

# **aiNet - Ein künstliches Immun Netzwerk zur Datenanalyse**

Ausarbeitung im Rahmen des Seminars  
**„Anwendungen und Algorithmen basierend auf  
Prinzipien eines künstlichen Immunsystems“**  
Sommersemester 2005

*Betreut durch:*

Thomas Stibor

Fachbereich Informatik  
Fachgebiet Sicherheit in der Informationstechnik  
Technische Universität Darmstadt

*von:*

**Christian Jacobi** (cjacobi@rbg.informatik.tu-darmstadt.de)

## **Zusammenfassung**

Diese Ausarbeitung beschäftigt sich mit aiNet, einem vom Immunsystem inspirierten Netzwerk, das von Leandro Nunes de Castro und Fernando José Von Zuben entwickelt wurde. Es werden die zugrundeliegende Motivation, die zugrundeliegenden Ideen des Immunsystems, die theoretischen Hintergründe von aiNet, der aiNet Algorithmus, die Charakterisierung von aiNet, die Knowledge Extraction und einige Beispiele der Anwendung vorgestellt. Letztendlich wird ein eher kritisches Fazit gezogen.

## Inhaltsverzeichnis

<b>1</b>	<b>Grundlegendes</b>	<b>3</b>
<b>1.1</b>	<b>Motivation</b>	<b>3</b>
<b>1.2</b>	<b>Ziel bei der aiNet Entwicklung</b>	<b>3</b>
<b>2</b>	<b>Das Immunsystem</b>	<b>4</b>
<b>3</b>	<b>Theoretische Grundlagen von aiNet</b>	<b>5</b>
<b>4</b>	<b>Der Lernalgorithmus von aiNet</b>	<b>7</b>
<b>5</b>	<b>Charakterisierungen von aiNet</b>	<b>8</b>
<b>5.1</b>	<b>Charakterisierung als Netzwerk</b>	<b>8</b>
<b>5.2</b>	<b>Charakterisierung als evolutionärer Algorithmus</b>	<b>8</b>
<b>6</b>	<b>Knowledge Extraction</b>	<b>9</b>
<b>6.1</b>	<b>Möglichkeit 1 der Knowledge Extraction</b>	<b>9</b>
<b>6.2</b>	<b>Anforderungen an Knowledge Extraction Algorithmus</b>	<b>9</b>
<b>6.3</b>	<b>Möglichkeit 2 der Knowledge Extraction</b>	<b>10</b>
<b>6.4</b>	<b>Möglichkeit 3 der Knowledge Extraction</b>	<b>11</b>
<b>6.5</b>	<b>Möglichkeit 4 der Knowledge Extraction</b>	<b>12</b>
<b>6.6</b>	<b>Diskussion der Möglichkeiten</b>	<b>12</b>
<b>7</b>	<b>Beispiele</b>	<b>13</b>
<b>7.1</b>	<b>SPIR</b>	<b>13</b>
<b>7.2</b>	<b>CHAINLINK</b>	<b>14</b>
<b>7.3</b>	<b>5-NLSC</b>	<b>15</b>
<b>8</b>	<b>Fazit</b>	<b>16</b>
<b>9</b>	<b>Literatur</b>	<b>17</b>

# **1 Grundlegendes**

## **1.1 Motivation**

Das Menschliche Immunsystem bietet vielfältige Möglichkeiten zur Inspiration bei der Entwicklung von Algorithmen, die in der Informatik verwendet werden können. Sinnvolle Ansätze aus der informationstheoretischen Sicht, die hier verwendet werden, sind die „immune network theory“, „clonal selection“ und „affinity maturation“.

## **1.2 Ziel bei der aiNet Entwicklung**

Bei der Entwicklung von aiNet ging es darum Datensätze handlicher zu machen, konkret bedeutet dies sie zu clustern, filtern und die Redundanz zu reduzieren. Clustern bedeutet, dass Datenhäufungen an bestimmten Stellen, also Strukturen im Datensatz, erkannt werden und zu Clustern zusammengefasst werden. Auf dieser Grundlage können dann die weiteren Schritte wie Filterung und Redundanzreduzierung anhand von Parametern geschehen. Die Datensätze die verarbeitet werden sollen können hochdimensional sein, hier werden allerdings zur besseren Visualisierung nur Datensätze bis zur 3ten Dimension gezeigt bzw. verarbeitet.

Ein wesentlicher Eckpunkt bei der aiNet Entwicklung war, die Ideen des Immunsystems zu benutzen, aber in keiner Weise das Immunsystem oder irgendwelche Immuneffekte nachzubilden. Man wollte sich also auf keinen Fall in der Entwicklung durch Gegebenheiten die im Immunsystem unumgänglich sind einschränken lassen, sondern nur das was man als sinnvoll erachtet verwenden.

## 2 Das Immunsystem

Da das Immunsystem im Laufe des Seminars oft und ausführlich behandelt wurde, wird an dieser Stelle darauf verzichtet die Grundlagen erneut zu behandeln. Hier soll die „immune network theory“, „clonal selection“ und „affinity maturation“ erläutert werden. Allerdings wird hier nicht der Anspruch auf Vollständigkeit erhoben, da dies für das Verständnis wie aiNet arbeitet nicht nötig ist. Es werden nur die für aiNet wichtigsten Eigenschaften vorgestellt. Die immune network theory wurde ursprünglich von Jerne (1974a) entwickelt, wobei es viele Abwandlungen und Interpretationen gibt.

In der immune network theory besteht das Immunsystem aus einem regulierten Netzwerk aus Zellen und Molekülen. Die relevanten Teile des Immunsystems sind nicht nur die Zellen, sondern auch ihre Interaktionen. Die Zellen und Moleküle erkennen sich gegenseitig und auch selbst. Antikörper erkennen also Antikörper und Antigene. Somit ist das Netzwerk auch aktiv wenn keine Antigene vorhanden sind. Die Immunzellen können positiv oder negativ auf eine Erkennung reagieren. Eine positive Antwort löst eine Zellteilung, eine Zellaktivierung oder Antikörper Ausschüttung aus. Eine negative Antwort ist eine Tolerierung und Netzwerkunterdrückung. Einfluss auf die Gesamtpopulation haben neue Zellen, der Tod unstimulierter Zellen und die Reproduktion von stimulierten Zellen.

Bei der clonalen Selektion geht es darum, dass stimulierte Zellen sich vermehren. Antikörper können sich nur vermehren wenn sie Antigene erkennen, sie also stimuliert werden. Dies stellt die Selektion dar. Der Prozess der Vermehrung besteht aus einfachen klonen, bis hin zu Mutationen der Clone. Durch die Mutation der Clone und die Vermehrung positiv stimulierter Zellen, sowie durch die Eliminierung der negativ stimulierten Zellen, verbessert sich stetig die Fähigkeit des Netzwerks Antigene zu erkennen. Dieser Gesamtprozess wird affinity maturation genannt.

### 3 Theoretische Grundlagen von aiNet

Um alles erfassen zu können was im aiNet Universum passieren kann und soll wurde der ShapeSpace  $S$  definiert. Der ShapeSpace ist eine  $L$  dimensionale Abbildung der Realität. Jede Dimension kann für eine mögliche Physio-Chemische Messung stehen, wenn man dies aus der Sicht des Immunsystem basierten Ansatzes betrachtet. Im Gegensatz zu vielen anderen immunen oder genetischen Algorithmen arbeitet aiNet auf realen bzw. floating point Variablen, und nicht auf ganzen Zahlen bzw. integer variablen. Oder kurz gesagt:  $S = \mathbb{R}^L$ .

Antikörper und Antigene werden, nach dem Prinzip alles im ShapeSpace  $S$  stattfinden zu lassen, als  $L$  dimensionaler String (oder Vektor) definiert. Zur Vereinfachung wird allerdings nicht zwischen der Oberfläche des Moleküls/Zelle und dem Molekül/Zelle selbst unterschieden, wodurch die Sichtweise als String nicht weiter verfeinert werden muss.

Antigene werden mit Ag abgekürzt. Die Antigene stehen in aiNet für die Daten, dargestellt im ShapeSpace  $S$ .

Antikörper (Antibody) werden mit Ab abgekürzt. Die Antikörper stehen für die Netzwerkknoten von aiNet und befinden sich auch im ShapeSpace. Das Ziel ist die gleiche räumliche Verteilung zu erreichen wie die Antigene.

Antikörper-Antikörper und Antigen-Antikörper Interaktionen werden als Konnetktivitätsgraph dargestellt. Wobei die Konnektivität eine Distanzmetrik ist, diese wird über die Affinität approximiert. Die Distanz ist invers proportional zur Affinität, es gilt also je ähnlicher – d.h. räumlich näher - sich Ab-Ab oder Ab-Ag sind, desto höher ist die Affinität. Die Folge der Interaktionen sind entweder Netzwerkunterdrückung oder Netzwerkaktivierung.

#### Definition:

Das aiNet ist ein **kantengewichteter Graph**, der nicht notwendigerweise vollständig verbunden ist, bestehend aus einem Satz aus Knoten, genannt **Antikörper** und einem Satz aus Knotenpaaren, genannt **Kanten**, mit einer Zahl, die **Gewicht** oder **Verbindungsstärke** genannt wird und jeder verbundenen Kante zugeordnet wird.

Bei Antikörper-Antigen-Paaren (Ab-Ag) ist die Affinität  $d$  das Gewicht. Die Affinität ist invers proportional zur Distanz. Eine Ab-Ag-Erkennung erfolgt, wenn die Affinität  $d$  hoch genug ist. Die Folge ist Netzwerkaktivierung. Netzwerkaktivierung wird durch Zellvermehrung realisiert, konkret durch Clonen und Mutieren.

Bei Antikörper-Antikörper-Paaren (Ab-Ab) ist die Ähnlichkeit (similarity)  $s$  das Kantengewicht, sie ist auch invers proportional zur Distanz. Die Erkennung findet statt wenn die Ähnlichkeit groß genug ist, dies wird über die Variable  $\sigma_s$ , den Suppression Threshold gesteuert. Findet eine Ab-Ab-Erkennung statt, so wird eine Netzwerkunterdrückung ausgelöst. Netzwerkunterdrückung bedeutet Zelltod. Hier bedeutet dies einfach Entfernung einer der sich erkennenden Antikörper.

Ist ein Antikörper weder zu einem Antigen affin, noch zu einem Antikörper ähnlich, führt dies auch zum Zelltod, der Antikörper wird also entfernt. Dies ist nötig da es sonst zu einer Überpopulation kommen würde. Des weiteren gilt zu bedenken, dass es das Ziel ist eine Antikörper Population zu erhalten, bei der die räumliche Verteilung möglichst ähnlich zu der Antigen Verteilung ist, dies wird so erreicht.

Cluster in aiNet sind Bilder der Eingabedaten bzw. Datencluster. Die Eigenschaften der aiNet-Cluster werden durch den Suppression Threshold  $\sigma_s$  gesteuert. Hiermit wird geregelt ab wann sich Antikörper und Antigene erkennen. Dies bedeutet, dass die Größe des entstehenden Netzwerkes durch den Suppression Threshold direkt beeinflusst wird. Für die Struktur der aiNet-Cluster gilt immer, es gibt weniger Daten (Ag) als Graphenknoten (Ab) und weniger Knoten (Ab) als Untergraphen bzw. Cluster. Untergraphen werden über die Distanz gebildet. Die direkte Folge hiervon ist eine Datenkompression. Allerdings muss beachtet werden, dass die Struktur (siehe Abb. 1(b)) nicht direkt vorliegt, sondern noch extrahiert werden muss, also ein wesentlicher Schritt noch fehlt.

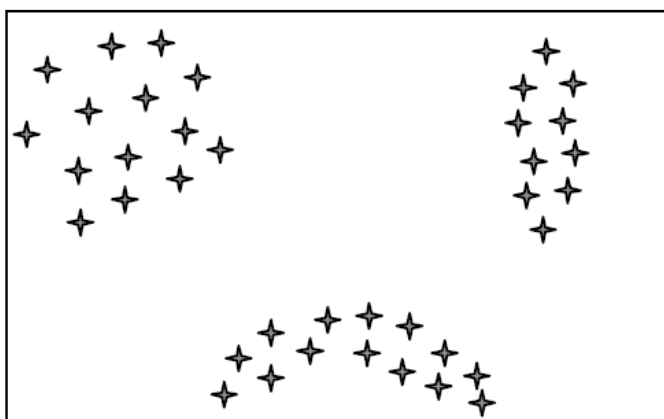
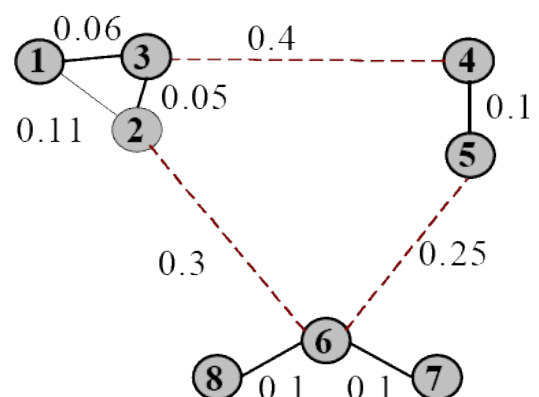


Abbildung 1: (a) Eingabedaten



(b) fertige aiNet Struktur

## 4 Der Lernalgorithmus von aiNet

### 1. Bei jedem Durchlauf / jeder Generation:

1.1 Für jedes Antigen  $Ag_j, j=1..M, (Ag_j \in Ag)$  :

1.1.1 Berechne die Affinität  $f_{ij}$  für alle  $Ab_i, i=1..N$

$$f_{ij} = 1/D_{ij} \text{ mit } D_{ij} = \|Ab_i - Ag_j\| \text{ (Distanz)}$$

1.1.2 Bilde  $Ab_{[n]}$  aus den n Antikörpern mit der höchsten Affinität

1.1.3 Aus  $Ab_{[n]}$  wird C erstellt indem proportional zu der antigenen Affinität  $f_{ij}$  geklont wird ( $N_c = \sum_{i=1}^n \text{round}(N \cdot D_{ij}^{-1})$ ): Je höher die Affinität, desto mehr Clone gibt es jeweils.

1.1.4 Es wird C\* generiert, indem die Clone C gezielt mutiert werden (affinity maturation). Die Mutationsrate  $\alpha_k$  ist invers proportional zu der Antigenen Affinität  $f_{ij}$  : je höher die Affinität desto geringer ist die Mutationsrate.

$$C_k^* = C_k + \alpha_k (Ag_j - C_k) \quad \alpha_k = 1/f_{ij} \quad k=1..N_c \quad i=1..N$$

1.1.5 Berechne die Affinität  $d_{kj} = 1/D_{kj}$  für  $Ag_j$  und alle Elemente von C\*

1.1.6 Bilde  $M_j$  (clonales Gedächtnis) aus den  $\zeta\%$  Antikörpern mit der höchsten Affinität  $d_{kj}$

1.1.7 Apoptose: Eleminiere alle Clone aus  $M_j$  deren Affinität  $D_{kj} > \sigma_d$  ist.

1.1.8 Berechne die Ähnlichkeit  $s_{ik}$  für die Clone in  $M_j$

$$s_{ij} = \|M_{ji} - M_{jk}\|, \forall i, k$$

1.1.9 Clonal Suppression: Eleminiere alle Clone aus  $M_j$  deren  $s_{ik} < \sigma_s$

1.1.10 Füge die verbliebenen Clone  $M_j$  den Antikörpern des aktuellen durchlaufs  $Ab_{[m]}$  hinzu

1.2 Berechne die Ähnlichkeit aller Antikörper  $Ab_{[m]}$

$$s_{ik} = \|Ab_{[m]}^i - Ab_{[m]}^k\| \quad \forall i, k$$

1.3 Netzwerkunterdrückung: Eleminiere alle Antikörper aus  $Ab_{[m]}$  deren  $s_{ik} < \sigma_s$

1.4 Bilde die neue Generation  $Ab \leftarrow [Ab_{[m]}; Ab_{[d]}]$   
(wobei  $Ab_{[d]}$  eine Auswahl aus Ab sein muss)

### 2 Überprüfe das Haltekriterium

Das Haltekriterium kann und wird hier variabel gehalten um eine grösstmögliche Flexibilität zu ermöglichen. Es ist zum Beispiel denkbar nach einer bestimmten Anzahl an Iterationen, oder bei einer bestimmten Anzahl von Antikörpern abubrechen. Auch der mittlere Fehler zwischen Ab-Ag kann ein Haltekriterium sein. Insbesondere ist es sinnvolles zu stoppen, wenn der mittlere Fehler sich nicht mehr verändert oder schwankt.

## **5 Charakterisierungen von aiNet**

AiNet kann auf verschiedene Weisen charakterisiert werden. Dies hängt allein von der Sicht des Autors ab, und welche Eigenschaften er für aiNet als signifikant bezeichnet. Somit ergeben sich Charakterisierungen als Netzwerk und als evolutionärer Algorithmus.

### **5.1 Charakterisierung als Netzwerk**

Aufgrund der Definition von aiNet als kantengewichteter Graph, ist aiNet als verbindungsorientiert zu sehen. Eine weitere Eigenschaft ist, dass aiNet als konkurrierend betrachtet werden kann. Schließlich hängt das Überleben der Antikörper davon ab, dass sie Antigene erkennen und sich somit vermehren können und nicht sterben. Hieraus ergibt sich auch, dass aiNet konstruktiv ist.

Der Zustand des Netzwerks wird durch Konzentration und Affinität der Antikörper und Antigene bestimmt. Die Zustandsänderungen erfolgen durch den Lernalgorithmus.

Da Antikörper zu den Antigenen korrespondieren, ist aiNet sehr anpassungsfähig. Konkret bedeutet dies, dass die entstehende aiNet Struktur durch den initialen Antigen Satz bestimmt wird.

Im Vergleich zu neuronalen Netzen ergibt sich, dass in aiNet die Knoten die Bilder der Daten sind, und nicht die verarbeitenden Elemente. In aiNet stehen die Verbindungen für die Ähnlichkeit bzw. Affinität und in neuronalen Netzwerken wird durch die Verbindungsstärken gelerntes wiedergespiegelt.

### **5.2 Charakterisierung als evolutionärer Algorithmus**

Wenn aiNet als evolutionärer Algorithmus klassifiziert wird, kann als erstes festgehalten werden, dass aiNet Populations basiert ist, und auch eine Anfangspopulation gegeben ist. Es muß eine Evaluationsfunktion geben, hierfür müssen Ähnlichkeitsmessungen und Affinitätsmessungen definiert werden. Ein weiterer wichtiger Aspekt ist, dass in aiNet die Mutation verwendet wird. Durch die Mutation werden Clone zu Nachkommen. Letztlich gilt, wie für jeden genetischen bzw. evolutionären Algorithmus, dass es sehr viele Parameter gibt z.B. maximale Affinität oder Ähnlichkeit, die Anzahl der zu selektierenden Antikörper, der natürliche Tod und die Netzwerkunterdrückung.



## 6 Knowledge Extraction

Die zentrale Frage bei der Wissensextraktion ist, warum diese überhaupt nötig ist. Das Problem ist, dass die Visualisierung des Netzwerks für Antikörper/Antigen Dimensionen größer 3 nicht ohne weiteres möglich ist. Der Lösungsansatz ist hier hierarchisches Clustern. Es kann festgehalten werden, dass die fertige Netzwerkstruktur erst in der Phase der Knowledge Extraction entsteht.

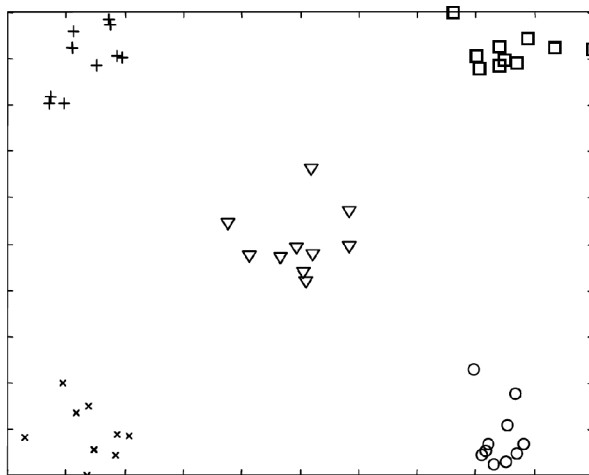
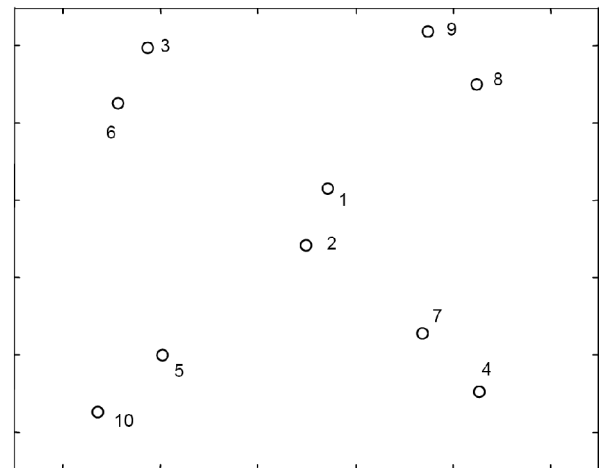


Abbildung 2: (a) zu clusternde Daten



(b) Resultierendes aiNet als Antikörper Netzwerk

Abbildung 2 wird im folgenden als Basis für die Visualisierung der verschiedenen Verfahren dienen.

### 6.1 Möglichkeit 1 der Knowledge Extraction

Die einfachste und intuitive Möglichkeit das Wissen aus aiNet zu extrahieren ist, alle Verbindungen aus dem Konnektivitätsgraph zu entfernen die größer als ein bestimmter Wert (Threshold) sind. Der Vorteil hierbei ist, dass dies schnell und einfach funktioniert. Das größte Problem ist allerdings, dass fehlerhafte Interpretationen hier nicht vermieden werden können. Es müssen also andere Verfahren verwendet werden um sichere Ergebnisse zu erhalten.

### 6.2 Anforderungen an den Knowledge Extraction Algorithmus

Da die intuitive Möglichkeit der Wissensextraktion nicht funktioniert, müssen Anforderungen an den Algorithmus zur Knowledge Extraction gestellt werden. Als erstes soll der Algorithmus eine hohe Aussagewahrscheinlichkeit haben. Dies bedeutet, dass die Ergebnisse die extrahiert werden auch wirklich die Situation im aiNet widerspiegeln und somit weiter verwendet werden können.

Ein weiteres Kriterium ist, dass der Algorithmus die Anzahl der Cluster korrekt wiedergeben kann, da dies ein zentraler Punkt bei der aiNet Entwicklung war. Auch soll die räumliche Verteilung der Cluster durch den Algorithmus wiedergespiegelt werden. Schließlich sollen die Antigene, also die Initialen Daten, den entstandenen Clustern zugeordnet werden können.

### 6.3 Möglichkeit 2 der Knowledge Extraction

Die nächste Möglichkeit der Wissensextraktion ist ein Dendrogramm.

#### Definition Dendrogramm:

Ein Dendrogramm ist ein gewichteter Baum bei dem alle Endknoten die selbe Distanz (Pfadlänge) bis zur Wurzel besitzen.

Konstruiert werden kann das Dendrogramm aus der Ähnlichkeit / Distanz zwischen den Antikörpern. Konkrete Algorithmen lassen sich bei Hartigan (1967) und Hubert, Arabie und Meulman (1998) finden.

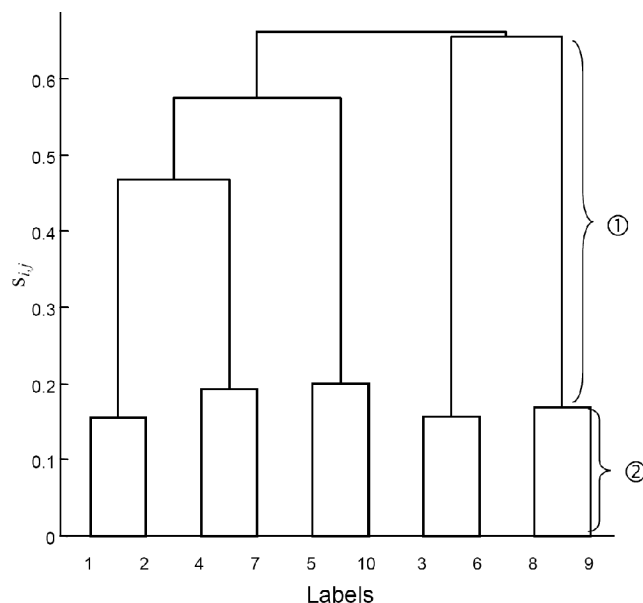


Abbildung 3: aiNet Dendrogramm

Das Dendrogramm in Abbildung 3 wurde aus den Daten bzw. aus den Antikörpernetzwerk aus Abbildung 2 erstellt.

Die Clusterung kann nach der Konstruktion des Dendrogramms über die Höhe erfolgen. Konkrete Algorithmen lassen sich bei Milligan und Cooper (1985) finden.

## 6.4 Möglichkeit 3 der Knowledge Extraction

Eine weitere Möglichkeit die Wissensextraktion durchzuführen ist, einen minimalen Spannbaum zu erstellen.

### Definition Spannbaum

Ein Baum ist ein Spannbaum eines Graphen wenn er ein Untergraph ist, der alle Knoten enthält und schleifenfrei ist.

### Definition minimaler Spannbaum

Ein Minimaler Spannbaum eines Graphen ist ein Spannbaum mit minimalen Gewicht.

### Definition Gewicht eines Baumes

Das Gewicht eines Baumes ist definiert als die Summe der Gewichte der einzelnen Kanten.

Die Konstruktion des minimalen Spannbaums kann beispielsweise über Prim's Algorithmus (Prim, 1957) erfolgen. Allerdings ist mit dem minimalen Spannbaum allein die Wissensextraktion noch nicht abgeschlossen, da die Clusterung noch nicht durchgeführt wurde. Aber der minimale Spannbaum ist hierfür die ideale Basis. Als nächster Schritt wird ein Histogramm des Spannbaums anhand der unterschiedlichen Längen der Kanten erstellt. Die Anzahl der Täler im Histogramm sind gleich der Anzahl der Cluster der Antikörper. Nun werden noch die inkonsistenten Kanten entfernt (in Abbildung 4(a) gestrichelt gezeichnet). Eine Kante ist inkonsistent genau dann, wenn das Gewicht der Kante signifikant grösser, als der Durchschnitt der benachbarten Kanten ist. Nach diesem Schritt liegt die finale aiNet Struktur vor. In Abbildung 4 werden die Ergebnisse, die aus Abbildung 2, bzw den zugrundeliegenden Daten, gewonnen wurden, dargestellt.

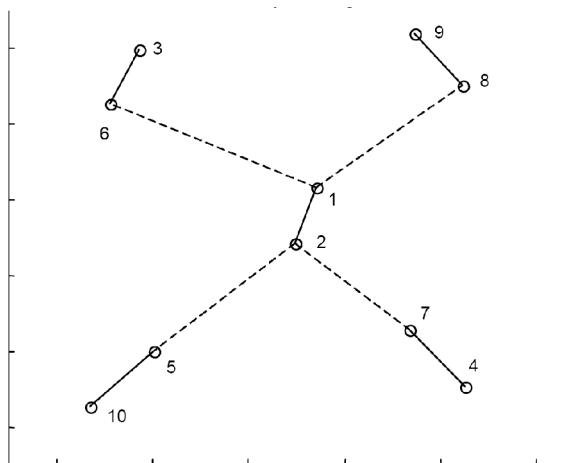
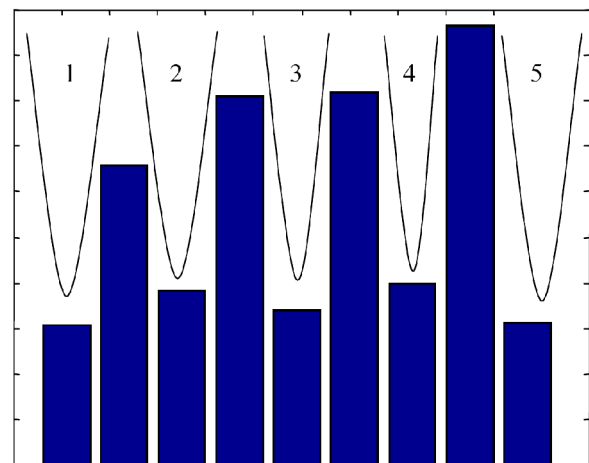


Abbildung 4: (a) Minimaler Spannbaum



(b) Histogramm des Minimalen Spannbaums

## 6.5 Möglichkeit 4 der Knowledge Extraction

Die vierte und letzte Möglichkeit der Wissensextraktion ist das Fuzzy Clustering. Beim Fuzzy Clustering werden die Knoten/Antikörper nicht fest einem Cluster zugeordnet. Es existiert eine membership function die die Zugehörigkeit der Knoten zu den jeweiligen Clustern einen Wert zwischen 0 und 1 zuweist, je höher der Wert, desto höher die Zugehörigkeit.

Konkrete Algorithmen wie fuzzy k-means und fuzzy c-means lassen sich bei Bezdek und Pal (1998) finden.

In Tabelle 1 beschreibt  $c_i$  die generierten Antikörper aus Abbildung 2 (b). Die durch den gewählten Algorithmus berechneten Cluster werden durch  $v_i$  beschrieben.

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$	$c_{10}$
$v_1$	1.00	1.00	0.67	0.71	0.76	0.69	0.84	0.75	0.71	0.66
$v_2$	0.58	0.63	0.50	1.00	0.68	0.50	1.00	0.60	0.56	0.64
$v_3$	0.67	0.50	0.63	0.56	0.50	0.58	0.57	1.00	1.00	0.50
$v_4$	0.50	0.55	0.54	0.62	1.00	0.57	0.64	0.50	0.50	1.00
$v_5$	0.60	0.50	1.00	0.50	0.59	1.00	0.50	0.60	0.56	0.64

Tabelle 1: Ergebnisse des Fuzzy Clustering

## 6.6 Diskussion der Möglichkeiten

Es gibt verschiedene Möglichkeiten um die Informationen die im aiNet stecken zu extrahieren. Es kann pauschal nicht eine Möglichkeit bevorzugt oder vernachlässigt werden. Wichtig ist, dass man für das jeweilige Problem die richtige Methode der Knowledge Extraction wählt. Ein komplizierteres Verfahren ist bei einem einfachen Problem meist nicht sinnvoll zu verwenden, da der overhead einfach zu groß ist. Umgekehrt kann das intuitive Verfahren aus 6.1 schon bei etwas schwereren Aufgaben keine Ergebnisse mehr liefern die verwendet werden können.

## 7 Beispiele

### 7.1 SPIR

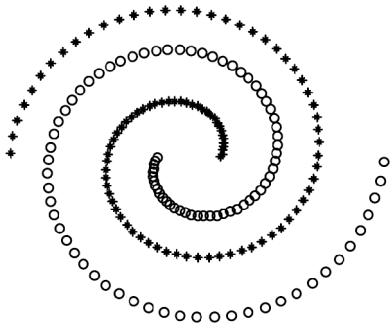


Abbildung 5: 2 Spirals Problem

Beim SPIR Problem geht es darum, dass Daten zweidimensional als sich Ineinander drehende Spiralen gegeben sind. Aufgabe ist es, jeden Datenpunkt bzw. jedes Antigen einem Cluster zuzuordnen. Zu bemerken ist, dass für aiNet jedes Antigen gleich aussieht, die unterschiedliche Darstellung dient hier nur der besseren Visualisierung für den Menschen.

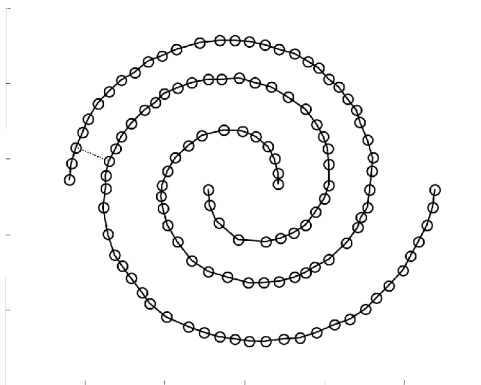


Abbildung 6: Minimaler Spannbaum

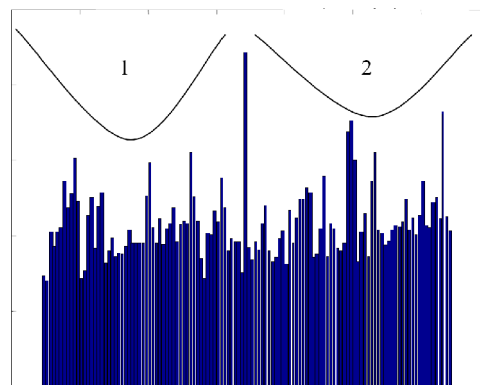


Abbildung 7: Histogramm mit Tälern

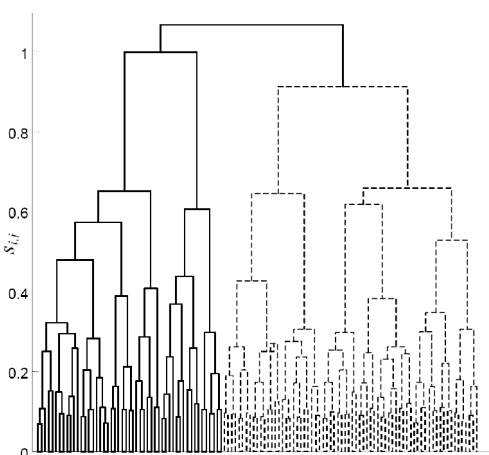


Abbildung 8: Dendrogramm

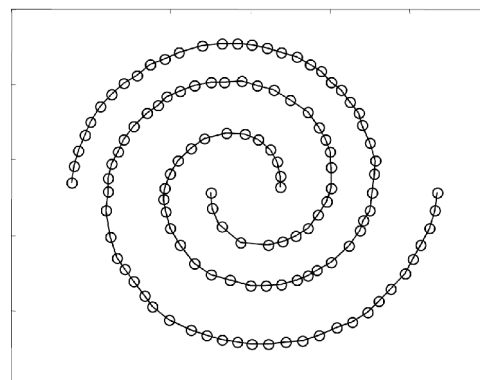


Abbildung 9: Finale aiNet Struktur

## 7.2 CHAINLINK

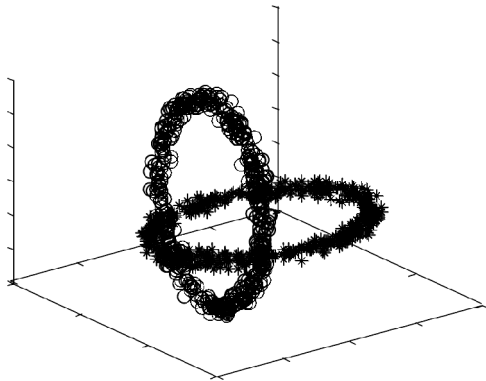


Abbildung 10: Chainlink Problem

Bei CHAINLINK sind zwei Datenringe im Dreidimensionalen ineinander gesteckt. Die Ergebnisse wurden mit Möglichkeit 2 und 3 der Knowledge Extraction erzielt.

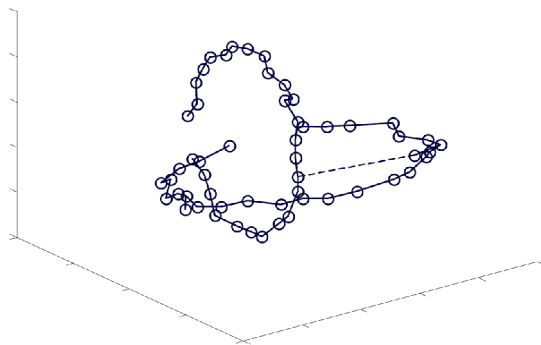


Abbildung 11: Minimaler Spannbaum

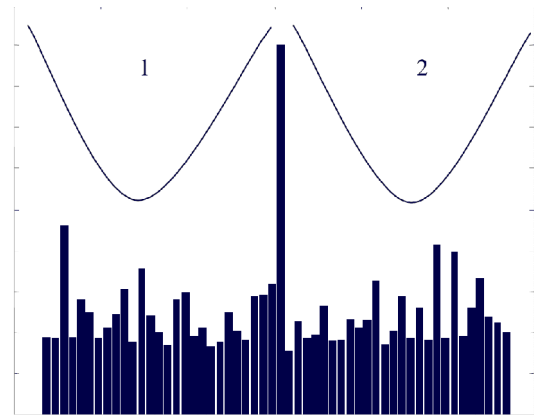


Abbildung 12: Histogramm mit Tälern

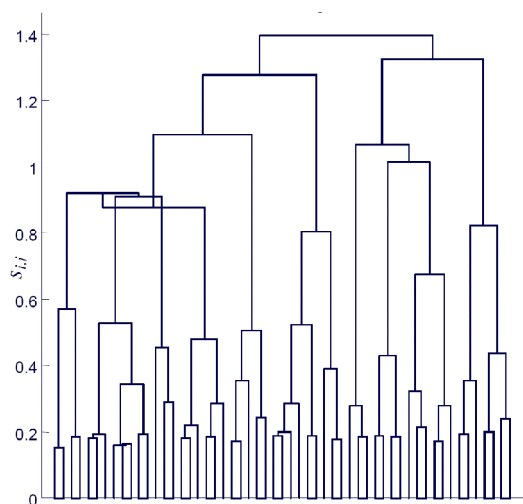


Abbildung 13: Dendrogramm

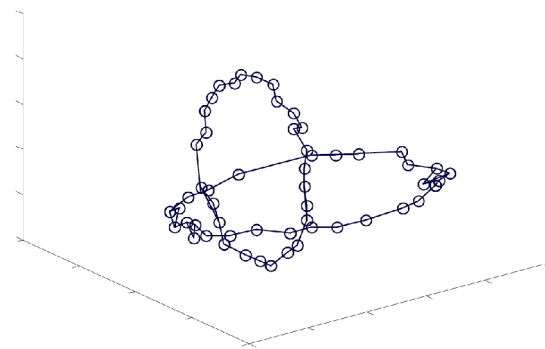


Abbildung 14: Finale aiNet Struktur

### 7.3 5-NLSC

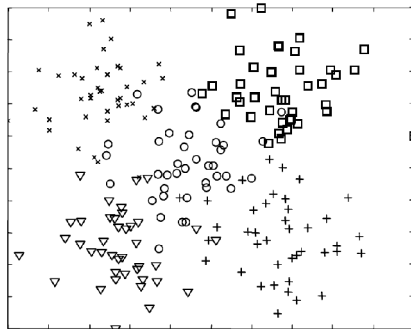


Abbildung 15: 5NLSC Prob.

Bei dem 5-NLSC Problem sind im zweidimensionalen 5 Cluster nah beieinander angeordnet und überlappen teilweise leicht. Auch hier dient die unterschiedliche Darstellung der Antigene/Daten nur der besseren Visualisierung und ist für aiNet nicht sichtbar.

Ergebnisse mit Möglichkeit 2 und Möglichkeit 3 der Knowledge Extraction:

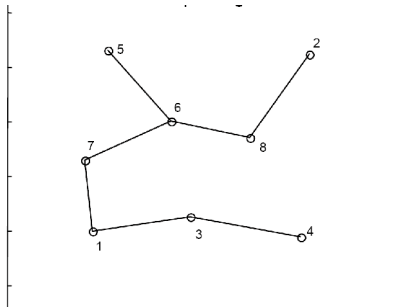


Abbildung 16: Minimaler Spannbaum

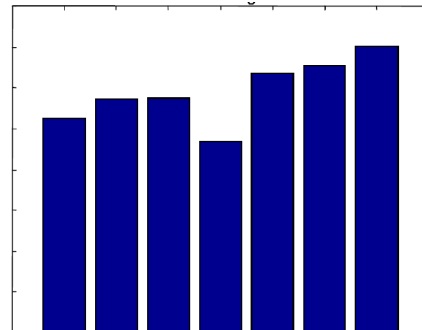


Abbildung 17: Histogramm

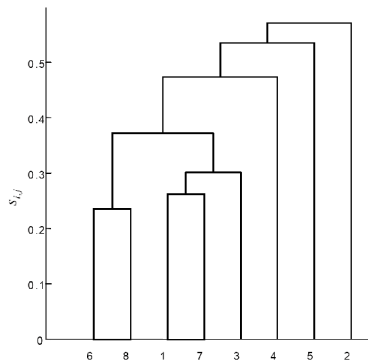


Abbildung 18: Dendrogramm

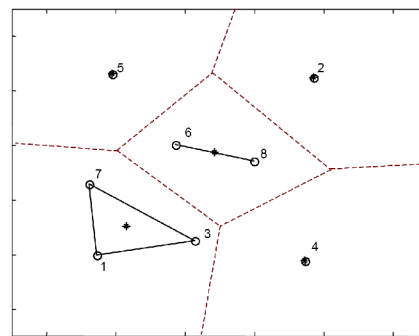


Abbildung 19: Finale aiNet Struktur

Ergebnisse mit Möglichkeit 4: FuzzyNet Clustering:

	$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$
$v_1$	1.00	0.50	1.00	0.50	0.50	0.86	1.00	0.63
$v_2$	0.80	0.50	1.00	0.50	0.50	1.00	0.97	1.00
$v_3$	0.65	0.50	0.55	0.50	1.00	0.89	0.94	0.50
$v_4$	0.71	0.50	1.00	1.00	0.50	0.50	0.59	0.72
$v_5$	0.50	1.00	0.50	0.50	0.50	0.63	0.50	0.80

Tabelle 2: FuzzyClustering Ergebnis

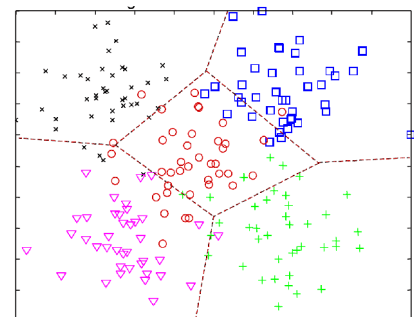


Abbildung 20: Finale aiNet Struktur

## 8 Fazit

Bei aiNet geht es hauptsächlich darum, Daten zu Clustern. Allerdings leistet dies das eigentliche aiNet nicht. Das Einzige was aiNet tut, ist die Eingabedaten zu reduzieren. Auf diese reduzierten Daten muss dann ein konventioneller Algorithmus, die Knowledge Extraction, aufsetzen. Hier wäre wichtig zu wissen, inwieweit aiNet Vorteile für die konventionellen Algorithmen bringt, vor allem gilt zu bedenken das der aiNet Lernalgorithmus die Laufzeit  $O(n^2)$  hat. Leider wird dieser Vergleich an keiner Stelle von den Autoren gezogen. Um zu zeigen wieviel Potential aiNet hat, wird es mit SOM verglichen. Es werden zwar viele Parameter von aiNet angegeben um die Ergebnisse nachvollziehbar erscheinen zu lassen, aber besonders beim 5NLSC Problem wirken die Fähigkeiten von aiNet magisch. Hier stellt sich besonders die Frage ob aiNet überhaupt funktioniert, wenn nicht per Hand günstige Parameter gewählt werden. Fraglich ist, ob bei der SOM Konfiguration genauso „ordentlich“ gearbeitet wurde und ob die Ergebnisse hierdurch nicht hinfällig sind. Aus diesem Grund verweise ich auf die originale Quelle und diskutiere SOM bei den Beispielen nicht.

Ein weiteres Problem bei aiNet - und wohl bei vielen anderen evolutionären Algorithmen oder immunen Netzwerken - ist, dass es sehr problematisch sein kann die Eingabedaten so zu präparieren, dass der angewendete Algorithmus richtig arbeitet. Es ist auf keinen Fall intuitiv Daten in den ShapeSpace zu bringen. Bei einfachen Clusterungsaufgaben von 2 oder 3 dimensionalen Punktmengen ist dies natürlich kein Problem, aber normalerweise formulieren sich Probleme nicht als solche Mengen.

Ein weiteres Problem ist, dass das Ziel von aiNet, sich vom Immunsystem zu lösen, nicht konsequent genug verfolgt worden ist. Sinnvoll wäre sicherlich gewesen sich vom Immunsystem inspirieren zu lassen, sich danach aber komplett davon zu lösen. Es ist für die Lösung von Problemen einfach nicht relevant ob man Daten als Antigene sieht oder einfach als Daten. Am Ende kommt es nur darauf an, was das entwickelte System leistet, und zwar im Vergleich zu konventionellen Systemen und nicht zu solchen, die auch einen biologischen Hintergrund haben.



## 9 Literatur

Leandro Nunes de Castro & Fernando José Von Zuben (2001) *aiNet: An Artificial Immune Network Model for Data Analysis* <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/lnunes/DMHA.pdf>

Bezdek, J. C. & Pal, S. K. (1992). *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*, New York, IEEE.

Prim, R. C. (1957). *Shortest Connection Networks and Some Generalizations*, Bell System Technology Journal, 1389-1401.

Jerne, N. K. (1974a). *Towards a Network Theory of the Immune System*, Ann. Immunol. (Inst. Pasteur) 125C, 373-389.

Hartigan, J. A. (1967). *Representations of Similarity Matrices by Trees*, Journal of the American Statistical Association, 62, 1440-1158.

Milligan, G. W., & Cooper, M. C. (1985). *An Examination of Procedures for Determining the Number of Clusters in a Data Set*, Psychometrika, 50(2). 159-179. 39

Hubert, L., Arabie, P., & Meulman, J. (1998). *Graph-Theoretic representations for Proximity*