A first approach to generating programs from data

Irvin Hwang

August 29, 2010

1 Learning to draw: A concrete problem

Given a set of images of triangles and primitives of a language how can we learn a program, which draws triangles.

2 Patterns, Expressions, and Models

If we want flexibility we need to be able to define new models. The problem is the space of expressions we can search over when finding a representation is huge. One possible approach is to use the power of abstraction.

3 Top Level Algorithm

We illustrate the high-level approach using the drawing problem. The setup begins with the learner having a language

$$
\begin{aligned}
exp ::= & \texttt{draw-pixel } exp\ exp \\
| & \texttt{True} \\
| & \texttt{False} \\
| & \texttt{If } exp\ exp\ exp \\
| & \texttt{while } exp\ exp \\
| & \texttt{Unknown} \\
| & \texttt{Act} \\
| & \texttt{NoAct} \\
| & \texttt{On} \\
| & \texttt{Off} \\
| & \texttt{Action} \\
| & \texttt{Feature} \\
| & \texttt{==} \ exp\ exp
\end{aligned}
$$

that consists of the draw-pixel, (a set of expressions or models) available to it. The high-level idea consists of three stages. In the first stage one tries to fit current models to the data. is to try and explain the data with the current models available. Then connect the best models together using the language as a glue. Given several models of the same object we can form more abstract models by finding common subtrees among instances.

3.1 Fitting Data Models

3.2 Gluing Models Together

3.3 Abstraction