Evaluation Datenbank
Projekt BierIdee

Danilo Bargen, Christian Fässler, Jonas Furrer

3. April 2012

1 Evaluation

1.1 Vorwort

Da es sich beim Projekt BierIdee - im Rahmen des Modules Software Engineering 2 Projekte - um ein Projekt mit stark beschränktem Zeitbudget handelt und zudem der Fokus auf der Anwendung des *Rational Unified Process* liegt, wird diese Evaluation bewusst einfach gehalten. Die Produkt-Auswahl enthält auch subjektive Kriterien, da die Zeit für eine ausgwogene objektive Beurteilung fehlt.

1.2 Einführung

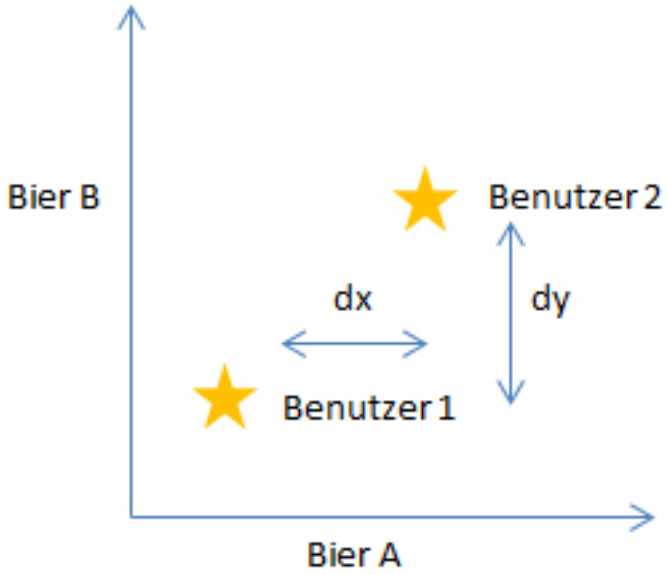
Es stellt sich die Frage wie sich unsere Problemdomain am besten modellieren lässt. Zwei Ansätze die sich herauskristalisiert haben sind Vektorräume und Graphen.

1.3 Verfahren

. Zur Auswahl Stehen lediglich die Zwei Verfahren Vektorräume und Graphen. Diese Auswahl entstand einerseits durch eine Besprechung mit einem Mathematik-Dozenten und durch Eigenständige Modellierung.

1.3.1 Vektorräume mit relationaler Datenbank

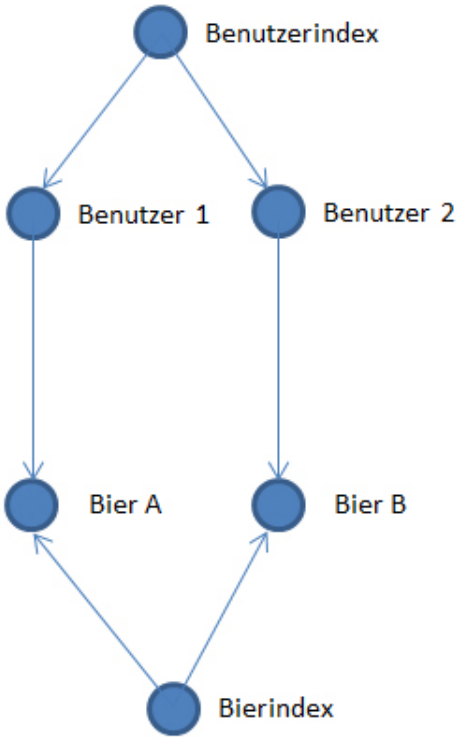
Bei der Modellierung mittels Vektorräumen werden die Bewertungen der Biere mittels Vektoren dargestellt. Jede Vektorkomponente stellt dabei ein Bier dar.



Möchte man

nun für einen Benutzer A eine Empfehlung erstellen, vergleicht man alle Bewertungen des gleichen Bieres der anderen Benutzer (selbe Vektorkomponente). Danach ërforscht"man die anderen Bierbewertungen der anderen Benutzer (restliche Vektorkomponenten). Um eine hohe Empfehlungsgenauigkeit zu erhalten ist eine sehr aufwendige aggregierung sämtlicher Konsumationen notwendig.

1.3.2 Graphen mittels Graphen-Datenbank



Die Benutzer und Bier werden mittels Knoten in einem Graph modelliert. Die Konsumationen und Bewertungen werden mittels Kanten (Beziehungen) zwischen den Benutzern und Bieren dargestellt. Möchte man nun dem einen Benutzer eine Empfehlung aufgrund eines vorgenommen Konsumes machen, lässt sich das sehr erstellen, indem man einfach auf andere Nutzer "traversiert", die das selbe Bier ebenfalls konsumiert haben. Z.B Benutzer 1 – Konsumiert - ¿ Bier A ¡ - Konsumiert – Benutzer 2.

1.4 Beispiel

Es soll eine Empfehlung für einen Benutzer 1 erstellt werden. Die Empfehlung soll auf den bereits Konsumierten Bieren (Bier A) bestehen. Es soll ein konsumiertes

Bier B gefunden werden welches vom Benutzer 2 konsumiert wurde. Benutzer 2 hat dasselbe Bier A auch konsumiert. Daher ist davon auszugehen das die beiden Benutzer einen ähnlichen Geschmack haben. Die Datenaggregation erfolgt also in den folgenden 2 Schritten:

- Schritt 1: Finden eines Benutzers 2 welcher auch ein Bier A konsumiert hat.
- Schritt 2: Finden eines Bieres welches von Benutzer 2 konsumiert wurde.

1.4.1 Vektoren

Konsumationen werden in einer Tabelle festgehalten. Ein Datensatz besteht aus einem Benutzer und einem Konsumierten Bier.

- Schritt 1: Für das finden eines Benutzers welcher auch ein Bier A konsumiert hat müssen im schlimmsten Falle alle Konsumationsdatensätze abgefragt werden.
- Schritt 2: Das finden eines anderen Bieres von Benutzer 2 als Bier A benötigt im schlimmsten Falle noch eine Laufzeit von O(AnzahlKonsumationen von Benutzer 2).

1.4.2 Graphen

- Schritt 1: Das finden eines Benutzers welcher auch ein Bier A konsumiert hat, kann der graph via Beziehung rückwärts traversiert werden zu einem Benutzer 2.
- Schritt 2: Das finden eines Bier B erfolgt dann wieder durch das traversieren zu einem konsumierten Bier B.

1.5 Auswahlverfahren

Die Auswahl zwischen den beiden Produkten passiert anhand von Kriterien die einfach und schnell zu evaluieren sind. Dazu gehören:

Laufzeit / Komplexität Wie Zeitintensiv ist die Generierung/Berechnung der Daten. Konkret, wie schnell können Empfehlungen gefunden werden.

Produkte/Frameworks Grober Überblick über vorhandene Frameworks, API's, DB Extensions welche bereits eine der angedachten Modellierungsvarianten ermöglichen.

Subjektivkriterien Wie Empfehlungen durch dritte, oder Vorlieben von Teammitgliedern. (Auf diese Kriterien wird nicht mehr weiter eingegangen)

1.6 Auswahl

Bei den gewählten Kriterien ist ein direkter Vergleich nur schwierig möglich. Deshalb werden hier lediglich die Gedanke die zur Produktwahl geführt haben festgehalten.

Komplexität Die Modellierung mittels Graphen zeigt, dass das Laufzeitverhalten für den erwähnten Anwendungsfall (Empfehlung erstellen), sehr viel besser ist. Softwaretechnisch gesehen, ist der umgang mit Graphen viel einfacher als das modellieren von Vektorräumen.

Popularität Bei der Suche nach einer Graphenbasierten Datenbank auf Google steht Neo4j sehr weit oben. Die Suche nach Referenzprojekten liefert leider wenig aussagekräftige Resultate. Twitter setzt auch auf eine Graphenbasierte Datenbank. Da unser Problem sehr verwandt mit diesem System ist, liegt die Entscheidung für eine solche Datenbank sehr nahe. Twitter arbeitet eng mit Neo4j zusammen, dieser Fakt ist als sehr entscheidend für die Wahl eines Produktes zu betrachten.

Low Representational Gap Die Graphenbasierte Lösung beschreibt sehr viel genauer die tatsächlich gegebene Beziehungsstruktur der Problemdomain und erlaubt somit ein viel intuitiverer und natürlicher Umgang mit den Daten.

1.7 Evaluationsergebnis

Das Resultat der Evaluation ist eine modellierung mittels Graphen. Als Produkt wird Neo4j verwendet.