Inf Syst Front (2017) 19:1123–1132
DOI 10.1007/s10796-016-9706-2

An evolutionary system for ozone concentration forecasting

Mauro Castelli[1] · Ivo Gonçalves[1,2] · Leonardo Trujillo[3] · Aleš Popovič[1,4]

Published online: 21 September 2016

Abstract Nowadays, with more than 50 % of the world's population living in urban areas, cities are facing important environmental challenges. Among them, air pollution has emerged as one of the most important concerns, taking into account the social costs related to the effect of polluted air. According to a report of the World Health Organization, approximately seven million people die each year from the effects of air pollution. Despite this fact, the same report suggests that cities could greatly improve their air quality through local measures by exploiting modern and efficient solutions for smart infrastructures. Ideally, this approach requires insights of how pollutant levels change over time in specific locations. To tackle this problem, we present an evolutionary system for the prediction of pollutants levels based on a recently proposed variant of genetic programming. This system is designed to predict the amount of ozone level, based on the concentration of other pollutants collected by sensors disposed in critical areas of a city. An analysis of data related to the region of Yuen Long (one of the most polluted areas of China), shows the suitability of the proposed system for addressing the problem at hand. In particular, the system is able to predict the ozone level with greater accuracy with respect to other techniques that are commonly used to tackle similar forecasting problems.

Keywords Evolutionary computation · Genetic programming · Smart cities · Forecasting · Air quality

✉ Aleš Popovič
ales.popovic@ef.uni-lj.si

Mauro Castelli
mcastelli@novaims.unl.pt

Ivo Gonçalves
igoncalves@novaims.unl.pt

Leonardo Trujillo
leonardo.trujillo@tectijuana.edu.mx

1 NOVA IMS, Universidade Nova de Lisboa, 1070-312 Lisbon, Portugal

2 CISUC, Department of Informatics Engineering, University of Coimbra, 3030-290, Coimbra, Portugal

3 Tree-Laboratory, Instituto Tecnológico de Tijuana, Blvd. Industrial y Ave. ITR Tijuana S/N, Mesa de Otay, Tijuana BC 22500, México

4 University of Ljubljana, Faculty of Economics, Kardeljeva ploscad 17. 1000, Ljubljana, Slovenia

1 Introduction

Cities have always represented the engine of industrial and technological growth and this fact has become even more evident in recent years: in 2009, more than half of the world's population lived in urban centers and by 2050 it is estimated that 70 % of the world's population will live in cities (United Nations 2014). While urbanization has created prosperity and new job opportunities, the high concentration of people living in urban areas has created noteworthy environmental challenges: urban centers, along with their traffic, industry, and energy needs, account for more than 70 % of the global gas emissions. Pollution composed of many different chemical components and small particles has raised an important public health problem that affects especially children and people presenting breathing difficulties: according to a report published in 2015 by the World Health Organization almost 90 % of the World's urban population breathes air with pollutant levels that are much higher than the recommended thresholds (World Health Organization 2015). Another study (Lim et al. 2012) has provided new evidence of the significant role that air pollution plays globally, placing it among the top ten risks faced by human beings. As reported in (Sharma et al. 2013; Kumar et al. 2013), many of the World's cities are unable to comply with the prescribed concentration limits of air pollutants and, in many cases, reported measurements far exceed these limits, resulting in millions of premature deaths (Kumar and Thiele 2014; Lim et al. 2012).

Among the several pollutants that exceed concentration limits are coarse (PM_{10}) and fine particulate matter ($PM_{2.5}$), and unregulated ultrafine particles ($< 100\ nm$) (Kittelson et al. 2004). Several studies have demonstrated the negative effect of these pollutants on human health (Kampa and Castanas 2008; Anderson et al. 2012; Kim et al. 2015). Despite all the measures that have been adopted for reducing air pollution, a report of the World Health Organization (World Health Organization 2014) suggested that the annual mean concentration of PM_{10} has increased by more than 5 % between 2008 and 2013 in 720 cities across the World. According to a study published in Medina et al. (2004), a reduction in long-term exposure to PM_{10} by 5 μg per cubic meter in Europe could avoid between 3000 and 8000 early deaths annually. Similar studies (Medina et al. 2004; Ayres 2010; Kumar and Thiele 2014) have been performed in order to evaluate the impact of $PM_{2.5}$ and ultrafine particles on human health and health costs in the UK. Results suggest an average loss of $7-8$ months in life expectancy for UK residents and about 20 billion pounds per year in corresponding health costs.

While these data highlight the importance of the problem, among the many environmental challenges facing urban areas, air quality is especially difficult to manage. One of the main issues is related to the high variability that affects air quality: pollutant levels are strongly related to weather conditions, wind speed, seasonality, and other occasional events. As reported in Karatzas and Kaltsatos (2007) and Sousa et al. (2007), to take effective measures able to counteract air pollution it is fundamental to make an accurate prediction of the air pollution level. To answer this call, this paper presents a system based on genetic programming. This system makes use of a recently proposed version of genetic programming, a machine learning technique belonging to the family of evolutionary computation. In particular, after introducing the concept of semantics in (the field of) genetic programming, we couple the semantics-based genetic programming system with a local search optimizer to increase the accuracy of predicting the ozone level based on the concentration of different air pollutants. To assess the performance of the system, we used data collected in 2014 in Yuen Long (a region of China). With the increasing interest in smart cities and internet of things (Li et al. 2015) and the usage of sensors (Corbette 2013) able to collect vast amount of data (Hota et al. 2015), the proposed system represents a viable option to analyze sensor data related to different phenomena.

The remainder of this paper proceeds as follows. Section 2 provides a general overview of genetic programming, focusing on the variant employed in this work and describing the method followed to include a local searcher in the evolutionary search. Section 3 presents the data used in this work to evaluate the performance of the proposed system and the experimental settings. Section 4 analyzes the results achieved by the proposed system and by other well-known machine learning techniques. Finally, Section 5 concludes the paper summarizing the main findings of this work.

2 Method

This section describes the system proposed to predict the ozone concentration level. Section 2.1 introduces basic concepts about genetic programming, an evolutionary computation technique that is the base of the framework that we developed. Section 2.2 presents a recently defined variant of genetic programming that is based on the concept of semantics. After giving the definition of semantics, the main features of this genetic programming variant are discussed. Finally, Section 2.3 shows how to combine semantic genetic programming with a local search optimizer, with the objective of improving performance and saving computational effort.

2.1 Genetic programming

Genetic Programming (GP) is a computational intelligence technique that belongs to the field of evolutionary computation (EC). EC deals with the development of global search and optimization algorithms which are designed based on some of the core principles of the neo-Darwinian theory (Koza 1992). However, the GP paradigm differs from other EC techniques in several key respects. GP is intended to solve problems that can be broadly defined as automatic program induction, most commonly following a supervised learning approach. In other words, the goal of GP is to evolve syntactic expressions that perform some form of computation, attempting to find the relation between a set of independent variables (inputs) and dependent variables (outputs). Conversely, most EC techniques, such as Genetic Algorithms (GAs), focus on function optimization. In the case of modeling, GAs can be used to optimize or tune model parameters, while GP is intended to automatically derive the syntactic structure of the model. Furthermore, unlike other machine learning paradigms, GP automatically defines the shape and size of the model with very little user involvement. On the other hand, GP follows the basic evolutionary process of other EC algorithms, that proceeds as follows: (1) a random generation of a set (population) of candidate solutions (individuals); (2) the use of a domain-specific fitness function that allows one to grade each solution; (3) a stochastic selection mechanism to probabilistically choose individuals (parents) that will be used to construct a new set of solutions (offspring); (4) stochastic search (genetic) operators, called mutation and crossover, that take selected parents as inputs and produce offspring as output, such that useful traits are inherited with the goal of progressively generating better solution; (5) this process is iteratively repeated (each iteration is called a generation) until a stopping criterion is met, such as a maximum computational effort which can be measured using the number of generations or the total number of function evaluations.

The representation used for individuals must be able to encode programs, mathematical formulas or other syntactic expressions. The most common representation used is tree structures, but other representations are possible. When using a tree-based representation, leaves can contain the problem's independent variables, constants and *0-arity* functions: all of these are referred to as terminals and compose the Terminal set Γ. Internal nodes are taken from a Function set F, that contains the primitive operations which can be used to construct the function or model. In general, Γ and F are chosen based on the problem features and domain, and together define the search space of the problem (the space of all possible programs that can be constructed by the evolutionary search). The search operators must be applicable to the chosen representation, and must allow for the evolution of unspecified expressions of different sizes and shapes. In standard tree-based GP, the operators are called subtree mutation and subtree crossover. More in detail, after choosing two individuals based on their fitness, subtree crossover performs the following operations: 1) selects a random subtree in each parent and 2) swaps the selected subtrees between the two parents (the resulting individuals are the offspring). Mutation introduces random changes in the structure of the individuals. Subtree mutation works as follows: 1) it randomly selects a node in a tree, 2) it removes whatever is currently at the selected point and whatever is below the selected point, and 3) it inserts a randomly generated subtree at that point. This operation is controlled by a parameter that specifies the maximum size (usually measured in terms of tree depth) for the newly created subtree that has to be inserted. Finally, fitness is usually expressed as an error function between the actual outputs of the evolved program and the target values over some data instances. Figure 1 provides a graphical representation of the basic GP process.

2.2 Semantic GP

Despite the large number of human-competitive results achieved with the use of GP (Koza 2010), researchers still continue to investigate new methods in order to improve the ability of GP to produce high-quality solutions. In recent years, one of the emerging ideas is to include the concept of semantics in the evolutionary process performed by GP. In this work we use the most common and widely accepted definition of semantics in GP literature (Krawiec and Lichocki 2009). The semantics of a program T_i is defined as the vector of outputs $\mathbf{s}_i = [T_i(\mathbf{x}_1), T_i(\mathbf{x}_2), ..., T_i(\mathbf{x}_n)]$ obtained after executing the program on a set of data $\mathbb{T}$

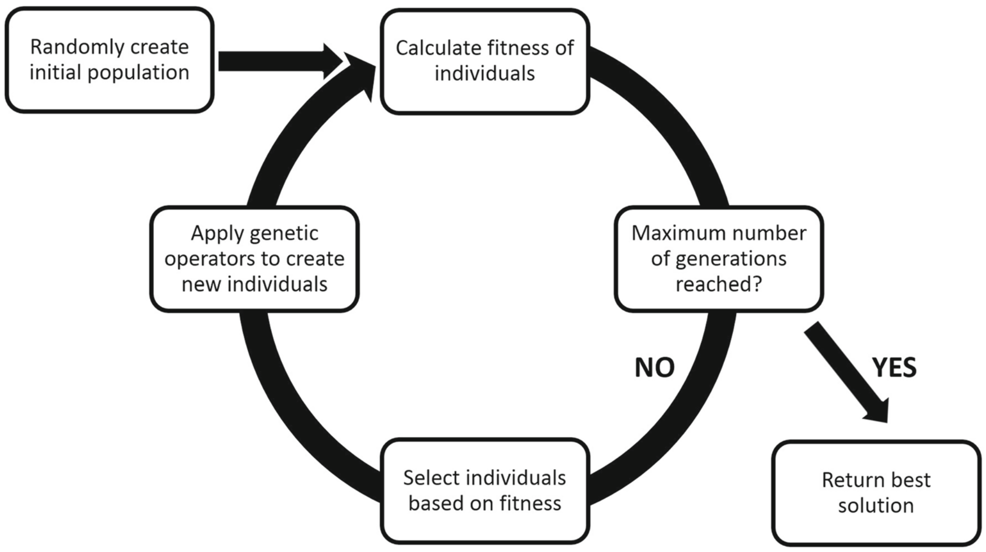

Randomly create initial population
Calculate fitness of individuals
Apply genetic operators to create new individuals
Maximum number of generations reached?
NO
YES
Select individuals based on fitness
Return best solution

Fig. 1 The GP algorithm

(Moraglio et al. 2012). When T_i represents a real-valued function then $\mathbf{s}_i \in \mathbb{R}^n$.

In the original definition of GP (Koza 1992), crossover and mutation produce offspring by manipulating the syntax of the parents. The idea of defining semantic methods in GP is to overcome one of the most important limitation of standard, syntax-based GP operators. In fact, while semantics determines what a program actually does, the traditional genetic operators manipulate programs only considering their syntax. Hence, traditional GP operators disregard the information about the behavior of programs provided by semantics. The drawback of this choice is that it is difficult to predict the effect that modifications of program syntax will have on the semantics of the generated offspring. To overcome this problem, genetic operators able to act on the semantics of the programs have been proposed (Moraglio et al. 2012). These operators, called Geometric Semantic Operators (GSOs), provide a simple but effective method to directly manipulate the semantics of the individuals and, more importantly, they are able to induce a unimodal fitness landscape (Stadler 1995) on any problem consisting in finding the match between a set of input data and a set of expected targets. This feature improves GP evolvability (the ability of GP to find high quality solutions) on this class of problems, since a unimodal fitness landscape makes the search much more effective with respect to the use of standard, syntax-based, genetic operators.

To have an intuition of this property, let us first consider a GAs problem in which the unique global optimum is known and the fitness of each individual (to be minimized) corresponds to its distance to the global optimum (our reasoning holds for any employed distance). In this problem, if we use, for instance, box mutation (i.e. a variation operator that slightly perturbs some of the coordinates of a solution), then any possible individual different from the global optimum has at least one fitter neighbor (individual resulting from its mutation). So, there are no locally suboptimal solutions. In other words, the error surface is unimodal, and consequently the problem is characterized by a good evolvability. Now, let us consider the typical GP problem of finding a function that maps sets of input data into known target values (as we said, regression and classification are particular cases). The fitness of an individual for this problem is typically a distance between its predicted output values and the target ones (error measure). GSOs simply define transformations on the syntax of the individuals that correspond to geometric crossover and box mutation in the semantic space, thus allowing us to map the considered GP problem into the previously discussed GA problem.

Considering that the problem addressed in this work perfectly fits the class of problems for which GSOs induce a unimodal fitness landscape, the use of GSOs is a natural choice for finding good quality solutions for the problem at hand. Here we report the definition of the geometric semantic operators for real functions domains, since these are the operators that will be used in the experimental phase.

Definition 1 (Geometric Semantic Crossover (GSC)). Given two parent functions $T_1, T_2 : \mathbb{R}^n \to \mathbb{R}$, the geometric semantic crossover returns the real function $T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2)$, where T_R is a random real function whose output values range in the interval $[0, 1]$.

Definition 2 (Geometric Semantic Mutation (GSM)). Given a parent function $T : \mathbb{R}^n \to \mathbb{R}$, the geometric semantic mutation with mutation step ms returns the real function $T_M = T + ms \cdot (T_{R1} - T_{R2})$, where T_{R1} and T_{R2} are random real functions.

Hereafter, GP that uses GSOs will be called Geometric Semantic GP (GSGP). As Moraglio et al. (2012) point out, geometric semantic operators create much larger offspring than their parents and the fast growth of the individuals in the population rapidly makes fitness evaluation unbearably slow, making the system unusable. Furthermore, while this growth produces fitter solutions, it may also be responsible for creating models that are too specialized on training data, hence generating overfitting. In Castelli et al. (2015b), a possible workaround to the problem related to the slowness of the fitness evaluation process was proposed, consisting in an implementation of these operators that makes them not only usable in practice, but also very efficient. As shown in Castelli et al. (2015b), the computational cost of evolving a population of n individuals for g generations is $O(ng)$, while the cost of evaluating a new, unseen, instance is $O(g)$. This is the implementation used in this work.

2.3 Local search optimizer in GSGP

While GSGP has been shown to produce good results (Castelli et al. 2013, 2014, 2015c, 2015e, Goncalves et al. 2015), its original formulation has several disadvantages. For instance, GSC is not effective when the semantics of the set of parents does not contain within its convex hull (i.e., does not surround) the target semantics. Similarly, since GSM produces offspring that are located randomly around the semantics of the parent, sometimes the offspring will have a worse fitness than the parent. While this is desirable when the fitness landscape is multimodal, in GSGP the landscape is unimodal so this type of exploration is not required. Furthermore, as pointed out in the previous section, the offspring produced by the GSOs will always be larger than the parents, meaning that program growth cannot be contained. The only way to control the size of the programs is to reduce the number of required applications of GSOs; i.e. reducing the number of generations required by the search.

In this section we describe how to integrate a local search (LS) strategy within GSGP. In particular, we include a local searcher within the GSM operator, since previous works have shown that GSGP achieves its best performance using only mutation during the search (Vanneschi et al. 2014; Gonçalves et al. 2015). This work follows (Castelli et al. 2015d), where a local search approach was integrated into GSM using the following construction. In particular, the GSM with LS (GSM-LS) of a tree T generates an individual:

$$T_M = \alpha_0 + \alpha_1 \cdot T + \alpha_2 \cdot (T_{R1} - T_{R2}) \tag{1}$$

where $\alpha_i \in \mathbb{R}$. Notice that α_2 replaces the mutation step parameter ms used in the definition of GSM. This in fact defines a basic multivariate linear regression problem, which can be solved, for example, by Ordinary Least Square regression (OLS). In this sense, after each mutation event, OLS is applied to the above expression to obtain the values of the model parameters $(\alpha_0, \alpha_1, \alpha_2)$ that best fit the training cases. Thus, GSM-LS is used to explore the region surrounding the tree that is to be mutated. We refer to this approach as a LS method, since our goal is to produce the best possible solution given T, T_{R1} and T_{R2}, which is the best solution possible based on these initial conditions. This allows the GSGP process to be more directed, using the target semantics to guide the search at each mutation event.

The idea of including a LS method is based on a very simple observation related to the properties of the GSOs: while these operators are effective in achieving good performance with respect to standard syntax-based operators, they may require many generations to converge to optimal solutions. Including a LS method we expect to improve the convergence speed of the search algorithm. Furthermore, by speeding up the search process, it is possible to limit the construction of over-specialized solutions that, at the end, could overfit the data.

3 Experimental phase

This section presents the data used in the experimental phase, as well as the experimental settings. In particular, Section 3.1 describes the data considered to assess the performance of the proposed system, presenting their main features. Section 3.2 gives details of the systems that have been taken into account, discussing their properties and the choice of the parameters' values.

3.1 Data

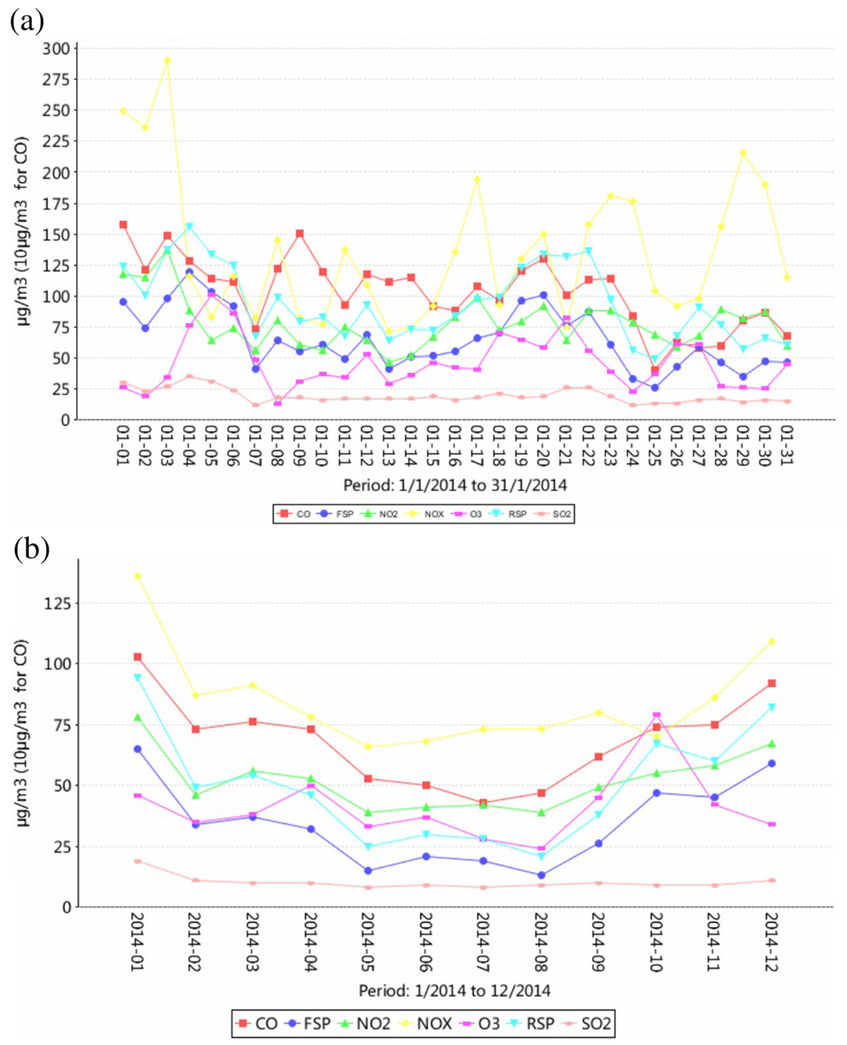
To assess the performance of the proposed system we considered data from the region of Yuen Long, China. As reported in Chan and Yao (2008), Ji et al. (2014), and Qin and Liao (2015), air quality of major Chinese cities is among the worst in the World, a consequence of three decades of double-digit economic growth with lax environmental regulation. According to the Asian Development Bank, less than 1 % of the largest Chinese cities meet the air quality standards recommended by the World Health Organization (Zhang and Crooks 2012). This severe air pollution has caused not only health problems, but also severe economic problems. According to a report of the World Bank, the monetized health costs of air pollution alone are estimated to be between 1.2 % and 3.8 % of Chinese GDP. All the reasons mentioned highlight the importance of having a system to predict, with a good level of accuracy, ozone level in such a way that human experts can timely take the adequate measures to improve air quality whenever necessary. As reported in Fig. 2a and in b, the ozone level presents a significant variation both on a daily time frame and on a monthly one. This feature highlights the difficulty of achieving an accurate prediction of the ozone level by using standard statistical techniques: the obtained solutions are usually not able to model this kind of phenomena (Breiman 2001). From this viewpoint, machine learning techniques could represent a better option to consider (Castelli et al. 2015e).

Data used in the experimental phase has been collected in 2014 by the Hong Kong Environmental Protection Department http://epic.epd.gov.hk/EPICDI/air/station/. An hourly time frame has been considered, hence the resulting dataset consists of 8760 observations. Each observation contains the concentration of different pollutants in the air and the target variable is the ozone level one hour later. In particular, the considered pollutants are the following: Carbon Monoxide (CO), Fine Suspended Particulates (FSP), Nitrogen Dioxide (NO_2), Nitrogen Oxides (NOX), Ozone (O_3), Respirable Suspended Particulates (RSP), and Sulphur Dioxide (SO_2).

3.2 Experimental settings

Several machine learning techniques have been considered in the experimental phase. In a first phase we performed a comparison between the proposed semantics-based system with a local search optimizer (LSGP) against GSGP. In a second phase, we extended the comparison by taking into account the performance of several techniques that are commonly used to tackle forecasting problems. In particular, we considered the following techniques: least square regression (SQ) (Seber and Wild 2003), Radial Basis

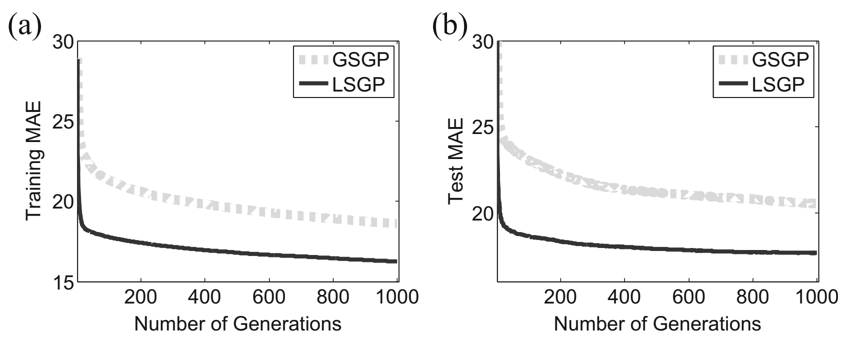
Fig. 2 Air quality: daily (**a**) and monthly (**b**) average of different pollutants for the monitoring station of Yuen Long. Pollutants: Carbon Monoxide (CO), Fine Suspended Particulates (FSP), Nitrogen Dioxide (NO_2), Nitrogen Oxides (NOX), Ozone (O_3), Respirable Suspended Particulates (RSP), Sulphur Dioxide (SO_2)



Function Network (RBF) (Haykin 1999), Isotonic Regression (ISO) (Hoffmann 2009), Multilayer Perceptron (NN) (Haykin 1999), Support Vector Machines with a polynomial kernel of first (SVM) and second order (SVM2) (Cristianini and Shawe-Taylor 2000).

Regarding the two GP systems, all the runs used populations of 200 individuals allowed to evolve for 1,000 generations. Tree initialization was performed with the Ramped Half-and-Half method (Koza 1992) with a maximum initial depth equal to 6. The function set contained arithmetic operators, including the protected division as in Koza (1992). 50 independent runs were executed for both the considered systems. Each run considers a different partition of the dataset, where 70 % of the data were used for training purpose, while the remaining 30 % as the test set. The terminal set contained 6 variables, each corresponding to a different

Fig. 3 Training and test MAE. The plots show the median over 50 independent runs



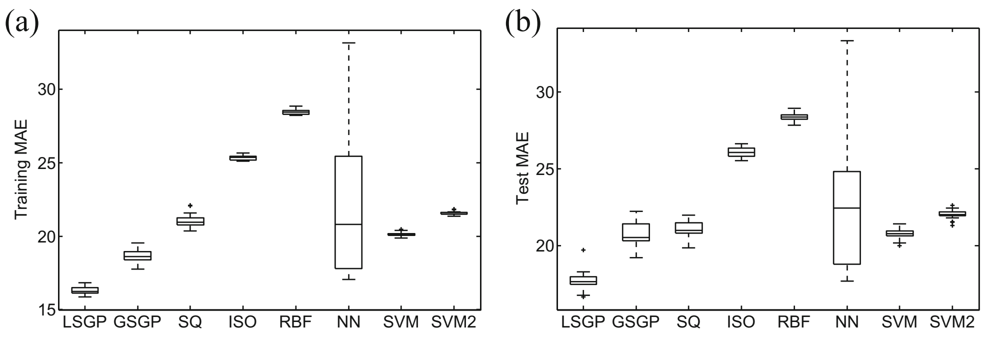


Fig. 4 Boxplots of MAE for (**a**) training and (**b**) test sets calculated over 50 runs. In each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers

feature in the dataset. Mutation and crossover probabilities have been automatically self-tuned as described in Castelli et al. (2016a). This system allows us to save the time-consuming task of tuning the GP parameters. Survival from one generation to the other was always guaranteed to the best individual of the population (elitism). For the GSM (used in GSGP) a random mutation step has been considered in each mutation event, as suggested in Vanneschi et al. (2013). The results discussed in the next section have been obtained using the GSGP implementation freely available at http://gsgp.sourceforge.net and documented in Castelli et al. (2015b).

For the non-evolutionary techniques, we used the implementation provided by the WEKA machine learning tool (Weka Machine Learning Project 2015). For these techniques, a preliminary tuning phase has been performed in order to find a satisfactory settings of the parameters. For each technique 50 independent runs have been executed and the same partitions of the dataset used in the previous phase have been considered. In particular, the tuning phase has been automatically performed by using the "multisearch" WEKA package. For a description of these meta-classifiers the reader is referred to the WEKA documentation available at http://www.cs.waikato.ac.nz/ml/weka.

We studied the performance of all the considered techniques by considering the mean absolute error (MAE). The definition of this error measure is the following:

$$MAE = \frac{1}{N}\sum_{i\in Q}|t_i - y_i| \qquad (2)$$

where $y_i = T(\mathbf{x}_i)$ is the output of the GP individual T on the input data $\mathbf{x}_i$ and t_i is the target value for instance $\mathbf{x}_i$. N denotes the number of samples in the training or testing subset, and Q contains the indices of that set.

The experimental results are discussed in the following section. For the two GP-based systems results are reported plotting the median error versus generation number on the training and test sets. In particular, for each generation we stored the error on the training and test sets of the best individual in the population. The reported curves finally plot the median of all these values collected at each generation. The median was preferred over the mean in the reported plots because of its higher robustness to outliers.

Table 1 Comparison between errors obtained on unseen examples with different techniques. For all the considered techniques we reported 10th, 25th, 50th, 75th and 90th percentile.

	LSGP	GSGP	SQ	ISO	RBF	NN	SVM	SVM2
TRAINING MAE								
10th	**16.017**	18.230	20.633	25.104	28.279	17.170	19.965	21.414
25th	**16.136**	18.398	20.758	25.179	28.324	17.813	20.049	21.505
50th	**16.254**	18.598	20.965	25.367	28.473	20.813	20.108	21.514
75th	**16.498**	18.954	21.238	25.435	28.574	25.443	20.189	21.620
90th	**16.597**	19.259	21.570	25.503	28.654	32.075	20.368	21.760
TEST MAE								
10th	**17.095**	19.922	20.537	25.723	28.081	18.496	20.227	21.540
25th	**17.485**	20.324	20.826	25.828	28.240	18.783	20.625	21.947
50th	**17.667**	20.526	21.011	26.079	28.388	22.450	20.799	22.019
75th	**17.965**	21.427	21.486	26.362	28.526	24.830	20.943	22.185
90th	**18.170**	22.088	21.789	26.645	28.737	32.859	21.146	22.406

Bold is used to denote the best (i.e., lower) median value among the considered techniques

Table 2 $P-$values returned by the Mann-Whitney statistical test

	GSGP	SQ	ISO	RBF	NN	SVM	SVM2
TRAINING							
LSGP	3.02E-11	3.00E-11	2.99E-11	2.98E-11	2.95E-11	3.00E-11	2.98E-11
TEST							
LSGP	3.69E-11	3.00E-11	3.00E-11	3.00E-11	1.52E-09	3.00E-11	3.00E-11

4 Results

We start the discussion of the obtained results by considering the performance of GSGP and LSGP. Results of this comparison are reported in Fig. 3a and b. As is possible to see in these plots, LSGP outperforms GSGP on both training and test instances. Furthermore, besides the fitness values reached at the end of the search process, it is interesting to notice how LSGP converges more quickly than GSGP. Hence, including a local searcher within the GSM operator is beneficial in speeding up the convergence of the search process and, as shown in the plots, it also results in a lower (better) training and test error with respect to GSGP. In detail, on the training instances LSGP produces a MAE of 16.254, while GSGP produces a MAE of 18.598. On the test set LSGP produces a MAE of 17.667, while GSGP produces a MAE of 20.526.

The boxplots reported in Fig. 4a and b show a comparison between the performance achieved by using the different machine learning techniques taken into account. On each box, the central mark is the median, the edges of the box are the 25th and 75th percentiles, and the whiskers extend to the most extreme data points not considered outliers. As can be observed, LSGP outperforms all other techniques considered on both training and test instances. Table 1 reports the numerical values of the median MAE for all the techniques considered, on both training and test instances, over the 50 independent runs that have been performed.

To analyze the statistical significance of the obtained results, a set of tests has been performed on the median errors. Firstly, the Kolmogorov-Smirnov test has shown that data are not normally distributed (p-value smaller than 10E-10) and hence a rank-based statistic has been applied. Subsequently, the Mann-Whitney rank-sum test for pairwise data comparison has been used (with a Bonferroni correction for the value of $\alpha = 0.05$) under the alternative hypothesis that the samples do not have equal medians. The p-values are reported in Table 2. As can be observed, LSGP produces results that are statistically better than the ones produced by the other techniques on both training and test instances.

To summarize, the proposed semantics-based GP system where the mutation operator has been redefined to include a local search optimizer is suitable for addressing the problem at hand. In particular, LSGP is able to converge in a smaller number of generations with respect to GSGP and, more importantly, to outperform existing state-of-the-art techniques used to tackle forecasting problems.

5 Conclusions

The prediction of the ozone concentration level is very important due to the negative impacts of ozone on human health. Improving the means to predict ozone concentration is thus very useful because it can provide early warnings to the population and also reduce the number of measuring sites. In this study a computational intelligence system to predict one hour ahead ozone concentration has been proposed. The system is based on geometric semantic genetic programming (GSGP), a recently proposed variant of genetic programming. While GSGP has shown its suitability in addressing real world applications over different domains, it presents an important drawback: the search process converges very slowly and this represents an important limitation for considering GSGP as a viable technique when a prompt prediction is needed. To tackle this problem and to improve the accuracy of the prediction of the ozone concentration, we proposed to couple GSGP with a local search optimizer included in the geometric semantic mutation operator. The main idea is to couple the exploration ability of GSGP with the exploitation provided by the local searcher.

The proposed system, called LSGP, has been compared against GSGP and other well-known machine learning techniques that are commonly used for addressing prediction problems. In particular, all the systems have been evaluated considering data related to the concentrations of seven environmental pollutants collected in 2014 in Yuen Long, one of the most polluted region in China. Experimental results have shown the suitability of LSGP in addressing the problem at hand. More in detail, LSGP is able to outperform all the considered techniques and, when compared against GSGP, it has shown its ability to produce a good quality model in a smaller number of generations.

Overall, the work provides two distinct and important contributions for advancing the prediction of ozone concentration: from the machine learning perspective we have shown that the inclusion of a local searcher in

GSGP is beneficial for improving the convergence speed of the search process; considering the ozone concentration prediction problem, LSGP has shown its ability in producing more accurate predictions with respect to other state-of the-art machine learning techniques.

References

Anderson, J.O., Thundiyil, J.G., & Stolbach, A. (2012). Clearing the air: a review of the effects of particulate matter air pollution on human health. *Journal of Medical Toxicology*, *8*(2), 166–175.

Ayres, J.G. (2010). The mortality effects of long-term exposure to particulate air pollution in the united kingdom. Report by the Committee on the Medical Effects of Air Pollutants.

Breiman, L. (2001). Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231.

Castelli, M., Castaldi, D., Giordani, I., Silva, S., Vanneschi, L., Archetti, F., & Maccagnola, D. (2013). An efficient implementation of geometric semantic genetic programming for anticoagulation level prediction in pharmacogenetics. In *Progress in Artificial Intelligence, Springer, pp 78–89*.

Castelli, M., Vanneschi, L., & Silva, S. (2014). Prediction of the unified Parkinson's disease rating scale assessment using a genetic programming system with geometric semantic genetic operators. *Expert Systems with Applications*, *41*(10), 4608–4616.

Castelli, M., Manzoni, L., Vanneschi, L., Silva, S., & Popovič, A. (2016a). Self-tuning geometric semantic genetic programming. *Genetic Programming and Evolvable Machines*, *17*(1), 55–74.

Castelli, M., Silva, S., & Vanneschi, L. (2015b). A C++ framework for geometric semantic genetic programming. *Genetic Programming and Evolvable Machines*, *16*(1), 73–81.

Castelli, M., Trujillo, L., Vanneschi, L., & Popovič, A. (2015c). Prediction of energy performance of residential buildings: a genetic programming approach. *Energy and Buildings*, *102*, 67–74.

Castelli, M., Trujillo, L., Vanneschi, L., Silva, S., Z-Flores, E., & Legrand, P. (2015d). Geometric semantic genetic programming with local search. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, ACM, New York, NY, USA, GECCO '15, pp 999–1006*.

Castelli, M., Vanneschi, L., & De Felice, M. (2015e). Forecasting short-term electricity consumption using a semantics-based genetic programming framework: The south Italy case. *Energy Economics*, *47*, 37–41.

Chan, C.K., & Yao, X. (2008). Air pollution in mega cities in china. *Atmospheric environment*, *42*(1), 1–42.

Corbette, J. (2013). Using information systems to improve energy efficiency: Do smart meters make a difference *Information Systems Frontiers*, *15*(5), 747–760.

Cristianini, N., & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York: Cambridge University Press.

Gonçalves, I., Silva, S., & Fonseca, C.M. (2015). On the generalization ability of geometric semantic genetic programming. In *Genetic Programming, Springer, pp 41–52*.

Haykin, S. (1999). *Neural networks: a comprehensive foundation*: Prentice Hall.

Hoffmann, L. (2009). Multivariate Isotonic Regression and Its Algorithms. Wichita State University, College of Liberal Arts and Sciences, Department of Mathematics and Statistics.

Hota, C., Upadhyaya, S., & Al-Karaki, J. (2015). Advances in secure knowledge management in the big data era. *Information Systems Frontiers*, *17*(5), 983–986.

Ji, D., Li, L., Wang, Y., Zhang, J., Cheng, M., Sun, Y., Liu, Z., Wang, L., Tang, G., Hu, B., et al. (2014). The heaviest particulate air-pollution episodes occurred in northern china in january, 2013: insights gained from observation. *Atmospheric Environment, 92*, 546–556.

Kampa, M., & Castanas, E. (2008). Human health effects of air pollution. *Environmental pollution*, *151*(2), 362–367.

Karatzas, K.D., & Kaltsatos, S. (2007). Air pollution modelling with the aid of computational intelligence methods in thessaloniki, greece. *Simulation Modelling Practice and Theory*, *15*(10), 1310–1319.

Kim, K.H., Kabir, E., & Kabir, S. (2015). A review on the human health impact of airborne particulate matter. *Environment international*, *74*, 136–143.

Kittelson, D., Watts, W., & Johnson, J. (2004). Nanoparticle emissions on minnesota highways. *Atmospheric Environment*, *38*(1), 9–19. doi:10.1016/j.atmosenv.2003.09.037.

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. USA: MIT Press, Cambridge.

Koza, J.R. (2010). Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, *11*(3-4), 251–284.

Krawiec, K., & Lichocki, P. (2009). Approximating geometric crossover in semantic space. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation, ACM, pp 987–994*.

Kumar, P., & Thiele, L. (2014). p-yds algorithm: An optimal extension of yds algorithm to minimize expected energy for real-time jobs. In *Proceedings of the 14th International Conference on Embedded Software, ACM, New York, NY, USA, EMSOFT '14, pp 12:1–12:10*. doi:10.1145/2656045.2656065.

Kumar, P., Jain, S., Gurjar, B., Sharma, P., Khare, M., Morawska, L., & Britter, R. (2013). New directions: Can a "blue sky" return to indian megacities *Atmospheric Environment*, *71*, 198–201. doi:10.1016/j.atmosenv.2013.01.055.

Li, D., Xu, L., & Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, *17*(2), 243–259.

Lim, S. et al. (2012). A comparative risk assessment of burden of disease and injury attributable to 67 risk factors and risk factor clusters in 21 regions, 1990-2010: a systematic analysis for the global burden of disease study 2010. *The Lancet*, *380*, 2224–2260.

Medina, S., Plasencia, A., Ballester, F., Mücke, H. G., & Schwartz, J. (2004). Apheis: public health impact of pm10 in 19 european cities. *Journal of Epidemiology and Community Health*, *58*(10), 831–836. doi:10.1136/jech.2003.016386.

Moraglio, A., Krawiec, K., & Johnson, C.G. (2012). Geometric semantic genetic programming, In Coello Coello, C.A., Cutello, V., Deb, K., Forrest, S., Nicosia, G., & Pavone, M. (Eds.) *Parallel Problem Solving from Nature, PPSN XII (part 1), Springer, Lecture Notes in Computer Science, vol 7491, pp 21–31*.

Qin, H., & Liao, T.F. (2015). The association between rural–urban migration flows and urban air quality in china. *Regional Environmental Change*, 1–13.

Seber, G., & Wild, C. (2003). *Nonlinear Regression. Wiley Series in Probability and Statistics*. Wiley.

Sharma, P., Sharma, P., Jain, S., & Kumar, P. (2013). An integrated statistical approach for evaluating the exceedence of criteria pollutants in the ambient air of megacity delhi. *Atmospheric Environment*, *70*(0), 7–17.

Sousa, S., Martins, F., Alvim-Ferraz, M., & Pereira, M.C. (2007). Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations. *Environmental Modelling & Software*, *22*(1), 97–103.

Stadler, P. (1995). Towards a theory of landscapes. In: Complex Systems and Binary Networks. *Lecture Notes in Physics*, *461-461*, 78–163. Springer Berlin Heidelberg.

United Nations, D.epartment.o.f.E.conomic.a.nd.S.ocial.A.airs, Population Division (2014). World urbanization prospects: The 2014 revision, highlights.

Vanneschi, L., Silva, S., Castelli, M., & Manzoni, L. (2013). Geometric Semantic Genetic Programming for Real Life Applications. In *Genetic Programming Theory and Practice XI GPTP 2013, University of Michigan, Ann Arbor, May 9-11, 2013, pp 191–209*.

Vanneschi, L., Castelli, M., & Silva, S. (2014). A survey of semantic methods in genetic programming. *Genetic Programming and Evolvable Machines*, *15*(2), 195–214.

Weka Machine Learning Project (2015). Weka. http://www.cs.waikato.ac.nz/~ml/weka.

World Health Organization (2014). Review of evidence on health aspects of air pollution.

World Health Organization (2015). Reducing global health risks through mitigation of short-lived climate pollutants.

Zhang, Q., & Crooks, R. (2012). *Toward an environmentally sustainable future: Country environmental analysis of the people's republic of China*: Report of the Asian Development Bank.

Mauro Castelli obtained his Master degree in Computer Science at the University of Milano Bicocca (Italy) in 2008 ("summa cum Laude") and his PhD at the University of Milano Bicocca in 2012. He is assistant professor at NOVA IMS, Universidade Nova de Lisboa (Portugal). His main research interests are in the field of Artificial Intelligence (in particular, Evolutionary Computation and Genetic Programming) and in the application of Machine Learning techniques to solve complex real-life problems. He has authored or coauthored more than 50 publications in the field of Evolutionary Computation and Machine Learning.

Ivo Gonçalves is a PhD candidate at the Doctoral Program in Information Science and Technology of the University of Coimbra (Portugal). He obtained his Master degree in Informatics Engineering at the University of Coimbra (Portugal) in 2011. He is an invited assistant professor at NOVA IMS, Universidade Nova de Lisboa (Portugal). His main research interests are in the fields of Artificial Intelligence, Evolutionary Computation, Genetic Programming, and Machine Learning.

Leonardo Trujillo received a degree in Electronic Engineering (2002) and a Masters in Computer Science (2004) from the Instituto Tecnológico de Tijuana. He also received a doctorate in Computer Science from the CICESE research center, in Ensenada México (2008), developing Genetic Programming (GP) applications for Computer Vision problems, focusing on feature extraction and image description. He is currently professor at Instituto Tecnológico de Tijuana (ITT) in México, where he is President of the Master Program, head of the Cybernetics research group and the head researcher of the TREE-LAB research lab. The work of Dr. Trujillo is mainly focused on developing data modelling and machine learning algorithms based on GP and evolutionary computation, with over 30 published journal papers and over 50 contributions published in top conference proceedings and edited books.

Aleš Popovič is an Associate Professor of Information Management at Faculty of Economics – University of Ljubljana and a visiting professor at NOVA IMS – Universidade Nova de Lisboa. His research interests include understanding IS value, success, and related business process change, both within and between organizations. Dr. Popovič is on the Editorial Board of International Journal of Information Management, Industrial Management & Data Systems, and International Journal of Business Intelligence Research.