Feature Space Transformation using Equation Discovery

Freek Stulp, Mark Pflüger, Michael Beetz

Intelligent Autonomous Systems Group, Technische Universität München, Munich, Germany
{stulp,pflueger,beetz}@cs.tum.edu

1 Motivation

In Machine Learning, the success and performance of learning often critically depends on the feature space provided. For instance, when learning a classifier $f_1, \ldots, f_n \rightarrow c$ that maps features $f_1, \ldots, f_n$ to classes c, appropriate encodings of $f_1, \ldots, f_n$ are often as important as the choice of learning algorithm itself.

This is particularly true for robot learning, where feature spaces are typically high dimensional and features are continuous. Consider an example from a robotic navigation, depicted in Figure 1. In [1], the robot learns to predict navigation execution duration given the current and goal pose, from observed experience. By exploiting translational and rotational invariances, the original 6-dimensional state space can be reduced to 3 dimensions. The benefit is that fewer examples are needed to learn an accurate prediction model.

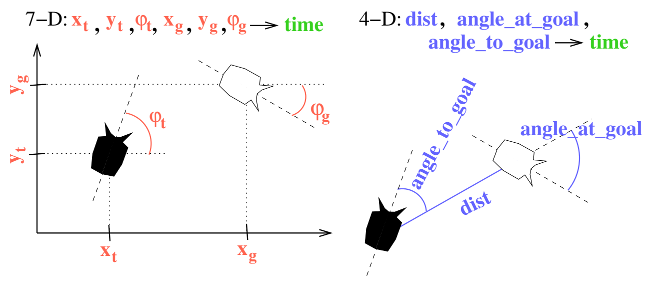
7–D: x_t, y_t, φ_t, x_g, y_g, φ_g → time
4–D: dist, angle_at_goal, angle_to_goal → time
y_g
y_t
φ_t
φ_g
x_t
x_g
angle_to_goal
angle_at_goal
dist

Fig. 1. Original and transformed feature spaces for a robotic navigation task. The 3-dimensional space is more appropriate for learning, as it contains the same information yet is more compact.

For many applications, it is common to design feature spaces manually. State variables are composed into higher level features using domain-specific knowledge. Unfortunately, manually designing these feature languages is tedious, because each new learning problem usually needs its own customized feature space. It is also error-prone, as relevant information in the original state space might be lost in the transformation. To overcome these problems, we propose an algorithm that automatically generates compact feature spaces, based on Equation Discovery.

2 Combining Equation Discovery and Machine Learning

Our feature space generation algorithm is based on *Equation Discovery* (ED). ED systems introduce new variables from a set of arithmetical operators and functions. The algorithm explores the hypothesis space of all equations, restricted by heuristics and constraints. A classical representative is BACON [2], which rediscovered Kepler's law ($T^2 = kR^3$). A graphic example can be seen to the left in the figure below, in which five input variables are mapped to the target by the equation $t = |i1| + (i2/i3) + \sqrt{i5}$. The advantage of ED is that it yields a compact representation and human readable output. For instance, would the simplicity and elegance of Kepler's law be obvious from the learned weights in a neural network? However, the equations are restricted by the operators provided, in contrast to for instance neural networks, which can learn complex non-linear relationships.

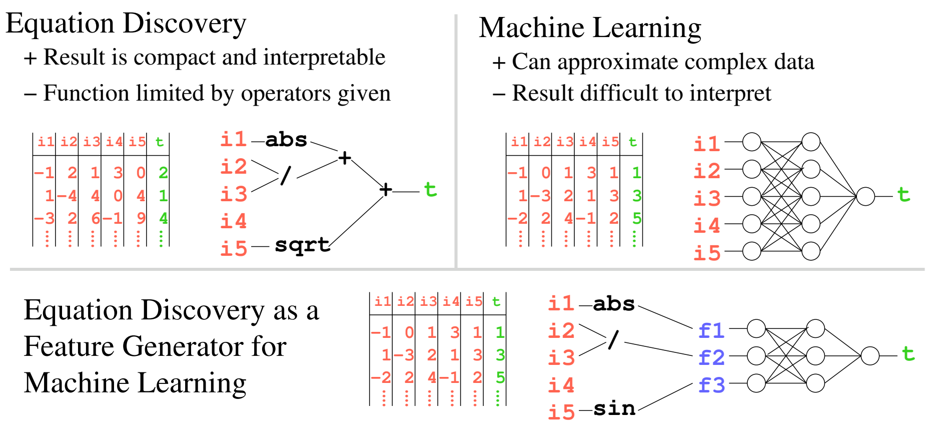

Fig. 2. Combining Equation Discovery and Machine Learning

Our novel approach combines the strengths of ED, being the compactness and interpretability of the resulting function, and ML, being its ability to approximate complex data. We do this by allowing ED to discover many equations, which, when applied to the input data, yield data that has a higher correlation with the target data. ED is halted at a certain depth, and from the multitude of generated equations (features), those most appropriate for learning are selected. The algorithm essentially searches for relationships between several input variables and the target variable that can be described well with operators, and leaves more complex relationships to machine learning.

The algorithm combines all n initial features with the k given operators, yielding new equations. These new features are added to the original set. This is repeated recursively d times, yielding equations with at most $2^{(d-1)}$ operators. Since the complexity of this algorithm is $\Theta(k^{2^d-1}n^{2^d})$, we should avoid generating irrelevant features. Mathematical constraints eliminate equations that generate neutral elements (e.g. x/x, x-x). Term reduction removes terms with the same semantics but different syntax (e.g. $x * 1/y = x/y$). Units are respected to avoid for example subtracting meters from millimeters, or meters from seconds. Furthermore, domain dependent operators can further control search. For example, in a geometrical domain it makes sense to add trigonometric operators and constraints how to use them, such as "apply $atan$ only to two distances".

We further direct search by choosing only features that predict the target value well. This is done by computing the linear correlation coefficient r of the feature with the target value. This approach is fast, but suffers the same problems as other filter methods [3]. At each depth, only the best $p\%$ of features are added to the set for further processing.

3 Results

First we evaluated our system with *Equation Rediscovery.* This is the process of discovering an equation from data generated by a known target equation. For every term containing up to seven operators, thirty random equations were generated and tested. Equations with up to two operators are always discovered, equations with three operators in most cases and everything above depends a lot on the structure of the term. An example of a short equation that was not discovered is $(w/(x * y)) - z$, while $f0 * f1 + f2 * f3 + f4 * f5 + f6 * f7 + f8 * f9$ was found. The runtime in any case is less than a minute on an x86 computer with 1500 MHz processor clock speed.

Prediction Models for Robots. As a real-world example we used the example from the introduction in which robots need to learn to predict expected action durations. The goal was to discover the Euclidean distance to the final position and the angle between the starting position and the shortest path and between there and the final position. Geometrical constraints and operators were added to guide search. Note that these are not *problem*, but *domain* specific. This means they are much more general, and will hold in many problems in the same domain.

These first results indicate that this approach can guide and sustain the design of feature languages for new problems. Nonetheless, some domain knowledge is still necessary when constructing sensible features for complex real-world tasks. Future work aims at processing nominal as well as numerical input values, and implementing more sophisticated relevance measures than linear correlation coefficient.

References

1. Stulp, F., Beetz, M.: Optimized execution of action chains using learned performance models of abstract actions. In: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI). (2005)
2. Langley, P., Simon, H., Bradshaw, G., Zytkow, J.: Scientific Discovery: Computational Explorations of the Creative Processes. MIT Press (1987)
3. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proceedings of International Conference on Machine Learning. (1994) 121–129