Understanding Computer Usage Evolution

David C. Anastasiu[†] and Al M. Rashid[‡] and Andrea Tagarelli[§] and George Karypis[†]

[†]Karypis Lab, Computer Science & Engineering, University of Minnesota, Twin Cities, U.S.A.
[‡]Intel Corporation, 1900 Prairie City Rd., Folsom, CA 95630, U.S.A.
[§]University of Calabria, Rende, Italy

http://cs.umn.edu/~dragos/orion

University of Minnesota
Supercomputing Institute

Introduction

- **Objective:** Analyze longitudinal user activity data in order to model and characterize changes in user behavior.
  - Focused on understanding how the usage of applications on personal computers (PCs) evolves over time.
- **Why?**
  - Identify user populations exhibiting different patterns of evolution.
  - Explain how external factors influence usage evolution.
  - Suggest required capabilities for future PC generations.
- **What?** Available data for each user's PC:
  - Daily summary statistics for executed applications, including execution CPU time, I/O time, number of times launched.
  - PC details, including system type, CPU type, rough geolocation.
- **How?**
  - We segment the application level usage of different users into a sequence of prototypical usage patterns (protos).
  - Using an iterative process, protos are automatically derived from the usage segmentation, and an optimal segmentation is determined from the protos.

What is computer usage evolution?

- Computer usage can be described by the distribution of execution CPU time across application categories.
- Usage evolution captures changes in this distribution.
- Characterizing usage evolution describes how the actions of user groups change over time.

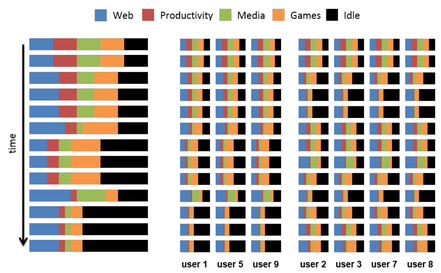

Figure: Computer usage evolution: a user's sequence of PC usage vectors (on the left), and sequences of two similar sets of users (on the right).

Characterizing usage evolution

- We follow a segmentation based approach:
  - Partition a user's usage sequence into disjoint consecutive sets of observations (segments) such that the usage in each segment remains fairly consistent.
  - Let $\langle \mathbf{w}_1, \ldots, \mathbf{w}_n \rangle$ be a sequence of usage vectors for a given user.
  - A segmentation into m segments optimizes a function of the form:

$$\min_{s_*, \mathbf{p}_l} \sum_{l=1}^{m} \sum_{j=s_{l-1}+1}^{s_l} ||\mathbf{w}_j - \mathbf{p}_l||^2.$$

  - The proto vector $\mathbf{p}_l$ captures the consistent usage during $\langle \mathbf{w}_{s_{l-1}+1}, \ldots, \mathbf{w}_{s_l} \rangle$.

Modeling assumptions

- Different users exhibit a rather small number of prototypical usage behaviors, captured by the protos.
- The usage behavior of users remains consistent over a certain period.
- The usage behavior of users can change from one prototypical behavior to another.

Orion: Cross-user usage segmentation

- Input:
  - Sequences of usage vectors of a set of users.
  - A predefined number of protos.
- Output:
  - A segmentation of the sequences of all users such that the error associated with modeling each segment by one of the protos is minimized,

$$\min_{s_*, m_*, \mathbf{p}_*} \sum_{i=1}^{n} \sum_{l=1}^{m_i} \sum_{j=s_{i,l-1}+1}^{s_{i,l}} ||\mathbf{w}_{i,j} - \mathbf{p}_{i,l}||^2. \quad (1)$$

- Iterative algorithm, whose iterations consist of two phases:
  - Segmentation identification:
    - Given the set of protos, identify the segmentation that minimizes the total error.
    - Uses a dynamic-programming algorithm to find the optimal segmentation. Complexity: $O(\#users \times \mu^2 \times \#protos)$, where μ is the average sequence length.
  - Optimal proto identification:
    - Given the segmentation, identify the protos that minimize the total error.
    - New proto is the mean of the usage vectors spanned by the proto.
- Initialization:
  - The initial protos are determined by performing a K-means clustering of all usage vectors across all users.
- Robustness:
  - Minimum length constraints on each segment.
  - A penalty associated with the creation of each additional segment within a user's sequence.
    - A segment is allowed to be created if it leads to a user-specified reduction in the approximation error.

Experimental evaluation

- Analyzed identified protos, their transitions, and correlation between proto transitions and system side-information.

Data:

- Generated from an anonymous data collection project run by Intel and its PC OEM partners, given specific user opt-in.
- Data are noisy. Focused on CPU time for a subset of users/applications:
  - App filtering:
    - Removed unknown, system, and internet apps
    - Removed records with $<$ 60s/week utilization
    - Removed apps with $<$ 2K records
  - User filtering:
    - Kept users with $>$ 5/week utilization in $>$ 20 weeks

Data statistics

category	*count*
#users	28360
#apps	762
#weeks	100
#records	11.05M

P4 Business communication

outlook.exe	33.1
skype.exe	32.7
winword.ex	11.6
excel.exe	8.3
acrord32.e	3.7
dropbox.ex	3
wmpnetwk.	2.3
powerpnt.e	1.3
all other	4

P10 Office

outlook.exe	47.3
winword.ex	16.7
excel.exe	12.4
acrord32.e	5.5
wmpnetwk.	4.7
dropbox.ex	3.8
powerpnt.e	2.1
itunes.exe	1.4
wmplayer.e	1.2
all other	4.9

Figure: Example protos discovered by `Orion`.

Discovered protos:

- Used `Orion` to identify 15 prototypical usage patterns.
- Protos proved to be quite informative, identifying usage in four categories: productivity, gaming, communication and media, and Asian applications.
- We named each proto based on its usage.

Usage evolution

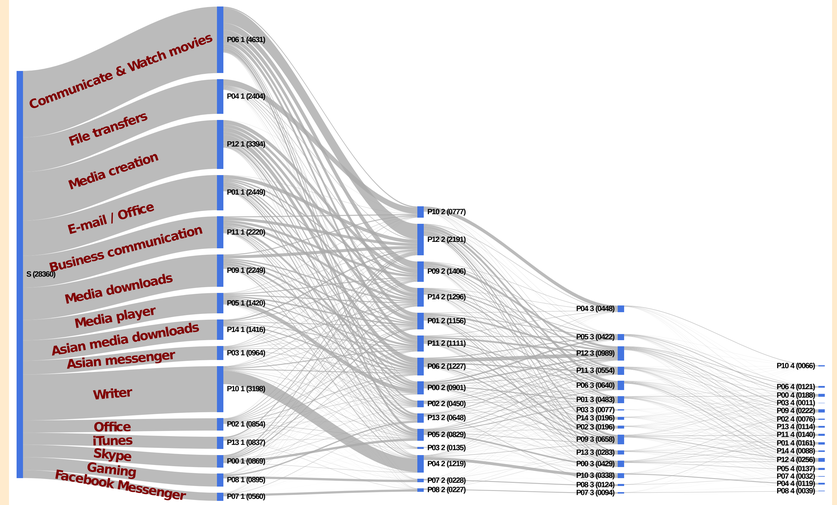

Figure: PC usage proto evolution. Each blue line represents a behavior state, either *Start* (S), or one of the protos. Gray lines show between-state transitions.

- We found that the usage patterns of nearly 50% of users change over time, and more than 20% of the users undergo multiple behavior changes.
  - Some protos are more stable than others; fewer users transition out of the state (P10, P4).
  - Others are "interior points", transition states in-between focussing on other tasks (P12).
  - Low *fan-in* (states transitioning into proto) and high *fan-out* (transitioning out of proto) indicate "early states" (P2, P3).
  - Low *fan-in* and low *fan-out* indicate niche groups, such as business or Asian users.

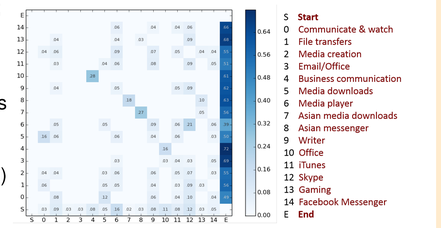

Figure: PC usage proto transitions. *S* and *E* are the *Start* and *End* states, and the numbers denote protos. Rows show the probabilities of a user transitioning from the proto identified by the row ID towards other protos, identified by the column ID.

Side information correlation

- For each proto transition, we computed the KL divergence between the side information distribution (geolocation, system type, or CPU type) of the users that belong to the "from" proto and the users that transition to the "to" proto.
- As an example, we found users transitioning into the *Office* proto state more likely to have higher-end systems, with high-end processors (i7 and i5, rather than i3, Pentium and Celeron).

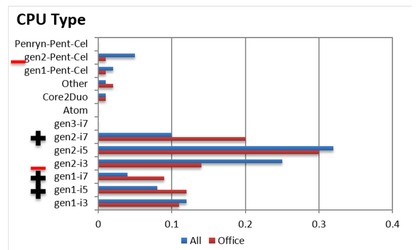

Figure: *Office* proto correlation with *CPU type* side information.

Beyond characterizing usage evolution

- `Orion` is versatile, applicable to diverse multivariate time-series domains.
- We used `Orion` to analyze purchase habit evolution of nearly 1000 users at an online grocery store, and obtained similar results.

Acknowledgements

This work was supported in part by NSF (IIS-0905220, OCI-1048018, CNS-1162405, IIS-1247632, IIP-1414153, IIS-1447788), Army Research Office (W911NF-14-1-0316), Intel Software and Services Group, and the Digital Technology Center at the University of Minnesota. Access to research and computing facilities was provided by the Digital Technology Center and the Minnesota Supercomputing Institute.