
Klassifikation der Schwierigkeitsgrade von Sudokus mit Methoden des maschinellen Lernens

Classification of Sudoku difficulty levels using methods of machine learning
Bachelor-Thesis von Michael Bräunlein



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fachbereich Informatik
Knowledge Engineering Group
Betreuer Prof. Dr. Johannes Fürnkranz

Zusammenfassung

Das Zahlenrätsel Sudoku ist weltweit bei Rätselliebhabern bekannt und beliebt. Seit seiner Veröffentlichung 1986 begeistern sich immer mehr Menschen für mehr oder weniger schwierige Exemplare. Sudokus finden sich im Internet, in der Rätsellecke der Tageszeitungen und sogar als ganze Bücher, um nur einige Erscheinungsorte zu nennen.

Die Regeln sind einfach zu lernen und doch kann man sich sehr lange mit Sudokus beschäftigen, da die schwersten Sudokus meist nur von Profis gelöst werden können.

Der Spielspass ist sehr stark davon abhängig, dass die Schwierigkeit zum persönlichen Können passt. Ist das Sudoku zu leicht, stellt es keine Herausforderung dar. Ist es zu schwer, kommt schnell ein Gefühl der Überforderung auf.

Die ausgewiesenen Schwierigkeitsstufen von Sudokus aus verschiedenen Quellen haben zwar oft die gleichen Namen wie zum Beispiel "Mittel", unterscheiden sich aber dennoch häufig nach Meinung des Spielers.

Das Ziel dieser Bachelorarbeit ist, Merkmale aus Sudokus zu extrahieren, anhand derer die Sudokus von einem Klassifizierer möglichst zuverlässig in Schwierigkeitsstufen eingeteilt werden können.

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Aufgabenstellung und Zielsetzung | 3 |
| 2 | Einführung | 4 |
| 2.1 | Die Regeln | 4 |
| 2.2 | Begriffserklärung | 5 |
| 3 | Lösungsmethoden | 6 |
| 3.1 | Kandidatenlisten | 6 |
| 3.2 | Full House | 7 |
| 3.3 | Naked Single | 8 |
| 3.4 | Hidden Single | 9 |
| 3.5 | Pointing Pair / Triple | 10 |
| 3.6 | Box-Line Reduction | 11 |
| 3.7 | Naked Subset | 12 |
| 3.8 | Hidden Subset | 13 |
| 3.9 | Fish | 14 |
| 3.9.1 | Basic Fish | 15 |
| 3.10 | Single Digit Patterns | 16 |
| 3.10.1 | Skyscarper | 17 |
| 3.10.2 | 2-String Kite | 18 |
| 3.10.3 | Turbot Fish | 19 |
| 3.10.4 | Empty Rectangle | 20 |
| 3.11 | Wings | 21 |
| 3.11.1 | XY-Wing | 22 |
| 3.11.2 | XYZ-Wing | 23 |
| 3.11.3 | W-Wing | 24 |
| 3.12 | Sue de Coq | 25 |
| 3.13 | Coloring | 26 |
| 3.14 | Almost Locked Set | 27 |
| 3.14.1 | ALS XZ | 28 |
| 3.14.2 | ALS XY Wing | 29 |
| 3.14.3 | ALS Chain | 30 |
| 4 | Klassifikation | 31 |
| 5 | Trainingsdaten | 32 |
| 6 | Merkmalsextrahierung | 33 |
| 7 | Ergebnisse | 34 |
| 7.1 | Allgemeines Vorgehen | 34 |
| 7.2 | Entkopplung von konkreten Zahlen | 34 |
| 8 | Zusammenfassung und Ausblick | 35 |

1 Aufgabenstellung und Zielsetzung

Diese Bachelorarbeit beschäftigt sich mit der Einteilung von Sudokus in verschiedene Schwierigkeitsstufen. Hierzu sollen Methoden des maschinellen Lernens verwendet werden.

Es soll eine Methode gefunden werden, mit der Merkmale aus Sudokus extrahiert werden können, die dann als Feature vektoren in einer .arff Datei¹ gesammelt werden. Die Feature vektoren werden anschließend mit Hilfe von Weka² klassifiziert.

Es werden verschiedene Klassifikatoren und unterschiedliche Parameter betrachtet. Ausserdem werden Optimierungen der Featurevektoren diskutiert.

¹ <http://www.cs.waikato.ac.nz/ml/weka/arff.html>

² <http://www.cs.waikato.ac.nz/ml/weka/>

2 Einführung

Die Vorfahren des heutigen Sudokus waren vermutlich die lateinischen Quadrate, mit denen sich vor allem der Mathematiker Leonhard Euler befasste. Hier ging es darum, in ein Quadrat mit n Zeilen und n Spalten Symbole so einzutragen, dass jedes Symbol in jeder Spalte und Zeile jeweils genau einmal vorkommt.

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 2 | 5 | 4 | 1 | 3 |
| 3 | 4 | 5 | 2 | 1 |
| 4 | 3 | 1 | 5 | 2 |
| 5 | 1 | 2 | 3 | 4 |

Abbildung 2.1: Lateinisches Quadrat

Daraus hat sich das heutige Sudoku entwickelt, das sich nicht nur bei Mathematikern großer Beliebtheit erfreut.

2.1 Die Regeln

Diese Arbeit beschäftigt sich nur mit der meist verbreiteten Art von Sudoku. Dabei spielt man auf einem 9x9 Felder großen Spielfeld, das wiederum in neun 3x3 Felder große Blöcke eingeteilt ist. Weiter handelt es sich nur dann um ein Sudoku, wenn genau eine Lösung vorhanden ist. Ein Sudoku gilt dann als gelöst, wenn jede Zeile, jede Spalte und jeder Block die Ziffern 1 bis 9 genau einmal enthält.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|--|
| 4 | 8 | 1 | 5 | 6 | | 3 | | |
| 7 | 6 | 9 | 3 | 2 | 4 | | 5 | |
| 3 | 5 | | | 7 | | 6 | | |
| | 9 | 7 | 2 | 8 | 5 | 2 | 1 | |
| 1 | | | 1 | 3 | 6 | 5 | 4 | |
| 5 | 4 | | 1 | 3 | 2 | 8 | 6 | |
| | 7 | 6 | 9 | 5 | 3 | | 2 | |
| 2 | | 5 | | 4 | | | 3 | |
| 9 | 3 | 4 | | | | 6 | 5 | |

Abbildung 2.2: Sudoku

2.2 Begriffserklärung

Ein Sudoku besteht aus 81 *Feldern* oder *Zellen*. Diese bilden ein Quadrat der Größe 9x9, das *Grid*. Aufgrund dieser Aufteilung hat ein Sudoku 9 *Zeilen* und 9 *Spalten*. Das Grid wird in 9 Unterquadrate geteilt, die jeweils 3x3 Felder groß sind. Diese werden *Blöcke* genannt. Zeilen, Spalten und Blöcke werden unter dem Begriff *Figur* zusammengefasst. Die Nummerierung der Blöcke erfolgt zeilenweise von links oben nach rechts unten.

Vorgaben sind Zahlen, die schon von Anfang an gegeben sind.

In **Abbildung 2.2** sieht man im mittleren Block sogenannte *Kandidaten*. Ein Kandidat ist eine Zahl, die in der Zelle noch möglich ist. Jede Zelle hat ihre eigene Liste mit Kandidaten.

In der Beschreibung der Lösungstechniken ist es notwendig bestimmte Felder zu betrachten. Hierzu wird eine Abkürzung verwendet, die Zeile und Spalte enthält und somit eine Zelle eindeutig indentifiziert. z2s3 meint zum Beispiel die Zelle in Zeile 2 und Spalte 3.

In der folgenden Abbildung sind die erläuterten Begriffe zum besseren Verständniss eingetragen.

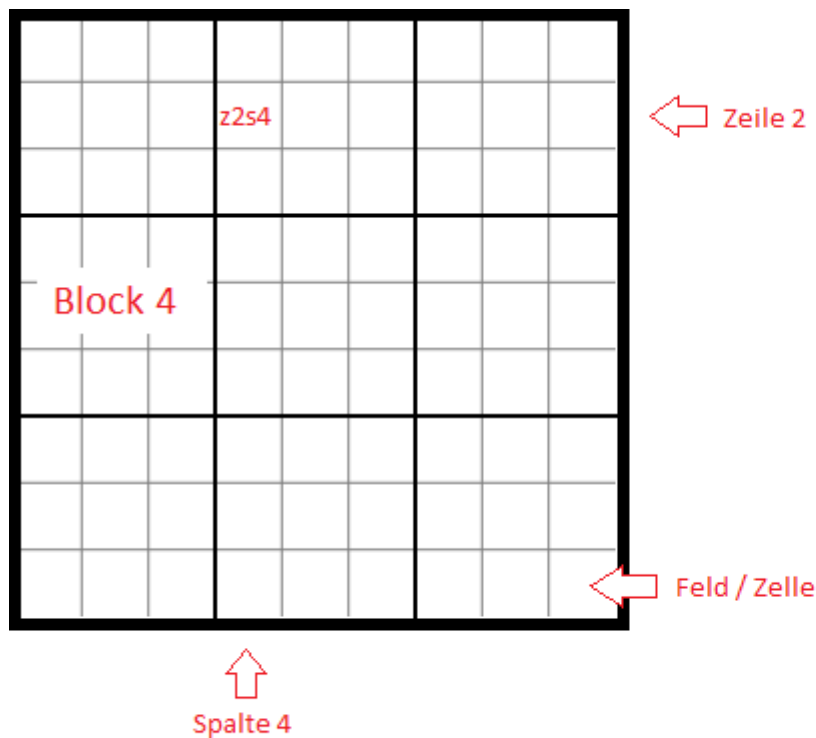


Abbildung 2.3: Begriffe

3 Lösungsmethoden

Alle in dieser Bachelorarbeit beschriebenen Techniken sind nicht im Rahmen dieser Arbeit entwickelt worden, sondern wurden aus verschiedenen Quellen zusammengetragen. Die Beschreibung der Lösungstechniken lehnt sich an die Beschreibung der Quellen an. Teile der Beispiele wurden aus den Quellen entnommen, dies ist entsprechend gekennzeichnet.

Grob kann man die Techniken zum Lösen von Sudokus in zwei Kategorien einteilen. Die erste Kategorie findet Zahlen heraus, die direkt in das Sudoku eingetragen werden können. Die Techniken der zweiten Kategorie entfernen Bedingungen in einzelnen Zellen des Sudokus.

3.1 Kandidatenlisten

Beim Lösen von Sudokus ist es üblich, in jedes Feld die Kandidaten einzutragen, die dort stehen können. Dabei wird vorerst nur die Sudoku Regel berücksichtigt, die besagt, dass in jeder Zeile die Zahlen 1 bis 9 vorkommen müssen. Wenn in einer Zeile nun die Zahl 3 vorkommt, dann kann sie in der selben Zeile nicht nochmal vorkommen, daher kann sie aus allen Kandidatenlisten der Zellen in der selben Zeile gelöscht werden. Dasselbe gilt für Spalten und Blöcke. Immer wenn eine Ziffer in ein Feld eingetragen wird, dann muss der Spieler die Liste der Kandidaten aktualisieren.

Kandidatenlisten sind keine eigene Lösungstechnik, sind aber wesentlicher Bestandteil vieler Techniken.

3.2 Full House

Wenn in einer Figur 8 Zahlen eingetragen sind, dann kann die Technik *Full House* angewendet werden. Da in jeder Figur die Zahlen 1 bis 9 stehen müssen, kann die fehlende Zahl einfach per Ausschluss ermittelt werden.

| | | | | | | | | |
|----------------------------|----------------------------|---------------------|--------------------------|------------------------|----------------------------|--------------------------|------------------------------|------------------------------|
| 2 | ¹ 4 5 6 | 7 | ¹ 4 5 8 | ^{1 3} 5 | ³ 8 | ¹ 6 | ^{1 3} 4 5 6 | ^{1 3} 5 6 9 |
| ¹ 4 5 | 8 | 4 5 6 | ^{1 2} 4 5 | 9 | ^{2 3} 7 | ^{1 2} 6 7 | ^{1 3} 4 5 6 7 | ^{1 2 3} 5 6 7 |
| ¹ 4 5 9 | 3 | 4 5 | 6 | ¹ 5 7 | ² 7 | 8 | ¹ 4 5 7 | ^{1 2} 5 7 9 |
| ³ 5 7 | ⁵ 7 | 8 | ¹ | 6 | 4 | 9 | ¹ 7 | ^{1 2} 7 |
| 6 | 9 | 2 | 7 | 8 | 5 | 3 | ¹ | 4 |
| ⁴ 7 | ⁴ 7 | 1 | 3 | 2 | ⁹ | 5 | ^{7 8} 6 7 | ⁶ |
| ^{4 5} 7 8 | ³ 4 5 6 7 | 9 | ⁵ 8 7 | ^{5 3} 5 | 1 | ⁷ 6 | 2 | ^{5 6} 7 |
| ^{1 3} 5 7 8 | ^{1 2} 5 6 7 | ³ 5 6 | ² 5 8 | 4 | ^{2 3} 6 7 8 | ¹ 6 7 | 9 | ^{1 3} 5 6 7 |
| ^{1 3} 5 7 | ^{1 2} 5 6 7 | ³ 5 6 | ² 5 9 7 | ³ 5 | ^{2 3} 6 7 9 | 4 | ^{1 3} 5 6 7 | 8 |

Abbildung 3.1: Full House

In **Abbildung 3.1** fehlt in Zeile 5 nur noch eine Ziffer. Da die Zahlen 2 bis 9 bereits vorhanden sind, kann in das Feld z5s8 die Zahl 1 eingetragen werden.

3.3 Naked Single

Bei der Technik *Naked Single* werden Kandidatenlisten verwendet. Diese Technik kann angewendet werden, wenn in der Kandidatenliste einer Zelle nur noch ein Kandidat steht. Dieser Kandidat kann dann in die Zelle eingetragen werden. Das funktioniert aufgrund des Aufbaus der Kandidatenlisten. Diese enthalten zuerst alle Kandidaten und es werden immer dann Kandidaten entfernt, wenn dieser Kandidat nicht mehr als Ziffer in der Zelle stehen könnte weil er dort eine Regel verletzen würde. Wenn also nur noch ein Kandidat in der Kandidatenliste steht, dann bedeutet das, dass dieser Kandidat die einzige Ziffer zwischen 1 und 9 ist, die in der Zelle stehen kann ohne eine Regel zu verletzen.

| | | | | | | | | |
|------------------------------|----------------------------|--------------------------|------------------------|------------------------|------------------------------|------------------------------|------------------------------|--------------------------------|
| <div>2 5 6 8</div> | 1 | <div>2 3 5 6</div> | 9 | <div>5 6</div> | <div>5 6</div> | 7 | 4 | <div>2 5 8</div> |
| <div>2 4 5 6 9</div> | <div>2 4 5 9</div> | <div>2 5 6</div> | 8 | <div>4 5 6</div> | <div>1 4 5 6 7</div> | <div>1 5</div> | <div>1 5</div> | 3 |
| <div>4 5 8</div> | 7 | 5 | 3 | 2 | <div>1 4 5</div> | 6 | 9 | <div>1 5 8</div> |
| <div>1 5 6 7 9</div> | <div>5 9</div> | 4 | <div>5 7</div> | 3 | <div>5 7 9</div> | 2 | <div>1 5 6 7 8</div> | <div>1 5 6 7 8 9</div> |
| <div>1 5 7 9</div> | <div>3 5 7 9</div> | <div>3 5</div> | 6 | <div>4 5 8 9</div> | 2 | <div>1 4 5 8 9</div> | <div>1 5 7 8</div> | <div>1 4 5 7 8 9</div> |
| <div>2 5 6 7 9</div> | <div>2 5 9</div> | 8 | <div>4 5 7</div> | 1 | <div>4 5 7 9</div> | 3 | <div>5 6 7</div> | <div>4 5 6 7 9</div> |
| <div>2 4 5</div> | 8 | 1 | <div>2 4 5</div> | 7 | <div>4 5 6 9</div> | <div>4 5 9</div> | 3 | <div>4 5 6 9</div> |
| 3 | <div>2 4 5</div> | <div>2 5 7</div> | <div>1 2 4 5</div> | <div>4 5 6 9</div> | 8 | <div>1 4 5 9</div> | <div>1 5 6 7</div> | <div>1 4 5 6 7 9</div> |
| <div>4 5 7</div> | 6 | 9 | <div>1 4 5</div> | <div>4 5</div> | 3 | <div>1 4 5 8</div> | 2 | <div>1 4 5 7 8</div> |

Abbildung 3.2: Naked Single

Im oben stehenden Beispiel **Abbildung 3.2** sieht man sofort, dass die Kandidatenliste in z3s3 nur noch einen Eintrag enthält. Dieser kann nun einfach eingetragen werden.

3.4 Hidden Single

Auch die Technck *Hidden Single* arbeitet wieder mit Kandidatenlisten. Wenn in einer Figur eine Kandidatenliste die einzige ist, in der eine bestimmte Zahl vorkommt, dann kann diese Zahl direkt in die Zelle eingetragen werden. Wenn in dieser Zelle die Zahl nicht stünde, dann gäbe es in der Figur keine Möglichkeit mehr, dass die Zahl auftaucht und damit wäre die Sudoku Regel verletzt, nach der jede Zahl genau einmal enthalten sein muss.

| | | | | | | | | |
|--------------|------------|------------|----------------|-----|--------------|------------|------------|-------------------|
| 4 5 3 9 | 2 | 8 | 4 5 6 1 9 5 | 7 | 4 6 3 9 9 | 3 | 3 | 3 |
| 4 5 9 | 1 | 6 | 4 5 9 | 8 | 3 | 4 2 9 | 7 | 4 2 9 |
| 4 3 7 9 | 4 | 4 3 9 | 4 6 9 | 2 | 4 | 8 | 5 | 1 |
| 1 | 3 | 7 | 2 | 9 | 5 8 | 4 5 6 8 | 6 | 4 5 6 8 |
| 4 5 6 8 9 | 4 5 6 8 | 4 5 2 9 | 7 | 3 | 1 5 8 | 4 5 6 9 | 1 2 8 9 | 2 4 5 6 8 9 |
| 5 8 9 | 5 8 | 2 5 9 | 1 5 5 | 4 | 6 | 3 | 1 2 8 9 | 7 |
| 2 | 9 | 1 3 4 5 | 1 4 5 | 7 | 1 4 5 | 5 6 | 3 6 8 | 3 5 6 8 |
| 7 3 5 | 7 5 | 5 3 5 | 8 | 6 | 2 5 9 | 1 | 4 | 2 3 5 9 |
| 4 5 6 8 | 4 5 6 8 | 1 4 5 | 3 | 1 5 | 1 2 4 5 9 | 7 | 2 6 8 9 | 2 5 6 8 9 |

Abbildung 3.3: Hidden Single

In **Abbildung 3.3** sieht man, dass die Zahl 6 in der Zeile 3 nur in z3s4 vorkommen kann. Daher kann man sie dort eintragen.

3.5 Pointing Pair / Triple

Bei der Technik *Pointing Pair / Triple* müssen zum ersten mal die Kandidatenlisten mehrerer Felder gleichzeitig betrachtet werden, was diese Technik etwas schwerer macht. Ausserdem ist diese Technik die erste, die Kandidaten aus Kandidatenlisten entfernt und nur bedingt zum Einsetzen von Zahlen in das Sudoku führt.

Es werden die Kandidatenlisten in Blöcken jeweils zeilen- und spaltenweise betrachtet. Die Technik *Pointing Pair / Triple* kann angewendet werden, wenn in einem Block eine Kandidat nur in Kandidatenlisten der selben Zeile oder Spalte vorkommt. Dann kann jedes weitere vorkommen der Zahl in einer Kandidatenliste der selben Zeile oder Spalte entfernt werden. Das gilt, da die Zahl genau einmal in dem Block vorkommen muss. Da alle möglichen Vorkommen der Zahl in der selben Zeile oder Spalte liegen ist klar, dass die Zahl in dieser Zeile oder Spalte vorkommt. Da sie aber kein zweites mal in der Zeile oder Spalte vorkommen darf muss sie aus den Kandidatenlisten entfernt werden, die nicht im selben Block liegen.

| | | | | | | | | |
|------------------------------------|--------------------------------------|----------------------------------|------------------------------------|----------------------------------|----------------------------------|------------------------------------|------------------|------------------------------------|
| 3 | 4 | ¹ 5 ₉ | ^{1 2} 5 ₈ | ⁵ 8 ₉ | 6 | ^{1 2} 5 | 7 | ^{1 2} 8 |
| ¹ 5 ₇ 6 | 8 | ¹ 5 ₆ | ^{1 2} 4 ₇ 5 | ^{4 5} 7 | ¹ 5 ₇ | 9 | 3 | ^{1 2} 4 |
| ¹ 5 ₇ 9 | ¹ 5 ₇ | 2 | ¹ 4 ₇ 5 | 3 | ¹ 5 ₇ 9 | ¹ 4 ₅ | 6 | ¹ 4 ₈ |
| ² 4 ₈ 5 | ^{2 3} 5 ₆ | ³ 4 ₈ 5 | ⁵ 7 ₈ | 1 | ⁵ 7 ₉ | ^{2 3} 4 ₆ | ² 4 | ^{2 3} 4 ₇ 6 |
| ^{1 2} | 9 | 7 | 3 | 6 | 4 | 8 | 5 | ^{1 2} |
| ¹ 4 ₈ 5 | ¹ 3 ₅ 6 | ¹ 3 ₄ 5 | ⁵ 7 ₈ | ⁵ 7 ₈ 9 | 2 | ¹ 3 ₄ 6 | ¹ 4 | ¹ 3 ₄ 6 |
| ^{1 2} 4 ₇ 5 | ^{1 2 3} 5 ₇ 6 | ¹ 3 ₈ 9 | ¹ 4 ₇ 5 | ^{4 5} 7 | ¹ 5 ₇ | ^{1 2 3} 4 ₆ | ^{1 2} 4 | ^{1 2 3} 4 ₆ |
| ^{1 2} 4 ₇ 5 | ^{1 2 3} 5 ₇ | ¹ 3 ₄ 5 | 6 | ^{4 5} 7 | 8 | ^{1 2 3} 4 | 9 | ^{1 2 3} 4 |
| ¹ 4 ₆ | ¹ 6 ₄ | ¹ 4 ₆ | 9 | 2 | 3 | 7 | 8 | 5 |

Abbildung 3.4: Pointing Pair / Triple

In **Abbildung 3.4** betrachten wir Block 8. Hier ist das Vorkommen der Zahl 1 in den Kandidatenlisten auf Zeile 7 beschränkt. Wie oben beschrieben können nun alle weiteren vorkommen in der selben Zeile, die nicht in Block 8 liegen aus den Kandidatenlisten entfernt werden. Im vorliegenden Beispiel führt das allerdings nicht dazu, dass eine neue Zahl in das Sudoku eingetragen wird. Dennoch ist das Sudoku nun genauer bestimmt, da weniger Möglichkeiten übrig sind.

3.6 Box-Line Reduction

Die Technik *Box-Line-Reduction* ist verwandt mit der Technik *Pointing Pair / Triple*. Hier wird das Sudoku zeilen- und spaltenweise betrachtet. Ist das Vorkommen einer Zahl in den Kandidatenlisten auf einen Block beschränkt, dann kann jedes weitere Vorkommen der Zahl aus den Kandidatenlisten der Zellen des selben Blocks gestrichen werden, die nicht in der Zeile oder Spalte liegen. Die Begründung dafür ist ähnlich der Begründung bei *Pointing Pair / Triple*. Da die Zahlen 1 bis 9 jeweils genau einmal in der Zeile oder Spalte vorkommen müssen und dieses Vorkommen auf einen Block beschränkt ist, ist klar, dass die Zahl letztendlich in diesem Block vorkommt und zwar in der gefundenen Zeile oder Spalte. Die Zahl kann aber nicht zweimal in dem Block vorkommen, daher kann sie aus den Kandidatenlisten des Blocks gelöscht werden, deren Zellen sich nicht in der Reihe oder Spalte befinden.

| | | | | | | | | |
|---|---|----------------|--|--|---------------------------------|-------------------------------|---|---|
| 7 | 6 | 2 | 4 ³ 5 | 4 ³ 5 ⁹ | 8 | 5 ⁹ 4 ³ | | 1 |
| 9 | 8 | 4 ³ | 1 ² 2 ³ 4 ⁷ 5 | 1 ² 2 ⁷ 4 ⁵ 5 | 4 ⁷ | 2 ⁵ 4 ³ | | 6 |
| 1 | 5 | 4 ³ | 4 ² 3 | 4 ² 3 | 6 ⁹ 4 ⁶ 6 | 2 ⁹ | 8 | 7 |
| 4 | 7 | 8 | 2 ⁵ | 2 ⁵ | 3 | 1 | 6 | 9 |
| 5 | 2 | 6 | 1 ⁴ | 1 ⁴ | 9 | 8 | 7 | 3 |
| 3 | 1 | 9 | 8 | 7 ⁶ 7 | 6 | 4 | 2 | 5 |
| 8 | 3 | 5 | 4 ⁷ | 4 ⁷ | 1 | 6 | 9 | 2 |
| 2 | 9 | 7 | 6 | 8 | 5 | 3 | 1 | 4 |
| 6 | 4 | 1 | 9 | 3 | 2 | 7 | 5 | 8 |

Abbildung 3.5: Box-Line Reduction

Wir betrachten Spalte 6 in **Abbildung 3.5**. Hier sieht man, dass das Vorkommen der Zahl 4 in dieser Spalte auf Block 2, den oberen Block, beschränkt ist. Anhand dieser Spalte sieht man also, dass die Zahl 4 entweder in z2s6 oder in z3s6 steht, also in jedem Fall in Block 2. Daher kann die Zahl 4 aus den Kandidatenlisten der anderen Zellen in Block 2 gestrichen werden.

3.7 Naked Subset

Die Technik *Naked Subset* ist ein Überbegriff für die Techniken *Naked Pair*, *Naked Triple* und *Naked Quadruple*. Alle Techniken arbeiten nach dem selben Prinzip, der Unterschied liegt in der Anzahl der verwendeten Kandidatenlisten. Bei *NakedSubsets* sucht man nach Paaren, Tripeln oder Quadrupeln von Zellen in Figuren, nach Kandidatenlisten einer bestimmten Eigenschaft. Die Vereinigung der Listen muss eine bestimmte Anzahl Elemente enthalten. Bei Paaren sind das zwei, bei Tripeln drei und bei Quadrupeln vier Einträge in den Kandidatenlisten.

Findet man zum Beispiel ein Paar, das nur noch die selben beiden Zahlen enthalten kann dann ist klar, dass keine der Zahlen anderswo in der Figur stehen kann, da sonst für eine der Zellen keine Zahl mehr übrig bleibt. Daher können die beiden Zahlen dann aus den Kandidatenlisten aller anderen Zahlen aus der Figur entfernt werden. Die Begründung für Tripel und Quadrupel ist analog.

Es ist nicht nötig nach mehr als Quadrupeln zu suchen, da für jedes Naked Quintupel ein Hidden Quadrupel existiert.

| | | | | | | | | | | | |
|---|--------|--------|---|--------|---|---|---|---|---|-------------|-------------|
| 7 | 6 | 1 7 | 6 | 1 7 | 5 | 2 | 9 | 4 | 3 | 8 | 7 5 |
| 2 | 3 | | 3 | 2 5 | 9 | 1 | 7 | 8 | 6 | 4 | 2 5 9 |
| 4 | 8 | | | 2 7 | 9 | 3 | 5 | 6 | 1 | | 2 9 7 |
| 2 | 6 | | 6 | 4 | | 8 | 3 | 7 | 5 | 2 6 9 | 1 |
| 2 | 3 | | 3 | 2 | | 4 | 1 | 5 | 7 | 2 6 9 | 2 9 |
| 5 | 1 7 | 1 7 | | | | 6 | 2 | 9 | 8 | 3 | 4 |
| 9 | 5 | 3 | | | | 7 | 8 | 2 | 4 | 1 | 6 |
| 1 | 2 | 6 | | | | 5 | 4 | 3 | 9 | 7 | 8 |
| 7 | 8 | | 4 | 7 | 8 | 9 | 6 | 1 | 2 | 5 | 3 |

Abbildung 3.6: Naked Subset - Naked Triple

In **Abbildung 3.6** findet man das *Naked Subset*, bei dem es sich um ein *Naked Triple* handelt, in Spalte 2. Hier hat die Vereinigung der Kandidatenlisten der Zellen z2s2, z4s2 und z5s2 genau drei Einträge: 3, 6 und 9. Es gibt offensichtlich keine andere Möglichkeit, als die drei Zahlen auf diese Felder zu verteilen, Demnach können sie in der Reihe sonst nicht vorkommen und können aus den Kandidatenlisten der anderen Zellen entfernt werden.

3.8 Hidden Subset

Analog zu den *Naked Subset*-Techniken ist auch *Hidden Subset* ein Sammelbegriff. Er beinhaltet die Techniken *Hidden Pair*, *Hidden Triple* und *Hidden Quadruple*. Auch hier ändert sich nur die Anzahl der betrachteten Kandidatenlisten. Hier soll exemplarisch die Technik *Hidden Tuple* erklärt werden, im folgenden Beispiel wird dann die Technik *Hidden Quadruple* angewendet.

Wenn man in einer Figur zwei Zahlen findet, die ausschließlich in den zwei gleichen Zellen vorkommen können, dann müssen diese beiden Zahlen in die beiden Zellen gesetzt werden. Daher kann man alle anderen Kandidaten in den Zellen von der Kandidatenliste streichen.

| | | | | | | | | |
|---|---|---|---------------------------|---------------------------------|-------------------------------|-----------------------------|---------------------------|---------------------------|
| 8 | 1 | 6 | 5 | 7 | 3 | 2 | 9 | 4 |
| 3 | 9 | 2 | ¹ | ¹ _{6 4 6 4} | ¹ ₈ | ¹ _{7 8} | ¹ ₅ | ⁵ ₇ |
| 4 | 5 | 7 | 2 | ¹ ₈ | 9 | ³ ₈ | ¹ ₃ | 6 |
| 9 | 4 | 1 | ³ ₇ | ^{2 3} ₇ | 2 | 5 | 6 | 8 |
| 7 | 8 | 5 | 4 | 9 | 6 | 1 | 2 | 3 |
| 6 | 2 | 3 | 8 | ¹ ₅ | ¹ ₅ | ⁷ ₉ | 4 | ⁷ ₉ |
| 2 | 7 | 9 | ³ ₆ | ³ _{4 5 6 8} | ³ _{4 5 8} | ⁴ ₆ | ⁵ ₃ | 1 |
| 1 | 3 | 8 | ⁶ ₉ | ² _{4 5 6 8} | ² _{4 5} | ⁴ ₆ | 7 | ⁵ ₉ |
| 5 | 6 | 4 | ¹ ₇ | ³ ₉ | ¹ ₇ | ³ ₉ | 8 | 2 |

Abbildung 3.7: Hidden Subset - Hidden Quadruple

In **Abbildung 3.7** betrachten wir den Block 8 und in diesem Block die Zellen z6s5, z6s6, z7s5 und z7s6. Nur in diesen Zellen können die Zahlen 4, 5, 6 und 8 vorkommen. Da wir diese vier Zahlen nun auf die vier Zellen verteilen müssen gibt es dort keinen Platz mehr für andere Zahlen. Diese können also aus den Kandidatenlisten entfernt werden.

3.9 Fish

Die *Fish* Methoden sind ein Sammelbegriff für eine ganze Gruppe von Methoden, die alle nach dem gleichen Prinzip arbeiten. Wie echte Fische hat dieses Prinzip eine sehr große Anzahl Unterarten hervorgebracht. Kleine Fische wie zum Beispiel X-Wing sind von geübten Sudoku Spielern noch zu finden, wenn die Fische allerdings größer werden, dann sind sie nur noch mit sehr hohem Aufwand manuell zu finden und daher eher zur Verarbeitung mit dem Computer gedacht.

Auf einer Internetseite, die sich unter anderem mit den Lösungsmethoden für Sudokus befasst, findet sich die folgende Erklärung zur Funktionsweise von Fischen.

[...] Man suche eine bestimmte Anzahl von Häusern, die sich nicht überschneiden. Diese Häuser werden als Base-Sets (Basismengen) bezeichnet (Set wird hier synonym für Haus verwendet), die in diesen Häusern enthaltenen Kandidaten sind die Basiskandidaten. Nicht überschneiden bedeutet hier, dass kein Basiskandidat in mehr als einem Haus enthalten sein darf, die Häuser selbst dürfen sich schon überlappen. Nun suche man eine gleiche Anzahl an sich nicht überschneidenden Häusern, die alle Basiskandidaten abdecken (engl.: cover). Diese neuen Häuser sind die Cover-Sets, sie enthalten die Coverkandidaten. Wenn eine solche Kombination existiert, können alle Coverkandidaten gelöscht werden, die nicht gleichzeitig Basiskandidaten sind.^{1 2}

¹ Quelle: http://hodoku.sourceforge.net/de/tech_fishg.php

² Häuser stehen hier für Figuren

3.9.1 Basic Fish

Die Techniken *X-Wing*, *Swordfish* und *Jellyfish* sind die einfachsten Unterarten der Fische. Sie funktionieren nach dem selben Prinzip, nur dass eine unterschiedliche Anzahl an Figuren betrachtet wird, ähnlich zu *Naked Subset* und *Hidden Subset*. Hier wird stellvertretend die Technik *X-Wing* erklärt und am Beispiel gezeigt.

Dazu sucht man zwei Spalten oder Zeilen, die ausschließlich in den selben zwei Zellen einen bestimmten Kandidaten, die Fischziffer, beinhalten. Nun kann man aus dem jeweils anderen Paar von Figuren (Spalte oder Zeile), deren Position durch die zwei gefundenen Zellen festgelegt wird, alle Fischziffern löschen, die nicht gleichzeitig in einer der zuerst ausgesuchten Figuren liegen.

Da die Fischziffer in den beiden zuerst ausgesuchten Figuren nur an jeweils zwei Stellen liegen kann und diese sich paarweise gegenseitig ausschließen ist klar, dass jedes Vorkommen der Fischziffer in den zuletzt ausgesuchten Figuren in der Überschneidung mit den ersten Figuren liegen muss.


| | | | | | | | | |
|--------|--------|---|----------|-------------|---------------|--------|--------|-----|
| 5 8 | 4 | 1 | 7 | 2 | 9 | 6 8 | 3 | 5 6 |
| 7 | 6 | 9 | 1 8 | 1 5 8 | 3 | 4 | 5 8 | 2 |
| 5 8 | 3 | 2 | 6 | 4 | 5 8 | 7 | 1 | 9 |
| 4 | 2 8 | 3 | 9 | 5 8 | 2 5 6 8 | 1 | 7 | 5 6 |
| 6 | 2 8 | 7 | 1 2 8 | 1 5 8 | 4 | 9 | 5 8 | 3 |
| 1 | 9 | 5 | 3 | 7 | 6 8 | 6 8 | 2 | 4 |
| 2 | 1 | 4 | 5 | 6 | 7 | 3 | 9 | 8 |
| 3 | 7 | 6 | 2 8 | 9 | 2 8 | 5 | 4 | 1 |
| 9 | 5 | 8 | 4 | 3 | 1 | 2 | 6 | 7 |

Abbildung 3.8: Basic Fish - X-Wing

Ein Beispiel für die Technik *X-Wing* findet sich in **Abbildung 3.8**. Hier wurden als erste Figuren die Zeilen 2 und 5 gewählt. Diese enthalten die Fischziffer 5 nur an den Stellen 5 und 8. Wichtig ist, dass es in den beiden Zeilen die gleichen Positionen sind. Da die Ziffer 5 und Zeile 2 nur an den Positionen 5 und 8 stehen kann werden beiden Fälle nun getrennt betrachtet. Steht in z2s5 die Ziffer 5, dann muss sie auch in z5s8 stehen, da sonst die Zeile 5 die Ziffer 5 nicht enthalten würde. Umgekehrt gilt: Steht in z2s8 die Ziffer 5, dann muss sie auch in z5s5 stehen. In jedem der beiden möglichen Fälle gilt, dass sowohl in Spalte 5 als auch in Spalte 8 die Ziffer 5 vorkommt und zwar in den Zeilen 2 und 5. Daher kann aus allen anderen Zellen der Spalten die Fischziffer 5 gelöscht werden, falls sie in den Kandidatenlisten vorhanden ist.

3.10 Single Digit Patterns

3.10.1 Skyscarper



3.10.2 2-String Kite

3.10.3 Turbot Fish

3.10.4 Empty Rectangle

3.11 Wings

3.11.1 XY-Wing

3.11.2 XYZ-Wing

3.11.3 W-Wing

3.12 Sue de Coq

3.13 Coloring

3.14 Almost Locked Set

3.14.1 ALS XZ

3.14.2 ALS XY Wing

3.14.3 ALS Chain

4 Klassifikation

Unter einer Klassifikation versteht man in der Informatik das Einteilen von Objekten in vorher festgelegte Klassen. Diese Einteilung wird von einem Algorithmus durchgeführt, der anhand von festgelegten Merkmalen jedem Objekt eine Klasse zuordnet. Einen solchen Algorithmus nennt man Klassifikator. Um die Qualität eines Klassifikators zu analysieren gibt es verschiedene Methoden.

- Accuracy - Die Anzahl der richtig zugeordneten Klassen
- Recall - Der Anteil der positiven Beispiele, die auch positiv klassifiziert wurden
- Precision - Der Anteil der positiv klassifizierten Beispiele, die auch positiv sind

Ein Klassifikator benötigt vor der Phase der Klassifikation aber zunächst einmal eine Trainingsphase, in der er anhand von Beispielen lernt. Mit dem erlerten Wissen wird anschließend die Einteilung in die Klassen vorgenommen.

Es gibt viele verschiedene Ansätze für Klassifikatoren, von denen die wichtigsten in einem open source Framework implementiert sind. Dieses Framework heisst Weka¹ und wurde im praktischen Teil dieser Bachelorthesis verwendet.

Weka arbeitet unter anderem mit dem .arff² Format. In einer .arff Datei befindet sich neben den Metadaten hauptsächlich eine Sammlung von Featurevektoren. Jeder Featurevektor beschreibt ein zu klassifizierendes Objekt. Ein Eintrag in einem Featurevektor beschreibt eine Eigenschaft des beschriebenen Objekts. Das könnte bei Fahrzeugen zum Beispiel die Anzahl der Reifen sein. Bezogen auf Sudokus bedeutet das, dass jedes Sudoku durch einen Featurevektor beschrieben wird. Jede zur Klassifikation verwendete Eigenschaft eines Sudokus ist dann ein Wert im entsprechenden Featurevektor. Die Schwierigkeitsgrade der Sudokus sind die vorgegebenen Klassen.

Jeder Klassifikator in Weka hat als Eingabe eine Liste von Featurevektoren. Möchte man also das Zuordnen von Sudokus zu Schwierigkeitsgraden mit Weka realisieren, dann muss eine Methode entwickelt werden, die aus einem gegebenen Sudoku einen Featurevektor extrahiert.

Die Qualität der resultierenden Klassifikation ist sehr stark von der Wahl der Einträge des Featurevektors abhängig.

¹ <http://www.cs.waikato.ac.nz/ml/weka/index.html>

² <http://weka.wikispaces.com/ARFF>

5 Trainingsdaten

Um fundierte Aussagen über die Qualität der Klassifikation treffen zu können, wird eine große Menge an Trainingsdaten benötigt. Diese müssen bereits vollständig in Schwierigkeitsstufen eingeteilt worden sein. Das ist nötig, da der Klassifikator eine Schwierigkeitsstufe zuordnet und die Qualität der Zuordnung evaluiert werden soll. Um also festzustellen, ob der Klassifikator die richtige Klasse zugeordnet hat muss diese bekannt sein.

Kostenlose und frei verfügbare Sudokus in digitaler Form mit definiertem Schwierigkeitsgrad lassen sich nicht leicht finden. Daher habe ich bei einigen großen Zeitungen, aus deren Websites Sudokus zu finden waren, nachgefragt, ob es möglich ist, ihre Sudokusammlungen zur Verfügung zu stellen. Die Anfragen wurden aber leider abgelehnt. Auf eine Anfrage an die Website <http://sudoku.soeinding.de/> wurden von sieben Schwierigkeitsgraden jeweils 32 Sudokus bereitgestellt. Da diese Trainingsdaten nicht ausreichten wurden mit dem open source Programm Hodoku¹ jeweils 1000 Sudokus von fünf unterschiedlichen Schwierigkeitsgraden generiert.

¹ <http://hodoku.sourceforge.net/de/index.php>

6 Merkmalsextrahierung

7 Ergebnisse

7.1 Allgemeines Vorgehen

7.2 Entkopplung von konkreten Zahlen

8 Zusammenfassung und Ausblick
