

Intelligente Systeme

Praktikum

Aufgabe 3

Dokumentation

Julian Parr, Thomas Jürgensen

Inhaltsverzeichnis

Aufgabenstellung.....	3
Die Idee.....	3
Ansatz des Mehrheitsentscheids.....	3
Ansatz des Quorums.....	4
Die Umsetzung.....	4
Rahmenprogramm.....	4
Mehrheitsentscheid.....	5
Quorum.....	6
Gegenüberstellung und Auswertung der Ideen.....	8

Aufgabenstellung

Die Aufgabe bestand zunächst darin, einen Agenten zu entwickeln, der auf eine Situation reagieren kann. Hierzu gab es eine harmlose Situation und eine gefährliche Situation, die mit den Werten 4 und 7 repräsentiert wurden. Der Agent sollte eine verschwommene Wahrnehmung besitzen, was dadurch realisiert werden sollte, dass die Situation an den Agenten gegeben werden sollte und mithilfe einer Gauß'schen Normalverteilung und einer zufällig gewählten Standardabweichung im Bereich $[3,5)$ ein zufälliger Fließkommawert die wahrgenommene Situation repräsentieren sollte. Die Reaktion sollte durch einen Schwellwert im Bereich $[5.5, 8.5)$ bestimmt werden.

Letztlich sollten mehrere Agenten nach diesem Prinzip erstellt werden; der eigentliche Kern der Aufgabe bestand darin, dass diese sich untereinander beobachten können sollten um somit die Wahrnehmung der Gruppe zu verbessern. Hierbei sollen auch Testdurchläufe erlaubt sein, in denen die Agenten Erkenntnisse übereinander gewinnen können.

Die Idee

Zunächst wurde sich in der Gruppe darauf geeinigt, dass jedes Mitglied das Rahmenprogramm selbst implementieren würde, um eine gleiche Wissensbasis zu schaffen, da die Bedingungen aus der Aufgabe hinreichend ersichtlich waren. Es stellte sich heraus, dass die Ansätze deckungsgleich waren und somit mit der subjektiv „schöneren“ Version weiter gearbeitet werden konnte. Eine Erläuterung des Rahmenprogramms folgt im Abschnitt „Die Umsetzung“; zunächst werden die Ansätze der Kernaufgabe beschrieben.

Ansatz des Mehrheitsentscheids

Der aus unserer Sicht pragmatischste Ansatz ist der des Mehrheitsentscheids: Jeder Agent bildet sich eine Meinung über die aktuelle Situation und zählt, wie viele weitere Agenten diese Einschätzung teilen. Hierbei sollen besonders „gute“ Agenten, die die Situation öfter richtig einschätzen, einen größeren Einfluss auf das Ergebnis haben als „schlechte“ Agenten, die besonders oft falsch lagen. Das Verhalten der Gruppe ergibt sich aus der Einschätzung, welche die Mehrheit der Agenten teilen.

Ansatz des Quorums

Ein anderer Ansatz ist der des Quorums.

Das Quorum ist ein Begriff aus dem Kontext der Politik, wenn es um Entscheidungsfindungen geht: Es wird ein Schwellwert, wie zum Beispiel das Erreichen eines bestimmten Prozentsatzes, festgelegt; sobald dieser Schwellwert erreicht wird, ist die Abstimmung erfolgreich. Dieses Prinzip kann bei den Agenten ebenso umgesetzt werden: Ein Agent schaut sich einen kleinen Teil der anderen Agenten an und schaut, ob sie zusammen ein vorher festgelegtes Quorum erfüllen. Falls dies nicht der Fall ist, sollen weitere Agenten betrachtet werden. Das Quorum soll aus Testdurchläufen ermittelt werden.

Die Umsetzung

Dieser Abschnitt befasst sich überwiegend mit der Umsetzung der Kernaufgabe. Das Rahmenprogramm wird nur grob Erläutert, da es sehr gut aus der Aufgabenstellung hervorgeht.

Rahmenprogramm

Das Rahmenprogramm besteht aus der Main-Klasse, einer Agenten-Klasse und einer AgentMemory-Klasse. Letztere wird lediglich für den Mehrheitsentscheid verwendet, enthält True- und False-Positive-Werte, die als Raten in Gettern zurückgegeben werden.

Zur Beschreibung des Agenten sollen folgende Tabellen dienen:

Attribut	Beschreibung
int agentID	ID zur wiedererkennung des Agenten. Wird lediglich für den Mehrheitsentscheid verwendet
AgentMemory[] agentsRating	Speichert die True- und False-Positive-Raten jedes anderen Agenten aus den Trainingsdurchläufen. Wird lediglich für den Mehrheitsentscheid verwendet
double standardDeviation	Eine zufällige Standardabweichung im Bereich [3, 5.5). Wird für die „verschwommene“ Wahrnehmung verwendet
double threshold	Ein zufälliger Schwellwert im Bereich [5.5, 8.5), der die Entscheidungsfindung beeinflusst
double minAmountOfAgentIds	Prozentsatz der bestimmt, wie viele andere

	Agenten betrachtet werden. Wird für das Quorum verwendet
--	--

Methode	Beschreibung
Agent(int agentID, int agentCount)	Konstruktor des Agenten agentID : die ID des Agenten agentCount : die Anzahl an Agenten, die insgesamt erstellt werden sollen
void newSituation(double Situation)	Erstellt eine neue Situation mithilfe einer Gauß'schen Funktion und der Standardabweichung des Agenten situation : 4: Harmlose Situation 7: gefährliche Situation
void observeWorld(Agent[] agents, boolean isDangerous)	Füllt das agentsRating-Array mit dessen True- und False-Positive-Werten während der Trainingsdurchläufe agents : das Array der Agenten isDangerous : gibt an, ob die Situation wirklich gefährlich ist
void isDangerousBasedOnWorld(Agent[] agents, boolean isDangerous)	Führt Evaluation der realen Situation durch. Hier befindet sich der Kern der Aufgabe agents : das Array der Agenten isDangerous : gibt an, ob die Situation wirklich gefährlich ist
boolean isDangerous()	Einschätzung des Agenten, ob es sich um eine gefährliche Situation handelt. Überprüft, ob die wahrgenommene Situation größer dem gesetzten Schwellwert ist.
Getters for Rates	Diverse Getter, die True- und False-Positive-Raten aus den Trainings- und realsituationen errechnen und zurückgeben

Zusätzlich existieren noch einige private Feldvariablen, die für interne Berechnungen zuständig sind. Einige hiervon werden in den folgenden Abschnitten aufgegriffen.

Die Main-Methode der Main-Klasse übernimmt die Erstellung der Agenten und ruft die Simulation auf. Die Simulation führt die Trainingsdurchläufe aus und erstellt reale Situationen.

Mehrheitsentscheid

Der Mehrheitsentscheid erfordert, dass sich jeder Agent merkt, wie sich jeder Agent in den Trainingsdurchläufen geschlagen hat um eine Gewichtung einzelner Agenten abzubilden. Dieses

Wissen befindet sich in der Hilfsklasse „AgentMemory“, wovon jeder Agent eine eigene Instanz besitzt und während der Trainingsdurchläufe befüllt.

Soll nun eine reale Situation evaluiert werden, „fragt“ jeder Agent jeden anderen Agenten nach seiner Einschätzung und schaut in der AgentMemory¹ nach, wie gut sich dieser in den Testdurchläufen geschlagen hat. Die Differenz aus den True- und False-Positives des Agenten aus den Testdurchläufen dient als Gewichtung der Einschätzung des Agenten; die summierten Differenzen aller Agenten bilden nun die Einschätzung des fragenden Agenten.

Diese Meinungsbildung ist nur abhängig von den anderen Agenten und ignoriert die eigene Wahrnehmung.

Quorum

Die Festlegung des Quorums ergab sich aus den Werten des Mehrheitsentscheids: Es wurde festgehalten, wie viele Agenten aus der Gruppe benötigt wurden, damit die Gruppe im Rahmen eines True-Positive-Ergebnisses bzw False-Positive-Ergebnisses handelte. Für eine Entscheidung gemäß eines True-Positive-Ergebnisses waren dies durchschnittlich 30%, für False-Positive-Ergebnisse waren dies 20%. Veränderung dieser Werte führte weiterhin zur Verschlechterung der Ergebnisse. Da diese Werte bekannt sind, muss mit dem Quorum keine Trainingsdurchläufe durchgeführt werden.

Aus diesem Wissen heraus wurden zwei Quoren mit genau diesen Werten gewählt, von denen eines erreicht werden muss, damit eine Entscheidung eines Agenten gefällt wird. Abbildung 1 veranschaulicht diese Entscheidungsfindung.

1 Die redundante Speicherung der AgentMemory in jedem einzelnen Agenten ergibt sich daraus, dass das Szenario möglichst “real” nachgestellt werden und jeder Agent sein eigenes Gedächtnis haben sollte. Effizienter wäre eine statische Speicherung, die sich die Agenten teilen, da die Inhalte für jeden Agenten identisch sind.

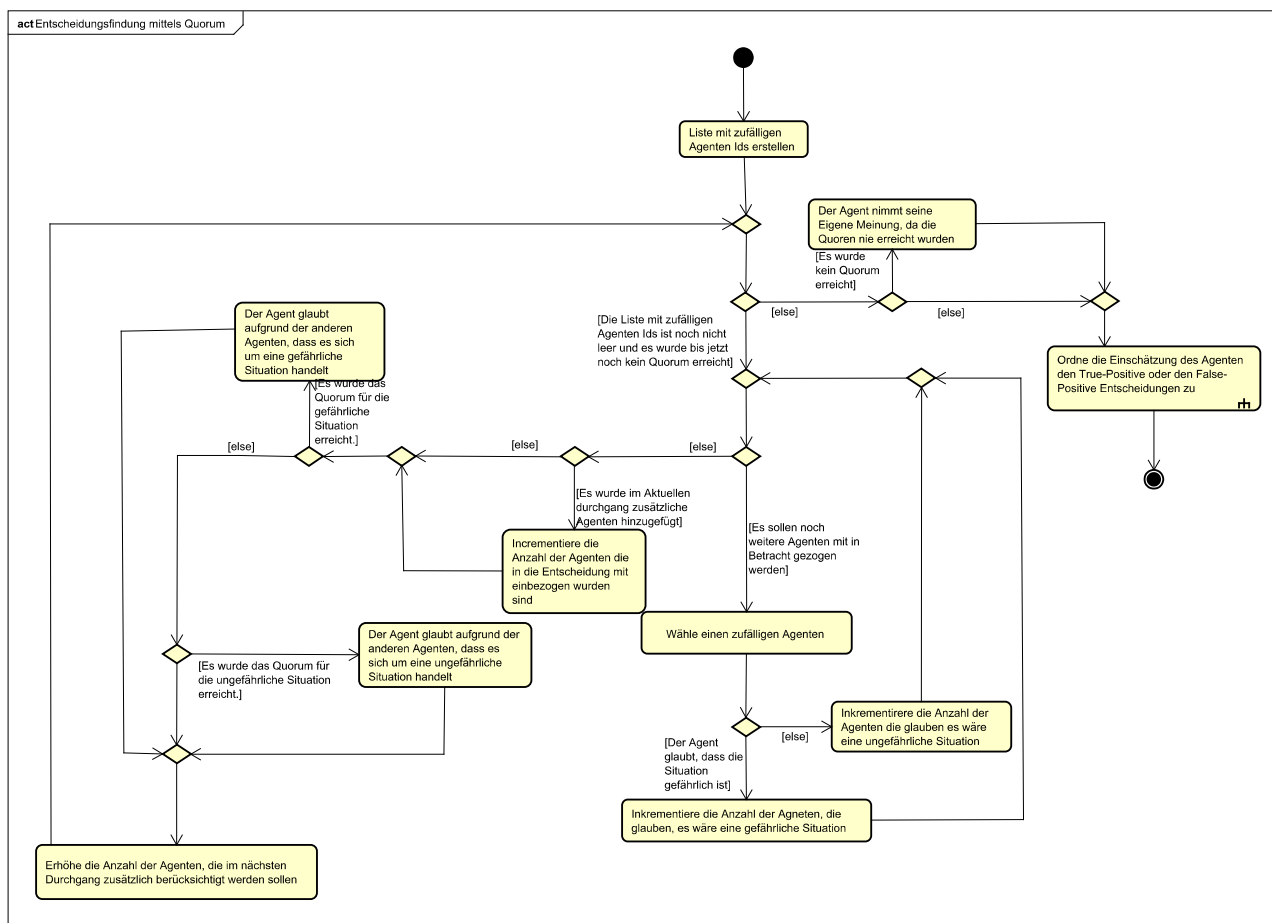


Abbildung 1: Aktivitätsdiagramm zur Entscheidung mittels Quorum

Ein Agent schaut sich 35%² der Gruppe an und prüft, ob die Entscheidung dieser Untergruppe eines der Quoren erfüllt. Ist dies nicht der Fall, so wird ein weiteres Individuum der Gruppe mit in Betracht gezogen. Ist eines der Quoren erfüllt, so ist die Entscheidung des Agenten getroffen. Falls in keinem Fall über mehrere Iterationen ein Quorum erfüllt wird, so agiert der Agent nach eigenem Ermessen ohne Berücksichtigung der anderen Agenten.

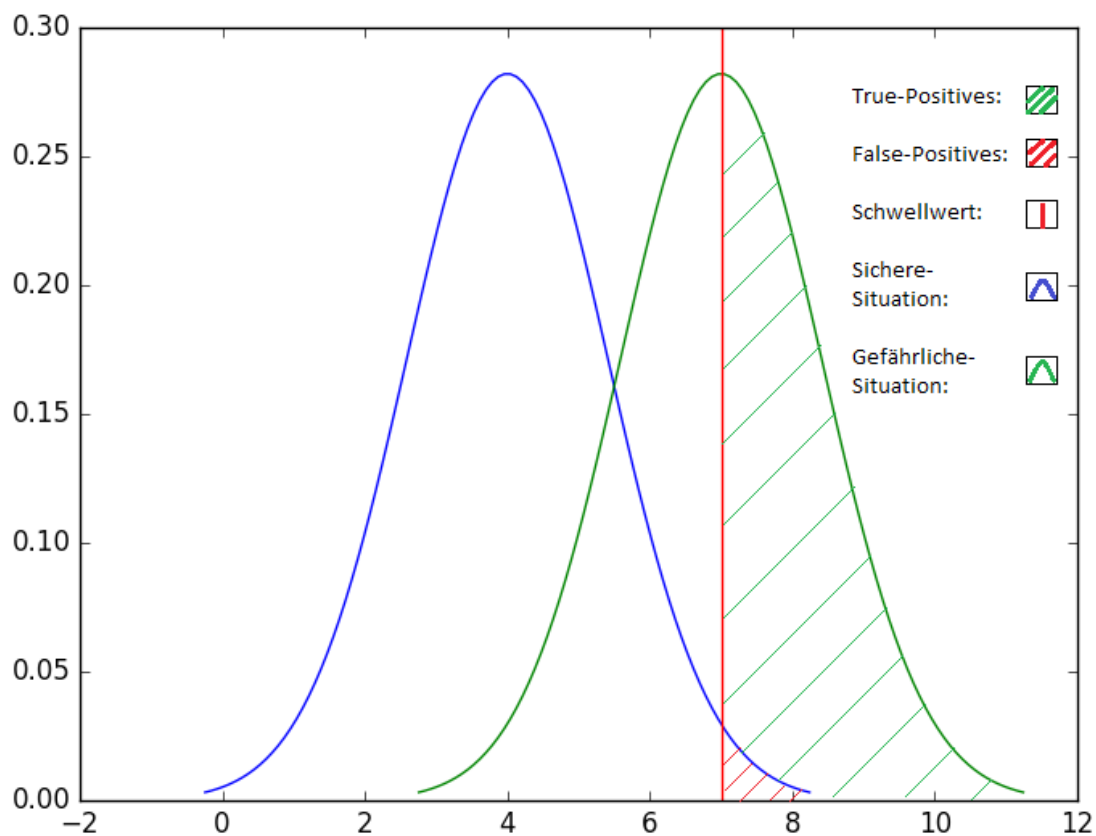
² Dieser Wert über 35% hat sich am ehesten geeignet herausgestellt. Unter diesem Wert wurden die Raten schlechter, wird der Wert erhöht, ähnelt dies eher einem Mehrheitsentscheid.

Gegenüberstellung und Auswertung der Ideen

Um die Ideen zu evaluieren, muss zunächst eine Datenbasis geschaffen werden.

Ohne Berücksichtigung des Trainings und anderer Agenten erreichen die Agenten im Durchschnitt eine True-Positive-Rate von ~51% für gefährliche Situationen und eine False-Positive-Rate von ~23% für ungefährliche Situationen.

Dies wird unter Betrachtung von Abbildung 2 ersichtlich, welche als Skizze dienen soll. Die X-Achse repräsentiert den Wert der Gefahr, die Y-Achse dient lediglich zur Darstellung der Verteilung. Die Wahrnehmung der Agenten wird durch eine Normalverteilung repräsentiert, dessen Breite die Streuung und der Höhepunkt beim Wert der jeweiligen Situation ist. Der Schwellwert liegt im Schnitt bei einem Wert von 7.



Die True-Positiv-Rate von ~51% ist darauf zurückzuführen, dass die Agenten keinerlei Informationen über die Art der Gefahr oder ihre Umgebung besitzen; in anderen Worten: Sie raten

lediglich. Der Schwellwert liegt im Bereich der gefährlichen Situationen, was dazu führt, dass diese Situationen generell besser erkannt werden.

Die False-Positive-Rate von ~23% hängt ebenfalls mit dem Intervall des Schwellwertes [5.5, 8.5) zusammen. Je weiter er in Richtung der X-Achse verschoben wird, desto geringer wird dieser Wert. Ist der Schwellwert geringer, so wird durch die Überschneidung der Verteilungen der Wert der False-Positive-Raten erhöht.

Der Mehrheitsentscheid basiert folglich auf dem Glück der Mehrheit. Durch die individuellen Schwellwerte sind einige Agenten „besser“ im Raten als andere, dies gleicht sich jedoch durch die „schlechteren“ Agenten wieder aus. Je mehr Agenten vorhanden sind, desto eher pendeln sich ihre durchschnittlichen True-Positive-Raten bei 25% und ihr False-Positive-Raten bei ~24% ein. Dies ist zwar eine kleine Verbesserung der False-Positive-Rate, jedoch eine deutliche Verschlechterung der True-Positive-Rate.

Die Idee der Quoren hat sich als deutlich besser erwiesen als der Mehrheitsentscheid. Folgende Tabelle zeigt die Ergebnisse mittels Quorum bei ansteigender Agentenzahl

Anzahl Agenten	10	20	30	40	50	60	70	80	90	100	1000
True-Positive Rate	75%	78%	82%	91%	93%	96%	97%	98%	98%	99%	<100%
False-Positive Rate	55%	19%	19%	19%	19%	19%	18%	17%	15%	14%	>0%

Bei diesen Werten handelt es sich um Durchschnittsbildungen aus lediglich wenigen Testreihen.

Um genauere Ergebnisse zu erzielen, müssten natürlich deutlich mehr Tests durchgeführt werden. Jedoch ist bereits nach so wenig deutlich erkennbar, dass dieses Verfahren besser geeignet ist. Ebenfalls ist anzumerken, dass eine kleinere Gruppe von Agenten zur Folge hat, dass die Ergebnisse stärker schwanken.

Aus den Daten wird ersichtlich, dass dieses Verfahren generell bessere Ergebnisse erzielt: Nicht nur die True-Positiv-Rate steigt enorm, auch die False-Positiv-Rate sinkt. Je Größer die Gruppe ist, desto höher gehen diese Werte auseinander; die richtige Erkennung der Situationen wird also deutlich präziser.

Die Effizienz des Quorums ist hiermit dargelegt, jedoch können wir nicht genauer herleiten, wieso dem so ist. Für ein besseres Verständnis hierfür wäre eine deutlich tiefere Recherche erforderlich