$ will have given rise to at time $t + 1$, the following steps are to be performed:

1. At time $t + 1$, in the region that candidate solution $x_{i,t}$ defines, determine the previous regional error based on the modeling errors that the model at time $t$ (taking $\tau = 1$) showed for the set $X_t$ of candidate solutions at time $t$,

$$\text{Err}(\mathcal{R}(x_{i,t}),M(T_t),V_t = X_t); \quad (4.20)$$

2. Re-train the model on the union of the training set used at time $t$ and the candidate solution $x_{i,t}$, which needs to have been sampled, and again determine the error in region $\mathcal{R}(x_{i,t})$, now defined as

$$\text{Err}(\mathcal{R}(x_{i,t}),M(T_t \cup \{x_{i,t}, f(x_{i,t})\}),V_{t+1} = X_{t+1}); \quad (4.21)$$
thus, the regional error at time $t + 1$ is based on the modeling errors observed in the new set of candidate solutions $X_{t+1}$;

3. By subtracting the regional error at time $t$ and $t + 1$ according to Equation 4.18, we obtain the learning progress that $x_{i,t}$ has induced.

For sake of completeness, it is pointed out that in case of using PE as modeling error within the regional error calculation, all candidate solutions are evaluated on quality. This is needed for determining their PE and later on for re-training the model. In estimating the modeling error using DP or PV, only the selected promising candidate solutions are sampled, needed for re-training the model.

### 4.3.3.1 · Actual Learning Progress

Schmidhuber [Schmidhuber, 1991, Oudeyer and Kaplan, 2007, Oudeyer et al., 2007] follows the steps laid out above, but at time $t$ already re-trains the model, for each candidate solution $x_{i,t}$, thus generating $|X_t|$ temporary models to be able to express each candidate’s interestingness. The requirement of being one time step in the future rises from the fact that a new set of candidate solutions $X_{t+1}$ is to be used in evaluating the updated model. This is, however, circumvented by evaluating the updated model on the same test set again, namely the candidate solutions at time $t$. We term this measure actual learning progress (ALP),

$$
\text{ALP}(x_{i,t}) = \frac{\text{Err}(R(x_{i,t}), M(T_t), V_t = X_t) - \text{error in region } R(x_{i,t}) \text{ before training on } x_{i,t}}{\text{error in region } R(x_{i,t}) \text{ after training on } x_{i,t}}.
$$

In Figure 4.14, the solutions generated in selecting on $\text{ALP}_{1PE}$ are displayed, as well as the development of the $\text{ALP}_{1PE}$ scores. $1PE$ stands for singular regional error values based on PE. Single time steps are used, as ALP compares the error directly before and after training on a candidate, and PE is used because of higher reliability compared to estimating. In selecting on $\text{ALP}_{1PE}$, the search is able to move through the unlearnable area, and exploration does not get stuck after. The unlearnability of

---

3 It could be argued that only $|X_t| - 1$ models are temporary, as the model of the best-ranked candidate can be preserved, but this does not hold in case multiple candidates are selected to proceed to the next iteration, requiring training a new model on all of these.
Figure 4.14 Actual Learning Progress. The bottom-row plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on ALP\textsubscript{1PE}, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored). The top-row plots display the development of the ALP\textsubscript{1PE} scores, where the red dots represent the lastly-selected parent population. The search is able to traverse the noisy area and keeps exploring after.

the region with extreme noise is clearly reflected in negative ALP\textsubscript{1PE} scores, while the “sine”-areas get slightly positive scores. ALP is computationally costly, however, because of the need of training a temporary model for each candidate solution.

4.3.3.2 · Previous Learning Progress

Oudeyer et al. propose in [Oudeyer et al., 2007, Oudeyer and Kaplan, 2007] for each region $\mathcal{R}$ to adopt the learning progress $\text{LP}(\mathcal{R}, \theta_\mathcal{R})$ that was realized at the last time step $\theta_\mathcal{R}$ at which a solution was generated in that region. For the used regional error value definition, it holds that, at time $t$, for all regions $\theta_\mathcal{R}$ equals $t$.

The previous learning progress (PLP) measure indicates for a region whether it has been possible to better approximate the to-be-learned mapping there at time step $t$, and should thereby detect regions in which the surrogate model does not improve. For a candidate $x_{i,t}$, PLP is taken as the learning progress that was realized in the region $\mathcal{R}(x_{i,t})$ that it defines, at time $t$ (candidate $x_{i,t}$ itself induces learning progress at time


\( t + 1 \),

\[
\text{PLP}(x_{i,t}) = \text{LP}(\mathcal{R}(x_{i,t}), t) = \frac{\text{Err}(\mathcal{R}(x_{i,t}), \mathcal{M}(T_{t-\tau}), V_{t-\tau} = X_{t-\tau}) - \text{Err}(\mathcal{R}(x_{i,t}), \mathcal{M}(T_{t}), V_{t} = X_{t})}{\text{error in region } \mathcal{R}(x_{i,t}) \text{ at time } t-\tau}.
\]

Note that in combination with the used region definition, the discriminative power between candidate solutions \( x_{i,t} \) is limited to the difference in regional error values that arises from the *distance* between the defined regions \( \mathcal{R}(x_{i,t}) \).

In Figure 4.15, the solutions generated in selecting on \( \text{PLP}_{s\text{PE}} \) are displayed, the development of the \( \text{PLP}_{s\text{PE}} \) scores, and a comparison of all solutions generated using the different regional error averaging schemes. \( s\text{PE} \) stands for separated averages of regional errors, as in [Oudeyer et al., 2007], based on PE because of higher reliability compared to estimating the modeling error.

First of all, it can be seen from the plots of the \( \text{PLP}_{s\text{PE}} \) scores that all candidate solutions get negative values, as it is the selected best values that are displayed and these are all negative. This would make sense as we are in the unlearnable area, however, there is an adjacent peak of positive scores. From the plots of all generated solutions for the different averaging schemes, it can be seen that exploration is moving back-and-forth in the same linear direction, seemingly in pursuit of the positive peak.

Remember, PLP expresses the learning progress realized around a candidate solution based on solutions that were generated in that region at earlier time steps. However, in the used regional error calculation, the less-visited a region is, the closer to zero its regional error value will be! Thus, as is moved into a certain area, the regional error values will *increase* at first, leading to *negative* PLP! Only after the regional error values have reached a sufficient level of reliability, based on a sufficient number of test points in the region, PLP values become positive. This is exactly the positive peak that is seen in the wake of the exploration.

PLP requires a different regional error scheme with a high initialization of regional errors of unvisited regions. In [Oudeyer et al., 2007], this is taken care of through starting from a *single* region that gets recursively split as exploration continues, based on some splitting condition to be satisfied. They use a threshold of a maximum number
Figure 4.15 Previous Learning Progress. In (a) through (c), the plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on PLP\textsubscript{sPE}, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored); the top-row plots display the development of the PLP\textsubscript{sPE} scores, where the red dots represent the lastly-selected parent population. In (d) through (f), the solutions generated in selecting on PE-based PLP using different error averaging schemes are displayed. In all tested setups, selecting on PLP moves back-and-forth in a linear direction because of incompatibility with the used regional error calculation regarding unvisited areas.
of solutions or test points that may be assigned to a region. When met, the region is split in two based on some clustering technique that minimizes the distance between the solutions in both clusters.

4.3.3.3 · Previous Competence Change

Next to promoting high learning progress, which reflects that the model is capable of improving its mapping in a certain region, [Baranes and Oudeyer, 2013] suggest valuing negative learning progress, as this could point to deterioration in a region in which improvement was shown earlier. That region should be focused on again for verifying the fact of (not) being able to improve there. Previous competence change (PCC) is, like PLP, based on the most-recent learning progress shown in a region, but taking the absolute value,

\[
PCC(x_{i,t}) = |PLP(x_{i,t})| = \\
\frac{\left| \text{Err}(R(x_{i,t}), M(T_{t-\tau}), V_{t-\tau} = X_{t-\tau}) - \right|}{\text{error in region } R(x_{i,t}) \text{ at time } t-\tau} - \\
\frac{\text{Err}(R(x_{i,t}), M(T_{t}), V_{t} = X_{t}) |.}{\text{error in region } R(x_{i,t}) \text{ at time } t}
\] (4.24)

In Figure 4.16, the solutions generated in selecting on PCC\textsubscript{aPE} are displayed, the development of the PCC\textsubscript{aPE} scores, and a comparison of all solutions generated using different regional error averaging schemes. aPE stands for adjacent averages of regional errors, as in [Baranes and Oudeyer, 2013], based on PE because of higher reliability compared to estimating the modeling error. Through taking the absolute value, PCC omits some of the problems of PLP with the used regional error scheme. Instead of negative values, little-visited areas get positive values. However, the problem of the trailing positive peak persists, and at times exploration will be drawn back to where it was before. For two of the error averaging schemes, selecting on PCC can be seen mostly returning to previously visited areas, but for short-term/long-term memory more directed behavior is shown.
Figure 4.16 Previous Competence Change. In (a) through (c), the plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on PCC$_{aPE}$, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored); the top-row plots display the development of the PCC$_{aPE}$ scores, where the red dots represent the lastly-selected parent population. In (d) through (f), the solutions generated in selecting on PE-based PCC using different error averaging schemes are displayed. While for separated and adjacent averaging of regional errors, selecting on PCC mostly returns to previously visited areas, more directed exploration can be seen for short-term/long-term memory.
### 4.3.3.4 · Reducible Error

The approach remaining is valuing only negative PLP. [Luciw et al., 2011] aim for error maximization, like in selecting on learning-based novelty directly, but account for regions that cannot be modeled. They subtract the *irreducible* error, that is, the error in the region at an earlier time step, from the novelty score, resulting in the *reducible error* (RE),

\[
RE(x_{i,t}) = - PLP(x_{i,t}) = \underbrace{Err(R(x_{i,t}), M(T_{t}), V_{t} = X_{t})}_{\text{current error in } R(x_{i,t}), \text{ at time } t} - \underbrace{Err(R(x_{i,t}), M(T_{t-\tau}), V_{t-\tau} = X_{t-\tau})}_{\text{irreducible error in } R(x_{i,t}), \text{ at time } t-\tau}.
\] (4.25)

Regions with a high error in the past but a lower error now, *exactly* those that are promoted by PLP, get a *negative* reducible error! These regions have shown learning progress, meaning that the model was improved there *already*, and therefore the model is now expected to be improved more in less-visited regions.

The irreducible error has to be initialized to a low value for new regions that have not been visited yet for the principle to work, not surprisingly, opposite to the requirement for PLP. As stated before, in the used regional error calculation, this is accounted for as regions that have not been frequently visited yet get errors near zero. A region will then feature *high reducible error* upon first visit: High current error, and low past, *irreducible* error. In *unlearnable* regions, the current error will remain high, making that the past error becomes high as well after repeated visits, resulting in a *low reducible error*.

In Figure 4.17, the solutions generated in selecting on RE\textsuperscript{mPE} are displayed, the development of the RE\textsuperscript{mPE} scores, and a comparison of all solutions generated using different regional error averaging schemes. mPE stands for PE-based long-term/short-term memories of regional errors, as in [Luciw et al., 2011]. As aimed for, newly-visited regions and thereby new candidate solutions get high RE\textsuperscript{mPE} scores. Furthermore, the trailing peak of high scores reported in PLP and PCC has been inverted to negative. As such, in selecting on RE\textsuperscript{mPE}, exploration is able to move through the search space in a clearly directed fashion towards regions not visited recently. It should be noted though that this only holds for the memories scheme.
Figure 4.17 Reducible Error — PE. In (a) through (c), the plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on $R_{mPE}$, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored); the top-row plots display the development of the $R_{mPE}$ scores, where the red dots represent the lastly-selected parent population. In (d) through (f), the solutions generated in selecting on PE-based RE using different error averaging schemes are displayed. In selecting on RE, exploration is able to move in a clearly directed pattern towards unvisited areas. This is, however, only the case in using short-term/long-term memories for regional error averaging.
Figure 4.18 Reducible Error — PV. In (a) through (c), the plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on RE<sub>mPV</sub>, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored); the top-row plots display the development of the RE<sub>mPV</sub> scores, where the red dots represent the lastly-selected parent population. In (d) through (f), the solutions generated in selecting on PV-based RE using different error averaging schemes are displayed. In selecting on PV-based RE, exploration is not able to match the performance of PE-based RE.
Figure 4.19 Reducible Error — DP. In (a) through (c), the plots show the solutions generated by a single run on test problem SineUnlearnable, of a (3,6)-CMA-ES selecting on $\text{RE}_{\text{mDP}}$, at three time steps (recently-generated solutions are dark-colored, early-generated solutions are bright-colored); the top-row plots display the development of the $\text{RE}_{\text{mDP}}$ scores, where the red dots represent the lastly-selected parent population. In (d) through (f), the solutions generated in selecting on DP-based RE using different error averaging schemes are displayed. While using only half of the quality evaluations for model training compared to PE-based RE, DP-based RE is able to deliver the same performance, for error averaging based on short-term/long-term memory.
Motivated by the possibility of matching this performance at lower cost of quality function evaluations, RE is tested with estimating the modeling errors via PV and DP. These results are displayed in Figure 4.18 and 4.19. In selecting on PV-based RE, exploration is not able to induce a similarly-clear pattern of directed behavior, even in using the memories scheme. As analyzed before, PV-based ranking is hindered through the model-wide stationarity, which makes the ranking purely density-based. Moreover, it is likely further hindered by the non-stationarity between different models in the used horizontal ensemble.

DP-based RE, conversely, is able to reproduce the PE-based behavior for the memories scheme, with no concessions to performance. Importantly, calculating DP is free of quality evaluations, but moreover, in DP, only half of the solutions are used for model training compared to in PE. Only the selected 3 parent individuals are evaluated and included in model training versus all 6 offspring. This makes RE_{mDP} the most-promising learning-based method, which is summarized in Figure 4.20.

4.4 · Summary

In this chapter, in search for exploration criteria for driving design-space exploration, is turned to work describing the use of novelty and interestingness measures for exploration in active learning, in developmental robotics. At the heart of these measures
<table>
<thead>
<tr>
<th>Exploration Measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVE ((V(T)))</td>
<td>Learning-based Novelty</td>
</tr>
<tr>
<td>(d_T(x))</td>
<td>Distance to reference solutions in an archive</td>
</tr>
<tr>
<td>(\sum_{i=1}^{n} \frac{d_T(x_i)}{D(x)})</td>
<td>Similar distance to solution (b) from archive, (d_T)</td>
</tr>
<tr>
<td>(\frac{\sum_{i=1}^{n} d_T(x_i)}{n})</td>
<td>Average distance to closest solutions, (d_T)</td>
</tr>
<tr>
<td>(D_{\max}(x))</td>
<td>Maximum distance to a solution from archive, (D)</td>
</tr>
</tbody>
</table>

**Table 4.1: Taxonomy of Exploration Measures**

Overview of the exploration measures for adaptive search that are described in this chapter. All exploration measures are to be maximized, i.e., a solution with the highest exploration-measure outcome gets the best ranking.
is a reference set of solutions that is updated during the search, making it possible to induce a sequential pattern in which exploration is first drawn to certain areas and then repelled again. Distance-based novelty compares candidate solutions to the solutions contained in the reference set using a domain-specific distance measure, whereas learning-based novelty expresses the error that a model generated based on the reference set makes in its approximation for a candidate solution.

The goal of the exploration is finding a diverse set of solutions with respect to the same domain-specific distance measure that is used in distance-based novelty. As such, the pattern of exploration induced by selecting on distance-based novelty will automatically constitute a diverse set. The optimal set of solutions for generating an accurate model of the search space is likely to, implicitly, maximize diversity as well. Learning-based novelty as direct predictor of a model’s gain of represented knowledge, however, is not sufficient. For inducing a pattern of exploration that maximizes this learning progress, regions in which the model does not improve are to be detected to prevent exploration of getting stuck in these areas. This is accounted for in interestingness, which derives from the development of learning-based novelty in a region over time.

This chapter identified two promising exploration criteria, out of nine examined, see Table 4.1, to be integrated into quality-based optimization. Uniqueness, a distance-based novelty expression, preserves the explorative behavior of the original sparseness expression but with less parameters involved, and leads to quick, oscillating behavior that generates a diverse set of solutions. Interestingness expression reducible error, based on short-term/long-term memories of dispersion-in-predictions estimations of learning-based novelty, matches the performance of its counterpart based on prediction-error learning-based novelty. It leads to directed behavior that generates a clear path through the search space, while managing to do so at half of the cost of the quality function evaluations used for model training compared to the prediction-error-based variant.