

Dokumentation zum Programmentwurf



Buchempfehlung

WISSENSBASIERTE SYSTEME

Eike-Lars Holle & Daniel Auler

Matrikelnummern:	7265951 & 5217026
Kurs:	TINF12C
Dozent:	Prof. Dr. Dirk Reichardt
Abgabedatum:	12.05.2015

Inhaltsverzeichnis

Abbildungsverzeichnis	1
Tabellenverzeichnis	1
1 Einleitung	2
2 Planung	3
2.1 Programmiersprache und Umgebung	3
2.2 Dempster's Regel	3
3 Praktische Umsetzung	4
3.1 GUI	4
3.2 Backend	5
3.3 Ausführen des Programmentwurfs	7
4 Fazit	8

Abbildungsverzeichnis

Figure 1 – Akkumulation der Werte m_1 und m_2	3
Figure 2 - Programmaufbau.....	5

Tabellenverzeichnis

Table 1 – Vorliegende Informationen zur Buchempfehlung	2
--	---

1 Einleitung

Im Rahmen der Vorlesung wurde die Evidenztheorie sowie insbesondere die Dempster's Regel näher behandelt. Die aus dem Bereich der Wahrscheinlichkeit stammende Theorie wird angewendet, wenn mehrere verschiedene Informationen zu einer Gesamtaussage zusammen gesetzt werden. Dabei wird die Glaubwürdigkeit, mit der diese Aussagen getroffen werden, besonders berücksichtigt.

In der vorliegenden Aufgabenstellung soll eine Buchempfehlung anhand vorgegebener Informationen (s. Tabelle 1) des Lesers ausgesprochen werden.

Informationen
Altersgruppe
Geschlecht
Familienstand
Kinderzahl
Einkommen
Bildungsstand
Beruf

Table 1 – Vorliegende Informationen zur Buchempfehlung

Jeder Information soll ein Basismaß zugeteilt werden, das basierend auf einer vorgegebenen CSV-Datei erstellt wird. Im Rahmen des Programmentwurfs werden die entsprechenden Klassifikationen mit Hilfe der Evidenztheorie berechnet und schlussendlich ausgegeben.

2 Planung

2.1 Programmiersprache und Umgebung

Bei der Umsetzung der Software entscheiden wir uns aufgrund unserer guten Kenntnisse für die Programmiersprache Java und die Entwicklungsumgebung IntelliJ IDEA. Die grafische Oberfläche wird mit Hilfe der Software „Java FX Scene Builder“ erstellt.

2.2 Dempster's Regel

Mithilfe der Dempster's Regel werden die jeweiligen Evidenzen der Informationen akkumuliert und zu einem Gesamtmaß errechnet.

Dempster-Shafer Regel (ohne Konflikt)

$$m_1 \oplus m_2 (Z) = \left(\sum_{Z=X \cap Y} m_1(X) \cdot m_2(Y) \right)$$

Figure 1 – Akkumulation der Werte m_1 und m_2

Die Basismaße der für den Programmentwurf vorliegenden Informationen werden anhand der Daten aus der CSV-Datei bestimmt. Hierzu werden die jeweiligen Evidenzen der einzelnen Informationen benötigt, die für die Umsetzung des Programmentwurfes definiert werden müssen. Wir legen fest, dass die benötigten Evidenzen anhand der Anzahl berechnet werden, wie häufig die selben Daten erscheinen. Diese Anzahl teilen wir durch die Gesamtzahl an Eigenschaften, um eine aussagekräftige Wahrscheinlichkeit zu erhalten.

3 Praktische Umsetzung

In diesem Kapitel wird die Umsetzung des Projektes näher beschrieben. Angefangen mit der GUI, welche eine einfach zu bedienende Nutzerschnittstelle darstellt. Anschließend wird das Backend näher beleuchtet.

3.1 GUI

Die GUI ermöglicht dem Benutzer einfache Eingaben über sogenannte Choice Boxes. Diese enthalten ein Dropdownmenü, über das alle Eingabemöglichkeiten aus der Beispieldatei eingegeben werden können. Eine dynamisch generierte Auswahl aus der Eingabedatei ist hier nicht vorgesehen, da diese den Rahmen der Arbeit sprengen würde.

Nach Auswahl der einzelnen Attribute kann durch den Button „Buch empfehlen lassen“ die Berechnung der Evidenzen erfolgen. Dieser Button öffnet ein Popup, in dem die Empfehlungen angezeigt werden. Sollte das Ergebnis bei der Gesamtevidenzberechnung eine leere Menge sein, wird angezeigt, dass keine Empfehlung ausgesprochen werden kann. Die errechnete Gesamtevidenz wird in der GUI nicht angezeigt, da diese bei den Beispielwerten sehr gering ist, eine geringe Aussagekraft darstellt und somit die Empfehlung abwerten würde.

Die GUI ermöglicht außerdem das Nutzen einer anderen Datei als der Beispieldatei. Diese Funktion ist über das Menü unter „Datei -> Neue Daten...“ erreichbar. Des Weiteren sind im Menü Funktionen für das Schließen des Programms, Öffnen der Dokumentation im PDF-Format und ein „Über“ enthalten, in dem die Entwickler erwähnt werden.

Die GUI ist über spezielle Controller an das Backend angebunden, das im Folgenden erklärt wird.

3.2 Backend

Wie bereits erwähnt sind die Controller der GUI die Schnittstelle zum Backend. Der GUI Controller erfasst die Eingaben des Nutzers und bringt diese in das Format zur weiteren Berechnung. Anschließend stößt er die Berechnung an und öffnet das Fenster zu Anzeige des Ergebnisses. Zudem enthält er Methoden, die Funktionen des Menüs triggern. Ein weiterer Controller ist der PopupController, welcher das Ergebnisfenster steuert. Diese beiden Klassen kommunizieren das Ergebnis über die Hilfsklasse „ModelContext“. Diese Klasse entspricht auch dem Model des MVC-Patterns, welches innerhalb dieses Projektes angewendet wird.

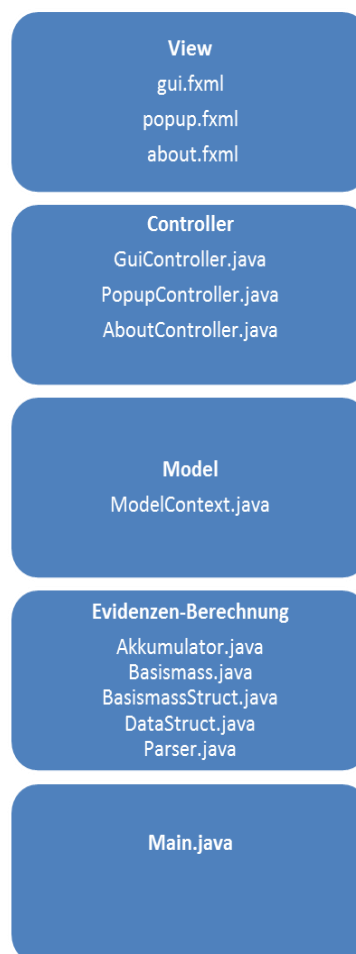


Figure 2 - Programmaufbau

Beim Start der Anwendung analysiert ein Parser die gesamte Beispieldatei und erzeugt eine Arraylist, welche einfacher zugänglich ist als die Datei. Dies geschieht auch, wenn eine neue Datei eingelesen wird. Aus dieser Datenstruktur werden weitere Daten zur definition des Basismaßes gewonnen. Diese Daten werden innerhalb der Klasse „DataStruct“ ermittelt und anschließend zum Errechnen der Evidenzen in die Klasse „Basismass“ gegeben. Nach der in Kapitel 2 beschriebenen Methodik werden dort die einzelnen Evidenzen berechnet.

Für jede Kategorie, beispielsweise das Alter, gibt es eine Instanz vom Basismaß, welche eine Arraylist für die einzelnen Basismaßmengen und dem dazugehörigen Wert speichert. Die Mengen und Werte werden zusätzlich mit dem gegebenen Merkmal in einer Hilfsklasse namens „BasismassStruct“ gespeichert.

Die Akkumulation der Basismaße findet innerhalb der Klasse „Akkumulator“ statt. Diese stellt Methoden zur Verrechnung von zwei oder mehreren Basismaßen zur Verfügung. Die Verrechnung findet über das Standardverfahren statt. Zuerst wird die Schnittmenge der jeweiligen Maße gebildet und anschließend die fokalen Mengen multipliziert. Bei gleichen Ergebnissen der Schnittmenge werden die Evidenzen jedoch nicht addiert und das ganze zusammengefasst, da sich die Empfehlung auf spezifische Merkmale bezieht. Bei leeren Mengen wird jedoch zusätzlich die Korrektur nach dem Standardverfahren berechnet. Das Ergebnis wird im GUIController gespeichert und kann dann ausgegeben werden.

3.3 Ausführen des Programmentwurfs

Um die entwickelte Software einfacher ausführen zu können, wurde eine „Runnable Jar-File“ erzeugt. Die ausführbare Datei „buchempfehlung.jar“ befindet sich im Unterverzeichnis „\out\artifacts\buchempfehlung.jar“. Hierbei ist zu beachten, dass sich die benötigten Ressourcen wie Dokumentation und die initiale Quelldatei im selben Unterverzeichnis befinden. Wird später über das Programm eine andere CSV-Datei geladen, kann diese jedoch an beliebiger Stelle gespeichert sein, da hierbei der vollständige Pfad eingelesen wird.

4 Fazit

Der Programmentwurf konnte erfolgreich umgesetzt werden. Eine grafische Oberfläche wertet die Logik erheblich auf und ermöglicht dem Benutzer, Eingaben auf eine einfache Weise zu tätigen. Dem Anwender wird nun individuell eine Empfehlung anhand vorgegebener Daten ausgesprochen und auf übersichtliche Weise dargestellt. Die Berechnungen konnten mithilfe des Dempster/Schafer-Algorithmus effektiv umgesetzt werden.

Optimierungspotenzial bietet die Festlegung der Evidenzen sowie der akkumulierten Werte, da diese relativ klein sind. Des Weiteren könnte der Algorithmus zum Auslesen der Daten aus dem Array verbessert werden, da die Daten bei jedem Abruf erneut aus dem Array ausgelesen werden. Dieser Vorgang ist weniger effizient, aber im Rahmen des Programmentwurfs aufgrund weniger Daten geeignet.