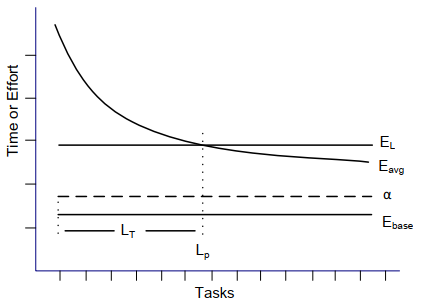
1. How would you test Learnability, Operability, and Understandability?
2. Learnability:
3. Firstly, we prepare a set of tasks which have the same scenario but different data and constraints. Users are not familiar with the application, but they are familiar with computer usage.
4. Recording the time they need to finished the task
5. Plotting the TOT or EOT average time for these users (subjects).
6. Find a curve with a strong fit to either a power law or exponential decay curve
7. When a tight fit is found, then the learning curve has occurred.
8. By using both the goodness of fit – R^2, and the learning rate – beta, it is possible to establish measurable requirements for the software
9. Then compare the learnability of different applications.
10. At the same time, we can also find learning point and, learning time
11. Operability
12. It can be get from the learning model.
13. Consider this equation we used for learning model: 
14. Several correlated parameters stem from this equation can be used to express operability
15. We can get operability from this expression: 
16. Thus, by testing learnability, we also tested the operability at the same time.
17. Understandability
18. compare the average subjects’ performance to the baseline performance
19. More specifically, 
20. Referring to the below chart, we see the gap between E-base and E – point



1. This is how we test the understandability
2. How would you perform Pinpoint analysis?

Answer:

1. We need accurate ways to identity and pinpoint issues in the software/interface.
2. This is so-called pinpoint analysis
3. There are two types:

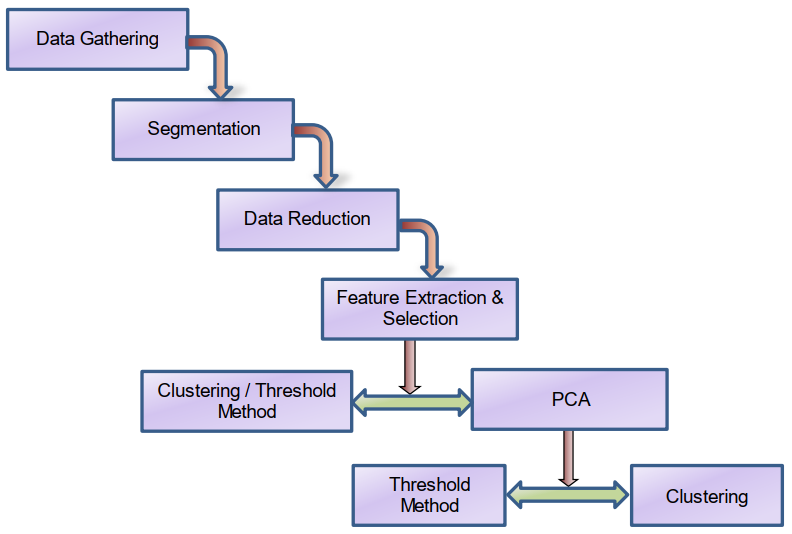
* Inter-pinpoint analysis refers to the identifying issue within tasks in a specific system. It involves detecting tasks that present anomalies and identifying the reasons for these anomalies at a high level.
* Intra-pinpoint analysis refers to identifying issues within tasks in a specific system itself. The analysis can be done manually by watching video recordings of users’ interactions with software and/or watching videos obtained from an eye-tracking device.

1. For the concrete experiments, we perform below steps:
2. Setup environment

* What kind of keyboards, mouse, and other input devices.
* Eye tracker hardware and relevant software
* Matlab for data analysis

1. Test procedures

* Data gathering, Segmentation, Data reduction, Feature extraction and Selection.



1. Adopted different method in the data analysis

* Threshold method
* K-means clustering
* PCA
* K-means on clustering

1. Result evaluation

* Classifying NE segments as E segments is regarded as false positive or **type-I error**.
* Segments that show excessive effort per manual classification but are
* Identified excessive effort segment as non-excessive effort segments is regarded as false negative or **type-II error**.

We check two aspects:

1. The number of type-II errors and
2. The minimal time to investigate the usability issues with an acceptable level of type-II errors.

The paper assumes that 15% of error of type-II is the upper bound for being considered as acceptable.

This is approximately the strategy to perform pinpoint analysis.