Inducing Shortcuts on a Mobile Phone Interface

Robert Bridle
Department of Computer Science
The Australian National University
rbridle@cs.anu.edu.au

Eric McCreath
Department of Computer Science
The Australian National University
ericm@cs.anu.edu.au

ABSTRACT

Due to size restrictions, mobile phone user interfaces are often difficult to use[8]. In this short paper, we investigated inducing shortcuts to replace the sequence of actions required to complete common tasks on a mobile phone. In particular, we used mobile phone interaction data to evaluate several methods for inducing shortcuts. We considered the balance between maximising interface efficiency and shortcuts that remained stable and hence predictable.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition, parameter learning*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*evaluation/methodology*

General Terms

Experimentation, Human Factors

Keywords

Adaptive User-Interface, Mobile Phone Interfaces, User Oriented Machine Learning

1. INTRODUCTION

Mobile phones have become commonplace and have now reached a stage where they can provide users with a range of applications and services. Despite this, mobile phones are often still regarded as being difficult to use[8]. Due to the size restrictions of mobile phones, providing a more usable interface is hard to achieve. In this paper we considered inducing shortcuts for common tasks. We assumed that mobile phone users would often repeat certain tasks, and we investigated generating and providing users with shortcuts for these tasks. For instance, consider the situation in which a user sends the same text-message to a friend daily, telling them what time to meet for lunch. This task could be automated by a shortcut that would ask the user for a meeting time and then complete the intervening steps to send the text-message. Other applications on a mobile phone could also benefit from shortcuts, including: calendar/diary appointment scheduling, web browsing and music/video players. Although shortcuts have a large potential benefit, in practise, achieving this is difficult. If too many shortcuts, or inappropriate shortcuts are presented to a user, then the effort involved to use such shortcuts quickly outweights their benefits. Similar work has focused on command-line prediction. Davison and Hirsh[2] demonstrated the use of a probabilistic bigram model to predict a user's next command in a UNIX environment. Korvemaker and Greiner[4] extended this work to the prediction of complete user commands. They showed that a relatively simple algorithm, one that considers only the previous command, could correctly predict user commands almost half the time. In this paper we investigated the challenges posed by inducing and applying shortcuts on a mobile phone. In the next Section we evaluate and compare several approaches for recognising shortcuts and consider approaches for selecting a method. Section 3 concludes this paper.

IUI'06, January 29–February 1, 2006, Sydney, Australia.
ACM 1-59593-287-9/06/0001.

2. INDUCING SHORTCUTS

We investigated inducing shortcuts on Nokia™Series 60 mobile phone. The speed-dial is a common form of shortcut, it allows a contact's number to be mapped on to the one of the 9 numeric keys for faster dialling. Since mobile phones are commonly used to make voice-calls and to send text-messages we considered inducing more complex shortcuts that would automate these tasks. To perform these tasks a user must perform a number of operations with a mobile phone's interface. Using a Nokia™Series 60 interface, two basic steps can be identified. The first step consists of looking up a contact's number and the second step consists of selecting the type of outgoing communication.

We used data collected and made publicly available[1] by Raento, et. al. as part of the ContextPhone[7] project. The ContextPhone consisted of a prototype platform that allowed contextual information to be gathered and used in mobile applications. In particular, we used the data gathered by a context logging application for Nokia™Series 60 mobile phones. The information collected by this application included: location based on cell-tower id, the active profile on the user's phone and the time/contact name/contact number/duration of all incoming and outgoing communications made on the phone. We used data from 13 anonymous users. The time period that the data was collected over, varied for each user. We evaluated several methods for producing shortcuts:

No Shortcut – This method does not produce any shortcuts for the user, and is used to bench-mark the other methods.

Last Performed – This method presents the user with the last outgoing communication performed. For exam-

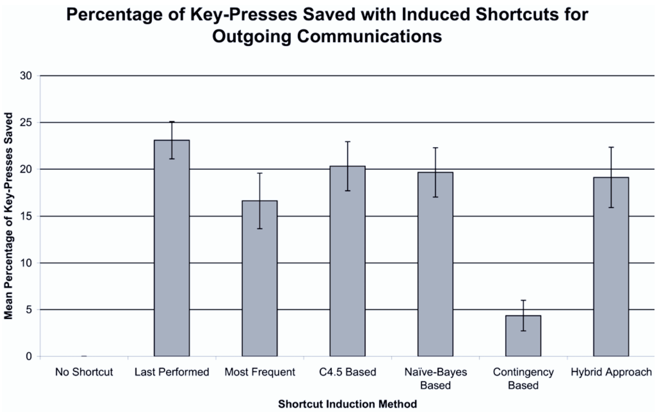

Figure 1: Mean percentage of key-presses saved if shortcuts are used when making outgoing communications.

ple, if the last outgoing communication was text-messaging one's wife, then the shortcut presented would set-up a text-message with the wife as the recipient.

Most Frequent – This method presents the most frequently occurring outgoing communication observed so far as a shortcut.

C4.5 Based – This method uses all previously observed outgoing communications as training examples to build a C4.5 decision tree[6]. For each outgoing communication the following attributes are recorded: the type of communication made and the name of the contact it was made to (these two attributes become an action-pair and this is the class attribute). The predictive attributes are the current day, time and location (based on cell id). And the day, time, communication type, contact name, contact number of the last outgoing communication made. The decision tree is employed to predict a contact name and communication type (action-pair) to present as a shortcut.

Naive-Bayes Based – This method uses previously observed outgoing communications as training examples, using the same attributes as the C4.5 method. However, a Naive-Bayes probabilistic concept is constructed[3]. The probability of each possible combination of contact name and communication type can be determined. The contact name and communication type (action-pair) of highest probability is presented as a shortcut.

Contingency Based – This method determines whether a contact name and communication type are dependent on one another. The contact name and communication type (action-pair) with the highest statistical significance of dependency (using Fisher's exact test) will be the shortcut presented to the user. For instance, if the most frequently occurring outgoing communications are voice calls to the contact name "john", and the second most frequent are text-messages to "john". Then because a large number of both voice-calls and text-messages follow-on from "john" no dependency will be detected.

Hybrid Approach – This method combines the *Most Frequent* method and the *Naive-Bayes Based* method. It operates in a similar fashion to the *Naive-Bayes Based* method. However, if the probability of the contact name and communication type (action-pair) predicted is less than 90%, the *Most Frequent* method is used.

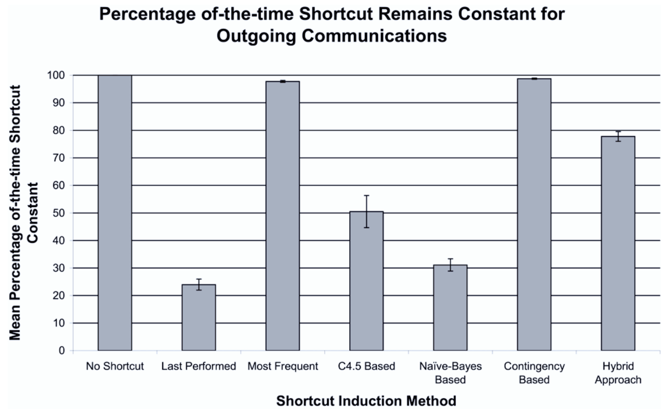

Figure 2: Mean percentage of-the-time the induced shortcut remained constant from one iteration to the next.

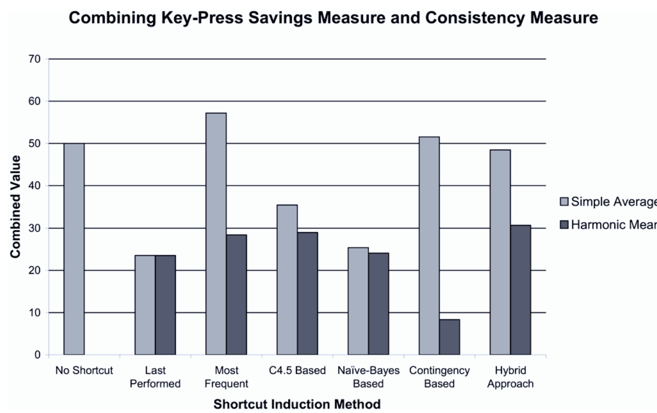

Figure 3: Average and Harmonic Mean approaches for combining key-press savings measure and the consistency measure.

2.1 Results

To evaluate each method of shortcut induction, we simulated how each method would behave on an actual mobile phone. Beginning with no examples of outgoing communications, the sequence of outgoing communication is replayed. At each point-in-time the method must guess a shortcut, if it guesses correctly than a saving in key presses is made (we assume the shortcut is used if possible). Clearly, the method may only use examples before the current point in the replayed sequence. Also, we assumed that a voice-call shortcut represented a saving of 7 key-presses and a text-message shortcut a saving of 10 key-presses. Figure 1 shows the percentage of key-presses each shortcut induction method could save a user. We also measured how constant the shortcuts were. We measured the number of times the shortcut presented to the user changed. Figure 2 shows the percentage of-the-time the induced shortcut remained constant with each method. We assumed that a method which produced stable shortcuts was desirable, since shortcuts that remain stable for long periods provide a more predictable interface.

In Figure 1, the *Last Performed* method stands out as having the potential to save the user the most key-press, approximately 24% of the total number of key-presses made. The two methods that employ a learning system, the *C4.5 Based* and the *Naive-Bayes Based* approaches, follow closely behind. The *Last Performed* method shows that the last communication made by the user has a strong bearing on the contact name and communication type the user will make next. It is likely that the learning approaches did not perform as well or better, because of too many irrelevant attributes in the training examples.

When comparing Figure 1 with Figure 2, we see that although the *Last Performed* method can save the user many key-presses, it does so by producing shortcuts that change often. This characteristic is also seen with the two learning approaches (*C4.5 Based* and *Naive-Bayes Based*). The *Most Frequent* and the *Contingency Based* methods produce shortcuts that are drastically more constant and stable. However, the *Contingency Based* method requires many examples for a statistically significant shortcut to be found, it therefore does not perform well in saving many key-presses in the short-term. The *Most Frequent* method is potentially the best candidate method for inducing shortcuts, as it provides both high savings in key-presses and low variability amongst shortcuts. However, this approach could be improved upon. The *Hybrid Approach* takes advantage of the stable yet successful strategy of the *Most Frequent* method, but also relies upon concepts generated by a Naive-Bayes learner. Whenever the Naive-Bayes learner can classify the next training example with a high probability, 90% or greater, then it predicts which shortcut to show, otherwise the *Most Frequent* method is used. In effect, the classifier is only invoked if it is confident in classifying which shortcut to present.

2.2 Stable vs. Changing Shortcuts

Efficiency and predictability are the two key aspects that are considered when mobile phone interfaces are designed[5]. These two aspects should be reflected in the approach we use to select shortcuts. Both of these characteristics can be simply measured. The first can be measure as the percentage of key presses saved(denoted p). The second can be measured as the percentage of times a shortcut remains the same over the entire interaction(denoted v). These two measures must be combined into a single measure to provide a way of comparing different approaches. We now state and discuss three possible ways of combining the measures:

Average – A simple average can be taken of the two measures : $(p + v)/2$. This assigns equal weight to each component. Note that the average evaluates to 50% for the bench-mark case where *No Shortcut* is used. The combined average for the results presented in this paper are shown in Figure 3. A variation of this measure would be to have a weighted average. This would enable the system designer to reflect the importance of saving key presses compared to reducing the number of times the shortcut changes.

Harmonic Mean – Another approach would be to use a harmonic mean for combining these measures : $2pv/(p+v)$. This means that you don't perform well in one measure at the expense of the other. The harmonic mean is used for combining precision and recall into the F1 measure, and is used by information retrieval researchers. The harmonic mean for the different methods is shown in Figure 3.

Threshold – Users may tolerate a limited amount of changing of the shortcut used. This motivates comparing methods based on key-press savings only when v is great than some threshold. If a threshold of 95% was applied to our empirical data then only the *Most Frequent*, *No Shortcut*, and *Contingency* approaches would be considered. And of these the *Most Frequent* would perform the best.

Interestingly, the average and harmonic mean approaches for combining the measures, rate the different methods for inducing shortcuts very differently. These differences motivate the need for an investigation into how these measures may be most appropriately combined. However, a number of other issues need to be addressed if such an approach is to be incorporated into a mobile phone, these include: how to present these changing shortcuts to the user, and how the user can scrutinise and control this feature.

3. CONCLUSIONS

We investigated inducing shortcuts to automate common tasks on a mobile phone. We compared several approaches and found that some relatively simple approaches saved a considerable number of key-presses. However, such approaches presented the user with shortcuts that varied frequently. A hybrid approach was introduced that combined frequency and Naive-Bayes approaches. The hybrid approach shows some potential in improving the mobile phone's user interface. This study also raises an important question of how the trade-off between efficiency and predictability should be evaluated. Given that this is a short paper we have focused on the issue of stable vs. changing shortcuts.

4. REFERENCES

[1] Context Project Dataset. Retrieved July 18, 2005, from Department of Computer Science and HIIT Basic Research Unit, University of Helsinki Web site: http://www.cs.helsinki.fi/group/context.

[2] B. Davison and H. Hirsh. Probabilistic online action prediction. In *Intelligent Environments: Papers from the AAAI 1998 Spring Symposium, Technical Report SS-98-02.*, pages 148–154, 1998.

[3] R. Duda and P. Hart. *Pattern classification and scene analysis.* John Wiley and Sons., New York, NY, USA, 1973.

[4] B. Korvemaker and R. Greiner. Predicting unix command lines: Adjusting to user patterns. In *Proceedings of the Seventeenth National Conference on AI and Twelfth Conference on Innovative Applications of AI*, pages 230–235, 2000.

[5] C. Lindholm and T. Keinonen. *Mobile Usability: How Nokia Changed the Face of the Mobile Phone.* McGraw-Hill Professional, New York, NY, USA, 2003.

[6] J. R. Quinlan. *C4.5: programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[7] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing*, 4(2):51–59, 2005.

[8] R. St. Amant, T. Horton, and F. Ritter. Model-based evaluation of cell phone menu interaction. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 343–350, 2004.