A2

I. Toward an adaptive command line interface

This articles introduces five different mechanisms (Omniscient, C4.5, Prefix, MFC and MRC) to predict the next possible command line input. The experiement is conducted based on 77 people's command histories. After collecting the data, the researchers find that C4.5 is the best of them, which has an average accuracy up to 45% and saves about 33% keystrokes typed. Based on the result, researchers implement an imperfect prototype `ilash` (Inductive Learning Apprentice Shell). But this prototype is not included in the experiment because they believe the required commitment to a single method contradicts the idea that they want to explore various methods, and the system's predictions on the interface could influence user's succeeding actions. In addtion, the best method, C4.5 is implemented with software outside shell. Since our group is studying the overall learning experience of CLI, we belive an adaptive command line interface is quite user-friendly especially to beginners, and the introduced mechanisms could be useful when we build our own prototype.

II. Predicting sequences of user actions

Basically, this article demonstrates an idea that, through repetitve application of a suitable algorithm, the patterns of user actions can be captured and thus the prediction of next action would seem more reliable. During the research, the authors also reveal an interesting fact that people tend to repeat themselves, consequently, the new commands only take very limited portion of total commands. By combining earlier results, they also believe that even the 'previously best' method C4.5 has its own drawbacks such as non-incremental, needs to be run outside of the command prediction loop and takes more time and memories. Then an algorithm named `Ideal Online Learning Algorithm (IOLA)` is introduced to readers with nine features and ideally suits for incorporation into many types of user interfaces. Consdering this algorithm, the `Incremental Probabilistic Action Modeling` performs about as accuracy as C4.5, but together with more advantages. For example, it's incremental, it can generate top-*n* predictions instead only one in C4.5 and last but not least, it runs faster than C4.5. Therefore, IPAM might be considered as a more competitive candidate for CLI prediction.

III. Learning to personalize

Comparing with previous two articles, this one is more theoratical. It emphasizes the recognition part in machine-learning algorithms, which decides 'what to learn' in those 'self-customizing software'. The authors suggest those routine processes and stereotypical sequences should be recognized by computer systems. Moreover, they also highlight a claim that due to differences between users, the prediction should not be settled and made ahead of time, but modified based upon users' personal preferences. In other words, the learning is not an instant action but has to be done in a long term gradually. Additionally, the collaboration is also believed to be vital in machine learning, i.e., the prediction should be suggested with some decent consideration in other users' behaviors in a similar scenario. In a nutshell, this article states some general ideas of how to make computer prediction 'personalized'. And this could be used in our prototype so that ours is able to offer related prediction of commands more efficiently and more accurately.

IV. Summary

When talking about learning CLI, one of the common problems people would encounter is remember every commands in details. So we think a command line interface with predictive function would be encouragingly handy for beginners. And these three articles not only introduce six different mechanisms for this purpose, but also show that how to improve such mechanisms.

note

The first reference, though not summarized, is still strongly related to the following two articles. It's because the former indicates an explicit data collection progress, whose result plays an important role in the latter two.

Davison, B. D., & Hirsh, H. (1997, August). Experiments in UNIX command prediction. In *AAAI/IAAI* (p. 827).

references

Davison, B. D., & Hirsh, H. (1997). Toward an adaptive command line interface. *HCI (2)*, 505-508.

Davison, B. D., & Hirsh, H. (1998, July). Predicting sequences of user actions. In *Notes of the AAAI/ICML 1998 Workshop on Predicting the Future: AI Approaches to Time-Series Analysis* (pp. 5-12).

Hirsh, H., Basu, C., & Davison, B. D. (2000). Learning to personalize. *Communications of the ACM, 43*(8), 102-106.