Regularization Approach to Inductive Genetic Programming

Nikolay Y. Nikolaev                     Hitoshi Iba
Dept. of Math. and Computing Sciences    Dept. of Inf. and Comm. Engineering
Goldsmiths College                      School of Engineering
University of London                    The University of Tokyo
New Cross                                7-3-1 Hongo, Bunkyo-ku
London SE14 6NW                          Tokyo 113-8656
United Kingdom                          Japan
nikolaev@mcs.gold.ac.uk                 iba@miv.t.u-tokyo.ac.jp

Abstract

This paper presents an approach to regularization of inductive genetic programming tuned for learning polynomials. The objective is to achieve optimal evolutionary performance when searching high-order multivariate polynomials represented as tree structures. We show how to improve the genetic programming of polynomials by balancing its statistical bias with its variance. Bias reduction is achieved by employing a set of basis polynomials in the tree nodes for better agreement with the examples. Since this often leads to overfitting, such tendencies are counteracted by decreasing the variance through regularization of the fitness function. We demonstrate that this balance facilitates the search as well as enables discovery of parsimonious, accurate, and predictive polynomials. The presented experimental results show that this regularization approach outperforms traditional genetic programming on benchmark data mining and practical time-series prediction tasks.

Keywords: Genetic Programming, Kolmogorov-Gabor polynomials, regularization, time series prediction.
1 Introduction

Inductive genetic programming (iGP) is considered a specialization of the original genetic programming (GP) paradigm [2], [14], [22] for inductive learning from examples. The reasons for using this specialized term are: 1) inductive learning is a search problem and GP is a versatile framework for exploring of large search spaces; 2) GP provides genetic learning operators that can be tailored to the particular task; and 3) GP flexibly reformulates program-like solutions that satisfy the constraints of the task adaptively. An advantage of iGP is that it discovers the structure and size of the solutions. The main components of a contemporary iGP system are: proportional selection, size-dependant fitness function, and size-dependant application of crossover and mutation operators.

Previous research points out that GP is successful for inductive learning tasks such as: Boolean concept formation [22], classification [42], data mining [7], [33], pattern recognition [35], image processing [34], [36], chaotic time-series prediction [1], [16], [24], [30], [33], [35], [48], [49], financial engineering [6], [23], etc. Most of these GP systems aim to discover nonlinear model descriptions of the regularities in the given data. This observation motivates us to direct attention to Kolmogorov-Gabor polynomials as general nonlinear models. Attempts for evolutionary induction of such polynomials represented according to the GMDH method [17], [18], [26], [29] already showed promising results on learning tasks [14], [15], [16], [21], [33].

The evolutionary systems from the STROGANOFF family [14], [15], [16], [33] search for high-order multivariate Kolmogorov-Gabor polynomials composed from computationally tractable low-order basis polynomials, which are allocated at the nodes of binary tree structures. During experiments with real data it has been detected that STROGANOFF suffers from several disadvantages: 1) it usually produces overfit polynomials that do not perform well when tested on unseen data [33], as it is quite sensitive to the noise in the data and to the sample size; 2) it often discovers oversized polynomials, despite the use of minimum description length (MDL) fitness functions [15] for imposing a pressure toward favoring concise solutions because the tree growth normally decreases the error; and 3) it is affected by multicollinearities in the data matrix due to the employment of an ordinary least squares fitting technique for estimating the coefficients.

This paper enhances STROGANOFF in order to overcome the above disadvantages. The objective is to achieve optimal search performance that locates not only parsimonious and accurate but also highly predictive solutions. We show how to improve the performance by balancing the statistical bias and variance of iGP. Statistical bias is the restriction that the basis polynomials, with which iGP constructs the target polynomials,
impose on the solution quality. Statistical variance is the deviation of the iGP learning efficacy from one data sample to another sample that could be described by the same target polynomial. STROGANOFF is enhanced in two ways: 1) a set of basis polynomials is designed to reduce the statistical bias and fit more flexibly the examples; and 2) a regularization technique is applied to the error component of the fitness function to reduce the variance, in the sense of a tendency to overfit the particularities and noise in the examples. Regularization is a manner for increasing the extrapolation capacity of the polynomials. We elaborate a regularized minimum description length (MDL) fitness function according to a method for polynomial synthesis [3].

The effect of extending the basis set is improving of the local search, which we analyze with the autocorrelation function [27]. The regularization in iGP has a primary effect of increasing the generalization and a secondary effect of facilitating the global search. We investigate the primary effect of increasing the generalization due to the regularization with several time-series prediction tasks using a variance measure [45], and with benchmark data mining tasks using a cross-validation technique [47]. The secondary effect of facilitating the global search is due to smoothing of the landscape by regularizing of the fitness function. The fitness landscape is a metaphor for studying the search space usually associated with the triple: an underlying set of trees; a fitness function that assigns a value to each tree; and a neighborhood relation specified by the genetic learning operator. The MDL fitness functions smooth the landscape by discarding complex solutions. Using the fitness-distance correlation measure [20] we show that the regularization contributes additionally for global landscape smoothing by preferring the simple polynomials having low magnitude coefficients.

The evolutionary search depends on the structure of the fitness landscape, as well as on the search navigation guided by genetic selection and the learning operators. An enhanced version of STROGANOFF is implemented using the regularized MDL fitness function referred to later here as reSTROGANOFF. The iGP mechanisms incorporated in the original system and its enhanced version include fitness-proportional selection and the application of crossover and mutation operators that depend on the tree size [32]. Mutation and crossover points are chosen with recombinative guidance by the largest residual error computed in the internal tree nodes [16]. Empirical evidence show that by using these same mechanisms the enhanced reSTROGANOFF system outperforms STROGANOFF on the time-series prediction and data mining tasks.

This paper is organized as follows. The next section defines the inductive learning problem to be addressed by the two STROGANOFF systems. It includes the scheme for tree-like polynomial construction from a set of
basis polynomials and the genetic mutation and crossover learning operators. Section three motivates the need for regularization in iGP and designs the regularized MDL fitness formula. It provides a formal derivation of the smoothing regularizer for the adopted set of basis polynomials. The local and global search performance of STROGANOFF and its enhanced regularized version reSTROGANOFF is investigated in section four. The applicability of the proposed regularization approach to practical time-series prediction and data mining tasks is demonstrated empirically and related to other GP systems in section five. Finally, a discussion ensues and conclusions are derived.

2 Inductive Learning and Regression

Instances of the inductive learning problem are of particular interest in contemporary computer science, as they arise in various real-world applications, including: classification, pattern recognition, data mining, time-series processing, financial engineering, and image processing.

The inductive learning problem can be formulated as a multivariate regression problem. Given example instantiated vectorss $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ of several independent variables, that is, pattern vectors $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{il}) \in \mathbb{R}^l$, and corresponding values of the dependent variable $y_i \in \mathbb{R}$, the goal is to find function models $y = f(\mathbf{x})$. A common assumption is that the examples are drawn independently from a certain probability distribution. The solution to the problem is the regression function $\hat{f}(\mathbf{x})$ that maps given $\mathbf{x}$ to the conditional mean $E[y|\mathbf{x}]$ with respect to the underlying probability distribution.

Since this assumption has a somewhat unknown nature, and the examples are often noisy in practice, the goal is reformulated. In practice the goal is to find a reasonably best approximation $f(\mathbf{x})$ of the regression $\hat{f}(\mathbf{x})$ of $y$ on $\mathbf{x}$ by minimizing the empirical error on the examples. Traditionally the Gaussian distribution is considered, and the least squares fitting criterion is used to search for the function $f(\mathbf{x})$ that minimizes the average squared residual (ASR):

$$ ASR = \frac{1}{N} \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2 $$

(1)

where $y_i$ is the true outcome of the $i$-th example, $f(\mathbf{x}_i)$ is the outcome estimated with the $i$-th input vector $\mathbf{x}_i$ in the same example, and $N$ is the sample size.
2.1 Tree-like Polynomials for GP

This paper focuses on inductive learning of polynomial function models by evolutionary search organized with the GP paradigm. Polynomials are represented as hierarchical compositions of basis functions allocated in the nodes of tree structures, borrowing ideas from the group method of data handling (GMDH) [17], [18], [26], [29]. The trees serve to cascade low-order basis polynomials allocated in the tree nodes and independent variables in the tree leaves. The basis polynomial outcomes from children nodes are fed forward to their parent nodes, where partial models are composed of received outcomes from the basis polynomials and/or independent variables passed by the children nodes. At the tree root, the output is interpreted as a high-order, high-dimensional Kolmogorov-Gabor polynomial:

\[ f(x) = a_0 + \sum_i a_i x_i + \sum_{i<j} a_{ij} x_i x_j + \sum_{i<j<k} a_{ijk} x_i x_j x_k + \ldots \]  

where \( a_i \) are the polynomial coefficients, and \( x_i \) are the independent variables.

The Kolmogorov-Gabor polynomial is a universal format for function modeling because with it one may approximate any continuous function mapping to an arbitrary precision, in an average squared residual sense, if there is a sufficient number of terms [8].

2.2 Evolving Polynomials with STROGANOFF

The appropriateness of tree-structured polynomials for GP-style processing was initiated with the system STROGANOFF (STructured Representation On Genetic Algorithms for NOnt-linear Function Fitting) [14], [15], [16]. It conducts an evolutionary search with a population of trees using the micromechanisms of a genetic algorithm: fitness evaluation, proportional selection, crossover and mutation operators. The search space is sampled by applications of the crossover and mutation operators, which transform probabilistically chosen trees from the population having above-average fitnesses. The fitness evaluation of a tree-like polynomial involves: 1) estimating the basis polynomial coefficients in the tree nodes with a least mean square technique from the multiple regression analysis, and 2) calculating the data fitting error at the tree root.

(Figure 1 here)

A tree-like STROGANOFF polynomial is a composition of second-order bivariate basis polynomials in the nodes and independent variables in the leaves (Figure 1). The bivariate polynomials with vector variables
\( \mathbf{x} = (x_1, x_2) \) require binary trees. The basis polynomials are:

\[
f(\mathbf{x}) = a_0 h_0(\mathbf{x}) + a_1 h_1(\mathbf{x}) + \ldots + a_5 h_5(\mathbf{x})
\]

(3)

where \( a_i \neq 0, \ 0 \leq i \leq 5 \), and the simple functions \( h_i(\mathbf{x}) \) that produce the terms are: \( h_0(\mathbf{x}) = 1 \), \( h_1(\mathbf{x}) = x_1 \), \( h_2(\mathbf{x}) = x_2 \), \( h_3(\mathbf{x}) = x_1x_2 \), \( h_4(\mathbf{x}) = x_1^2 \), and \( h_5(\mathbf{x}) = x_2^2 \).

The target \( f(\mathbf{x}) \) is built by bottom-up tree traversal and composing the basis polynomials in the nodes. The basis polynomial \( f_1 \) at a node, for example at node 1 in Figure 1, is applied \( f_1(y_3, y_4) \) with the outcomes from the polynomials \( y_3 \) and \( y_4 \) supplied by the children nodes, \( y_3 = f_3(\mathbf{x}) \) from node 3 and \( y_4 = f_4(x_8, x_9) \) from node 4. In case of a node like node 3 with a polynomial child like node 2: \( y_2 = f_2(x_9, x_7) \), and an independent variable \( x_5 \) as a second child, the application \( f_3(f_2(x_9, x_7), x_5) \) yields outcome\(^1\):

\[
y_3 = f_3(y_2, x_5)
\]

(4)

The computed target polynomial in this case is the composition:

\[
f(\mathbf{x}) = f_1(f_3(f_2(x_9, x_7), x_5), f_4(x_8, x_9))
\]

(5)

These binary tree-like polynomials used in the system STROGANOFF \cite{14}, \cite{15}, \cite{16} are more efficient than those in GMDH \cite{17}, \cite{18}, \cite{26}, \cite{29} since they feature a reduced representation complexity.

The problem of learning tree-like polynomials concerns three main issues: 1) how to find the polynomial coefficients in the tree nodes, 2) which polynomial terms with which independent variables to select, and 3) how to avoid overfitting the given examples.

(Figure 2 here)

Since a polynomial is a combination of monomials of degree up to 2 in the independent variables, the polynomial term coefficients at each node of the tree structure are calculated with the following matrix equation:

\[
\mathbf{a} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y}
\]

(6)

where \( \mathbf{a} \) is the vector of coefficients \( \mathbf{a} = (a_0, a_1, \ldots, a_5) \), \( \mathbf{H} \) is a \( N \times 6 \) design matrix of row vectors \( \mathbf{h}(\mathbf{x}_i) = (h_0(\mathbf{x}_i), h_1(\mathbf{x}_i), \ldots, h_5(\mathbf{x}_i)), i = 1..N \), and \( \mathbf{y} \) is a \( N \times 1 \) output vector. Equation (6) defines the normal equations of the ordinary least-squares (LS) fitting method which has a unique solution, that is a best fitting polynomial

\(^1\text{Strictly speaking, the function } f \text{ depends on its arguments, and each occurrence of } f \text{ should be specialized. In order to simplify the notation when we write, for example: } y_j = f(y_k, x), \text{ we mean that it should be considered as: } y_j = f_{y_k,x}(y_k, x). \)
\[ f(x) = Ha \] which is unique at the given data. Since the covariance matrix \( H^T H \) is symmetric, Cholesky decomposition \([37]\) is used for its fast inversion.

The main advantage of STROGANOFF over traditional GP systems for function approximation \([2], [22], [42]\) is that STROGANOFF attempts to discover the proper tree-like structure of the polynomial models and avoids the necessity to conduct search for finding their numeric coefficients.

### 2.3 Enhancement with a Set of Polynomials

The developments into GMDH showed that its performance is improved with the use of several basis polynomials in the nodes instead of using only the complete bivariate polynomial \([26], [29]\). We continue this idea further and enhance the system STROGANOFF with all distinctive complete and incomplete (first-order and second-order) bivariate polynomials. The motivations for using them as basis polynomials are: 1) when composed higher-order basis polynomials rapidly increase the order of the target polynomial, which causes overfitting even with small trees, 2) first-order and second-order basis polynomials are fast to process, and 3) they define a search space of reasonable dimensionality for the GP to explore. The extended basis set serves to make STROGANOFF fit the data more flexibly compared with the use of the complete bivariate polynomial (3) only, and to enable more precise identification of the interactions between the variables.

The incomplete basis polynomials are derived from the complete bivariate polynomial (3) assuming that some of its term coefficients are zero. The total number of such incomplete polynomials is 25 from all \( 2^5 - 1 \) possible combinations of monomials \( a_i h_i(x), 1 \leq i \leq 5 \), and always the leading constant term \( a_0 \), that contain two different independent variables. We use a subset \( \Phi = \{16\} \) out of them after elimination of the symmetric polynomials (see Appendix). This subset \( f_j \in \Phi, 1 \leq j \leq 16 \) includes the complete first-order and the complete second-order basis polynomials. An example tree-structured polynomial using some of these basis polynomials is illustrated in Figure 2.

### 2.4 Genetic Learning Operators

The second issue in learning nonlinear multivariate trees, suggested in subsection II.B, concerns the iGP search for polynomial terms to enter the model. The iGP system STROGANOFF performs evolutionary search using mutation and crossover operators that are applied with probabilities which are proportional to the size of the
tree-like polynomial representations. The rationale behind the development of size-dependant operators is to guard against a degenerated search when used in combination with minimizing fitness functions.

The mutation operator is context preserving [32]. It transforms a tree structure with three elementary submutations: 1) substitution of a chosen node by another randomly generated one, 2) insertion of a random node as a parent of a subtree so that the subtree becomes the leftmost child of the new node and the right child is a randomly generated terminal leaf, and 3) deletion of a chosen node only when no subtree below is to be cut. This mutation is applied with probability $p_m = m, |g|^2$ [11], where $m$ is a free parameter and $|g|$ is the tree size.

The crossover operator splices two trees with probability $p_c = c/\sqrt{|g|}$ [32], where $c$ is a free parameter, or swaps them. Thus, the offspring trees become of larger size when their parent trees are relatively short. When a tree is of short size, it is not cut but simply added as a whole subtree at the crossover point in the other tree. If the two trees are short they are spliced together: the first with the second, and the second with the first. Overall, this crossover by cut and splice contributes to a better global search by preventing the trees from rapid shrinking in the case of minimizing fitness functions. The proper values for the free parameters $m$ and $c$ are found with the autocorrelation function (see subsubsection IV.B.1).

2.5 Recombinative Guidance

The convergence of STROGANOFF is accelerated by recombinative guidance [16]. The recombinative guidance suggests to select as a cut/mutation point the tree node that shows the largest residual error $ASR$. This improves the local search ability of the iGP system because it helps for exploiting promising zones in the vicinity of the parent tree. The usefulness of such recombinative guidance is strengthened here showing that the error component $(y - Ha)^T (y - Ha)$ of the residual error $ASR$ (1) can be computed efficiently as follows:

$$ASR = \frac{1}{N}(y^T y - a^T H^T y)$$

(7)

where $y$ is the outcome vector, $a$ is the $A \times 1$ coefficient vector, and $N$ is the number of the data. Formula (7) takes only $A$ multiplications because the quantity $y^T y$ is constant for the task, and $H^T y$ is ready from the process of estimating the coefficients by (6).
3 Regularization of the iGP

The third issue in learning tree-like polynomials, raised in subsection II.B, is avoiding overfitting. The problem is to find a polynomial with terms and coefficients such that it approximates the examples optimally. In practice, one is interested in adequate approximation by polynomials having a finite number of terms, rather than with a large number of terms that can represent the unknown function exactly. That is why our goal is the design of a fitness function with which the iGP search can be controlled so as to discover parsimonious, accurate, and predictive solutions. Aiming at optimal search guidance, we rely on an information-theoretic principle and on regularization theory to favor polynomials with such qualities.

3.1 The MDL-based Fitness Function

Criteria for learning accurate and parsimonious models are given by the minimum description length (MDL) principle [39]. Adapted for the purpose of inducing polynomials, the MDL principle can be stated as follows: given a set of examples and an effective enumeration of their polynomial models, prefer with greatest confidence the polynomial that has both high learning accuracy (i.e., very closely approximates the examples), and low structural complexity (i.e., has a small number of terms or coefficients). Such MDL functions are suitable for iGP as they prevent the trees from rapid growth, which causes premature convergence to inferior solutions.

We adopt a concrete MDL-based fitness function with a learning accuracy component proportional to the empirical error over the available training examples, and a structural complexity component proportional to the number of polynomial coefficients [3]:

$$MDL = ASR + \frac{A}{N} s^2 \log(N)$$

(8)

where $ASR$ is the average squared residual (1), $A$ is the number of polynomial term coefficients, $N$ is the number of the examples, and $s^2$ is a rough estimate of the unknown error variance. An estimate of the error variance can be derived using the deviation of the dependent variable values in the provided examples:

$$s^2 = \frac{1}{N} \sum_{i=1}^{N} (y_i - \overline{y})^2$$

(9)

where $\overline{y}$ is the sample mean of the dependent variable $\overline{y} = (1/N) \sum_{i=1}^{N} y_i$. This empirical error variance $s^2$ is practical since it does not depend on the model [3].
Although the MDL fitness function (8) may help to evolve concise polynomials, it does not guarantee abilities to learn polynomials that will predict well. One of the reasons is that its ASR component depends on the particularities of the examples, which often leads to overfitting.

3.2 Statistical Bias and Variance

Criteria for avoiding overfitting with the data and increasing the predictability through balancing the statistical bias and variance when learning function models are given by regularization theory.

When learning from a fixed and finite example set, the ASR error may be low but this is not enough for a high generalization since the examples are not reliable due to noise and uncertainties. Such difficulties may be addressed by decomposing the error of the solution $f(x)$, being an approximation of the true function $\hat{f}(x) = E[y|x]$ with respect to the training data $D$, in two components [10]:

$$E_D[(f(x) - E[y|x])^2] = (E_D[f(x)] - E[y|x])^2 + \left(\frac{1}{n_D}\sum_{x \in D} (f(x) - E_D[f(x)])^2\right)$$

$$= BIAS^2(f(x)) + VAR(f(x))$$

(10)

where $BIAS(f(x)) = E_D[f(x)] - E[y|x]$ is statistical bias, and $VAR(f(x)) = E_D[(f(x) - E_D[f(x)])^2]$ is statistical variance.

The statistical bias accounts only for the degree of fitting the given training data $D$, but not for the level of generalization. This is the statistical variance that accounts for the generalization of whether or not the polynomials fit the examples without regard to the specificities of the provided data. When high-order polynomial models are evolved the statistical bias is reduced, but the variance increases. A way to decrease the statistical variance is to consider polynomial curvature-driven smoothing implemented by regularization of the fitness function.

3.3 Smoother Polynomials

Viewing evolutionary inductive learning as approximation, our goal is to identify a polynomial surface that passes through the space of example points. Polynomials that define less varying surfaces are desirable since they do
not tightly fit the example data points. We call them smoother polynomials. Concise smoother polynomials are good approximators for two main reasons [37]: 1) usually there are a large number of examples, and we seek a polynomial with a small number of coefficients that describes them sufficiently well; 2) the given examples are typically noisy and sparse in some regions of the example space, which introduces undulations in the polynomial surface.

We address the first reason with the employment of the MDL fitness function (8). Concerning the second reason, there are no effective criteria to judge as to what degree the examples are correct and to what degree they are corrupted by noise. Therefore, one needs some a priori criteria to combat noisy data. Such a priori criteria are given by variance functionals that enforce the smoothness requirement. A proper variance functional should assign lower values to smoother polynomials. Variance in learning polynomials may be considered as sensitivity to slight changes in the coefficients. This could be explained with the curvature of the polynomial surface. The surface curvature is a property of a polynomial, which depends on the rates of change of the derivatives of its independent variables.

The choice of a proper smoothing factor depends on the concrete task, and on the algorithm with which one decides to address it. The curvature of a polynomial surface $f(x)$ may be estimated with the following simple variance functional $V[f]$ [43], [44]:

$$V[f] = \int \left| \frac{\partial^2 f(x)}{\partial x^2} \right|^2 dx$$  \hspace{1cm} (11)

The polynomial $f(x)$ is required to have continuous first and second derivatives, i.e., $f(x)$ has to be a twice-differentiable function mapping. A large second derivative indicates a high curvature. This quadratic functional diminishes the influence of the low-order term coefficients, since they do not affect the bending of the surface, and amplifies the influence of the coefficients over the high-order terms in the polynomial.

When smoothing polynomials composed of predefined bases, the integral may be replaced by summation over the set of available examples [3]. In our case of a bivariate quadratic basis polynomial, from which the basis polynomials are derived, the functional $V[f]$ is:

$$V[f] = \sum_{i=1}^{N} \left( \frac{\partial^2 f(x_i)}{\partial x_{1i}^2} + 2 \frac{\partial^2 f(x_i)}{\partial x_{1i} \partial x_{2i}} + \frac{\partial^2 f(x_i)}{\partial x_{2i}^2} \right)^2$$  \hspace{1cm} (12)

where $\partial^2 f(x_i)/\partial x_{1i}^2$, $\partial^2 f(x_i)/\partial x_{2i}^2$ are the instantiated second partial derivatives of $f(x)$ with respect to the independent variables $x_1$ and $x_2$, and $\partial^2 f(x_i)/\partial x_{1i} \partial x_{2i}$ is the instantiated cross derivative of $f(x)$ with respect
to the same independent variables $x_1$ and $x_2$. The partial derivatives are instantiated with the corresponding values of $x_i$ from the $i$-th example.

Applying this functional to the basis polynomials from the set $\Phi$, we find that these derivatives, as well as the cross derivative, are constants that do not depend on the independent variables in the vector $x_i$. Then the variance functional $V[f]$ for the basis polynomials in $\Phi$ reduces to:

$$V[f] = \left( \frac{\partial^2 f(x)}{\partial x^2} \right)^2 = \left( \frac{\partial^2 f(x)}{\partial x_1^2} + 2 \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} + \frac{\partial^2 f(x)}{\partial x_2^2} \right)^2$$  \hspace{1cm} (13)

Substituting the complete basis polynomial $f_1$ in the variance functional $V[f]$, if we suppose that the index of the complete polynomial in $\Phi$ is 1, we obtain:

$$V[f_1] = (2a_4 + 2a_3 + 2a_5)^2$$  \hspace{1cm} (14)

Corresponding variance functionals are calculated in the same way for all remaining basis functions $f_j \in \Phi$, $2 \leq j \leq 16$, i.e., complete and incomplete bivariate quadratic basis polynomials from the set $\Phi$.

(Figure 3 here)

3.3.1 Regularized Average Error

The risk of overfitting the examples could be minimized if the variance functional is used in the fitness function to penalize polynomials with high curvatures. For this reason we accommodate the functional $V[f]$ in a regularized average error RAE as a fitness criterion for encouraging polynomial models with low-magnitude high-order term coefficients:

$$RAE = \frac{1}{N} \left( \sum_{i=1}^{N} (y_i - f(x_i))^2 + k \left| \frac{\partial^2 f(x)}{\partial x^2} \right|^2 \right)$$  \hspace{1cm} (15)

where $k$ is a regularization parameter. The regularization is a roughness penalty since small high-order term coefficients imply a more “regular” approximation. A choice of $k = 0$ favors polynomial surfaces interpolating the example points tightly, while a large $k \rightarrow +\infty$ favors flat surfaces.

Tuning the regularization parameter $k$ favors polynomials that compromise between the goodness of fit and smoothness of the surface. Figure 3 shows approximations of a curve by three different polynomials. The smoothest polynomial, showed with a bold solid line, is computed using a regularization parameter and appears closer to the original function than its recalculated version using least-squares fitting shown with a dashed line, and the polynomial of coefficients estimated directly by ordinary least-squares fitting shown with a dotted line.
The errors of the same smoothest polynomial when changing the values of the parameter \( k \) are given in Figure 8.

Similar regularization techniques have been applied successfully to neural networks [4], [45]. The advantage of the regularization functional \( V[f] = |\partial^2 f(x)/\partial x^2|^2 \) is that it is fast to compute. This is because our basis polynomials \( f_j \in \Phi, 1 \leq j \leq 16 \) are fixed, which means that their functionals \( V[f] \) can be programmed in advance. There remains only to instantiate these functionals in run time with concrete coefficient values, to square the results, and to multiply them by the regularization parameter \( k \). Since the largest basis polynomial (3) has six coefficients, three of which participate in its functional \( V[f_1] \) (14), there will be up to four necessary multiplications plus two summations, and one squaring.

We use the regularized error \( RAE \) in the \( MDL \) fitness function instead of using the residual error \( ASR \) component. The synthesized novel fitness function we will denote from now on \( MDL_R \):

\[
MDL_R = RAE + \frac{A}{N} \sigma^2 \log(N) \tag{16}
\]

Another formula for calculation of the coefficients, instead of (6), is derived by solving the minimization problem \( \sum_{i=1}^{N} (y_i - f(x_i))^2 + kV[f] \) with respect to the coefficients \( a_j, 1 \leq j \leq A \), assuming the least squares fitting criterion. After applying calculus, one finds the matrix formula:

\[
a = (H^T H + kI)^{-1} H^T y \tag{17}
\]

where \( a \) is the vector of coefficients \( a = (a_0, a_1, ..., a_5) \), \( I \) is the square \( 6 \times 6 \) identity matrix. It is important to note that this augmentation of the matrix \( H^T H \) with \( kI \) also helps to avoid ill-effects of multicollinearity on ordinary least squares fitting [12], and makes the approach practically appealing. Even small changes in the elements of the design matrix \( H \) may cause large changes in the inverse matrix \( (H^T H)^{-1} \), which will influence essentially the numeric computation of the coefficients.

Adding the term \( kI \) may be viewed as training with artificial noise on the examples. The open questions include how much noise should be tolerated and from which range \( k \) may take values?

### 3.3.2 The Regularization Parameter

The choice of proper values for the regularization parameter \( k \) is subtle since it determines the degree of fitting of the examples and governs the amount of smoothing. Having in mind that the examples are usually noisy, a fitness function with too small \( k \) will tend to overfit the data, hence the noise. This will lead to wildly undulating
surfaces that are not likely to make good predictions. At the other extreme, very large values of $k$ will produce quite smooth surfaces that underfit the examples, and does not also give good predictions.

Proper values for the regularization parameter $k$ may be selected from a certain interval relying on a proof that as long as $k$ lies within this interval:

$$0 < k < 2\sigma^2/\mathbf{a}^T \mathbf{a}$$  \hspace{1cm} (18)

the mean squared error of the RAE approximator is smaller than that of the best least squares approximator without regularization \[12\]. As an unbiased estimator of $\sigma^2$ the average squared residual error $ASR$ over the training examples can be used.

We propose to find values for $k$, needed in (18), with the following algorithm, which should be done once for a particular data set of examples:

1) pick initially a fit polynomial with coefficients calculated by ordinary least squares fitting, for example pick the best polynomial model from a large number of runs with the iGP system, e.g., 100 polynomials, on the Mackey-Glass series given in Table 2.b;

2) compute the quantity $2ASR/\mathbf{a}^T \mathbf{a}$, and assign a step size of $\Delta k = (2ASR/\mathbf{a}^T \mathbf{a})/10$;

3) using $k_1 = 0$ reevaluate the chosen polynomial in 1) and estimate its $ASR_1$;

4) repeat step 3 with $k_i = \theta_{i-1} + \Delta k$, for a number of iterations $2 \leq i < 10$;

5) finally, select the $k_i$ which yields the medium residual $ASR_1 \leq ASR_i \leq ASR_{10}$.

The rationale for step 5 relies on empirical observations, which demonstrate that the average value of $k_i$ is the most suitable compromise (Figure 8).

According to this scheme, we conducted experiments from $k_1 = 0.00025$ which causes residual $ASR_1 = 0.00479$, on the first 100 points from the Mackey-Glass series \[25\], to $k_{10} = 0.005$ which causes $ASR_{10} = 0.00488$. We selected the value $k = 0.001$ leading to a residual $ASR = 0.00481$ somewhere between $ASR_1$ and $ASR_{10}$ (see Figure 8).

4 Search Performance

Stated more precisely, the proposed regularization approach prescribes how to design a fitness function for iGP with respect to the employed basis polynomials so as to attain optimal solutions. The ultimate fitness function design goal is to facilitate the discovery of accurate, short size and well predicting polyno-
mial models. We examine the performance of ordinary STROGANOFF and its regularized enhanced version reSTROGANOFF. The intention is to find out whether the regularization approach improves the previous ordinary system STROGANOFF.

First, we examine whether or not the use of a basis polynomial set can facilitate the iGP search. The evolutionary iGP search is viewed as navigation, by the fitness function and a concrete genetic operator, and structure, determined by the fitness landscape of the same genetic operator. A correlation analysis of the global and local characteristics of the fitness landscape of the mutation operator is made in subsection IV.B to gain insight into how the landscape structure impacts the search difficulties.

Second, we examine how guidance by the regularized fitness function influences the iGP search process. Experiments are conducted to reveal whether or not the learning accuracy and the generalization of the best polynomials found using regularization are better. The results are given in subsection IV.C.

The parameters of the two iGP systems used in all experiments are given in Table 1. The values of the regularization parameter $k$ are application dependent and calculated with the above five step algorithm (subsection III.C.2) separately for each task.

(Table 1 here)

4.1 The Task: Mackey-Glass Equation Series

Time-series prediction may be regarded as an inductive learning problem. The task is to identify the regularities among a series of points: ..., $x^t$, $x^{t+1}$, $x^{t+2}$, ..., sampled at discrete time intervals: ..., $t$, $t+1$, $t+2$, .... The goal is to discover how future points depend on previous, historic points in the series. This is accomplished by searching for a series description with the most relevant from the available points. We assume an autoregressive model with high-order multivariate polynomial models (linear in the coefficients and nonlinear in the independent variables), represented as binary trees. Vectors with the independent variable $x_i$ are created from the available points with embedding dimension $d$, and delay time $\tau$:

$$x_i = (x^{t-(d-1)\tau}, x^{t-(d-2)\tau}, ..., x^t)$$

(19)

that is lagged vectors from $d$ nearest previous points starting at a point $t$. The dependent variable is the immediate next point to the starting $y_t = x^{t+1}$. Theoretical studies reveal that difficulties in time-series
prediction arise from the high dimensionalities of the series. This makes the choice of the embedding dimension $d$ and delay time $\tau$ important design issues.

Series produced with the Mackey-Glass differential equation [25] for simulating blood flow are used for learning time-series descriptions. An embedding dimension $d = 10$, and delay time $\tau = 1$ are selected heuristically. A trajectory of 400 points is derived with parameters: $a = 0.2, b = 0.1$, and differential $\Delta = 17$. The first 100 points are used for training, and the remainder for testing.

4.2 The Methodology: Fitness Landscapes

The fitness landscape structure provides valuable information for navigation within the search space of the inductive task. This information has two aspects: local and global search characteristics of the landscape. The local search characteristics depend on the relation between the fitness and the operator. The global search characteristics depend on the correspondence between the fitness and the distance to a known global optima. We conduct empirical studies into the fitness landscape characteristics to analyze the search mechanisms, and their effects on the evolutionary search performance. Statistical correlation measures are employed to investigate how the proposed extended set of basis polynomials, and the recombinative guidance influence the landscape structure and improve STROGANOFF.

4.2.1 Local Search

The autocorrelation function of a fitness landscape, in context of iGP, reveals the statistical correlation between the fitnesses of trees separated by a series of applications of the genetic learning operator, from which the landscape arises. A high autocorrelation indicates a locally smooth fitness landscape that is easy to search with the operator in most cases. We calculate the autocorrelation function (AC) of the mutation landscape with the fitnesses of parent $p$ and child $c$ pairs of trees, taken during a random walk from an arbitrary initial tree through a set of trees generated by successive mutations [27]:

$$AC(p, c) = \frac{\text{Cov}(Mp, Mc)}{s(Mp)s(Mc)}$$

(20)

where: $\text{Cov}(Mp, Mc)$ is the covariance between the MDL-based fitnesses $Mp$ of $p$ and $Mc$ of $c$; $s(Mp)$ and $s(Mc)$ are the standard deviations of the fitnesses $Mp$ and $Mc$ respectively. The covariance is computed with
a large number \( n \) trees using the fitness means \( \overline{M} \) as follows:

\[
\text{Cov}(M_p, M_c) = \frac{1}{n} \sum_{i=1}^{n} (M_{p_i} - \overline{M})(M_{c_i} - \overline{M})
\]  

(21)

A correlation analysis of different fitness landscapes was performed. Four fitness landscapes arising in four modifications of the original system STROGANOFF were considered: 1) with one complete quadratic bivariate basis polynomial, defined by (3), and random selection of the mutation point, 2) with one complete quadratic bivariate basis polynomial (3) and selection of the mutation point by recombinative guidance\(^2\) (subsection II.E), 3) with a set of basis polynomials (given in Appendix) and random selection of the mutation point, and 4) with a set of basis polynomials and recombinative guidance.

The empirical autocorrelations of these four fitness landscapes, displayed in Figure 4, were produced using \( MDL \) fitness function (8) evaluated with the Mackey-Glass [25] series points as a training example set. Each \( AC \) curve plotted in Figure 4 is taken after conducting 10 random walks over 2000 trees, that is with a population having 1 tree and 2000 offspring produced by successive applications of the context-preserving mutation operator (subsection II.D), starting from an arbitrary initial tree.

(Figure 4 here)

The objective was to reveal whether or not the local search characteristics of the fitness landscape become smoother when recombinative guidance is used for choosing mutation points, and whether or not it becomes smoother when the extended set of basis polynomials is employed. The reason is that the smoother local fitness landscapes may be considered as indication of a possible local search improvement. It was found that the four autocorrelations computed with (20) have a very small standard deviation, each less than 0.0035\%. Since the landscapes seem similar looking in these subspaces, one may think that the walked subspaces have locally isotropic structures. Hence, formula (20) for \( AC \) may be applied to study the four landscapes subject to the assumption that the analysis is restricted to the concrete landscape subspaces [13].

The upper two curves in Figure 4 produced with the extended set of basis polynomials show higher correlation between the fitnesses of parent and children trees, which means that their fitness landscapes are smoother and this will make local search on them easier. A large number of basis polynomials enables flexible term-by-term construction of partial polynomial models, rather than composing polynomials from the complete quadratic basis only. We note that STROGANOFF equipped with the extended set of basis polynomials

\(^2\)Recombinative guidance when doing mutation suggests selecting the node with highest \( ASR \) error as the point for mutation.
behaves best with guided mutation (see the bold curve in Figure 4). This evidence makes it possible to predict that the recombinative guidance will also contribute to facilitating the local search in comparison with the random mutation node selection. One observes that in the two cases of using only the complete quadratic basis polynomial, or the set of 16 basis polynomials (given in the Appendix), recombinative guidance implies smoother local fitness landscapes.

(Figure 5 here)

4.2.2 Global Search

The global search difficulty depends on the global fitness landscape structure. The relationship between the fitness and the distance to a known global optimum gives evidence of the global character, that is the smoothness and ruggedness, of the fitness landscape. The distance to the global optimum should decrease as the fitness improves. This will be an indication for a relatively smooth landscape, which is considered easy to search.

We use the fitness distance correlation (FDC) [20] measure to examine the degree of mutual connection between the fitness and the distance:

$$FDC(M, d) = \frac{Cov(M, d)}{s(M)s(d)}$$ (22)

where: $Cov(M, d)$ is the covariance between the $MDL$ fitness $M$ and the distance $d$; $s(M), s(d)$ are the standard deviations of $M$ and $d$. The tree-to-tree distance $d$ between two trees is the number of one-point mutations needed to produce one of them from the other [32].

Random samplings of trees were made, and their tree-to-tree distances to the best tree found from 1000 runs were computed. The rationale is that, in practice, large samplings are used to derive statistical evidence for the landscape characteristics, since one does not know the global optima. The fitness evaluations were made using the Mackey-Glass series. The calculated $FDC$ values were as follows: 1) from the regularized $MDL_R$ fitness function (16) in the interval $[0.15, 0.28]$, and 2) from the ordinary $MDL$ function (8) in the interval $[0.05, 0.16]$. These empirical revelations allow us to think that the fitness landscape of the $MDL_R$ function (16) is smoother globally, hence the global performance of the iGP system could be improved when $MDL_R$ is used to guide the evolutionary search.

(Figure 6 here)

Figure 5 reveals that the regularized $MDL_R$ produces higher $FDC$. The average line through the scatter plot

18
obtained with the $MDL_R$ is at a greater angle with the abscissa than the one with $MDL$. This is because the regularization smoothens the fitness landscape more than the $MDL$. The $MDL$ penalizes polynomial models with a large number of coefficients, while $MDL_R$ strengthens this penalty requiring that in case of polynomials with close or equal number of coefficients the one with smaller coefficient values should be preferred.

4.3 Learning Accuracy and Generalization

The landscape correlation analysis is only a necessary condition for tuning the iGP performance, but it is not sufficient to guarantee that the iGP system will search progressively. The reason is that an iGP is a complex system whose behaviour depends not only on the fitness function and the genetic operators, but on other coupled components, such as the selection mechanism and the reproduction scheme. Since the intention is to realize to what degree the regularization of the fitness function influences the evolutionary search process, we study below an enhanced implementation of the system STROGANOFF using the same set of 16 basis polynomials (given in the Appendix) as in the regularized enhanced version reSTROGANOFF. The intention is to make fair comparisons of two iGP systems with equal representation potential.

We examine the performance of these iGP systems by tracking the changes in the learning accuracy of the bred polynomials and their generalization capacity. The intention is to understand: 1) whether or not the regularization causes premature convergence to suboptimal solutions, or it may help to focus gradually on fitter polynomials, and 2) whether or not the regularization enables to yield polynomials with lower variance.
(Table 2 here)

**Accuracy.** The learning accuracy curves, made with the average and best residuals $ASR$ over the training examples, depicted in Figure 6, show the tendency of reSTROGANOFF to discover fitter tree-like polynomials than those found by the ordinary approach. The regularized version pushes the population faster and longer, that is during a longer number of generations, and it orients on the fitness landscape toward better solutions, compared to ordinary STROGANOFF, which ceases to learn earlier.

We think that the search improvement is due mostly to the fitness landscape smoothing made with the correcting variance factor incorporated in the regularized fitness function $MDL_R$. The best discovered polynomials given in Table 2 have accuracies as follows: the regularized polynomial (Table 2,a) $ASR_R = 0.000244$, and the ordinary one (Table 2,b) $ASR = 0.000451$. Note that although the two trees in Table 2 have the same number of nodes, they imply different polynomials. The structural complexity of the regularized polynomial is much smaller, as the regularized polynomial (Table 2,a) is of fourth-order while the ordinary polynomial (Table 2,b) is of ninth-order.

**Generalization.** The problem is how to estimate the generalization, that is the degree of predicting the dependent variable given unseen values of the independent variables. We use the average relative variance $ARV$ [45]:

$$ARV = \frac{\sum_{i=1}^{N} (y_i - f(x_i))^2}{\sum_{i=1}^{N} (y_i - \overline{y})^2} \quad (23)$$

where $y_i$ is the true outcome of the $i$-th example, $f(x_i)$ is the outcome estimated with the $i$-th input vector $x_i$ in the same example, and $\overline{y}$ is the mean of the true outcome.

(Figure 7 here)

(Figure 8 here)

Figure 7 shows that the variance errors $ARV$ of the regularized polynomial discovered by reSTROGANOFF (Table 2,a) are lower than those of its ordinary polynomial version with coefficients recalculated by least squares fitting without the correcting functional.

Pursuing moderate smoothing, so as to achieve together acceptable accuracy on the training data and adequate generalization on future data, we propose selecting a value of the regularization parameter $k$ that causes medium average squared error $ASR$. This is because one cannot be certain as to what degree a polynomial overfits the available training examples. The effects from different values of the regularization parameter $k$ on
the ARV are measured over the unseen series points (Figure 8). It is evident in Figure 8 that the selection of a medium regularization parameter \( k = 0.001 \) is a reasonable choice.

(Table 3 here)

This \( k = 0.001 \) is a compromise value between the two values \( k_1 = 0.00025 \) and \( k_{10} = 0.005 \) which allows to produce really good generalizations. Near the lower boundary value \( k_1 = 0.00025 \), the generalization error on near future points is higher, but after that it diminishes on distant points. Near the higher boundary value \( k_{10} = 0.005 \), the generalization error on close future points is lower than those caused by \( k = 0.001 \), whereas on distant future points the error increases more.

The performance of the ordinary STROGANOFF, and its regularized version reSTROGANOFF, are compared with a traditional Koza-style GP [30]. This Koza-style GP also evolves nonlinear expressions using however a different function set: \( +, - , \times , \div , \sin , \cos , \text{power} \), and 10 terminal variables, with the same embedding \( d = 10 \). The best non-linear expression found by Mulloy and colleagues [30]: \((x_1 \times x_1 \times x_1)/(x_2 \times x_2 \times x_2) \times x_3\), was taken and estimated with the ARV measure (Table 3). It is clear from Table 3 that iGP evolves better solutions of the Mackey-Glass task with the selected parameters.

The systems from the STROGANOFF family use basis function trees, like the neural network algorithm LMS Tree [40]. LMS Tree builds a tree-structured network that models various nonlinear functions depending on which basis functions are used in its nodes. When the one dimensional function basis: \( \phi(x_i) = x_i \) is considered, LMS Tree infers exactly a high-order multivariate polynomial. The network is trained by the least mean square (LMS) error algorithm for learning the weights. It grows the tree-structure downward from the root to the leaves by expanding subtrees below the child nodes with largest error variance at each layer. The restriction is that each input variable may appear only once along a path from the root to the leaves. Because of this, the tree depth can be predefined to limit the polynomial order.

We ran the LMS Tree with a grow limit up to 20 subtrees on the Mackey-Glass series generated with the above parameters. Table 3 shows that even growing a complex solution the LMS Tree algorithm is inferior to GP, STROGANOFF and reSTROGANOFF. These systems produce better results than LMS Tree on this task with less complexity, higher accuracy on the training data, and better generalization on future data (Table 3).

(Table 4 here)
5 Range of Applicability

An iGP system evolving high-order multivariate polynomials is especially suitable for addressing real-world problems that feature nonlinear characteristics. We demonstrate the applicability of STROGANOFF and its regularized version to chaotic time-series prediction and data mining tasks. The examples' matrices in these tasks are often ill-conditioned and regularization methods should be preferred. We do not claim, however, that there exist ver accurate analytical polynomial models for these real tasks. Rather, we present our solutions found by the iGP systems.

The enhanced implementation of STROGANOFF using the same set of 16 incomplete bivariate basis polynomials (see Appendix) using the MDL fitness function (8) is compared with reSTROGANOFF using the regularized MDLR (16) in order to determine the effect of the regularization. In order to facilitate the comparisons both systems use the same iGP micromechanisms (subsections IIL, ILE), and the same parameters given in Table 1.

Learning the sunspot series [45]. The not-exactly-periodic appearances of sunspots are natural phenomena. The sunspot series contain 280 data points, divided into three subsets: the data from years 1700 – 1920 for training, and those from 1921 – 1955 as well as from 1956 – 1979 for testing. The best polynomial from 50 runs by ordinary STROGANOFF has size 9 with 18 coefficients, and that by reSTROGANOFF has size 9 with 18 coefficients.

The accuracies in Table 4 ASR achieved by STROGANOFF and reSTROGANOFF are very close. Note that the best regularized polynomial performs considerably better on long-term prediction than all other models. Compared to the reported results from a traditional Koza-style GP [19], which uses symbolic regression models of elementary functions, like +, –, ·, /, sin, cos, exp etc., we realize that the cascaded polynomials used in iGP have not only better accuracy, but also better forecasts on the three studied future periods. Related to the results from neural networks trained by backpropagation (BP-NN) [45], we see that STROGANOFF compares favorably to their generalization in the interval 1700 – 1920, while reSTROGANOFF is much more predictive. It is important to note that the two iGP systems are better on short-term forecasting, and considerably better on long-term forecasting in the interval 1956 – 1979. Figure 9 shows the best regularized polynomial approximation graphed using a solid line.

(Figure 9 here)
Learning of currency exchange rates (Set C) [46]. The financial problem of learning the trend in observed series of foreign currency exchange rates aims at forecasting future profit increase. Learning high-frequency exchange rate data sequences is most difficult since they feature almost a random character. We consider the benchmark currency exchange rate time series that contains 10 segments of 3000 points each, which give tickwise bid quotes from a bank for the rate between the Swiss franc (CHF) and the U.S. dollar (USD). All 30000 points are divided in two halves: one half for training and one half for testing.

One of the difficulties with these real data is that the exchange rates are taken at irregular time intervals. That is why the construction of examples as lagged independent variable vectors is not straightforward, moreover it impacts the embedding scheme and it is crucial for successful learning. We preprocessed the currency exchange series in order to derive 1-minute, 15-minute, and 60-minute predictors. From the given series of 30000 points we filtered out six: 3 training and 3 corresponding testing subseries of almost equally spaced in time currency rates.

The first set includes points at a distance $1 \pm 0.5$ minutes apart (one including 1652 training points and another one with 3169 testing points), the second set includes points at a distance $15 \pm 1.5$ minutes apart (with 938 and 1439 points), and the third set includes points at a distance $60 \pm 6$ minutes apart (with 482 and 625 points) [50]. If at a particular clock tick there is no corresponding data point, we compute artificially such a point as the average from its immediate neighbors in the series. Note that even if we do such space filling we cannot change the random walk character of the financial series.

Such a strategy for preprocessing of the data series assumes a continuity in the currency exchange rate changes from one day to the next, that is the consecutive days are pasted one after the other. This may not be quite plausible, and we do not claim that this is a perfect strategy.

(Figure 10 here)

Experiments were conducted to learn a separate polynomial from each of the three subseries, thus finding three separate predictors. The best polynomial from 50 runs on the second subseries at distance $15 \pm 1.5$ minutes apart evolved by the ordinary STROGANOFF was of size 11 with 21 coefficients, and the best regularized polynomial from reSTROGANOFF was of size 13 with 27 coefficients. Figure 10 displays the approximation of a segment from the second currency exchange rate series (at distance $15 \pm 1.5$ minutes apart) by the best polynomial discovered by reSTROGANOFF. Just for a comparison, the best expression evolved by the traditional Koza-style
GP on the second subseries was: \(((x_6 - x_4)/x_5) \times \log_2(x_2) \times \sin(x_3)) + x_1.\)

Our intention is to make the evolved patterns comparable with those available in the literature. In order to facilitate comparisons, we adopt a specialized measure for evaluating time series approximations. This is the ratio \(RAT\) measure [46] proposed for assessment of the error at each next series point with respect to the tendency after the last point in the series:

\[
RAT = \frac{\sum_{i=1}^{N} (y_i - f(x_i))^2}{\sum_{i=1}^{N} (y_i - y_{i-1})^2}
\]

(24)

where \(y_i\) is the true current observation of the \(i\)-th example, \(f(x_i)\) is the model outcome estimated with the \(i\)-th input vector \(x_i\) in the same example \(i\), and \(y_{i-1}\) is the last true observation of example \(i-1\). When the ratio measure \(RAT < 1\) the prediction is assumed better than random, and respectively when the ratio measure \(RAT > 1\) signifies a worse than by-chance prediction.

The learning results from all of the three subseries are given in Table 5. They are related to the best results from the Santa Fe competition achieved by a neural network (NN) [28].

(Table 5 here)

Table 5 indicates that all the genetic programming systems: STROGANOFF, \(re\)STROGANOFF, and traditional Koza-style GP produce very close results on such exchange rate prediction tasks. This is because the financial data series exhibit quite an aperiodic, chaotic character that is extremely difficult to capture by functional models. Even small improvements like that shown by the best results from \(re\)STROGANOFF are considered promising, and acceptable from an economical point of view.

On such extremely complex tasks the GP systems seem to be worse than the neural network on approximation showing accuracies \(RAT > 0.9\), while the neural network shows accuracies \(RAT < 0.9\). Note, however, that the iGP system \(re\)STROGANOFF better extrapolates the second series (at distance 15 ± 1.5 minutes apart) achieving highest generalization \(RAT = 1.021\).

**Data mining.** The iGP performance is studied with five data mining tasks from the machine learning database repository [5]: Iris, Ionosphere, Glass, Credit, and Vehicle. Experiments were conducted with the ordinary and regularized iGP systems STROGANOFF to find one polynomial for each particular class from each of these tasks.

Table 6 provides the error rates of learned concepts for each of the selected data mining tasks, given in percentage of the correctly recognized training examples. The variances, in the sense of the mean and standard
deviation of the error of correctly recognized examples, of 10 good solutions discovered in 10 different runs, are estimated according to the 10-fold cross-validation technique [47]. The two iGP systems are related in Table 6 to a machine learning system Ltree [9] that learns oblique multivariate trees, since it also finds nonlinear approximations of the data, and to the standard decision tree machine learning system C4.5 [38]. An oblique tree as well as a decision tree classifies the data into all classes, and the comparison with the polynomials is not straightforward. For this reason, we estimated the error variances of the evolved best STROGANOFF and reSTROGANOFF polynomials as average variances from each group of polynomials that together learn all classes of the concrete task.

(Table 6 here)

Table 6 suggests that the regularization approach converges on average to more accurate polynomials than those identified by ordinary STROGANOFF. Furthermore, the variances of the polynomials from reSTROGANOFF are smaller in most of the cases, therefore they feature a higher generalization potential. One may observe that iGP discovers apparently better solutions than the machine learning systems on the complex tasks, like Glass, Credit, and Vehicle. The reason is that iGP is able to evolve high-order polynomials that very closely approximate the decision boundaries among the data. The non-evolutionary systems Ltree and C4.5 are slightly better on the simpler data sets Iris and Ionosphere.

On complex tasks iGP discovers shorter trees, for example of size 8 with 22 coefficients on the Glass data while Ltree produces trees of size approximately 32 and C4.5 acquires trees of size 44. Although iGP induces trees of almost equal size to these learned by Ltree and C4.5 on the simple tasks, 5 on the Iris data set and respectively 12, 14, 18 on the Ionosphere data set, the evolved polynomials include more terms. The best polynomial found by reSTROGANOFF from the Iris data of accuracy 99.90% contained 8 coefficients.

Summary. After an analysis of the iGP performance on these tasks it could be summarised that:

1) the ordinary system STROGANOFF usually produces overfitting polynomials, since their generalization capacity seems to be worse than that of the regularized polynomials discovered by reSTROGANOFF. Empirical evidence for the better forecasting capacity of reSTROGANOFF are given in Tables 3-5;

2) the ordinary polynomials evolved by STROGANOFF do not improve even if we recalculate their coefficients with the regularized matrix formula (17). This means that there is no guarantee to find satisfactory results on such tasks if one uses STROGANOFF, and after that simply recalculate the coefficients of the best
discovered polynomial with formula (17): \( a = (H^T H + kI)^{-1} H^T y \), that is by adding a regularization factor \( kI \) to the covariance matrix \( H^T H \) before doing inversion. From another point of view, the plots in Figure 7 reveal that the best polynomial from reSTROGANOFF shows a worse performance when we recalculate its coefficients simply by ordinary least squares fitting.

An explanation for such phenomena could be that in real world tasks the data often suffer from multicollinearities in the example matrix, and that is why some good polynomials are discarded by the ordinary iGP during evolutionary search. When regularization is applied, these may appear as promising solutions that survive and help direct the search toward good polynomials.

6 Discussion

The iGP approaches from the STROGANOFF family evolve polynomials represented as nonlinear multivariate trees. An advantage of these approaches is that the polynomial coefficients in the trees are computed directly as least-squares solutions by the method of normal equations, which avoids searching their values. The STROGANOFF systems achieve very accurate results trying only to discover the inherent model structure, that is tree topology, of basis polynomials that constitute the target polynomial.

Strengths of the regularization approach to iGP. Three main strengths of the proposed regularization approach to iGP can be noted.

First, when regularization is applied to an iGP system it will lose slightly in speed of computation, but it will gain increased predictability.

Second, without regularization STROGANOFF often produces oversized polynomials, because the tree growth usually decreases the error. When regularization is applied, the covariances among the polynomial terms are considered, and thus the interactive behavior of the variables is identified more precisely. The overall effect of regularization is a decrease of the error variance.

Third, the regularization improves STROGANOFF providing a reliable scheme for computing the coefficients and, thus, avoiding problems of ill-posedness of the example matrix. This is important since in real applications ill-effects of multicollinearity often arise due to omission of observations and sensitivity of the polynomials to slight changes in the examples. The improvement in accuracy and generalization shown in this paper may seem small, but during the experiments we did not use the original STROGANOFF but rather its extended version.
with the set of basis polynomials to emphasize especially the need for regularization.

**Evolving Kolmogorov-Gabor polynomials.** The STROGANOFF systems evolve universally approximating Kolmogorov-Gabor polynomials, which are preferable representations over many other symbolic expressions of inhomogeneous elementary functions in the traditional Koza-style GP systems [2], [22], [42]. Employing such polynomials allows for learning reliable mathematical models [26]. The traditional Koza-style GP may use the proposed regularization approach directly for elaboration of fitness functions. This will be helpful as many of them exhibit a tendency to capture specific details in the provided training examples, and cannot identify the systematic aspects in the training sample well.

An early genetic algorithm has been suggested for evolving Kolmogorov-Gabor polynomials represented as polynomial networks [21]. Such a network representation makes the inductive task, however, much more difficult since it requires finding the complex network topology for the given task, that is, identifying the network connectivity. This difficulty is overcome to a great degree with the tree-like network representations used in the STROGANOFF systems as they attempt to discover hierarchical network topologies in the known format of fixed binary tree structures.

**STROGANOFF vs. statistical learning algorithms.** The STROGANOFF systems relate closely to the statistical learning algorithms GMDH [17], [18], [26], [29] and MAPS [3] in that they discover the structure of the polynomials. A principle distinction from GMDH and MAPS is that STROGANOFF performs a search in the space of whole polynomials, while the others grow a single polynomial iteratively. STROGANOFF maintains a population of polynomials competing to reproduce and reorient the search, which allows them to escape from local optima. The search is navigated with the fitnesses of the completed polynomials with all their terms and coefficients completely identified. This is different from GMDH and MAPS where one polynomial network is built in such a way that the formation of the next layer of basis polynomials is guided by the partial fitnesses from the previous levels. This is an obstacle that makes these learning networks susceptible to entrapment at local optima solutions [31].

**The GMDH and STROGANOFF families.** Since there are various GMDH algorithms [26], we would like to point that the iGP systems studied here from the STROGANOFF family are related to the multilayer GMDH algorithm. The multilayer GMDH algorithm syntesizes a polynomial network layer by layer doing heuristic search for pairs of essential variables to feed the next node so that its residual outcome error is
decreased. This, however, constrains the feeding of higher layers since they depend on previously learned nodes that may not be optimal. STROGANOFF avoids this insufficiency by forming basis polynomials at a certain layer with randomly selected independent variables from the set of all variables, thus avoiding any dependance on learned nodes.

The main difference in the polynomial models between STROGANOFF systems and those used in the multilayer GMDH is that when employing tree-structured networks we reduce the representation complexity, which makes the models easier to process. The GMDH polynomial networks feature a large number of internode connections, because there are links going out of every internal node to every node in next layer. More precisely, the multilayer GMDH constructs strictly layered networks of basis polynomials. The implementations of GMDH systems may benefit from the presented MDL fitness function and it regularization as they are directly applicable to it.
7 Conclusion

This paper contributes to the research into increasing the search abilities of inductive GP. We have enhanced an ordinary GP system STROGANOFF and designed an improved fitness function. This is a regularized fitness function that serves to compromise between agreement with the examples and stability of the solution with respect to the variation in the examples. This function enables discovery of polynomials that interpolate well, with a good ability to extrapolate on unseen examples. This claim holds for inductive learning tasks with training samples with a close number of examples, and small-to-moderate dimensionality of the examples as these studied in this paper. In the limit, exact approximation is not guaranteed since the maximal polynomial size is predefined by a design decision, since one always gives the upper polynomial size which imposes the acceptable error boundary.

This paper indicates a general approach to iGP with polynomial models of practical importance. The regularization approach to learning polynomials can be used successfully for nonparametric approximation due to the following advantages: 1) it discovers the model structure from the data automatically, that is, the size and terms of the polynomials, 2) it generates explicit representations in the form of multivariate high-order polynomials amenable to human comprehension, and 3) it makes the polynomials well-conditioned, thus computationally stable and suitable for practical purposes.

Acknowledgments

The authors are grateful to Professor Alexey G. Ivakhnenko for the the invaluable discussions on GMDH, and to Terry Jones and Wim Hordijk for the use of correlation measures to analyze the fitness landscapes arising in GP. We thank to Terence Sanger for providing the implementation of the LMS Tree algorithm. Thanks also to the anonymous reviewers and special thanks to David Fogel who helped improve the paper quality considerably.
**Appendix: Bivariate Basis Polynomials**

<table>
<thead>
<tr>
<th></th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( f_1(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 )</td>
</tr>
<tr>
<td>2</td>
<td>( f_2(x) = a_0 + a_1x_1 + a_2x_2 )</td>
</tr>
<tr>
<td>3</td>
<td>( f_3(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 + a_4x_2^2 )</td>
</tr>
<tr>
<td>4</td>
<td>( f_4(x) = a_0 + a_1x_1 + a_2x_1x_2 + a_3x_1^2 )</td>
</tr>
<tr>
<td>5</td>
<td>( f_5(x) = a_0 + a_1x_1 + a_2x_2^2 )</td>
</tr>
<tr>
<td>6</td>
<td>( f_6(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_1^2 )</td>
</tr>
<tr>
<td>7</td>
<td>( f_7(x) = a_0 + a_1x_1 + a_2x_1^2 + a_3x_2^2 )</td>
</tr>
<tr>
<td>8</td>
<td>( f_8(x) = a_0 + a_1x_1^2 + a_2x_2^2 )</td>
</tr>
<tr>
<td>9</td>
<td>( f_9(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 )</td>
</tr>
<tr>
<td>10</td>
<td>( f_{10}(x) = a_0 + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 )</td>
</tr>
<tr>
<td>11</td>
<td>( f_{11}(x) = a_0 + a_1x_1 + a_2x_1x_2 + a_3x_1^2 + a_4x_2^2 )</td>
</tr>
<tr>
<td>12</td>
<td>( f_{12}(x) = a_0 + a_1x_1 + a_2x_1^2 + a_3x_2^2 )</td>
</tr>
<tr>
<td>13</td>
<td>( f_{13}(x) = a_0 + a_1x_1 + a_2x_1x_2 + a_3x_2^2 )</td>
</tr>
<tr>
<td>14</td>
<td>( f_{14}(x) = a_0 + a_1x_1 + a_2x_1x_2 )</td>
</tr>
<tr>
<td>15</td>
<td>( f_{15}(x) = a_0 + a_1x_1x_2 )</td>
</tr>
<tr>
<td>16</td>
<td>( f_{16}(x) = a_0 + a_1x_1x_2 + a_2x_1^2 )</td>
</tr>
</tbody>
</table>
References


Dr Nikolay Nikolaev received the Ph.D. degree in computer science and engineering in 1992 from the Technical University in Sofia, Bulgaria. From 1992 to 1993 he conducted postdoctoral research in machine learning from the University of Wales, Cardiff, United Kingdom. Since the fall of 1993 he was an Assistant Professor in Computer Science at the American University in Bulgaria. In fall of 2000 he joined the Department of Mathematical and Computing Sciences at Goldsmiths College, University of London as a Lecturer in Computing. His theoretical research interests include: evolutionary computation- genetic algorithms, inductive genetic programming and classifier systems; neural networks- time-delay neural networks, polynomial networks, and immune networks; machine learning- non-linear and distance-based decision trees; His application interests include: time-series prediction, financial engineering, pattern recognition, classification, and data mining.

Dr. Hitoshi Iba graduated from Dept. Science of University of Tokyo in 1985 and received his Ph.D. degree from Department of Engineering of University of Tokyo in 1990. Since then, he had been working in ETL (ElectroTechnical Lab.) He left from ETL and moved to the University of Tokyo in April, 1998. He is an associate editor of IEEE Transactions on EC and Journal of Genetic Programming and Evolvable Machines (GPEM). His research interests include: Evolutionary Computation, Genetic Programming, Bio-infomatics, Foundation of Artificial Intelligence, Machine Learning, Robotics, Vision.
Figure 1. A tree-like polynomial representation considered in STROGANOFF which composes complete bivariate basis polynomial functions $f_j(x)$, $f_j: \mathcal{R}^2 \rightarrow \mathcal{R}$, $x = (x_1, x_2)$, applied with input variables:

$$x_i, 1 \leq i \leq 10, \ x_i \in \mathcal{R}.$$
Figure 2. A tree-like polynomial representation considered in the enhanced reSTROGANOFF system which composes incomplete bivariate basis polynomial functions \( f_j(x) \), \( f_j: \mathbb{R}^2 \rightarrow \mathbb{R} \), \( x = (x_1, x_2) \), applied with input variables: \( x_i, 1 \leq i \leq 10, x_i \in \mathbb{R} \).
Figure 3. Approximation of the Mackey-Glass curve by a relatively smooth regularized polynomial (Table 2,a) obtained with $k = 0.001$, the same polynomial (Table 2,a) with recalculated coefficients by LS, and a more fluctuating polynomial evolved by STROGANOFF (Table 2,b) which coefficients are estimated by LS. These are segments from the real polynomials with knots computed at discretely sampled points with step size 1.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value / Kind</th>
</tr>
</thead>
<tbody>
<tr>
<td>population size</td>
<td>100</td>
</tr>
<tr>
<td>initial population</td>
<td>random</td>
</tr>
<tr>
<td>reproduction</td>
<td>steady-state (elite 50%)</td>
</tr>
<tr>
<td>selection</td>
<td>fitness-proportional</td>
</tr>
<tr>
<td>selection technique</td>
<td>elitist (the best 2 survive)</td>
</tr>
<tr>
<td>mutation</td>
<td>25% (i.e. half of the elite)</td>
</tr>
<tr>
<td>crossover</td>
<td>25% (i.e. half of the elite)</td>
</tr>
<tr>
<td>mutation probability</td>
<td>$p_m = 0.01$</td>
</tr>
<tr>
<td>crossover probability</td>
<td>$p_c = 1.75$</td>
</tr>
<tr>
<td>function set</td>
<td>$f_1(x),...,f_{16}(x)$ (16 basis polynomials)</td>
</tr>
<tr>
<td>terminal set</td>
<td>$x_1,...,x_{10}$ (10 independent variables)</td>
</tr>
<tr>
<td>maximal tree size</td>
<td>40 treenodes (functional + terminal)</td>
</tr>
</tbody>
</table>

**Table 1.** Parameters for both iGP systems STROGANOFF and reSTROGANOFF used in all runs analysed in this paper.
Figure 4. Averaged autocorrelations (AC) from 10 runs, each computed after a random walk over 2000 tree-like polynomials sampled by the context-preserving mutation operator with recombinitive guidance and random node selection, using the complete quadratic basis polynomial or a set of basis polynomials.
Figure 5. Scattered plots of the fitness/tree-to-tree distance correlation calculated with 2000 randomly sampled trees, evaluated by the ordinary $MDL$ and regularized $MDL_R$ fitness functions.
Figure 6. Average and best errors from one particular run conducted by each of the ordinary STROGANOFF and its regularized version reSTROGANOFF using: $MaxTreeSize=40$, $PopSize=100$, and 100 data points from the Mackey-Glass series.
<table>
<thead>
<tr>
<th>rSTROGANOFF</th>
<th>STROGANOFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>(node9)</td>
<td>(node10)</td>
</tr>
<tr>
<td>(node7)</td>
<td>(node13)</td>
</tr>
<tr>
<td>x4</td>
<td>x4</td>
</tr>
<tr>
<td>x2</td>
<td>x4</td>
</tr>
<tr>
<td>(node2)</td>
<td>x1</td>
</tr>
<tr>
<td>x1</td>
<td>x2</td>
</tr>
<tr>
<td>x2)</td>
<td>x6)</td>
</tr>
</tbody>
</table>

where the nodes contain the following basis polynomials:

node2: \( f_2(x) = a_0 + a_1 x_1 + a_2 x_2 \)

node7: \( f_7(x) = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_2^2 \)

node9: \( f_9(x) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_1 x_2 + a_4 x_1^2 + a_5 x_2^2 \)

node10: \( f_{10}(x) = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_1 x_2 + a_4 x_1^2 \)

node13: \( f_{13}(x) = a_0 + a_1 x_1 + a_2 x_1 x_2 + a_3 x_2^2 \)

Table 2. Best tree-like polynomials found from 50 runs: a) with the $MDL_R$ (16) using RAE with a regularization $k = 0.001$; and b) with the ordinary $MDL$ (8) using LS only on examples derived with the Mackey-Glass series equation.
Figure 7. Prediction (ARV) as a function of the unseen series points, measured with the best polynomial found from 50 runs by reSTROGANOFF (Table 2,a), and its ordinary version with recalculated coefficients by LS.
Figure 8. Influence of the values of the regularization parameter $k$ on the prediction potential of the best regularized polynomial, estimated with future series points from the Mackey-Glass equation.
<table>
<thead>
<tr>
<th></th>
<th>Accuracy (ARV)</th>
<th>Generalization (ARV)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-100</td>
<td>100-200</td>
</tr>
<tr>
<td>STROGANOFF</td>
<td>0.00552</td>
<td>0.00461 0.00482 0.00457</td>
</tr>
<tr>
<td>rSTROGANOFF</td>
<td>0.00467</td>
<td>0.00348 0.00361 0.00332</td>
</tr>
<tr>
<td>GP</td>
<td>0.00826</td>
<td>0.00725 0.00744 0.00721</td>
</tr>
<tr>
<td>LMS Tree (NN)</td>
<td>0.01128</td>
<td>0.00867 0.00892 0.00995</td>
</tr>
</tbody>
</table>

Table 3. Results from different leaning systems on the Mackey-Glass series generated with differential delay $\Delta = 17$. The best polynomials found by the iGP systems from Table 2 are considered, as well as the best nonlinear expression found by the traditional Koza-style GP and the LMS Tree neural network.
<table>
<thead>
<tr>
<th></th>
<th>Accuracy (ARV)</th>
<th>Generalization (ARV)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1700-1920</td>
<td>1921-1955</td>
</tr>
<tr>
<td>STROGANOFF</td>
<td>0.0803</td>
<td>0.0825</td>
</tr>
<tr>
<td>reSTROGANOFF</td>
<td>0.0731</td>
<td>0.0739</td>
</tr>
<tr>
<td>GP</td>
<td>0.1250</td>
<td>0.1820</td>
</tr>
<tr>
<td>BP-NN</td>
<td>0.0820</td>
<td>0.0860</td>
</tr>
</tbody>
</table>

Table 4. Learning results from ordinary STROGANOFF, reSTROGANOFF using parameter \( k = 0.0015 \), a traditional Koza-style GP system, and a neural network trained by backpropagation over the Sunspots data series.
Figure 9. Approximation and extrapolation of the sunspot series by the best regularized polynomial discovered by reSTROGANOFF over the Sunspots data series using $k = 0.0015$. 
Figure 10. Detrended interpolation of the currency exchange rate series, formed of points sampled at distance $15 \pm 1.5$ minutes apart, by the best polynomial from reSTROGANOFF using $k = 0.00025$. 
<table>
<thead>
<tr>
<th></th>
<th>Accuracy (RAT)</th>
<th>Generalization (RAT)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1 min</td>
<td>15 min</td>
</tr>
<tr>
<td>STROGANOFF</td>
<td>[0.892, 1.002]</td>
<td>[0.978, 1.001]</td>
</tr>
<tr>
<td>reSTROGANOFF</td>
<td>[0.986, 0.997]</td>
<td>[0.991, 1.001]</td>
</tr>
<tr>
<td>GP</td>
<td>[0.998, 1.443]</td>
<td>[0.992, 1.007]</td>
</tr>
<tr>
<td>NN</td>
<td>[0.889, .]</td>
<td>[0.891, .]</td>
</tr>
</tbody>
</table>

Table 5. Learning results from ordinary STROGANOFF, reSTROGANOFF, using \( k = 0.00025 \), a traditional Koza-style GP, and a Neural Network over the space filled and day-after-day pasted *Currency exchange rates* data. These are the intervals between the best and the worst polynomials found by each of the iGP systems.
<table>
<thead>
<tr>
<th>Data/class</th>
<th>variance</th>
<th>STROGANOFF</th>
<th>reSTROGANOFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>(%)</td>
<td>99.74±0.16</td>
<td>99.90±0.6</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>(%)</td>
<td>89.23±0.75</td>
<td>89.81±0.29</td>
</tr>
<tr>
<td>Glass</td>
<td>(%)</td>
<td>72.55±5.04</td>
<td>76.27±4.51</td>
</tr>
<tr>
<td>Credit</td>
<td>(%)</td>
<td>77.15±3.48</td>
<td>79.25±3.44</td>
</tr>
<tr>
<td>Vehicle</td>
<td>(%)</td>
<td>75.11±5.12</td>
<td>78.23±3.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ltree</td>
<td>C4.5</td>
</tr>
<tr>
<td>Iris</td>
<td>(%)</td>
<td>97.15±2.85</td>
<td>95.15±4.85</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>(%)</td>
<td>90.6±4.0</td>
<td>90.9±5.0</td>
</tr>
<tr>
<td>Glass</td>
<td>(%)</td>
<td>65.5±8.0</td>
<td>67.7±12.0</td>
</tr>
<tr>
<td>Credit</td>
<td>(%)</td>
<td>73.6±5.0</td>
<td>70.9±4.0</td>
</tr>
<tr>
<td>Vehicle</td>
<td>(%)</td>
<td>77.5±5.0</td>
<td>71.2±4.0</td>
</tr>
</tbody>
</table>

Table 6. Variance of results discovered by the systems reSTROGANOFF using a parameter value for $k = 0.001$, ordinary STROGANOFF, Ltree and C4.5. The variance measures are the mean and the standard deviations of error rates (in percentage of correctly recognized examples) estimated according to the 10-fold cross-validation technique with solutions derived in different runs.