Eine Metrik zur Klassifizierung von Neuronen

Diplomarbeit
bei Prof. Dr. G. Wittum

vorgelegt von Holger Heumann
aus Strausberg
am Fachbereich Mathematik der
Universität Heidelberg

Heidelberg, 27. August 2006

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Einleitung

Die Nervenzellen des Gehirns erinnern in ihrer Form stark an die verzweigte Struktur von natürlichen Bäumen. Ähnlich wie Botaniker, die verschiedene Baumarten allein an ihrem Wuchs unterscheiden können, versuchen Neurowissenschaftler, Nervenzellen aufgrund ihrer Form zu klassifizieren. Die Hoffnung ist, dass die so strukturierten Nervenzellen zu einem besseren Verständnis der Funktion des Gehirns führen. Dabei wird davon ausgegangen, dass Form und Funktion der Zellen sich gegenseitig bedingen. Um nun eine solche Einteilung nach morphologischen Aspekten objektiv durchführen zu können, werden geeignete Hilfsmittel zur Analyse der Morphologie, d.h. von Form und Gestalt der Zellen, benötigt. Weiterhin können diese Hilfsmittel bei der Untersuchung von Morphologieveränderungen in verschiedenen Wachstumsphasen, bei der Suche nach Zusammenhängen zwischen Dysfunktionen und Form der Zellen sowie bei der Generierung künstlicher Neuronen zu Simulationszwecken verwendet werden.

Es gibt die verschiedensten Ansätze, die Morphologie von Zellen zu quantifizieren [**?**]. Die am naheliegendsten Ansätze beschreiben Zellen durch morphologische Kennzahlen, wie etwa Volumen, Oberfläche oder Anzahl der Verzweigungspunkte. Betrachten wir jedoch beispielsweise die vielfältigen Formen, die binäre Bäume mit gleicher Anzahl an Verzweigungspunkten annehmen können, so wird deutlich, dass für eine akzeptable morphologische Analyse komplexere Maße benötig werden. Eine Möglichkeit besteht in der Definition von Abstandsfunktionen, die eine Aussage über die Ähnlichkeit von Neuronen machen sollen. Für zwei verschiedene Populationen von Zellen kann damit untersucht werden, ob zwei Zellen aus derselben Population einen kleineren Abstand haben und damit ähnlicher sind als zwei Zellen aus verschiedenen Populationen. Ist dies der Fall, so repräsentieren die beiden Populationen verschiedene morphologische Gruppen. In [**?**] wird beispielsweise eine solche Abstandsfunktion über eine Metrik auf Mengen, die sogenannte Hausdorffmetrik, definiert, indem die Morphologie von Zellen durch Punktenmengen im $\mathbb{R}^3$ approximiert wird.

Wir wollen in dieser Arbeit einen anderen Weg zur Definition von Abstandsfunktionen auf Neuronen gehen. Dazu betrachten wir die Verallgemeinerung der Edit-Distanz [**?**] von Wörtern auf Bäume. Bei der Bestimmung der Edit-Distanz werden Folgen von elementaren Transformationsoperationen betrachtet, die das eine Objekt in ein anderes überführen, und es wird diejenige Folge bestimmt, deren Gewicht minimal ist. Über geeignete Definitionen der Kosten von Transformationsoperationen können verschiedene Aspekte von Ähnlichkeit

modelliert werden. Dieses Konzept kann auf Bäume übertragen werden, und eine kleine Modifikation liefert dann eine berechenbare Abstandsfunktion für Bäume.

Das erste Kapitel beinhaltet eine kurze Einführung zum biologischen Hintergrund dieser Arbeit und soll Einblick in die untersuchten Objekte, die Neuronen, geben. Im zweiten Kapitel stellen wir dann die grundlegenden Ideen von Wagner und Fischer [**?**] zur Edit-Distanz von Wörtern dar und formulieren die Verallgemeinerung auf Bäume. Da die Bestimmung dieser Abstandsfunktion NP-vollständig ist, wird in Kapitel **??** eine Metrik auf Bäumen eingeführt, die die Edit-Distanz leicht modifiziert und effektiv berechenbar ist. Anschließend zeigen wir in Kapitel **??**, wie wir diese Metrik auf die Abstandsbestimmung von Neuronen anwenden können und stellen eine Implementierung vor, die den Abstand zweier Nervenzellen aus ihrer Kodierung im hoc-Format bestimmt. Mit diesem Programm demonstrieren wir dann in Kapitel **??**, dass unsere Abstandsfunktion in der Tat in der Lage ist, Zellen bekannter morphologischer Klassen zu unterscheiden und damit ein geeignetes Instrument der morphologischen Analyse ist. Abschließend werden in Kapitel **??** die Ergebnisse dieser Arbeit zusammengefasst.

Kapitel 1

Biologische Grundlagen

Dieses Kapitel soll lediglich einen ersten Einblick in Funktion und Aufbau von Gehirn und Nervenzellen geben. Kapitel 1, 2 und 5 in [**?**], aus denen die meisten Abbildungen dieses Kapitels stammen, liefern eine umfassendere Übersicht.

1.1 Makroskopische Struktur des Gehirns

Das Gehirn läßt sich anatomisch in mehrere makroskopische Teilstrukturen unterteilen. Im embryonalen Zustand unterscheidet man zwischen Vorderhirn, Mittelhirn (*Mesencephalon*) und Rautenhirn. Im ausgewachsenen Zustand differenziert sich dann das Rautenhirn zur *Medulla ablongata*, *der Pons* und *dem Cerebellum* und das Vorderhirn zu dem Zwischenhirn (*Diencephalon*) und den beiden Großhirnhälften (*cerebrale Hemispähren*). Diese Regionen gliedern zusammen mit den Mittelhirn und dem Rückenmark das Zentralnervensystem in sieben anatomische Regionen (Abb. **??**).

Jedem dieser Bereiche sind spezielle Funktionalitäten zugeordnet. So befindet sich zum Beispiel im Diencephalon der Hypothalamus, der das autonome Nervensystem und die Abgabe von Hormonen reguliert. Die größte Region sind die beiden cerebralen Hemisphären, die aus Großhirnrinde (*Cortex*), den Basalganglien, dem Hippokampus und dem Mandelkern bestehen und für höhere motorische und kognitive Funktionen sowie Gedächtnis und Emotionen zuständig sind. Interessanterweise sind die Grundbausteine des Nervensystems in allen Funktionsbereichen dieselben speziellen Zelltypen, die Neuronen.

1.2 Die Nervenzelle

Es gibt zwei verschiedene Zelltypen im Nervensystem: die Gliazellen oder Stützzellen und die Nervenzellen (Neuronen). Die Neuronen, auf die wir uns hier beschränken wollen, sind verantwortlich für die Informationsweiterleitung und -verarbeitung. Erst die große

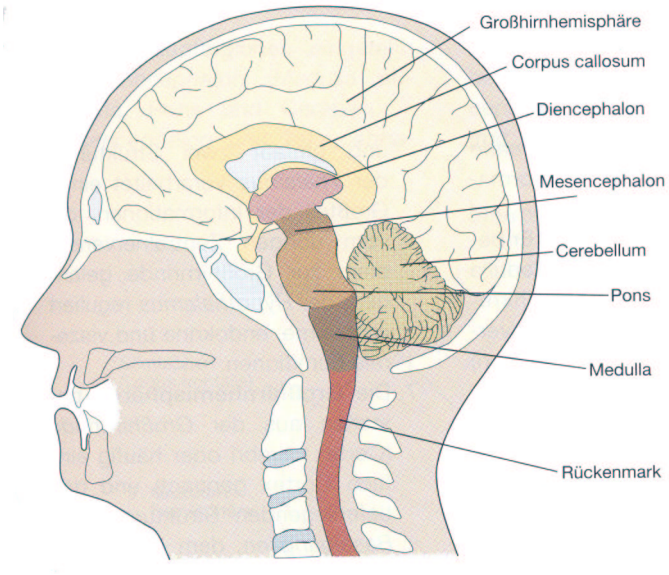


Abbildung 1.1: *Anatomie des Gehirns (aus [?]).*

Konzentration von Neuronen und ihr hoher Verschaltungsgrad, ermöglichen die vielfältige und komplexe Funktionalität des Gehirns. In manchen Bereichen des Gehirns befinden sich mehrere Millionen Zellen pro Kubikzentimeter, und eine Zelle ist teilweise mit tausend und mehr Zellen vernetzt.

1.2.1 Morphologie

Trotz der großen Variabilität, haben die verschiedenen Neuronen eine grundlegende gemeinsame Struktur (Abb. **??**). An einem Zellkörper (*Soma*), der den Zellkern und die Zellorganellen enthält, befinden sich mehrere röhrenartige Fortsätze, die in *Axon* und *Dendriten* unterteilt werden. Die Axone, von denen jede Zelle höchstens eines besitzt, sind sehr dünn im Verhältnis zum Zellkörper und können bis zu einem Meter lang werden. Es ist der Ausgangskanal für Signale. Die teilweise zahlreichen Dendriten (griech: $\tau o\ \delta\epsilon\nu\delta\rho o\nu$ - der Baum) sind stark verzweigt und verjüngen sich schnell. Sie dienen dazu Signale, von anderen Nervenzellen zu empfangen und zum Soma weiterzuleiten. Die Synapse bezeichnet die Stelle, an der Signale von einer axonalen Endigung auf die Dendriten einer anderen Nervenzelle übertragen werden.

1.2.2 Elekrophysiologie

Entscheidend für die elektrische Signalleitung innerhalb einer Nervenzelle ist die Spannungsdifferenz an der Membran, die durch unterschiedliche Ionenkonzentrationen im In-

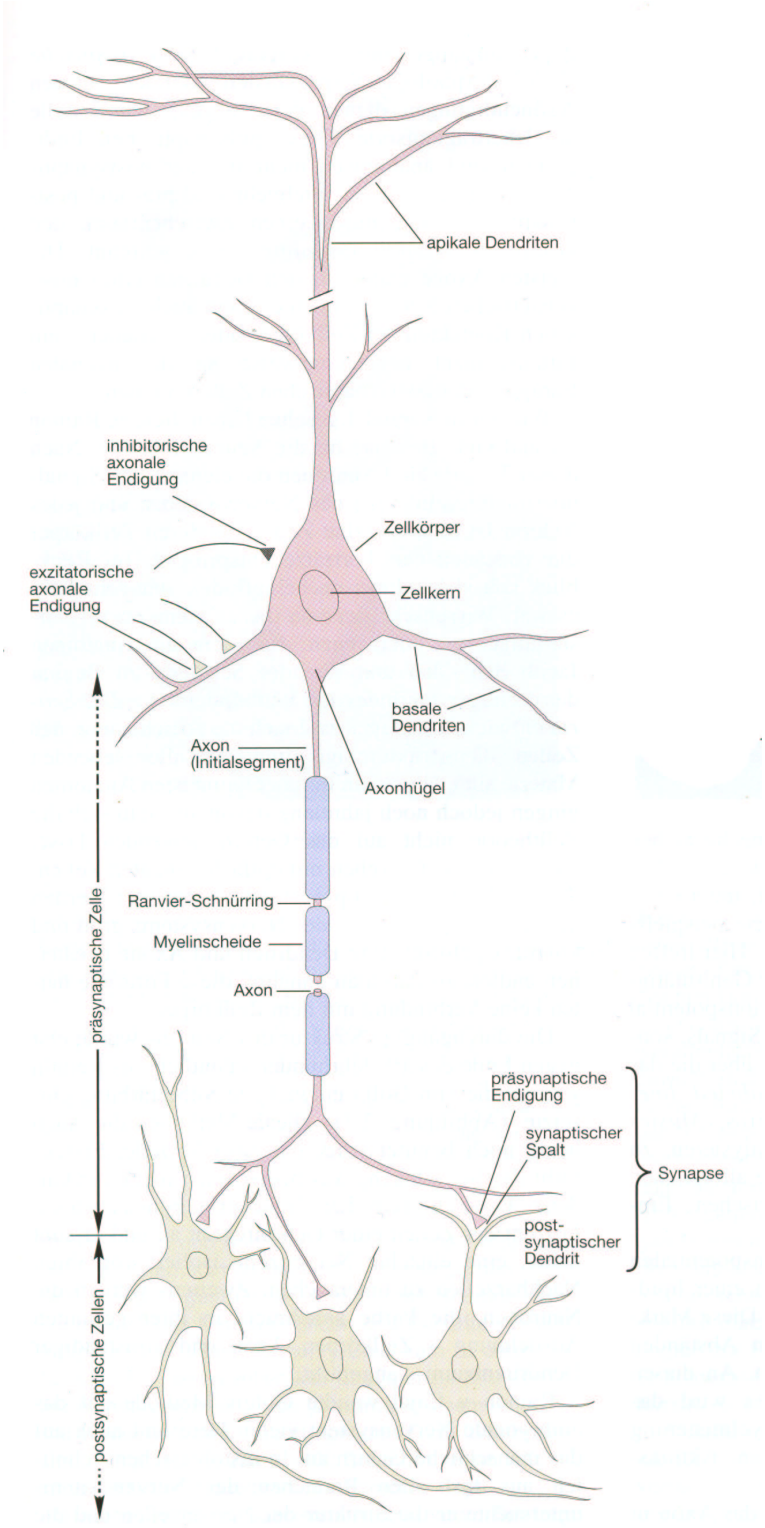


Abbildung 1.2: *Schematischer Aufbau einer Nervenzelle (aus [?]).*

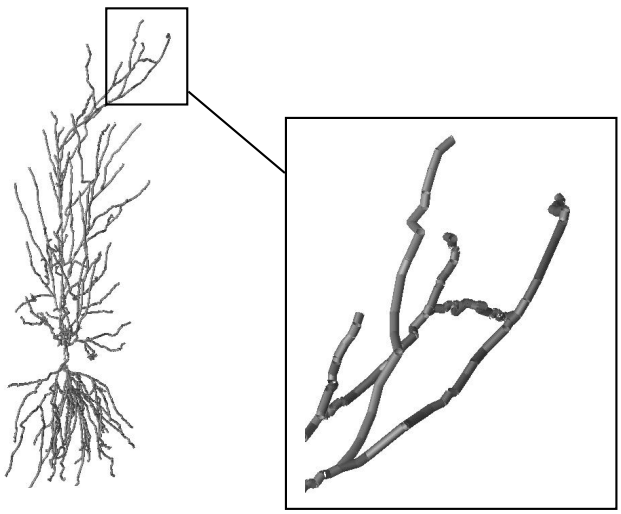


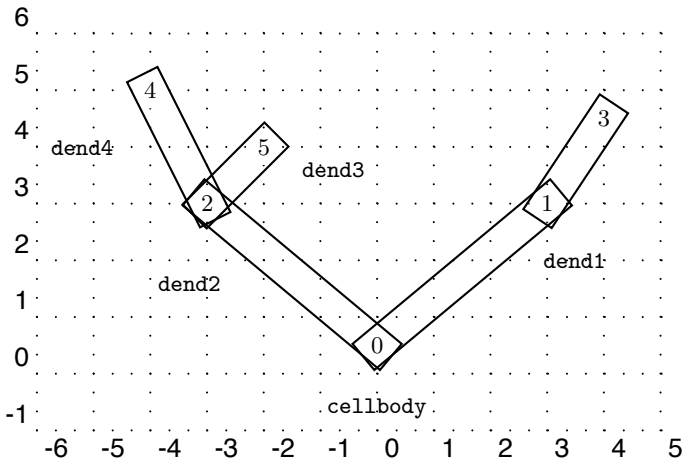
Abbildung 1.3: *Die Zelle n*412 *aus dem Duke-Southampton-Archiv [***?***] und ein vergrößerter Ausschnitt. Zur besseren Darstellung wurde ein konstanter Durchmesser gewählt.*

neren und Äußeren der Zelle verursacht wird. Die in die Membran eingebauten Ionenkanäle und -pumpen ermöglichen durch einen komplizierten Mechanismus die Weiterleitung von Spannungsausschlägen, den *Aktionspotentialen*. Bei einer synaptischen Signalübertragung wird ein Spannungsausschlag im Dendriten der postsynaptischen Zelle erzeugt, der sich dann abgeschwächt entlang des Dendriten in Richtung Soma ausbreitet und eventuell mit Ausschlägen aus anderen Dendritenzweigen überlagert. Die genaue Form der Signalleitung im Dendriten wird von elektrophysiologischen Parametern wie Leitfähigkeit, von geometrischen Parametern wie dem Durchmesser und der Länge und von der topologischen Form, d.h. der Verzweigungsstruktur, bestimmt [**?**].

1.3 Modellierung neuronaler Morphologie

Dank Markierungsmethoden und hochauflösender Mikroskopietechnik ist es heute möglich, 3-dimensionale Bilder einzelner Nervenzellen zu erzeugen. Durch diverse Filter- und Segmentierschritte [**?**] können solche Bilder computerunterstützt aufbereitet werden, so dass mittels eines geeigneten Rekonstruktionsalgorithmus ein mathematischer Graph als Modell für eine Nervenzelle generiert werden kann. Das Modell der Zellmorphologie besteht damit aus einer Menge von Punkten (x, y, z, d), die die Koordinaten und den Durchmesser kodieren, und einer Zuordnungstabelle, die angibt, welcher Punkt mit welchem verbunden ist. Zwischen zwei Punkten (x_1, y_1, z_1, d_1) und (x_2, y_2, z_2, d_2) wird die morphologische Struktur einer Zelle durch Kegelstümpfe approximiert (Abb. **??**). Es gibt diverse Datei-Formate, in denen dieses Modell gespeichert werden kann. In Abb. **??** ist die Kodierung einer einfachen morphologischen Struktur im hoc- und im swc-Format dargestellt.

a)



b)

ID	Typ	X	Y	Z	Radius	Father
0	1	0	0.5	0	0	-1
1	3	3	3	0	0.5	0
2	3	-3	3	0	0.5	0
3	3	4	4.5	0	0.5	1
4	3	-4	5	0	0.5	2
5	3	-2	4	0	0.5	2

c)

```
{create cellbody}
{access cellbody}
{nseg=2}
{pt3dclear()}
{pt3dadd(0, 0.5, 0, 1)}
{pt3dadd(0, 0.5, 0, 1)}
{create dend1}
{connect dend1(0), cellbody(1)}
{access dend1}
{nseg=3}
{pt3dclear()}
{pt3dadd(0, 0.5, 0, 1)}
{pt3dadd(3, 3, 0, 1)}
{pt3dadd(4, 4.5, 0, 1)}
{create dend2}
{connect dend2(0), cellbody(1)}
{access dend2}
{nseg=2}
{pt3dclear()}
{pt3dadd(0, 0.5, 0, 1)}
{pt3dadd(-3, 3, 0, 1)}
{create dend3}
{connect dend3(0), dend2(1)}
{access dend3}
{nseg=2}
{pt3dclear()}
{pt3dadd(-3, 3, 0, 1)}
{pt3dadd(-2, 4, 0, 1)}
{create dend4}
{connect dend4(0), dend2(1)}
{access dend4}
{nseg=2}
{pt3dclear()}
{pt3dadd(-3, 3, 0, 1)}
{pt3dadd(-4, 5, 0, 1)}
```

Abbildung 1.4: *Vergleich von hoc und swc-Format. a) schematische Darstellung einer zweidimensionalen Zellmorphologie. b) swc-Format: Punktbasierte Kodierung, jedem Punkt (X,Y,Z,Radius) der Struktur wird eine eindeutige ID und die ID des Vorgängerpunktes in Richtung Soma zugewiesen. c) hoc-Format: Sektionsbasierte Darstellung, eine Sektion ist die Menge aller Punkte zwischen zwei Verzweigungspunkten. Die Kodierung erfolgt in der Programmiersprache hoc des NEURON-Simulationsprogramms (http://www.neuron.yale.edu/neuron).*

Kapitel 2

Grundlegende Konzepte

Zunächst werden in diesem Kapital kurz die Ideen vorgestellt, die beim Vergleich von Zeichenketten verwendet werden. Dies ist ein Spezialfall für das allgemeinere Problem bei Bäumen, verdeutlicht aber den Ansatz, der später für den Vergleich von Bäumen benutzt wird. Der Aufbau des Kapitels orientiert sich an der Arbeit von Wagner und Fischer ([**?**]).

2.1 Distanzmaß und Metrik

Ein Vergleich von Objekten ergibt nur Sinn, wenn anhand eines allgemeinen Verfahrens der Grad der Ähnlichkeit beliebiger Objekte bestimmt werden kann. Wie diese Ähnlichkeit von Objekten genau zu definieren ist, hängt von der Fragestellung ab. Umgekehrt wird durch ein bestimmtes Verfahren schon im Voraus festgelegt, was als ähnlich oder identisch angesehen wird und was nicht.

In jedem Fall wird jedoch versucht ein Ähnlichkeits- bzw. Distanzmaß für Tupel (i, j) von Objekten i und j aus der Objektmenge $\mathbb{I}$ zu definieren:

Definition 2.1 (Distanzmaß)**.** *Sei* $\mathbb{I} = \{1 \dots n\}$ *eine beliebige Indexmenge. Eine Abbildung* $d : \mathbb{I} \times \mathbb{I} \to \mathbb{R}_0^+$ *heißt* Distanzmaß *, falls sie die Eigenschaften*

1. $d(i, i) = 0$,
2. $d(i, j) = d(j, i)$

für $i, j \in \mathbb{I}$ *besitzt. Die symmetrische* $n \times n - Matrix$ $(d(i, j))_{i \in \mathbb{I}, j \in \mathbb{I}}$ *heißt* Distanzmatrix. *Ein Distanzmaß heißt* metrisch *falls zusätzlich gilt:*

3. $d(i, k) \leq d(i, j) + d(j, k)$ *für* $i, j, k \in \mathbb{I}$.

Die Definition einer Metrik auf einem beliebigen Raum Ω setzt dagegen einen Gleichheitsbegriff, also eine Äquivalenzrelation, auf Ω voraus.

Definition 2.2 (Metrik)**.** *Sei Ω ein beliebiger Raum. Eine Abbildung $\delta : \Omega \times \Omega \longrightarrow \mathbb{R}_0^+$ heißt* Metrik*, falls für $x, y, z \in \Omega$ gilt :*

1. $\delta(x, x) = 0$,
2. $\delta(x, y) = 0 \qquad \Rightarrow \qquad x = y$,
3. $\delta(x, y) = \delta(y, x)$,
4. $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$.

Da insbesondere jede Metrik auch ein metrisches Distanzmaß ist, wird versucht, beliebige Objekte durch mathematische Objekte darzustellen, für die eine Metrik definiert ist. Deshalb werden Objekte häufig durch Merkmalsvektoren $\mathbf{x}^i \in \mathbb{R}^n$ dargestellt und das Distanzmaß d über die l^p-Normen definiert:

$$d(i, j) := d_p(\mathbf{x}^i, \mathbf{x}^j) := \left(\sum_{k=1}^{n} \left| x_k^i - x_k^j \right|^p \right)^{1/p} \quad p \in \mathbb{N},\ \mathbf{x}^i, \mathbf{x}^j \in \mathbb{R}^n.$$

Für Objekte, die als Zeichenketten bzw. Bäume dargestellt werden können, bietet sich ein grundsätzlich anderes Vergleichsverfahren an. Dieses basiert auf dem Prinzip der sogenannten **Edit-Distanz** d_{edit}**:**
Es wird eine Folge von Transformationsoperationen auf Buchstaben bzw. Knoten bestimmt, die das eine Objekt in das andere überführen und minimal in Bezug auf eine zu definierende Kostenfunktion ist.
Über die Darstellungen u_i und u_j zweier Objekte i und j als Baum oder Wort kann dann ein Distanzmaß durch

$$d(i, j) := d_{edit}(u_i, u_j)$$

definiert werden.

2.2 Der Vergleich von Zeichenketten

Der Vergleich von Zeichenketten oder -folgen ist ein häufig auftretendes Problem, sei es beim Vergleich zweier verschiedener Versionen desselben Textdokumentes oder allgemeiner zweier Folgen durch Buchstaben kodierter Objekte. Ein populäres Beispiel aus der Bioinformatik ist der Vergleich von DNA-Sequenzen. In beiden Fällen will man wissen ob sich die beiden Zeichenketten unterscheiden und quantifizieren wie groß der Unterschied ist. Je nach Fragestellung sind sogar die konkreten Unterschiede gesucht. Ein Ansatz zur Lösung solcher Fragestellungen ist die Suche nach einer Zuordnung der Elemente der Zeichenketten, die gewisse Voraussetzungen, wie die Erhaltung der Anordnung, erfüllt und optimal in Bezug auf eine zu definierende Kostenfunktion ist. Dieser Ansatz als Optimierungsproblem birgt natürlich die Gefahr, dass der Raum zulässiger Lösungen und damit auch die Komplexität des Problems sehr groß wird. Mit geeigneten Voraussetzungen an zulässige Zuordnungen und der Formulierung als dynamisches Programm kann man dem entgegenwirken.

2.2.1 Grundbegriffe

Im Folgenden werden einige Standardbezeichnungen eingeführt, die beim Vergleich von Zeichenketten verwendet werden.

- Ein *Alphabet* $(\Sigma, <)$ ist eine endliche Menge Σ zusammen mit einer totalen Ordnung $<$ auf Σ. Die Elemente von Σ heißen *Buchstaben*.
- Eine endliche Folge von Buchstaben bezeichnet man als *Wort* oder *Zeichenfolge*. Das *leere Wort* wird mit λ bezeichnet. Σ^* bezeichnet die Menge aller Wörter über Σ und Σ^n die Menge der Wörter mit Länge n.
- Für ein Wort V bezeichnet $V[i]$ den Buchstaben an der $i-ten$ Stelle.
- Für ein Wort V bezeichnet $|V|$ die Länge von V, d.h. die Anzahl der Buchstaben. Insbesondere gilt $V = V[1]V[2]\dots V[|V|]$ und $|\lambda| = 0$.
- Ein Wort V heißt *Faktor* eines Wortes U, falls $U = U_1VU_2$ für zwei Wörter U_1, U_2.
- Ein Wort V heißt *Präfix* eines Wortes U, falls $U = VU_1$ für ein Wort U_1. Insbesondere bezeichnet $U(i) = U[1]\dots U[i]$ das Präfix von U, das aus den ersten i Buchstaben besteht.
- Ein Wort V heißt *Suffix* eines Wortes U, falls $U = U_1V$ für ein Wort U_1. Mit $U - U(i)$ bezeichnen wir das Suffix $U[i+1]\dots U[|U|]$.

2.2.2 Die Edit-Distanz für Zeichenketten

Hamming-Distanz Die Hamming-Distanz [**?**] wurde ursprünglich für binäre Wörter definiert, kann aber auf Wörter über beliebigen Alphabeten verallgemeinert werden. Sie ist definiert als die Anzahl der Buchstaben, in denen zwei Wörter gleicher Länge nicht übereinstimmen. Das heißt, es werden die Buchstaben gezählt, die in dem ersten Wort ersetzt werden müssen, um es in das zweite Wort zu überführen (Abb. **??**).

Definition 2.3 (Hamming-Distanz)**.** *Seien U, $V \in \Sigma^*$, Σ ein Alphabet und $|U| = |V|$, dann ist die* Hamming-Distanz $d_H(U,V)$ *definiert durch:*

$$d_H(U,V) = |\{\ i \mid 0 < i \leq |U| \text{ und } U[i] \neq V[i]\}|$$

Die Hamming-Distanz ist eine Metrik auf Σ^n [**?**] und definiert damit ein Distanzmaß auf der Indexmenge der Wörter gleicher Länge.

Levenshtein-Distanz Als eine Erweiterung der Hamming-Distanz kann das von Levenshtein [**?**] eingeführte Distanzmaß angesehen werden, bei dem neben den Ersetzungen auch das Einfügen bzw. das Löschen von Buchstaben erlaubt ist.

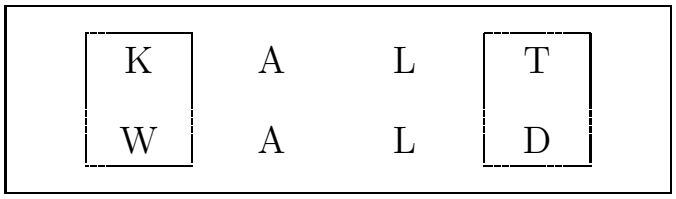


Abbildung 2.1: *Die Hamming-Distanz:* $d_H = 2$.

Definition 2.4 (elementare Edit-Operationen)**.** *Eine* Edit-Operation *oder auch* Transformation *ist ein Tupel von Wörtern* u *und* v *mit maximaler Länge eins*

$$u, v \in \Sigma \cup \{\lambda\}.$$

Unter Verwendung der Schreibweise $u \longrightarrow v$ *für die Transformation* (u, v) *bezeichnet* $U \Rightarrow V$ *via* $u \longrightarrow v$ *die* Herleitung *von* V *aus* U*, falls* $U = \sigma u \tau$ *und* $V = \sigma v \tau$ *mit* $\sigma, \tau \in \Sigma^*$*. Eine Folge von Wörtern* $U_0, U_1, \ldots U_m \in \Sigma^*$ *mit* $U_0 = U$ *und* $U_m = V$ *heißt* S-Herleitung von V aus U*, falls es ein Folge von Edit-Operationen* $S = (s_1, \ldots s_m)$ *gibt, so dass* $U_{i-1} \Rightarrow U_i$ *via* s_i *für* $1 \leq i \leq m$ *. Insbesondere bezeichnen wir eine Edit-Operation* $u \longrightarrow v$ *als*

1. *Substitution (sub), falls* $u \neq \lambda$ *und* $v \neq \lambda$*,*
2. *Einfügeoperation (ins), falls* $u = \lambda$ *und* $v \neq \lambda$*,*
3. *Löschoperation (del), falls* $v = \lambda$ *und* $u \neq \lambda$*.*

und nennen $\mathcal{E}$ *die Menge aller Edit-Operationen.*

Abbildung **??** veranschaulicht die Wirkung der Edit-Operationen. Es ist zu beachten, dass insbesondere jedes Wort U aus U selbst mit der leeren Folge $S = ()$ S-herleitbar ist und dass zwei verschiedene Wörter $V_1 \neq V_2$ mit der gleichen Folge S aus U S-herleitbar sein können (Abb. **??**).

Levenshtein konnte zeigen [**?**], dass die minimale Anzahl solcher elementaren Edit-Operationen, die ein Wort U in ein Wort V überführen, eine Metrik ist und damit als Distanzmaß verwendet werden kann.

Definition 2.5 (Levenshtein-Distanz)**.** *Die* Levenshtein-Distanz $d_L(U, V)$ *für zwei Wörter* U*,* $V \in \Sigma^*$ *ist definiert als:*

$$d_L(U, V) = min_{S \in \mathcal{S}}\{|S|\}.$$

$\mathcal{S}$ *ist dabei die Menge aller möglichen Folgen* $S = (s_1 \ldots s_n)$*,* $s_i \in \mathcal{E}$*, von Edit-Operationen, die eine S-Herleitung von* V *aus* U *induzieren.*

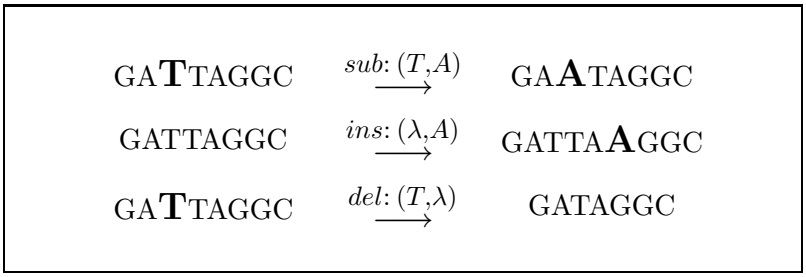


Abbildung 2.2: *Beispiele für Edit-Operationen aus Definition* **??**.

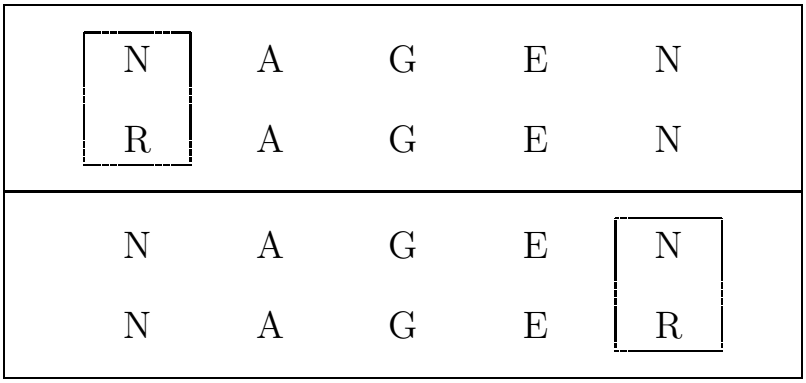


Abbildung 2.3: Die Wörter *RAGEN* und *NAGER* sind durch Substitution (n, r) aus *NAGEN* herleitbar.

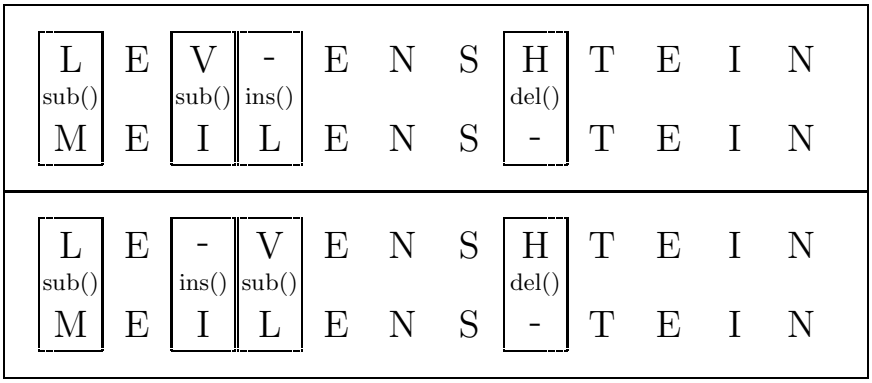


Abbildung 2.4: *Levenshtein-Distanz: Zwei verschiedene Operationsfolgen für* $d_L = 4$ *(http://www.levenshtein.de).*

Im nächsten Abschnitt wollen wir dies in einem allgemeineren Fall beweisen. Im Gegensatz zur Hamming-Distanz ist die Levenshtein-Distanz für Wörter beliebiger Länge definiert. Allerdings ist dann die Folge der Edit-Operationen nicht mehr eindeutig (Abb. **??**).

Weitere Verallgemeinerungen des *Edit-Distanz-Prinzips* basieren auf der Hinzunahme von anderen elementaren Edit-Operationen, wie etwa der Vertauschung von Buchstaben oder der unterschiedlichen Gewichtung der einzelnen elementaren Operationen. Wir werden uns im Folgenden auf die Einführung einer lokalen Kostenfunktion beschränken.

Allgemeine Edit-Distanz Beim Vergleich von DNA-Sequenzen, Zeichenketten über dem Alphabet $\Sigma = \{C, A, G, U, T\}$, können Substitutionen als Modell für Mutationen angesehen werden. Da nicht jede Mutation gleich wahrscheinlich ist, legt man für jede Substitution gewisse Gewichte, auch Kosten genannt, fest und versucht nun, die wahrscheinlichste Folge von Transformations-Operationen zu bestimmen, die eine Zeichenkette in die andere überführen, d.h. wir suchen eine S-Herleitung mit minimalen Kosten. Auch formal definieren wir zuerst eine *lokale Kostenfunktion* für die Edit-Operationen und erweitern diese dann auf beliebige Folgen.

Definition 2.6 (lokale Kostenfunktion)**.** *Eine Funktion $\gamma : \mathcal{E} \to \mathbb{R}_0^+$ heißt* lokale Kostenfunktion *auf der Menge der Edit-Operationen $\mathcal{E}$, falls γ aufgefasst als Funktion von u und v:*

$$d_\gamma(u, v) = \gamma(u \longrightarrow v)$$

die Axiome einer Metrik auf $\Sigma \cup \{\lambda\}$ erfüllt. Der Definitionsbereich von γ wird auf Folgen von Edit-Operationen erweitert, indem

$$\gamma(S) = \sum_{i=1}^{m} \gamma(s_i)$$

für eine Folge $S = (s_1, \ldots s_m)$, $s_i \in \mathcal{E}$ und

$$\gamma(S) = 0$$

für die leere Folge $S = ()$ definiert wird.

Die allgemeine Edit-Distanz zweier Wörter wird nun über diejenige S-Herleitung definiert, deren Gewicht minimal ist:

Definition 2.7 (Edit-Distanz)**.** *Seien U, $V \in \Sigma^*$ zwei Wörter. Mit den Bezeichnungen aus* **??** *ist die allgemeine* Edit-Distanz $d_{edit}(U, V)$ *definiert als:*

$$d_{edit}(U, V) = min_{S \in \mathcal{S}}\{\gamma(S)\}$$

Offensichtlich erfüllt die Edit-Distanz die ersten drei Bedingungen einer Metrik (Definition **??**). Wagner und Fischer ([**?**]) konnten zeigen, dass sie sogar die Dreiecksungleichung erfüllt und damit eine Metrik auf dem Raum der Wörter Σ^* ist.

Satz 2.8 (Wagner und Fischer 1974)**.** *Die in* **??** *definierte Edit-Distanz ist eine Metrik auf* Σ^* *und*

$$d(i,j) := d_{edit}(U_i, U_j)$$

ein Distanzmaß auf der Indexmenge von Σ^**.*

Mit Hilfe der im nächsten Abschnitt definierten Spurabbildung wird Satz **??** dann bewiesen.

Die Levenshtein-Distanz d_L ist der Spezialfall einer Edit-Distanz mit lokaler Kostenfunktion

$$\gamma_L(u,v) = \begin{cases} 0 & falls \quad u = v \\ 1 & sonst \end{cases}$$

und damit ebenfalls eine Metrik auf Σ^*.

2.2.3 Die Edit-Distanz und die Spur

Jede Folge von Edit-Operationen S, die ein Wort U in ein Wort V überführt, induziert eine Zuordnung zwischen den Buchstaben von U und denen von V. Diese Zuordnung wird als Alignment oder auch als Spur der Edit-Distanz bezeichnet.

Definition 2.9 (Spur)**.** *Sei* U*,* $V \in \Sigma^*$ *und* M *eine Menge von Tupeln* $(i,j) \in \mathbb{N} \times \mathbb{N}$*. Dann ist das Tripel* (M,U,V) *eine Spur von* U *und* V*, falls für die Tupel* $(i,j) \in M$ *gilt:*

1. $1 \leq i \leq |U|$ *und* $1 \leq j \leq |V|$*,*
2. *für Tupel* (i_1,j_1) *und* (i_2,j_2) *aus* M *gilt:* $i_1 = i_2 \iff j_1 = j_2$*,*
3. *für Tupel* (i_1,j_1) *und* (i_2,j_2) *aus* M *gilt:* $i_1 \leq i_2 \iff j_1 \leq j_2$*.*

Anstatt (M,U,V) *verwenden wir auch nur* M*, wenn klar ist, um welche Wörter es sich handelt. Die Verknüpfung*

$$M_1 \circ M_2 := \{\ (i,j) \mid \exists k \text{ mit } (i,k) \in M_1 \text{ und } (k,j) \in M_2\}$$

zweier Spuren M_1 *und* M_2 *ist ebenfalls eine Spur.*

Anschaulich definiert die Spur M eine bijektive Abbildung von einem Teil der Buchstaben von U nach einem Teil der Buchstaben von V. Das Tupel (i,j) steht für die Zuordnung von $U[i]$ zu $V[j]$. Die dritte Bedingung aus Definition **??** stellt sicher, dass die Anordnung der Buchstaben erhalten bleibt. Das Beispiel zur Levenshtein-Distanz (Abb. **??**) veranschaulicht eine solche Spur.

Unter Verwendung der in **??** eingeführten lokalen Kostenfunktion $\gamma(s)$, $s \in \mathcal{E}$, wird nun eine Kostenfunktion für die Spur definiert.

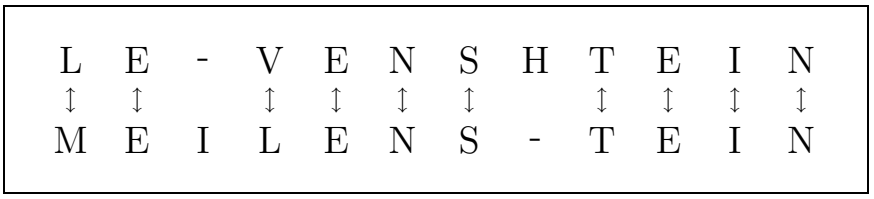


Abbildung 2.5: *Veranschaulichung der Spur:* $M = \{(1,1),(2,2),(3,4),(4,5),(5,6),(6,7),(8,8),(9,9),(10,10),(11,11)\}$.

Definition 2.10 (Gewicht der Spur)**.** *Seien U und V zwei Wörter und $\mathcal{M}$ die Menge aller Spuren (M,U,V). Die Funktion $\Gamma : \mathcal{M} \to \mathbb{R}_0^+$ mit*

$$\begin{aligned}\Gamma(M) := & \sum_{(i,j)\in M} \gamma(U[i] \to V[j]) \\ & + \sum_{\{i|\forall V[j]\in V,(i,j)\notin M\}} \gamma(U[i] \to \lambda) \\ & + \sum_{\{j|\forall U[i]\in U,(i,j)\notin M\}} \gamma(\lambda \to V[j])\end{aligned}$$

wird als Gewicht der Spur *M bezeichnet.*

Die Elemente aus M tragen zum Gewicht als Substitutionen bei. Buchstaben $U[i]$, die von M nicht getroffen werden, werden als Löschoperation gewichtet, die Buchstaben aus V als Einfügeoperation.

Da $\gamma(s)$ der Dreiecksungleichung genügt, ist das Gewicht der Spur subadditiv.

Lemma 2.11 (Subadditivität)**.** *Seien U, V und W Wörter und M_1 und M_2 Spuren von U nach V bzw. von V nach W. Damit ist $M_1 \circ M_2$ eine Spur von U nach W und für das Gewicht $\Gamma(M_1 \circ M_2)$ gilt:*

$$\Gamma(M_1 \circ M_2) \leq \Gamma(M_1) + \Gamma(M_2)$$

Weiterhin können wir über die Dreiecksungleichung von $\gamma(s)$ eine obere Schranke für das Gewicht einer Spur finden.

Bemerkung 2.12. *Sei M eine Spur von zwei Wörtern U und V. Dann wird $\Gamma(M)$ majorisiert durch das Gewicht der leeren Spur $\emptyset$:*

$$\Gamma(M) \leq 0 + \sum_{\{i|U[i]\in U\}} \gamma(U[i] \to \lambda) + \sum_{\{j|V[j]\in V\}} \gamma(\lambda \to V[j])$$

Den Zusammenhang zwischen der Kostenfunktion der Spur (M,U,V) und der einer S-Herleitung von V aus U stellt die folgende Proposition dar.

Proposition 2.13. *Seien U, V zwei Wörter. Dann gilt:*

1. *Zu jeder Spur M existiert eine S-Herleitung, so dass $\gamma(S) = \Gamma(M)$.*
2. *Umgekehrt existiert zu jeder S-Herleitung eine Spur M, so dass $\Gamma(M) \leq \gamma(S)$.*

Beweis. 1. Sei M eine beliebige Spur von U und V. Jedes Tupel $(i, j) \in M$ stellt eine Substitutionsoperation $U[i] \longrightarrow V[j]$ dar. Buchstaben $U[i]$ von U, für die für alle $V[j] \in V$ $(i, j) \notin M$ gilt, werden gelöscht. Umgekehrt werden Buchstaben $V[j]$, für die für alle $U[i] \in U$ $(i, j) \notin M$ gilt, in U eingefügt, um U in V zu transformieren. Für die so definierte S-Herleitung gilt $\Gamma(M) = \gamma(S)$.

2. Wir beweisen über Induktion nach der Länge m der Folge $S = (s_1, \ldots s_m)$, dass zu jeder S-Herleitung von $V = U_m$ aus $U = U_0$ eine Spur M mit $\Gamma(M) \leq \gamma(S)$ gefunden werden kann. Für $m = 0$ ist durch $M = \{(i, i)|1 \leq i \leq |U|\}$ eine Spur definiert mit $\Gamma(M) = 0 = \gamma(S)$. Im Induktionsschritt nehmen wir an, dass eine Spur M_1 für U_0 und U_{m-1} mit $\Gamma(M_1) \leq \gamma(s_1, \ldots s_{m-1})$ gegeben ist. Da $U_{m-1} \Rightarrow U_m$ via $s_m = u \longrightarrow v$, existieren $\sigma, \tau \in \Sigma^*$, so dass $U_{m-1} = \sigma u \tau$ und $U_m = \sigma v \tau$. Wir unterscheiden nun drei Fälle und definieren jeweils eine Spur M_2 mit $\Gamma(M_2) = \gamma(u \longrightarrow v)$:

   (a) s_m ist Substitution, d.h. $u \neq \lambda$ und $v \neq \lambda$:

   $$M_2 = \{\ (i, i) \mid 1 \leq i \leq |U_m|\}$$

   (b) s_m ist Löschoperation, d.h. $u \neq \lambda$ und $v = \lambda$:

   $$M_2 = \{\ (i, i) \mid 1 \leq i \leq |\sigma|\} \cup \{\ (i + 1, i) \mid |\sigma| < i \leq |U_m|\}$$

   (c) s_m ist Einfügeoperation, d.h. $u = \lambda$ und $v \neq \lambda$:

   $$M_2 = \{\ (i, i) \mid 1 \leq i \leq |\sigma|\} \cup \{\ (i, i + 1) \mid |\sigma| < i \leq |U_{m-1}|\}$$

   Die Verknüpfung $M = M_1 \circ M_2$ ist eine Spur von U und V und für $\Gamma(M)$ gilt unter Verwendung der Subadditivität aus Lemma **??**:

   $$\Gamma(M) \leq \Gamma(M_1) + \Gamma(M_2) \leq \gamma(s_1, \ldots s_{m-1}) + \gamma(s_m) = \gamma(S)$$

□

Als direkte Folgerung dieser Proposition kann die Edit-Distanz d_{edit} damit auch als das Minimum über alle Spuren definiert werden.

Satz 2.14. *Es gilt $d_{edit}(U, V) = min\{\ \Gamma(M) \mid M$ Spur von U und $V\}$*

Beweis. Sei S_{min} diejenige Folge, so dass

$$\gamma(S_{min}) = d_{edit}(U,V) = min_{S\in\mathcal{S}}\{\gamma(S)\}.$$

Nach der Proposition existiert ein Spur M mit $\Gamma(M) \leq \gamma(S_{min})$. Die Annahme $\Gamma(M) < \gamma(S_{min})$ liefert ebenfalls nach der Proposition einen Widerspruch zur Minimalität von $\gamma(S_{min})$. □

Damit ist ebenfalls gezeigt, dass die Edit-Distanz der Dreiecksungleichung genügt.

Korollar 2.15. *Seien U, V und W Wörter, dann gilt:*

$$d_{edit}(U,W) \leq d_{edit}(U,V) + d_{edit}(V,W).$$

Beweis. Seien M_1 und M_2 diejenigen Spuren mit $\Gamma(M_1) = d_{edit}(U,V)$ bzw. $\Gamma(M_2) = d_{edit}(V,W)$. Dann gilt

$$d_{edit}(U,W) \leq \Gamma(M_1 \circ M_2) \leq \Gamma(M_1) + \Gamma(M_2) = d_{edit}(U,V) + d_{edit}(V,W).$$

□

Wir wissen jetzt also, dass d_{edit} eine Metrik auf der Menge der Wörter Σ^* ist. Im nächsten Abschnitt wollen wir mit Hilfe der Spur einen Algorithmus zur Berechnung von d_{edit} entwickeln.

2.2.4 Algorithmus

Der Algorithmus zur Berechnung der Edit-Distanz zweier Wörter U und V basiert auf der Idee, dass die zugehörige Spur (M,U,V) in zwei Spuren $(M_1, U(i), V(j))$ und $(M_2, U - U(i), V - V(j))$ von Präfixen und Suffixen von U und V zerlegt werden kann. Das Gewicht von M ist dann die Summe der Gewichte von M_1 und M_2:

$$\Gamma(M) = \Gamma(M_1) + \Gamma(M_2).$$

Das folgende Theorem bildet die Grundlage für eine dynamische Berechnung der Edit-Distanz.

Satz 2.16. *Wir bezeichnen die Edit-Distanz d_{edit} zweier Präfixe $U(i)$ und $V(j)$ von beliebigen Wörtern U bzw. V mit $D(i,j)$:*

$$D(i,j) = d_{edit}(U(i), V(j)) \qquad \text{mit} \qquad 0 \leq i \leq |U|,\ 0 \leq j \leq |V|.$$

Es gilt dann für alle i, j, $1 \leq i \leq |U|$, $1 \leq j \leq |V|$:

$$D(i,j) = min \begin{cases} D(i-1, j-1) + \gamma(U[i] \longrightarrow V[j]), \\ D(i-1, j) + \gamma(U[i] \longrightarrow \lambda), \\ D(i, j-1) + \gamma(\lambda \longrightarrow V[j]). \end{cases}$$

Beweis. Sei M eine Spur von $U(i)$ und $V(j)$ mit $\Gamma(M) = D(i,j)$. Wir unterscheiden drei Fälle.

1. Es gibt kein Tupel (i,l), $1 \leq l \leq |V|$ in M. Damit ist
$$\Gamma(M) = D(i-1,j) + \gamma(U[i] \longrightarrow \lambda).$$
2. $(i,j) \in M$, d.h. $U[i]$ wird auf $V[j]$ abgebildet und
$$\Gamma(M) = D(i-1,j-1) + \gamma(U[i] \longrightarrow V[j]).$$
3. Es gibt ein $l \neq j$, so dass $(i,l) \in M$. Dann gibt es aber kein Tupel (k,j), $1 \leq k \leq |U|$ in M. Damit ist
$$\Gamma(M) = D(i,j-1) + \gamma(\lambda \longrightarrow V[j]).$$

□

Satz 2.17. *Mit den Bezeichnungen aus Satz* **??** *gilt*

1. $D(0,0) = 0$
2. $D(i,0) = \sum_{k=1}^{i} \gamma(U[k] \longrightarrow \lambda) \qquad 1 \leq i \leq |U|$
3. $D(0,j) = \sum_{l=1}^{j} \gamma(\lambda \longrightarrow V[l]) \qquad 1 \leq j \leq |V|$

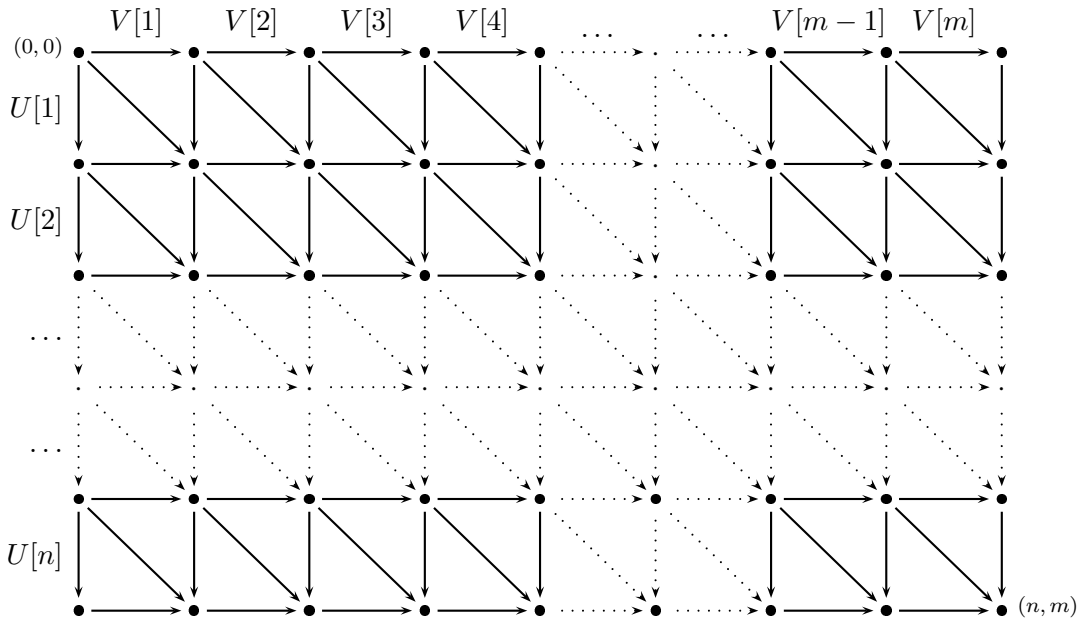
Beweis. Falls $i = 0$ oder $j = 0$, so ist $M = \emptyset$ die einzig mögliche Spur. Der Satz folgt damit unmittelbar aus der Definition von Γ (Definition **??**). □

Edit-Distanz und das Shortest-Path-Problem

Eine anschauliche Erklärung der Rekursionsformel aus Satz **??** liefert die Interpretation als Shortest-Path-Problem. Dazu betrachten wir den gitterähnlichen kantengewichteten Graphen aus Abbildung **??**. Jeder zusammenhängende Weg zwischen den beiden Knoten $(0,0)$ und (m,n) ist äquivalent zu einer Spur von U und V. Die diagonalen Kanten der Wege bestimmen die Zuordnung der Buchstaben während vertikale und horizontale festlegen, welche Buchstaben nicht berücksichtigt werden. Die Summe der Kantengewichte, die Weglänge, ist dann das Gewicht der Spur und der kürzeste Weg entspricht einer Spur mit minimalem Gewicht. Es ist dann klar, dass die Länge $D(i,j)$ des kürzesten Weges von $(0,0)$ nach (i,j) das Minimum von $D(i-1,j-1) + \gamma(U[i],V[j])$, $D(i-1,j) + \gamma(U[i],\lambda)$ und $D(i,j-1) + \gamma(\lambda,V[j])$ ist.

Wir sind nun in der Lage einen vollständigen Algorithmus zur Berechnung der Edit-Distanz zweier Wörter U und V zu formulieren.

Der Algorithmus in Abbildung **??** berechnet aus der Eingabe zweier Wörter U und V die Distanz d_{edit} dynamisch, d.h. er bestimmt zuerst $D(i,j)$ für $0 \leq i < |U|$ und $0 \leq j < |V|$ und errechnet daraus $D(|U|,|V|)$. Die Komplexität wird durch die Länge der Wörter bestimmt und ist von der Ordnung $O(|U||V|)$. Nachdem wir die Matrix $D(i,j)$ berechnet haben, kann nun mit einem einfachen Backtracking-Algorithmus (Abb. **??**) die Spur M bestimmt werden.



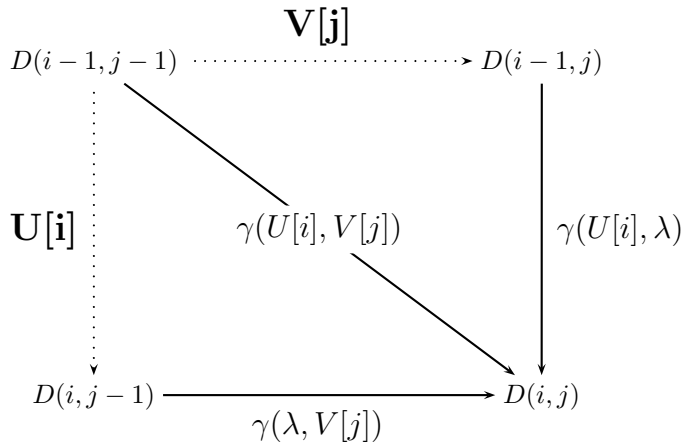


Abbildung 2.6: *Edit-Distanz und das Shortest-Path-Problem.*

Require: $U, V \in \Sigma^*$
1: $D(0,0) \leftarrow 0$
2: **for** $i = 1$ to $|U|$ **do**
3: $\quad D(i,0) \leftarrow D(i-1,0) + \gamma(U[i] \longrightarrow \lambda)$
4: **end for**
5: **for** $j = 1$ to $|V|$ **do**
6: $\quad D(0,j) \leftarrow D(0,j-1) + \gamma(\lambda \longrightarrow V[j])$
7: **end for**
8: **for** $i = 1$ to $|U|$ **do**
9: $\quad$ **for** $j = 1$ to $|V|$ **do**
10: $\quad\quad m1 \leftarrow D(i-1,j-1) + \gamma(U[i] \longrightarrow V[j])$
11: $\quad\quad m2 \leftarrow D(i-1,j) + \gamma(U[i] \longrightarrow \lambda)$
12: $\quad\quad m1 \leftarrow D(i,j-1) + \gamma(\lambda \longrightarrow V[j])$
13: $\quad\quad D(i,j) \leftarrow min(m1, m2, m3)$
14: $\quad$ **end for**
15: **end for**

Abbildung 2.7: *Algorithmus zur Berechnung von* $d_{edit}(U, V)$

Require: $D(i,j)$ $0 \leq i \leq |U|$, $0 \leq j \leq |V|$
1: $i \leftarrow |U|$
2: $j \leftarrow |V|$
3: **while** $i \neq 0$ && $j \neq 0$ **do**
4: $\quad$ **if** $D(i,j) = D(i-1,j) + \gamma(U[i] \longrightarrow \lambda)$ **then**
5: $\quad\quad i \leftarrow i - 1$
6: $\quad$ **else if** $D(i,j) = D(i,j-1) + \gamma(\lambda \longrightarrow V(j))$ **then**
7: $\quad\quad j \leftarrow j - 1$
8: $\quad$ **else**
9: $\quad\quad$ **print** (i,j)
10: $\quad\quad i \leftarrow i - 1$
11: $\quad\quad j \leftarrow j - 1$
12: $\quad$ **end if**
13: **end while**

Abbildung 2.8: *Backtracking-Algorithmus*

2.3 Der Vergleich von Bäumen

Beim Vergleich von Bäumen muss zwischen geordneten Bäumen, Bäumen, in denen die Kinderknoten einer Ordnungsrelation unterliegen und den allgemeineren Bäumen, hier als ungeordnete bezeichnet, unterschieden werden. Während man für erstere analog zu der Edit-Distanz von Wörtern einen Algorithmus findet [**?**, **?**], konnte Zhang [**?**] zeigen, dass das allgemeine Problem NP-vollständig ist. Da die in dieser Arbeit betrachteten Bäume, die Modelle der Zellmorphologie, keine ausgewiesene Ordnung innerhalb der Kinderknoten haben, beschränken wir uns auf den allgemeinen Fall. Im restlichen Teil dieses Kapitels werden die grundlegenden Begriffe und Notationen aus der Graphentheorie eingeführt und die Edit-Distanz und die damit einhergehenden Konzepte auf Bäume verallgemeinert. Im nächsten Kapitel wird eine Abwandlung des Spurbegriffs vorgestellt, die es dann ermöglicht einen Algorithmus für eine obere Schranke der Edit-Distanz zu entwickeln.

2.3.1 Grundbegriffe

Zunächst wollen wir die grundlegenden Begriffe und Notationen aus der Graphentheorie einführen. Wir werden uns dabei auf gerichtete Graphen beschränken.

- Ein *gerichteter Graph* $G = (V, E)$ besteht aus einer Knotenmenge V (engl. vertex) und einer Kantenmenge E (engl. edge). Eine Kante $e = (a, b) \in E$ ist ein geordnetes Paar von Knoten. a ist *Anfangs-* und b *Endpunkt*. Im Folgenden soll mit Graph immer ein gerichteter Graph gemeint sein.

- Ein *knotengewichteter Graph* oder auch *etikettierter Graph* $(V, E, label)$ ist ein Graph (V, E) zusammen mit einer Gewichtsfunktion $label : V \longrightarrow \Sigma$. Σ bezeichnen wir als *Attributmenge*.

- Ein *kantengewichteter Graph* ist ein Graph (V, E) zusammen mit einer Gewichtsfunktion $cost : E \longrightarrow \mathbb{R}$.

- Eine Folge von Knoten $W = v_0, \ldots v_m$ heißt (v_0, v_m)-*Kette*, falls es für zwei Folgenglieder v_i und v_{i+1}, $0 \leq i < m$ eine Kante $e_i \in E$ mit $e_i = (v_i, v_{i+1})$ gibt. m ist die *Länge* der Kette.

- Eine (v_0, v_m)-Kette, in der alle Knoten v_i voneinander verschieden sind, heißt (v_0, v_m)-*Weg*.

- Zwei Knoten u und v heißen *zusammmenhängend*, falls es einen (u, v)-Weg gibt.

- Ein gerichteter Graph G heißt *zusammmenhängend*, falls alle Paare (u, v) von Knoten aus G zusammenhängend sind.

- Eine (u, v)-Kette heißt *geschlossen*, falls ihre Länge nicht Null ist und $u = v$ gilt.

- Eine geschlossene (v, v)-Kette $W = (v, v_1, v_2 \dots v)$, in der alle inneren Knoten v_i von v verschieden sind, heißt *Kreis*.

- Ein Graph, der keinen Kreis enthält, heißt *azyklisch*.

- Ein *Wald* F ist ein azyklischer Graph, in dem jeder Knoten höchstens eine einlaufende Kante hat.

- Ein *Baum* T (engl. tree) ist ein zusammenhängender Wald. Es gibt einen ausgezeichneten Knoten *root*, der mit jedem anderen Knoten v aus T über einen $(root, v)$-Weg zusammenhängend ist. Der ausgezeichnete Knoten *root* wird als *Wurzel* des Baumes bezeichnet. Der *leere Baum* $T = (\emptyset, \emptyset)$ wird mit Θ bezeichnet. $|T|$ bezeichnet die Anzahl der Knoten von T[1].

- Eine Menge $F = \{T_1, \dots, T_m\}$ von Bäumen T_i ist wieder ein Wald.

- Für zwei Knoten u und v eines Baumes heißt u *Vorgänger* von v und v *Nachfolger* von u, falls u auf dem $(root, v)$-Weg liegt. Die *Vorgänger-Nachfolger-Relation* ist eine partielle Ordnung auf der Menge der Knoten. Sie wird mit $\leq_T$ bezeichnet. Sind zwei Knoten u und v direkt verbunden, d.h. es gibt einen (u, v)-Weg der Länge 1, und es gilt $u \leq_T v$, so heißt u *Vater* von v, $father(v)$, und v *Kind* von u, $child(u)$. Die Menge aller Kinder von u wird mit $children(u)$ bezeichnet. Zwei verschiedene Knoten v_1 und v_2 mit $father(v_1) = father(v_2)$ heißen *Geschwister*.

- Der letzte gemeinsame Vorgänger $lca(u, v)$ zweier Knoten u und v ist der Vorgänger von u und v, so dass $x \leq_T lca(u, v)$ für alle gemeinsamen Vorgänger x mit $x \leq_T u$ und $x \leq_T v$.

- Ein *geordneter Baum* ist ein Baum mit einer Ordnung $\leq_G$ auf der Menge $children(v)$, d.h. die Kinderknoten jedes Knotens unterliegen einer Reihenfolge. $\leq_G$ und $\leq_T$ bilden eine totale Ordnung auf der Knotenmenge V.

- Für einen Baum T und einen Knoten v von T bezeichnet $T - v$ den Baum, der entsteht, wenn die Kantenmenge so modifiziert wird, dass der Vater von v der Vater der Kinder von v wird und v und alle Kanten, die v enthalten, gelöscht werden (Abb. **??**).

- Ein *Baum über* Σ ist ein knotengewichteter Baum mit Attributmenge Σ.

- Es ist üblich, die Knoten eines gegebenen Baumes zu nummerieren. Wir wollen im Folgenden eine beliebige Nummerierung der Knoten eines Baumes T als gegeben annehmen.

- Der i-te Knoten von T wird dann mit $t[i]$ bezeichnet.

[1] Um zwischen gerichteten Graphen und ungerichteten Graphen zu unterscheiden, werden auch die Begriffe Branching statt Wald und Aboreszenz statt Baum verwendet.

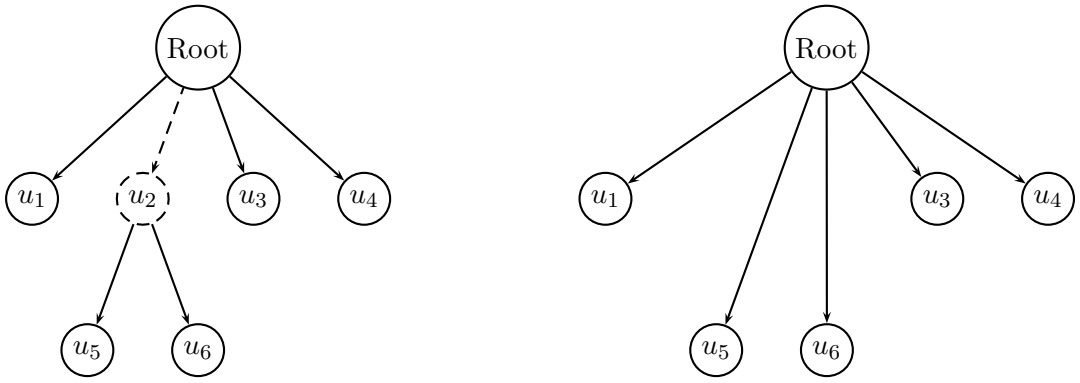


Abbildung 2.9: *Die Bäume T und $T - u_2$*

- Ein *Teilbaum* von T ist ein Baum, dessen Wurzel ein Knoten v_{root} von T ist und der alle Knoten v mit dazugehörenden Kanten aus T, für die $v_{root} \leq_T v$ gilt, enthält.
- Der Teilbaum von T, der als Wurzel den Knoten $t[i]$ hat, wird mit $T[i]$ bezeichnet.
- Der Wald, der entsteht, wenn von einem Teilbaum $T[i]$ die Wurzel $t[i]$ und die dazugehörenden Kanten gelöscht werden, wird mit $F[i]$ bezeichnet.

2.3.2 Die Edit-Distanz für knotengewichtete Bäume

Wir wollen wieder Edit-Operationen bestimmen, die diesmal einen gegebenen Baum T_1 in einen anderen Baum T_2 transformieren. Formal können wir die Menge der Edit-Operationen $\mathcal{E}$ analog zu **??** definieren. Lediglich die Lösch- und Einfügeoperation müssen detaillierter erklärt werden (Abb. **??**).

Definition 2.18 (elementare Edit-Operationen auf Bäumen)**.** *Seien Σ eine Menge von Attributen, λ ein beliebiges Symbol mit $\lambda \notin \Sigma$ und $\Sigma_\lambda = \Sigma \cup \{\lambda\}$. Eine* Edit-Operation *oder auch* Transformation *eines Baumes T über einer Attributmenge Σ ist ein Tupel (u, v) mit*

$$u, v \in \Sigma_\lambda.$$

Die Edit-Operation $u \longrightarrow v$ wird bezeichnet als

1. Substitution (sub)*, falls $u \neq \lambda$ und $v \neq \lambda$,*
2. Einfügeoperation (ins)*, falls $u = \lambda$ und $v \neq \lambda$,*
3. Löschoperation (del)*, falls $v = \lambda$ und $u \neq \lambda$.*

$\mathcal{E}$ ist die Menge aller Edit-Operationen. Ein Baum T_2 ist durch Substitution (u, v) aus T_1 herleitbar, falls es Knoten $t_1[i] \in T_1$ und $t_2[j] \in T_2$ gibt, mit label$(t_1[i]) = u$ und

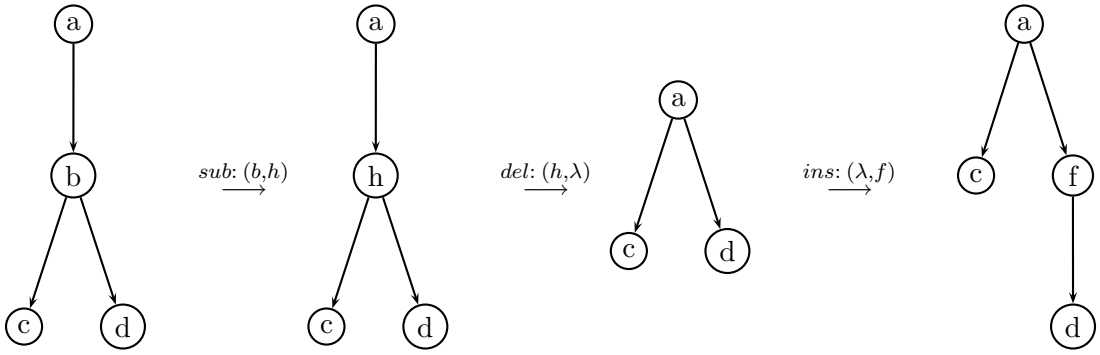


Abbildung 2.10: *Beispiel einer Folge von Edit-Operationen*

$label(t_2[j]) = v$ und $T_1 - t_1[i] = T_2 - t_2[j]$. T_2 ist herleitbar aus T_1 durch eine Einfügeoperation (λ, v), falls es einen Knoten $t_2[j] \in T_2$ gibt, mit $label(t_2[j]) = v$ und $T_1 = T_2 - t_2[j]$. T_2 ist herleitbar aus T_1 durch eine Löschoperation (u, λ), falls es einen Knoten $t_1[i] \in T_1$ mit $label(t_1[i]) = u$ und $T_1 - t_1[i] = T_2$ gibt. Unter Verwendung der Schreibweise $u \longrightarrow v$ für die Transformation (u, v) bezeichnet $T_1 \Rightarrow T_2$ via $u \longrightarrow v$ die Herleitung *von T_2 aus T_1. Eine Folge von Bäumen $T_0, T_1, \ldots T_m$ mit $T_0 = T$ und $T_m = T'$ heißt* S-Herleitung von T' aus T*, falls es ein Folge von Edit-Operationen $S = (s_1, \ldots s_m)$ gibt, so dass $T_{i-1} \Rightarrow T_i$ via s_i für $1 \leq i \leq m$.*

Wir betrachten nun eine lokale Kostenfunktion $\gamma(u, v)$ für die Elemente $(u, v) \in \mathcal{E}$, die den Axiomen einer Metrik auf Σ_λ genüge. γ wird wieder auf Folgen $S = (s_1, \ldots s_m)$ von Edit-Operationen verallgemeinert (vgl. Definition **??**)

$$\gamma(S) = \sum_{i=1}^{m} \gamma(s_i),$$

und die Edit-Distanz zweier Bäume T_1 und T_2 sind dann wiederum die minimalen Kosten einer S-Herleitung von T_2 aus T_1:

$$d_{edit}(T_1, T_2) = min_{S \in \mathcal{S}}\{\Gamma(S)\}$$

Definition 2.19. *(Edit-Distanz für Bäume) Seien T_1 und T_2 zwei Bäume über Σ und Σ_λ wie in **??** definiert. Ferner sei $\gamma : \mathcal{E} \longrightarrow \mathbb{R}_0^+$ eine Metrik auf Σ_λ. Die* Edit-Distanz *von T_1 und T_2 ist dann*

$$d_{edit}(T_1, T_2) = min_{S \in \mathcal{S}}\{\gamma(S)\}.$$

Bei der Edit-Distanz zwischen Wörtern war die Einführung des Spurbegriffs entscheidend für den Nachweis der Gültgkeit der Dreiecksungleichung. Ähnlich verfahren wir jetzt bei Bäumen.

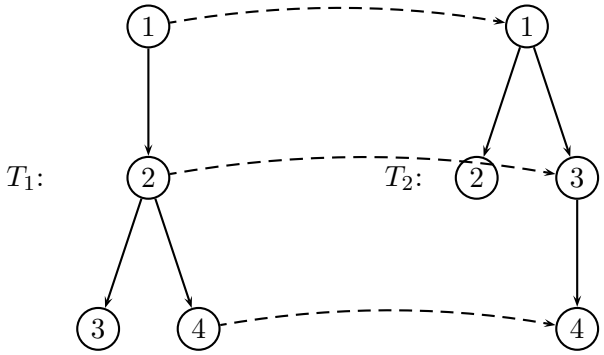


Abbildung 2.11: $M = \{(1,1),(2,3),(4,4)\}$ *ist eine Spur von* T_1 *und* T_2. $M \cup \{(3,2)\}$ *ist keine Spur, da* $t_1[2] \leq_T t_1[3]$ *aber* $t_2[3] \not\leq t_2[2]$.

Definition 2.20 (Spur)**.** *Sei* M *eine Menge von geordneten Tupeln* $(i,j) \in \mathbb{N} \times \mathbb{N}$*. Das Tripel* (M, T_1, T_2) *heißt* Spur *von* T_1 *und* T_2*, falls:*

1. $1 \leq i \leq |T_1|$, $1 \leq j \leq |T_2|$ *für alle* $(i,j) \in M$,
2. *für Tupel* (i_1,j_1) *und* (i_2,j_2) *aus* M *gilt:* $i_1 = i_2 \iff j_1 = j_2$,
3. *für Tupel* (i_1,j_1) *und* (i_2,j_2) *aus* M *gilt:* $t_1[i_1]$ *ist Vorgänger von* $t_1[i_2]$ *genau dann, wenn* $t_2[j_1]$ *Vorgänger von* $t_2(j_2)$ *ist:*

$$t_1[i_1] \leq_T t_1[i_2] \Leftrightarrow t_2[j_1] \leq_T t_2[j_2].$$

Die zweite Bedingung stellt sicher, dass jedem Knoten höchstens ein Knoten des anderen Baumes zugeordnet wird. Die dritte Bedingung sorgt dafür, dass das Vorgänger-Nachfolger-Verhältnis erhalten bleibt (Abb. **??**). Wir definieren nun wie in **??** das Gewicht der Spur M

$$\begin{aligned}\Gamma(M) := \quad & \sum_{(i,j)\in M} \gamma\big(label(t_1[i]) \to label(t_2[j])\big) \\ + \quad & \sum_{\{i|\forall t_2[j],(i,j)\notin M\}} \gamma\big(label(t_1[i]) \to \lambda\big) \\ + \quad & \sum_{\{j|\forall t_2[i],(i,j)\notin M\}} \gamma\big(\lambda \to label(t_2[j])\big).\end{aligned}$$

Da die Verknüpfung

$$M_1 \circ M_2 = \{\ (i,k) \mid (i,j) \in M_1, (j,k) \in M_2\}$$

zweier Spuren wieder eine Spur ist und Γ subadditiv ist

$$\Gamma(M_1 \circ M_2) \leq \Gamma(M_1) + \Gamma(M_2),$$

kann Proposition **??** jetzt für das Verhältnis von Spur und S-Herleitung bei Bäumen formuliert und bewiesen werden.

Proposition 2.21. *Seien T_1 und T_2 zwei Bäume. Dann gilt:*

1. *Zu jeder Spur M existiert eine S-Herleitung, so dass $\gamma(S) = \Gamma(M)$.*
2. *Umgekehrt existiert zu jeder S-Herleitung eine Spur M, so dass $\Gamma(M) \leq \gamma(S)$.*

Beweis. Der Beweis zu Proposition **??** kann auf diese Proposition übertragen werden. Dass es zu jeder Spur M eine S-Herleitung mit $\gamma(S) = \Gamma(M)$ gibt, folgt direkt aus der Definition von Γ. Der zweite Teil der Behauptung wird wieder über Induktion nach der Länge der Folge S und unter Verwendung der Subadditivität des Gewichtes von Spuren gezeigt. □

Als Konsequenz kann die Edit-Distanz auch als das Minimum über die Menge der Spuren definiert werden

$$d_{edit}(T_1, T_2) = min\{\ \Gamma(M) \mid M \textit{ Spur von } T_1 \textit{ und } T_2\}$$

Die Edit-Distanz d_{edit} ist damit eine Metrik auf dem Raum der Bäume über Σ.

Dieses Resultat war der Ausgangspunkt der Berechnung der Edit-Distanz von Wörtern. Leider hilft dies im Falle allgemeiner Bäume nicht weiter. Zhang konnte in [**?**] zeigen, dass die Berechnung der Edit-Distanz in der Tat NP-vollständig ist.

Satz 2.22 (Zhang [**?**])**.** *Seien T_1 und T_2 zwei ungeordnete Bäume. Die Berechnung von $d_{edit}(T_1, T_2)$ ist NP-vollständig.*

Dies bedeutet, dass es keinen Algorithmus mit polynomialer Laufzeit gibt, der dieses Optimierungsproblem löst.

Durch die Einführung einer weiteren Restriktion für zulässige Spuren zwischen zwei Bäumen kann jedoch eine obere Schranke der Edit-Distanz berechnet werden, die ebenfalls den Axiomen einer Metrik genügt. Im nächsten Kapitel wird diese modifizierte Edit-Distanz nach Zhang vorgestellt.

Kapitel 3

Die Edit-Distanz für Bäume

Im letzten Kapitel wurde die Edit-Distanz für Bäume aus der Edit-Distanz für Wörter hergeleitet. In diesem Kapitel wird der von Zhang [**?**] entwickelte Algorithmus vorgestellt und analysiert. Er baut auf dem intuitiven Ähnlichkeitskonzept der Edit-Distanz auf, die im allgemeinen Fall nicht in polynomialer Zeit berechenbar ist.

Die Idee der Modifikation von Zhang beruht darauf, dass die Knoten eines Teilbaums des einen Baumes nur den Knoten eines Teilbaumes des anderen Baumes zugeordnet werden dürfen. Dieser Ansatz geht auf Überlegungen von Tanaka und Tanaka [**?**] zurück, die zeigen konnten, das eine solche Zuordnung bei Klassifikationsprobleme geeigneter ist als die allgemeine Edit-Distanz.

3.1 Die eingeschränkte Edit-Distanz d^c_{edit}

Die Spur aus **??** erhält lediglich die partielle Ordnung $\leq_T$. Die Modifikation, die wir jetzt einführen wollen, sorgt dafür, dass die Ordnung zwischen dem letzten gemeinsamen Vorgänger zweier Knoten und einem dritten Knoten erhalten bleibt.

Definition 3.1 (eingeschränkte Spur)**.** *Seien T_1 und T_2 Bäume über einer Attributmenge Σ und M eine Menge von Tupeln $(i,j) \in \mathbb{N}\times\mathbb{N}$. Das Tripel (M, T_1, T_2) heißt* eingeschränkte Spur*, falls die folgenden Bedingungen erfüllt sind:*

1. *M ist eine Spur nach Definition **??**.*
2. *Für drei beliebige Elemente (i_1, j_1), (i_2, j_2) und (i_3, j_3) aus M seien*

   $$t_1[I] = lca(t_1[i_1], t_1[i_2]) \text{ und } t_2[J] = lca(t_2[j_1], t_2[j_2])$$

   die letzten gemeinsamen Vorgänger von $t_1[i_1]$ und $t_1[i_2]$ bzw. von $t_2[j_1]$ und $t_2[j_2]$. Dann ist $t_1[I]$ Vorgänger von $t_1[i_3]$ dann und nur dann, falls $t_2[J]$ Vorgänger von $t_2[j_3]$ ist, d.h.

   $$t_1[I] \leq_T t_1[i_3] \Leftrightarrow t_2[J] \leq_T t_2[j_3].$$

Abbildung **??** veranschaulicht den Unterschied zwischen Spur und eingeschränkter Spur.

Inwiefern ist diese Definition konform mit der Forderung, dass nur solche Spuren zulässig sein sollen, die Teilbäume wieder auf Teilbäume abbilden? Nehmen wir einmal an, dass $t_1[I]$ Nachfolger von $t_1[i_3]$ ist:

$$t_1[i_3] \leq_T t_1[I].$$

Damit ist $t_1[i_3]$ Vorgänger von $t_1[i_1]$ und $t_1[i_2]$:

$$t_1[i_3] \leq_T t_1[i_1] \qquad \text{und} \qquad t_1[i_3] \leq_T t_1[i_2],$$

und die erste Bedingung aus Definition **??** impliziert dann

$$t_2[j_3] \leq_T t_2[j_1] \qquad \text{und} \qquad t_2[j_3] \leq_T t_2[j_2].$$

Damit ist $t_2[j_3]$ auch Vorgänger des letzten gemeinsamen Vorgängers von $t_2[j_1]$ und $t_3[j_2]$:

$$t_2[j_3] \leq_T t_2[J].$$

Analog schließen wir mit vertauschten Rollen und erhalten

$$t_1[i_3] \leq_T t_1[I] \;\Leftrightarrow\; t_2[j_3] \leq_T t_2[J].$$

Kombinieren wir dies mit der zweiten Bedingung aus Definition **??**, so erhalten wir die folgende Eigenschaft einer eingeschränkten Spur: *Betrachtet man drei verschiedene Knoten eines Baumes, so erhält die eingeschränkte Spur das Vorgänger-Nachfolger-Verhältnis eines Knotens zu dem letzten gemeinsamen Vielfachen der beiden anderen Knoten.*

Betrachten wir nun die eingeschränkte Spur M zweier Bäume T_1 und T_2. Seien $T_1[i_1]$ und $T_1[i_2]$ zwei disjunkte Teilbäume, d.h $t_1[i_1] \not\leq_T t_1[i_2]$ und $t_1[i_2] \not\leq_T t_1[i_1]$. Die beiden Mengen

$$M_1 = \{\, m \mid (m,n) \in M \text{ und } t_1[m] \in T_1[i_1]\}$$

und

$$M_2 = \{\, m \mid (m,n) \in M \text{ und } t_2[m] \in T_1[i_2]\}$$

enthalten die Indizes von Knoten von $T_1[i_1]$ bzw. $T_1[i_2]$, die unter M einen Bildknoten in T_2 haben. Komplementär dazu sind die Mengen

$$\overline{M}_1 = \{\, n \mid (m,n) \in M \text{ und } t_1[m] \in T_1[i_1]\}$$

und

$$\overline{M}_2 = \{\, n \mid (m,n) \in M \text{ und } t_2[m] \in T_1[i_2]\},$$

die Indizes von Knoten aus T_2, die unter M Urbilder in den Teilbäumen $T_1[i_1]$ bzw. $T_1[i_2]$ besitzen. Die Definition der eingeschränkten Spur besagt jetzt, dass der letzte gemeinsame Vorgänger $lca(\bar{M}_1)$ der Knoten in $\bar{M}_1$ genau dann kein Vorgänger oder Nachfolger von Knoten in $\bar{M}_2$ ist, falls $lca(M_1)$ dies für keinen Knoten aus M_2 ist. Da aber $lca(M_1) = t_1[i_1]$, ist dies wegen der Disjunktheit von $T_1[i_1]$ und $T_1[i_2]$ erfüllt. Wir wissen also:

$$lca(\bar{M}_1) \not\leq t_2[n] \qquad \text{und} \qquad t_2[n] \not\leq lca(\overline{M}_1), \qquad n \in \overline{M}_2.$$

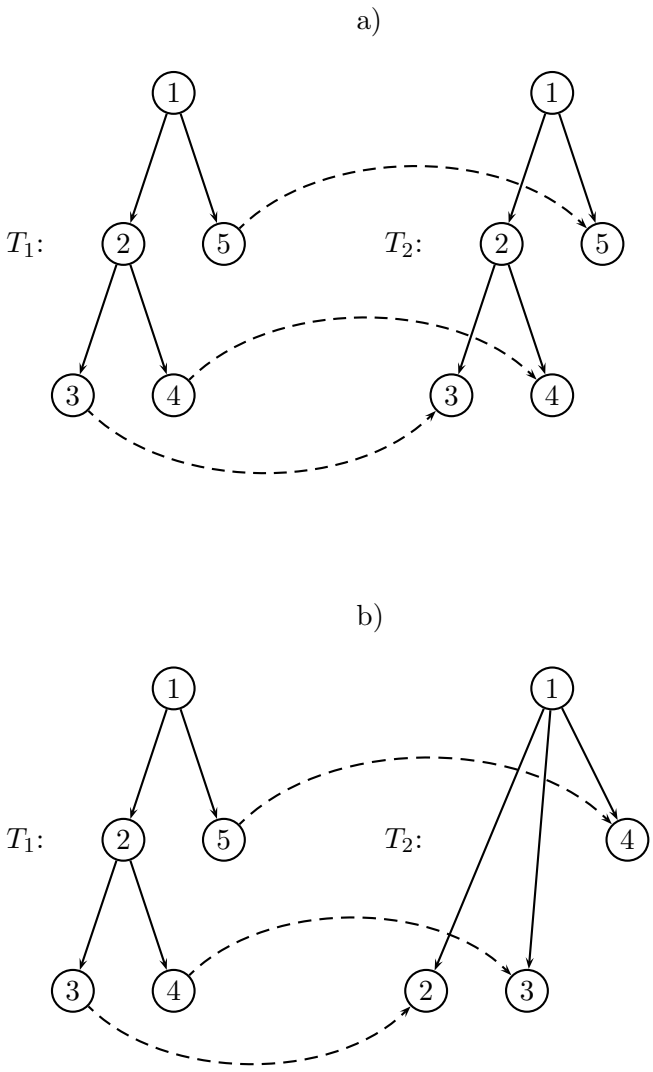


Abbildung 3.1: *Die gestrichelten Pfeile skizzieren eine Zuordnung der Knoten. Im Beispiel a) ist diese sowohl Spur als auch eingeschränkte Spur. Die Zuordnung in b) definiert eine Spur, aber keine eingeschränkte Spur, da* $lca(t_2[2], t_2[3]) \leq_T t_2[4]$ *aber* $lca(t_1[3], t_2[4]) \not\leq_T t_1[5]$.

Genauso folgert man

$$lca(\overline{M}_2) \not\leq t_2[n] \qquad \text{und} \qquad t_2[n] \not\leq lca(\overline{M}_2), \qquad n \in \overline{M}_1.$$

Damit ist $lca(\overline{M}_1)$ weder Vorgänger noch Nachfolger von $lca(\overline{M}_2)$:

$$lca(\overline{M}_1) \not\leq lca(\overline{M}_2) \qquad \text{und} \qquad lca(\overline{M}_2) \not\leq lca(\overline{M}_1)$$

und die Bilder der disjunkten Bäume $T_1[t_1]$ und $T_2[t_2]$ sind die disjunkten Bäume $T_2[lca(\overline{M}_1)]$ und $T_2[lca(\overline{M}_2)]$. Im zweiten Beispiel der Abbildung **??**, in dem die Spur keine eingeschränkte Spur ist, werden die beiden disjunkten Teilbäume $T_1[5]$ und $T_1[2]$ auf denselben Teilbaum von T_2, nämlich $T_2[1]$ abgebildet.

Wir wollen nun eine Abstandsfunktion über die Gewichtsfunktion $\Gamma(M)$ einer eingeschränkten Spur M definieren.

Definition 3.2. *Seien T_1 und T_2 zwei Bäume über einer Attributmenge Σ. Dann ist*

$$d^c_{edit}(T_1, T_2) = min\{\, \Gamma(M) \mid \text{M ist eingeschränkte Spur}\}$$

die eingeschränkte Edit-Distanz *von T_1 und T_2.*

Im Gegensatz zur Edit-Distanz ist die eingeschränkte Edit-Distanz nicht mehr über das Gewicht der besten Folge von Transformationsoperationen, sondern direkt über das minimale Gewicht einer Zuordnung von Knoten definiert, die gewisse Anordnungen der Knoten zueinander berücksichtigt.

Aufgrund dieser Definition ist zunächst klar, dass d_{edit} durch d^c_{edit} majorisiert wird. Zum Nachweis, dass d^c_{edit} sogar eine Metrik ist, benötigen wir zuerst das folgende Lemma.

Lemma 3.3. *Seien (M_1, T_1, T_2) und (M_2, T_2, T_3) zwei eingeschränkte Spuren.*

1. *Die Verknüpfung*

$$M_1 \circ M_2 = \{\, (i,j) \mid (i,k) \in M_1 \text{ und } (k,j) \in M_2\}$$

   ist wieder eine eingeschränkte Spur.

2. *Für das Gewicht von $\Gamma(M_1 \circ M_2)$ gilt:*

$$\Gamma(M_1 \circ M_2) \leq \Gamma(M_1) + \Gamma(M_2).$$

Beweis. 1. Da dies für jede Spur gilt, müssen wir nur die zweite Bedingung (**??**) einer eingeschränkten Spur nachweisen. Seien also (i_1, k_1), (i_2, k_2) und (i_3, k_3) drei Tupel aus $M_1 \circ M_2$. Dann gibt es $j_1, j_2, j_3 \in \mathbb{N}$, sodass $(i_1, j_1), (i_2, j_2), (i_3, j_3) \in M_1$ und $(j_1, k_1), (j_2, k_2), (j_3, k_3) \in M_2$. Unter Verwendung der Notationen $t_1[I] =$

$lca(t_1[i_1], t_1[i_2])$, $t_2[J] = lca(t_2[j_1], t_2[j_2])$ und $t_3[K] = lca(t_3[k_1], t_3[k_2])$ kann man die einschränkende Spurbedingung für M_1 und M_2 folgendermaßen schreiben:

$$t_1[I] \leq_T t_1[i_3] \Leftrightarrow t_2[J] \leq_T t_2[j_3]$$

und

$$t_2[J] \leq_t t_2[j_3] \Leftrightarrow t_3[K] \leq_T t_3[j_3].$$

Damit folgt direkt

$$t_1[I] \leq_T t_1[i_3] \Leftrightarrow t_3[K] \leq_T t_3[k_3].$$

$M_1 \circ M_2$ ist damit wieder eine eingeschränkte Spur.

2. Die Gültigkeit der Ungleichung $\Gamma(M_1 \circ M_2) \leq \Gamma(M_1) + \Gamma(M_2)$ beruht wieder auf der Dreiecksungleichung der lokalen Kostenfunktion $\gamma(s)$ und der Definition von $\Gamma(M)$.

□

Wir können nun zeigen, dass d^c_{edit} eine Metrik ist.

Satz 3.4. *Die in ?? definierte Abstandsfunktion $d^c_{edit}(T_1, T_2)$ stellt eine Metrik auf dem Raum der Bäume über Σ dar.*

Beweis. Der Beweis für die Gültigkeit der Dreiecksungleichung kann mit Lemma ?? wie der für Korollar ?? geführt werden. Die übrigen Eigenschaften einer Metrik folgen aus der Definition von $\Gamma(M)$, wobei M eine eingeschränkte Spur mit Gewicht $d^c_{edit}(T_1, T_2)$ ist. □

Die folgende Bemerkung verallgemeinert die Distanz d^c_{edit} auf Wälder.

Bemerkung 3.5. *Der Begriff der Spur und der eingeschränkten Spur kann für Wälder F eingeführt werden. Dazu betrachten wir den Baum $T(F)$, der aus einem Wald $F = \{T_1 \dots T_n\}$ entsteht, wenn ein Knoten root hinzugefügt und dieser zum Vater der Wurzeln aller Bäume $T_i \in F$ gemacht wird.*

- *Seien F_1 und F_2 zwei Wälder. Eine Tupelmenge M_F heißt (eingeschränkte) Spur von F_1 und F_2, falls $M_F \cup \{(root_1, root_2)\}$ eine (eingeschränkte) Spur von $T(F_1)$ und $T(F_2)$ ist.*
- *Das Gewicht einer Spur M_F ist über die lokale Kostenfunktion γ definiert.*

$$\begin{aligned}\Gamma(M_F) := & \sum_{(i,j)\in M_F} \gamma\big(label(t_1[i]) \to label(t_2[j])\big) \\ & + \sum_{\{i|\forall t_2[j],(i,j)\notin M_F\}} \gamma\big(label(t_1[i]) \to \lambda\big) \\ & + \sum_{\{j|\forall t_2[i],(i,j)\notin M_F\}} \gamma\big(\lambda \to label(t_2[j])\big).\end{aligned}$$

- *Die Distanzen $d_{edit}(F_1, F_2)$ und $d^c_{edit}(F_1, F_2)$ sind über das minimale Gewicht einer Spur bzw. eingeschränkten Spur von F_1 und F_2 definiert.*

3.2 Eigenschaften von d^c_{edit}

Der Algorithmus zur Berechnung der Edit-Distanz zweier Wörter beruht auf der Berechnung der Edit-Distanz von Präfixwörtern. Wir wollen jetzt zeigen, dass wir die Distanz d^c_{edit} zweier Bäume T_1 und T_2 aus der Distanz von Teilbäumen bestimmen können. Wir führen $D(T_1[i], T_2[j])$ und $D(F_1[i], F_2[j])$ als abkürzende Schreibweise für die Distanz $d^c_{edit}(T_1[i], T_2[j])$ und $d^c_{edit}(F_1[i], F_2[j])$ ein und schreiben $\gamma(i,j)$ für $\gamma\big(label(t_1[i]), label(t_2[j])\big)$, $\gamma(i,\lambda)$ für $\gamma\big(label(t_1[i]), \lambda\big)$ und $\gamma(\lambda, j)$ für $\gamma\big(\lambda, label(t_2[j])\big)$.

Lemma 3.6. *Seien $t_1[i_1], \dots t_1[i_{n_i}]$ die Kinderknoten von $t_1[i]$ in T_1 und $t_2[j_1], \dots t_2[j_{n_j}]$ die Kinderknoten von $t_2[j]$ in T_2. Dann ist*

- $D(\Theta, \Theta) = 0$,
- $D(F_1[i], \Theta) = \sum_{k=1}^{n_i} D(T_1[i_k], \Theta)$,
- $D(\Theta, F_2[j]) = \sum_{l=1}^{n_j} D(\Theta, T_2[j_l])$,
- $D(T_1[i], \Theta) = D(F_1[i], \Theta) + \gamma(i, \lambda)$,
- $D(\Theta, D(T_2[j])) = D(\Theta, F_2[j]) + \gamma(\lambda, j)$.

Beweis. Die einzige zulässige Spur ist $M = \emptyset$. Die Behauptung folgt dann aus der Definition von d^c_{edit} und $\Gamma(M)$. □

Lemma 3.7. *Seien $t_1[i_1], \dots t_1[i_{n_i}]$ die Kinderknoten von $t_1[i]$ in T_1 und $t_2[j_1], \dots t_2[j_{n_j}]$ die Kinderknoten von $t_2[j]$ in T_2. Dann ist*

$$D(T_1[i], T_2[j]) = min \begin{cases} D(\Theta, T_2[j]) + \min\limits_{1 \le t \le n_j} \{D(T_1[i], T_2[j_t]) - D(\Theta, T_2[j_t])\}, \\ D(T_1[i], \Theta) + \min\limits_{1 \le s \le n_i} \{D(T_1[i_s], T_2[j]) - D(T_1[i_s], \Theta)\}, \\ D(F_1[i], F_2[j]) + \gamma(i,j). \end{cases}$$

Beweis. Sei M eine eingeschränkte Spur von $T_1[i]$ und $T_2[j]$ mit $\Gamma(M) = d^c_{edit}(T_1, T_2)$. Dann können wir die folgenden vier Fälle unterscheiden:

1. $(i,l) \in M$ und $(k,j) \notin M$, $k, l \in \mathbb{N}$.
   Dann ist $t_2[l]$ ein Knoten in $F_2[j]$ und da M eingeschränkte Spur ist, werden alle Knoten von $T_1[i]$ auf einen Teilbaum $T_2[j_t]$ abgebildet. Es gibt also ein $t \in \mathbb{N}$ mit $t_2[j_t] \le_T t_2[l']$ für alle $(k,l') \in M$. Damit gilt

   $$D(T_1[i], T_2[j]) = D(T_1[i], T_2[j_t]) + \gamma(\lambda, j) + \sum_{q \ne t} D(\Theta, T_2[j_q])$$

   Benutzt man $D(\Theta, T_2[j]) = \gamma(\lambda, j) + \sum_{q=1}^{n_j} D(\Theta, T_2[j_q])$, so vereinfacht sich dies zu:

   $$D(T_1[i], T_2[j]) = D(\Theta, T_2[j]) + D(T_1[i], T_2[j_t]) - D(\Theta, T_2[j_t]).$$

Damit ist

$$D(T_1[i], T_2[j]) = D(\Theta, T_2[j]) + \min_{1 \leq t \leq n_j} \{D(T_1[i], T_2[j_t]) - D(\Theta, T_2[j_t])\}.$$

2. $(i,l) \notin M$ und $(k,j) \in M$, $k,l \in \mathbb{N}$.
   Analog zu **??** können wir

   $$D(T_1[i], T_2[j]) = D(T_1[i], \Theta) + \min_{1 \leq s \leq n_i} \{D(T_1[i_s], T_2[j]) - D(T_1[i_s], \Theta)\}$$

   zeigen.

3. $(i,l) \in M$ und $(k,j) \in M$, $k,l \in \mathbb{N}$.
   Damit ist aber $i = k$ und $l = j$. Da $M \setminus \{(i,j)\}$ eine eingeschränkte Spur von $F_1[i]$ und $F_2[j]$ ist und umgekehrt zu jeder eingeschränkten Spur M' von $F_1[i]$ und $F_2[j]$ die Tupelmenge $M' \cup \{(i,j)\}$ eine eingeschränkte Spur von $T_1[i]$ und $T_2[j]$ ist, können wir

   $$D(T_1[i], T_2[j]) = D(F_1[i], F_2[i]) + \gamma(i,j)$$

   schreiben.

4. $(i,l) \notin M$ und $(k,j) \notin M$, $k,l \in \mathbb{N}$.
   Analog zu **??** können wir

   $$D(T_1[i], T_2[j]) = D(F_1[i], F_2[i]) + \gamma(i,\lambda) + \gamma(\lambda,j)$$

   schreiben. Da aber immer

   $$\gamma(i,j) \leq \gamma(i,\lambda) + \gamma(\lambda,j)$$

   braucht dieser Fall nicht betrachtet zu werden.

□

Das Lemma **??** besagt jetzt, dass die Edit-Distanz zweier Teilbäume $T_1[i]$ und $T_2[j]$ aus der Edit-Distanz eines Teilbaumes mit den Bäumen an den Wurzelkindern des anderen und der Edit-Distanz zwischen den Wäldern $F_1[i]$ und $F_2[j]$ bestimmt werden kann. Um eine ähnliche Eigenschaft für die Wälder angegeben zu können, führen wir einen Namen für eine spezielle eingeschränkte Spur ein.

Definition 3.8 (limitierte Spur)**.** *Eine eingeschränkte Spur $M_F(i,j)$ zweier Wälder $F_1[i]$ und $F_2[j]$ heißt* limitierte Spur $M_{lim}(i,j)$*, falls gilt:*

- *Gibt es $(k,l) \in M_F(i,j)$, so dass $t_1[k]$ Knoten von $T_1[i_s]$ und $t_2[l]$ Knoten von $T_2[j_t]$ sind, dann ist für ein Tupel $(k_1,l_1) \in M$ der Knoten $t_1[k_1]$ im Teilbaum $T_1[i_s]$ genau dann, wenn $t_2[l_1]$ in $T_1[i_s]$ ist.*

Die limitierten Spuren sind also nur diejenigen eingeschränkten Spuren, die Knoten in dem Teilbaum $T_1[i_s]$ vollständig auf einen Teilbaum $T_2[j_t]$ abbilden und umgekehrt.

Lemma 3.9. *Seien $t_1[i_1], \ldots t_1[i_{n_i}]$ die Kinderknoten von $t_1[i]$ in T_1 und $t_2[j_1], \ldots t_2[j_{n_j}]$ die Kinderknoten von $t_2[j]$ in T_2. Dann ist*

$$D(F_1[i], F_2[j]) = min \begin{cases} D(\Theta, F_2[j]) + \min\limits_{1 \leq t \leq n_j} \{D(F_1[i], F_2[j_t]) - D(\Theta, F_2[j_t])\}, \\ D(F_1[i], \Theta) + \min\limits_{1 \leq s \leq n_i} \{D(F_1[i_s], F_2[j]) - D(F_1[i_s], \Theta)\}, \\ \min\limits_{M_{lim}(i,j)} \Gamma(M_{lim}). \end{cases}$$

Beweis. Sei M eine eingeschränkte Spur von $F_1[i]$ und $F_2[j]$ mit $\Gamma(M) = d^c_{edit}(F_1, F_2)$. Dann können wir wieder vier Fälle unterscheiden:

1. $\exists_{t \in \mathbb{N}} \forall_{(k,l) \in M} : t_2[l] \in T_2[j_t]$ und $\nexists_{s \in \mathbb{N}} \forall_{(k,l) \in M} : t_1[k] \in T_1[i_s]$:
   Es existiert ein $t, 1 \leq t \leq n_j$, so dass für alle $(k,l) \in M$ der Knoten $t_2[l]$ in $T_2[j_t]$ liegt und es gibt $(k_1, l_1), (k_2, l_2) \in M$, so dass $t_1[k_1]$ und $t_1[k_2]$ in zwei verschiedenen Teilbäumen $T_1[i_{s_1}]$ und $T_1[i_{s_2}]$, $s_1 \neq s_2$, liegen. Damit kann aber (r, j_t) nicht in M liegen. Analog zum Fall **??** von Lemma **??** folgt damit
   $$D(F_1[i], F_2[j]) = D(\Theta, F_2[j]) + \min_{1 \leq t \leq n_j} \{D(F_1[i], F_2[j_t]) - D(\Theta, F_2[j_t])\}$$

2. $\nexists_{t \in \mathbb{N}} \forall_{(k,l) \in M} : t_2[l] \in T_2[j_t]$ und $\exists_{s \in \mathbb{N}} \forall_{(k,l) \in M} : t_1[k] \in T_1[i_s]$:
   Wie in **??** erhalten wir
   $$D(F_1[i], F_2[j]) = D(F_1[i], \Theta) + \min_{1 \leq s \leq n_i} \{D(F_1[i_s], F_2[j]) - D(F_1[i_s], \Theta)\}.$$

3. $\exists_{t \in \mathbb{N}} \forall_{(k,l) \in M} : t_2[l] \in T_2[j_t]$ und $\exists_{s \in \mathbb{N}} \forall_{(k,l) \in M} : t_1[k] \in T_1[i_s]$:
   Für jedes Tupel $(k,l) \in M$ liegen die Knoten $t_1[k]$ und $t_2[l]$ in $T_1[s]$ bzw. $T_2[t]$. M ist damit limitierte Spur und
   $$D(F_1[i], F_2[j]) = \min_{M_{lim}(i,j)} \Gamma(M_{lim}(i,j))$$

4. $\nexists_{t \in \mathbb{N}} \forall_{(k,l) \in M} : t_2[l] \in T_2[j_t]$ und $\nexists_{s \in \mathbb{N}} \forall_{(k,l) \in M} : t_1[k] \in T_1[i_s]$:
   Wir wollen durch einen Widerspruchsbeweis zeigen, dass M auch in diesem Fall eine limitierte Spur ist. Falls M nicht limitierte Spur ist, gibt es $(a_1, b_1), (a_2, b_2) \in M$ mit $t_1[a_1], t_2[a_2] \in T_1[i_s]$ aber $t_2[b_1] \in T_2[j_{t_1}]$ und $t_2[b_2] \in T_2[j_{t_2}]$ $(t_1 \neq t_2)$. Weiter wählen wir $(a_3, b_3) \in M$ so, dass $t_1[a_3] \notin T_1[i_s]$. Damit gilt aber $lca(t_1[a_1], t_1[a_2]) \not\leq_T t_1[a_3]$ und $t_1[a_3] \not\leq_T lca(t_1[a_1], t_1[a_2])$. Da aber $lca(t_2[b_1], t_2[b_2]) = t_2[j] \leq_T t_2[b_3]$ haben wir hier einen Widerspruch zur Definition der eingeschränkten Spur.
   $$D(F_1[i], F_2[j]) = \min_{M_{lim}(i,j)} \Gamma(M_{lim}(i,j))$$

□

Die Berechnung der Distanz d^c_{edit} beruht damit wesentlich auf der Berechnung von $\min_{M_{lim}(i,j)} \Gamma(M_{lim})$. Wir zeigen nun, dass dieses Problem so formulierbar ist, dass es

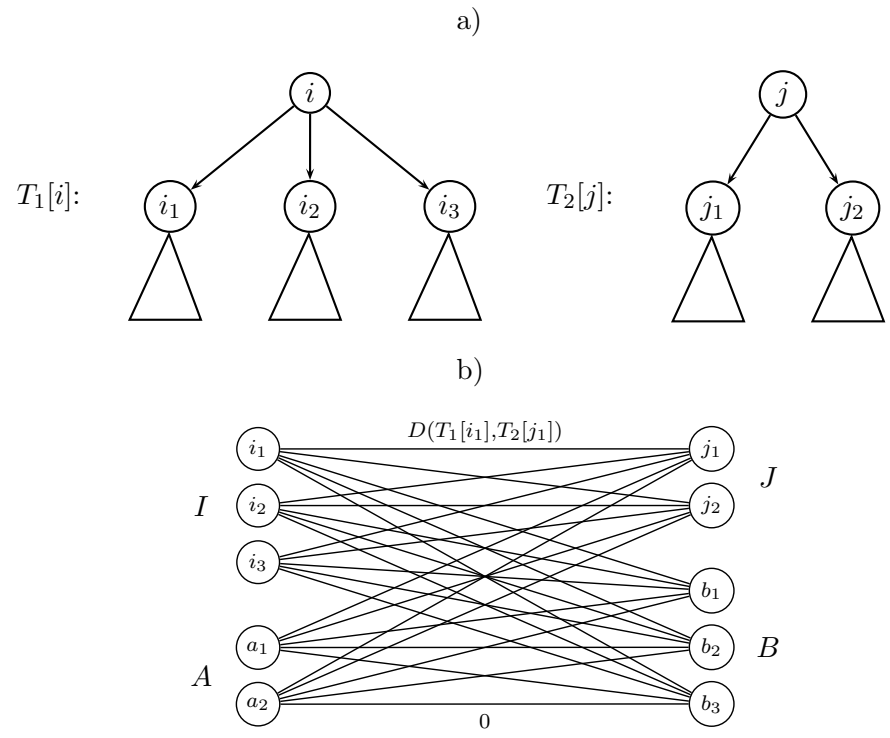


Abbildung 3.2: *Der Graph $G_H(i,j)$ für die Bäume $T_1[i]$ und $T_2[j]$.*

mit Hilfsmitteln aus der diskreten Mathematik gelöst werden kann. Dass eine Spur M zwischen $T_1[i]$ und $T_2[j]$ limitiert ist, bedeutet, dass die Knoten eines Teilbaums $T_1[i_s]$ entweder komplett auf einen Teilbaum $T_2[j_t]$ oder auf den leeren Baum Θ abgebildet werden. Wir können daher $\min_{M_{lim}(i,j)}\Gamma(M_{lim})$ über die Distanzen $d^c_{edit}(T_1[i_s],T_2[j_t])$, $d^c_{edit}(\Theta,T_2[j_t])$ und $d^c_{edit}(T_1[i_s],\Theta)$ bestimmen. Dazu betrachten wir einen kantengewichteten Graphen $G_H(i,j) = (V,E,cost)$ (Abb. **??**). Mit den Bezeichnungen $I = \{i_1,\dots i_{n_i}\}$, $J = \{j_1,\dots j_{n_j}\}$, $A = \{a_1,\dots a_{n_j}\}$, $B = \{b_1,\dots b_{n_i}\}$, $L = I \cup A$ und $R = J \cup B$ definieren wir $G_H(i,j)$ folgendermaßen:

1. $V = L \cup R$,
2. $E = \{\ (l,r) \mid l \in L, r \in R\}$,
3. $cost(l,r) = D(T_1[l],T_2[r])$ falls $l \in I, r \in J$,
   $cost(l,r) = D(T_1[l],\Theta)$ falls $l \in I, r \in B$,
   $cost(l,r) = D(\Theta,T_2[r])$ falls $l \in A, r \in J$,
   $cost(l,r) = 0$ falls $r \in A, l \in B$.

Ein *Matching* auf $G_H(i,j)$ ist eine Teilmenge $MT \subseteq E$ der Kanten, so dass für alle $v \in L$ höchstens eine Kante (v,r), $r \in R$, und für alle $w \in R$ höchstens eine Kan-

te (l, w), $l \in L$, in MT ist. Ein *maximales Matching* MT_{max} ist ein Matching, mit $|MT| \leq |MT_{max}|$ für alle Matchings MT. Für das maximale Matching $MT_{max}(i,j)$ auf $G_H(i,j)$ gilt $|MT_{max}(i,j)| = n_i + n_j$. Das Gewicht eines maximalen Matchings MT_{max} wird über die Gewichte der Kanten in MT_{max} definiert.

$$cost(MT_{max}) = \sum_{e \in MT_{max}} cost(e)$$

Das Lemma **??** stellt nun den Zusammenhang zwischen einem maximalen Matching auf $G_H(i,j)$ und der limitierten Spur $M_{lim}(i,j)$ heraus.

Lemma 3.10. *Sei $M_{lim}(i,j)$ eine limitierte Spur von $F_1[i]$ und $F_2[j]$ und $MT_{max}(i,j)$ ein maximales Matching des Graphen $G_H(i,j)$. Dann gilt:*

$$\min_{M_{lim}(i,j)} \Gamma(M_{lim}(i,j)) = \min_{MT_{max}(i,j)} cost(MT_{max}(i,j))$$

Beweis. Aufgrund der Konstruktion von $G_H(i,j)$ ist klar, dass jede limitierte Spur ein maximales Matching mit

$$\Gamma(M_{lim}(i,j)) = cost(MT_{max}(i,j))$$

und umgekehrt jedes maximale Matching eine limitierte Spur mit

$$cost(MT_{max}(i,j)) = \Gamma(M_{lim}(i,j))$$

induziert. □

In der diskreten Mathematik gibt es nun verschiedene Algorithmen, die ein maximales Matching mit minimalem Gewicht berechnen ([**?**] S. 470ff). Wir sind damit in der Lage, die Distanz d^c_{edit} effektiv zu bestimmen.

3.3 Algorithmus und Komplexität

Die Ergebnisse aus den Lemmata **??**, **??** und **??** ergeben eine rekursive Beschreibung der eingeschränkten Edit-Distanz zwischen zwei Bäumen. Der Nachteil eines rekursiven Algorithmus wäre die hohe Komplexität. Durch eine geeignete Nummerierung der Knoten kann aber ein iterativer Algorithmus angegeben werden, der die Mehrfachberechnung von Teilergebnissen umgeht und damit deutlich schneller ist. Da wir zur Berechnung von $d^c_{edit}(T_1[i], T_2[j])$ nur Distanzwerte zwischen Teilbäumen von $T_1[i]$ und $T_2[i]$ benötigen, müssen wir lediglich sicherstellen, dass in einem iterativen Algorithmus diese Distanzwerte zuerst bestimmt werden. Wählen wir also eine postorder-Nummerierung der Knoten, so erhalten wir den Algorithmus in Abbildung **??** zur Berechnung der Distanz d^c_{edit} zweier Bäume T_1 und T_2. In jedem Iterationsschritt wird zur Berechnung von $D(F_1[i], F_2[j])$ mit $\min_{MT_{max}(i,j)} cost(MT_{max}(i,j))$ das maximale Matching mit minimalem Gewicht auf

Require: T_1 und T_2
1: $D(\Theta, \Theta) \leftarrow 0$
2: **for** $i = 1$ to $|T_1|$ **do**
3: $\quad D(F_1[i], \Theta) \leftarrow \sum_{k=1}^{n_i} D(T_1[i_k], \Theta)$
4: $\quad D(T_1[i], \Theta) \leftarrow D(F_1[i], \Theta) + \gamma(t_1[i] \longrightarrow \lambda)$
5: **end for**
6: **for** $j = 1$ to $|T_2|$ **do**
7: $\quad D(\Theta, F_2[j]) \leftarrow \sum_{l=1}^{n_j} D(\Theta, T_2[n_l])$
8: $\quad D(\Theta, T_2[j]) \leftarrow D(\Theta, F_2[j]) + \gamma(\lambda \longrightarrow t_2[j])$
9: **end for**
10: **for** $i = 1$ to $|T_1|$ **do**
11: $\quad$ **for** $j = 1$ to $|T_2|$ **do**
12: $\quad\quad f_1 \leftarrow D(\Theta, F_2[j]) + \min_{1 \le t \le n_j} \{D(F_1[i], F_2[j_t]) - D(\Theta, F_2[j_t])\}$
13: $\quad\quad f_2 \leftarrow D(F_1[i], \Theta) + \min_{1 \le s \le n_i} \{D(F_1[i_s], F_2[j]) - D(F_1[i_s], \Theta)\}$
14: $\quad\quad f_3 \leftarrow \min_{MT_{max}(i,j)} cost(MT_{max}(i, j))$
15: $\quad\quad D(F_1[i], F_2[j]) \leftarrow \min\{f_1, f_2, f_3\}$
16: $\quad\quad t_1 \leftarrow D(\Theta, T_2[j]) + \min_{1 \le t \le n_j} \{D(T_1[i], T_2[j_t]) - D(\Theta, T_2[j_t])\}$
17: $\quad\quad t_2 \leftarrow D(T_1[i], \Theta) + \min_{1 \le s \le n_i} \{D(T_1[i_s], T_2[j]) - D(T_1[i_s], \Theta)\}$
18: $\quad\quad t_3 \leftarrow D(F_1[i], F_2[j]) + \gamma(i, j)$
19: $\quad\quad D(T_1[i], T_2[j]) \leftarrow \min\{t_1, t_2, t_3\}$
20: $\quad$ **end for**
21: **end for**

Abbildung 3.3: *Algorithmus zur Berechnung von* $d^c_{edit}(T_1, T_2)$*,* $|T_i|$ *ist dabei die Anzahl der Knoten im Baum* T_i*.*

dem Graphen $G_H(i,j)$ als Teilproblem bestimmt. Die Komplexität des Algorithmus hängt damit auch wesentlich von der Komplextität des Teilproblems ab. Beachten wir, dass der Graph $G_H(i,j)$ $2(n_i+n_j)$ Knoten und $(n_i+n_j)^2$ Kanten hat, so macht der folgenden Satz [**?**] eine Aussage über die Komplexität des Teilproblems macht.

Satz 3.11 ([**?**] S. 246)**.** *Die Laufzeit zur Bestimmung des maximalen Matchings auf $G_H(i,j)$ mit minimalem Gewicht ist*

$$O\big((n_i+n_j)^3+(n_i+n_j)^2 log(n_i+n_j)\big).$$

Bezeichnen wir mit deg_i eine obere Schranke für die Anzahl der Kinderknoten im Baum T_i, so können wir eine Aussage über die Komplexität des Algorithmus machen.

Satz 3.12. *Die Laufzeit des Algorithmus aus Abbildung* **??** *zur Bestimmung der eingeschränkten Edit-Distanz von zwei Bäumen T_1 und T_2 ist*

$$O\big(|T_1||T_2|((deg_1+deg_2)^3+(deg_1+deg_2)^2 log(deg_1+deg_2))\big).$$

Beweis. Der Aufwand setzt sich hauptsächlich aus der doppelten Iteration und dem Teilproblem zusammen.

$$\sum_{i=1}^{|T_1|}\sum_{j=1}^{|T_2|} O\big((n_i+n_j)^3+(n_i+n_j)^2 log(n_i+n_j)\big)$$

Wir benutzen nun $n_i \leq deg_1$ und $n_j \leq deg_2$ und erhalten so die Aussage des Satzes. □

Bei Problemstellungen, bei denen die Konstanten deg_1 und deg_2 sehr groß werden können, kann die Laufzeit noch verbessert werden, indem die besondere Struktur von $G_H(i,j)$ ausgenutzt wird, um das Teilproblem direkt durch einen min-cost-flow-Algorithmus zu lösen. Für die genaue Definition des min-cost-flow-Problems verweisen wir auf [**?**] und [**?**]. Wir wollen hier lediglich die Formulierung des Teilproblems angeben. Wir definieren dafür einen Graphen $G_{flow}(i,j)$, die Kapazität $cap : E \longrightarrow \mathbb{N}_0^+$ und die Kosten $c : E \longrightarrow \mathbb{R}_0^+$

der Kanten (Abb. **??**):

$$
\begin{aligned}
V &= \{s,t,e_i,e_j\} \cup I \cup J \\
\\
E &= \{(s,e_i),(e_j,t),(e_i,e_j)\} \\
&\quad \cup \{(s,i_k)|i_k \in I\} \cup \{(i_k,e_j)|i_k \in I\} \\
&\quad \cup \{i_k,j_l|i_k \in I, j_l \in J\} \\
&\quad \cup \{(e_i,j_k)|j_k \in J\} \cup \{(j_k,t)|j_k \in J\}
\end{aligned}
$$

$$
cap(e) = \begin{cases} n_j & falls \quad e=(s,e_i), \\ n_i * n_j & falls \quad e=(e_i,e_j), \\ n_i & falls \quad e=(e_j,t), \\ 1 & sonst. \end{cases}
$$

$$
c(e) = \begin{cases} 0 & falls \quad e \in \{(s,e_i),(e_j,t),(e_i,e_j)\} \\ 0 & falls \quad e \in \{(s,i_k)|i_k \in I\} \cup \{(j_k,t)|j_k \in J\} \\ D(T_1[i_k],T_2[j_l]) & falls \quad e=(i_k,j_l) \\ D(T_1[i_k],\Theta) & falls \quad e=(i_k,e_j) \\ D(\Theta,T_2[j_l]) & falls \quad e=(e_i,j_l) \end{cases}
$$

Ein (s,t)-Fluß auf $G_{flow}(i,j)$ ist eine Funktion $flow : E \longrightarrow \mathbb{N}_0^+$, die den folgenden Erhaltungsgleichungen genügt:

$$
\begin{aligned}
flow(e) &\leq cap(e) \qquad \forall e \in E \\
\\
\sum_{e=(v_i,\cdot)} flow(e) &= \sum_{e=(\cdot,v_i)} flow(e) \qquad \forall v_i \in V \setminus \{s,t\} \\
\sum_{e=(s,\cdot)} flow(e) &= \sum_{e=(\cdot,t)} flow(e)
\end{aligned}
$$

Der Wert des (s,t)-Flusses ist $\sum_{e=(s,\cdot)} f(e)$. Der maximale Wert $flow_{max}(i,j)$ eines (s,t)-Flusses in $G_{flow}(i,j)$ ist $n_i + n_j$. Das Gewicht eines Flusses ist $C_{flow} = \sum_{e \in E} c(e) flow(e)$. Aufgrund der Konstruktion ist klar, dass ein maximaler Fluss auf $G_{flow}(i,j)$ ein maximales Matching $M_{max}(i,j)$ auf $G_H(i,j)$ mit $C(flow) = cost(MT_{max}(i,j))$ induziert und umgekehrt. Es gilt daher:

$$\min_{MT_{max}(i,j)} cost(MT_{max}(i,j)) = \min_{flow_{max}(i,j)} C(flow_{max}(i,j))$$

Verwenden wir nun einen geeigneten Algorithmus [**?**] um den Fluss mit minimalen Kosten auf $G_{flow}(i,j)$ zu bestimmen, so kann der Abstand zweier Bäume T_1 und T_2 in $O\big(|T_1||T_2|(deg_1+deg_2)log(deg_1+deg_2)\big)$ bestimmt werden ([**?**]).

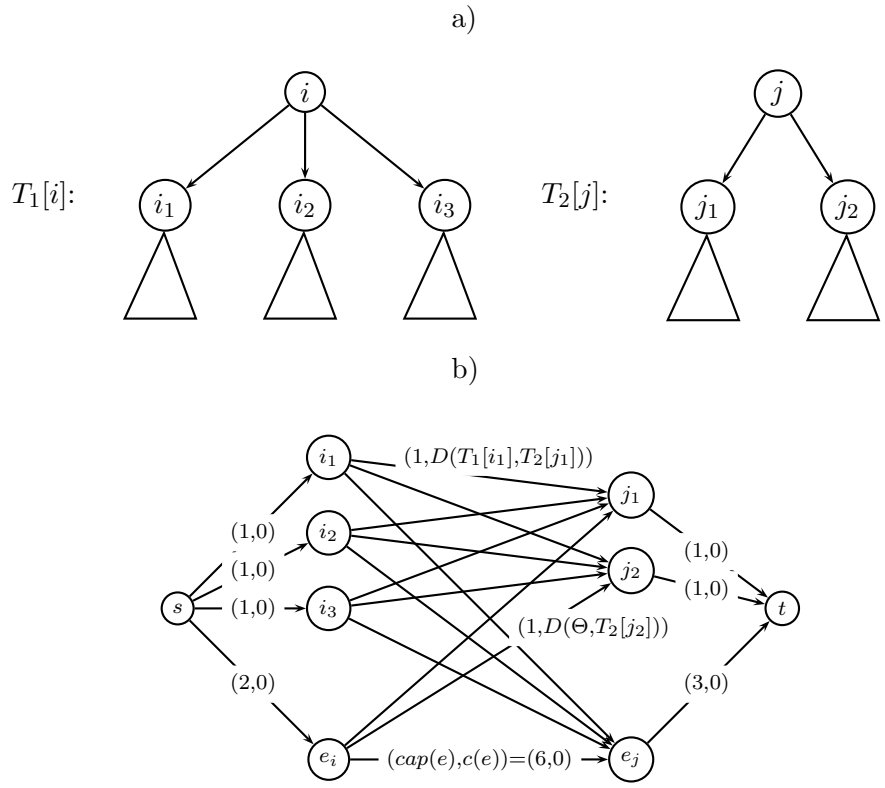


Abbildung 3.4: *Der Graph $G_{flow}(i,j)$ für die Bäume $T_1[i]$ und $T_2[j]$.*

Kapitel 4

Neuronen und die Edit-Distanz

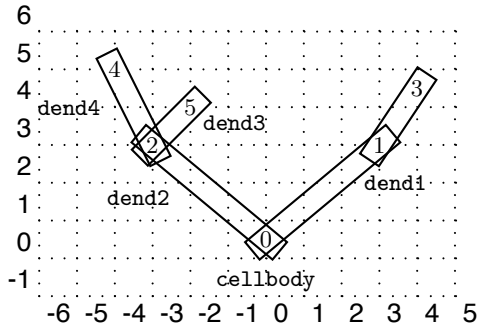
In Kapitel **??** haben wir mit der eingeschränkten Edit-Distanz eine Metrik auf dem Raum der ungeordneten knotengewichteten Bäume eingeführt. Wir wollen nun zeigen, wie wir diese Metrik verwenden können um ein metrisches Distanzmaß auf der Menge der Nervenzellen zu definieren. Dazu müssen wir das Modell der Morphologie aus Kapitel **??** in einen knotengewichteten Baum überführen und eine geeignete lokale Kostenfunktion auf der Menge der Knotengewichte finden.

4.1 Darstellung von Neuronen als knotengewichtete Bäume

In Abschnitt **??** hatten wir die Zellmorpholgie stückweise durch Kegelstümpfe approximiert. Wir wollen nun die Kegelstümpfe zwischen zwei Verzweigungspunkten als eine Einheit auffassen. Wir nennen diese Einheit in Anlehnung an das hoc-Format (Abb. **??**) *Sektion*. Die topologische Struktur der Zelle, d.h. die Verzweigungen der Dendriten, gibt nun für jede Sektion der Zelle die Vorgänger- und Nachfolgersektionen vor. Wir haben damit eine Knotenmenge und eine Kantenmenge, die zusammen einen Baum T_{cell} definieren (Abb. **??**). Da grundsätzlich davon ausgegangen wird, dass jeder dendritische Verzweigungspunkt eine Bifurkation ist, sind die Teilbäume von T_{cell} binäre Bäume.

Als nächstes muss eine geeignete Attributmenge Σ für die Knoten von T_{cell}, die Sektionen, gefunden werden. Da T_{cell} bisher nur die topologische Struktur einer Zelle wiedergibt, bietet es sich an, geometrische Eigenschaften der Sektionen wie etwa Länge, Volumen, Oberfläche oder Abstand zum Soma als Knotenattribute zu verwenden. Es ist sinnvoll anzunehmen, dass die geometrischen Attribute einer Sektion nicht alle Null sind. Wir können also die Bestimmung des Abstandes zweier Nervenzellen auf die Bestimmung des Abstandes zweier Bäume T_{cell_1} und T_{cell_2} über einer Attributmenge $\mathbb{R}^n \setminus \{0\}$ zurückführen. Das Prinzip der Edit-Distanz besagt nun, dass der Abstand von T_{cell_1} und T_{cell_2} als die minimalen Kosten einer Folge von Edit-Operationen definiert ist, die T_{cell_1} in T_{cell_2} überführt. Die Edit-Operationen auf T_{cell} entsprechen dann anschaulich gesprochen, der Änderung

a)



b)

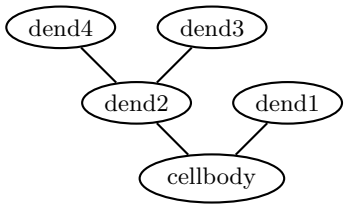


Abbildung 4.1: *Die Sektionen cellbody, dend*1*, dend*2*, dend*3*, dend*4 *der Zellmorphologie in a) bilden die Knoten des Baumes in b).*

geometrischer Eigenschaften einer Sektion, wie etwa der Länge. Da die Edit-Distanz für Bäume nicht in polynomialer Zeit bestimmbar ist, wollen wir die eingeschränkte Edit-Distanz d^c_{edit} verwenden, um ein Maß für die Ähnlichkeit von Nervenzellen zu erhalten. Nach Tanaka und Tanaka [**?**] ist die eingeschränkte Spur bei Klassifikationsproblemen sogar geeigneter als die allgemeine Spur.

4.2 Lokale Kostenfunktionen für Neuronen

Zunächst wollen wir eine lokale Kostenfunktion einführen, die unabhängig von den Attributen der Knoten definiert ist.

Definition 4.1 (lokale topologische Kostenfunktion)**.** *Für zwei Bäume* T_1 *und* T_2 *über der Attributmenge* $\Sigma = \mathbb{R}^n \setminus \{0\}$ *und* $\boldsymbol{x}, \boldsymbol{y} \in \Sigma$ *ist durch*

$$\gamma_{top}(\boldsymbol{x}, \lambda) := \gamma_{top}(\lambda, \boldsymbol{x}) := 1$$

und

$$\gamma_{top}(\boldsymbol{x}, \boldsymbol{y}) := 0$$

die lokale topologische Kostenfunktion γ_{top} *definiert.* γ_{top} *ist insbesondere eine Metrik auf* Σ_λ.

Verwenden wir γ_{top} zur Definition des Gewichtes einer Spur von T_1 und T_2, so gibt $d^c_{edit}(T_1, T_2)$ die minimale Anzahl der Knoten an, die in T_1 eingefügt bzw. gelöscht werden müssen, um T_2 zu erhalten.

Kommen wir nun zu lokalen Kostenfunktionen, die die Geometrie der Sektionen berücksichtigen. Da wir Nervenzellen durch Bäume über der Attributmenge $\Sigma = \mathbb{R}^n \setminus \{0\}$ dar-

stellen, liegt es nahe, bei der Definition von γ die l^p-Normen

$$\begin{aligned} \|\mathbf{x}\|_p &:= \left(\sum_{k=1}^{n} |x_k|^p\right)^{1/p} && falls\, p \in \mathbb{N},\ \mathbf{x} \in \mathbb{R}^n \\ \|\mathbf{x}\|_p &:= \max_{1 \le k \le n} |x_k| && falls\, p = \infty,\ \mathbf{x} \in \mathbb{R}^n \end{aligned}$$

zu verwenden. Die Edit-Operationen entsprechen der Änderung geometrischer Eigenschaften von Sektionen und es ist bietet sich daher an, das Gewicht von Einfüge- und Löschoperationen $\lambda \longrightarrow \mathbf{x}$ bzw. $\mathbf{x} \longrightarrow \lambda$ über den Abstand von $\mathbf{x}$ zum Nullvektor $\mathbf{0}$ zu definieren:

$$\gamma(\mathbf{x}, \lambda) = \gamma(\lambda, \mathbf{x}) = \|\mathbf{x}\|_p.$$

Mit

$$\gamma(\mathbf{x}, \mathbf{y}) := \|\mathbf{x} - \mathbf{y}\|_p.$$

für Substitutionsoperationen $\mathbf{x} \longrightarrow \mathbf{y}$ ist γ damit eine Metrik auf Σ_λ.

Definition 4.2 (lokale Kostenfunktion)**.** *Für zwei Bäume T_1 und T_2 über der Attributmenge $\Sigma = \mathbb{R}^n \setminus \{0\}$ und $\boldsymbol{x}, \boldsymbol{y} \in \Sigma$ ist durch*

$$\gamma_p(\boldsymbol{x}, \lambda) := \gamma_p(\lambda, \boldsymbol{x}) := \|\boldsymbol{x}\|$$

und

$$\gamma_p(\boldsymbol{x}, \boldsymbol{y}) := \|\boldsymbol{x} - \boldsymbol{y}\|$$

die lokale Kostenfunktion γ_p definiert. γ_p ist insbesondere eine Metrik auf Σ_λ.

Es ist zu beachten, dass in einem Baum, in dem alle Knoten v das Attribut $label(v) = 1$ haben, die lokale Kostenfunktion γ_p der topologischen Kostenfunktion γ_{top} entspricht. Weiterhin gilt $\gamma_p(x, y) = \gamma_q(x, y)$ für $x,\ y \in \mathbb{R},\ q,\ p \in \mathbb{N}$.

4.3 Eine metrische Distanz für Neuronen

Fassen wir noch einmal die zentrale Idee dieser Arbeit zusammen.

Bemerkung 4.3 (metrische Distanz für Nervenzellen)**.** *Stellt man beliebige Nervenzellen $cell_1$ und $cell_2$ als knotengewichtete Bäume T_{cell_1} und T_{cell_2} über $\mathbb{R}^n$ dar, so ist die eingeschränkte Edit-Distanz ein geeignetes mathematisches Verfahren, um den Grad der Ähnlichkeit von Nervenzellen zu bestimmen.*

Mit dem bisher Gezeigten, können wir in jedem Fall durch

$$d(cell_1, cell_2) := d^c_{edit}(T_{cell_1}, T_{cell_2})$$

Beschreibung	Klasse
Datenstruktur zur Verwaltung der Sektionen	`Section`
Containerklasse Baum	`Tree`
Datenstruktur zur Verwaltung einer Nervenzelle	`TreeGraph`
Berechnung der eingeschränkten Edit-Distanz	`TreeMatch`
Verwaltung verschiedener lokaler Kostenfunktionen	`LocalCost`
Löser für min-cost-max-flow-Problem	`MCFClass`

Tabelle 4.1: *Übersicht über die verwendeten Klassen.*

ein metrisches Distanzmaß auf der Menge der Nervenzellen definieren. Wir sagen dann, dass eine Zelle $cell_2$ einer Zelle $cell_1$ ähnlicher ist als eine Zelle $cell_3$, falls

$$d(cell_1, cell_2) < d(cell_1, cell_3).$$

Um zu untersuchen, ob umgekehrt dieses Distanzmaß geeignet ist, die etablierten Vorstellungen von Ähnlichkeit zwischen Nervenzellen zu reproduzieren, wurde ein Programm in C++ implementiert, das aus Eingabedaten im hoc-Format die eingeschränkte Edit-Distanz bestimmt. Tabelle **??** gibt eine Übersicht der verwendeten Klassen.

Die Klasse `Section`.

Eine Instanz der Klasse `Section` verwaltet eine Sektion einer Nervenzelle. Sie besitzt eine eindeutige ID und eine Liste *pt3dPoints*, die die Punkte (x_i, y_i, z_i, d_i) der approximierenden Morphologie enthält. Außerdem hat sie eine Liste *ChildsID* mit den IDs der Kindersektionen und die ID des Vorgängers *parent*. Die verschiedenen Attribute, die einer Sektion zugeschrieben werden können, werden in einem Vektor *label* gespeichert.

Die Klasse `Tree`.

Die Klasse `Tree` wurde von Kasper Peeters[1] implementiert und ist unter der GNU General Public License frei verfügbar. Sie ist eine Containerklasse für Bäume, die in Aufbau und Schnittstelle den Containerklassen der STL ähnelt. Es gibt diverse Iteratoren, die die Knoten des Baumes in verschiedenen Reihenfolgen durchlaufen. Es können Knoten und Teilbäume gelöscht oder an beliebigen Stellen eingefügt und strukturspezifische Eigenschaften, wie etwa die Tiefe eines Knotens, ausgegeben werden.

Die Klasse `TreeGraph`.

Zur Verwaltung einer Nervenzelle dient die Klasse `TreeGraph`. Die Methode *Loadhocfile* liest die Daten aus einer hoc-Datei ein und erzeugt für jede Sektion der Zelle ein Instanz

[1] http://www.aei.mpg.de/peekas/tree/

von `Section` und schreibt die Zeiger in eine Liste *sectlist*. Die Methode *makeTree* erzeugt dann daraus einen Baum-Container *tr*, in dessen Knoten Zeiger auf die Sektionen stehen. Gewisse geometrische Attribute einer Zelle, wie etwa der Abstand zum Soma entlang des Dendriten, können nun schnell bestimmt werden. Außerdem können die Sektionen in geeigneter Weise, d.h. in postorder, nummeriert werden. Das Array *tgraph* ermöglicht dann den indizierten Zugriff auf die Sektionen. $tgraph[i]$ ist ein Zeiger auf die Sektion, die in der postorder-Nummerierung die Nummer i hat.

Die Klasse `LocalCost`.
Die verschiedenen lokalen Kostenfunktionen auf den Attributen der Sektionen, die in die Berechnung der eingeschränkten Edit-Distanz eingehen können, sind in der Klasse `LocalCost` implementiert. Diese Klasse hat zwei Referenzen auf Instanzen von `TreeGraph` und damit indizierten Zugriff auf die Sektionen und ihre Attribute.

Die Klasse `MCFClass`.
Die Klasse `MCFClass` ist eine abstrakte, rein virtuelle Basisklasse, die die Schnittstelle zu einem Löser eines Min-Cost-Max-Flow-Problems definiert. Diese Klasse sowie verschiedene Löser wurden von Antonio Frangioni[2] implementiert. In der hier vorgestellten Implementierung wird der Algorithmus von D. Bertsekas [**?**] verwendet.

Die Klasse `TreeMatch`.
Dem Konstruktor der Klasse `TreeMatch` werden zwei Referenzen auf Instanzen von `TreeGraph` übergeben. Die Methode *matching()* ist dann die Implementierung von Algorithmus **??**. Sie berechnet sukzessive die eingeschränkte Edit-Distanz zwischen den Teilbäumen und Teilwäldern und speichert die Ergebnisse in den beiden zwei-dimensionalen Feldern *Dforest* und *Dtree* ab. Dabei wird in jedem Iterationsschritt eine Instanz von MCFClass zur Lösung des Min-Cost-Max-Flow-Problems aufgerufen. Die eingeschränkte Edit-Distanz zwischen den Zellen, die durch die Instanzen von TreeGraph repräsentiert werden, entspricht dem Wert von $DTree[n_1, n_2]$, wobei n_1 und n_2 die Anzahl der Sektionen in den beiden Zellen ist. Wie in Satz **??** gezeigt, ist die Laufzeit sowohl von der Anzahl der Sektionen als auch von der Anzahl der Kinderknoten abhängig. Da bis auf den Wurzelknoten in den hier betrachteten Bäumen alle Knoten nur zwei Kinderknoten haben, sind Laufzeitabschätzungen wie die in Satz **??** zu grob. Numerische Experimente zeigen, dass die Laufzeit in diesem Fall linear mit $|T_1||T_2|$ wächst (Abb. **??**).

[2] http://www.di.unipi.it/di/groups/optimize/Software/MCF.html

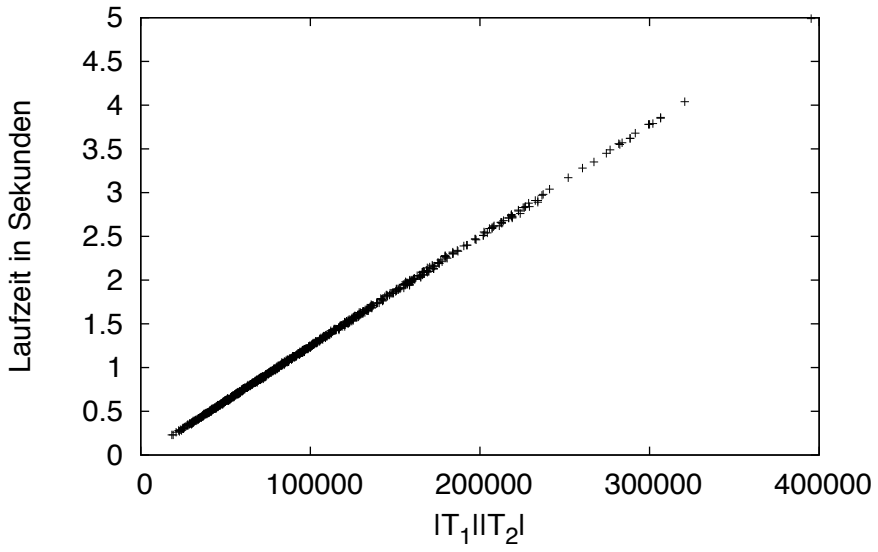


Abbildung 4.2: *Der Graph legt einen linearen Zusammenhang zwischen der Laufzeit der Methode* matching *und dem Produkt der Knotenanzahl nahe.*

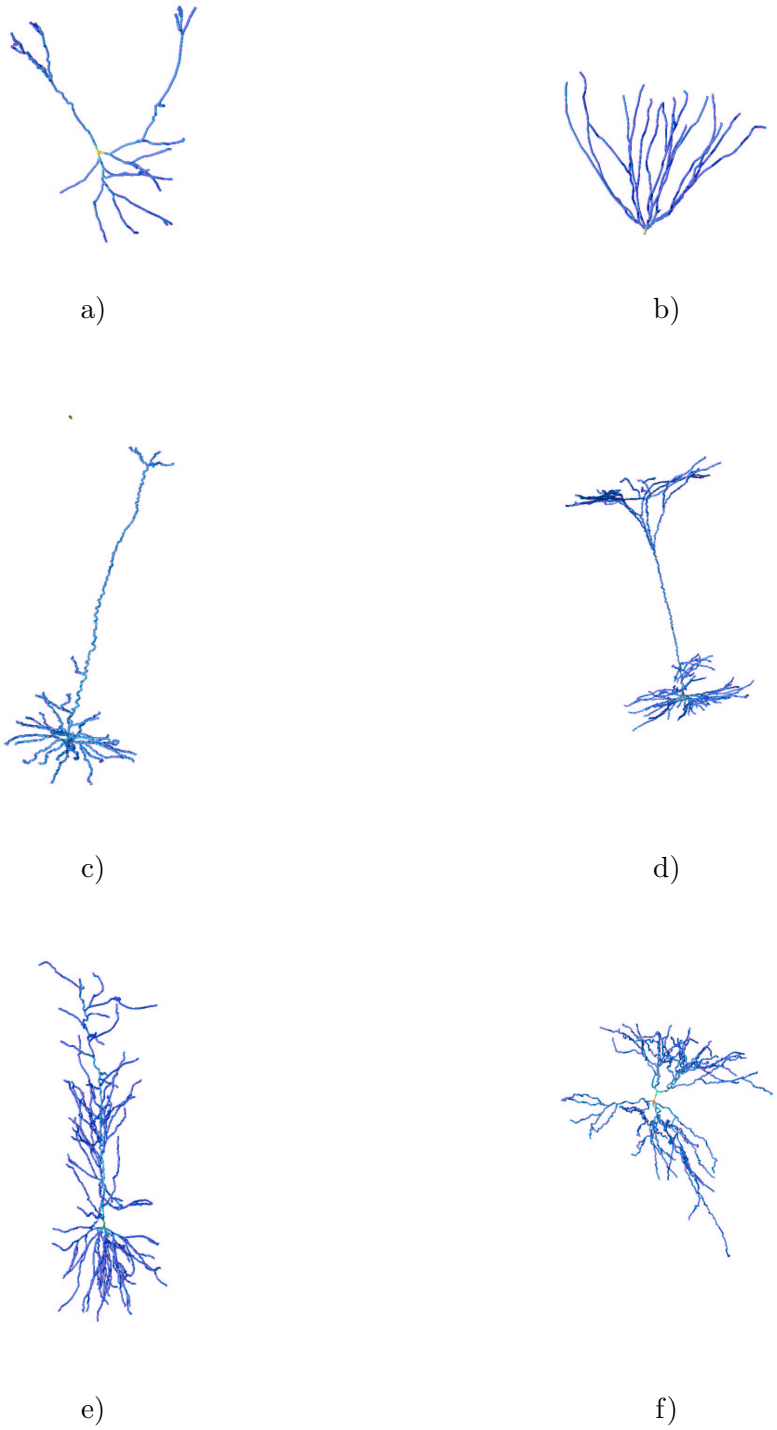
Kapitel 5

Ergebnisse

Wir wollen in diesem Kapitel untersuchen, ob es möglich ist, über die eingeschränkte Edit-Distanz ein Distanzmaß für Neuronen zu definieren, das bereits existierende Vorstellungen von Ähnlichkeit bzw. Unähnlichkeit wiedergeben kann.

5.1 Klassifikation von Neuronen

In den Neurowissenschaften wird zunächst zwischen pyramidalen und nicht pyramidalen Zellen unterschieden. Pyramidenzellen haben einen annähernd dreieckigen Zellkörper. Neben den kleineren *Basaldendriten* an der Basis entspringt an der Spitze des Zellkörpers ein prominenter Dendrit, der sogenannte *Apikaldendrit*. Die sogenannten Interneuronen und Granularzellen sind Nervenzellen, die nicht zur Klasse der Pyramidalzellen gehören (Abb. **??** a) und b)). Innerhalb der Klasse der Pyramidenzellen selbst gibt es eine sehr große Vielfalt an morphologischen Strukturen. Interessant dabei ist die Tatsache, dass Pyramidenzellen, die in demselben anatomischen Bereich des Gehirns liegen, offensichtlich ähnlicher sind als solche aus verschiedenen Bereichen. Die Bereiche, die hierbei betrachtet werden, sind sehr kleine Teilbereiche der anatomischen Bereiche aus Kapitel **??**. Der Cortex beispielsweise wird in sechs übereinanderliegende Schichten unterteilt. Die Pyramidenzellen jeder Schicht sind nach einem festgelegten Muster mit den Zellen in anderen Schichten verbunden und unterscheiden sich teilweise deutlich in der Morphologie. Zusammen bilden sie räumlich kompakte funktionelle Einheiten, sogenannte Kolumnen oder Säulen, die beispielsweise einem einzigen Tasthaar einer Maus zugeordnet werden können. In Abbildung **??** sind charakteristische Morphologien von Zellen verschiedener anatomischer Lokalisierung dargestellt. Beobachtungen dieser Art legen es nahe, dass sich die Klassifizierung von Neuronen nach ihrer Lokalisierung in der morphologischen Struktur wiederspiegelt. Zellen, die also aus demselben anatomischen Bereich des Gehirns kommen, sind sich ähnlicher als solche aus verschiedenen Bereichen. Um diese Behauptung quantifizieren zu können, werden Parameter definiert, die die morphologische Struktur von Neuronen beschreiben. Diese Parameter sollten für Zellen verschiedener Klassen signifikante Unterschiede aufweisen. Die



a) b) c) d) e) f)

Abbildung 5.1: *a) Interneuron und b) Granularzelle aus dem Duke-Southampton-Archiv; c) L5a- und d) L5b-Pyramidenzellen aus der Datenbank CortexDB, L steht dabei für Layer (Schicht); e) Ca1- und f) Ca3- Pyramiden des Hippokampus aus dem Duke-Southampton-Archiv.*

Parameter können sowohl feste Kennzahlen, wie die Anzahl der Verzweigungspunkte einer Zelle als auch Funktionen, etwa die Anzahl der Verzweigungspunkte mit Abstand d zum Soma, sein. Uylings und Pelt geben in [**?**] eine Übersicht über morpholgische Parameter, die bisher in den Neurowissenschaften verwendet werden. Dazu zählen neben offensichtlichen Parametern, wie der Anzahl der Sektionen und der Membranfläche, auch Parameter, die eine Aussage über die Asymmetrie des dendritischen Baumes machen oder die fraktale Dimension der Morphologie beschreiben. Soll also gezeigt werden, dass zwei verschiedene Populationen von Zellen tatsächlich zwei verschiedene Klassen sind, muss die Verteilung der morphologischen Parameter statistisch analysiert werden. Wird kein Unterschied gefunden, so kann keine Aussage über die Klasseneinteilung getroffen werden. Es ist möglich, dass lediglich ein geeigneter Parameter definiert werden muss, um einen signifikanten Unterschied zu erhalten.

5.2 Daten

Wir wollen nun zeigen, dass die eingeschränkte Edit-Distanz ein geeignetes Maß ist, um bekannte morphologische Klassen zu unterscheiden. Dazu müssen sowohl genügend umfangreiche Daten (Zellen im hoc-Format) vorhanden sein als auch die konkrete Einteilung dieser Zellen in morphologische Klassen bekannt sein. Wir konzentrieren uns daher auf Neuronen aus dem Hippokampus und dem Cortex.

Hippokampus Cannon [**?**] analysiert die Verteilungen von 32 verschiedenen Parametern für Interneuronen Granularzellen, Ca1- und Ca3-Pyramidenzellen aus dem Hippokampus und kann zeigen, dass Interneuronen, Granularzellen und die Pyramidenzellen eigene morphologische Klassen sind. Innerhalb der Klasse der Pyramidenzellen selbst konnte er keinen signifikanten Unterschied zwischen den Ca1- und Ca3-Pyramiden erkennen. Die Morphologien der Interneuronen und der Pyramidenzellen, die in dieser Arbeit verwendet werden, liegen im swc-Format im frei zugänglichen Duke-Southampton-Archiv[1] vor. Zur Konvertierung vom swc- in das benötigte hoc-Format wurde das Programm cvapp[2] verwendet.

Cortex Wie bereits erwähnt, wird der Cortex in 6 Schichten unterteilt und die Nervenzellen werden nach der Lage ihrer Somata in diesen Schichten klassifiziert. Die Zellen jeder Schicht besitzen eine charakteristische Funktionalität und sind über Synapsen mit Zellen anderer Schichten verbunden. Dies legt den Schluß nahe, dass sich die Zellen der verschiedenen Schichten auch in ihrer Morphologie unterscheiden müssen. Da aber die Funktionalität der Zellen in einer kortikalen Kolumne noch nicht vollständig verstanden ist, ist die Klassifizierung der Zellen nach Schichten nicht endgültig. Beispielsweise unterscheidet man innerhalb Schicht 5 noch einmal zwischen L5a- und L5b-Pyramidenzellen ([**?**]), während die Zellen in Schicht 2 und 3 zu den L2/3-Pyramidenzellen zusammengefasst werden. Obwohl in verschiedenen Arbeiten die Verteilung morphologischer Parameter für

[1] http://neuron.duke.edu/cells/
[2] http://compneuro.org/CDROM/nmorph/download.html

Zellen der verschieden Schichten publiziert wurden, konnte keine Arbeit gefunden werden, die wie im Falle des Hippokampus die Zellen dieser Klassen auf signifikante Unterschiede in morphologischen Parametern untersucht. Der Umfang der verfügbaren Zellmorphologien beschränkt sich auf 10 Zellen aus der Datenbank *CortexDB*, die in Zusammenarbeit mit dem MPIMF[3] in der Arbeitsgruppe SIT[4] entwickelt wurde und 29 Zellmorphologien aus Schicht 5, die Andreas Schäfer [**?**] zur Verfügung gestellt hat.

In Tabelle **??** sind die verfügbaren Morphologien zusammengefasst. Neben den Zellen aus dem Hippokampus, die Cannon für seine Analyse verwendet, sind im Duke-Southampton-Archiv weitere Morphologien veröffentlicht, die sich von den analysierten durch ihr Alter oder die Präparierung unterscheiden. Desweiteren können mit dem Programm NeuGen ([**?**]) beliebig viele nicht identische Morphologien verschiedener Zellklassen des Cortex generiert werden. Der Algorithmus greift dabei auf publizierte morphologische Parameter zurück. Erste Evaluationen und der visuelle Vergleich zeigen, dass die generierten Morphologien den echten stark ähneln.

5.3 Analyse von Distanzmatrizen

Die Klassifikation von Objekten aufgrund ihrer metrischen Distanzen ist ein klassisches Problem in der multivariaten statistischen Analyse. Zunächst treten solche Fragestellungen bei Objekten auf, die eine Darstellung als Vektoren $\mathbf{x} \in \mathbb{R}^n$ besitzen. Die Abstände dieser Objekte sind dann durch eine der bekannten Metriken im $\mathbb{R}^n$ gegeben und werden in einer sogenannten Distanzmatrix $D \in \mathbb{R}^{n \times n}$ zusammengefasst. Der Eintrag d_{ij} entspricht dem Abstand von Objekt i und j (vgl. Definition **??**). In unserem Fall ist die Distanzmatrix D durch die eingeschränkte Edit-Distanz gegeben:

$$d_{ij} = d^c_{edit}(T_{cell_i}, T_{cell_j}).$$

Mit statistischen Verfahren wie der *multidimensionalen Skalierung* und der *Clusteranalyse* [**?**] kann daraus eine Klassifizierung der Objekte bestimmt werden.

5.3.1 Multidimensionale Skalierung

Definition 5.1 (euklidische Distanzmatrix)**.** *Eine Distanzmatrix $D \in \mathbb{R}^{n \times n}$ heißt euklidische Distanzmatrix, falls es Vektoren $\boldsymbol{x}_1, \ldots \boldsymbol{x}_n$, $\boldsymbol{x}_i \in \mathbb{R}^p$ gibt, so dass*

$$d^2_{ij} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T (\boldsymbol{x}_i - \boldsymbol{x}_j)$$

Der Vorteil euklidischer Distanzmatrizen liegt darin, dass abstrakten Objekten, für die nur die Abstände untereinander bekannt sind, Vektoren zugeordnet werden können. Mittels

[3] Max-Plank-Institut für medizinische Forschung in Heidelberg
[4] Simulation in Technology, IWR Heidelberg

Quelle	Zelltypen	Anzahl
Neuronen aus der CortexDB einer Datenbank von MPIMF und SIT Simulation in Technology, Heidelberg	L2/3-Pyramiden	2
	L5a-Pyramiden	3
	L5b-Pyramiden	3
	L4-Spiny-Stellate	2
pyramidale Neuronen aus [?]	L5-Pyramiden	29
Neuronen aus dem Duke-Southampton-Archive http://neuron.duke.edu/cells/	Ca1-Pyramiden	52
	Ca3-Pyramiden	16
	Granularzellen	35
	Interneuronen	13
Zellen der Analyse von Cannon[?]	Ca1-Pyramiden	22
	Ca3-Pyramiden	16
	Granularzellen	18
	Interneuronen	13
künstliche Neuronen, generiert mit NeuGen http://neugen.uni-hd.de	L2/3-Pyramiden	50
	L5a-Pyramiden	50
	L5b-Pyramiden	50
	L4-Spiny-Stellate	50
	L4-Star-Pyramiden	50

Tabelle 5.1: *Übersicht über die verfügbaren Morphologien.*

Hauptkomponentenanalyse können diese Vektoren dann beispielsweise auf den $\mathbb{R}^2$ projiziert werden, um die relative Lage der Objekte zueinander zu visualisieren. Der folgende Satz gibt ein Kriterium, wann eine Distanzmatrix euklidisch ist.

Satz 5.2 (Multidimensionale Skalierung I)**.** *Sei $D \in \mathbb{R}^{n\times n}$ eine Distanzmatrix, $A \in \mathbb{R}^{n\times n}$ mit $a_{ij} = -\frac{1}{2}d_{ij}^2$, $I \in \mathbb{R}^{n\times n}$ die Einheitsmatrix und $E_n \in \mathbb{R}^{n\times n}$ die Matrix, deren Einträge alle eins sind. D ist genau dann euklidisch, falls die doppelt zentrierte Matrix*

$$B = (I - \frac{1}{n}E_n)A(I - \frac{1}{n}E_n)$$

positiv semidefinit ist.

Ist die doppelt zentrierte Matrix B indefinit, so besagt der folgende Satz, dass eine Konstante $c \in \mathbb{R}$ gefunden werden kann, so dass die doppelt zentrierte Matrix B_c der Distanzmatrix $D_c = D + cE_n$ positiv semidefinit ist.

Satz 5.3 (Multidimensionale Skalierung II)**.** *Ist die Matrix B aus Satz* **??** *nicht positiv semidefinit, d.h. es existiert ein Eigenwert $\lambda \leq 0$, und sei μ der kleinste Eigenwert der Matrix $\frac{-1}{2}(I - \frac{1}{n}E_n)D(I - \frac{1}{n}E_n)$, dann ist die doppelt zentrierte Matrix B_c der Distanzmatrix $D_c = D + cE_n$ positiv semidefinit für alle $c \in \mathbb{R}$ mit*

$$c \geq \sqrt{4\mu^2 - 2\lambda} - 2\mu.$$

Wir wollen den Beweis der letzen beiden Sätze hier nicht vorführen, sondern lediglich auf [**?**] verweisen. Interessant ist für uns die Berechnung der Vektoren $\mathbf{x}_i$. Nehmen wir also an, dass die doppelt zentrierte Matrix B positiv definit ist. Dann gibt es eine unitäre Matrix $R \in \mathbb{R}^{n\times n}$ und eine Diagonalmatrix $\Lambda \in \mathbb{R}^{n\times n}$, deren Diagonaleinträge λ_i, mit $\lambda_i \geq \lambda_j$ für $i \geq j$, die Eigenwerte von B sind, so dass $B = R\Lambda R^T$. Definieren wir nun

$$\mathbf{x}_i^T = (R\Lambda^{\frac{1}{2}})_i$$

über die i-te Zeile der Matrix $R\Lambda^{\frac{1}{2}}$, so gilt für den euklidischen Abstand von $\mathbf{x}_i$ und $\mathbf{x}_j$:

$$\begin{aligned}(d_2(\mathbf{x}_i, \mathbf{x}_j))^2 &= (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)\\ &= \mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T\mathbf{x}_j - \mathbf{x}_j^T\mathbf{x}_i + \mathbf{x}_j^T\mathbf{x}_j\\ &= b_{ii} - b_{ij} - b_{ji} + b_{jj}\\ &= d_{ij}^2\end{aligned}$$

Die ersten beiden Komponenten der Vektoren $\mathbf{x}_i$ entsprechen den beiden Hauptkomponenten [**?**] und liefern eine graphische Darstellung der Objekte, die die tatsächliche Anordnung in einem höherdimensionalen Vektorraum umso besser darstellt, je kleiner die vernachlässigten Komponenten $r_{i3}\sqrt{\lambda_3}, \ldots r_{in}\sqrt{\lambda_n}$ sind. Die Differenz zwischen dem exakten

Abstand d_{ij} und dem dargestellten Abstand kann nämlich durch den drittgrößten Eigenwert λ_3 abgeschätzt werden, wie aus der folgenden Abschätzung für $q = 2$ hervorgeht [**?**].

$$\begin{aligned}
0 &\leq d_{ij} - \sqrt{\sum_{k=1}^{q} \lambda_k (r_{ik} - r_{jk})^2} \\
&\leq \sqrt{\sum_{k=q+1}^{n} \lambda_k (r_{ik} - r_{jk})^2} \\
&\leq \sqrt{\sum_{k=q+1}^{n} \lambda_k r_{ik}} + \sqrt{\sum_{k=q+1}^{n} \lambda_k r_{jk}} \\
&\leq \sqrt{\lambda_{q+1}} \left(\sqrt{\sum_{k=q+1}^{n} r_{ik}^2} + \sqrt{\sum_{k=q+1}^{n} r_{jk}^2} \right) \\
&\leq \sqrt{\lambda_{q+1}} \left(\sqrt{\sum_{k=1}^{n} r_{ik}^2} + \sqrt{\sum_{k=1}^{n} r_{jk}^2} \right) \\
&\leq 2\sqrt{\lambda_{q+1}}
\end{aligned}$$

Durch die multidimensionale Skalierung kann also ein erster visueller Eindruck der Verteilung der Objekte gewonnen werden. Dabei sollte jedoch beachtet werden, dass bei einer großen Anzahl an Objekten die Reduktion auf zwei Komponenten zu groben Fehlern in der Darstellung führen kann. Interessant sind noch solche Fälle, in denen die doppelt zentrierte Matrix mehrere Eigenwerte λ_i, mit $\lambda_i = 0$ besitzt, da dann die n Objekte eine Darstellung in einem Vektorraum besitzen, dessen Dimension kleiner als n ist.

5.3.2 Cluster-Analyse

Eine Partitionierung einer Indexmenge $\mathbb{I}$ ist eine Menge $P_k = \{I_1, \ldots I_k\}$ von disjunkten Teilmengen $I_j \in \mathbb{I}$, deren Vereinigung die gesamte Objektmenge $\mathbb{I}$ ist. Die Teilmengen I_k werden auch Cluster genannt. In der Cluster-Analyse wird nun versucht, eine Menge von Objekten anhand ihrer Distanzmatrix zu partitionieren. Die Objekte eines Clusters bilden eine Klasse innerhalb der Menge aller Objekte. Es gibt diverse heuristische Algorithmen, die eine Partitionierung erzeugen. Wir wollen in unseren Analysen zwei verschiedene Algorithmen verwenden.

Partitioning around Medoids

Der Partitionierungsalgorithmus *Partitioning around Medoids (PAM)* [**?**] bestimmt zu einer vorgegebenen Anzahl an k Clustern eine Partionierung der Objektmenge. Im Initialsierungsschritt werden k Objekte, die Medoide, zufällig ausgewählt und die anderen

Objekte dem Medoiden zugeordnet, der ihnen am nächsten liegt. Ziel des Algorithmus ist die Bestimmung von k repräsentativen Medoiden, so dass die Summe der Abstände der Objekte zu ihren Medoiden minimal wird. Dazu wird im Iterationsschritt ein weiteres Objekt zufällig bestimmt und geprüft, ob der Tausch dieses Objektes mit einem aktuellen Medoid den Wert der Zielfunktion verkleinert.

Hierarchische Cluster-Analyse

In der *agglomerativen hierarchischen Cluster-Analyse (HCA)* werden ausgehend von der feinsten Partitionierung, d.h. $P = \{\{1\}, \ldots \{n\}\}$, sukzessive gröbere Partionierungen bestimmt, indem die zwei Cluster, deren Abstand minimal ist, vereinigt werden. Im letzen Iterationsschritt werden alle Objekte zu einem einzigen Cluster zusammengefasst. Die verschiedenen Algorithmen der HCA variieren in der Definition des Abstandes zweier Cluster. Die Single-Link-Methode beispielsweise definiert den Abstand $d(C_1, C_2)$ zweier Cluster C_1 und C_2 als den minimalen Abstand zweier Objekte aus C_1 und C_2. Entscheidend für eine schnelle Berechnung ist die Lance-Williams-Formel

$$\begin{aligned} d(C_1 \cup C_2, Q) &= \alpha_1 * d(C_1, Q) \\ &\quad +\alpha_2 * d(C_2, Q) \\ &\quad +\beta * d(C_1, C_2) \\ &\quad +\gamma * |d(C_1, Q) - d(C_2, Q)|, \end{aligned}$$

mit der der Abstand der vereinigten Menge $C_1 \cup C_2$ zu jeder anderen Menge Q aus den Abständen $d(C_1, Q)$, $d(C_2, Q)$ und $d(C_2, Q)$ bestimmt werden kann. Die verschiedenen Methoden unterscheiden sich in der Wahl der Parameter α_1, α_2, β und γ (Abb. **??**), die auch von den Kardinalitäten von C_1, C_2 und Q abhängen können. Das Ergebnis der hierarchischen Cluster-Analyse wird in einem Dendrogramm visualisiert. Ein Dendrogramm ist ein vollständiger binärer Baum mit n Blättern, die die n Objekte repräsentieren. Jeder innere Knoten stellt einen Cluster dar, der die Objekte enthält, die in den Blättern darüber stehen. Außerdem werden die inneren Knoten in einem Graphen so aufgetragen, dass ihre y-Komponente dem Abstand der Cluster an den beiden Kinderknoten entspricht (Abb. **??**). Genauere Betrachtungen der verschiedenen agglomerativen Verfahren zeigen, welche charakteristische Form die Cluster der jeweiligen Partitionierung haben bzw. bei welcher Gruppierung der Objekte die Verfahren versagen. Da wir in dem Klassifizierungsproblem von Nervenzellen keine Aussage über die genaue Form der Klassen machen können, wollen wir uns damit begnügen zu zeigen, dass eine bestimmte agglomerative Methode, die Methode von Ward [**?**], anhand einer Distanzmatrix Zellen nach ihrer morphologischen Klasse gruppiert. Dies impliziert die Annahme, dass die Zellen innerhalb der einzelnen Klassen normalverteilt sind [**?**].

Sämtliche statistischen Analysen von Distanzmatrizen wurden mit dem Statistik-Programm R[5] durchgeführt. Die Funktionen *mdscale*(), *pam*() und *hclust*() sind die Implementie-

[5] http://cran.r-project.org

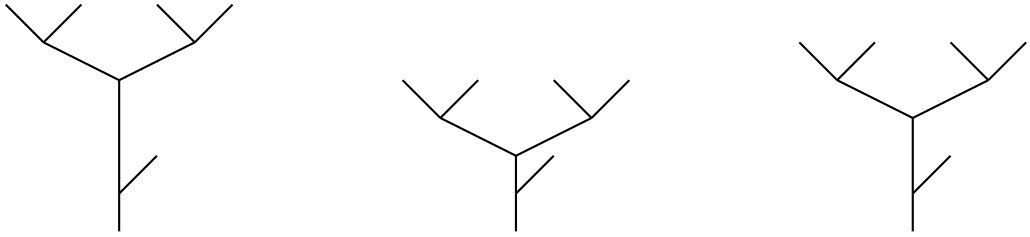
Verfahren	α_1	α_2	β	γ
Single Link.	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete Link.	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Average Link. 1	$\frac{1}{2}$	$\frac{1}{2}$	0	0
Average Link. 2	$\frac{\lvert C_1\rvert}{\lvert C_1\rvert+\lvert C_2\rvert}$	$\frac{\lvert C_2\rvert}{\lvert C_1\rvert+\lvert C_2\rvert}$	0	$\frac{-\lvert C_2\rvert\lvert C_2\rvert}{(\lvert C_1\rvert+\lvert C_2\rvert)^2}$
Zentroid	$\frac{\lvert C_1\rvert}{\lvert C_1\rvert+\lvert C_2\rvert}$	$\frac{\lvert C_2\rvert}{\lvert C_1\rvert+\lvert C_2\rvert}$	$\frac{\lvert C_2\rvert}{\lvert C_1\rvert+\lvert C_2\rvert}$	0
Ward	$\frac{\lvert C_1\rvert+\lvert Q\rvert}{\lvert C_1\rvert+\lvert C_2\rvert+\lvert Q\rvert}$	$\frac{\lvert C_2\rvert+\lvert Q\rvert}{\lvert C_1\rvert+\lvert C_2\rvert+\lvert Q\rvert}$	$\frac{-\lvert Q\rvert}{\lvert C_1\rvert+\lvert C_2\rvert+\lvert Q\rvert}$	0
Median	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0

Tabelle 5.2: *Die Parameter verschiedener agglomerativer Cluster-Verfahren.*

rungen der multidimensionalen Analyse, des PAM-Algorithmus und der hierarchischen Clusteranalyse.

5.4 Wahl der Attribute

Wir haben in Kapitel **??** gezeigt, dass wir Nervenzellen in einen knotengewichteten Baum überführen und dann eine metrische Distanz zweier Nervenzellen durch die eingeschränkte Edit-Distanz ihrer Baumdarstellungen definieren können. Während die topologische Struktur dieses Baumes durch die Zelle vollständig bestimmt ist, kann durch die Wahl der Attribute die intuitive Vorstellung von Ähnlichkeit in die Abstandsbestimmung mit eingehen. Wählt man für alle Knoten das Attribut 1 so spielt die geometrische Form der Sektionen keine Rolle. Zellen sind dann ähnlich, sobald ihre Topologie ähnlich ist. Sollen darüber hinaus ähnliche Zellen eine ähnliche räumliche Ausdehnung haben, so muss die Länge der Sektionen oder ihr Abstand zum Soma in die Definition der Attribute eingehen (Abb. **??**). Grundsätzlich ist es möglich, dass die Attribute, die bei der Bestimmung der eingeschränkten Edit-Distanz berücksichtigt werden aus mehreren Komponenten bestehen. Es sollte dann jedoch eine Standardisierung durchgeführt werden [**?**], damit alle Komponenten der Attribute in etwa die gleiche Größenordnung haben und keine Komponente den Wert der lokalen Kostenfunktion dominiert. Dazu werden zunächst die Mittelwerte



a) b) c)

Abbildung 5.2: *Wahl der Attribute; reduziert auf ihre topologische Struktur sind die drei Bäume gleich, wird jedoch die Länge der Sektionen berücksichtige, so ähnelt b) stärker c) als a).*

und Standardabweichungen für jede Komponente bestimmt

$$\begin{aligned} \mu_i &= \frac{1}{|T_1| + |T_2|} \sum_{v_k \in T_1 \cup T_2} label_i(v_k) \\ \sigma_i &= \sqrt{\frac{1}{|T_1| + |T_2|} \sum_{v_k \in T_1 \cup T_2} (label_i(v_k) - \mu_i)^2} \end{aligned}$$

und danach die standardisierten Attribute

$$slabel_i(v_k) = \frac{label_i(v_k) - \mu_i}{\sigma_i}$$

definiert. Es ist zu beachten, dass die standardisierten Attribute negative Werte annehmen können, auch wenn die nicht standardisierten immer positiv sind. Damit ist aber die in Kapitel **??** beschriebene anschauliche Interpretation von Edit-Operationen als Änderung geometrischer Eigenschaften hinfällig und es wird schwieriger, konkrete Vorstellungen von lokaler Ähnlichkeit in die Wahl geeigneter Attribute umzusetzen. Ein weiterer Nachteil dieser Standardisierung besteht darin, dass beim Vergleich mehrerer Zellen die Mittelwerte μ_i und die Varianzen σ_i über die Knoten aller Zellen bestimmt werden müssen, um die Abstände zwischen den Zellen vergleichen zu können. Betrachtet man also nur eine weitere Zelle, so müssen die Abstände zwischen allen Zellen neu bestimmt werden. Wir wollen zunächst nur untersuchen, ob es geeignete Attribute gibt, die zu einer eingeschränkten Edit-Distanz führen, die Ähnlichkeit und Unähnlichleit von Nervenzellen wiederspiegelt und beschränken uns daher auf solche lokalen Kostenfunktionen, die auf einzelnen Komponenten operieren. Wir bezeichnen dann mit $d^c_{edit,i}$ die eingeschränkte Edit-Distanz, die auf der i-ten Komponente der Knotenattribute basiert. Im Anschluß daran versuchen wir, durch die Kombination der verschiedenen Attribute die Aussagekraft der eingeschränkten Edit-Distanz zu verbessern.

In dieser Arbeit wurde der Einfluß von 21 verschiedenen Attributen auf die Abstandsbestimmung von Nervenzellen untersucht.

5.4.1 Topologische Attribute

Zwei der untersuchten Attribute basieren lediglich auf Kennzahlen der topologischen Baumstruktur:

$$\begin{aligned} label_1(t[i]) &= 1 \\ label_2(t[i]) &= \frac{1}{|T|} \end{aligned}$$

Bei der Verwendung von $label_1$ ist eine eingeschränkte Edit-Distanz durch die Anzahl der Knoten in beiden Bäumen beschränkt. Bezeichnet M die Spur von zwei Bäumen T_1 und T_2 und D_1 bzw. D_2 die Anzahl der Knoten aus T_1 und T_2, die nicht in die Spur eingehen, so gilt nach der Definition des Gewichtes $\Gamma(M)$ für die Distanz

$$d^c_{edit,1}(T_1, T_2) = D_1 + D_2 \leq |T_1| + |T_2|.$$

Aus der Definition der Spur folgt

$$|T_1| - D_1 = |T_2| - D_2$$

und damit

$$D_2 = D_1 + |T_2| - |T_1|.$$

Die eingeschränkte Edit-Distanz modifiziert also in diesem Fall das Distanzmaß, das auf der Differenz der Anzahl der Sektionen beruht

$$d^c_{edit,1}(T_1, T_2) = 2D_1 + |T_2| - |T_1|.$$

Die Verwendung von $label_2$ anstatt $label_1$ führt zu einer eingeschränkten Edit-Distanz, die für beliebige Bäume nach oben beschränkt ist.

$$d^c_{edit,2}(T_1, T_2) \leq \frac{D_1}{|T_1|} + \frac{D_2}{|T_2|} \leq 1 + 1$$

Im Gegensatz zu $label_1$ wird durch die Wahl von $label_2$ den einzelnen Knoten in größeren Bäumen weniger Gewicht beigemessen als in kleineren. Hat ein Baum sehr viele Knoten, so ist ein einzelner Knoten nicht so entscheidend wie in einem Baum mit sehr wenigen Knoten.

5.4.2 Geometrische Attribute

Die geometrischen Attribute unterteilen sich in solche, die auf Längen-, Volumen- oder Oberflächeneigenschaften der Sektionen basieren. Während die Längenattribute Ähnlichkeit in der räumlichen Ausbreitung modellieren sollen, basiert die Definition von Oberflächenattributen auf der Idee, elektrophysiologische Ähnlichkeit zu modellieren, da die Membranoberfläche entscheidend zur Signalausbreitung beiträgt. Die genaue Definition verschiedener Attribute soll hier am Beispiel der Längeneigenschaft diskutiert werden. Neben der Länge der Sektion

$$label_3(t[i]) = lenght(t[i])$$

wurden der Abstand zum Soma und die Gesamtlänge des Dendriten oberhalb der Sektion

$$\begin{aligned} label_4(t[i]) &= LengthToSoma(t[i]) \\ label_5(t[i]) &= lenght(T[i]) \end{aligned}$$

als Attribute gewählt. Die klare Trennung von lokaler und globaler Ähnlichkeit geht damit verloren. Analog zu der Normierung der topologischen Distanz erhalten wir eine eingeschränkte Edit-Distanz, die nach oben durch zwei beschränkt ist, wenn wir die Länge jeder Sektion mit dem Kehrwert der Gesamtlänge des Baumes multiplizieren. Gewichten wir die beiden anderen Längenattribute entsprechend, so ist die eingeschränkte Edit-Distanz zweier Bäume T_1 und T_2 durch die Summe der Sektionen $|T_1| + |T_2|$ beschränkt.

$$\begin{aligned} label_6(t[i]) &= \frac{length(t[i])}{length(T)} \\ label_7(t[i]) &= \frac{LengthToSoma(t[i])}{length(T)} \\ label_8(t[i]) &= \frac{length(T[i])}{length(T)} \end{aligned}$$

In Anlehnung an eine von Cannon [**?**] untersuchte morphologische Kennzahl, $\frac{Gesamtvolumen}{Gesamtflache}$, die signifikante Unterschiede zwischen Interneuronen und Granularzellen aufweist, wollen wir noch

$$label_{21}(t[i]) = \frac{volume(t[i])}{surface(T)}$$

untersuchen. Ein weiteres Attribut berücksichtigt die räumliche Orientierung.

$$label_{22}(t[i]) = angle(children(t[i]))$$

$label_{22}$ ist der summierte und über die Anzahl der Kinderknoten gemittelte Winkel zwischen den Kindersektionen.

In Tabelle **??** sind noch einmal alle Attribute zusammengefasst.

k	$label_k(t[i])$	
1	1	Topologie
2	$\frac{1}{\|T\|}$	
3	$lenght(t[i])$	Länge
4	$LengthToSoma(t[i])$	
5	$lenght(T[i])$	
6	$\frac{length(t[i])}{length(T)}$	
7	$\frac{LengthToSoma(t[i])}{length(T)}$	
8	$\frac{length(T[i])}{length(T)}$	
9	$volume(t[i])$	Volumen
10	$VolumeToSoma(t[i])$	
11	$volume(T[i])$	
12	$\frac{volume(t[i])}{volume(T)}$	
13	$\frac{VolumeToSoma(t[i])}{volume(T)}$	
14	$\frac{volume(T[i])}{volume(T)}$	
15	$surface(t[i])$	Oberfläche
16	$SurfaceToSoma(t[i])$	
17	$surface(T[i])$	
18	$\frac{surface(t[i])}{surface(T)}$	
19	$\frac{SurfaceToSoma(t[i])}{surface(T)}$	
20	$\frac{surface(T[i])}{surface(T)}$	
21	$\frac{volume(t[i])}{surface(T)}$	
22	$angle(children(t[i]))$	Winkel

Tabelle 5.3: *Übersicht der untersuchten Attribute.*

5.5 Ergebnisse

Wir wollen nun untersuchen, welche der Attribute geeignet sind, morphologische Zellklassen zu erkennen. Dazu betrachten wir n_a Zellen einer Klasse A und n_b Zellen einer Klasse B und bestimmen dann die eingeschränkte Edit-Distanz für alle Paare von Zellen und alle 22 Attribute. Wir erhalten damit 22 verschiedene Distanzmatrizen $D^i \in \mathbb{R}^{n_a \times n_b}, 1 \leq i \leq 22$, die die Abstände der Zellen zueinander unter der Berücksichtigung verschiedener Attribute enthalten. Um zu überprüfen, ob die verschiedenen Distanzmatrizen die Klasseneinteilung widerspiegeln, wenden wir die beschriebenen statistischen Analyseverfahren Clusteranalyse und multidimensionale Skalierung an. Während der Clusteralgorithmus PAM die Gesamtmenge der Zellen in zwei Teilmengen C_1^{pam} und C_2^{pam} partitioniert, generiert der agglomerative Algorithmus WARD eine Folge von Partitionierungen, von denen wir nur diejenige betrachten, die aus zwei Partitionen C_1^{ward} und C_2^{ward} besteht. Wir untersuchen dann, wie gut diese Partitionierungen mit den vorgegebenen Klassen übereinstimmen. Dazu definieren wir den Partitionierungsfehler Δ_{abs} als den Anteil der falsch zugeordneten Zellen:

$$\Delta_{abs} = min\left\{|A \cap C_1| + |B \cap C_2|, |A \cap C_2| + |B \cap C_1|\right\}.$$

Da

$$\Delta_{abs} = min\left\{|A \cap C_1| + |B \cap C_2|, |A| - |A \cap C_1| + |B| - |B \cap C_2|\right\}.$$

ist der abslute Fehler immer kleiner als $\frac{|A|+|B|}{2}$. Wir definieren daher den relativen Partitionierngsfehler Δ folgendermaßen:

$$\Delta = \frac{2\,\Delta_{abs}}{|A| + |B|}.$$

Bei einem Partitionierungsfehler von $\Delta = 0.5$ sind dann ein Viertel aller Zellen falsch zugeordnet. Die multidimensionale Skalierung wollen wir lediglich zur Visualisierung der Verteilung der Zellen verwenden.

5.5.1 Hippokampus

Pyramidenzellen und „Nicht-Pyramidenzellen “

Als erstes wollen wir zeigen, dass wir pyramidale von nicht pyramidalen Zellen unterscheiden können. Dazu fassen wir die 52 Ca1-Pyramidenzellen und die 16 Ca3-Pyramidenzellen aus dem Duke-Southampton-Archiv in einer Population A und die 13 Interneuronen und die 35 Granularzellen in der Population B zusammen, bestimmen den Abstand von jeder Zelle mit jeder anderen über die eingeschränkten Edit-Distanzen $d^c_{edit,i}$ und untersuchen dann, ob ein Clusteralgorithmus aus den Distanzmatrizen D^i die Partitionierung in pyramidale und nicht pyramidale Zellen reproduzieren kann. In den Tabellen **??** und **??** sind die Partitionierungen und der Partitionierungsfehler der Clusteralgorithmen PAM und WARD dargestellt. Die Algorithmus PAM kann aus den Distanzmatrizen D^1, D^2, D^6, D^{12}, D^{18}

Attribut	$\|A \cap C_1\|$	$\|B \cap C_2\|$	$\|A \cap C_2\|$	$\|B \cap C_1\|$	Δ
1	57	48	11	0	0.19
2	67	48	1	0	0.02
3	27	0	41	48	0.47
4	52	48	16	0	0.28
5	22	0	46	48	0.38
6	68	48	0	0	0.00
7	66	48	2	0	0.03
8	26	0	42	48	0.45
9	58	0	10	48	1
10	62	0	6	48	0.93
11	50	0	18	48	0.86
12	65	48	3	0	0.05
13	39	48	29	0	0.50
14	23	0	45	48	0.40
15	34	0	34	48	0.59
16	37	48	31	0	0.53
17	46	0	22	48	0.80
18	67	48	1	0	0.02
19	37	48	31	0	0.53
20	24	0	44	48	0.41
21	24	0	44	48	0.41
22	64	48	4	0	0.07

Tabelle 5.4: *Partitionierung von PAM anhand der verschiedenen Distanzmatrizen. Die Menge A sind 68 Pyramidenzellen. Die Menge B sind 48 Interneuronen und Granularzellen.*

und D^{22} die urpsprüngliche Partitionierung gut ($\Delta \leq 0.2$) reproduzieren. WARD dagegen erzielt mit deutlich mehr Distanzmatrizen gute Ergebnisse. Lediglich bei D^8, D^9, D^{10}, D^{11},D^{14}, D^{16}, D^{17}, D^{20} und D^{21} ist der Fehler größer als 0.2. In beiden Fällen liefern bereits die beiden rein topologischen Varianten $d^c_{edit,1}$ und $d^c_{edit,2}$ sehr gute Ergebnisse. Die besten Ergebnisse liefert die Verwendung der Attribute $label_6$ und $label_{18}$, die den Sektionen den relativen Anteil an der Gesamtlänge bzw. dem Gesamtvolumen zuschreiben. Die Distanz $d^c_{edit,22}$, die den Winkel zwischen den Kindersektionen berücksichtigt liefert ebenfalls gute Ergebnisse.

Nun wollen wir untersuchen, wie gut die vier verschiedenen Zellklassen der Ca1-Pyramiden, Ca3-Pyramiden, Interneuronen und Granularzellen aufgrund der Distanzmatrizen getrennt werden können. Wir geben im Folgenden in einer kompakteren Darstellung nur noch die Größenordnung des Partitionierungsfehlers an. Wir tragen die vier verschiedenen Namen der Zellklassen an den x- und y-Achsen eines Graphen an und stellen den Partitionierungs-

Attribut	$\lvert A \cap C_1 \rvert$	$\lvert B \cap C_2 \rvert$	$\lvert A \cap C_2 \rvert$	$\lvert B \cap C_1 \rvert$	Δ
1	67	48	1	0	0.02
2	67	48	1	0	0.02
3	67	48	1	0	0.02
4	68	48	0	0	0.00
5	66	48	2	0	0.03
6	67	48	1	0	0.02
7	66	47	2	1	0.05
8	25	0	43	48	0.43
9	53	0	15	48	0.91
10	61	0	7	48	0.94
11	53	0	15	48	0.91
12	67	47	1	1	0.03
13	60	48	8	0	0.14
14	20	0	48	48	0.35
15	68	48	0	0	0.00
16	53	0	15	48	0.91
17	50	0	18	48	0.86
18	68	48	0	0	0.00
19	68	47	0	1	0.02
20	21	0	47	48	0.36
21	24	0	44	48	0.41
22	67	48	1	0	0.02

Tabelle 5.5: *Partitionierung von WARD anhand der verschiedenen Distanzmatrizen. Die Menge A sind 68 Pyramidenzellen. Die Menge B sind 48 Interneuronen und Granularzellen.*

fehler beim Vergleich von Klasse X und Y durch den Grauwert eines Quadrates im Punkt (X,Y) dar. Je dunkler die Quadrate eingefärbt sind, umso kleiner ist der Partitionierungsfehler (Abb. **??**).

Wie erwartet, ist der Partitionierungsfehler generell klein beim Vergleich einer pyramidalen mit einer nicht pyramidalen Klasse. Bei den topologischen Distanzen ist, die beschränkte Distanz $d^c_{edit,2}$ geringfügig besser als $d^c_{edit,1}$. Neben den beiden Attributen $label_6$ und $label_{18}$, die sich auf Länge und Oberfläche beziehen, liefert auch das entsprechende Attribut für das Volumen $label_{12}$ gute Ergebnisse. Generell ergeben die Attribute, die die relativen Eigenschaften von Sektionen beschreiben bessere Ergebnisse (D^6-D^8, D^{12}-D^{14}, D^{18}-D^{20}). Längen- und Oberflächeattribute führen zu kleineren Partitionierungsfehlern als Volumenattribute. Leider ist der Partitionierungsfehler beim Vergleich der beiden pyramidalen Klassen (Quadrat(A,B)) und der beiden nicht pyramidalen Klassen (Quadrat(C,D) zu groß. Eine genauere Betrachtung des Partitionierungsfehlers zeigt, dass dieser beim Vergleich von Interneuronen und Granularzellen nicht unter 0.30 fällt. Wir könnten nun versuchen, durch die Kombination verschiedener Metriken und Attribute eine Distanz zu definieren, die diese Klassen besser trennt. Aus der Analyse von Cannon [**?**] wissen wir jedoch, dass die bisher betrachten Zellen aus unterschiedlichen Experimenten stammen und während verschiedener Entwicklungsstufen entnommen wurden. Es kann damit nicht ausgeschlossen werden, dass die Art der Präparierung, in-vivo oder in-vitro oder das Alter der Zellen, einen Einfluß auf die Homogenität der verschiedenen Zellklassen hat. Wir wollen uns daher, genau wie Cannon auch, auf Zellen beschränken, die denselben experimentellen Ursprung haben.

Zellen der Analyse von Cannon

Die von Cannon untersuchten Zellen unterscheiden sich nur in der Zusammensetzung der Klasse der Ca1-Pyramiden und der Granularzellen von den bisher betrachteten Zellen. Die Anzahl der Ca1-Pyramidenzellen reduziert sich auf 22 und die der Granularzellen auf 18. Die Klasse der Ca3-Pyramidenzellen und Interneuronen besteht wie in den vorherigen Betrachtungen aus 16 bzw. 13 Zellen. Wir führen nun für diese Zellen eine vergleichbare Analyse, wie im vorherigen Abschnitt durch und untersuchen den Partitionierungsfehler beim Vergleich von jeder Klasse mit jeder anderen (Abb. **??**).

Mehrere Attribute führen beim Vergleich der beiden Pyramiden-Klassen (Quadrat (A,B)) jetzt zu einem kleinen Partitionierungsfehler. Dies legt den Schluß nahe, dass sich die Zellen der beiden Klassen damit tatsächlich signifikant in der Morphologie unterscheiden. Diese Aussage erweitert das Ergebnis von Cannon, der lediglich Granularzellen, Interneuronen und Pyramidenzellen als eigenständige morphologische Klassen identifizieren konnte. In Abbildung **??** ist die Verteilung der Zellen, projiziert auf zwei Dimensionen und die Partitionierung von PAM sowie das Dendrogramm von WARD dargestellt.

Wir wollen nun die Partitionierungsfehler beim Vergleich von Interneuronen und Granularzellen genauer untersuchen. In Tabelle **??** sind die gerundeten Werte für die Partitionierungen von PAM und WARD dargestellt. Wir erkennen, dass Attribute, die bei den

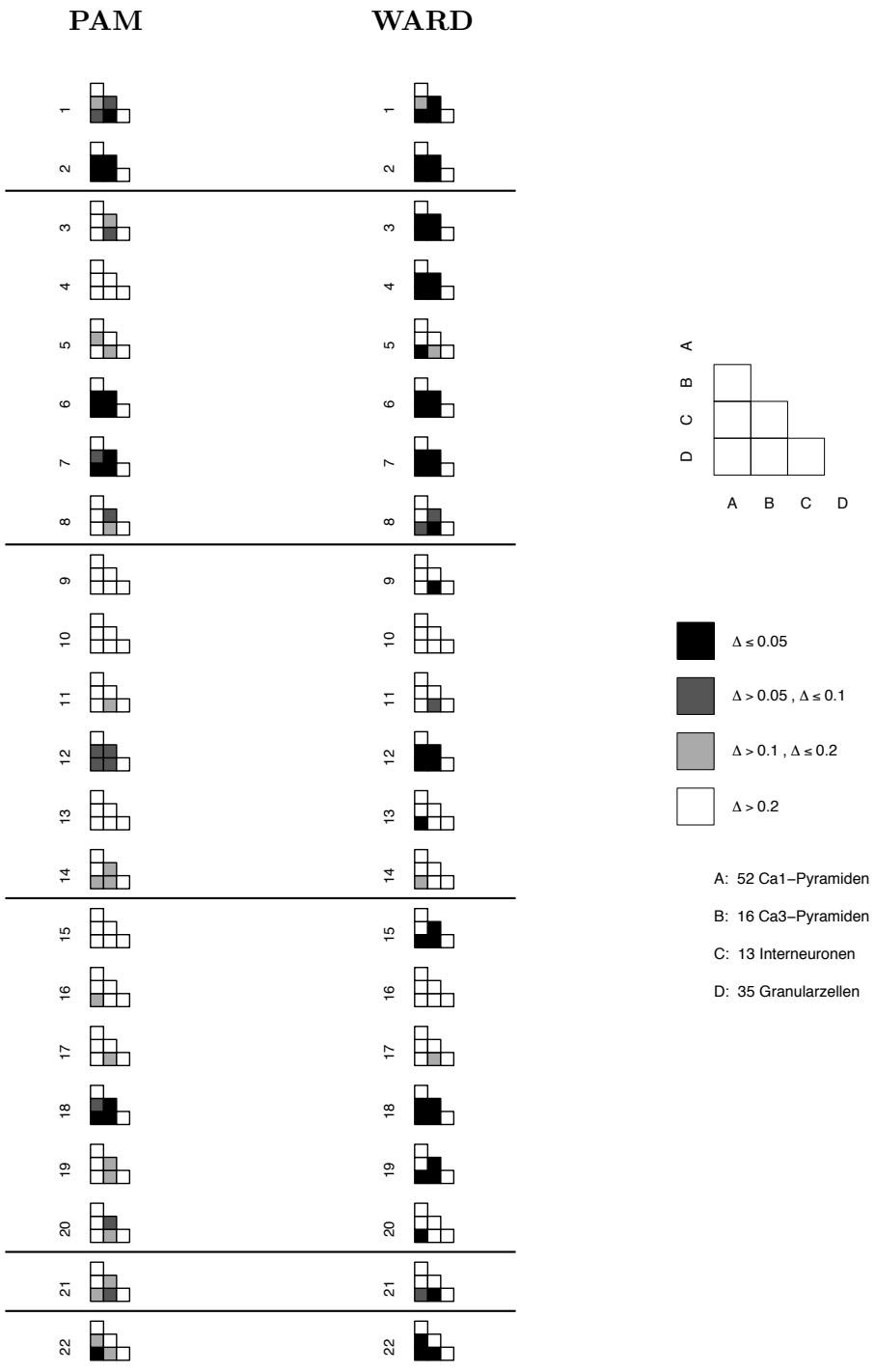


Abbildung 5.3: *Klassifizierung von* 116 *Zellen des Hippokampus: Die Färbung jedes Quadrats gibt die Größe des Partitionierungsfehlers beim Vergleich zweier verschiedener Klassen wieder. In der linken Spalte sind die Fehler des Clusteralgorithmus PAM für die 22 verschiedenen Distanzmatrizen dargestellt. Die rechte Spalte fasst die Ergebnisse der Methode WARD zusammen.*

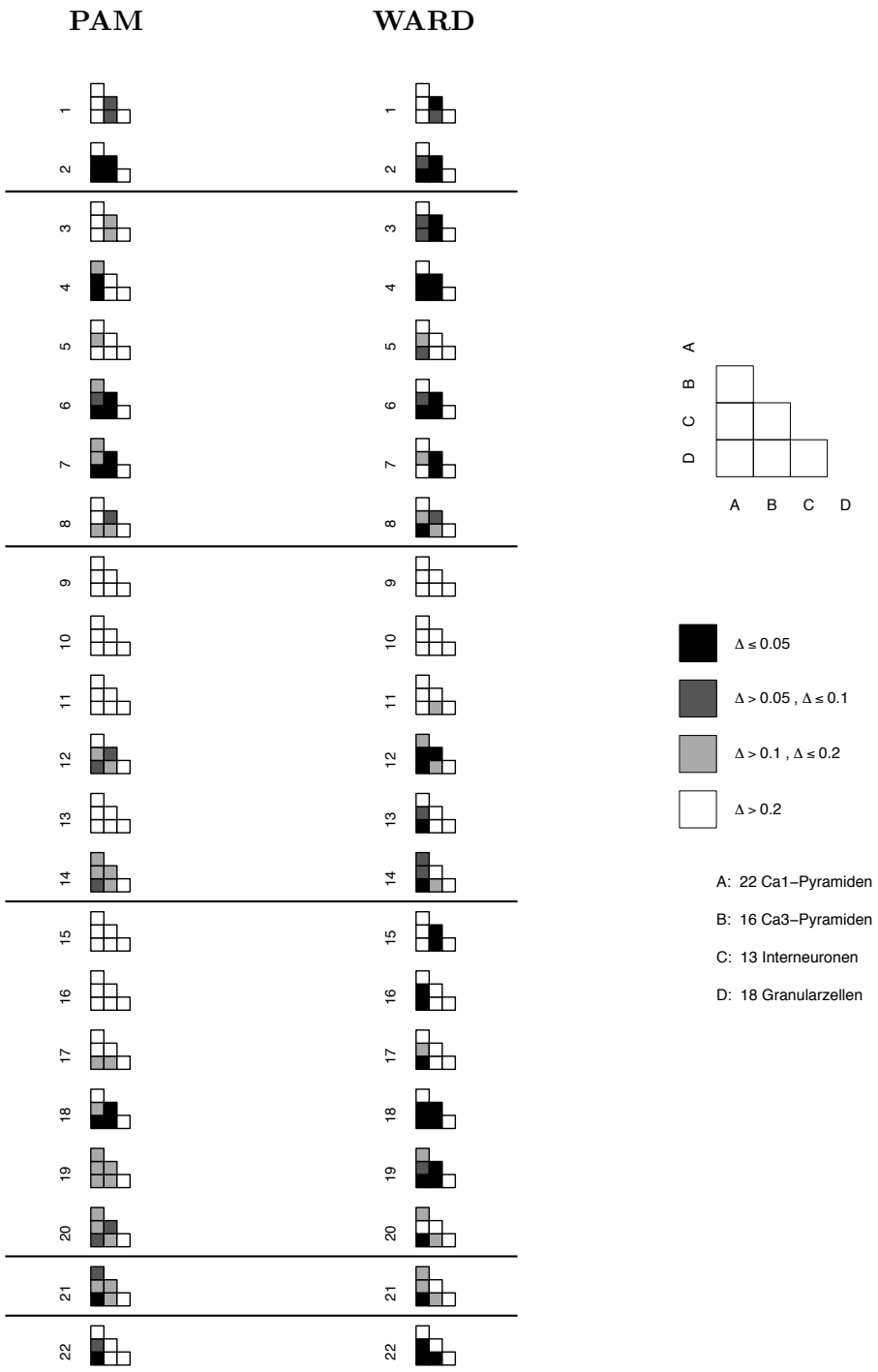


Abbildung 5.4: *Klassifizierung von* 69 *Zellen des Hippokampus: Die Färbung jedes Quadrats gibt die Größe des Partitionierungsfehlers beim Vergleich zweier verschiedener Klassen wieder. In der linken Spalte sind die Fehler des Clusteralgorithmus PAM für die 22 verschiedenen Distanzmatrizen dargestellt. Die rechte Spalte fasst die Ergebnisse der Methode WARD zusammen.*

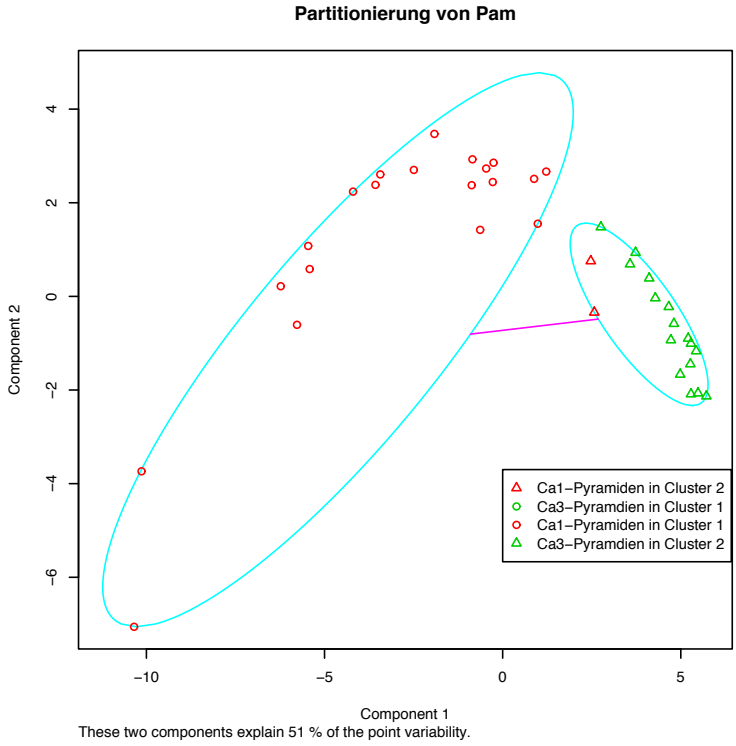


Abbildung 5.5: *Klassifizierung von Pyramidenzellen unter Verwendung des Attributes* $label_{14}$*. Sowohl die Partitionierung von PAM (oben) als auch die von WARD stimmt mit der Einteilung nach morphologischen Klassen nahezu überein.*

Attribut	$PAM : \Delta$	$WARD : \Delta$
1	0.58	0.45
2	0.77	0.97
3	0.39	0.39
4	0.39	0.45
5	0.65	0.32
6	0.39	0.39
7	0.77	0.32
8	0.58	0.71
9	0.97	0.77
10	0.77	0.58
11	0.65	0.71
12	0.32	0.26
13	0.84	0.84
14	0.32	0.58
15	0.71	0.39
16	0.90	0.84
17	0.90	0.90
18	0.45	0.39
19	0.84	0.45
20	0.52	0.77
21	0.39	0.65
22	0.26	0.32

Tabelle 5.6: *Partitionierungsfehler beim Vergleich von Interneuronen und Granularzellen.*

Attribut	$PAM : \Delta$	$WARD : \Delta$
6	0.39	0.39
12	0.32	0.26
13	0.84	0.84
19	0.84	0.45
22	0.26	0.32
23	0.32	0.32
24	0.32	0.39
25	0.19	0.39
26	0.26	0.26

Tabelle 5.7: *Partitionierungsfehler beim Vergleich von Interneuronen und Granularzellen unter der Berücksichtigung von zwei Attributen.*

Vergleichen anderer Klassen zu kleinen Partitionierungsfehlern führten auch in diesem Fall die besten Ergebnisse liefern. Dazu gehört $label_{22}$, $label_6$, $label_{12}$ und $label_{18}$. Betrachten wir die Zellen der beiden verschiedenen Klassen (Abb. **??**), so fällt auf, dass die Interneuronen sich offensichtlich in der Orientierung der Dendriten am Soma von den Granularzellen unterscheiden. Es war daher zu erwarten, dass das Attribut $label_{22}$, das beste Ergebnis liefert. Der Fehler liegt mit 0.26 (PAM) und 0.32 (WARD) zumindest für PAM nur geringfügig über dem als akzeptabel angesehen Wert von 0.2. Es wäre interessant zu testen, wie sich der Partitionierungsfehler bei einem größeren Datenumfang verhält.

Wir können aber auch versuchen bei der Bestimmung des Abstandes mehr als ein Attribut zu berücksichtigen, um bessere Ergebnisse zu erzielen.

$$
\begin{aligned}
label_{23}(t[i]) &= \begin{pmatrix} label_{22}(t[i]) \\ label_{12}(t[i]) \end{pmatrix} \\
label_{24}(t[i]) &= \begin{pmatrix} label_{22}(t[i]) \\ label_{6}(t[i]) \end{pmatrix} \\
label_{25}(t[i]) &= \begin{pmatrix} label_{22}(t[i]) \\ label_{13}(t[i]) \end{pmatrix} \\
label_{26}(t[i]) &= \begin{pmatrix} label_{22}(t[i]) \\ label_{19}(t[i]) \end{pmatrix}
\end{aligned}
$$

Die Attribute $label_{23}$, $label_{24}$, $label_{25}$ und $label_{26}$ berücksichtigen neben dem Winkelattribut $label_{22}$ jeweils ein weiteres Attribut. Lediglich die Kombination von $label_{22}$ und $label_{13}$ lässt den Partitionierungsfehler auf unter 0.2 fallen (Tabelle **??**). Eine andere Möglichkeit mehr als ein Attribut bei der Abstandsbestimmung zu berücksichtigen, besteht in der Linearkombination von Attributen.

Grundsätzlich sind beliebig viele weitere Kombinationen von Attributen denkbar. Eine strukturierte Analyse dieser Kombinationen ist sehr aufwändig, da die Werte der Attribu-

Attribut	$PAM : \Delta$	$WARD : \Delta$
13	0.84	0.84
18	0.45	0.39
19	0.84	0.45
22	0.26	0.32
26	0.26	0.26
22+19	0.19	0.32
22+13	0.19	0.32
26+21	0.19	0.32
26+18	0.19	0.32

Tabelle 5.8: *Partitionierungsfehler beim Vergleich von Interneuronen und Granularzellen. Die Summe zweier Edit-Distanzen ist wieder ein metrisches Distanzmaß für Neuronen und kann zu einem kleineren Partitionierungsfehler führen.*

te verschiedene Skalenniveaus besitzen. Die in Abschnitt **??** beschriebene Standardisierung ist aufgrund des hohen Aufwandes der Abstandsberechnung bei der Hinzunahme weiterer Zellen nicht praktikabel. Nur durch die Analyse einer sehr großen Anzahl von Zellen könnten geeignete Normierungsfaktoren gefunden werden, die dann für alle weiteren Zellen verwendet werden.

Neben den eben beschriebenen Kombinationsmöglichkeiten können neue Abstandsfunktionen durch Summation der bisher betrachteten eingeschränkten Edit-Distanzen definiert werden:

$$d^c_{edit,sum} = \sum_{i=1}^{22} a_i d^c_{edit,i}.$$

Addieren wir beispielsweise $d^c_{edit,19}$ und $d_{edit,22}$ so ist der Partitionierungsfehler von PAM kleiner als 0.2. In der Tabelle **??** ist der Partitionierungsfehler von PAM für verschiedene Kombinationen der bisher betrachteten Distanzmatrizen dargestellt.

Die Diskussion des Partitionierungsfehlers bei Interneuronen und Granularzellen zeigt, dass durch die Kombination verschiedener Attribute bzw. die Summation mehrerer Abstandsfunktionen Distanzmaße gefunden werden, die gewisse Zellklassen besser unterscheiden. Es stellt sich dann die Frage, ob nicht auch ein Distanzmaß gefunden werden kann, dass sämtliche Zellklassen unterscheiden kann. Für die hier betrachteten Zellen ist der Partitionierungsfehler des Cluster-Verfahren PAM beim Vergleich von jeweils zwei Klassen relativ gering, wenn wir die Abstände über

$$d^c_{edit,sum} = d^c_{edit,21} + 0.3 * d_{edit,22} + 1.5 * d^c_{edit,8}$$

bestimmen (Abb. **??**).

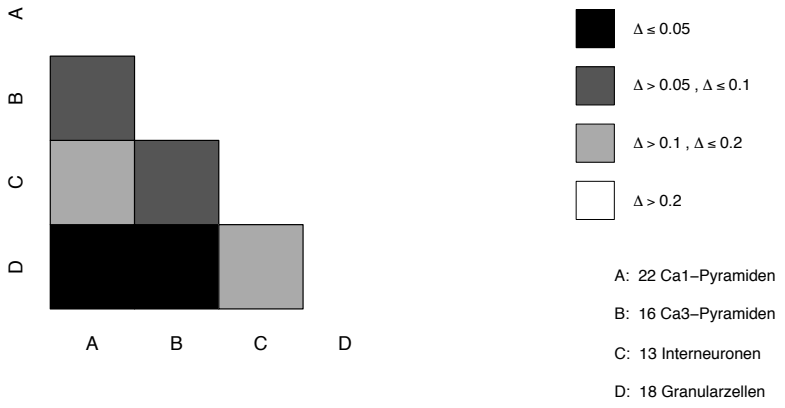
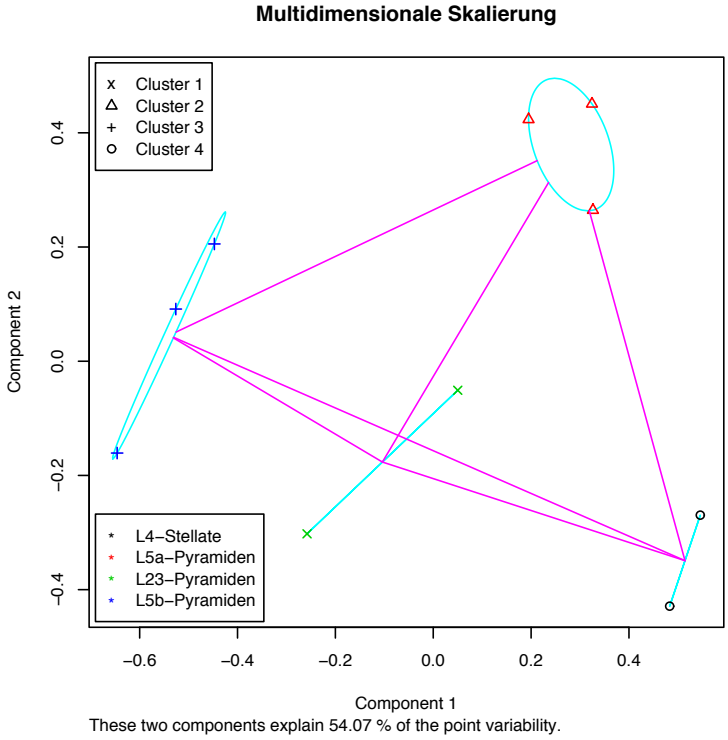


Abbildung 5.6: *Klassifizierung von Zellen des Hippopkampus.*

5.5.2 Cortex

Leider ist der Umfang an Morphologien aus der Datenbank CortexDB zu gering, um zu überprüfen, ob die eingeschränkte Edit-Distanz verschiedene Zellklassen des Cortex unterscheiden kann. In Abbildung **??** sind die Partitionierungen von PAM und WARD anhand der Distanzmatrix D^6 dargestellt. Beide Methoden können die Klassifizierung in L5a-Pyramidenzellen, L5b-Pyramidenzellen, L2/3-Pyramidenzellen und L4-Spiny-Stellate reproduzieren. Aufgrund der geringen Anzahl der verwendeten Morphologien ist es noch fraglich wie allgemeingültig dieses Ergebnis ist. Da die bisherigen Untersuchungen jedoch zeigen, dass das Attribut $label_6$, das jedem Knoten die Länge der repräsentierten Sektion relativ zur Gesamtlänge des Dendriten zuweist, zu guten Ergebnissen führt, kann ein solches Ergebnis erwartet werden.

Neben den Zellen der CortexDB, standen noch 29 L5-Pyramidenzellen zur Verfügung, die aber nicht weiter in L5a- und L5b-Pyramidenzellen unterschieden wurden. Andreas Schäfer konnte in seiner Arbeit [**?**] zeigen, dass es innerhalb dieser L5-Pyramidenzellen zwei Gruppen mit unterschiedlichen elektrophysiologischen Eigenschaften gibt, die sich zudem in ihrer Morphologie unterscheiden. Wir konnten dieses Ergebnis mit der hier vorgestellten Analyse bisher nicht reproduzieren. Der Grund liegt wohl darin, dass der morphologische Unterschied zwischen diesen beiden Gruppen sehr diffizil ist. Schäfer fand heraus, dass die Anzahl der Verzweigungspunkte in einem bestimmten Abstand vom Soma entscheidend für die Eigenschaft des BAC-Firing ist. Der Unterschied ist dabei wohl zu gering, als dass er den Wert der eingeschränkten Edit-Distanz signifikant beeinflußt.



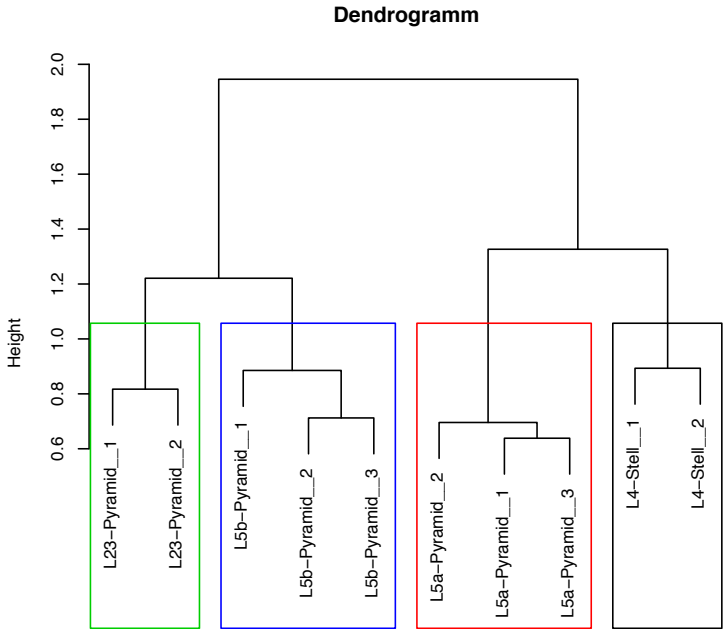


Abbildung 5.7: *Klassifizierung von Zellen aus der CortexDB unter Verwendung des Attributes* $label_6$*. Sowohl die Partitionierung von PAM (oben) als auch die von WARD stimmt mit der Einteilung nach morphologischen Klassen überein.*

5.5.3 NeuGen

Neben den Morphologien echter Zellen können wir künstlich generierte Zellen verwenden, um die verschiedenen Versionen der eingeschränkten Edit-Distanzen zu testen. Abbildung **??** fasst die Ergebnisse zusammen. Es fällt auf, dass insgesamt die Partitionierungsfehler sehr klein sind. Dies war zunächst auch so zu erwarten, da der Generierungsalgorithmus gerade dafür entwickelt wurde, Zellen unterschiedlicher morphologischer Klassen zu erzeugen. Das Ergebnis bestätigt noch einmal, dass die generierten Zellen jeder Klasse sich tatsächlich signifikant von Zellen anderer Klassen unterscheiden und der Algorithmus nicht identische Zellen unterschiedlicher morphologischer Klassen generieren kann.

In Abbildung **??** ist das Ergebnis der hierarchischen Clusteranalyse am Beispiel des Attributes $label_{18}$, das eine Sektion durch seine anteilige Oberfläche an der Gesamtoberfläche $\frac{surface(section)}{surface(Tree)}$ beschreibt, dargestellt. Das Dendrogramm zeigt, dass die Edit-Distanz, die über dieses Attribut definiert ist, die 50 verschiedenen Neuronen in den entsprechenden fünf Klassen zusammenfasst.

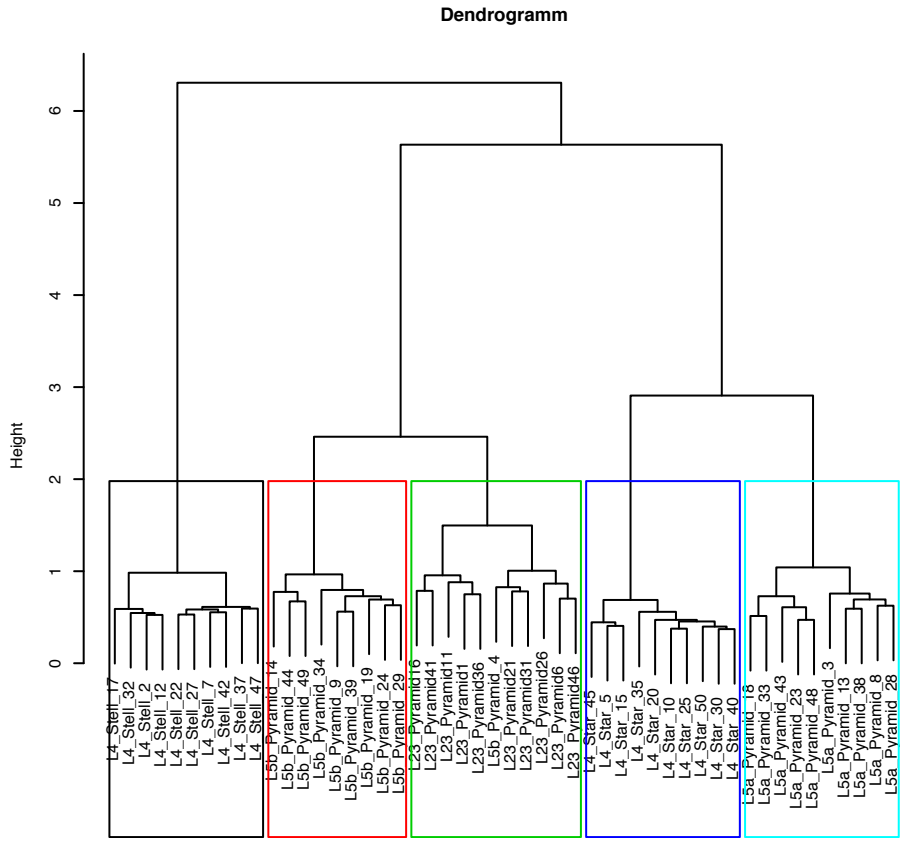


Abbildung 5.9: *Partitionierung von Zellen aus NeuGen unter Verwendung des Attributes* $label_{18}$.

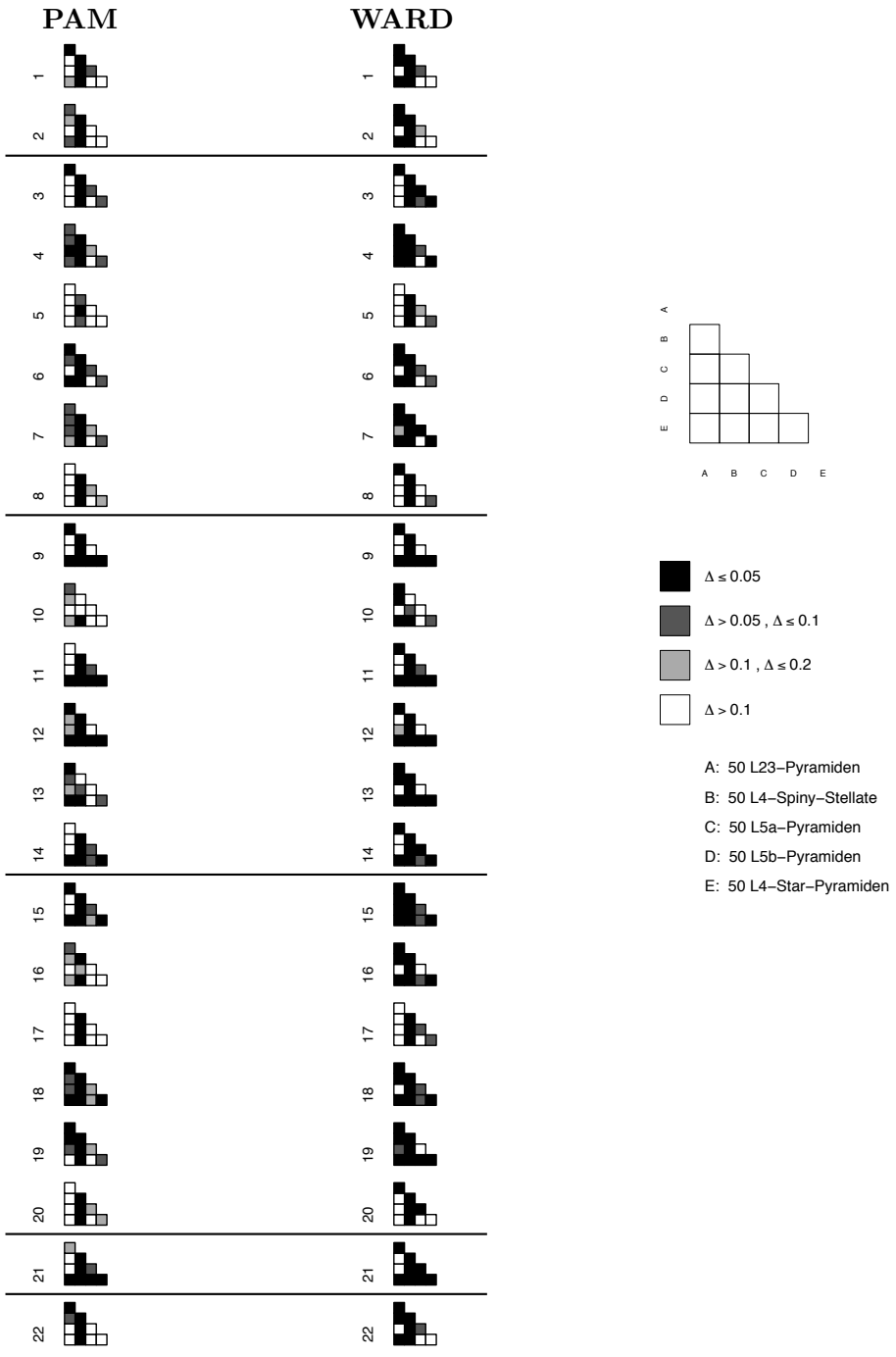


Abbildung 5.8: *Klassifizierung von mit NeuGen generierten künstlichen Zellen: Die Färbung jedes Quadrats gibt die Größe des Partitionierungsfehlers beim Vergleich zweier verschiedener Klassen wieder. In der linken Spalte sind die Fehler des Clusteralgorithmus PAM für die 22 verschiedenen Distanzmatrizen dargestellt. Die rechte Spalte fasst die Ergebnisse der Methode WARD zusammen.*

Wir können natürlich auch untersuchen, ob es Unterschiede zwischen echten und generierten Zellen gibt, indem wir die Abstände der Zellen zueinander über eine eingeschränkte Edit-Distanz bestimmen und dann überprüfen, ob die automatische Partitionierung echte und generierte Zellen derselben morphologischen Klasse zusammenfasst. In Abbildung **??** sind die Dendrogramme solcher Evaluationen für L5a- und L5b-Pyramidenzellen dargestellt. Da in beiden Fällen die echten Zellen mit den künstlichen Zellen ihrer Klasse in einer Partion liegen, generiert der Algorithmus tatsächlich realitätsnahe L5a- und L5b-Pyramiden.

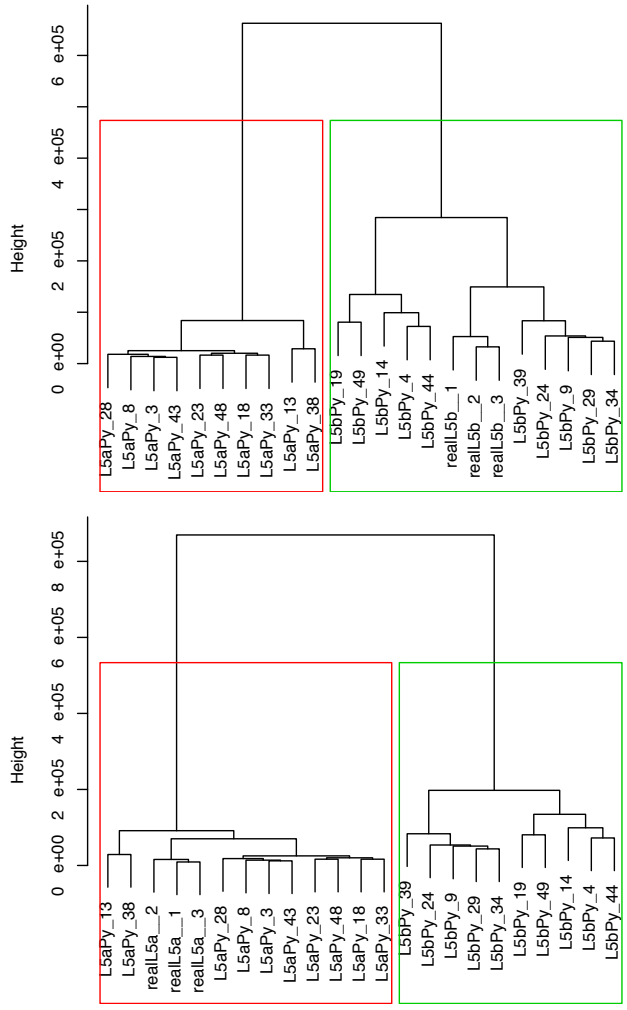


Abbildung 5.10: *Vergleich von echten und künstlich generierten L5a und L5b-Pyramidenzellen anhand des Attributs* $label_4$*. Die echten Zellen, hier durch das Präfix* real *gekennzeichnet, werden mit den generierten Zellen der richtigen Klasse in einer Partition zusammengefasst.*

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

Der Gegenstand dieser Arbeit war die Definition einer Abstandsfunktion für Neuronen, die in der Lage ist, morphologische Unterschiede zwischen verschiedenen Zellklassen zu detektieren. Wir konnten zeigen, dass die eingeschränkte Edit-Distanz nach Zhang [**?**] zusammen mit der Clusteranalyse eine geeignete Methodik ist, um morphologische Klassen zu detektieren. Sicherlich ist der Umfang der Daten zu gering, um die Aussagen über die Trennungsschärfe der verschiedenen Metriken ohne weitere Untersuchungen zu verallgemeinern. Dennoch gibt es mehrere Punkte, die dafür sprechen, dass morphologische Analysen mit Hilfe der eingeschränkten Edit-Distanz zu befriedigenden Ergebnissen führen. Zunächst ist der Spezialfall der Edit-Distanz auf Wörtern eines der Standard-Verfahren in der DNA-Sequenz-Analyse. Ein großer Bereich der Bioinformatik beschäftigt sich heute mit der Weiterentwicklung der Methoden von Wagner und Fischer [**?**]. Die eingeschränkte Edit-Distanz für Bäume selbst wird bereits erfolgreich in der Analyse von RNA-Sekundärstrukturen [**?**, **?**] und von botanischen Bäumen [**?**] eingesetzt. Das Prinzip der Edit-Distanz scheint also prädestiniert für medizinisch-biologische Anwendungen zu sein. Dies wird noch deutlicher, wenn wir die Folgen von Edit-Operationen, die dabei betrachtet werden, als Transformationen der Struktur, als Wachsen oder Schrumpfen einzelner Bereiche, ansehen. Die zu jeder Edit-Distanz gehörende Folge von Edit-Operationen stellt dann also den wahrscheinlichsten Transformationsprozess dar.

Wichtig für das Verständis der eingeschränkten Edit-Distanz ist der Begriff der Spur. Die Spur ist eine bijektive Abbildung von einer Teilmenge der Knoten des einen Baumes auf eine Teilmenge der Knoten des anderen, die gewisse Ordnungsrelationen der Knoten erhält. Jedes Knotenpaar einer Spur ordnet einem Bereich des einen Baumes einen Bereich des zweiten Baumes zu, der in gewisser Weise äquivalent ist (Abb. **??**). Unter Äquivalenz wird hier in erster Linie die topologische Ähnlichkeit, d.h. eine vergleichbare Lage innerhalb der Baumstruktur verstanden.

Die zur Edit-Distanz gehörende Spur ist damit diejenige Spur, die am besten topologisch

äquivalente Bereiche von zwei Bäumen identifiziert. Der Betrag der Edit-Distanz ist bestimmt durch die Bereiche, die kein topologisches Äquivalent besitzen und durch die lokale Unähnlichkeit topologisch äquivalenter Bereiche. Die Spuren der eingeschränkten und der normalen Edit-Distanz unterscheiden sich durch die Ordnungsrelationen, die sie erhalten. Eine eingeschränkte Spur impliziert neben der Erhaltung des Vorgänger-Nachfolger-Verhältnisses zusätzlich, dass Teilbäume wieder Teilbäumen zugeordnet werden. Erst diese Einschränkung macht es überhaupt möglich, einen Abstand von Bäumen, der auf dem Prinzip von Edit-Operationen beruht, effektiv zu berechnen.

Überträgt man den Zusammenhang von Edit-Distanz und Spurbegriff direkt auf die Zellmorphologien, so ergibt sich eines der Hauptargumente, das für die Verwendung der eingeschränkten Edit-Distanz in der Analyse von Zellmorphologien spricht. Die Bestimmung des Abstandes zweier Zellen geht einher mit der Bestimmung einer Substruktur, die in beiden Zellen enthalten ist, und optimal in dem Sinne ist, dass alle anderen gemeinsamen Substrukturen zu einem größeren Abstand führen. Die Abstandsbestimmung durch die Edit-Distanz berücksichtigt also automatisch äquivalente Teilstrukturen, wie etwa den prominenten Apikaldendriten bei Pyramidenzellen.

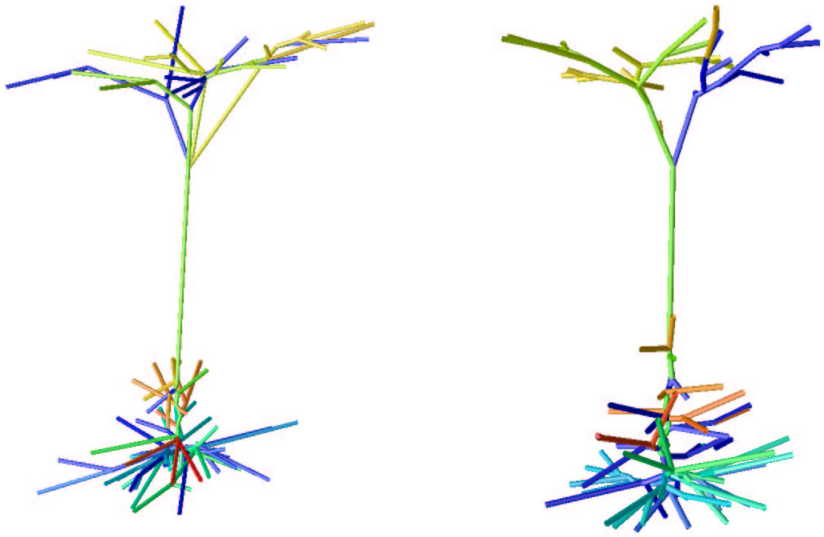


Abbildung 6.1: *Visualisierung einer eingeschränkten Spur zwischen Bäumen. Sektionen gleicher Farbe stellen ein Element der eingeschränkten Spur dar. Blau gefärbte Sektionen werden in der Spur nicht berücksichtigt. Die Spur ordnet deutlich erkennbar äquivalente Bereiche wie Apikaldendriten (türkis), Trunk (grün) oder Tuft (gelb, grün) einander zu.*

6.2 Ausblick

Wir konnten zeigen, dass die zwei Cluster-Algorithmen PAM und WARD in der Lage sind, aus den Abstandswerten der eingeschränkten Edit-Distanz verschiedene morphologische Klassen zu extrahieren. Es stellt sich die Frage, ob vielleicht andere Clusterverfahren, wie etwa dichte- oder modellbasierte Clusterverfahren [**?**], zu besseren Ergebnissen führen. Viele Clusterverfahren setzten eine vektorbasierte Darstellung der zu partitionierenden Objekte voraus. Um diese Verfahren verwenden zu können, müssen den Nervenzellen daher durch die beschriebene multidimensionale Skalierung Vektoren zugeordnet werden, deren euklidische Abstände den Edit-Distanzen entsprechen.

Weiterhin ist es möglich, durch die Definition neuer Attribute, die Verwendung anderer lokaler Abstandsfunktionen und die Kombination von verschiedenen eingeschränkten Edit-Distanzen bessere Ergebnisse zu erzielen. Interessant wäre auch die Betrachtung nicht metrischer Attribute wie etwa der Verteilung von Ionenkanälen, die dann neben der Morphologie auch elektrophysiologische Eigenschaften bei der Abstandsbestimmung berücksichtigt. Auch könnte versucht werden, anstatt des vorgestellten sektionsbasierten Ansatzes Teilstücke gleicher Länge als Knoten einer Baumdarstellung von Nervenzellen zu definieren. Da diese Teilstücke sehr klein sein müssen, um die tatsächliche Struktur gut zu approximieren, kann die Anzahl der Knoten und damit der zeitliche Aufwand allerdings sehr groß werden.

Neben der vorgestellten eingeschränkten Edit-Distanz gibt es einen weiteren Ansatz, durch das Einführen einer Restriktion eine berechenbare Edit-Distanz zu erhalten. Diese sogenannte Alignment-Distanz ist über das minimale Gewicht derjenigen Folgen von Edit-Operationen definiert, die das eine Objekt in das andere überführen und in denen alle Einfügeoperationen vor den Löschoperationen stattfinden. Es ist bekannt, dass die Edit-Distanz und die Alignment-Distanz für Wörter identisch sind [**?**]. Jiang, Wang und Zhang zeigen in [**?**], dass die Alignment-Distanz sowohl für geordnete als auch für ungeordnete Bäume in polynomialer Laufzeit bestimmt werden kann. Im Gegensatz zur eingeschränkten Edit-Distanz wird zur Berechnung der Alignment-Distanz ein Baum bestimmt, der die beiden zu vergleichenden Bäume komplett enthält, also eine Art Superbaum.

Literaturverzeichnis

[AMO93] R.K. Ahuja, T.L. Magnanti und J.B. Orlin. *Network flows: theory, algorithms and applications*. Prentice Hall, 1993.

[BSL+04] P.J. Broser, R. Schulte, S. Lang, A. Roth, F. Helmchen, J. Waters, B. Sakmann und G. Wittum. Nonlinear anisotropic diffusion filtering of three-dimensional image data from two-photon microscopy. *Journal of Biomedical Optics*, 9(6):1253–1264, 2004.

[BT88] D.P. Bertsekas und P. Tseng. Relax: A computer code for the minimum cost network flow problem. *Annals of Operation Research*, 13:127–190, 1988.

[CTPW99] R.C. Cannon, D.A. Turner, G.K. Pyapali und H.V. Wheal. An on-line archive of recontructed hippocampal neurons. *Journal of Neuroscience Methods*, 84:48–54, 1999.

[CWT99] R.C. Cannon, H.V. Wheal und D.A. Turner. Dendrites of classes of hippocampal neurons differ in structural complexity and branching pattern. *The Journal of Comparativ Neurology*, 413:619–633, 1999.

[EWW06] J.P. Eberhard, A. Wanner und G. Wittum. A tool for the generation of realistic morphology of cortical neurons and neural networks in 3d. *Neurocomputing*, accepted for publication 2006.

[FBM95] M. Falk, R. Becker und F. Marohn. *Angewandte Statistik mit SAS: Eine Einführung*. Springer, 1995.

[FG00] P. Ferraro und C. Godin. A distance measure between plan architectures. *Ann. For. Sci.*, 57:445–461, 2000.

[FR02] C. Fraley und A.E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97:611, 2002.

[Gus97] D. Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, 1997.

[Ham80] Richard Hamming. *Coding and Information Theory*. Prentice Hall, 1980.

[HS03] W. Härdle und L. Simar. *Applied Multivariate Statistical Analysis*. Springer, 2003.

[HTGK03] M. Höchsmann, T. Töller, R. Giegerich und S. Kurtz. Local similarity in rna secondary structures. *Proccedings of the IEEE Bioinformatic Conference*, Seiten 159–168, 2003.

[JWZ95] T. Jiang, L. Wang und K. Zhang. Alignment of trees - an alternative to tree edit. *Theoretical Computer Science*, 143:137–158, 1995.

[Kle98] P.N. Klein. Computing the edit-distance between unrooted ordered trees. *Proceedings of the 6th annual European Symposium on Algorithms (ESA)*, Seiten 91–102, 1998.

[KR90] L. Kaufmann und P.J. Rousseuw. *Finding Groups in Data: An Introduction to Data Analysis*. Wiley, 1990.

[KSJ95] E.R. Kandel, J.H. Schwartz und T.M. Jessell. *Neurowissenschaften: Eine Einführung*. Spektrum Akademischer Verlag, 1995.

[KV05] B. Korte und J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer, 2005.

[Lev66] V. I. Levenshtein. Binary codes capable of correcting insertions and reversals. *Sov. Phys. Dokl.*, 10:707–710, 1966.

[MBNK+00] A. Mizrah, E. Ben-Ner, M.J. Katz, K. Kedem, J.G. Glusman und F. Libersat. Comparative analysis of dendritic architecture of indentified neurons using the haussdorff distance metric. *Journal of Comparative Neurology*, 422:415–428, 2000.

[MSB04] I.A. Manns, B. Sakmann und M. Brecht. Sub- and suprathreshold receptive field properties of pyramidal neurons in layers 5a and 5b of rat somatosensory barrel cortex. *Journal of Physiology*, 556(2):601–622, 2004.

[Sch00] A. Schäfer. *Untersuchung morphologischer Korrelate elektrophysiologischer Eigenschaften von Pyramidenzellen im Neocortex der Ratte*. Diplomarbeit, Fakultät für Physik und Astronomie, 2000.

[SLSR03] A.T. Schäfer, M.E. Larkum, B. Sakman und A. Roth. Coincidence detection in pyramidal neurons is tuned by their dendritic branching pattern. *J Neurophysiology*, 89:3143–3154, 2003.

[Tic03] L. Tichit. *Algorithmique des structures biologiques: l'edition d'arborescenes pour la comparision de structures secondaires d'ARN*. Dissertation, L'UNIVERSITY Bordeaux I, 2003.

[TT88] E. Tanaka und K. Tanaka. The tree-to-tree editing problem. *Internat. J. Pattern Recog. Artificial Intell.*, 2(2):221–240, 1988.

[UvP02] H.B.M. Uylings und J. van Pelt. Measures for quantifying dentritic arborization. *Network: Computation in Neural Systems*, 13:397–414, 2002.

[War63] J.H. Ward. Hierarchical groupings to optimize an objective function. *Journal of the American Statistical Association*, 58:234–244, 1963.

[WF74] R.A. Wagner und M.J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 12(1):168–173, 1974.

[Zha96] K. Zhang. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15:205–222, 1996.

[ZS89] K. Zhang und D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal of Computing*, 18:1245–1262, 1989.

[ZSS92] K. Zhang, R. Statman und D. Shasha. On the editing distance between unordered labeled trees. *Information Processing Letters*, 42:133–139, 1992.

Danksagung
Ich danke Herrn Prof. Dr. Gabriel Wittum für die Vergabe dieses interessanten Themas. Insbesondere möchte ich mich dafür bedanken, dass ich in seiner Arbeitsgruppe die Möglichkeit hatte, einen Einblick in ein mir bis dahin völlig unbekanntes Wissenschaftsgebiet, die Neurowissenschaften, zu bekommen. Die Seminare, die Konferenz in Hohenwart und die verschiedenen Praktika zur Bildverarbeitung von Mikroskopaufnahmen waren sehr hilfreich bei der Bearbeitung der Diplomarbeit.

Heidelberg, 27. August 2006 Holger Heumann