Qualitatively Constrained Equation Discovery

Jure Žabkar and **Aleksander Sadikov** and **Ivan Bratko** and **Janez Demšar**
Artificial Intelligence Laboratory, Faculty of Computer and Information Science,
University of Ljubljana, SI-1000 Ljubljana, Slovenia

Abstract

Equation discovery is a very lively area of artificial intelligence which deals with explaining phenomena by mathematical formulae induced from the data. One successful approach to the problem are algorithms which construct thousands of formulae and report the simplest ones with the best fit to the data. Another, sub-symbolic, fits (piecewise) regression hyper-planes; their advantage is that they may be made to conform to qualitative constraints. We propose an algorithm that shares the qualities of the two approaches: EDGAR searches for simple qualitatively faithful equations which fit the data well. The algorithm performs very well on simple problems, but in its current implementation fails to solve more complex ones.

Introduction

The field of equation discovery can be defined as "given a set of (numerical) observations, find a set of laws, expressed as mathematical equations, which govern the observed system". An amazing example of such a venture is Kepler's use of Brahe's data to discover the rules of planetary motion. The task is far easier if the researcher knows what he is looking for, that is, if he wants to discover the relation between a set of independent variables and a dependent variable.

The physicist's approach is to derive a new law from the known laws. For instance, the motion of planets is a direct consequence of Newton's universal laws of gravity. This will fail when the laws are not there yet (like they were not in Kepler's time) or, as is more often the case nowadays, if the domain is too complex, non-linear, or has too many variables to be analytically solvable. A typical example of such a problem is modeling weather. In such cases, the expert may be able to come up with an approximate model where the constants are fit to the existing data by applying statistical methods like minimization of squared error. When the domain is not understood well enough, even this may be unfeasible. Some well known examples of this kind occur in ecological modeling (Langley *et al.* 2002; Kompare, Todorovski, & Džeroski 2001).

Machine learning and statistics offer two alternatives. One is to generate numerical models, such as piecewise linear regression or LOESS (Cleveland, Devlin, & Grosse 1998). The property of this approach, which is of particular interest for our paper, is that it can be implemented to also conform to qualitative constraints (Šuc, Vladušič, & Bratko 2004) given by an expert or by algorithms like QUIN (Bratko & Šuc 2003) or Padé (Žabkar, Bratko, & Demšar 2007). These models may be very accurate, but they are useful only for *predicting* and not *explaining* the domain and so fail to fulfill our goal of finding "a set of laws governing the system".

The alternative, symbolic models are better in this respect. Algorithms like Goldhorn (Križman, Džeroski, & Kompare 1995) and Lagramge (Todorovski & Džeroski 1997) produce a number of equations with missing constants, fit the constants to the data and rank the equations by their simplicity and fit. For better control, they may also allow the expert to define a grammar for the equations (Todorovski & Džeroski 1997; Langley *et al.* 2002).

The problem with symbolic models is their ignorance of qualitative constraints, which can lead to meaningless results. For a simple test we modeled the free fall acceleration at different distances from the Earth. The correct equation (if the experiment is done above the Earth surface) is

$$g = G\frac{M}{r^2} = \frac{3.99 \times 10^{14}}{r^2}$$

where G is the gravitational constant, M is the Earth's mass and r is the distance from the Earth's center at which we measure the acceleration.

We generated artificial experimental data by sampling the function $g(r)$ with step 200 in the interval $[6371, 39971]$ (from the ground to the height of satellites) obtaining 169 samples. We added Gaussian noise with $N(0, 0.5)$. We tried to reconstruct the formula as a linear combination of terms obtained by generating all subsets of elementary functions

$$\{1, r, r^{-1}, r^2, r^{-2}, r^3, r^{-3}, \sin r, \log r, \cos r, \exp r\},$$

i.e. we were fitting the coefficients of functions like $a + br^2 + \sin r$ and $a \log r + br^{-2}$. We sorted the functions using the state-of-the-art combination of root mean squared error (RMSE) and minimum description length (MDL) measures from (Todorovski & Džeroski 1997). The optimal fit was a constant function, and the second best fit was (RMSE=0.5013):

$$g(r) = \frac{3.928 \cdot 10^{14}}{r^2} - 0.124\cos(r).$$

The first term is quite correct, while the second term only fits the (random) noise. The problem with this solution is that it suggests that free fall acceleration oscillates with r — which we (today) know is not true. The obvious remedy to this problem is to exclude the sine and cosine from the list of base functions. We can also tune the scoring function's bias on description length, but this can only be done if we know the correct formula in advance. Besides, the emphasis on MDL may already be too high, as witnessed by the fact that the best ranked function is simply a constant.

In this paper we propose a new algorithm, EDGAR, that offers a third approach, combining the advantages of numeric and symbolic approaches: it searches for symbolic equations by fitting the template functions constructed as a combination of terms (like in the example above) or from a grammar given by the expert, but at the same time also ensures that the solutions match the prescribed qualitative constraints.

Algorithm EDGAR

EDGAR (Equation Discovery with Grammars And Regression) is an algorithm for discovery of equations from a set of measurements of independent and dependent variables, a set of qualitative constraints, and the grammar specifying the templates of equations. The constraints may also specify a region, like in "y increases with x for all positive values of x". The algorithm consists of the following four steps.

1. Use a function generator to generate general forms of functions (templates). For instance, $a + bx + cx^2$ is a template for second degree polynomials in x.
2. Compute a symbolic derivative of each generated function, *e.g.*
$$\frac{\partial(a + bx + cx^2)}{\partial x} = b + 2cx.$$
3. Symbolically solve the system that puts the constraints on the coefficients of the initial function, respecting the qualitative constraint. For instance, if we know (from an expert or a qualitative model) that the function increases with x for all positive x, the algorithm needs to find the values of b and c which satisfy
$$\forall x, x > 0 : b + 2cx > 0.$$
The solution is:
$$(b = 0 \wedge c > 0) \vee (b > 0 \wedge c \geq 0).$$
4. Finally, fit the coefficients of the function to minimize RMSE, with respect to the constraints on the coefficients that were computed in the previous step to guarantee that the induced function will satisfy the given qualitative constraints. For instance, the algorithm would find the values of a, b and c within $(b = 0 \wedge c > 0) \vee (b > 0 \wedge c \geq 0)$, for which $a + bx + cx^2$ fits the data as close as possible.

For the first step, the algorithm currently supports two forms of specifying the function templates. One is to provide a set of elementary (basic) functions from which we can automatically generate candidate functions for further processing, like we did in the example in the introduction. For instance $\{1, x, x^2\}$ is used to generate all possible second degree polynomials. The alternative is to use context free grammars to generate candidate functions. This approach has several advantages over the first one, among them offering a simple way for the user to provide background knowledge and the use of declarative bias (Todorovski & Džeroski 1997).

The second step, computing the symbolic derivative of the function from the previous step, is trivial.

The overall simplicity of the idea is unfortunately spoiled by the extremely difficult realization of the third step. Its task translates to the problem of quantifier elimination and is generally insolvable. We used the state-of-the-art algorithms coded in Mathematica's (Wolfram Research, Inc. 2005) function Reduce. For polynomials, it uses cylindrical algebraic decomposition (Collins 1975). Algebraic functions are translated into equivalent purely polynomial systems. For transcendental functions, Reduce generates polynomial systems composed with transcendental conditions, then reduces these using functional relations and a database of inverse image information. Piecewise functions are symbolically expanded to construct a collection of continuous systems. The user can also help by adding some background knowledge into the logical formula.

The remaining step, minimization of RMSE given the constraints from the previous step, is generally a nonlinear constraint satisfaction problem, which we solve using Nelder-Mead methods (Luersen & Le Riche 2002).

The first step of the algorithm was partially implemented in Prolog. Everything else was implemented in Mathematica, which already contains the derivation, methods for quantifier elimination, and nonlinear minimization.

Experiments and Discussion

We tried the algorithm on the problem of modeling the gravitational acceleration with artificial data generated as described in the introduction. The Gaussian noise was again N(0, 0.5). We generated the function templates with a grammar that can induce symbolic rational functions up to the second order, e.g.: $ax^2 + bx\sin(c + dx)$, or $ax/[\sin(b + cx) - dx]$. The sine terms were included only for the sake of comparison, although it was obvious that all functions with such terms would be discarded in the third step. As a qualitative constraint, we told EDGAR that the gravitation decreases with the distance, $g = Q(-r)$.

The generated function with the optimal RMSE was
$$g(r) = -0.0259 + \frac{4.096 \cdot 10^{14}}{r^2}$$
with a RMSE of 0.4968.

Acting as domain experts, we noted that the formula, despite obeying the given qualitative constraints, still made no physical sense, since the negative term reverses the sense of gravitation for distances above 125,000 kilometers.

EDGAR makes it easy to add new constraints. We thus additionally stated that $g(r)$ should always be positive, which reported
$$g(r) = \frac{4.070 \cdot 10^{14}}{r^2}$$

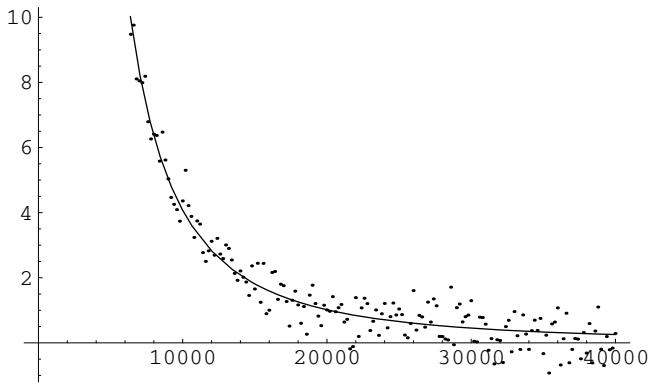

Figure 1: Best fit by EDGAR with enforced $Q(-r)$ and $\forall r : g(r) > 0$.

as the best ranked function with a RMSE of 0.4972 (see Fig. 1). This function is correct, except for the 3.6% error in the constant due to the noise.

We repeated the experiment with different amounts of noise: N(0, 1) and N(0, 0.2). EDGAR's results were the same (correct) as in the experiment with N(0, 0.5), except for the constant slightly varying due to different amounts of noise in the data. On the other hand, RMSE alone always selected an overly complex overfitted function, and adding MDL to the scoring function resulted in always preferring a constant as a solution.

Yet, despite this success — and a few others on similarly simple domains, for instance on the XPERO robot data described in (Žabkar, Bratko, & Demšar 2007) — there remains a lot of further work to make the algorithm practically useful. We describe the problems and our proposed solutions below.

Depending on the complexity of the templates (or, more accurately, their derivatives) the task of the third step may be too complex. In the current implementation, this would result in a suboptimal, yet still qualitatively faithful solution. We are working on replacing the Reduce function with probabilistic alternatives.

When the solution includes periodic functions, these can generate a lot of local minima, which the minimization procedure can fall into. We do not yet know whether this will cause any real problems and whether restarting the minimization from different initial points will amend them.

The algorithm needs a few minutes on an average PC for solving rather simple problems (gravitational acceleration, XPERO robot data) and does not seem to scale well. This is again due to the complexity of the Reduce function. Besides replacing it, the algorithm can also be accelerated by using exact or heuristic methods to eliminate as many functions as possible before they reach the third step of the algorithm.

Conclusion

We described an algorithm called EDGAR which discovers symbolic equations that fit the given data as well as possible and, at the same time, match the given qualitative constraints. The algorithm is conceptually simple and was easy to implement using the existing functions for derivation, quantifier elimination and minimization available in Mathematica. The successful tests on a few simple domains show the algorithm as promising, yet there remain quite a few technical problems to be solved before it will also be practically useful.

Acknowledgments

This work was supported by the Slovenian research agency ARRS, and by the European project XPERO: Learning by Experimentation (IST-29427).

References

Bratko, I., and Šuc, D. 2003. Learning qualitative models. *AI Magazine* 24(4):107–119.

Cleveland, W.; Devlin, S.; and Grosse, E. 1998. Regression by local fitting. *Journal of Econometrics* 37:87–114.

Collins, G. E. 1975. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. *In Lecture Notes In Computer Science* 33:134–183.

Kompare, B.; Todorovski, L.; and Džeroski, S. 2001. Modeling and prediction of phytoplankton growth with equation discovery : Case study - Lake Glumsø, Denmark. *Verh. - Int. Ver. Theor. Angew. Limnol.* 27:3626–3631.

Križman, V.; Džeroski, S.; and Kompare, B. 1995. Discovering dynamics from measured data. *Electrotechnical Review* 62:191–198.

Langley, P.; Sanchez, J.; Todorovski, L.; and Džeroski, S. 2002. Inducing process models from continuous data. In *Proceedings of The Nineteenth International Conference on Machine Learning*.

Luersen, M. A., and Le Riche, R. 2002. Globalized Nelder-Mead method for engineering optimization. In *ICECT'03: Proceedings of the third international conference on Engineering computational technology*, 165–166. Edinburgh, UK: Civil-Comp press.

Todorovski, L., and Džeroski, S. 1997. Declarative bias in equation discovery. In *In Proceedings of the 14th International Conference on Machine Learning*.

Šuc, D.; Vladušič, D.; and Bratko, I. 2004. Qualitatively faithful quantitative prediction. *Artificial Intelligence* 158(2):189–214.

Žabkar, J.; Bratko, I.; and Demšar, J. 2007. Learning qualitative models through partial derivatives by Padé. In *Proceedings of the 21st International Workshop on Qualitative Reasoning (submitted)*.

Wolfram Research, Inc. 2005. *Mathematica, Version 5.2*. Champaign, IL, USA.