predix: Understanding Desktop Users

Practical Applications of Machine Learning for Predicting User Action

Ian Cohen, James Walker, Faculty Advisor: Lyle Ungar

Abstract

In this project we attempted to employ standard machine learning methods on the Linux desktop with the goal of answering the question, given the user’s past behavior, *What will the user do next?*

It was intended as a sister project to gnome Do, which can use our output as a component in their ranking scheme and provide us with on-line feedback.

What we’re dealing with

**gnome** is the leading Linux desktop suite. It is highly portable and customizable, and contains only open-source applications, making it a great candidate for usage-logging. **gnome** follows many modern desktop conventions, such as application menus and shortcuts, known collectively as the “panel.”

**D-Bus**

**gnome** includes an underutilized interprocess communication service known as **D-Bus**. Applications can subscribe to signals and make method calls on other applications using D-Bus proxy objects exposed as a sort of API. This is how predix collects data and answers queries.

The Design

![Diagram of predix system]

**D-Bus**

**gnome Do**

Panel

Firefox

**Signal Passing**

**Prediction**

**Predix proxy**

**Context**

**Logger**

**Online Learning**

**Orange**

**Offline Learning**

**Features**

**Addins**

**Machine Learning**

We tried a variety of algorithms to predict relevant user actions and objects. The simplest and possibly most effective is titled the *Incremental Probably Action Modeling* algorithm, which is an online Markov model.

We built a framework for adding new features in a simple way, given objects encapsulating the event and context data. We automatically generate meta-features such as *n*-grams of specified features.

Conclusion

Additionally, a third party developer can reveal new aspects of the **gnome** environment to our data collecting system. The code is as simple as in the following complete example which reveals the name of the focused desktop window:

```python
import Xlib.display
from predix.Addins import context_method
display = Xlib.display.Display()

@context_method
def focused_window_name():
    window = display.get_input_focus().focus
    if isinstance(window, int):
        return window
    else:
        return window.get_wm_name()
```

**predix** is a fully functioning machine learning desktop prediction system, fully extensible by third parties, and ready for professional testing on large data sets with fancier plugins. We are very excited at the prospect of determining which algorithms perform better on real data. **predix** makes for a great topic of future research, and could be extended in several interesting ways.