

Rhys Smith

Department of Computer Science,
Cardiff University, Cardiff, UK.
R.O.Smith@cs.cardiff.ac.uk

Abstract

The idea of using user preferences to personalise e-services has caught the attention of the research community recently. User preferences can be used in several applications in this context, including helping with information filtering and providing the “best” answers to user queries. To this end, several frameworks for formulating user preferences and their embedding into relational query languages have been proposed. However, there are major issues as to the trustworthiness of the results returned and general privacy concerns inherent in the paradigm shared by these frameworks.

To solve these problems we need to thoroughly investigate how user preferences are represented, creating a well-defined language for preference representation. We need to propose a new paradigm for future frameworks to take, and investigate the applications that this framework can feasibly be applied to.

Introduction

User preferences can be used to help with information filtering and providing “relevant” answers to user searches.

Several preference-based query frameworks have been proposed, all of which share the same basic paradigm:

- A user expresses their preferences in a preference statement. This defines exactly what particular values (or type of values) the user would prefer particular attributes to take (or not take).
- These preferences are then transmitted to an e-business’ system (usually a preference-enabled DBMS).
- Their system then searches through the data it holds, returning items that best matches the user’s preferences, ranked according to these preferences.

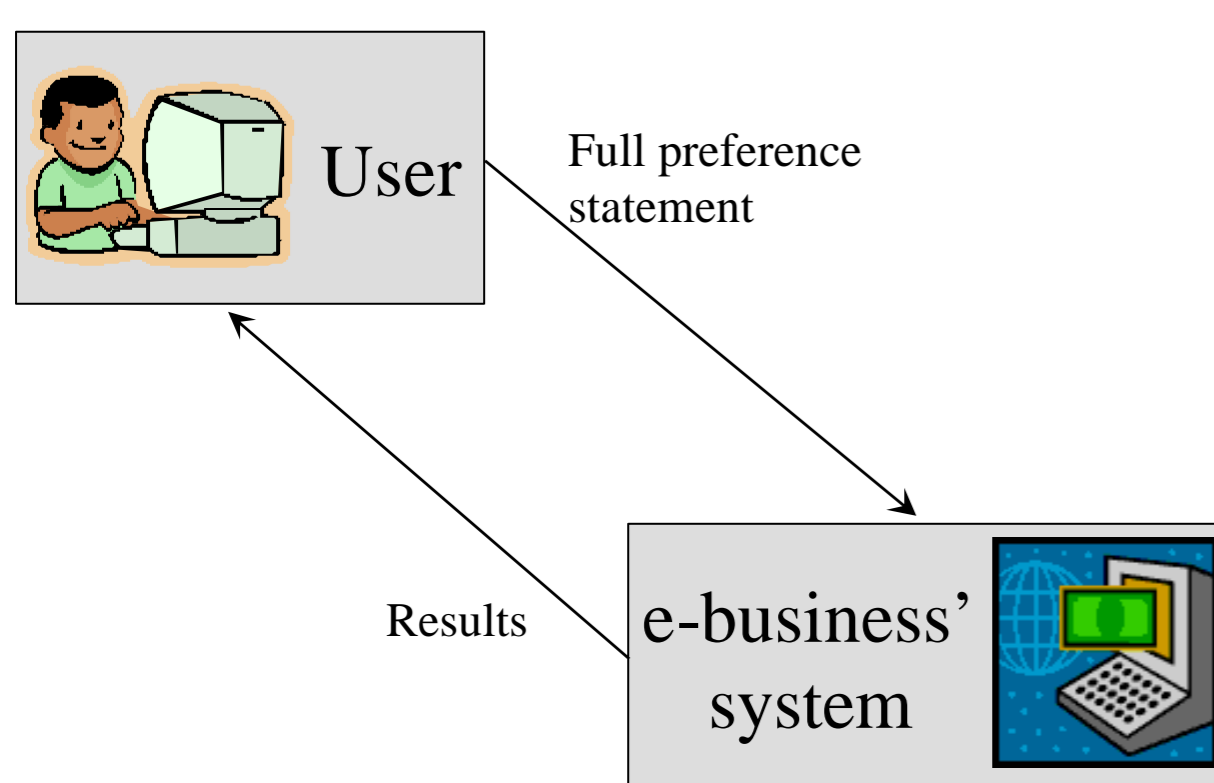


Figure 1: The current paradigm

However, when this paradigm is applied to e-business and systems that are not fully trusted by the user, there exists two main issues – the trustworthiness of results, and general privacy concerns. These are explained below.

Trustworthiness issues

When users’ preferences are sent in full to an e-business’ system, the knowledge contained in this full preference statement can be exploited by an untrustworthy system. This could result in the user being returned a set of results that are not optimal for them, instead being optimal for the business.

To illustrate this, consider the following:

Example 1: Speedee SuperBoats is a company that sells boats. Alice wants to buy a speedboat that is made by either Oceanic or Marine (preferably Oceanic), costing between £25,000 and £40,000 (the lower the better).

The company have in stock one Oceanic boat costing £30,000, and three Marine boats costing between

£34,000 and £38,000. If the company were trustworthy, then the search should return the Oceanic boat. However, the company is not trustworthy, and have somewhat unscrupulously set up their system so their own preferences are added to, and supersede, the user’s preferences.

By exploiting the knowledge contained in the customer’s preferences available to them, Speedee SuperBoats could return a Marine boat, knowing that this will still satisfy Alice’s preferences, but at the same time maximising the company’s profit and keeping the last Oceanic boat in stock.

Privacy issues

There is a major conflict between users and e-business in the area of privacy – e-businesses are thirsty for ever increasing amounts of information about their users, which undermines the users’ fundamental right to privacy.

When users’ preferences are sent in full to an e-business’ system, the knowledge contained in this full preference statement about a user’s likes and dislikes could be stored by an e-business. Although this could help them to adapt their service to each customer’s needs, it is breaching the customer’s right to privacy. Recent studies have shown that although a large percentage of internet users are concerned about privacy, many of these privacy-concerned users disclose a lot of information about themselves. Basically, users cannot be trusted to keep sensitive information about themselves private!

If there were a method available of retrieving optimal results for a user (according to their preferences) while only releasing part of the full preference knowledge, there is no reason to release the full knowledge. In fact, doing so impacts on users privacy, for no adequate reason.

Gradual Preference Release

If a user wants to obtain trustworthy results from systems they do not fully trust, whilst retaining as much privacy as possible, the current preference based paradigm is not adequate.

Our Approach

- An agent acts on behalf of the user.
- The user expresses all of their preferences to their agent, which analyses these preferences and splits them into “parts”.
- It gradually releases these parts to the e-business’ system one at a time until satisfactory results were returned to the agent
- The agent then sends these results back to the user.

This minimises the preference knowledge transmitted, thus preventing exploitation of the knowledge contained in the full preference statement and maximising the user’s privacy.

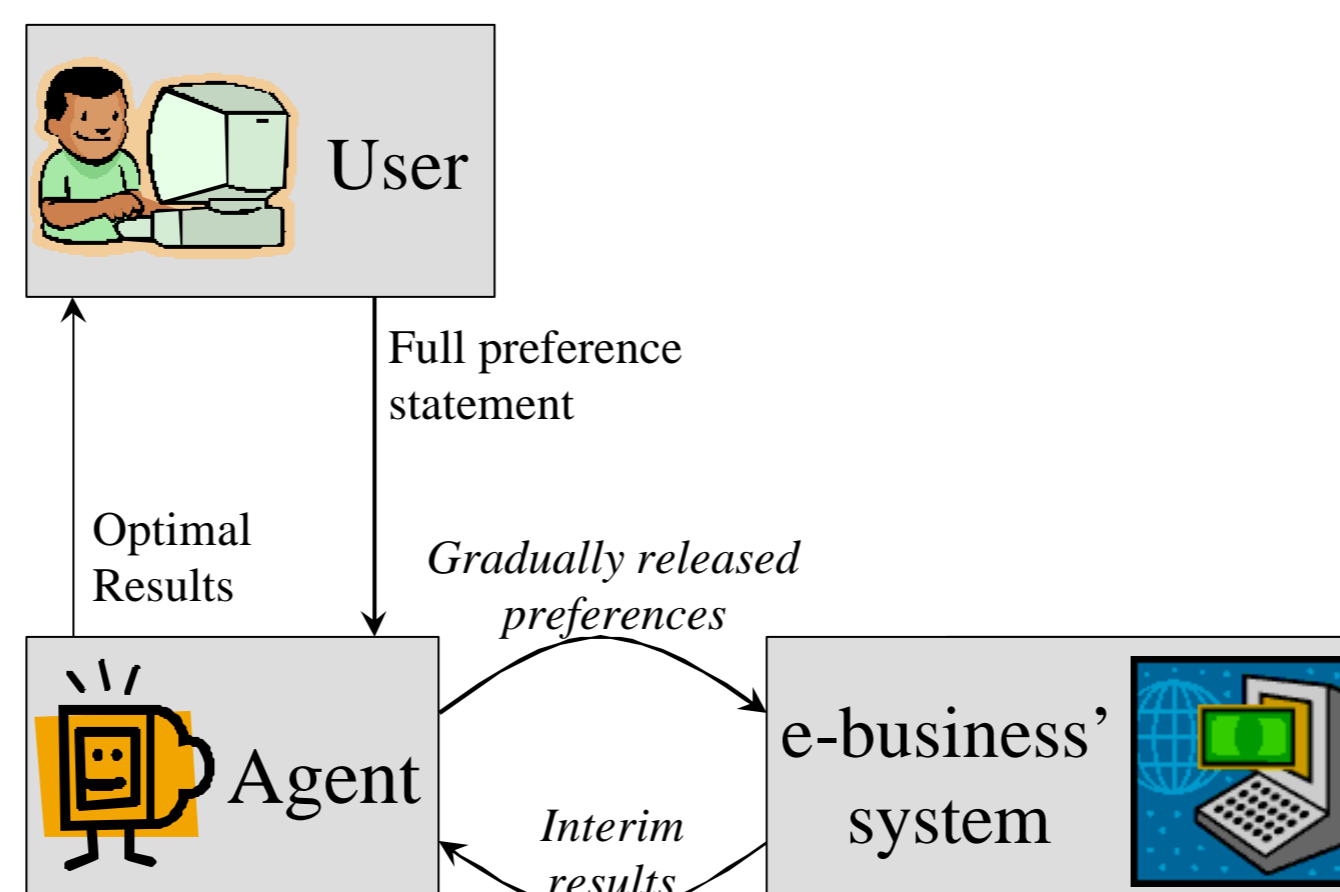


Figure 2: Our proposed framework

A Selection of the Main Challenges

- How do we take the user’s preferences and split them up so that they can be released gradually?
- Is our approach actually feasible, performance-wise? Intuitively, our approach should send more queries to the e-business’ system than the standard approach, however, the result sets returned should be a lot smaller. Does this balance out to produce a framework that can actually be used?
- How can our framework be optimised? Can the way the “parts” of preferences are sent change the performance of the approach? As each interim query sent to the e-business’ system will be only slightly different to the last, is there some way the previous query or results could be cached somewhere, for example? How else can we optimise our framework?
- What applications can our framework be successfully applied to?

Stating Preferences

In order for our approach to work, there must exist a formal method of stating a user’s preferences. This method must be able to represent exactly a user’s preferred (and non-preferred) values across multiple attributes, and the internal ordering of these preferences.

However, current work in the area of preference based searching has approached the problem from the data end, by applying preferences gathered by “some means” and analysing the resulting sets of tuples. No major work has been done to date on the representation of the actual preferences themselves, and their properties.

As there is currently no formal method of stating a user’s preferences available, we need to define such a method – how to represent single preferences, and how to combine these single preferences into a full preference statement. We need to define the operators involved and their properties, and analyse the properties of the resulting language as a whole. Investigation should take place into whether this language can be very simple, or does it need to include more complicated ideas such as allowing a user to state things like “If x then I prefer y, otherwise I prefer z”?

In order for the gradual releasing algorithm to work, there needs to be a fully defined ordering between all of the preferences, so the language we define needs to produce a fully ordered preference graph.

Applications

Once a preference language and a gradual release framework has been defined, and both have been thoroughly analysed, we can then think about the applications of our privacy preserving trustworthy preferences – there are many possible applications, but we need to investigate which will feasibly be able to use such an approach.

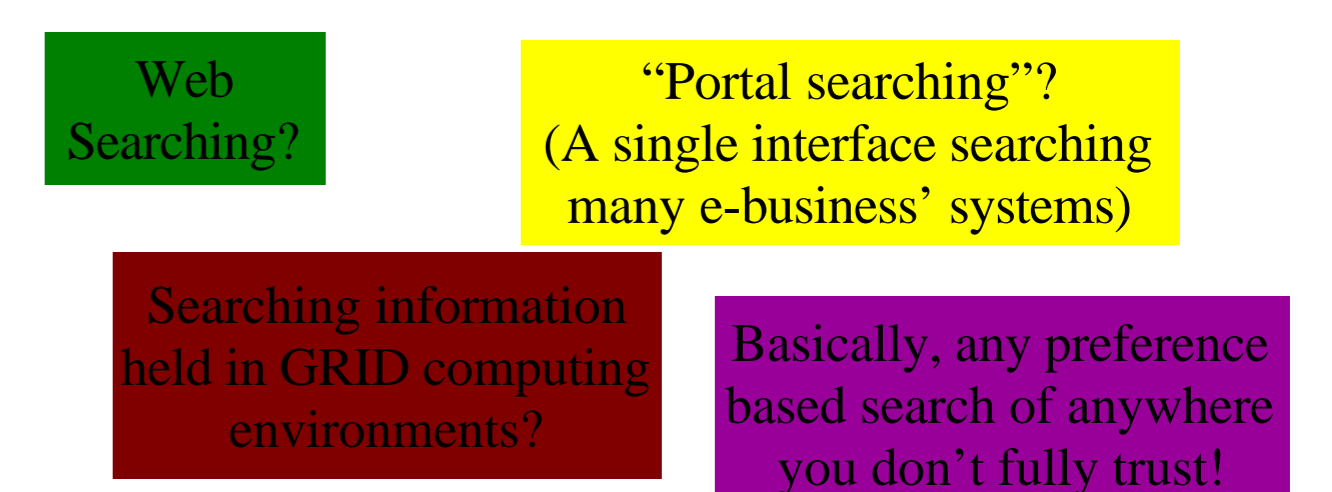


Figure 3: Some Possible Applications

We would need to try to adapt and apply our framework to each of the areas we identify, and evaluate their effectiveness.