Expert Systems with Applications 40 (2013) 6856–6862

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

# Prediction of high performance concrete strength using Genetic Programming with geometric semantic genetic operators

Mauro Castelli [a,b,*], Leonardo Vanneschi [a,b], Sara Silva [b]

[a] ISEGI, Universidade Nova de Lisboa, 1070-312 Lisboa, Portugal
[b] INESC-ID, IST, Universidade Técnica de Lisboa, 1000-029 Lisboa, Portugal

ARTICLE INFO

*Keywords:*
High performance concrete
Strength prediction
Artificial intelligence
Genetic Programming
Geometric operators
Semantics

ABSTRACT

Concrete is a composite construction material made primarily with aggregate, cement, and water. In addition to the basic ingredients used in conventional concrete, high-performance concrete incorporates supplementary cementitious materials, such as fly ash and blast furnace slag, and chemical admixture, such as superplasticizer. Hence, high-performance concrete is a highly complex material and modeling its behavior represents a difficult task. In this paper, we propose an intelligent system based on Genetic Programming for the prediction of high-performance concrete strength. The system we propose is called Geometric Semantic Genetic Programming, and it is based on recently defined geometric semantic genetic operators for Genetic Programming. Experimental results show the suitability of the proposed system for the prediction of concrete strength. In particular, the new method provides significantly better results than the ones produced by standard Genetic Programming and other machine learning methods, both on training and on out-of-sample data.

© 2013 Elsevier Ltd. All rights reserved.

## 1. Introduction

Concrete is a composite construction material made primarily with aggregate, cement, and water. There are many formulations of concrete, which provide varied properties, and concrete is the most-used man-made product in the world (Lomborg, 2001). Modern concrete mix designs can be complex. The choice of a concrete mix depends on the need of the project both in terms of strength and appearance and in relation to local legislation and building codes. The design begins by determining the requirements of the concrete. These requirements take into consideration the weather conditions that the concrete will be exposed to in service, and the required design strength. Many factors need to be taken into account, from the cost of the various additives and aggregates, to the trade offs between, the "slump" for easy mixing and placement and ultimate performance. A mix is then designed using cement, coarse and fine aggregates, water and chemical admixtures. The method of mixing will also be specified, as well as conditions that it may be used in. This allows a user of the concrete to be confident that the structure will perform properly. As reported in Yeh (1998), high-performance concrete (HPC) is a new terminology used in the concrete construction industry. In addition to the basic ingredients in conventional concrete the making of HPC needs to incorporate supplementary cementitious materials, such as fly ash and blast furnace slag, and chemical admixture, such as superplasticizer (Kumar, Singh, & Singh, 2012). High-performance concrete is such a highly complex material that modeling its behavior is a difficult task.

The Abrams' water-to-cement ratio (w/c) law (Abrams, 1927; Nagaraj & Banu, 1996) has been described as the most useful and significant advancement in the history of concrete technology. According to Abrams's law, the compressive strength of concrete varies inversely with the W/C ratio. Hence, an increase in the w/c decreases the concrete strength, whereas a decrease in the w/c ratio increases the strength. The implication, therefore, is that the strengths of various but comparable concrete are identical as long as their w/c ratios remain the same, regardless of the details of the compositions.

The Abrams rule implies that only the quality of the cement paste controls the strength of comparable cement. The paste quantity does not matter. Analysis of a variety of experimental data shows that this is not quite true (Popovics, 1990). For instance, if two comparable concrete mixtures have the same w/c ratio, the strength of the concrete with the higher cement content is lower (Popovics, 1990).

As reported in Yeh (1998), several studies independently have shown that concrete strength development is determined not only by the w/c ratio, but that it is also influenced by the content of other ingredients (Bhanja & Sengupta, 2005). Therefore, although experimental data have shown the practical acceptability of this rule within wide limits, a few deviations have been reported. The current empirical equations for estimating compressive strength

* Corresponding author. Tel.: +351 213828628.
*E-mail addresses:* mcastelli@isegi.unl.pt (M. Castelli), lvanneschi@isegi.unl.pt (L. Vanneschi), sara@kdbio.inesc-id.pt (S. Silva).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.06.037

are based on tests of concrete without supplementary cementitious materials. The validity of these relationships for concrete with supplementary cementitious materials (fly ash, blast furnace slag, etc.) should be investigated (Bhanja & Sengupta, 2005). The more we know about the concrete composition versus strength relationship, the better we can understand the nature of concrete and how to optimize the concrete mixture.

All these aspects highlight the need of reliable and accurate techniques that allow modeling the behavior of concrete materials.

In this paper, for the first time, we propose an intelligent system based on Genetic Programming for the prediction of the concrete strength. More in detail, in this work we use recently defined geometric semantic genetic operators for Genetic Programming. These operators allow to include the concept of semantics in the search process and have several advantages with respect to standard genetic operators used in Genetic Programming.

The paper is organized as follows: Section 2 introduces basic concepts about Genetic Programming with a particular focus on the standard operators used in the evolutionary process; Section 3 presents the geometric semantic operators used in this work and outlines some of their properties that have a direct effect on the search process; Section 4 presents a literature review on using computational intelligence methods in simulating the behavior of concrete materials; Section 5 describes the data used in this work, the experimental settings and proposes an accurate discussion and analysis of the results; Section 6 concludes the paper summarizing the results that have been achieved.

## 2. Genetic Programming

Models lie in the core of any technology in any industry, be it finance, manufacturing, services, mining, or information technology. The task of data-driven modeling lies in using a limited number of observations of system variables for inferring relationships among these variables. The design of reliable learning machines for data-driven modeling tasks is of strategic importance, as there are many systems that cannot be accurately modeled by classical mathematical or statistical techniques. Reliable learning in the field of Machine Learning (ML) revolves around the notion of generalization, which is the ability of a learned model to correctly explain data that are drawn from the same distribution as training data, but have not been presented during the training process, and it is this very important property that ML algorithms aim to optimize. Genetic Programming (GP) (Koza, 1992; Poli, Langdon, & Mcphee, 2008) is one of the youngest paradigms inside the computational intelligence research area called Evolutionary Computation (EC) and consists in the automated learning of computer programs by means of a process mimicking Darwinian evolution. GP tackles learning problems by means of searching a computer program space for the program that better respects some given functional specifications.GP is an evolutionary computation technique that automatically solves problems without requiring the user to know or specify the form or structure of the solution in advance. At the most abstract level GP is a systematic, domain-independent method for getting computers to solve problems automatically starting from a high-level statement of what needs to be done. In GP a population of computer programs is evolved. That is, generation by generation, GP stochastically transforms populations of programs into new, hopefully better, populations of programs. The search is performed using an EA. The recipe for solving a problem with GP is as follows:

- Chose a representation space in which candidate solutions can be specified. This consists of deciding on the primitives of the programming language that will be used to construct programs. A program is built up from a terminal set (the variables in the problem) and a function set (the basic operators).
- Design the fitness criteria for evaluating the quality of a solution. This involves the execution of a candidate solution on a suite of test cases, reminiscent of the process of black-box testing. In case of supervised learning, a distance-based function is employed to quantify the divergence of a candidate's behavior from the desired one.
- Design a parent selection and replacement policy. Central to every EA is the concept of fitness-driven selection in order to exert an evolutionary pressure towards promising areas of the program space. The replacement policy determines the way in which newly created offspring programs replace their parents in the population.
- Design a variation mechanism for generating offspring from a parent or a set of parents. Standard GP uses two main variation operators: crossover and mutation. Crossover recombines parts of the structure of two individuals, whereas mutation stochastically alters a portion of the structure of an individual.
- After a random initialization of a population of computer programs, an iterative application of selection-variation-replacement is employed to improve the programs quality in a stepwise refinement way.

In GP learning, both terms of program and model refer to the same entity and will be used interchangeably. Supervised learning of regression models in GP relies on the extraction of implicit relationships that may exist in the input–output mappings specified by the training examples. The discovered relationships are expressed in symbolic form, which is traditionally represented by an expression-tree structure. For a complete introduction to GP the reader is referred to Koza (1992). In this work we used genetic operators that, diversely from the canonical ones, are based on the concept of semantics that will be introduced in the next section. To understand the differences between the genetic operators used in this work (described in Section 3) and the ones used in the standard GP algorithm, the latter are briefly described.

*The "Standard" Crossover Operator*. The crossover operator is traditionally used to combine the genetic material of two parents by swapping a part of one parent with a part of the other. More specifically, tree-based crossover proceeds by the following steps:

- Choose two individuals as parents, based on mating selection policy (the better the fitness of the individual the higher its probability of being selected).
- Select a random subtree in each parent. The selection of subtrees can be biased so that subtrees constituting terminals are selected with lower probability than other subtrees.
- Swap the selected subtrees between the two parents. The resulting individuals are the children.

This process is shown, using two arbitrary simple Boolean expressions as parents, in Fig. 1.

*The Mutation Operator*. The mutation operation introduces random changes in the structures of the individuals in the population. While there are many different mutation operators, here subtree mutation (that is the most well-known one and also the one used in this work) is described. Subtree mutation begins by selecting a point at random within the "parent" tree. Then, it removes whatever is currently at the selected point and whatever is below the selected point and 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 is to be inserted. This parameter typically has the same value as the parameter for the maximum size of trees in the initial population. This process is shown in Fig. 2, where an arbitrary simple Boolean expression is mutated by replacing one of its subtrees with a random tree.

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
✓ امکان دانلود نسخه ترجمه شده مقالات
✓ پذیرش سفارش ترجمه تخصصی
✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
✓ دانلود فوری مقاله پس از پرداخت آنلاین
✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات