

Carl von Ossietzky
Universität Oldenburg

Diplom Mathematik mit Schwerpunkt Biowissenschaften

DIPLOMARBEIT

Vergleich Metrik-basierter Verfahren zur Dekodierung neuronaler Signale

vorgelegt von Imke Christel Gudrun Reimer

am 23.08.2007

Erste Gutachterin:
Zweiter Gutachter:
Betreuerin:

Prof. Dr. Christine Müller
Dr. Peter Harmand
Juniorprof. Dr. Jutta Kretzberg

Inhaltsverzeichnis

1	Einleitung	4
1.1	Motivation	4
1.1.1	Retinale Informationsverarbeitung	4
1.1.2	Kodierung	5
1.2	Überblick und Zielsetzung	7
2	Abstandsmaße	10
2.1	$D^{euklid}[\tau]$	10
2.2	$D^{spike}[q]$ und $D^{interval}[q]$	13
2.2.1	Definition und Berechnung	13
2.2.2	Eigenschaften	17
2.3	Multineuronale Erweiterung von $D^{spike}[q]$	21
2.3.1	Definition und Berechnung	21
2.3.2	Nicht-Einbettbarkeit in euklidischen Raum	24
3	Stimulusabhängiges Clustern	38
3.1	Algorithmus	38
3.2	Güte des stimulusabhängigen Clusters	41
3.2.1	Informationstheoretische Grundlagen	41
3.2.2	Wechselseitige Information und Bias	46
4	Simulationen	52
4.1	Poisson-Prozesse	52
4.2	Anwendung auf künstliche Daten	58
4.2.1	Ratenunterscheidung	58
4.2.2	Zeitliche-Phasen-Unterscheidung	62
5	Anwendung auf retinale Daten	65
5.1	Experiment und Datenselektion	65
5.2	Ergebnisse	68
5.2.1	Dekodierung der Geschwindigkeit	68
5.2.2	Dekodierung der Geschwindigkeitsübergänge	76
5.2.3	Multineuronale Dekodierung der Geschwindigkeit	82
6	Diskussion	86
6.1	Methodische Aspekte	86

6.1.1	Abstandsmaße	86
6.1.2	Gruppenbildung	88
6.1.3	Prozent-Korrekt und wechselseitige Information	89
6.2	Biologische Aspekte	91
6.2.1	Neuronale Kodierung	91
6.2.2	Multineuronale Kodierung	94
6.3	Schlusswort	97
A	Anhang	98
A.1	Beschreibung der Matlab-Programme	98
A.1.1	Datenaufbereitung	98
A.1.2	Charakterisierung des Antwortverhaltens	99
A.1.3	Erzeugung künstlicher Daten	101
A.1.4	Abstandsberechnung	103
A.1.5	Stimulusabhängiges Clustern	108
A.2	Tuningkurven der verwendeten Zellen	113

1 Einleitung

1.1 Motivation

1.1.1 Retinale Informationsverarbeitung

Die Wahrnehmung unserer Umwelt erfordert eine zuverlässige Übertragung der Information über Eigenschaften wie Form, Farbe, Kontrast und Geschwindigkeit eines betrachteten Gegenstandes vom Auge ins Gehirn. Unser Auge ist ein auf visuelle Reize angepasster dioptrischer Apparat (siehe Abb. 1.1), bei dem zunächst die Linse das Licht bricht und den Gegenstand auf unseren Augenhintergrund fokussiert. Hier befindet sich die Netzhaut (Retina), die die erste Station neuronaler Verarbeitung der visuellen Information darstellt und aus einem Netzwerk verschiedener Zelltypen besteht. So lösen die auf die Retina eintreffenden Lichtquanten zunächst in den Photorezeptoren (Zapfen und Stäbchen) biochemische Prozesse aus. In Form von chemischen und elektrischen Signalen gelangt die Information dann über die Bipolar-, Amakrin- und Horizontalzellen zu den sogenannten Ganglienzellen.

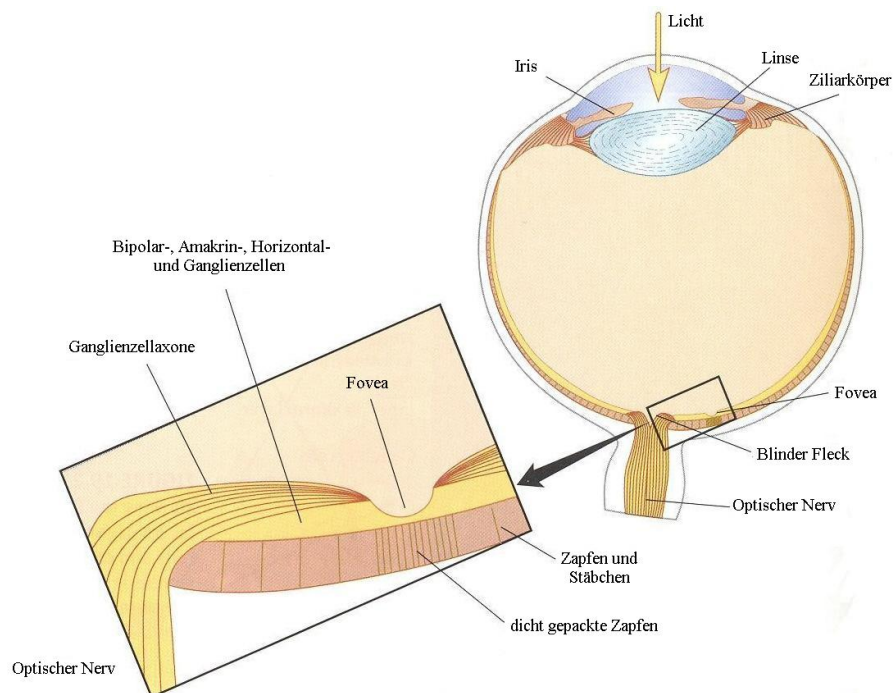


Abbildung 1.1: Aufbau des Auges und der Retina (modifiziert aus [NMWF01])

Zu einem besseren Verständnis der weiteren Informationsübertragung ist es wichtig zu wissen, dass zwischen dem Innen- und Außenmedium einer Nervenzelle (Neuron) eine elektrische Spannung, das sogenannte Membranpotential, besteht. Bei Erregung werden in den meisten Nervenzellen vom Ruhepotential ausgehende schnelle Potentialänderungen, sogenannte Aktionspotentiale (Spikes, siehe Abb. 1.2), ausgelöst. Diese dauern etwa ein bis zwei Millisekunden und haben einen charakteristischen Zeitverlauf.

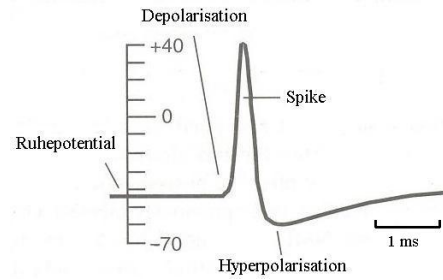


Abbildung 1.2: Typisches Aktionspotential. Membranpotential (in mV) in Abhängigkeit von der Zeit (modifiziert aus [Tr02])

Die Ganglienzellen zeigen bei Reizung diese elektrischen Änderungen. So wird die Information über einen Stimulus über ihre langen Fortsätze (Axone), die den optischen Nerv bilden (siehe Abb. 1.1), in Form einer Folge von Spikes ans Gehirn geleitet. ([SS00])

1.1.2 Kodierung¹

Eine wichtige neurobiologische Fragestellung ist, wie die Information über einen visuellen Stimulus in den Mustern der elektrischen Aktivität einer Ganglienzelle enthalten ist. Eine Herangehensweise ist nun, die Aktionspotentiale, obwohl sie in Dauer, Amplitude und Form variieren, als stereotypische Ereignisse zu betrachten. Man geht also davon aus, dass die Information in der Folge $\{T_1, T_2, T_3, \dots, T_N : T_i \text{ ist der Zeitpunkt, an dem das } i\text{-te Aktionspotential auftritt}\}$ enthalten ist. Im Folgenden wird anstelle von Folge der Aktionspotentialzeitpunkte auch der Begriff *Spiketrain* verwendet.

Lange Zeit ging man davon aus, dass ein Reiz über die Rate der Spikes kodiert wird. Dies war bedingt durch die Tatsache, dass man oft beobachten kann, dass die Spikeanzahl mit einer Reizverstärkung anwächst. Ein Zugrundeliegen eines solchen Ratencodes würde dann nahelegen, dass bei zwei hintereinandergeschalteten Neuronen das eine Neuron die Spikes des es innervierenden Neurons integrieren würde. Es besteht aber auch die Möglichkeit, dass ein Neuron nur dann ein Aktionspotential generiert, wenn die Spikes zweier es innervierenden Neurone es gleichzeitig erreichen. Für dieses als Koinzidenz-Detektor bezeichnetes Neuron ist also das Timing der einzelnen Spikes wichtig. Zudem spielt es manchmal eine Rolle, wie weit aufeinanderfolgende Aktionspotentiale zeitlich auseinander liegen (zum Beispiel bei der Langzeitpotenzierung an den Pyramidenzellen), die Länge der sogenannten *Interspikeintervalle* ist also von Bedeutung. In diesen Fällen ist die Information über einen Stimulus also in der zeitlichen Struktur eines Spiketrains enthalten. Dieser Zeitkode wird auch motiviert durch die Tatsache, dass das „Zählen“ von Spikes eine langsame Verschlüsselung darstellt und somit den Anforderungen einer korrekten Informationsübertragung von Umweltreizen nicht gerecht werden könnte.

Als Kodierungsstrategien vermutet man also Ratencode und Zeitkode, wobei es sich aber

¹Alle angesprochenen Aspekte und zum Teil weitergehende Ausführungen sind in [ARMV03], [DA01], [K01], [KS99], [MB99], [RWRB97], [Tr02], [VP97] und [V99] zu finden.

nicht nur um einen qualitativen sondern vor allem um einen quantitativen Unterschied handelt. Ihnen liegt eine unterschiedliche Bedeutsamkeit der zeitlichen Präzision der Spikes zugrunde. In diesem Sinne kann der Zeitkode interpretiert werden als ein Spikeanzahlen-Kode bei einem entsprechend hohen Grad der zeitlichen Auflösung. Es stellt sich somit die Frage nach der der Verschlüsselung und der Übertragung einer Information zugrundeliegenden Zeitskala.

Die Frage nach dem zugrundeliegenden Kode ist aber nicht auf individuelle Ganglienzellen zu reduzieren. So überlappen sich die rezeptiven Felder (die Bereiche der Netzhaut, von denen aus die Aktivität einer Zelle beeinflusst werden kann) von Ganglienzellen. Spiket trains mehrerer Neurone beinhalten also die Information über den gleichen Teil eines visuellen Bildes, weshalb betrachtet werden muss, inwieweit die jeweiligen Antworten zusammenwirken. Außerdem weist das neuronale Antwortverhalten eine gewisse Variabilität auf. Dieses Problem der Zuverlässigkeit könnte durch die Mittelung über die elektrische Aktivität einer Population umgangen werden und somit zu einer besseren Kodierung führen. Neben der Frage der Zeitskala ist somit auch zu untersuchen, inwieweit die Spikes der einzelnen Zellen als individuelle Ereignisse in die Informationsverarbeitung eingehen, wie lokal eine solche Populationsmittelung also ist.

1.2 Überblick und Zielsetzung

In meiner Diplomarbeit werde ich basierend auf vor allem den Arbeiten von Victor et al. (insbesondere [VP96], [VP97]) ein Verfahren einführen und verwenden, bei dem über eine Dekodierung von Reizinformationen in neuronalen Signalen Aufschluss über eine zugrundeliegende Kodierungsstrategie der Nervenzellen gewonnen werden soll. Die Vorgehensweise basiert dabei auf der Idee, dass sich die neuronalen Antworten auf denselben Stimulus ähnlicher sind als die Antworten auf verschiedene Stimuli und dieser Unterschied durch ein Abstandsmaß beschrieben werden können sollte. Eine gute Unterscheidbarkeit der Spiketrains hinsichtlich der Stimuli, und somit eine gute Dekodierung der Stimulusinformation, ist also dann möglich, wenn man bei der Distanzbestimmung die Aspekte betont, die für die Kodierung der Information bedeutsam sind.

Eine Vermutung hinsichtlich dieser Aspekte ist, dass die Stimulusinformation in den Zeitpunkten der Spikes auf einer bestimmten Zeitskala liegt. Dies ist in den beiden Familien von Abstandsmaßen $D^{spike}[q]$ nach Victor et al. ([VP96], [VP97]) und $D^{euklid}[\tau]$ nach Kretzberg et al. ([KWE01]) umgesetzt, wobei je mit den Parametern q und τ die zeitliche Präzision der Spikezeitpunkte gewichtet wird. Ersteres Maß stellt den minimalen Preis dar, um einen Spiketrain mittels sogenannter *elementarer Umformungen* einem anderen anzugleichen. Diese elementaren Schritte sind dabei mit unterschiedlichen *Kosten* assoziiert und sind zum Teil von dem Parameter q abhängig. Bei Letzterem wird die euklidische Norm auf mit einem Rechteckkern der Länge τ gefilterte Spiketrains angewandt. Des Weiteren wird die Familie von Abstandsmaßen $D^{interval}[q]$ nach Victor et al. eingeführt, bei der es sich wie bei $D^{spike}[q]$ um eine Familie von Metriken handelt. Die Motivation von $D^{interval}[q]$ ist, dass die Zeit zwischen zwei Aktionspotentialen bedeutsam sein kann. Hier wird die Information über einen Reiz also in der Folge der Interspikeintervalle vermutet und dementsprechend diese Folgen betrachtet. Wieder stellt der Parameter q die zeitliche Auflösung dar und ist der Abstand gerade der minimale Preis, um zwei Spiketrains mit elementaren Umformungen einander anzugleichen.

Ausserdem wird die multineuronale Erweiterung $D^{multi}[q, k]$ von $D^{spike}[q]$ (insbesondere [A03], [AV04]) definiert, der die Idee zugrundeliegt, dass das Zusammenwirken mehrerer Neurone für die Kodierung von Stimulusinformationen wichtig ist. Hier werden sogenannte *gekennzeichnete Spiketrains* betrachtet, die Folgen von Spikezeitpunkten von mehreren Nervenzellen darstellen. Dabei sind die Spikezeitpunkte entsprechend dem Neuron, durch das das jeweilige Aktionspotential ausgelöst wurde, gekennzeichnet. Neben dem Parameter q , der auch hier die zeitliche Auflösung repräsentiert, gibt es einen Parameter k . Dieser stellt die Wichtigkeit der Unterscheidung individueller Neurone dar. Der Abstand zweier gekennzeichnete Spiketrains ist auch hier der minimale Preis, um diese mittels elementarer Umformungen anzugleichen. Dabei bestimmen die Parameter q und k einen Teil der Kosten der elementaren Schritte.

Neben dem Definieren dieser Abstandsmaße werden ihre Eigenschaften basierend auf vor allem biologischen Arbeiten mathematisch ausgearbeitet.

Mit Hilfe dieser Abstandsmaße ist es möglich quantitativen und qualitativen Aufschluss über die zugrundeliegende Kodierung zu gewinnen. Dazu ist das Abstandsmaß zu bestimmen, das zur besten Unterscheidbarkeit der Spiketrains hinsichtlich des Stimulus, durch den diese je erzeugt wurden, im Vergleich aller Abstandsmaße führt. Handelt es sich hierbei zum Beispiel um das Maß, das gerade die absolute Differenz der Spikeanzahlen zweier neuronaler Antworten darstellt, so bedeutet das, dass die Reizinformation in der Spike-rate kodiert ist. Um nun das optimale Abstandsmaß auszumachen, werden Spiketrains, die durch unterschiedliche Stimuli hervorgerufen wurden, basierend auf verschiedenen Abstandsmaßen mittels des sogenannten *stimulusabhängigen Clusters* in Gruppen eingeteilt. Daraufhin wird für jedes Abstandsmaß diese Einordnung auf Basis des Wissens, welcher Spiketrain zu welchem Stimulus gehört, bewertet.

In den Algorithmus des stimulusabhängigen Clusters geht in die Gruppenbildung die Information ein, durch welchen Stimulus die Spiketrains hervorgerufen wurden, welcher Klasse sie also eigentlich angehören. Insofern unterscheidet sich diese Methode von herkömmlichen Clusteranalysen beziehungsweise Clusterformationen ([Sp83]), bei denen diese Information im Allgemeinen nicht vorliegt und somit auch nicht in die Einteilung von Gruppen eingeht.² Hinsichtlich der Einbeziehung der eigentlichen Gruppenstruktur erinnert das Verfahren des stimulusabhängigen Clusters an eine Klassifikation.³ ([ES00])

Die Bestimmung der Güte des stimulusabhängigen Clusters erfolgt dann mittels *Prozent-Korrekt* (nach [KWE01]) und *wechselseitiger Information* (nach [VP96], [VP97]), welche zu diesem Zweck eingeführt werden. Hierbei sei angemerkt, dass es sich nicht um die im Allgemeinen betrachtete Güte des Clusters handelt,⁴ sondern die Stimulusabhängigkeit der Einteilung bewertet wird.

Nach den theoretischen Einführungen und Betrachtungen soll eine Anwendung der Abstandsmaße auf simulierte Daten zu einem besseren Verständnis dieser und der gesamten Methode führen. Dabei werden die künstlichen Daten mit Hilfe von *Poisson-Prozessen* erzeugt, in deren Theorie kurz eingeführt wird.

Abschließend wird dann eine Anwendung auf retinale Daten erfolgen, die zwei Ziele hat. Zum einen soll mit Hinblick auf spätere Untersuchungen herausgefunden werden, welches der Verfahren sich am besten eignet. Dies bezieht sich insbesondere auf die verschiedenen Abstandsmaße, aber auch auf die Maße Prozent-Korrekt und wechselseitige Information, mit denen die Güte des stimulusabhängigen Clusters bewertet wird. Zum anderen sollen erste Aussagen über mögliche Kodierungsstrategien in den betrachteten Datensätzen gemacht werden. Hierbei steht im Vordergrund die Zeitskala zu ermitteln, die für eine bestmögliche Entschlüsselung der Information über die Reize Geschwindigkeit (mit unterschiedlicher Richtung) und Geschwindigkeitsübergänge durch das visuelle System zu-

²aus [ES00], S.4: „Clustering: Ziel des Clustering ist die Partitionierung einer Datenbank in Gruppen (Cluster) von Objekten, so dass Objekte eines Clusters möglichst ähnlich, Objekte verschiedener Cluster möglichst unähnlich sind.“

³aus [ES00], S.4: „Klassifikation: Gegeben sind hier Trainingsobjekte mit Attributwerten, die bereits einer Klasse zugeordnet sind. Es soll eine Funktion gelernt werden, die zukünftige Objekte auf Grund ihrer Attributwerte einer der Klassen zuweist.“

⁴Wie es etwa in [H00] ausführlich behandelt wird.

grunde gelegt werden sollte. Dies beinhaltet auch die Fragestellung nach unterschiedlicher Kodierung der Zelltypen hinsichtlich ihrer Richtungssensitivität. Ausserdem sollen erste Schlüsse darüber gewonnen werden, ob ein Zusammenwirken zweier Zellen zu einer besseren Kodierung führen kann und im gegebenen Fall unter welchen Bedingungen. Dabei werden die Aspekte Zeitskala und Wichtigkeit der Unterscheidung der Spikes nach Neuron betrachtet.

2 Abstandsmaße

2.1 $D^{euclid}[\tau]$

Zunächst wird für Spiketrains, also Folgen von Aktionspotentialzeitpunkten, das Abstandsmaß $D^{euclid}[\tau]$ nach Kretzberg et al. ([KWE01]) definiert. Hierbei wird zunächst aus einem Spiketrain S ein Vektor gebildet, der die Mittelwerte der Spikeanzahlen pro Sekunde innerhalb eines Zeitfensters der Größe τ , das um v Sekunden verschoben wird, enthält. Der Abstand zwischen zwei Antworten entspricht dann einer um die Anzahl an Datenpunkten standardisierten euklidischen Norm der so gefilterten Antworten.

Definition 2.1.1 ($D^{euclid}[\tau]$)

Sei $\mathcal{S} = \{\emptyset\} \cup \{\{T_1, T_2, \dots, T_k\} : 0 \leq T_1 < T_2 < \dots < T_k \leq z, k \in \mathbb{N}, z \in \mathbb{R}_{\geq 0}\}$ die Menge aller Spiketrains und seien $S, S' \in \mathcal{S}$ zwei Spiketrains mit $|S| = M$. Außerdem sei τ das Zeitfenster in Sekunden, das um v Sekunden verschoben wird.

Dann sei X der Vektor mit n -ten Eintrag x_n , wobei mit $T_i \in S$ gilt:

$$x_n = \frac{1}{\tau} \sum_{i=1}^M I_{[(n-1)v, (n-1)v+\tau]}(T_i), \quad \text{für } n = 1, \dots, \frac{z - \tau + v}{v}$$

Sei X' dementsprechend. Dann wird definiert:

$$D^{euclid}[\tau](S, S') = \sqrt{\frac{v}{z - \tau + v}} \cdot \|X - X'\|_2$$

Dabei gelte $\frac{0}{0} = 1$.

Die Normierung mit der Anzahl an Datenpunkten bewirkt, dass die Größenordnung der Abstände $D^{euclid}[\tau_1](S, S')$ und $D^{euclid}[\tau_2](S, S')$ mit $\tau_1 \neq \tau_2$ gleich ist. Dies wird sich aber für das an späterer Stelle eingeführte Verfahren des stimulusabhängigen Clusters als unwichtig erweisen.

Bemerkung 2.1.2

Der Parameter τ stellt die Zeitfenstergröße dar, in dem das Vorhandensein eines Spikes von Bedeutung ist. Je größer man ihn wählt, desto unwichtiger wird die zeitliche Präzision des Spiketimings. Ist τ gerade so groß wie der Zeitraum z , in dem die Spikezeitpunkte

ermittelt wurden (und v somit gleich Null), so ist der Abstand gerade die Differenz der Spikeanzahlen dividiert durch τ , also pro Sekunde. Der andere extreme Fall ist, wenn τ gleich der Verschiebung v und der Genauigkeit, mit der die Spikezeitpunkte vorliegen, gewählt wird. Denn dann ergibt $D^{euclid}[\tau]$ gerade die Wurzel aus der Anzahl der zeitlich nicht zusammenfallenden Aktionspotentiale multipliziert mit $\sqrt{\tau/z}$.

Das folgende Beispiel möge die vorhergehenden Erläuterungen veranschaulichen.

Beispiel 2.1.3

Seien etwa $S_a = \{0.43\}$ und $S_b = \{0.31, 0.7\}$ zwei Spiketrains mit Zeitpunkten in Sekunden, wobei diese in einem Zeitraum von $z = 1s$ ermittelt worden sein sollen. Dann erhält man für

a) $\tau = 1s$:

$$X_a = 1 \text{ und } X_b = 2 \text{ und}$$

$$D^{euclid}[1](\{0.43\}, \{0.31, 0.7\}) = \sqrt{1} \cdot \|1 - 2\|_2 = 1$$

b) $\tau = 0.2s, v = 0.1s$:

$$X_a = [0, 0, 0, 5, 5, 0, 0, 0, 0] \text{ und } X_b = [0, 0, 5, 5, 0, 5, 5, 0, 0] \text{ und}$$

$$\begin{aligned} & D^{euclid}[0.2](\{0.43\}, \{0.31, 0.7\}) \\ &= \sqrt{\frac{0.1}{1 - 0.2 + 0.1}} \cdot \|[0, 0, 0, 5, 5, 0, 0, 0, 0] - [0, 0, 5, 5, 0, 5, 5, 0, 0]\|_2 \\ &= \frac{1}{3} \cdot \sqrt{25 + 25 + 25 + 25} \\ &= \frac{10}{3} \end{aligned}$$

c) $\tau = 0.01s, v = 0.01s$:

$$X_a = \underbrace{[0, \dots, 0]}_{42\text{-mal}}, 1, \underbrace{[0, \dots, 0]}_{57\text{-mal}} \text{ und } X_b = \underbrace{[0, \dots, 0]}_{30\text{-mal}}, 1, \underbrace{[0, \dots, 0]}_{38\text{-mal}}, 1, \underbrace{[0, \dots, 0]}_{30\text{-mal}} \text{ und}$$

$$D^{euclid}[0.01](\{0.43\}, \{0.31, 0.7\}) = \sqrt{1} \cdot \|X_a - X_b\|_2 = \sqrt{3}$$

Im Fall $\tau = v$ entspricht $D^{euclid}[\tau]$ qualitativ der Hamming-Distanz ([H00]).

Definition 2.1.4 (Hamming-Distanz)

Seien $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ und $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)^T$ zwei Zufallsvektoren mit endlicher Wertemenge $\mathcal{Z}^n = \{z_1, z_2, \dots, z_k\}^n$. Dann ist die Hamming-Distanz definiert als die Anzahl paarweise unterschiedlicher Realisierungen:

$$D^H(\mathbf{X}, \mathbf{Y}) = |\{i : x_i \neq y_i, 1 \leq i \leq n\}|$$

Bemerkung 2.1.5

Für $\tau = v$ gilt mit den Bezeichnungen aus Definition 2.1.1 und $\mathcal{Z} = \{0, 1\}$:

$$D^{euclid}(S, S')[\tau] = \sqrt{\frac{\tau}{z} \cdot D^H(X, X')}$$

2.2 $D^{spike}[q]$ und $D^{interval}[q]$

2.2.1 Definition und Berechnung

Victor und Purpura ([VP96], [VP97], [V05]) haben eine Metrik konstruiert, die den minimalen 'Betrag' darstellt, um einen Spiketrain S in sogenannten elementaren Schritten in einen Spiketrain S' umzuwandeln. Dabei werden die elementaren Umformungen mit verschiedenen Kosten assoziiert. Die allgemeinste Definition lautet wie folgt.

Definition 2.2.1

Sei $\mathcal{S} = \{\emptyset\} \cup \{\{T_1, T_2, \dots, T_k\} : 0 \leq T_1 < T_2 < \dots < T_k, T_i \in \mathbb{R}_{\geq 0} \text{ f\"ur } 1 \leq i \leq k, k \in \mathbb{N}\}$ die Menge aller Spiketrains. Ausserdem sei $c : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ eine symmetrische Abbildung, die nur f\"ur zwei Folgen von Spikezeitpunkten S_j und $S_{j-1} \in \mathcal{S}$ definiert sei, die sich durch einen elementaren Schritt ineinander umformen lassen. Dann wird f\"ur $S, S' \in \mathcal{S}$ definiert:

$$D(S, S') := \inf_{S_0, S_1, \dots, S_r} \sum c(S_j, S_{j-1})$$

wobei $S_0 = S$, $S_r = S'$ gilt und S_1, S_2, \dots, S_{r-1} Spiketrains auf dem Weg von S_0 nach S_r darstellen.

Im Folgenden wird, sofern nicht anders angegeben, immer auf die zuvor genannten Arbeiten von Victor et al. Bezug genommen. Des Weiteren werden stets Spiketrains mit Zeitpunkten in Sekunden betrachtet.

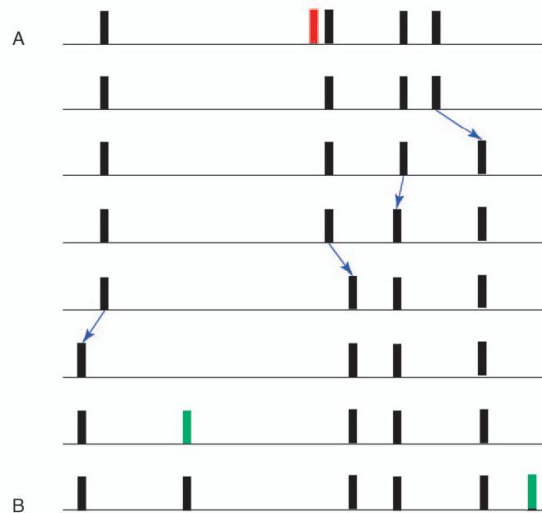


Abbildung 2.1: Veranschaulichung von D^{spike} . Der Spiketrain A wird in elementaren Schritten in Spiketrain B umgeformt. Jeder Balken stellt einen Spike dar, wobei die Zeit durch eine Linie repräsentiert wird. Eine Verschiebung wird durch einen blauen Pfeil, das Löschen eines Spikes durch einen roten Balken und das Einfügen durch einen grünen Balken angezeigt. (aus [V05])

Zunächst wird das Maß $D^{spike}[q]$ konstruiert, dem die Idee zugrunde liegt, dass das Timing eines einzelnen Spikes wichtig ist. Die möglichen elementaren Schritte sind dabei das Löschen oder Einfügen eines Spikes und das Verschieben eines Spikes in der Zeit, wie es in Abbildung 2.1 beispielhaft veranschaulicht ist. Die zeitliche Präzision, die für das Spiketiming relevant ist, wird durch einen Parameter q repräsentiert.

Definition 2.2.2 ($D^{spike}[q]$)

Mit den Bezeichnungen aus Definition 2.2.1 und $q \in \mathbb{R}_{\geq 0}$ mit einer Einheit von s^{-1} gelte:

$$c_{sp}(S_j, S_{j-1}) = \begin{cases} 1 & \text{Angleichen von } S_j \text{ und } S_{j-1} \text{ durch} \\ & \text{das Löschen oder Einfügen eines Spikes} \\ q|\Delta t| & \text{Angleichen von } S_j \text{ und } S_{j-1} \text{ durch} \\ & \text{das Verschieben eines Spikes um } \Delta t \end{cases}.$$

Dann wird definiert:

$$D^{spike}[q](S, S') := \inf_{S_0, S_1, \dots, S_r} \sum c_{sp}(S_j, S_{j-1})$$

Beispiel 2.2.3

Wie in Beispiel 2.1.3 seien $S = \{0.43\}$ und $S' = \{0.31, 0.7\}$ und ausserdem $q = 1$. Dann ist der minimale Weg gegeben durch: $S_0 = S$, $S_1 = \{0.31\}$ und $S_2 = S'$ und als Abstand erhält man: $D^{spike}[1](\{0.43\}, \{0.31, 0.7\}) = 1 \cdot 0.12 + 1 = 1.12$.

Bemerkung 2.2.4

Um sich der Bedeutung des Parameters q klar zu werden, betrachte man etwa zwei Spike-trains $S = \{a\}$ und $S' = \{b\}$ mit $a \neq b$. Dann ist $D^{spike}[q](S, S') = \min\{q|a - b|, 2\}$. Wenn $|a - b| > 2/q$ gilt, ist es also günstiger einen Spike zu löschen und wieder einzufügen, als ihn zu verschieben. Demnach ist es genauso teuer ein einzelnes Aktionspotential um $1/q$ zu verschieben wie es zu löschen. Das bedeutet, dass $1/q$ gerade dem Parameter τ des Abstandmaßes $D^{euclid}[\tau]$ entspricht. Wird also ein Abstand mit $D^{spike}[q]$ bestimmt, so drückt $1/q$ aus, welche zeitliche Präzision eines Spikezeitpunktes betrachtet wird. Die Parameter $1/q$ und τ geben also die bei einem Spiketrain betrachtete Zeitskala beziehungsweise zeitliche Auflösung an.

Des Weiteren wird ein Maß $D^{interval}[q]$ eingeführt, bei dem Interspikeintervalle, also die Zeit zwischen je zwei Spikes, betrachtet wird. Motiviert wird dieses Abstandsmaß durch neurobiologische Beispiele, in denen der Effekt eines Aktionspotentials stark von der zeitlichen Differenz zu vorhergehenden Spikes abhängt. Hierbei stellen das Löschen oder

Einfügen eines Interspikeintervalls elementaren Schritte dar. Ebenfalls kann zum Umformen einer Folge von Interspikeintervallen in eine andere die Länge eines Intervalls geändert werden, wobei die übrigen Intervalle unberührt bleiben.

Definition 2.2.5 ($D^{\text{interval}[q]}$)

Seien S_a und S_b wie zuvor aus der Menge aller Spiketrains \mathcal{S} und S_a^I und S_b^I die je darauf basierenden Folgen der Interspikeintervalle. Seien S_j^I und S_{j-1}^I zwei Folgen von Interspikeintervallen, die durch eine elementare Umformung angeglichen werden können.

Mit $q \in \mathbb{R}_{\geq 0}$ und

$$c_{\text{int}}(S_j^I, S_{j-1}^I) := \begin{cases} 1 & \text{Angleichen von } S_j \text{ und } S_{j-1} \text{ durch das} \\ & \text{Löschen oder Einfügen eines Interspikeintervalls} \\ q|\Delta t| & \text{Angleichen von } S_j \text{ und } S_{j-1} \text{ durch die} \\ & \text{Längenänderung eines Interspikeintervalls um } \Delta t \end{cases}$$

ist dann

$$D^{\text{interval}[q]}(S_a^I, S_b^I) := \inf_{S_0^I, S_1^I, \dots, S_r^I} \sum c_{\text{int}}(S_j^I, S_{j-1}^I).$$

Zur Bestimmung einer Interspikeintervallfolge S^I aus bei einem Versuch gewonnenen Spiketrain S gibt es verschiedene Möglichkeiten (siehe [VP97]). Sei etwa

$S = \{T_1, T_2, \dots, T_n\}$, dann ist es naheliegend $S^I = \{T_2 - T_1, T_3 - T_2, \dots, T_n - T_{n-1}\}$ zu setzen, also $D^{\text{interval:ign}}$ nach Victor ([VP97]) zu berechnen. Sind jedoch leere Spiketrains vorhanden, gibt es keine zu bevorzugende Art die Interspikeintervalle fest zu legen. Seien zum Beispiel $S_a = \{T_1\}$, $S_b = \{T_2, T_3\}$ und $S_c = \{\}$. Dann könnte etwa $S_a^I = \{0\}$, $S_b^I = \{T_3 - T_2\}$ und $S_c^I = \{\}$ sein. Dies würde bedeuten, dass $D^{\text{interval}}(S_a^I, S_c^I) = D^{\text{interval}}(S_b^I, S_c^I) = 1$. Es würde also keinen Unterschied machen, ob ein Neuron ein- oder zweimal gefeuert hat. Ebenfalls möglich wäre es, $S_a^I = \{\}$, $S_b^I = \{T_3 - T_2\}$ und $S_c^I = \{\}$ zu setzen. Das hieße, dass $D^{\text{interval}}(S_a^I, S_c^I) = 0$ und $D^{\text{interval}}(S_b^I, S_c^I) = 1$ wäre. In diesem Fall würde es also als gleichwertig angesehen werden, wenn ein Neuron gar kein oder nur ein Aktionspotential generiert hätte. Daher macht es Sinn, für jeden Spiketrain den gleichen ersten und letzten Spikezeitpunkt fest zu legen. Im Folgenden wird der erste Spikezeitpunkt immer auf 0 und der letzte auf das Ende des Zeitraums, in dem die neuronale Antwort ermittelt wurde, gesetzt.

Beispiel 2.2.6

Wie in Beispiel 2.1.3 seien wieder $S_a = \{0.43\}$ und $S_b = \{0.31, 0.7\}$ in einem Zeitraum von einer Sekunde ermittelt worden. Dann ist $S_a^I = \{0.43, 0.57\}$ und

$S_b^I = \{0.31, 0.39, 0.3\}$. Mit $q = 1$ ist der minimale Weg gegeben durch $S_0^I = S_a^I$, $S_1^I = \{0.31, 0.57\}$, $S_2^I = \{0.31, 0.39\}$, $S_3^I = S_b^I$ zu einem Preis von

$$D^{\text{interval}[1]}(\{0.43, 0.57\}, \{0.31, 0.7\}) = 1 \cdot 0.12 + 1 \cdot 0.18 + 1 = 1.3.$$

Ein auf den Arbeiten von Sellers [Se74] für genetische Sequenzen basierender Algorithmus gewährleistet eine effiziente Berechnung der beiden Maße.

Algorithmus 2.2.7

Seien $S_a = \{a_1, \dots, a_m\}$, $S_b = \{b_1, \dots, b_n\} \in \mathcal{S}$. Dann setze: $G_{i,0} := i$ für $0 \leq i \leq m$, $G_{0,j} := j$ für $0 \leq j \leq n$ und

$$G_{i,j} = \min \{G_{i-1,j} + 1, G_{i,j-1} + 1, G_{i-1,j-1} + q |a_i - b_j|\} \text{ für } 1 \leq i, j \leq m$$

Dann gilt: $D^{spike}[q](S_a, S_b) = G_{m,n}$.

Beweisskizze

Die Anfangssetzungen gelten per Definition. So ist der minimale Preis, um einen leeren Spiketrain in einen anderen Spiketrain zu überführen, gerade die Anzahl der Spikes des zweiten Spiketrains: $D^{spike}[q](\{\}, \{\}) = 0$, $D^{spike}[q](\{a_1, \dots, a_i\}, \{\}) = |\{a_1, \dots, a_i\}| = i = G_{i,0}$ für $1 \leq i \leq m$ und $D^{spike}[q](\{\}, \{b_1, \dots, b_j\}) = j = G_{0,j}$ für $1 \leq j \leq n$.

Sei ein mit minimalem Preis assoziierter Weg $S_a = S_0, S_1, S_2, \dots, S_b = S_r$ gefunden. Dann kann man die „Entstehungsgeschichte“ jedes Spikes in Form eines Diagramms ähnlich Abbildung 2.1 darstellen. Dabei sei die Änderung der ursprünglichen Position eines Spikes in die neue Position in Form einer „Trajektorie“ gekennzeichnet (wie blaue Pfeile in 2.1). Es ergeben sich einige Beschränkungen dieses Diagramms unter der Annahme, dass es sich hierbei um den minimalen Weg handelt. So besteht keine Notwendigkeit einen Fall zu betrachten, bei dem ein Spike von S_a auf eine Position verschoben wird und dann in einem zweiten Schritt auf seine endgültige Position in S_b bewegt wird. Des Weiteren kann es nicht vorkommen, dass ein Spike in mehr als eine Richtung verschoben wird. Beim Verschieben zweier Aktionspotentiale kommt es zu keinem Überkreuzen der entsprechenden Trajektorien. Hieraus ergeben sich drei Möglichkeiten, die gerade den induktiven Schritt des Algorithmus darstellen: a) Der letzte Spike des Spiketrains S_a ist ein Spike, der gelöscht werden muss. b) Der letzte Spike in S_b ist ein Spike, der eingefügt werden muss. c) Die je letzten Spikes beider Folgen müssen über ein Verschieben einer der beiden Aktionspotentiale angeglichen werden. \square

Bemerkung 2.2.8

Während das Abstandsmaß $D^{spike}[q]$ sich nicht direkt auf das von Sellers konstruierte Maß übertragen lässt, ist dies für $D^{interval}[q]$ durchaus möglich. So entsprechen die Interspikereintervallfolgen den genetischen Sequenzen, deren Abstand gerade die minimale gewichtete Summe bestehend aus gewichteten Mutationen und Löschungen von Folgeelementen, die zu einem Angleichen der Sequenzen führen, darstellt. Der elementare Schritt des Löschens ist hier ebenso enthalten, wie die Längenänderung eines Interspikereintervalls in Form einer Mutation.

Die Berechnung von $D^{interval}[q]$ erfolgt wie im Algorithmus 2.2.7 dargestellt, wobei an Stelle der Spikezeitpunkte die Interspikeintervalle betrachtet werden.

Für $q = 0$ entsprechen die beiden Maße je gerade der Differenz der Spikeanzahlen und für $q \rightarrow \infty$ ist $D^{spike}[q]$ gerade die Anzahl der zeitlich nicht zusammenfallenden Aktionspotentiale der beiden Spiketrains.

Definition und Lemma 2.2.9

Definiere $D^{spike}[\infty](S_a, S_b) := \lim_{q \rightarrow \infty} D^{spike}[q](S_a, S_b)$ und $D^{count}(S_a, S_b) :=$

$D^{spike}[0](S_a, S_b)$. Sei $|S_a| = k$ und $|S_b| = l$ mit $|S_a \cap S_b| = m$.

Dann ist $D^{count}(S_a, S_b) = D^{interval}[0](S_a^I, S_b^I) = |k - l|$ und $D^{spike}[\infty](S_a, S_b) = k + l - 2m$.

Beweis

Für $q = 0$ gilt

$$c_{sp}(S_j, S_{j-1}) = c_{int}(S_k^I, S_{k-1}^I) = \begin{cases} 1 & \text{Löschen/Einfügen} \\ 0 & \text{Verschieben/Längenänderung} \end{cases}$$

und für $q \rightarrow \infty$ ist

$$c_{sp}(S_j, S_{j-1}) = \begin{cases} 1 & \text{Löschen/Einfügen eines Spikes} \\ \infty & \text{Verschieben um } \Delta t \text{ mit } |\Delta t| > 0 \\ 0 & \text{Verschieben um } \Delta t \text{ mit } |\Delta t| = 0 \end{cases}$$

Zusammen mit der Tatsache, dass die Anzahl der Interspikeintervalle je Spiketrain gerade gleich der Anzahl der Spikes plus 1 ist, folgt nun nach Definition und Algorithmus 2.2.7 die Aussage. \square

2.2.2 Eigenschaften

Im Folgenden werden die Abstandsmaße $D^{spike}[q]$ und $D^{interval}[q]$ charakterisiert. So bilden diese je zusammen mit der Menge der Spiketrains einen pseudometrischen Raum. Sie erfüllen also alle Eigenschaften einer Metrik bis auf diejenige, dass bei Vorliegen eines Abstands von Null folgt, dass die betrachteten Elemente gleich sind.

Satz 2.2.10

$D^{spike}[q]$ und $D^{interval}[q]$ sind Pseudometriken.

Der Beweis dieser Aussage stellt eine Formalisierung und geringfügige Erweiterung der Begründungen von Victor et al. ([VP97]) dar.

Beweis

Mit den Bezeichnungen aus den vorhergehenden Definitionen stehe im Folgenden $D[q]$ sowohl für $D^{spike}[q]$ als auch für $D^{interval}[q]$ und $c(S_j, S_{j-1})$ sowohl für $c_{sp}(S_j, S_{j-1})$ als auch für $c_{int}(S_j^I, S_{j-1}^I)$.

Seien $S_a, S_b \in \mathcal{S}$. Mit $S_0 = S_a, S_r = S_b$ folgt aus $c(S_j, S_{j-1}) \geq 0$, dass $D[q](S_a, S_b) =$

$$\inf_{S_0, S_1, \dots, S_r} \sum c(S_j, S_{j-1}) \geq 0 \text{ gilt.}$$

Die Symmetrie von $D[q]$ folgt aus der Symmetrie von $c(S_j, S_{j-1})$: $D[q](S_a, S_b) =$

$$\inf_{S_0, S_1, \dots, S_r} \sum c(S_j, S_{j-1}) = \inf_{S_0, S_1, \dots, S_r} \sum c(S_{j-1}, S_j) = D[q](S_b, S_a)$$

Die Dreiecksungleichung ergibt sich aus der Konstruktion von $D[q]$: Seien $S_a, S_b, S_c \in \mathcal{S}$ und ausserdem $S_0^*, S_1^*, \dots, S_t^*$ die Spiketrains mit $S_a = S_0^*$ und $S_c = S_t^*$, so dass gilt: $D[q](S_a, S_c) = \sum_{j=1}^t c(S_j^*, S_{j-1}^*)$. Erster Fall: Es gibt ein $k \in \{0, \dots, t\}$ mit $S_k^* = S_b$. Dann ist $D[q](S_a, S_c) = D[q](S_a, S_b) + D[q](S_b, S_c)$, denn sonst wäre $S_0^*, S_1^*, \dots, S_t^*$ nicht der minimale Weg gewesen. Zweiter Fall: Es gibt kein $k \in \{0, \dots, t\}$ mit $S_k^* = S_b$. Dann gilt per Definition von $D[q]$ aber sicherlich $D[q](S_a, S_c) < D[q](S_a, S_b) + D[q](S_b, S_c)$. \square

Lemma 2.2.11

D^{count} ist keine Metrik.

Beweis

Seien etwa $|S| = |S'| = n$ mit $|S \cap S'| \leq n - 1$. Die Spiketrains S und S' unterscheiden sich also mindestens in einem Zeitpunkt der Aktionspotentiale. Mit Lemma 2.2.9 ergibt sich $D^{count}(S, S') = 0$, aber es ist $S \neq S'$. \square

Hätte man $D^{interval}$ nicht auf den Folgen der Interspikeintervallen definiert, sondern auf den Spiketrains und würde mit $D^{interval:ign}$ (siehe nach Definition 2.2.5) rechnen, so wäre $D^{interval}[q]$ für $q \in \mathbb{R}_{\geq 0}$ keine Metrik. Denn sei etwa $S_a = \{1, 2, \dots, n\}$ und $S_b = \{1 + c, 2 + c, \dots, n + c\}$ mit einem $c \in \mathbb{R}^+$, dann ist $D^{interval}[q](S_a^I, S_b^I) = D^{interval}[q](\underbrace{\{1, 1, \dots, 1\}}_{(n-1)\text{-mal}}, \underbrace{\{1, 1, \dots, 1\}}_{(n-1)\text{-mal}}) = 0$, aber es gilt $S_a \neq S_b$.

Wie es in der Analysis eine gängige Vorgehensweise ist und von Victor et al. angemerkt wird, werden die Pseudometriken über die Bildung von Äquivalenzklassen wie folgt zu Metriken.

Lemma 2.2.12

Durch \sim_D mit $S_i \sim_D S_j :\Leftrightarrow D^{spike}[q](S_i, S_j) = 0$ wird eine Äquivalenzrelation auf der Menge aller Spiketrains \mathcal{S} definiert.

Beweis

Der Beweis lässt sich mit den Pseudometrikeigenschaften von $D^{spike}[q]$ leicht durchführen:

$$\text{reflexiv: } S_i \in \mathcal{S} \Rightarrow D^{spike}(S_i, S_i) = 0 \Rightarrow S_i \sim_D S_i$$

$$\text{symmetrisch: } S_i, S_j \in \mathcal{S} : S_i \sim_D S_j \Rightarrow D^{spike}[q](S_i, S_j) = 0$$

$$\Rightarrow D^{spike}[q](S_j, S_i) = 0 \Rightarrow S_j \sim_D S_i$$

$$\text{transitiv: } S_i, S_j, S_k \in \mathcal{S} : S_i \sim_D S_j \wedge S_j \sim_D S_k$$

$$\Rightarrow D^{spike}[q](S_i, S_j) = 0 \wedge D^{spike}[q](S_j, S_k) = 0$$

$$\Rightarrow D^{spike}[q](S_i, S_k) \leq D^{spike}[q](S_i, S_j) + D^{spike}[q](S_j, S_k) = 0$$

$$\Rightarrow D^{spike}[q](S_i, S_k) = 0 \Rightarrow S_i \sim_D S_k \quad \square$$

Satz 2.2.13

Mit den Äquivalenzklassen $[S_i] = \{S : D^{spike}[q](S, S_i) = 0\}$ sei $\mathcal{S} = \bigoplus_i [S_i]$. Dann bildet \mathcal{S} zusammen mit $D^{spike}[q]^*([S_i], [S_j]) := D^{spike}[q](S_i, S_j)$ einen metrischen Raum.

Beweis

Symmetrie und Dreiecksungleichung folgen direkt aus den Eigenschaften von $D^{spike}[q]$, somit muss nur noch die positive Definitheit gezeigt werden:

$$\text{Seien } S_i, S_j \in \mathcal{S}, \text{ dann: } D^{spike}[q]^*([S_i], [S_j]) = 0 \Leftrightarrow D^{spike}[q](S_i, S_j) = 0$$

$$\Leftrightarrow S_i \in [S_j] \Leftrightarrow [S_i] \cap [S_j] \neq \emptyset \Leftrightarrow [S_i] = [S_j]. \quad \square$$

Die Aussagen 2.2.12 und 2.2.13 gelten natürlich auch für $D^{interval}[q]$. Im Folgenden bezeichne S_i immer die Äquivalenzklasse $[S_i]$, dementsprechend sei $D^{spike}[q]$ immer $D^{spike}[q]^*$ und $D^{interval}[q]$ immer $D^{interval}[q]^*$.

Um die Zusammenhänge zwischen den verschiedenen Metriken charakterisieren zu können, werden nun zwei Begriffe aus [G64] eingeführt.

Definition 2.2.14

Seien (\mathcal{S}, D_1) und (\mathcal{S}, D_2) metrische Räume und $S, S' \in \mathcal{S}$.

(i) Eine Metrik D_1 verfeinert eine Metrik D_2 , wenn gilt:

$$\forall \epsilon > 0 \exists \delta > 0 : D_1(S, S') < \delta \Rightarrow D_2(S, S') < \epsilon$$

(ii) Die Metriken D_1 und D_2 heißen äquivalent, wenn D_1 die Metrik D_2 verfeinert und D_2 die Metrik D_1 verfeinert.

Nun können die Aussagen nach Victor et al. ([VP97]) formuliert werden. Die Beweise sind dort auch entnommen, jedoch an einigen Stellen leicht abgeändert beziehungsweise ergänzt worden.

Satz 2.2.15

Für $0 < q_a < q_b$ sind sowohl $D^{spike}[q_a]$ und $D^{spike}[q_b]$ als auch $D^{interval}[q_a]$ und $D^{interval}[q_b]$ äquivalent.

Beweis

Der Beweis wird nur für D^{spike} geführt, für $D^{interval}$ funktioniert er analog. Seien $S, S' \in \mathcal{S}$ und $0 < q_a < q_b$.

$D^{spike}[q_b]$ verfeinert $D^{spike}[q_a]$: Sei $\epsilon > 0$. Sicher gilt $D^{spike}[q_a](S, S') \leq D^{spike}[q_b](S, S')$. Also setze $\delta := \epsilon$, dann gilt: $D^{spike}[q_b](S, S') < \epsilon \Rightarrow D^{spike}[q_a](S, S') < \delta$.

$D^{spike}[q_b]$ verfeinert $D^{spike}[q_a]$: Sei $\epsilon > 0$ und $\delta < \min(\frac{\epsilon q_a}{q_b}, 1)$. Damit gilt

$D^{spike}[q_a](S, S') < \delta < 1$. Daraus folgt, dass die elementaren Schritte bei der Umformung von S in S' nur aus Verschiebungen von Spikes bestehen können, da das Löschen bzw. Einfügen eines Spikes bereits einen Preis von 1 hat. Also:

$D^{spike}[q_a](S, S') = \sum_i q_a |\Delta_i t| < \delta$ und wegen der Wahl von δ folgt: $\sum_i |\Delta_i t| < \frac{\delta}{q_a} < \frac{\epsilon}{q_b}$ und somit $D^{spike}[q_b](S, S') \leq \sum_i q_b |\Delta_i t| < \epsilon$. \square

Satz 2.2.16

$D^{spike}[q]$ verfeinert $D^{interval}[q]$ für $q > 0$, aber die beiden Metriken sind nicht äquivalent.

Beweis

$D^{spike}[q]$ verfeinert $D^{interval}[q]$: Die Verschiebung eines Spikes um Δt zu einem Preis von $q|\Delta t|$ in $D^{spike}[q]$ kann ebenso durch die Längenänderung zweier Interspikeintervalle in $D^{interval}[q]$ realisiert werden. Das vorangehende Intervall wird um Δt und das anschließende um $-\Delta t$ geändert, insgesamt also zu einem Preis von $2q|\Delta t|$. Analog wird das Verschieben eines Spikes um $-\Delta t$ umgesetzt. Das Einfügen eines Spikes in $D^{spike}[q]$ kann mit einer maximalen Kost von 3 in $D^{interval}[q]$ durch das Löschen von zwei Interspikeintervallen und das Einfügen von einem entsprechend umgewandelt werden. Analoges gilt für das Löschen eines Spikes. Setze also $\delta = \frac{\epsilon}{3}$, dann folgt die Behauptung.

$D^{interval}[q]$ verfeinert nicht $D^{spike}[q]$: Zeige dazu, dass es Folgen von Spiketrams $(S_n)_{n \in \mathbb{N}}$ und $(S'_n)_{n \in \mathbb{N}}$ gibt, so dass $D^{interval}[q](S_n, S'_n)$ gegen 0 konvergiert, aber

$D^{spike}[q](S_n, S'_n)$ nicht. Sei dazu $S_n := \{0, 1, 2, 3, \dots, (n-1), n\}$ und $S'_n := \{0, 1 + \frac{1}{n}, 2 + \frac{1}{n}, 3 + \frac{1}{n}, \dots, (n-1) + \frac{1}{n}, n + \frac{1}{n}\}$. Die n Spikezeitpunkte unterscheiden sich somit je um $\frac{1}{n}$. Mit T als letzten Spikezeitpunkt ist $S_n^I = \{0, \underbrace{1, \dots, 1}_{n\text{-mal}}, T - n\}$ und $S_n^{II} = \{0, 1 + \frac{1}{n}, \underbrace{1, \dots, 1}_{(n-1)\text{-mal}}, T - n - \frac{1}{n}\}$. Die je zweiten und letzten Interspikeintervalle unterscheiden sich also um $\frac{1}{n}$. Somit ergibt sich

$$\lim_{n \rightarrow \infty} D^{interval}[q](S_n^I, S_n^{II}) = \lim_{n \rightarrow \infty} 2 \cdot \min\{2, \frac{q}{n}\} = \lim_{n \rightarrow \infty} \frac{2q}{n} = 0,$$

da für n gegen unendlich $q < 2n$ gilt. Dagegen ist

$$\lim_{n \rightarrow \infty} D^{spike}[q](S_n, S'_n) = \lim_{n \rightarrow \infty} n \cdot \min\{2, \frac{q}{n}\} = \lim_{n \rightarrow \infty} n \cdot \frac{q}{n} = q.$$

\square

2.3 Multineuronale Erweiterung von $D^{spike}[q]$

2.3.1 Definition und Berechnung

Die bisher betrachteten Abstandsmaße beziehen sich immer nur auf die Antworten eines Neurons. Jedoch spielt für die neuronale Kodierung sicherlich das „Zusammenspiel“ mehrerer Zellen auch eine Rolle. Daher haben sich Victor et al. ([VP97], [A03], [AV04]) eine multineuronale Erweiterung von $D^{spike}[q]$ überlegt. Hierbei wird ein Spiketrain als eine Abfolge von Aktionspotentialzeitpunkten M verschiedener Nervenzellen betrachtet, die dementsprechend gekennzeichnet sind. Auch hier ist der Abstand definiert als der mit dem geringsten Preis assoziierte Weg, um einen Spiketrain einem anderen anzugleichen (siehe Abb. 2.3 (A)). Neben der zeitlichen Präzision q bringt dann der weitere Parameter k zum Ausdruck, ob eher das Entladungsmuster jedes einzelnen Neurons oder eher das der gesamten Population entscheidend ist für die Informationsübertragung.

Definition 2.3.1 ($D^{multi}[q, k]$)

Sei $\mathcal{S}_m = \{\emptyset\} \cup \{\{T_1^m, T_2^m, \dots, T_{n_m}^m\} : 0 \leq T_1^m < T_2^m < \dots < T_{n_m}^m, T_i^m \in \mathbb{R}_{\geq 0} \text{ für } 1 \leq i \leq n_m, n_m \in \mathbb{N}\}$ die Menge aller Spiketrains hervorgerufen durch ein mit m bezeichnetes Neuron für $1 \leq m \leq M$. Dann ist die Menge aller gekennzeichneten Spiketrains definiert als

$$\mathcal{S} = \{S : \exists S_m \in \mathcal{S}_m \text{ für } 1 \leq m \leq M : S = \bigcup_{m=1}^M S_m\}.$$

Seien S, S' zwei Spiketrains, die sich um eine elementare Umformung unterscheiden. Das heißt, sie können durch das Löschen oder Einfügen eines Spikes, durch das Verschieben eines Spikes in der Zeit um Δt oder durch das Ändern der Markierung eines Spikes von m in n angeglichen werden. Mit $q, k \in \mathbb{R}_{\geq 0}$ sei nun

$$c_{mu}(S, S') := \begin{cases} 1 & \text{Angleichen von } S \text{ und } S' \text{ durch das} \\ & \text{Löschen oder Einfügen eines Spikes} \\ q|\Delta t| & \text{Angleichen von } S \text{ und } S' \text{ durch das} \\ & \text{Verschieben eines Spikes um } \Delta t \\ k & \text{Angleichen von } S \text{ und } S' \text{ durch das} \\ & \text{Ändern der Kennzeichnung eines Spikes} \end{cases}.$$

Dann wird für $S_a, S_b \in \mathcal{S}$ definiert:

$$D^{multi}[q, k](S_a, S_b) := \min_{S_0, S_1, \dots, S_r} \sum c_{mu}(S_j, S_{j-1})$$

wobei $S_0 = S_a, S_r = S_b$ gilt und S_1, S_2, \dots, S_{r-1} Spiketrains auf dem Weg von S_0 nach S_r darstellen.

Bemerkung 2.3.2

Der Parameter k stellt die Wichtigkeit der Unterscheidung individueller Neurone dar. Im Fall $k = 0$ spielt es keine Rolle, von welcher Zelle das Aktionspotential hervorgerufen wurde. Genau entgegengesetzt verhält es sich für $k \geq 2$, denn dann ist es immer günstiger einen Spike zu löschen und einen anders gekennzeichneten einzufügen (Kosten von 2) als die Markierung zu ändern (Kosten von k). Also werden Antworten von unterschiedlichen Neuronen nie als gleich betrachtet. Wenn zwei unterschiedlich markierte Spikes zeitlich um $|\Delta t|$ auseinanderliegen ist der Abstand gerade gleich dem Minimum von $q|\Delta t| + k$ und 2. Für $0 < k < 2$ sollten sie also durch eine Verschiebung angeglichen werden, sofern sie innerhalb eines Zeitraums von $(2 - k)/q$ Sekunden hervorgerufen wurden.

Zur Berechnung des Abstands zwischen zwei gekennzeichneten Spiketrains kann der Algorithmus zu D^{spike} (vgl. Algorithmus 2.2.7) nicht direkt übertragen werden, da dieser Vorschriften beinhaltet, die im multineuronalen Fall zur Bestimmung des minimalen Wegs ungeeignet sind. Seien etwa zwei gekennzeichnete Spiketrains wie in Abbildung 2.2 und $k < 2$ gegeben. Hier stellen die Balken unterschiedlicher Färbung jeweils Spikes (zu den Zeitpunkten 0 beziehungsweise Δt) dar, die von einem der zwei Neurone gefeuert wurden. Sofern nun kein Überkreuzen zweier Trajektorien (Pfeile in Abb. 2.2) erlaubt wäre, würde der minimale Weg aus dem Ändern der Kennzeichnung zweier Spikes zu einem Preis von $2k$ bestehen (durchgezogene Pfeile). Das Überkreuzen (gestrichelte Pfeile) hätte dagegen einen Preis von $2q\Delta t$. Der minimale Weg zur Umformung der Spiketrains wird für $\Delta t < k/q$ also durch das Überkreuzen realisiert. Das Verbot des Überkreuzens muss also eingeschränkt werden auf den Fall, dass es sich um Trajektorien von Spikes gleicher Kennzeichnung handelt.

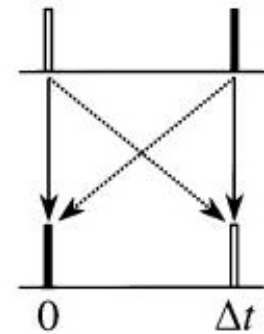


Abbildung 2.2: Einfaches Beispiel (aus [A03]). Erklärung siehe Text.

Man betrachte nun zwei Spiketrains S_a und S_b mit M verschiedenen Kennzeichnungen, um die Berechnung und somit den Algorithmus zu $D^{multi}[q, k]$ zu motivieren. Der Spiketrain S_b wird nun aufgefasst als Zusammensetzung von M Subspiketrains (siehe Abb. 2.3 B), die gerade aus den Spikes je einer Nervenzelle bestehen. Damit ergeben sich folgende Möglichkeiten einer Umformungen des letzten Spikes von S_a in einem minimalen Weg: Entweder er wird gelöscht oder er muss mit einem Spike in S_b über eine Verschiebung „verbunden“ werden. Er kann mit einem der letzten Spikes einer der Subtrains verknüpft werden, was M verschiedene Alternativen bedeutet. Zudem kann es sein, dass der letzte Spike in S_a mit einem Spike in S_b verbunden werden muss, der nicht der Letzte in dem jeweiligen Subspiketrain ist. Das bedeutet, dass das letzte Aktionspotential in dem entsprechenden Subtrain gelöscht werden muss, damit es nicht zu einem unerlaubten Überkreuzen der Trajektorien von Spikes einer Kennzeichnung kommt. Dieses Löschen stellt weitere M Möglichkeiten

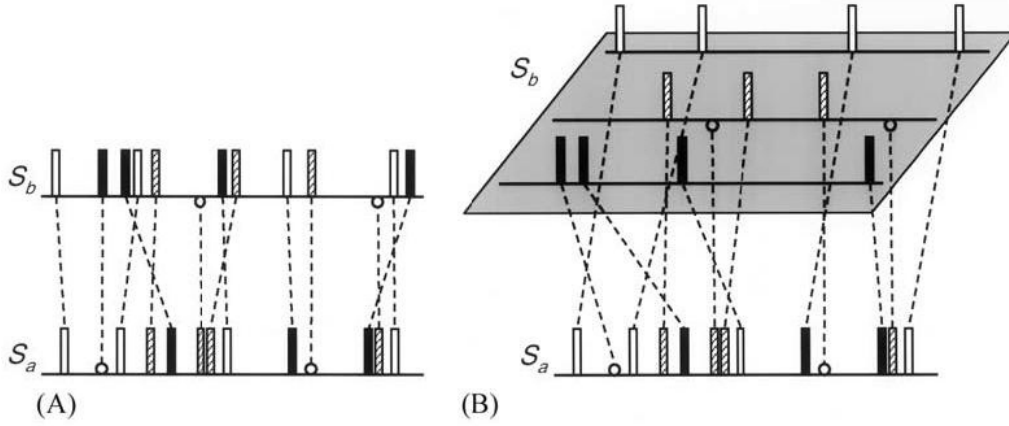


Abbildung 2.3: Algorithmus zur Berechnung von $D^{multi}[q, k]$. (A) Minimaler Weg zur Umformung zweier Spiketrains, die je aus den gleichzeitig abgeleiteten Antworten von drei Neuronen (unterschiedliche Balkenfärbung) bestehen. Gestrichelte Linien zwischen zwei Spikes stehen für die „Entstehungsgeschichte“: Entweder handelt es sich um eine Verschiebung und, sofern nötig, ein Ändern der Kennzeichnung oder um das Löschen/Einfügen eines Spikes (kleine Kreise). (B) Im Algorithmus wird ein Spiketrain, hier S_b , in Subspiketrains unterteilt, die je aus den Spikes eines der drei Neurone bestehen. Die jeweiligen Verbindungen zweier Spikes zum Angleichen von S_a und S_b auf einem minimalen Weg ist wie in (A) durch die gestrichelten Linien (Trajektorien) dargestellt. Dabei beachtet der Algorithmus, dass sich Trajektorien innerhalb einer durch S_a und einem der Subtrains von S_b gebildeteten Ebenen nicht überkreuzen dürfen. (aus [A03])

für den letzten Spike in S_a dar, so dass insgesamt $2M + 1$ (bei Vorliegen nicht-leerer Subspiketrains) Alternativen in einem induktiven Algorithmus betrachtet werden müssen. Mit diesen Überlegungen erhält man nun folgenden Algorithmus für $D^{multi}[q, k]$ nach [A03].

Algorithms 2.3.3

Seien S_a und S_b zwei gekennzeichnete Spiketrains, wobei die Zeitpunkte der Aktionspotentiale in S_a mit $\{a_1^m, a_2^m, \dots, a_{n_m}^m\}$ und in S_b mit $\{b_1^m, b_2^m, \dots, b_{k_m}^m\}$ für $m = 1, 2, \dots, M$ bezeichnet werden. Außerdem gelte für S_a folgende Notation: Die Zeitpunkte der Aktionspotentiale in S_a seien $\alpha_1, \alpha_2, \dots, \alpha_N$, $\alpha_1 < \alpha_2 < \dots < \alpha_N$ und r_1, r_2, \dots, r_N die zugehörigen Kennzeichnungen. Die Gesamtanzahl Spikes in S_a ist demnach $N = \sum n_m$. Bezeichne $G_{i;j_1, j_2, \dots, j_M}$ den Abstand zwischen dem Spiketrain bestehend aus den ersten i Spikes von S_a und dem Spiketrain bestehend aus den ersten j_m Spikes jeder Kennzeichnung m von S_b .

Setze: $G_{i;0,0,\dots,0} = i$ für $0 \leq i \leq N$, $G_{0;j_1, j_2, \dots, j_M} = \sum j_m$ für $0 \leq j_m \leq k_m$ mit $m = 1, \dots, M$ und

$$G_{i;j_1, j_2, \dots, j_M} = \min \left\{ \begin{array}{l} G_{i-1; j_1, \dots, j_M} + 1, \\ \min_{m, j_m > 0} G_{i-1; j_1, j_2, \dots, j_{m-1}, \dots, j_M} + q|\alpha_i - b_{j_m}^m| + k(1 - \delta_{m, r_i}), \\ \min_{m, j_m > 0} G_{i-1; j_1, \dots, j_{m-1}, \dots, j_M} + 1 \end{array} \right\}, \quad (2.1)$$

wobei $\delta_{ij} = 1$ für $i = j$ und 0 sonst. Dann gilt:

$$D^{multi}[q, k](S_a, S_b) = G_{N; k_1, k_2, \dots, k_M}$$

Dabei stellt die erste Zeile in 2.1 gerade den Fall dar, den letzten Spike in S_a zu löschen. Die M Möglichkeiten den letzten Spike in S_a mit einem in S_b über eine Verschiebung zu verbinden, sind in der zweiten Zeile umgesetzt. Die Bedingung $j_m > 0$ zeigt an, dass für diesen elementaren Schritt nur die nicht-leeren Subspiketrains betrachtet werden müssen. Das Kronecker-Delta gewährleistet, dass die Kosten für das Ändern einer Kennzeichnung im notwendigen Fall mit einbezogen werden. Die letzte Zeile entspricht dann den M Möglichkeiten, bei denen ein Spike von S_b gelöscht wird.

Bemerkung 2.3.4

Zur Bestimmung von $D^{multi}[q, k](S_a, S_b)$ wird also eine Tabelle G erstellt, für die so viele Einträge berechnet werden müssen, wie sich Kombinationsmöglichkeiten von i und j_m ergeben, die nicht durch die Anfangssetzungen gegeben sind. Ohne Betrachtung leerer Subspiketrains, also $1 \leq j_m \leq k_m$ anstelle von $0 \leq j_m \leq k_m$ für $m = 1, 2, \dots, M$, handelt es sich um $N \cdot \prod k_m$ Einträge. Ein Vertauschen der Rollen von S_a und S_b kann somit den Rechenaufwand unter Umständen verringern. Jeder dieser Einträge für $1 \leq j_m \leq k_m$ stellt das Minimum von $2M + 1$ zu berechnenden Werten dar. Besteht also jeder Subspiketrain durchschnittlich aus L Aktionspotentialzeitpunkten, sind, wie Aronov in [A03] schreibt, in etwa $(2M + 1) \cdot N \cdot L^M = (2M + 1)ML^{M+1}$ Berechnungen durchzuführen.

Um zum Beispiel Rechenzeiten hochzurechnen, empfiehlt es sich eine etwas genauere Approximation der Anzahl an Berechnungen zu verwenden. Insbesondere bei einer geringen mittleren Spikeanzahl L pro Subspiketrain sollten auch die Einträge basierend auf leeren Subspiketrains mit einbezogen werden. So müssen alle Einträge mit l leeren Subspiketrains für $1 \leq l \leq M - 1$ betrachtet werden. Hiervon gibt es jeweils $\binom{M}{l} \cdot L^{M-l}$ Stück in Kombination mit den ersten i von LM Spikes von S_a , für die lediglich $2(M - l) + 1$ Werte bestimmt werden müssen. Insgesamt erhalte ich so eine Abschätzung von

$$(2M + 1)ML^{M+1} + LM \cdot \sum_{l=1}^{M-1} (2(M - l) + 1) \cdot \binom{M}{l} \cdot L^{M-l} \quad (2.2)$$

durchzuführenden Berechnungen.

2.3.2 Nicht-Einbettbarkeit in euklidischen Raum

Bevor auf die Fragen der Einbettbarkeit eingegangen wird, sollten zunächst einige Begrifflichkeiten geklärt werden ([HKC83]). Dabei erfolgen die Definitionen für semimetrische

Räume, die mit den metrischen Räumen alle Eigenschaften bis auf die Dreiecksungleichung gemein haben ([B70]).

Definition 2.3.4 (Kongruenz)

Eine Kongruenz ist eine Funktion Φ , die einen semimetrischen Raum (\mathcal{S}, d_1) in einen anderen semimetrischen Raum (\mathcal{S}', d_2) abbildet, so dass für alle $x, y \in \mathcal{S}$ gilt:

$$d_1(x, y) = d_2(\Phi(x), \Phi(y))$$

Zwei semimetrische Räume heißen kongruent, wenn eine Kongruenz zwischen ihnen existiert.

Definition 2.3.5 (kongruent einbettbar)

Ein semimetrischer Raum \mathcal{S} heißt kongruent einbettbar in einen semimetrischen Raum \mathcal{T} , wenn er kongruent zu einer Teilmenge von \mathcal{T} ist.

Eine Veranschaulichung der Problemstellung der Einbettbarkeit liefert folgendes Beispiel ([AV04]). Zugleich wird es sich als Motivation für einen Teil des Beweises der Hauptaussage in diesem Kapitel erweisen.

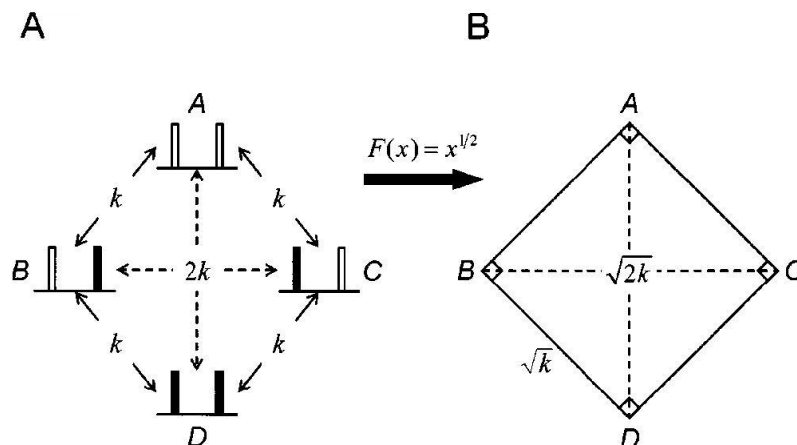


Abbildung 2.4: Beispiel nicht-einbettbarer Menge von Spiketrains. A Die vier Spiketrains bestehen je aus Antworten von zwei gleichzeitig abgeleiteten Neuronen (Balken mit unterschiedlicher Färbung). Die Aktionspotentiale in einem Spiketrain sind zeitlich um mindestens $2/q$ getrennt, so dass Verschiebungen in einem minimalen Weg zur Umformung nicht vorkommen. B Mit $F(x) = \sqrt{x}$ können die paarweisen Abstände so abgeändert werden, dass eine Abbildung der Spiketrains auf Ecken eines Quadrats im euklidischen Raum möglich ist. (aus [AV04])

Beispiel 2.3.6

Seien etwa vier durch zwei Neurone erzeugte Spiketrains A, B, C, D wie in Abbildung 2.4A gegeben. Diese bilden zwei Gruppen (A und D , sowie B und C), wobei gilt, dass der Abstand von Spiketrains innerhalb einer Gruppe stets $2k$ und verschiedener Gruppen stets k ist. Wie in Beispiel 2.3.14 nachgewiesen wird, ist es nicht möglich diese Spiketrains kongruent, also abstandserhaltend, in einen euklidischen Raum l_2^n einzubetten. Dies lässt sich

etwa für \mathbb{R}^2 veranschaulichen. So würden bei einem Einbetten zum Beispiel die Punkte A und D eine Gerade bilden, in deren Mitte der Punkt B liegen müsste. Damit der Punkt C aber zu A und D je einen Abstand von k hat, müsste er wie B genau auf der Mitte zwischen diesen liegen. Dann haben B und C aber keinen Abstand von $2k$.

Auf Grund dieser Tatsache erscheint es interessant zu untersuchen, ob es nicht eine Transformation F gibt, die die grundlegenden Eigenschaften einer Metrik erhält, so dass zumindest $F(D^{multi}[q, k](S, S')) = \|\Phi(S), \Phi(S')\|_2$ mit $S, S' \in \mathcal{S}$ gilt. Im vorliegenden Fall kann dies etwa durch $F(x) = x^p$ mit $p = 1/2$ gewährleistet werden, wobei die Spiketrains als Ecken eines Quadrats mit Seitenlänge \sqrt{k} abgebildet werden. Der Quotient der Abstände wird somit von $2k/k = 2$ auf $\sqrt{2}$ abgeändert. Wird die Transformation F so gesetzt, dass dieser Quotient größer als $\sqrt{2}$ ist, ist laut Aronov et al. ([AV04]) eine kongruente Einbettung nicht möglich. Bei einer Wahl eines Exponenten p , so dass das Verhältnis kleiner als $\sqrt{2}$ ist, ist eine Einbettung zwar möglich, allerdings nicht als Viereck, sondern als Tetraeder. Der Quotient der Abstände ist also für die Einbettbarkeit entscheidend.

Aronov et al. formulieren einige gewünschte Forderungen an eine derartige Transformation F , wie etwa, dass zwei geometrische Figuren, die im originalen metrischen Raum ähnlich sind, dies auch nach einer Transformation sein sollen. Auf Grund dieser und weiterer Eigenschaften folgern sie dann, dass es sich bei der Transformation um eine Potenzfunktion handeln muss (siehe [AV04]).

Bemerkung 2.3.7

Eine Transformation F sollte ordnungserhaltend sein, also:

$$F(b) > F(a) \quad \text{falls } b > a$$

Ausserdem sollte sie Verhältnis-skaleninvariant sein, das heißt:

$$\frac{F(ca)}{F(cb)} = \frac{F(a)}{F(b)} \quad \text{für } c > 0$$

Die Dreieckungleichung sollte auch gelten: $F(a + b) \leq F(a) + F(b)$

Damit ergibt sich:

$$F(x) = \alpha x^p \quad \text{mit } \alpha > 0 \text{ und } 0 < p \leq 1$$

Im Folgenden werden nur Transformationen nach Bemerkung 2.3.7 betrachtet.

Definition 2.3.8

Ein semimetrischer Raum (\mathcal{S}, d_1) heißt unter einer Transformation kongruent zu einem anderen semimetrischen Raum (\mathcal{S}', d_2) , wenn es eine Transformation F nach Bemerkung 2.3.7 und eine Funktion Φ gibt, so dass für alle $x, y \in \mathcal{S}$ gilt:

$$F(d_1(x, y)) = d_2(\Phi(x), \Phi(y))$$

Dementsprechend sei unter einer Transformation kongruent einbettbar definiert.

Zunächst werden einige Spezialfälle nach [AV04] behandelt, die nicht von dem zentralen Satz in diesem Kapitel abgedeckt werden.

Bemerkung 2.3.9

Für das Abstandsmaß D^{count} lassen sich alle Spiketrains in \mathbb{R} kongruent einbetten, indem man sie auf die Anzahl ihrer Spikes abbildet.

Im Fall $q = \infty$ gibt $D^{spike}[q]$ ja gerade die Anzahl der zeitlich nicht zusammenfallenden Spikes an. Man kann also immer die Zeit so diskretisieren, dass $D^{spike}[\infty]$ gerade der Hammingdistanz D^H entspricht. Für diese gilt aber gerade mit $X, Y \in \mathbb{Z}_2^n \subset \mathbb{R}^n$: $\sqrt{D^H(X, Y)} = \|X - Y\|_2$. Die Spiketrains sind also zumindest unter einer Transformation kongruent einbettbar.

Für $0 < q < \infty$ behaupten Aranov et al. ([AV04]), dass $D^{spike}[q]$ lokal euklidisch ist. Das heißt, es soll zu jedem Spiketrain S_a Mengen von Spiketrains geben, die in l_2^n kongruent einbettbar sind. Sei etwa $S_a = \{a_1, a_2, \dots, a_n\}$ und $\mathcal{B}(S_a, \frac{1}{2}) = \{S : D^{spike}[q](S_a, S) < \frac{1}{2}\}$. Für alle $S \in \mathcal{B}(S_a, \frac{1}{2})$ muss gelten, dass sie genauso wie S_a aus N Spikes bestehen, da das Einfügen oder Löschen eines Spikes bereits einen Preis von 1 hat. Natürlich bedeutet das, dass die Zeitpunkte der Aktionspotentiale eines Spiketrains aus der Kugel dementsprechend nah bei derer von S_a liegen. Ein Spike S aus der Kugel lässt sich somit darstellen als $S = \{a_1 + \delta_1, a_2 + \delta_2, \dots, a_n + \delta_n\}$ mit $\sum_i q|\delta_i| < \frac{1}{2}$. Für $S, S' \in \mathcal{B}(S_a, \frac{1}{2})$ gilt $D^{spike}[q](S, S') = \sum_i q|\delta_i - \delta'_i|$. Nach Aranov et al. soll es nun möglich sein, diese Menge in einen euklidischen Raum einzubetten. Dazu soll man einen Spiketrain $\{a_1 + \delta_1, a_2 + \delta_2, \dots, a_n + \delta_n\}$ abbilden auf $[\delta_1, \delta_2, \dots, \delta_n]$ und dann für die Transformation F nach Lemma 2.3.7 $F(x) = x^{\frac{1}{2}}$ wählen. Doch hiermit erhält man nicht die gewünschte Aussage.

Die Nicht-Einbettbarkeit von Spiketrains, die aus Antworten von weniger als drei Nervenzellen bestehen, führen Victor et al. einen Computerbeweis, auf den hier nicht eingegangen werden soll. Für mindestens drei Neurone lautet nun die wichtige Aussage in diesem Kapitel:

Satz 2.3.10

Sei $(\mathcal{S}, D^{multi}[q, k])$ der semimetrische Raum ¹ der gekennzeichneten Spiketrains nach Definition 2.3.1 und $M \geq 3$. Dann existiert keine Transformation F nach Bemerkung 2.3.7 und Dimension n , so dass jede Teilmenge von \mathcal{S} unter einer Transformation kongruent einbettbar ist in den n -dimensionalen euklidischen Raum l_2^n .

Zum Beweis der vorhergehenden Aussage bedarf es eines Satzes über die kongruente Einbettbarkeit. Dieser basiert auf der sogenannten Cayley-Menger Determinante, die nun nach [B70] (siehe auch [HKC83] und [M83]) eingeführt wird.

¹Streng gesehen handelt es sich hier wieder um Äquivalenzklassen (vgl. Kapitel 2.2.2)

Definition 2.3.11 (Cayley-Menger Determinante)

Sei (\mathcal{S}, d) mit $\mathcal{S} = \{S_0, S_1, \dots, S_l\}$ ein semimetrischer Raum. Dann heißt

$$D(S_0, S_1, \dots, S_l) := \det \begin{pmatrix} 0 & 1 & 1 & \dots & 1 \\ 1 & 0 & d_{01}^2 & \dots & d_{0l}^2 \\ 1 & d_{10}^2 & 0 & \dots & d_{1l}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & d_{l0}^2 & d_{l1}^2 & \dots & 0 \end{pmatrix} \quad \text{mit } d_{ij} := d(S_i, S_j)$$

die Cayley-Menger Determinante des semimetrischen Raumes (\mathcal{S}, d) . Dabei sei angemerkt, dass ihr Wert unabhängig von der Ordnung beziehungsweise Nummerierung der Elemente S_0, S_1, \dots, S_l ist.

Mit Hilfe dieser Definition kann eine notwendige Bedingung für kongruente Einbettbarkeit eines semimetrischen $(r+1)$ -Tupels, und somit einer $(r+1)$ -elementigen Menge, formuliert werden. Für den Nachweis von Satz 2.3.10 wird keine hinreichende Bedingung benötigt. Zu einem besseren Verständnis des einführenden Beispiels werden aber beide Bedingungen nach [B70] und [M83] aufgeführt.

Satz 2.3.12

Ein semimetrisches $(r+1)$ -Tupel S_0, S_1, \dots, S_r ist genau dann in einen euklidischen Raum \mathbb{R}^n kongruent einbettbar, wenn für alle l Punkte mit $l \leq n+1$ die Cayley-Menger Determinante verschwindet oder das Vorzeichen $(-1)^{l+1}$ hat und der Wert aller Cayley-Menger Determinanten von mehr als $n+1$ Punkten Null ist.

Im Kontext des Inhalts bzw. Volumens eines Simplex, welches zunächst definiert wird (siehe [GB03],[HR05]), lässt sich der Beweis recht anschaulich durchführen.

Definition 2.3.13 (Simplex)

Ein d -Simplex mit Eckpunkten S_0, S_1, \dots, S_d ist die konvexe Hülle dieser $d+1$ Punkte. Ein Simplex in \mathbb{R}^2 heißt Dreieck und ein Simplex in \mathbb{R}^3 heißt Tetraeder. Ein Simplex heißt nicht-entartet, falls S_0, S_1, \dots, S_d affin unabhängig sind.

Der Beweis der notwendigen Bedingung lautet nach [B70] ([M83]):

Beweis von Satz 2.3.12

Seien $k+1$ Punkte S_0, S_1, \dots, S_k in \mathbb{R}^k gegeben. Habe S_i die Koordinaten $(x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(k)})$ für $0 \leq i \leq k$. Bezeichne $V(S_0, S_1, \dots, S_k)$ das Volumen des (möglicherweise entarteten) Simplex mit Eckpunkten S_0, S_1, \dots, S_k . Dann erhält man nach einer

Formel aus der analytischen Geometrie:

$$V(S_0, S_1, \dots, S_k) = \frac{1}{k!} \det \begin{pmatrix} x_0^{(1)} & x_0^{(2)} & \cdots & x_0^{(k)} & 1 \\ x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(k)} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_k^{(1)} & x_k^{(2)} & \cdots & x_k^{(k)} & 1 \end{pmatrix} \quad (2.3)$$

Der Wert der Determinante bleibt nach dem Hinzufügen einer $k + 2$ -ten Spalte und Zeile mit lauter Nullen außer einer Eins auf der Diagonalen, wie folgt, unverändert:

$$= \frac{1}{k!} \det \begin{pmatrix} x_0^{(1)} & x_0^{(2)} & \cdots & x_0^{(k)} & 1 & 0 \\ x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(k)} & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & 0 \\ x_k^{(1)} & x_k^{(2)} & \cdots & x_k^{(k)} & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix} \quad (2.4)$$

Nach einem Vertauschen der letzten beiden Zeilen und Transponieren der Matrix gilt:

$$= -\frac{1}{k!} \det \begin{pmatrix} x_0^{(1)} & x_1^{(1)} & \cdots & x_k^{(1)} & 0 \\ x_0^{(2)} & x_1^{(2)} & \cdots & x_k^{(2)} & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ x_0^{(k)} & x_1^{(k)} & \cdots & x_k^{(k)} & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \quad (2.5)$$

Nach Multiplikation von 2.4 und 2.5 erhält man:

$$V^2(S_0, S_1, \dots, S_k) = -\frac{1}{(k!)^2} \det \begin{pmatrix} (S_0, S_0) & (S_0, S_1) & \cdots & (S_0, S_k) & 1 \\ (S_1, S_0) & (S_1, S_1) & \cdots & (S_1, S_k) & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ (S_k, S_0) & (S_k, S_1) & \cdots & (S_k, S_k) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \quad (2.6)$$

Dabei bezeichne (S_i, S_j) das Skalarprodukt von S_i und S_j . Dann gilt $(S_i, S_j) = \frac{1}{2}(S_i, S_i) + \frac{1}{2}(S_j, S_j) - \frac{1}{2}\|S_i - S_j\|_2^2$. Ersetzt man so die jeweiligen Skalarprodukte und zieht sowohl von der i -ten Spalte das Produkt aus der letzten Spalte und $\frac{1}{2}(S_{i-1}, S_{i-1})$ als auch von der i -ten Zeile das entsprechende Produkt ab, so ergibt dies:

$$= -\frac{1}{(k!)^2} \det \begin{pmatrix} -\frac{1}{2}\|S_0 - S_0\|_2^2 & -\frac{1}{2}\|S_0 - S_1\|_2^2 & \cdots & -\frac{1}{2}\|S_1 - S_k\|_2^2 & 1 \\ -\frac{1}{2}\|S_1 - S_0\|_2^2 & -\frac{1}{2}\|S_1 - S_1\|_2^2 & \cdots & -\frac{1}{2}\|S_1 - S_k\|_2^2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ -\frac{1}{2}\|S_k - S_0\|_2^2 & -\frac{1}{2}\|S_k - S_1\|_2^2 & \cdots & -\frac{1}{2}\|S_k - S_k\|_2^2 & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \quad (2.7)$$

Nach Multiplikation aller Spalten mit -2 und dann Multiplikation der letzten Zeile mit $-\frac{1}{2}$ erhält man:

$$= \frac{(-1)^{k+1}}{2^k(k!)^2} \det \begin{pmatrix} 0 & \|S_0 - S_1\|_2^2 & \cdots & \|S_1 - S_k\|_2^2 & 1 \\ \|S_1 - S_0\|_2^2 & 0 & \cdots & \|S_1 - S_k\|_2^2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \|S_k - S_0\|_2^2 & \|S_k - S_1\|_2^2 & \cdots & 0 & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \quad (2.8)$$

Die Determinante der Matrix in 2.8 entspricht der Cayley-Menger Determinante. Diese steht also wie folgt im Verhältnis zum Volumen des Simplex:

$$V^2(S_0, S_1, \dots, S_k) = \frac{(-1)^{k+1}}{2^k(k!)^2} D(S_0, S_1, \dots, S_k) \quad (2.9)$$

Die Cayley-Menger Determinante ist offensichtlich kongruenzinvariant und auf Grund der Tatsache, dass $V^2(S_0, S_1, \dots, S_k) \geq 0$ für jedes euklidische $(k+1)$ -Tupel gilt, muss das Vorzeichen der Determinante wie behauptet mit der Anzahl an Punkten variieren. \square

Beispiel 2.3.14

In der Situation von Beispiel 2.3.6 erhält man mit $S_0 = A, S_1 = B, S_2 = C, S_3 = D$ als Cayley-Menger Determinante (nach *Maple*):

$$D(S_0, S_1, S_2, S_3) = \det \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & k^2 & 4k^2 & k^2 \\ 1 & k^2 & 0 & k^2 & 4k^2 \\ 1 & 4k^2 & k^2 & 0 & k^2 \\ 1 & k^2 & 4k^2 & k^2 & 0 \end{pmatrix} = -128k^6$$

Wollte man die vier Spiketrains in \mathbb{R}^3 einbetten, müsste unter anderem $\text{sgn}D(S_0, S_1, S_2, S_3) = (-1)^{3+1} = 1$ gelten. Dies ist hier aber nicht der Fall. Für eine kongruente Einbettung in \mathbb{R}^2 hätte u.a. dagegen die Cayley-Menger Determinante aller Spiketrains verschwinden müssen. Also ist diese Menge von Spiketrains tatsächlich nicht kongruent einbettbar in einen euklidischen Raum.

Wendet man dagegen die Transformation $F(x) = \sqrt{x}$ auf die Abstände an, erhält man:

$$\begin{aligned} \text{sgn}D(S_0, S_1) &= \text{sgn}D(S_0, S_2) = \text{sgn}D(S_0, S_3) = \text{sgn}D(S_1, S_2) \\ &= \text{sgn}D(S_1, S_3) = \text{sgn}D(S_2, S_3) = (-1)^{1+1} \\ \text{sgn}D(S_0, S_1, S_2) &= \text{sgn}D(S_0, S_2, S_3) = \text{sgn}D(S_1, S_2, S_3) = \text{sgn}D(S_0, S_1, S_3) = (-1)^{2+1} \\ \text{sgn}D(S_0, S_1, S_2, S_3) &= 0 \end{aligned}$$

Nach Satz 2.3.13 ist also $(\{S_0, S_1, S_2, S_3\}, D^{\text{multi}}[q, k])$ unter einer Transformation kongruent einbettbar in $(\mathbb{R}^2, \|\cdot\|_2)$.

Die gleichen Vorzeichen ergeben sich für $F(x) = x^{1/4}$ für die entsprechenden Cayley-Menger Determinanten aller l Spiketrains mit $l \leq 3$. Allerdings gilt $\text{sgn}D(S_0, S_1, S_2, S_3) = 1$. Die Spiketrains bilden also unter dieser Transformation die Eckpunkte eines Simplex in \mathbb{R}^3 , sind aber nicht in \mathbb{R}^2 kongruent einbettbar.

Bemerkung 2.3.15

Es ist stets möglich eine endliche Menge von N Spiketrains unter einer Transformation in einen euklidischen Raum einzubetten. So gibt es immer einen kleinsten und einen größten Abstand. Der Exponent p kann daher immer so klein gewählt werden, dass die abgeänderten Abstände ein Vielfaches von höchstens $1 + \epsilon$ voneinander darstellen. Daher können die transformierten Abstände bei entsprechend klein ϵ ein $(N - 1)$ -dimensionales approximiertes (nicht-entartetes) Simplex bilden. Somit wächst mit wachsender Anzahl an Spikes die Dimension, die zur Einbettung notwendig ist, unbeschränkt an. ([AV04])

Nachdem alle notwendigen Definitionen und Aussagen zusammengetragen wurden, kann der Beweis des Satzes über die Einbettbarkeit nach [AV04] ausgeführt werden.

Beweis von Satz 2.3.10

Für den Beweis wird eine Folge von Mengen gekennzeichneter Spiketrains konstruiert. Dabei wird für jedes fest betrachtete Folgenglied eine Transformation existieren, so dass die notwendige Bedingung zur Einbettbarkeit nach Satz 2.3.13 erfüllt ist. Jedoch wird die Betrachtung des Grenzwertes das Festlegen einer Transformation, die für alle Folgenglieder zur Einbettbarkeit in Frage kommt, als unmöglich zeigen.

Der erste Teil des Beweises besteht aus der Konstruktion dieser besonderen Folge von Mengen gekennzeichneter Spiketrains, deren Idee sich an Beispiel 2.3.6 orientiert. Daran anschließend wird dann das Kriterium basierend auf der Cayley-Menger Determinante angewendet.

Konstruktion von zwei Gruppen Spiketrains

Im Folgenden wird eine Folge von Spiketrains konstruiert, für die keine feste Wahl von $p > 0$ (vgl. Bemerkung 2.3.7) möglich ist, so dass die Bedingungen erfüllt werden, damit alle Folgenglieder unter der selben Transformation einbettbar sind. Dabei werden zwei Gruppen von Spiketrains analog zu Beispiel 2.3.6 entworfen, wobei die Anzahl Spiketrains N in jeder Gruppe anwächst. Der Abstand innerhalb einer Gruppe wird dann konstant ρ_N und der Abstand zweier Spiketrains unterschiedlicher Gruppen eine weitere Konstante ζ_N sein.

Zur Konstruktion dieser Folge gehe man nun von $2M$ Spiketrains gleicher Länge aus, die von 0 bis $2M - 1$ durchnummeriert sind und im Folgenden als „Bausteine“ verwendet werden. Diese sind in zwei Gruppen eingeteilt: Der einen gehören die Spiketrains mit gerader Zahl $0, 2, 4, \dots, 2M - 2$ an, die zweite wird gebildet durch die ungerade nummerierten

sind mit ungeraden Ziffern bzw. geraden Ziffern nummeriert. Mit dem aufgezeigten Bildungsprinzip bedarf es für die Erzeugung von $2N$ Spiketrains (N pro Gruppe) M^{N-1} Bausteine für jeden einzelnen Spiketrain. Nach Gleichung 2.10 ergibt sich so ein Abstand von $\zeta_N = M^{N-1}w$ zwischen Spiketrains mit unterschiedlicher Gruppenzugehörigkeit. Dagegen ist der Abstand von Spiketrains einer Gruppe $\rho_N = (M-1)M^{N-2}v$, da immer $1/M$ der M^{N-1} Bausteine übereinstimmen.

Der Quotient der Abstände ist konstant:

$$\frac{\rho_N}{\zeta_N} = \frac{(M-1)v}{Mw} \quad (2.11)$$

Wie schon in Beispiel 2.3.6 gesehen, bildet dieser Quotient die entscheidende Rolle für die Einbettbarkeit der Spiketrains. Es gilt:

$$\frac{v}{w} > \frac{M}{M-1} \Rightarrow \frac{\rho_N}{\zeta_N} > 1 \quad (2.12)$$

Die Bausteine sind also so zu konstruieren, dass das Verhältnis der Abstände v und w die Voraussetzung nach 2.12 erfüllt. Abbildung 2.5 B zeigt, wie die Bausteine im Fall von $M = 3$ zu konstruieren sind.

Lemma 2.3.16

Werden die Bausteine für $M = 3$ wie in Abbildung 2.5 B zu sehen konstruiert, so gilt $\frac{v}{w} > \frac{M}{M-1}$ und somit $\frac{\rho_N}{\zeta_N} > 1$.

Für $M = 3$ ist also die Voraussetzung nach 2.12 erfüllt und es gilt $\frac{\rho_N}{\zeta_N} > 1$. Wie der folgende Teil des Beweises zeigen wird, verhindert dies die Einbettbarkeit. Es gilt dann auch die Nicht-Einbettbarkeit für $M > 3$. Denn wenn die Einbettbarkeit für ein M mit $M > 3$ gegeben wäre, müsste dies auch für $M = 3$ gelten, da letzterer Fall in ersterem enthalten ist.

Cayley-Menger Determinanten Kriterium

Der Quotient der Abstände innerhalb einer und zwischen zwei Gruppen ρ_N/ζ_N ist also unabhängig von N stets gleich einer Konstanten $\rho/\zeta > 1$. Somit gilt $\rho_N = \alpha_N\rho$ und $\zeta_N = \alpha_N\zeta$ mit einer Folge α_N positiver Koeffizienten.

Seien nun $S_0, S_1, \dots, S_{2N-1}$ nach vorhergehenden Erläuterungen gebildete gekennzeichnete Spiketrains, wobei die Spiketrains mit ungerader Nummerierung die eine und die mit

gerader Nummerierung die andere Gruppe darstellen. Dann gilt:

$$D(S_0, S_1, \dots, S_{2N-1}) = \det \begin{pmatrix} 0 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & \zeta_N^2 & \rho_N^2 & \dots & \rho_N^2 & \zeta_N^2 \\ 1 & \zeta_N^2 & 0 & \zeta_N^2 & \dots & \zeta_N^2 & \rho_N^2 \\ 1 & \rho_N^2 & \zeta_N^2 & 0 & \dots & \rho_N^2 & \zeta_N^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \rho_N^2 & \zeta_N^2 & \rho_N^2 & \dots & 0 & \zeta_N^2 \\ 1 & \zeta_N^2 & \rho_N^2 & \zeta_N^2 & \dots & \zeta_N^2 & 0 \end{pmatrix} \quad (2.13)$$

Multiplikation der ersten Spalte mit α_N^2 und Ersetzen von ρ_N mit $\alpha_N \rho$ und ζ_N mit $\alpha_N \zeta$ ergibt:

$$= \alpha_N^{-2} \cdot \det \begin{pmatrix} 0 & 1 & 1 & 1 & \dots & 1 & 1 \\ \alpha_N^2 & 0 & \alpha_N^2 \zeta^2 & \alpha_N^2 \rho^2 & \dots & \alpha_N^2 \rho^2 & \alpha_N^2 \zeta^2 \\ \alpha_N^2 & \alpha_N^2 \zeta^2 & 0 & \alpha_N^2 \zeta^2 & \dots & \alpha_N^2 \zeta^2 & \alpha_N^2 \rho^2 \\ \alpha_N^2 & \alpha_N^2 \rho^2 & \alpha_N^2 \zeta^2 & 0 & \dots & \alpha_N^2 \rho^2 & \alpha_N^2 \zeta^2 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_N^2 & \alpha_N^2 \rho^2 & \alpha_N^2 \zeta^2 & \alpha_N^2 \rho^2 & \dots & 0 & \alpha_N^2 \zeta^2 \\ \alpha_N^2 & \alpha_N^2 \zeta^2 & \alpha_N^2 \rho^2 & \alpha_N^2 \zeta^2 & \dots & \alpha_N^2 \zeta^2 & 0 \end{pmatrix} \quad (2.14)$$

Mit $r := \rho^2$ und $s := \zeta^2$ erhält man:

$$= \alpha_N^{4N-2} \cdot \det \begin{pmatrix} 0 & 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & s & r & \dots & r & s \\ 1 & s & 0 & s & \dots & s & r \\ 1 & r & s & 0 & \dots & r & s \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & r & s & r & \dots & 0 & s \\ 1 & s & r & s & \dots & s & 0 \end{pmatrix} =: \alpha_N^{4N-2} \cdot \det A_N \quad (2.15)$$

Lemma 2.3.17

Es gilt: $\det A_N = 2Nr^{2N-2}[Ns - (N-1)r]$

Nach Satz 2.3.12 muss nun das Vorzeichen von $\det A_N$ gleich $(-1)^{2N}$ sein (α_N ist positiv), also die Determinante muss größer oder gleich Null sein. Der erste Faktor ist immer größer Null, also muss dies auch für den zweiten gelten, woraus folgt:

$$\frac{r}{s} = \left(\frac{\rho}{\zeta}\right)^2 \leq \frac{N}{N-1} \quad (2.16)$$

Die Anwendung einer Transformation $F(x) = \alpha x^p$ liefert:

$$\left(\frac{\rho}{\zeta}\right)^{2p} \leq \frac{N}{N-1} \quad (2.17)$$

Für eine immer größer werdende Anzahl Spiketrains N in einer Gruppe konvergiert die rechte Seite der Ungleichung 2.17 gegen 1. Da der Quotient ρ/ζ aber größer als 1 ist, kann die Ungleichung ungeachtet des Exponenten p nicht erfüllt werden. \square

In Beispiel 2.3.6 gilt gerade $\frac{\rho_N}{\zeta_N} = \frac{2k}{k} = 2 > 1$. Diese vier Spiketrains eignen sich jedoch nicht als Bausteine für eine wie zuvor beschriebene Verkettung. So ist die Bedingung von 2.12 nicht erfüllt, denn es ist $\frac{v}{w} = \frac{2k}{k} = \frac{M}{M-1}$.

Beweis von Lemma 2.3.16

Zunächst ist nachzuweisen, dass $\frac{v}{w} > \frac{2}{3}$ gilt. Für zwei Spiketrains S und S' derselben Gruppe ($\{0, 2, 4\}$ oder $\{1, 3, 5\}$) aus Abbildung ist $v = D^{multi}[q, k](S, S') = \min\{4, q\Delta t + 2, 2q\Delta t + k\}$. Denn der Abstand ist gerade der minimale Preis von drei Möglichkeiten einen Spiketrain in den anderen umzuformen. Dies kann realisiert werden durch das Löschen von zwei Spikes eines Spiketrains und dann Einfügen von zwei richtigen Spikes. Eine weitere Möglichkeit stellt das Verschieben eines Spikes und Löschen des zweiten Spikes und dann Einfügen eines richtig gekennzeichneten Spikes dar. Ebenfalls kann ein Angleichen durch das Verschieben beider Spikes und dann Ändern der Kennzeichnung des einen Spikes erfolgen. Somit erhält man mit $\Delta t = k/2q$:

$$v = \begin{cases} 2k, & \text{für } 0 < k \leq \frac{4}{3} \\ \frac{k}{2} + 2 & \text{für } \frac{4}{3} < k < 2 \end{cases} \quad (2.18)$$

Für zwei Spiketrains S und S' verschiedener Gruppen gilt für den Abstand mit $\Delta t = k/2q$ unabhängig von k :

$$w = D^{multi}[q, k](S, S') = k \quad (2.19)$$

Der minimale Weg ist hier jedoch abhängig von der Kombination der Bausteine. Für die Spiketrains 0 und 1, 2 und 3, und 4 und 5 ist es am günstigsten zwei Spikes eines Bausteins um je $\Delta t = k/2q$ zu verschieben. Alle anderen Paare (S, S') werden durch Änderung einer Kennzeichnung eines Spikes angeglichen. Somit folgt nun aus den Gleichungen 2.18 und 2.19:

$$\frac{v}{w} = \begin{cases} 2 & \text{für } 0 < k \leq \frac{4}{3} \\ \frac{1}{2} + \frac{2}{k} & \text{für } \frac{4}{3} < k < 2 \end{cases} \quad (2.20)$$

Der Quotient v/w ist also immer größer als $3/2$. Mit 2.11 folgt nun aus Gleichung 2.20, dass $\rho_N/\zeta_N > 1$ gilt, denn:

$$\frac{\rho_N}{\zeta_N} = \begin{cases} \frac{4}{3} & \text{für } 0 < k \leq \frac{4}{3} \\ \frac{1}{3} + \frac{4}{3k} & \text{für } \frac{4}{3} < k < 2 \end{cases} \quad (2.21)$$

□

Die Berechnung der Determinante von A_N ist noch durchzuführen:

Beweis von Lemma 2.3.16

Da es sich bei A_N um eine symmetrische Matrix handelt und somit diagonalisierbar ist, kann die Determinante leicht über die Eigenwerte berechnet werden. Sei nun λ ein Eigenwert und $x = (x_0, x_1, \dots, x_{2N})$ der zugehörige Eigenvektor, dann gilt:

$$A_N x = \begin{pmatrix} x_1 + x_2 + \dots + x_{2N} \\ x_0 + r(x_1 + x_3 + \dots + x_{2N-1}) + s(x_2 + \dots + x_{2N}) - rx_1 \\ x_0 + r(x_2 + x_4 + \dots + x_{2N}) + s(x_1 + \dots + x_{2N-1}) - rx_2 \\ \vdots \\ x_0 + r(x_1 + x_3 + \dots + x_{2N-1}) + s(x_2 + \dots + x_{2N}) - rx_{2N-1} \\ x_0 + r(x_2 + x_4 + \dots + x_{2N}) + s(x_1 + \dots + x_{2N-1}) - rx_{2N} \end{pmatrix} = \begin{pmatrix} \lambda x_0 \\ \lambda x_1 \\ \lambda x_2 \\ \vdots \\ \lambda x_{2N-1} \\ \lambda x_{2N} \end{pmatrix} \quad (2.22)$$

Es ergibt sich aus Gleichung (2.22):

$$\begin{aligned} \lambda x_0 &= \sum_i x_i \\ (r + \lambda)x_j &= x_0 + r \sum_{\text{ungerade } i} x_i + s \sum_{\text{gerade } i > 0} x_i \quad \text{für ungerade } j > 0, \\ (r + \lambda)x_j &= x_0 + r \sum_{\text{gerade } i > 0} x_i + s \sum_{\text{ungerade } i} x_i \quad \text{für gerade } j > 0. \end{aligned} \quad (2.23)$$

Aus den letzten beiden Zeilen von 2.23 folgt:

$$\begin{aligned} (r + \lambda)x_1 &= (r + \lambda)x_3 = \dots = (r + \lambda)x_{2N-1} \\ (r + \lambda)x_2 &= (r + \lambda)x_4 = \dots = (r + \lambda)x_{2N} \end{aligned} \quad (2.24)$$

Für $N > 1$ ist eine Lösung der Gleichung $\lambda = -r$. Für die Vielfachheit dieses Eigenwertes, also dieser Nullstelle des charakteristischen Polynoms χ_{A_N} , gilt auf Grund der Diagonalisierbarkeit:

$$\begin{aligned} \mu(\chi_{A_N}, -r) &= \dim \text{Eig}(A_N, -r) = \dim(\text{Lös}(A_N + rI_{N \times N}, 0)) \\ &= 2N + 1 - \text{rang}(A_N + rI_{N \times N}) = 2N + 1 - 3 = 2N - 2 \end{aligned} \quad (2.25)$$

Wenn $\lambda \neq -r$ ist, dann gilt $x_i = x_{\text{ungerade}}$ für alle ungerade i und $x_i = x_{\text{gerade}}$ für alle gerade $i > 0$. Somit können die beiden letzten Gleichungen in 2.23 auch geschrieben werden

als:

$$\begin{aligned} x_j &= \frac{1}{r+\lambda}x_0 + \frac{Nr}{r+\lambda}x_{\text{ungerade}} + \frac{Ns}{r+\lambda}x_{\text{gerade}} && \text{für ungerade } j \\ x_j &= \frac{1}{r+\lambda}x_0 + \frac{Nr}{r+\lambda}x_{\text{gerade}} + \frac{Ns}{r+\lambda}x_{\text{ungerade}} && \text{für gerade } j \end{aligned} \quad (2.26)$$

Eingesetzt in die erste Gleichung von 2.23 erhält man:

$$\lambda x_0 = \frac{2N}{r+\lambda}x_0 + \frac{Nr}{r+\lambda}Nx_{\text{ungerade}} + \frac{Nr}{r+\lambda}Nx_{\text{gerade}} + \frac{Ns}{r+\lambda}Nx_{\text{ungerade}} + \frac{Ns}{r+\lambda}Nx_{\text{gerade}} \quad (2.27)$$

Zusammen mit der Tatsache, dass $\sum x_i = Nx_{\text{ungerade}} + Nx_{\text{gerade}}$ gilt, ergibt dies unter erneuter Benutzung der ersten Gleichung von 2.23:

$$\lambda x_0 = \frac{2N}{r+\lambda}x_0 + \frac{\lambda N(r+s)}{r+\lambda}x_0 \quad (2.28)$$

Eine Lösung von 2.28 ist $x_0 = 0$. Da $x_1 = x_3 = \dots = x_{2N-1}$ und $x_2 = x_4 = \dots = x_{2N}$ gilt, folgt aus der ersten Zeile von 2.23, dass der entsprechende Eigenvektor $x = (0, 1, -1, 1, \dots, -1, 1, -1)$ lautet. Somit:

$$A_N x = \left(0 \quad rN - sN - r \quad -rN + sN + r \quad \dots \quad rN - sN - r \quad -rN + sN + r \right)^T \quad (2.29)$$

Der zugehörige Eigenwert lautet also $(N-1)r - Ns$.

Wenn $x_0 \neq 0$, so wird aus 2.28:

$$\begin{aligned} \lambda(r+\lambda) &= 2N + \lambda N(r+s) \\ \Leftrightarrow 0 &= \lambda^2 - [(N-1)r + Ns]\lambda - 2N \end{aligned} \quad (2.30)$$

Hieraus folgt, dass das Produkt der beiden Nullstellen gerade $-2N$ ist. Also hat das charakteristische Polynom χ_{A_N} vier verschiedene Nullstellen: $\lambda_1 = -r$, $\lambda_2 = (N-1)r - Ns$ und die Nullstellen λ_3 und λ_4 der obigen Gleichung. Dabei haben die letzten drei Eigenwerte je die Vielfachheit 1, da die Matrix ja diagonalisierbar ist und λ_1 eine Vielfachheit von $2N-2$ hat. Damit ergibt sich:

$$\begin{aligned} \det A_N &= (-r)^{2N-2}[(N-1)r - Ns](-2N) \\ &= 2Nr^{2N-2}[Ns - (N-1)r] \end{aligned} \quad (2.31)$$

□

3 Stimulusabhängiges Clustern

Nachdem die verschiedenen Abstandsmaße charakterisiert wurden, bedarf es einer Vorgehensweise, um dasjenige Maß zu bestimmen, welches den Unterschied zweier Spiketrains quantifiziert über die für die Informationsverschlüsselung wichtigsten Aspekte. Hierzu wird zunächst ein Algorithmus eingeführt, der dazu dient, neuronale Antworten auf verschiedene Stimuli auf Basis eines der zuvor beschriebenen Abstandsmaßes in Klassen einzuteilen. Danach werden Maße definiert, mit denen die Güte dieser Gruppierung bestimmt werden kann. So ist bekannt, welcher Spiketrain auf welchem Stimulus basiert und wie daher die Einteilung hätte sein müssen.

3.1 Algorithmus

Seien KL Spiketrains gewonnen, indem einem Neuron K verschiedene Stimuli je L mal präsentiert wurden. Dabei erwartet man hinsichtlich einer korrekten Informationsverschlüsselung, dass auf den selben Stimulus immer gleich reagiert wird. Bezeichne im Folgenden Stimulusklasse die Menge von Spiketrains, die durch den gleichen Stimulus hervorgerufen wurden.¹ Den Erwartungen nach sollte also ein durch Stimulus i hervorgerufener Spiketrain zu einem Spiketrain, der der selben Stimulusklasse s_i angehört, einen geringeren Abstand haben als zu einem Spiketrain einer anderen Stimulusklasse s_j .

Partitioniert man nun die Menge aller Spiketrains unter Verwendung der verschiedenen Abstandsmaße in K Antwortklassen r_1, r_2, \dots, r_K , so sollte dies mit demjenigen Abstandsmaß, bei dem die für die Kodierung relevanten Aspekte betrachtet werden, zu erwartungsgemäßen Ergebnissen führen. So sollte die Ähnlichkeit zweier Spiketrains einer Stimulusklasse dazu führen, dass sie auch in dieselbe Antwortklasse geordnet werden.

Victor et al. ([VP96],[VP97]) haben zur Bildung der Antwortklassen einen Algorithmus konstruiert, der die Information nutzt, welcher Spiketrain durch welchen Stimulus ausgelöst wurde. So wird zunächst für einen Spiketrain S_{ij} , der durch den i -ten Stimulus bei der j -ten Präsentation dieses Stimulus hervorgerufen wurde, und jede Stimulusklasse s_k für $1 \leq k \leq K$ die Distanz $d(S_{ij}, s_k)$ berechnet. Diese Distanz kann durch einen Parameter z modifiziert werden, stellt in ihrer einfachsten Form aber gerade den mittleren Abstand von S_{ij} zu allen Spiketrains der Klasse s_k dar. Wird das Minimum von $d(S_{ij}, s_k)$ über alle Stimulusklassen bei der Klasse s_l angenommen, so wird der Spiketrain S_{ij} in die Antwortklasse r_l eingeteilt.

¹Der Begriff Stimulusklasse hat bei Victor et al. ([VP96],[VP97]) eine etwas andere Bedeutung.

Neben dieser Bildung der Antwortklassen, wird in einer Matrix $N = (n_{ij})_{1 \leq i, j \leq K}$ die Anzahl Spiketrains festgehalten, die aus Stimulusklasse s_i in Antwortklasse r_j geordnet wurden. Dies bietet die Möglichkeit die Güte dieses stimulusabhängigen Clusters (siehe Kapitel 3.2) zu bestimmen.

Victors Erläuterungen des Algorithmus habe ich wie folgt als Pseudocode formuliert:

Algorithmus 3.1.1

Bezeichne S_{ij} den Spiketrain, der durch den i -ten Stimulus bei der j -ten Präsentation dieses Stimulus hervorgerufen wurde und D ein Abstandsmaß. Seien $s_i = \{S_{i1}, \dots, S_{iL}\}$ für $1 \leq i \leq K$ die Stimulusklassen und r_k für $1 \leq k \leq K$ die Antwortklassen. Dann:

$r_m := \{\}$ für $1 \leq m \leq K$

$n_{ij} := 0$ für $1 \leq i, j \leq K$

$N := (n_{ij})_{1 \leq i, j \leq K}$

für $1 \leq i \leq K$

 für $1 \leq j \leq L$

 für $1 \leq k \leq K$

 wenn $k=i$

$L := L - 1$

 ende

$$d(S_{ij}, s_k) := \left(\frac{1}{L} \sum_{S \in s_k \setminus \{S_{ij}\}} D(S_{ij}, S)^z \right)^{1/z}$$

 ende

$M := \arg \min_{1 \leq l \leq K} d(S_{ij}, s_l)$

 wenn $m \in M$

$r_m = r_m \cup \{S_{ij}\}$

$n_{im} = n_{im} + \frac{1}{|M|}$

 ende

ende

ende

Wie bereits erwähnt, wird ein Spiketrain S_{ij} also für $z = 1$ der Antwortgruppe r_l zugeordnet, wenn der mittlere Abstand zur Stimulusklasse s_l am geringsten ist. Durch den Clusterexponenten z kann dies ein wenig modifiziert werden. Für z gegen minus unendlich wird $d(S_{ij}, s_k)$ zur Minimumsnorm. Der kürzeste Abstand zwischen S_{ij} und einem Spiketrain aus der Klasse s_k fällt also desto stärker ins Gewicht je kleiner z wird. Dagegen wird für z gegen unendlich der Abstand zur Maximumsnorm, weshalb nicht der kürzeste

sondern der größte Abstand die entscheidende Rolle spielt. Ein Spiketrain wird dementsprechend in die Gruppe r_l eingeteilt, wenn der Abstand zu dem „Weitentferntesten“ der Klasse s_l kleiner ist als alle Abstände zu dem jeweiligen der anderen Gruppen.

Natürlich lässt sich das stimulusabhängige Clustern nach Victor et al. auch Anwenden auf Stimulusklassen mit unterschiedlicher Anzahl an Spiketrains. Der Algorithmus ist dann dementsprechend zu modifizieren.

Des Weiteren können anstelle von neuronalen Antworten einer Zelle auch die gleichzeitig abgeleiteten Antworten von mehreren Zellen in Form von gekennzeichneten Spiketrains (siehe vorhergehendes Kapitel) betrachtet werden.

3.2 Güte des stimulusabhängigen Clusters

Für die Bewertung des stimulusabhängigen Clusters gibt es verschiedene Möglichkeiten. Dabei sei angemerkt, dass die Stimulusabhängigkeit der Einteilung im Vordergrund steht und dies nicht der im Allgemeinen betrachteten Frage nach der Güte des Clusters, wie es zum Beispiel in [H00] behandelt wird, entsprechen muss.

Eine naheliegende Möglichkeit die Güte des stimulusabhängigen Clusters anzugeben, stellt die Bestimmung des mittleren prozentualen Anteils an korrekten Zuordnungen dar (vgl. [KWE01]).

Definition und Bemerkung 3.2.1

Mit den Bezeichnungen aus dem vorhergehenden Kapitel sei Prozent-Korrekt definiert als

$$P := \frac{1}{K} \cdot \sum_{i=1}^K \frac{n_{ii}}{L} \cdot 100\%.$$

Dann würde optimales stimulusabhängiges Clustern bedeuten:

$$n_{ij} = \begin{cases} L & \text{für } i = j \\ 0 & \text{für } i \neq j \end{cases}$$

und $P = 100\%$.

Dagegen verwenden Victor et al. ([VP96]) ein informationstheoretisches Maß, um zu bewerten, inwieweit systematisch stimulusabhängig zugeordnet wurde. Diese wechselseitige Information bezieht sich auf die gesamte Matrix N und wird im Folgenden, sofern nicht anders angegeben, basierend auf [CT91] eingeführt.

3.2.1 Informationstheoretische Grundlagen

Zunächst wird die Entropie definiert, die ein Maß für die Unsicherheit einer Zufallsvariable X darstellt .

Definition 3.2.2 (Entropie)

Sei X eine diskrete Zufallsvariable mit Wertemenge \mathcal{X} und Wahrscheinlichkeitsdichte $p(x) = P^X(\{x\})$, $x \in \mathcal{X}$. Die Entropie $H(X)$ ist definiert als

$$H(X) = \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{1}{p(x)}.$$

Bemerkung 3.2.3

1. Den Logarithmus hätte man ebenso zu einer anderen Basis als 2 wählen können. So hat die Entropie nun die Einheit Bit.
2. Nach den Regeln von l' Hospital ist $\lim p \log_2 \frac{1}{p} = 0$ für $p \rightarrow 0$, was eine Setzung von $0 \log_2 \frac{1}{0} = 0$ rechtfertigt.
3. Die Entropie kann ebenso als Erwartungswert von $\log_2 \frac{1}{p(X)}$ (Information) betrachtet werden.

Beispiel und Bemerkung 3.2.4

Sei X eine $\mathcal{B}(1, p)$ -verteilte Zufallsvariable, dann ist

$$H(X) = -p \log p - (1 - p) \log(1 - p) =: H(p).$$

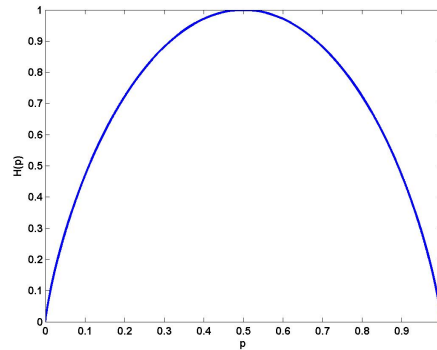


Abbildung 3.1: $H(p)$ in Abhängigkeit von p

Die Abbildung 3.2.1 verdeutlicht einige wichtige Eigenschaften der Entropie. $H(p)$ ist eine konkave Funktion und spiegelt das Maß der Unsicherheit gut wieder. Auf Grund der Tatsache, dass $0 \leq p(x) \leq 1$ und somit $\log_2 \frac{1}{p(x)} \geq 0$ gilt, ist stets $H(X) \geq 0$. In den Fällen $p = 0$ und $p = 1$ ist $H(p) = 0$, in Übereinstimmung damit, dass X nun über keinerlei Zufallscharakter verfügt. Dagegen wird die Unsicherheit für $p = \frac{1}{2}$ maximal, was sich an späterer Stelle auch allgemeiner formulieren lassen wird.

Die Entropie für eine Zufallsvariable wird nun ausgedehnt auf ein Paar von Zufallsvariablen.

Definition 3.2.5 (gemeinsame Entropie)

Die gemeinsame Entropie $H(X, Y)$ eines Paares von diskreten Zufallsvariablen (X, Y) mit (gemeinsamer) Dichte $p(x, y)$ ist definiert als

$$H(X, Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 \frac{1}{p(x, y)}.$$

Oder anders ausgedrückt:

$$H(X, Y) = E \left(\log_2 \frac{1}{p(X, Y)} \right)$$

Ebenfalls wird die bedingte Entropie definiert:

Definition 3.2.6 (bedingte Entropie)

Sei wieder (X, Y) ein Zufallsvektor und $p(x, y)$ die zugehörige Wahrscheinlichkeitsdichte, dann ist die bedingte Entropie

$H(Y|X)$ definiert als

$$\begin{aligned} H(Y|X) &= \sum_{x \in \mathcal{X}} p(x) H(Y|X = x) \\ &= - \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log_2 p(y|x) \\ &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(y|x) \\ &= - E_{p(x,y)} (\log_2 p(Y|X)). \end{aligned}$$

Den Zusammenhang von Entropie, gemeinsamer und bedingter Entropie charakterisiert der folgende Satz.

Satz 3.2.7 (Kettenregel)

$$H(X, Y) = H(X) + H(Y|X)$$

Beweis

$$\begin{aligned} H(X, Y) &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(x, y) \\ &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(x) p(y|x) \\ &= - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(x) - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(y|x) \\ &= - \sum_{x \in \mathcal{X}} p(x) \log_2 p(x) - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 p(y|x) \\ &= H(X) + H(Y|X) \end{aligned}$$

□

Ein Maß für die Distanz zweier Verteilungen stellt die relative Entropie dar.

Definition 3.2.8 (relative Entropie)

Die relative Entropie oder Kullback Leibler Divergenz zwischen zwei Wahrscheinlichkeitsdichten $p(x)$ und $q(x)$ ist definiert als

$$\begin{aligned} D(p||q) &= \sum_{x \in \mathcal{X}} p(x) \log_2 \frac{p(x)}{q(x)} \\ &= E_p \left(\log_2 \frac{p(x)}{q(x)} \right). \end{aligned}$$

Mit folgendem Satz kann eine obere Grenze für die Entropie bestimmt und an späterer Stelle auf eine wichtige Eigenschaft der wechselseitigen Informationen geschlossen werden.

Satz 3.2.9

Seien $p(x)$, $q(x)$, $x \in \mathcal{X}$, zwei Wahrscheinlichkeitsdichten. Dann gilt:

$$D(p||q) \geq 0 \text{ und}$$

$$D(p||q) = 0 \Leftrightarrow p(x) = q(x) \text{ für alle } x \in \mathcal{X}.$$

Für den Beweis wird die Jensen'sche Ungleichung benötigt, die hier nur ohne Beweis angegeben wird.

Hilfssatz 3.2.10 (Jensen'sche Ungleichung)

Sei X eine Zufallsvariable und f eine konkave oder konvexe Funktion, dann gilt

$$E(f(X)) \begin{cases} \geq f(E(X)) & \text{wenn } f \text{ konvex} \\ \leq f(E(X)) & \text{wenn } f \text{ konkav.} \end{cases}$$

Ist f sogar streng konvex bzw. konkav, so impliziert Gleichheit, dass $X = E(X)$ mit Wahrscheinlichkeit 1 gilt, also X konstant ist.

Beweis von Satz 3.2.9

Sei $A = \{x : p(x) > 0\}$ der Träger von $p(x)$. Dann gilt:

$$\begin{aligned} -D(p||q) &= -\sum_{x \in A} p(x) \log_2 \frac{p(x)}{q(x)} \\ &= \sum_{x \in A} p(x) \log_2 \frac{q(x)}{p(x)} \\ &\leq \log_2 \sum_{x \in A} p(x) \frac{q(x)}{p(x)} \\ &= \log_2 \sum_{x \in A} q(x) \\ &\leq \log_2 \sum_{x \in \mathcal{X}} q(x) \\ &= \log_2 1 \\ &= 0 \end{aligned}$$

Die erste Ungleichung folgt aus der Tatsache, dass $\log_2 t$ konkav ist, und der Jensen'schen Ungleichung. Da der Logarithmus zur Basis 2 insbesondere eine streng konkave Funktion ist, liegt stattdessen genau dann eine Gleichheit vor, wenn überall $q(x)/p(x) = 1$, also $q(x) = p(x)$ gilt. Damit wurde gezeigt, dass $D(p||q) = 0$ genau dann vorliegt, wenn $p(x) = q(x)$ für alle x gilt. \square

Satz 3.2.11

Bezeichne $|\mathcal{X}|$ die Anzahl Elemente der Bildmenge von X . Es gilt $H(X) \leq \log |\mathcal{X}|$, wobei Gleichheit genau dann besteht, wenn X eine Gleichverteilung über \mathcal{X} besitzt.

Beweis

Sei $u(x) = \frac{1}{|\mathcal{X}|}$ die Zähldichte der Gleichverteilung über \mathcal{X} und sei $p(x)$ die Zähldichte von X . Dann gilt

$$\begin{aligned} D(p||u) &= \sum p(x) \log \frac{p(x)}{u(x)} \\ &= \sum p(x) \log \frac{1}{u(x)} - \sum p(x) \log \frac{1}{p(x)} \\ &= \log |\mathcal{X}| \sum p(x) - H(X) \\ &= \log |\mathcal{X}| - H(X) \end{aligned}$$

Mit der Nicht-Negativität der relativen Entropie (Satz 3.2.9) folgen die Behauptungen. \square

Nun kann auch erklärt werden, warum in Beispiel 3.2.4 für $p = \frac{1}{2}$ die Entropie mit $H(p) = 1$ maximal ist. So liegt hier eine Gleichverteilung vor mit $|\mathcal{X}| = 2$ und $\log_2(2) = 1$.

3.2.2 Wechselseitige Information und Bias

Da jetzt alle benötigten Begriffe und Aussagen eingeführt wurden, kann die wechselseitige Information definiert werden. Sie ist ein Maß für die Menge an Information, die eine Zufallsvariable über eine andere Zufallsvariable enthält.

Definition 3.2.12 (wechselseitige Information)

Seien X und Y zwei Zufallsvariablen mit gemeinsamer Wahrscheinlichkeitsdichte $p(x, y)$ und Randdichten $p(x)$ und $p(y)$. Die wechselseitige Information $I(X; Y)$ ist die relative Entropie zwischen der gemeinsamen Dichte und der Produktdichte, das heißt

$$\begin{aligned} I(X; Y) &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \\ &= D(p(x, y) || p(x)p(y)) \\ &= E_{p(x, y)} \left(\log_2 \frac{p(X, Y)}{p(X)p(Y)} \right). \end{aligned}$$

Nun werden einige wichtige Eigenschaften der wechselseitigen Information aufgeführt, wie etwa, dass sie stets nicht-negativ und genau dann Null ist, wenn X und Y unabhängig sind (vgl. auch [H00]).

Satz 3.2.13

Seien X und Y zwei Zufallsvariablen, dann gilt:

1. $I(X; Y) = I(Y; X)$
2. $I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$
3. $I(X; Y) = H(X) + H(Y) - H(X, Y)$
4. $I(X; Y) \geq 0$ und $I(X; Y) = 0 \Leftrightarrow X$ und Y sind stochastisch unabhängig.
5. $I(X; Y) \leq \min \{H(X), H(Y)\}$

Beweis

1. Die erste Eigenschaft ist trivial.
2. Es wird nur die erste Gleichheit gezeigt, da der Beweis von $I(X; Y) = H(Y) - H(Y|X)$

analog verläuft.

$$\begin{aligned}
I(X; Y) &= \sum_{x,y} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \\
&= \sum_{x,y} p(x, y) \log_2 \frac{p(x|y)}{p(x)} \\
&= - \sum_{x,y} p(x, y) \log_2 p(x) + \sum_{x,y} p(x, y) \log_2 p(x, y) \\
&= - \sum_x \left(\log_2 p(x) \sum_y p(x, y) \right) + \sum_{x,y} p(x, y) \log_2 p(x, y) \\
&= - \sum_x p(x) \log_2 p(x) - \left(- \sum_{x,y} p(x, y) \log_2 p(x|y) \right) \\
&= H(X) - H(X|Y).
\end{aligned}$$

3. Folgt aus 2. und Satz 3.2.7.

4. Nach Satz 3.2.8 gilt $I(X; Y) = D(p(x, y) || p(x)p(y)) \geq 0$, wobei die Gleichheit genau dann besteht, wenn $p(x, y) = p(x)p(y)$, also X und Y stochastisch unabhängig sind.

5. Nach 2. und 4. gilt $0 \leq I(X; Y) = H(X) - \underbrace{H(X|Y)}_{\geq 0} \Rightarrow I(X; Y) \leq H(X)$ und entsprechend für Y . □

Die (relative) wechselseitige Information wird von Herwig ([H00]) genutzt, um die Güte des Clusters zu bestimmen. Hier wird sie aber, wie bereits erwähnt, angewendet um die Güte des stimulusabhängigen Clusters, also die Stimulusabhängigkeit der Einteilung, zu bewerten.

Bemerkung 3.2.14

Die aus unserem Algorithmus des stimulusabhängigen Clusters erhaltene Stimulus-/ Antwortklassenmatrix N lässt sich auch als Kontingenztafel zweier Zufallsvariablen X mit Wertemenge $\mathcal{X} = \{s_1, s_2, \dots, s_K\}$ und Y mit Wertemenge $\mathcal{Y} = \{r_1, r_2, \dots, r_K\}$ interpretieren. Dann handelt es sich bei dem von Victor ([VP97]) angegebenen Maß zur Bestimmung

	Y				
		r_1	r_2	\dots	r_K
X					
s_1		n_{11}	n_{12}	\dots	n_{1K}
s_2		n_{21}	n_{22}	\dots	n_{2K}
\vdots		\vdots	\vdots	\vdots	\vdots
s_K		n_{K1}	n_{K2}	\dots	n_{KK}

der Güte des stimulusabhängigen Clusters

$$I := \frac{1}{KL} \sum_{i,j} n_{ij} \left(\log_2 n_{ij} - \log_2 \sum_i n_{ij} - \log_2 \sum_j n_{ij} + \log_2 KL \right)$$

gerade um einen Schätzer der wechselseitigen Information von X und Y . Bezeichne $\hat{p}(x)$, $\hat{p}(y)$ und $\hat{p}(x, y)$ die jeweiligen empirischen Dichten, dann gilt:

$$\begin{aligned} \hat{I}(X; Y) &= \sum_{x,y} \hat{p}(x, y) \log_2 \frac{\hat{p}(x, y)}{\hat{p}(x)\hat{p}(y)} \\ &= \sum_{i,j} \frac{n_{ij}}{KL} \log_2 \frac{\frac{n_{ij}}{KL}}{(\sum_i \frac{n_{ij}}{KL})(\sum_j \frac{n_{ij}}{KL})} \\ &= \frac{1}{KL} \sum_{i,j} n_{ij} \log_2 \frac{n_{ij}KL}{(\sum_i n_{ij})(\sum_j n_{ij})} \\ &= \frac{1}{KL} \sum_{i,j} n_{ij} \left(\log_2 n_{ij} - \log_2 \sum_i n_{ij} - \log_2 \sum_j n_{ij} + \log_2 KL \right) \end{aligned}$$

Mit Hilfe der vorhergehenden Resultate und einigen Äußerungen von Victor (vgl. etwa [VP97]) kann ich nun den Zusammenhang von wechselseitiger Information und der Güte des stimulusabhängigen Clusters charakterisieren. Mit den Bezeichnungen aus der vorhergehenden Bemerkung gilt:

Lemma 3.2.15

Zufälliges, also stimulusunabhängiges, Clustern bedeutet, dass die wechselseitige Information einen Wert von 0 annimmt. Dahingegen wird sie maximal für optimales stimulusabhängiges Clustern mit $\hat{I}(X; Y) = \log_2 K$.

Beweis

Zufälliges Clustern, also alle Einträge der Matrix N sind gleich, entspricht einer stochastischen Unabhängigkeit von X und Y . Die Behauptung ergibt sich direkt aus der vierten Aussage von Satz 3.2.13. Optimales stimulusabhängiges Clustern liegt dagegen vor, wenn die Diagonalelemente der Matrix N alle gleich L sind und alle anderen Einträge gleich 0 und somit sowohl X als auch Y gleichverteilt sind. Leichtes Nachrechnen zusammen mit Satz 3.2.11 und der dritten Aussage von Satz 3.2.13 zeigt:

$$I(X; Y) = \log_2 K + \log_2 K - K \cdot \left(\frac{1}{K} \log_2 K \right) = \log_2 K.$$

Nach der fünften Eigenschaft von Satz 3.2.13 und Satz 3.2.11 ist dies aber gerade der maximale Wert, den die wechselseitige Information annehmen kann. \square

Bemerkung 3.2.16

1. Die wechselseitige Information wird bei Vorliegen jeder Matrix maximal, bei der in jeder Zeile und jeder Spalte genau ein Eintrag gleich L und Null sonst vorkommt, da dies ebenfalls einer Gleichverteilung von X und Y entspricht und $H(X;Y)$ den gleichen Wert wie im zuvor betrachteten Fall annimmt. Hohe Werte der wechselseitigen Information bedeuten also nicht zwingend, dass die Zuordnung der Spiketrains in Antwortklassen der als optimal verstandenen stimulusabhängigen entspricht, sondern allgemeiner einer systematisch stimulusabhängigen Einteilung.
2. Es sei auch darauf hingewiesen, dass ein Wert von Null für die wechselseitige Information nicht zwingend ein zufälliges Zuordnen der Spiketrains bedeutet. Werden etwa alle in die gleiche Antwortklasse eingeteilt, so gilt $H(X) = H(X;Y)$ und $H(Y) = 0$ und somit $I(X;Y) = 0$ nach der dritten Aussage von Satz 3.2.13.
3. Sind die Stimulusklassen nicht gleich groß und liegt der Zufallsvariable X somit keine Gleichverteilung zugrunde, so gilt $I(X;Y) \leq H(X)$. Man erinnere sich, dass stets $|\mathcal{X}|=|\mathcal{Y}|$ gilt, dann folgt dies wieder aus der fünften Eigenschaft von Satz 3.2.13 und Satz 3.2.11.

Bei der Bestimmung der wechselseitigen Information über die empirischen Dichten ist zu beachten, dass es sich hierbei um keinen erwartungstreuen Schätzer handelt. Dies lässt sich über die Verfälschtheit des Schätzers der Entropie erklären.

Satz 3.2.17

Sei X eine Zufallsvariable mit K -elementiger Wertemenge $\mathcal{X} = \{s_1, s_2, \dots, s_K\}$ und Wahrscheinlichkeitsdichten $p_i^X := P^X(s_i)$, $1 \leq i \leq K$. Bezeichne n_i die Häufigkeit mit der X bei einer n -fachen unabhängigen Versuchswiederholung den Wert r_i angenommen hat. Dann ist der Schätzer

$$\hat{H}(X) = - \sum_{i=1}^K \frac{n_i}{n} \log_2 \frac{n_i}{n}$$

für die Entropie nicht erwartungstreu und es gilt:

$$H(X) - E(\hat{H}(X)) = \frac{\log_2 e}{2n} (K-1) - \frac{\log_2 e}{12n^2} \left[1 - \sum_{i=1}^K \frac{1}{p_i^X} \right] + O\left(\frac{1}{n^3}\right), \quad (3.1)$$

wobei $O\left(\frac{1}{n^3}\right)$ einen Term der Ordnung $\frac{1}{n^3}$ bedeutet.

Die Gleichung 3.1 zeigt, dass $\hat{H}(X)$ den wahren Wert der Entropie $H(X)$ unterschätzt. Es handelt sich hierbei um eine der ersten Abschätzungen für den Bias, die in [M55] von Miller hergeleitet wird. Ist die Anzahl der Beobachtungen n ähnlich groß wie K , also die Anzahl der Werte, die X annehmen kann, sollte diese aber nicht verwendet werden. Ist ausserdem X gleichverteilt, sei für die entsprechende Abschätzung des Bias auf eine Arbeit von Rogers und Green [RG55] verwiesen.

Für den Bias der wechselseitigen Information erhält man nun ([M55]):

Satz 3.2.18

Seien X und Y zwei Zufallsvariablen mit Wertemenge $\mathcal{X} = \{s_1, s_2, \dots, s_K\}$ beziehungsweise $\mathcal{Y} = \{r_1, r_2, \dots, r_C\}$ mit gemeinsamer Wahrscheinlichkeitsdichte $p_{ij}^{XY} := P^{X,Y}(s_i, r_j)$ und Randdichten p_i^X beziehungsweise p_j^Y mit $1 \leq i \leq K$ und $1 \leq j \leq C$. Bezeichne n_{ij} die Häufigkeit mit der bei einer n -fachen unabhängigen Versuchswiederholung der Zufallsvektor (X, Y) den Wert (s_i, r_j) angenommen hat und $n_{i\cdot} = \sum_j n_{ij}$. Dann ist der Schätzer

$$\hat{I}(X; Y) = \frac{1}{n} \sum_{i,j} \frac{n_{ij}}{n} \log_2 \frac{n \cdot n_{ij}}{n_{i\cdot} \cdot n_{\cdot j}}$$

nicht erwartungstreu und es gilt:

$$\begin{aligned} & I(X; Y) - E(\hat{I}(X; Y)) \\ &= -\frac{\log_2 e}{2n} (KC - K - C + 1) - \frac{\log_2 e}{12n^2} \left(\sum_i \frac{1}{p_i^X} + \sum_j \frac{1}{p_j^Y} - \sum_{i,j} \frac{1}{p_{ij}^{XY}} - 1 \right) + O\left(\frac{1}{n^3}\right) \end{aligned} \quad (3.2)$$

Beweis

Aus der dritten Eigenschaft von Satz 3.2.13 folgt zusammen mit dem vorhergehenden Satz:

$$\begin{aligned} E(\hat{I}(X; Y)) &= E(\hat{H}(X)) + E(\hat{H}(Y)) - E(\hat{H}(X, Y)) \\ &= \left[H(X) - \frac{\log_2 e}{2n} (K - 1) - \frac{\log_2 e}{12n^2} \left(1 - \sum_i \frac{1}{p_i^X} \right) \right] + \\ &\quad \left[H(Y) - \frac{\log_2 e}{2n} (C - 1) - \frac{\log_2 e}{12n^2} \left(1 - \sum_j \frac{1}{p_j^Y} \right) \right] - \\ &\quad \left[H(X, Y) - \frac{\log_2 e}{2n} (KC - 1) - \frac{\log_2 e}{12n^2} \left(1 - \sum_{i,j} \frac{1}{p_{ij}^{XY}} \right) \right] + O\left(\frac{1}{n^3}\right) \\ &= I(X; Y) + \frac{\log_2 e}{2n} [(KC - 1) - (K - 1) - (C - 1)] + \\ &\quad \frac{\log_2 e}{12n^2} \left[\left(1 - \sum_{i,j} \frac{1}{p_{ij}^{XY}} \right) - \left(1 - \sum_i \frac{1}{p_i^X} \right) - \left(1 - \sum_j \frac{1}{p_j^Y} \right) \right] + O\left(\frac{1}{n^3}\right) \\ &= I(X; Y) + \frac{\log_2 e}{2n} (KC - K - C + 1) + \\ &\quad \frac{\log_2 e}{12n^2} \left(\sum_i \frac{1}{p_i^X} + \sum_j \frac{1}{p_j^Y} - \sum_{i,j} \frac{1}{p_{ij}^{XY}} - 1 \right) + O\left(\frac{1}{n^3}\right) \end{aligned}$$

Hieraus erhält man den behaupteten Bias. □

Der Satz besagt also, dass der wahre Wert der wechselseitigen Information überschätzt wird. Folgendes, für die spätere Anwendung relevante, Beispiel möge dies verdeutlichen.

Beispiel 3.2.19

Mit den Bezeichnungen aus Bemerkung 3.2.14 seien X und Y zwei gleichverteilte stochastisch unabhängige Zufallsvariablen. Dann ist $I(X; Y) = 0$, aber bei einer Schätzung erhält man nach Satz 3.2.18 je nach Anzahl der Stimulusklassen folgenden Bias:

(a) Mit $K = C = 9$ und $L = 640$, also $n = KL = 5760$ gilt:

$$\begin{aligned}
& I(X; Y) - E(\widehat{I}(X; Y)) \\
&= -\frac{\log_2 e}{2n} (K^2 - 2K + 1) - \frac{\log_2 e}{12n^2} (2K^2 - K^4 - 1) + O\left(\frac{1}{n^3}\right) \\
&= -\frac{\log_2 e}{2 \cdot 5760} (81 - 18 + 1) - \frac{\log_2 e}{12 \cdot 5760^2} (2 \cdot 81 - 6561 - 1) + O\left(\frac{1}{n^3}\right) \quad (3.3) \\
&= 0.008015 + 0.000023 + O\left(\frac{1}{n^3}\right) \\
&= 0.008038 + O\left(\frac{1}{n^3}\right)
\end{aligned}$$

(b) Mit $K = C = 72$ und $L = 80$, also $n = KL = 5760$ gilt:

$$\begin{aligned}
& I(X; Y) - E(\widehat{I}(X; Y)) \\
&= -\frac{\log_2 e}{2 \cdot 5760} (5184 - 144 + 1) - \frac{\log_2 e}{12 \cdot 5760^2} (2 \cdot 5184 - 26873856 - 1) + O\left(\frac{1}{n^3}\right) \\
&= 0.6313 + 0.0973 + O\left(\frac{1}{n^3}\right) \\
&= 0.7286 + O\left(\frac{1}{n^3}\right) \quad (3.4)
\end{aligned}$$

Es sei angemerkt, dass mit $72 \cdot 72 < 72 \cdot 80 = 5760$ die empfohlene Voraussetzung zur Anwendung von 3.1 zur Abschätzung des Bias für die gemeinsame Entropie $H(X; Y)$ und somit für die wechselseitige Information gerade noch erfüllt ist.

Bemerkung 3.2.20

Bei der Schätzung des Bias ist wieder zu beachten, dass diese problematisch werden könnte, wenn n klein ist. Viele der p_{ij}^{XY} können fast Null sein, so dass es nach [M55] besser sein könnte, ein Model zu betrachten, in dem viele der p_{ij}^{XY} gleich Null gesetzt werden. Dementsprechend muss dann für jede dieser Setzungen die Anzahl der Freiheitsgrade von $KC - K - C + 1 = (K - 1)(C - 1)$ je um eins reduziert werden.

Im Folgenden wird von Prozent-Korrekt und wechselseitiger Information als Fehler- oder Gütemaße gesprochen.

4 Simulationen

4.1 Poisson-Prozesse

Zur Analyse der Abstandsmaße ist eine Anwendung auf künstliche Daten sinnvoll. Diese sollen mit Hilfe von Poisson-Prozessen erzeugt werden, die häufig genutzt werden um die elektrische Aktivität von Nervenzellen zu simulieren. Diese Näherung wird unter anderem verwendet, weil ein Neuron nie zwei Aktionspotentiale zur gleichen Zeit generiert und die Verteilung von Interspikeintervallen der einer Exponentialverteilung ähnelt. Da es bei meiner Analyse der Abstandsmaße nicht das Ziel ist, das Antwortverhalten möglichst adäquat nach zu bilden, werden die Poisson-Prozesse im Folgenden kaum neurobiologisch motiviert.

Bevor nun ein homogener Poisson-Prozess definiert werden kann, werden noch einige Begriffe benötigt. Dabei basieren die folgenden Aussagen, soweit nicht anders angegeben, auf [MS05].

Definition 4.1.1 (Wartezeit-, Sprungzeit-, Sprungprozess)

Sei $(X_t)_{t \geq 0}$ ein rechtsstetiger stochastischer Prozess mit Zustandsraum \mathbb{N}_0 . Dann ist der Prozess der Sprungzeiten $(T_n)_{n \in \mathbb{N}_0}$ induktiv definiert durch

$$T_0 := 0 \quad \text{und} \quad T_{n+1} := \inf \{t \geq T_n : X_t \neq X_{T_n}\}.$$

Dabei ist $\inf \emptyset = \infty$. Der Prozess der Wartezeiten $(W_n)_{n \in \mathbb{N}}$ ist definiert durch

$$W_n := \begin{cases} T_n - T_{n-1} & \text{für } T_{n-1} < \infty \\ \infty & \text{sonst.} \end{cases}$$

Der Sprungprozess $(S_n)_{n \in \mathbb{N}_0}$ ist definiert durch

$$S_n := \begin{cases} X_{T_n} & \text{für } T_n < \infty \\ X_a, \text{ mit } a := \max \{r \in \mathbb{N}_0 : T_r < \infty\} & \text{für } T_n = \infty. \end{cases}$$

Im neuronalen Kontext ist die Sprungzeit gerade der Zeitpunkt, zu dem die Zelle ein Aktionspotential generiert. Dementsprechend sind die Wartezeiten gerade die Zeiten zwischen zwei Spikes, also die Interspikeintervalle.

Definition 4.1.2 (homogener Poisson-Prozess)

Ein rechtsstetiger Prozess $(N_t)_{t \geq 0}$ mit Zustandsraum \mathbb{N}_0 und $N_0 = 0$ fast sicher heißt homogener Poisson-Prozess mit der Rate λ , falls die folgenden zwei Bedingungen gelten:

- (i) Die Folge der Wartezeiten $(W_n)_{n \in \mathbb{N}}$ ist unabhängig, und für jedes $n \in \mathbb{N}$ ist W_n $\text{Exp}(\lambda)$ -verteilt.
- (ii) Der Sprungprozess $(S_n)_{n \in \mathbb{N}_0}$ ist gegeben durch

$$S_n = n, \quad n \in \mathbb{N}_0.$$

Der folgende Satz ist für die Konstruktion von Spiketrains hilfreich. Ausserdem verdeutlicht er, dass dem Sprungprozess gerade der Prozess der Summe der Spikeanzahlen entspricht. Natürlich ist dieser nur theoretisch gesehen unendlich und in Wirklichkeit durch das Absterben einer Zelle begrenzt.

Satz 4.1.3

Sei $\lambda > 0$ und $(W_n)_{n \in \mathbb{N}}$ eine Folge unabhängiger $\text{Exp}(\lambda)$ -verteilter Zufallsvariablen. Setze

$$T_0 := 0 \quad \text{und} \quad T_n := \sum_{i=1}^n W_i, \quad n \in \mathbb{N}$$

sowie

$$N_t := \sum_{i=0}^{\infty} I_{]0,t]}(T_i), \quad t \geq 0$$

Dann ist $(N_t)_{t \geq 0}$ ein Poisson-Prozess mit der Rate λ .

Für den Beweis wird der folgende Hilfssatz benötigt.

Hilfssatz 4.1.4

Sei $(X_t)_{t \geq 0}$ ein stochastischer Prozess mit Zustandsraum \mathbb{N}_0 . Dann ist $(X_t)_{t \geq 0}$ genau dann rechtsstetig, wenn für fast alle $\omega \in \Omega$ zu jedem $t \geq 0$ ein $\epsilon > 0$ existiert, so dass

$$X_t(\omega) = X_s(\omega) \quad \text{für alle } t \leq s \leq t + \epsilon.$$

Beweis von Satz 4.1.3

Um nachzuweisen, dass $(N_t)_{t \geq 0}$ ein stochastischer Prozess ist, muss gezeigt werden, dass für jedes $n \in \mathbb{N}$ die Menge der Urbilder $\{N_t = n\}$ für alle $t \in [0, \infty[$ messbar ist. Da die Exponentialverteilung fast sicher positiv ist, gilt $P(T_{n+1} - T_n > 0) = P(W_{n+1} > 0) = 1$ für alle $n \in \mathbb{N}$, also $T_0 < T_1 < T_2 < \dots$ fast sicher. Mit W_i sind auch T_n messbare Abbildungen, so dass für jedes $n \in \mathbb{N}$ insgesamt folgt: $\{N_t = n\} = \{T_n \leq t < T_{n+1}\}$ ist messbar.

Nach Hilfssatz 4.1.4 ist $(N_t)_{t \geq 0}$ rechtsstetig, denn: Aus $P(T_{n+1} - T_n > 0) = 1$ folgt, dass es zu fast allen $\omega \in \Omega$ und $T_n(\omega) \leq t < T_{n+1}(\omega)$ ein $\epsilon > 0$ gibt, so dass auch $T_n(\omega) \leq t + \epsilon < T_{n+1}(\omega)$ und damit $N_{t+\epsilon}(\omega) = n = N_t(\omega)$ gilt.

Die verbleibenden Aussagen von 4.1.2 lassen sich mit Hilfe von Definition 4.1.1 zeigen. Seien dazu $(\tilde{T}_n)_{n \in \mathbb{N}_0}$ die Sprungzeiten des Prozesses $(N_t)_{t \geq 0}$. Auf Grund der strengen Monotonie der $(T_n)_{n \in \mathbb{N}_0}$ gilt:

$$N_{T_n} := \sum_{i=0}^{\infty} I_{]0, T_n]}(T_i) = \sum_{i=1}^n 1 = n$$

Zunächst wird hiermit induktiv gezeigt, dass $\tilde{T}_n = T_n$ für $n \geq 0$ gilt. Für $n = 0$ erhält man dies per Definition, für den Induktionsschritt benutzt man nun 4.1.1:

$$\begin{aligned} \tilde{T}_{n+1} &= \inf\{t \geq T_n : N_t \neq N_{T_n}\} \\ &= \inf\{t \geq T_n : \sum_{i=0}^{\infty} I_{]0, t]}(T_i) \neq n\} \\ &= \inf\{t \geq T_n : \sum_{i=n+1}^{\infty} I_{]0, t]}(T_i) > 0\} \\ &= \inf\{t \geq T_n : t \geq T_{n+1}\} = T_{n+1} \end{aligned}$$

Die Sprungzeiten des Prozesses $(N_t)_{t \geq 0}$ sind also gerade $(T_n)_{n \in \mathbb{N}_0}$. Hieraus folgt nach Definition 4.4.1, dass (W_n) die Wartezeiten sind. Diese sind nach Voraussetzung unabhängig und $\text{Exp}(\lambda)$ -verteilt. Nun kann man auch schließen, dass der Sprungprozess $(S_n)_{n \in \mathbb{N}_0}$ gegeben ist durch $S_n = N_{T_n} = n$ für alle $n \in \mathbb{N}_0$. Damit wurde bewiesen, dass $(N_t)_{t \geq 0}$ ein homogener Poisson-Prozess mit Rate λ ist. \square

Mit Hilfe der folgenden Aussage können die exponentialverteilten Zufallsvariablen erzeugt und somit nach Satz 4.1.3 die Sprungzeiten eines homogenen Poisson-Prozess simuliert werden (Beweis siehe [PS07]).

Lemma 4.1.5

Sei F eine beliebige Verteilungsfunktion und Y eine über dem Einheitsintervall $[0, 1]$ stetig gleichverteilte Zufallsvariable, so besitzt die Zufallsvariable $X := F^{-1}(Y)$ die Verteilungsfunktion F , wobei F^{-1} die Pseudo-Inverse von F bezeichne.

Zur Analyse und einem besseren Verständnis der Abstandsmaße ist es sinnvoll auch Prozesse zu betrachten, bei dem die Interspike-Intervalle weniger stark variieren. Victor beschreibt in [VP97], dass sich eine solche Simulation durch das Herausgreifen jedes k -ten Spikezeitpunkts eines homogenen Poisson-Prozesses konstruieren lässt. Ich habe dies wie folgt formal zusammengefasst:

Definition 4.1.6 (iterierter Poisson-Prozess)

Sei $(N_t)_{t \geq 0}$ ein homogener Poisson-Prozess mit Rate $k\lambda$ mit einem $k \in \mathbb{N}$, Sprungzeiten T_n und Wartezeiten W_n . Setze

$$T_0^* := T_0 \quad \text{und} \quad T_n^* := \sum_{i=1}^{k(n-1)+z} W_i, \quad n \in \mathbb{N} \quad \text{und} \quad z \in \{1, 2, \dots, k-1, k\} \text{ fest}$$

sowie

$$N_t^* := \sum_{i=0}^{\infty} I_{]0, t]}(T_i^*), \quad t \geq 0$$

Dann heißt $(N_t^*)_{t \geq 0}$ ein iterierter Poisson-Prozess der Ordnung k mit Rate λ .

Die Charakterisierung der Verteilung der Wartezeiten eines iterierten Poisson-Prozesses der Ordnung k mit Rate λ erfordert folgende Bemerkung:

Lemma 4.1.7

Seien Y_1, \dots, Y_n unabhängig und $\text{Exp}(\lambda)$ -verteilt. Dann besitzt $Z_n := \sum_{i=1}^n Y_i$ die Dichte

$$d_{Z_n} : \mathbb{R} \rightarrow \mathbb{R},$$

$$d_{Z_n}(t) := \begin{cases} \lambda \exp(-\lambda t) \cdot \frac{(\lambda t)^{n-1}}{(n-1)!} & \text{für } t > 0, \\ 0 & \text{für } t \leq 0 \end{cases}$$

Z_n ist also $\Gamma(n, \lambda)$ -verteilt und es gilt: $E(Z_n) = \frac{n}{\lambda}$ und $\text{Var}(Z_n) = \frac{n}{\lambda^2}$.

Beweis

Der Beweis, dass Z_n $\Gamma(n, \lambda)$ -verteilt ist, wird per Induktion geführt. Für $n = 1$ erhält man

$$d_{Z_1} = \begin{cases} \lambda \exp(-\lambda t) & \text{für } t > 0, \\ 0 & \text{für } t \leq 0, \end{cases}$$

also gerade die Dichte der Exponentialverteilung. Der Induktionsschritt $n \rightarrow n+1$ lautet:

$$\begin{aligned} d_{Z_{n+1}}(t) &= \int_0^t d_{Z_n}(s) \lambda \exp(-\lambda(t-s)) ds \\ &= \int_0^t \lambda \exp(-\lambda s) \cdot \frac{(\lambda s)^{n-1}}{(n-1)!} \lambda \exp(-\lambda(t-s)) ds \\ &= \exp(-\lambda t) \lambda^2 \int_0^t \frac{(\lambda s)^{n-1}}{(n-1)!} ds \\ &= \lambda \exp(-\lambda t) \frac{(\lambda t)^n}{n!}. \end{aligned}$$

Dabei basiert die Darstellung der Faltung im ersten Schritt auf der Unabhängigkeit von Z_n und Y_{n+1} (Beweis siehe [PS07]).

Erwartungswert und Varianz folgen direkt aus deren Eigenschaften und der Tatsache, dass $E(Y_i) = \frac{1}{\lambda}$ und $Var(Y_i) = \frac{1}{\lambda^2}$ für $1 \leq i \leq n$ gilt. \square

Nun kann ich zeigen, dass, wie in [VP97] behauptet, die Wartezeiten eines iterierten Poisson-Prozesses für $k > 1$ regulärer sind als im homogenen Fall.

Lemma 4.1.8

Im Fall eines homogenen Poisson-Prozess mit Rate λ haben die Interspike-Intervalle W_n einen Variationskoeffizienten von $VarK = 1$.

Liegt dagegen ein iterierter Poisson-Prozesses der Ordnung k mit Rate λ vor, so gilt $VarK = \frac{1}{\sqrt{k}}$.

Beweis

Seien W_n die Wartezeiten eines homogenen Poisson-Prozesses mit Rate λ , dann gilt:

$$VarK = \frac{\sqrt{Var(W_n)}}{E(W_n)} = \frac{1/\lambda}{1/\lambda} = 1.$$

Mit den Bezeichnungen aus Definition 4.1.6 gilt für die Wartezeiten $(W_n^*)_{n \in \mathbb{N}}$ eines iterierten Poisson-Prozesses der Ordnung k :

$$W_1^* = T_1^* \text{ und } W_n^* := \sum_{i=z+(n-2)k+1}^{k(n-1)+z} W_i, \quad n \in \mathbb{N}_{\geq 2}$$

Die Wartezeiten sind also gebildet aus der Summe von k unabhängigen $Exp(k\lambda)$ -verteilten Zufallsvariablen, so dass nach Lemma 4.1.7 folgt, dass $W_i^* \Gamma(k, k\lambda)$ -verteilt sind. Daher erhält man hierfür einen Variationskoeffizienten von

$$VarK = \frac{\sqrt{Var(W_n^*)}}{E(W_n^*)} = \frac{\sqrt{k/(k\lambda)^2}}{k/k\lambda} = \frac{\sqrt{k}}{k\lambda} \frac{k\lambda}{k} = \frac{1}{\sqrt{k}}.$$

\square

Allerdings feuern Neurone nicht in jedem gleichgroßen Zeitintervall mit gleicher Wahrscheinlichkeit. Die Zuwächse der Aktionspotentiale sind also nicht stationär wie es durch einen homogenen Poisson-Prozess beschrieben wird. Das Wirken der Abstandsmaße sollte also auch auf nichthomogenen Poisson-Prozessen untersucht werden, die eine etwas angemessenere Charakterisierung der elektrischen Aktivität von Nervenzellen darstellen.

Definition 4.1.9 (Nicht homogener Poisson-Prozess)

Es sei $(N_t)_{t \geq 0}$ ein rechtsstetiger Prozess mit Zustandsraum \mathbb{N}_0 und $\lambda : \mathbb{R}^+ \rightarrow \mathbb{R}_{\geq 0}$ eine messbare Abbildung. Dann heißt $(N_t)_{t \geq 0}$ nicht homogener Poisson-Prozess mit der lokalen Rate λ , wenn folgende Bedingungen gelten:

- (i) $N_0 = 0$ fast sicher.
- (ii) $(N_t)_{t \geq 0}$ hat unabhängige Zuwächse.
- (iii) Für $t \geq s$ ist $N_t - N_s$ Poisson-verteilt mit der Rate

$$\int_s^t \lambda(u) du.$$

Im Allgemeinen sind die Wartezeiten eines nicht homogenen Poisson-Prozesses weder exponentialverteilt noch unabhängig. Eine Simulation der Sprungzeiten kann also nicht ähnlich wie im homogenen Fall erfolgen. In Kapitel 4.3 wird aber beschrieben, wie man Spiketrains mit Hilfe von Definition 4.1.8 erzeugen kann.

4.2 Anwendung auf künstliche Daten

Mit Hilfe des vorangehenden Kapitels werden einige Simulationen durchgeführt um einerseits durch den Vergleich zu [VP97] die Korrektheit meiner implementierten Algorithmen für D^{spike} und $D^{interval}$ zu prüfen und andererseits um deren Wirkungsweise im Vergleich zu D^{euclid} zu analysieren. Ausserdem können so die Maße Prozent-Korrekt und die wechselseitige Information zur Bestimmung der Güte des stimulusabhängigen Clusters besser gegenüber gestellt werden.

Berechnung und Implementierung

Es werden jeweils die Distanzmatrizen für $D^{spike}[q]$ und $D^{interval}[q]$ mit Werten von q von 0, 0.25, 0.5, 1, 2, 4, 8, 16, 32, 64, 128 und $256 s^{-1}$ bestimmt. Für das Abstandsmaß $D^{euclid}[\tau]$ werden für τ die gerundeten Kehrwerte von q gewählt, um so die Güte des stimulusabhängigen Clusters für die verschiedenen Abstandsmaße auf einer entsprechenden Zeitskala vergleichen zu können. Bei der Implementierung des Algorithmus zur Berechnung von $D^{euclid}[\tau]$ wurde auf die Normierung mit der Anzahl an Datenpunkten der gefilterten Spiketrains nach Definition 2.1.1 verzichtet, da dies für die Bestimmung der Güte des stimulusabhängigen Clusters irrelevant ist. Die Verschiebung des Zeitfenster τ erfolgte stets um $v = 0.1ms$.

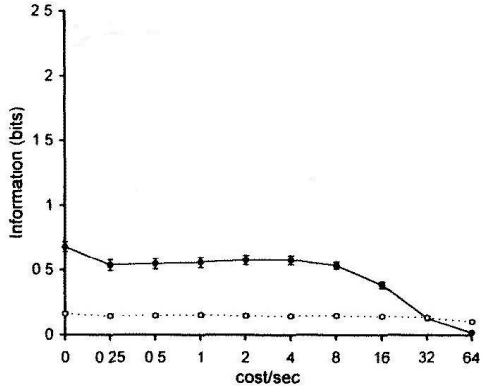
Beim Clustern mit einem negativen Exponenten z ist zu beachten, dass in *Matlab* für 0^z der Wert unendlich angenommen wird. Da wir aber von ausgewählten Einträgen der Distanzmatrix zur Potenz z den Mittelwert bestimmen, würde bereits ein solcher Ausreißer den Mittelwert beliebig verfälschen. Insbesondere würde das bedeuten, dass die mittleren Abstände zu zwei verschiedenen Stimulusklassen gleich wären, sofern mindestens je ein Abstand gleich Null wäre. Daher wird dieser unendliche Wert in allen folgenden Berechnungen auf 0.00001^z gesetzt.

4.2.1 Ratenunterscheidung

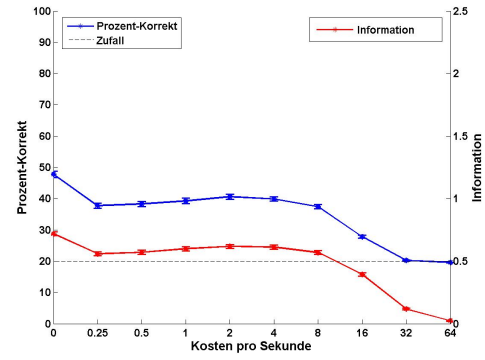
Homogener Poisson-Prozess

Es werden Antworten von einer Sekunde Länge erzeugt, so dass deren Summe der Spikeanzahlen einen homogenen Poisson-Prozess (betrachtet bis zum Zeitpunkt 1) bilden. Die Simulation stellt eine Zelle dar, der fünf verschiedene Stimuli je zwanzig mal präsentiert wurden, so dass man insgesamt hundert Spiketrains erhält. Auf die fünf Reize reagiert das künstliche Neuron je mit einer Spikerate von 2, 4, 6, 8 und 10 Spikes pro Sekunde. Um die Zuverlässigkeit der Schätzung der Güte des stimulusabhängigen Clusters bestimmen zu können, werden Daten dieser Form vierzig mal erzeugt.

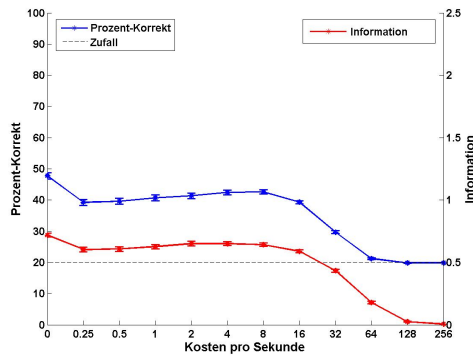
Die Güte des stimulusabhängigen Clusters mit Clusterexponent $z=-2$ im Mittel für die beschriebene Simulation ist in Abbildung 4.1 jeweils für die Fehlermaße Prozent-Korrekt



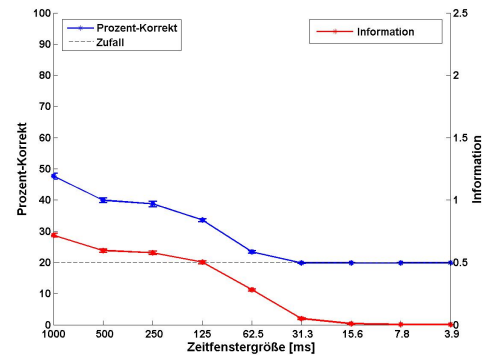
(a) D^{spike} modifiziert nach [VP97], dort Abb. 4A



(b) D^{spike}



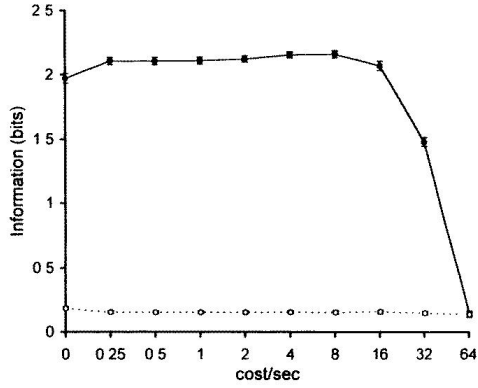
(c) $D^{interval}$



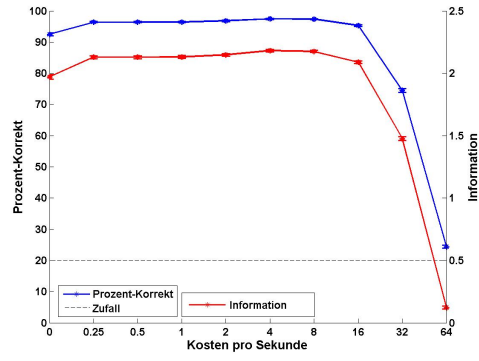
(d) D^{euklid}

Abbildung 4.1: Ratenunterscheidung ($CV=1.0$). Mittlere Güte des stimulusabhängigen Clusters mit Exponent $z = -2$ für Simulationen von homogenen Poisson-Prozessen. Balken stellen das Zweifache des jeweiligen Standardfehlers dar. 4.1a zeigt D^{spike} (durchgezogene Linie) und empirisches Zufallslevel (gepunktete Linie).

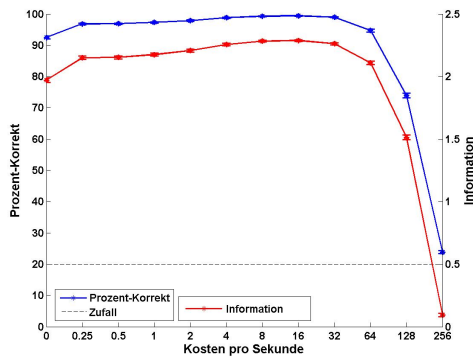
und wechselseitiger Information dargestellt. Die Ähnlichkeit des Verlaufs der wechselseitigen Information in den Abbildungen 4.1a und 4.1b bestätigt die Richtigkeit meines implementierten Algorithmus. Für D^{count} , also $D^{spike}[0]$, werden die Spiketrams am meisten in die richtigen Antwortklassen eingeordnet. Dies würde man auch erwarten, da in den simulierten Antworten die genauen Zeitpunkte der aufgetretenen Spikes keine Information über den Stimulus beinhalten. Für $0 < q \leq 16$ ist die stimulusabhängige Unterscheidung nur geringfügig schlechter. Dies indiziert, dass $D^{spike}[q]$ mit einem q zwischen 0 und 16 auf einer Mischung von Rate und Spikezeitpunkt basiert. Bei $q = 32$ fällt die wechselseitige Information auf das empirische Zufallslevel (siehe Abb. 4.1a). Das Verschieben eines Spikes ist nun meist teurer als das Löschen und wieder Einfügen zu einem anderen Zeitpunkt, auch innerhalb einer Stimulusklasse, da die Spikezeitpunkte in der vorliegenden Simulation ja zufällig sind. Somit sind die Abstände zu Antworten mit der kleinsten Rate auch immer öfter minimal und alle Antworten werden in die entsprechende Antwortklasse eingeordnet. Mittels Prozent-Korrekt kommt man zur qualitativ gleichen Bewertung des stimulusabhängigen Clusters. Das Zufallslevel hierfür liegt im Falle von fünf Stimuli bei 20%.



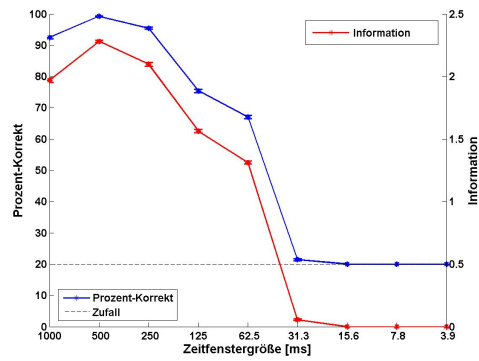
(a) D^{spike} modifiziert nach [VP97], dort Abb. 4E



(b) D^{spike}



(c) $D^{interval}$



(d) D^{euclid}

Abbildung 4.2: Ratenunterscheidung ($CV=0.125$). Mittlere Güte des stimulusabhängigen Clusters mit Exponent $z = -2$ für Simulationen von iterierten Poisson-Prozessen der Ordnung 64. Balken stellen das Zweifache des jeweiligen Standardfehlers dar. 4.2a zeigt D^{spike} (durchgezogene Linie) und empirisches Zufallslevel (gepunktete Linie).

Für $D^{interval}[q]$ kann man ein sehr ähnliches Verhalten feststellen. Zufriedenstellende stimulusabhängige Unterscheidung ist sogar bis $q = 32$ möglich. Bei einer Zeitfenstergröße von einer Sekunde ist die Unterscheidung mit $D^{euclid}[\tau]$ genauso gut wie mit D^{count} , was den Erwartungen entspricht, da der Abstand je gerade die absolute Differenz der Spikeanzahlen bis auf ein Vielfaches (was für den Algorithmus des stimulusabhängigen Clusters an sich unbedeutend ist) ist. Es sei daran erinnert, dass die Zeitfenstergröße τ gerade dem Kehrwert der Kosten q entspricht, also zum Beispiel $\tau = 62.5m$ entspricht $q = 16$. Für größere Werte von τ ist kein qualitativer Unterschied zu D^{spike} fest zu stellen. Insgesamt zeigt sich eine recht große Ähnlichkeit sowohl der verschiedenen Abstandsmaße als auch der beiden Maße zur Bestimmung der Güte des stimulusabhängigen Clusters bei Vorliegen eines homogenen Poisson-Prozesses.

Iterierter Poisson-Prozess

In Abbildung 4.2 sieht man die Ergebnisse für eine ähnliche Simulation, bei der die Aktionspotentiale jedoch etwas regulärer erzeugt wurden. So handelt es sich hier nicht um

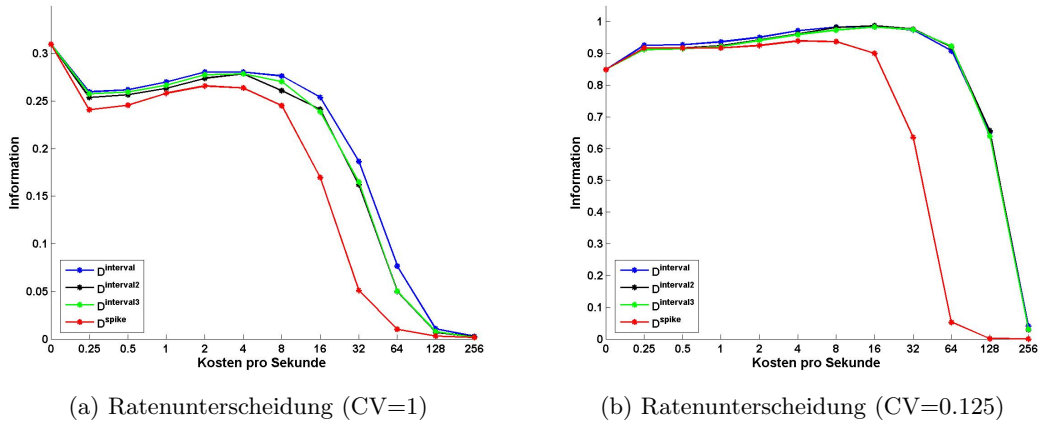


Abbildung 4.3: Mittlere Güte des stimulusabhängigen Clusters mit Exponent $z = -2$. Wechselseitige Information wurde mit $\log_2(5)$ normiert (vgl. Kapitel 3.2.2).

homogene Poisson-Prozesse, sondern um iterierte Poisson-Prozesse der Ordnung k mit $k=64$ (vgl. Kapitel 4.1). Der Variationskoeffizient für die interspikeintervalle ist daher nach Lemma 4.2.8 nun 0.125. Die Raten liegen wie in der vorherigen Simulation bei 2, 4, 6, 8 und 10 Spikes pro Sekunde.

Für D^{count} wird nahezu 100% der Antworten richtig zugeordnet, was daran liegt, dass die Anzahl der Spikes innerhalb einer Stimulusklasse weniger stark variieren. Bei allen drei Typen an Abstandsmaßen ist ein Anstieg zum zweitgrößten Parameter q bzw. zweitkleinsten Parameter τ festzustellen. Dies ist damit zu erklären, dass bei ähnlichem ersten Spikezeitpunkt auch die folgenden ähnlich sind und somit Spiketrains einer Stimulusklasse einen kleineren Abstand zueinander haben als wenn dieser nur über die Spikeanzahl bestimmt wird.

Zudem lassen sich ähnliche Unterschiede und Gemeinsamkeiten feststellen wie im vorherigen Fall. So führt $D^{interval}$ für größere Werte von q als im Fall von D^{spike} zu einer guten stimulusabhängigen Unterscheidung. Für D^{euclid} liegt Prozent-Korrekt für einen D^{spike} ähnlichen Bereich zeitlicher Auflösung über dem Zufallslevel. Die Verläufe sind jedoch nicht so übereinstimmend wie in der ersten Simulation. So nimmt die Güte des stimulusabhängigen Clusters für D^{euclid} bereits ab einem Zeitfenster von $500ms$ ab und nicht erst ab $\tau = 62.5ms$ wie es für D^{spike} bei diesem Wert entsprechenden q von 16 der Fall ist.

$D^{interval}[q]$

Die Abnahme der Güte des stimulusabhängigen Clusters für $D^{interval}$ bei einer höheren zeitlichen Auflösung als für D^{spike} könnte dadurch bedingt sein, dass die Länge der interspikeintervalle nicht so zufällig sind wie die Spikezeitpunkte (in dem Sinne, dass erstere exponentialverteilt sind, wohingegen das Vorhandensein eines Spikes in jedem gleichgroßen

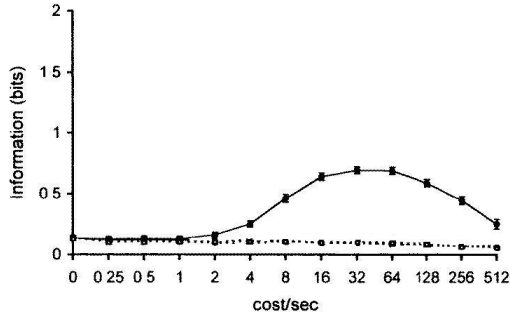
Intervall gleichwahrscheinlich ist). Der Erwartungswert der Interspikeintervalllängen unterscheidet sich zwischen den Stimulusklassen minimal um $1/8 - 1/10 = 0.025$ Sekunden. Mit einer entsprechenden zeitlichen Auflösung, was ein q von etwa $1/0.025 = 40$ bedeutet, sollte also bei Betrachtung der Interspikeintervalle eine Unterscheidung der Stimulusklassen über dem Zufallslevel erreicht werden, wie es die Ergebnisse auch zeigen. Abbildung 4.3b untermauert diese Vermutung. Hier bedingt der deutlich kleinere Variationskoeffizient eine quantitativ bessere Unterscheidung der neuronalen Antworten (vgl. Abbildung 4.3a,4.3b). Ausserdem trägt er dazu bei, dass sich die Stimulusklassen hinsichtlich der Interspikeintervalllängen eindeutiger auf einer feinen Zeitskala unterscheiden.

Mit Hinblick auf die in Kapitel 2.2.1 angesprochene Problematik der Bestimmung der Interspikeintervalle könnte es sein, dass das bessere Wirken von $D^{interval}$ auf einer feineren Zeitskala als D^{spike} durch die Implementierung bedingt sein könnte. Um diesen Verdacht zu überprüfen, werden zwei weitere Abstandsmaße betrachtet: Bei $D^{interval2}$ entfällt bei der Festsetzung nach $D^{interval}$ der stets gleiche letzte Aktionspotentialzeitpunkt, dem Reizende, und bei $D^{interval3}$ der erste immer gleiche Zeitpunkt, der Reizanfang. Die Ergebnisse der hiermit erfolgten Berechnungen sind in Abbildung 4.3 im Vergleich zum Standardverfahren $D^{interval}$ zu sehen. Dabei wird auf die Darstellung von Prozent-Korrekt verzichtet, da dies qualitativ das Gleiche zeigt. Offensichtlich kann die Information über die Zeit zwischen ersten Spike und Stimulusanfang oder über die Zeit zwischen letzten Spike und Stimulusende die Diskriminierung auf einer feineren Zeitskala verbessern (4.3a). Ist die Verteilung der Länge der Interspikeintervalle besonders regulär, fällt eine derartige Information aber nichts in Gewicht. Insgesamt lässt sich demnach der Unterschied zu $D^{interval}$ nicht hauptsächlich durch diesen Effekt erklären.

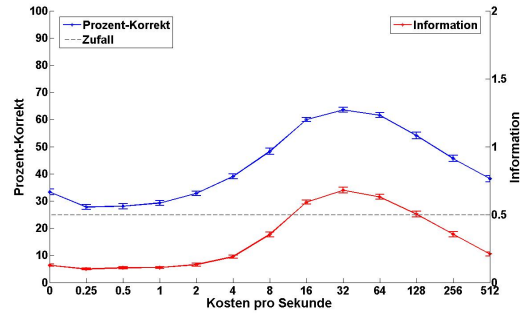
4.2.2 Zeitliche-Phasen-Unterscheidung

Erzeugung eines inhomogenen Poisson-Prozesses

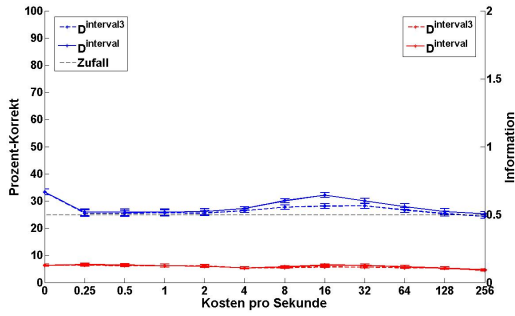
Im Folgenden wurden Spiketrains einer Länge von einer Sekunde erzeugt, so dass die Summe der Spikeanzahlen einen inhomogenen Poisson-Prozess mit lokaler Rate $\lambda(u) = R [1 + m \cos(2\pi f u + \phi)]$ bilden. Dabei bezeichnet R die mittlere Feuerrate, die auf 20 Spikes/Sekunde gesetzt wurde. Die Modulationfrequenz f wurde auf 4 Hertz eingestellt und die Modulationstiefe m auf 0.5. Durch den Parameter ϕ , der die Modulationsphase darstellt, wurden vier Stimulusklassen simuliert mit $\phi \in \{0, \pi/2, \pi, 3\pi/2\}$. Wie in den beiden vorangehenden Fällen wurden je zwanzig Präsentationen eines Stimulus simuliert und zur Ermittlung der Zuverlässigkeit der ermittelten Werte von Prozent-Korrekt und wechselseitiger Information die Simulation vierzig mal wiederholt.



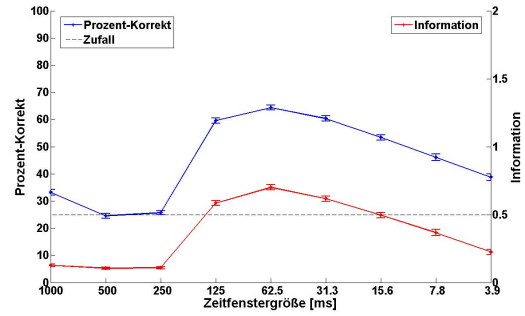
(a) D^{spike} modifiziert nach [VP97], dort Abb. 5
A



(b) D^{spike}



(c) Prozent-Korrekt (blaue Linien) und wechselseitige Information (rote Linien) für $D^{interval}$ und $D^{interval3}$



(d) D^{euklid}

Abbildung 4.4: Zeitliche-Phasen-Unterscheidung. Mittlere Güte des stimulusabhängigen Clusters mit Exponent $z = -2$ für Simulationen von inhomogenen Poisson-Prozessen. Balken stellen das Zweifache des jeweiligen Standardfehlers dar. 4.4a zeigt D^{spike} (durchgezogene Linie) und empirisches Zufallslevel (gepunktete Linie).

$$\begin{aligned}
 & \int_t^{t+r} 20 + 10 \cos(8\pi u + \phi) du \\
 &= 20(t+r) + 5 \frac{\sin(8\pi(t+r) + \phi)}{4\pi} - 20t - 5 \frac{\sin(8\pi t + \phi)}{4\pi} \\
 &= 20r + 5 \frac{\sin(8\pi(t+r) + \phi) - \sin(8\pi t + \phi)}{4\pi}
 \end{aligned} \tag{4.1}$$

Zur Simulation der Spiketrains wurden zunächst die Wahrscheinlichkeiten über Gleichung 4.1 und Definition 4.1.9 für das Auslösen eines Spikes in einem Zeitfenster der Größe $r = 0.00001$ (Sekunden) zwischen 0 und 1 (Sekunde) ermittelt. Ist diese Wahrscheinlichkeit dann größer oder gleich einer erzeugten gleichverteilten Zufallszahl zwischen 0 und 1, so wird in dem entsprechenden Intervall ein Spike generiert. Dies wird durch eine 1 in einem Vektor mit 100000 Einträgen an entsprechender Stelle notiert. Die Positionen der nicht-leeren Einträge je geteilt durch 100000 gibt dann auf Grund der Wahl sehr kleiner Zeitfenster eine gute Näherung für den Spikezeitpunkt in Sekunden an.

Vergleich der Abstandsmaße

Wie in Abbildung 4.4 zu sehen ist, zeigen D^{spike} und $D^{interval}$ ein unterschiedliches Verhalten. Für D^{spike} ist bei kleinen Werten von q kein stimulusabhängiges Clustern zu erkennen. Ab $q = 4$ kommt es zu einem kleinen Anstieg, bei $q = 32$ wird ein Maximum angenommen und ab $q = 512$ liegt keine gute stimulusabhängige Unterscheidung mehr vor. Dies ist damit zu erklären, dass die Stimulusklassen nicht durch die Anzahl der Spikes charakterisiert sind, sondern durch die Modulationsphase. Die Spikezeitpunkte der verschiedenen Stimulusklassen sind gerade um maximal $\frac{3\pi}{2}$ und minimal um $\frac{\pi}{2}$ von 8π pro Sekunde verschoben. Es bedarf also einer Zeitskala von etwa $\frac{3\pi}{2}/(8\pi s^{-1}) = 187.5ms$, um die neuronalen Antworten nach Stimuli überhaupt unterscheiden zu können. Dagegen erfordert eine gute Unterscheidbarkeit der neuronalen Antworten nach Stimuli mindestens eine Auflösung von $q = (\frac{\pi}{2}/8\pi)^{-1} = 16$. Dies spiegelt die Güte des stimulusabhängigen Clusters mit D^{spike} gut wieder. Mit $q = 16$ können die Antworten schon ganz gut auseinander gehalten werden, mit $q = 32$ wird der maximale Wert erreicht. Höher werdende zeitliche Präzision führt zu einer schlechter werdenden stimulusabhängigen Zuordnung.

Da das Abstandsmaß $D^{interval}$ dagegen auf den Interspikeintervallen beruht, deren Verteilungen kein Charakteristikum der Stimulusklassen bilden, ist hierfür keine gute Unterscheidung der Antworten zu verzeichnen (vgl. Abbildung 4.4c). Die kleine Erhebung im Verlauf von Prozent-Korrekt könnte dadurch begründet sein, dass das erste Interspikeintervall (zwischen Zeitpunkt 0 und erstem Spikezeitpunkt) und das letzte Interspikeintervall eine geringfügige Information über die zugehörige Stimulusklasse liefert, sofern die zeitliche Präzision entsprechend gewählt wird. Dies wird unterstützt durch Berechnungen mit $D^{interval2}$ und $D^{interval3}$, für die die Güte des stimulusabhängigen Clusters das Zufallslevel noch geringfügiger übersteigt.

D^{euklid} zeigt wieder einen ähnlichen Verlauf wie D^{spike} . Das Maximum liegt hier aber bei $\tau = 62.5ms$, also gerade der Zeitfenstergröße, um die sich die vier Stimulusklassen minimal unterscheiden.

5 Anwendung auf retinale Daten

5.1 Experiment und Datenselektion

Experiment

Eine Schildkrötenretina wurde mit einer eindimensional bewegten Fläche, wie sie in Abbildung 5.1 zu sehen ist, gereizt. Die Fläche wurde in fünf verschiedenen Geschwindigkeiten von 0mm/s , 0.625mm/s , 1.25mm/s , 1.875mm/s und 2.5mm/s bewegt und das sowohl nach links ($-x\text{mm/s}$) als auch nach rechts ($+x\text{mm/s}$), so dass es insgesamt neun unterschiedliche Stimuli hinsichtlich der Geschwindigkeit gibt. Diese wurden zufällig je zwischen 80- und 83-mal für je 500ms repräsentiert und dieser insgesamt 365000ms lange Zufallsvektor wurde nochmal sieben bis acht mal wiederholt.

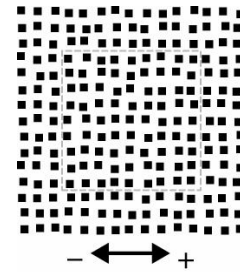


Abbildung 5.1: Reiz

Während dieser Stimulation wurden die elektrischen Aktivitäten der Ganglienzellen mittels eines Multielektrodenarrays bestehend aus 100 Elektroden abgeleitet. Dabei wurden an vielen Elektroden die Aktionspotentiale mehrerer Zellen aufgenommen. Basierend auf einer *k - means* Clustermethode wurde ein sogenanntes Spikesorting durchgeführt, bei dem diese neuronalen Signale an einer Elektrode unter Einbeziehung ihrer Form nach verschiedenen möglichen Zellen aufgeteilt wurden. Nach diesem Spikesorting wurden dann je Ganglienzelle (d.h. Prototyp an einer Elektrode) die Zeitpunkte der jeweiligen Spikes bestimmt. Für eine ausführlichere Beschreibung des Experiments sei auf [Th07] verwiesen.

Der oben geschilderte Versuch wurde dreimal durchgeführt, wobei die letzten beiden Datensätze von der gleichen Retina erhoben wurden. Die Datensätze werden im folgenden mit Daten02, Daten10 und Daten14 bezeichnet. In Daten02 und Daten10 gibt es je acht Wiederholungen, in Daten14 nur sieben. Nach dem Spikesorting umfasst der Datensatz Daten02 nach dem Spikesorting 87 Zellen, Daten10 und Daten14 je 107 Zellen.

(Nicht-)Richtungssensitive Ganglienzellen

In Abbildung 5.2 sind vier Beispiele für die mittlere Spikeanzahl pro Stimulus, sogenannte Tuningkurven, zu sehen. Wie man dieser Abbildung entnehmen kann, unterscheiden sich

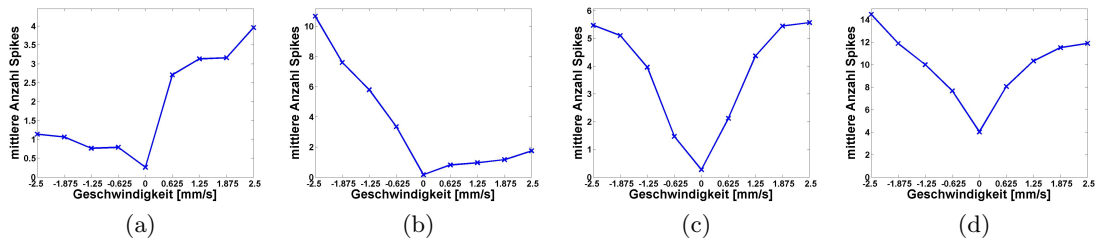


Abbildung 5.2: Tuningkurven von vier verschiedenen Zellen aus den drei Experimenten. Die Bewegungsrichtung des Stimulus wird dabei durch das Vorzeichen der Geschwindigkeit angegeben.

die Ganglienzellen in ihrem Antwortverhalten. So existieren Neurone, die für eine Richtung sensibler sind als für die andere (5.2a und 5.2b). Andere reagieren aber unabhängig von der Richtung mit etwa gleicher Spikeanzahl, wenn die Fläche betragsmäßig um die identische Geschwindigkeit bewegt wurde (5.2c und 5.2d). Hierbei handelt es sich um den größten Anteil an Zellen. So wurde in [AMK95] gezeigt, dass 30 – 40% aller Ganglienzellen selektiv für eine Richtung sind.

Stabilität des Antwortverhaltens

Eine wichtige Annahme des anzuwendenden Verfahrens ist, dass ein Neuron auf jede Präsentation des gleichen Stimulus sehr ähnlich antwortet. Daher sollte zuvor ausgeschlossen werden, dass die im Versuch gewonnenen Antworten zu stark variieren oder sich systematisch im Lauf des Experiments ändern (siehe Abbildung 5.3). Mögliche Ursachen hierfür könnten das Absterben einer Zelle oder Adaptation sein. Ebenfalls kann es passieren, dass das Multielektrodenarray ganz leicht verrutscht und somit an einer Elektrode eine andere Zelle näher liegt, deren Signale dann aufgezeichnet werden.

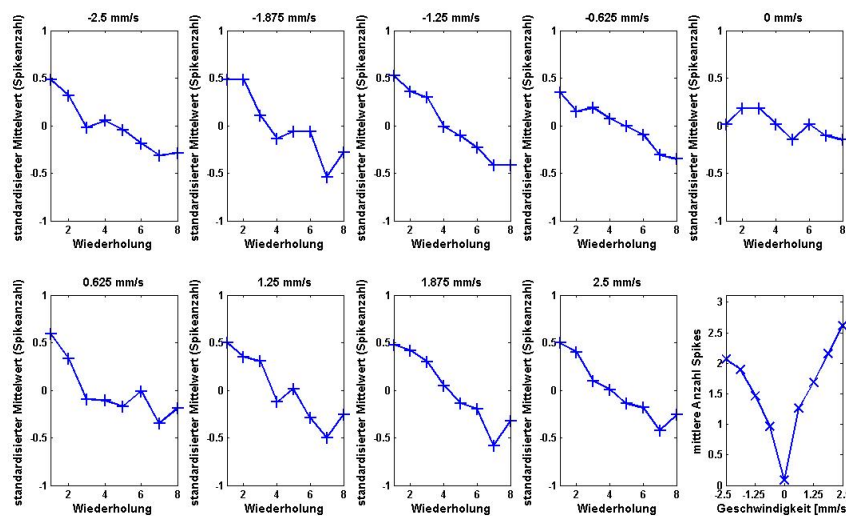


Abbildung 5.3: Beispiel für nicht-stabiles Antwortverhalten einer Zelle

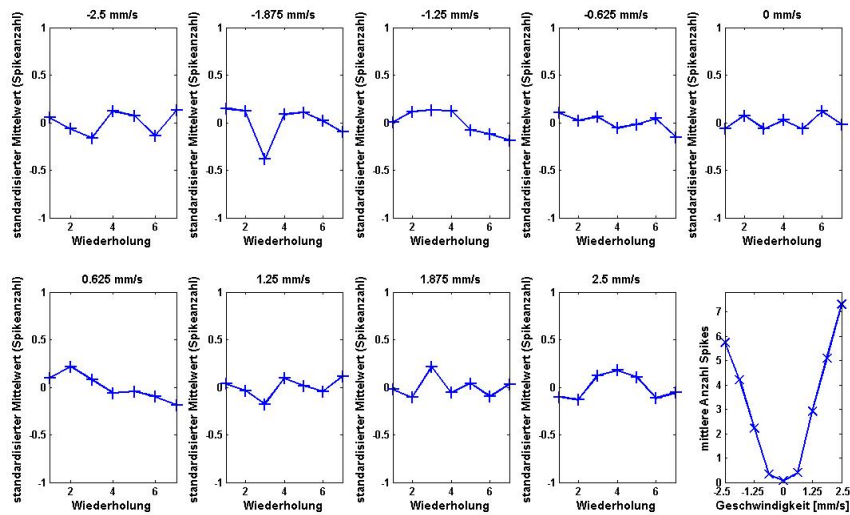


Abbildung 5.4: Beispiel für stabiles Antwortverhalten einer Zelle

Mit dem Ziel diejenigen Zellen mit der stabilsten Ableitqualität zu selektieren, werden die standardisierten Mittelwerte der Spikeanzahlen einer Zelle pro Wiederholung für jeden der neun Stimuli abgetragen. Abbildung 5.3 zeigt beispielhaft eine Zelle, deren mittlere Spikeanzahlen über die Wiederholungen abnimmt. Dagegen ist in Abbildung 5.4 ein vergleichsweise stationäres Antwortverhalten zu sehen.

Auf diese Weise wurden aus Daten02 und Daten10 je zehn Zellen und aus Daten14 sechs Zellen für das weitere Vorgehen als verwendbar bestimmt (siehe auch Anhang).

5.2 Ergebnisse

5.2.1 Dekodierung der Geschwindigkeit

Die Antworten der Ganglienzellen werden zunächst in neun Stimulusklassen (vgl. Kapitel 3.1) eingeteilt, in denen sich jeweils die Spiketrains befinden, die durch denselben Stimulus ausgelöst wurden. Dabei bildet je eine Geschwindigkeit mit der zugehörigen Richtung einen Stimulus. Um zu gewährleisten, dass für jeden Stimulus gleich viele Antworten in die Analyse einbezogen werden, werden innerhalb jeder der sieben bis acht Subsequenzen jeweils nur die ersten achtzig Präsentationen verwendet. So stehen einem pro Zelle für jeden Stimulus 640 beziehungsweise 560 neuronale Antworten zur Verfügung. Es werden für die Spiketrains der ausgewählten je zehn Zellen von Daten02 und Daten10 und sechs Zellen von Daten14 die Distanzmatrizen berechnet. Wie in Kapitel 4.2 beschrieben werden diese anschließend mit den Clusterexponenten -8 , -2 , 1 , 2 und 8 geclustert und die Güte des stimulusabhängigen Clusters bestimmt. Dies geschieht für D^{spike} und $D^{interval}$ also mit folgenden zwölf Werten von q : 0 , 0.25 , 0.5 , 1 , 2 , 4 , 8 , 16 , 32 , 64 , 128 und $256 s^{-1}$. Für den Parameter τ des Abstandsmaßes D^{euklid} werden die gerundeten Kehrwerte von q zwischen $2s^{-1}$ und $256s^{-1}$ gewählt.

Bias

Bevor die Güte des stimulusabhängigen Clusters für die verschiedenen Zellen betrachtet wird, sei an den Bias der wechselseitigen Information (siehe Kapitel 3.2.2) erinnert. Der Bias wird zunächst in Anlehnung an Victor et al. ([VP96]) empirisch abgeschätzt. Dabei werden die Spiketrains zufällig den Stimuli zugeordnet, also die Distanzmatrizen permutiert und anschließend geclustert. Die über zehn solcher Durchführungen gemittelte Güte des stimulusabhängigen Clusters gibt dann an, um wieviel der wahre Wert der wechselseitigen Information im Falle von zufälliger Zuordnung überschätzt wird. Diese Vorgehensweise wird motiviert durch die Tatsache, dass beim Clustern die Antworten stark interagieren, die Clustermatrix N sich also nicht als Kontingenztafel unabhängiger Versuchswiederholungen interpretieren lässt, was aber bei analytischen Abschätzungen des Bias vorausgesetzt wird (zum Beispiel [C69],[PT96]). Zudem könnte der Bias abhängig sein von der Art des Distanzmaßes oder dem Parameter q beziehungsweise τ .

Abbildung 5.5b zeigt die über zehn Zellen von Daten10 gemittelte Güte des Clusters bei Vorliegen stimulusunabhängiger Antworten für D^{spike} . Sie stellt ein empirisches Zufallslevel dar. Da das theoretische Zufallslevel bei 0 liegt, gibt diese Güte somit zugleich einen empirisch abgeschätzten Bias an. Für positive Clusterexponenten ist keine Abhängigkeit von q zu erkennen. Dagegen scheint für negative Exponenten der Bias für $q = 0$ geringfügig kleiner zu sein als für Werte größer als Null. Des Weiteren ist ein analytisch bestimmter Bias für stimulusunabhängiges Clustern, also stochastisch unabhängige gleichverteilte Zufallsvariablen X und Y , nach Gleichung 3.2 abgetragen. Wird der analytische Bias nur

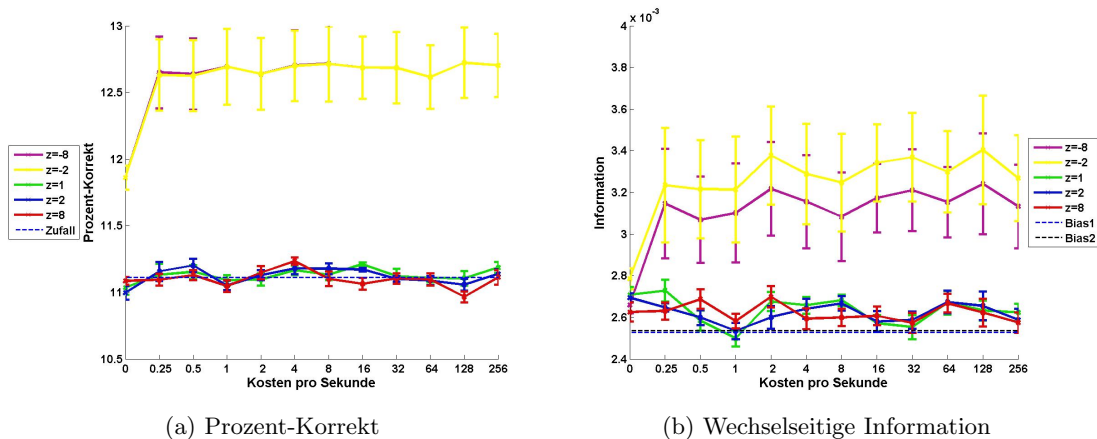


Abbildung 5.5: Güte des Clusters bei Vorliegen stimulusunabhängiger Antworten und analytische Abschätzung des entsprechenden Bias. Mittelung über zehn Zellen von Daten10 für D^{spike} . Balken zeigen das Zweifache des Standardfehlers an. Wechselseitige Information mit $\log_2 9$ normiert.

über den ersten Summand berechnet (Bias1, vgl. Beispiel 3.2.19 a)), ist er dem empirischen (für positive Clusterexponenten) bereits ähnlich. Durch eine Abschätzung des Bias über beide Summanden (Bias2) wird die Annäherung an den empirisch ermittelten kaum verbessert.

Bei der Bestimmung des empirischen Bias der wechselseitigen Information wurde die Güte des stimulusabhängigen Clusters mittels Prozent-Korrekt gleichzeitig mitbestimmt. Die Ergebnisse für positive Clusterexponenten entsprechen der Erwartungstreue des Mittelwertschätzers. Die Tatsache, dass die Werte für Prozent-Korrekt wie bei der wechselseitigen Information für negative Clusterexponenten höher sind, lässt vermuten, dass dies auf den Algorithmus des stimulusabhängigen Clusters oder die Daten selbst zurückzuführen ist.

Die Ergebnisse für $D^{interval}$ und D^{euclid} sind nicht gezeigt, diese sind aber sehr ähnlich. Der empirisch geschätzte Bias ist für die vorliegende Daten demnach unabhängig vom Abstandsmaß. Ebenfalls nicht dargestellt sind die Ergebnisse der beiden anderen Datensätze, da auch sie kaum Unterschiede aufweisen.

Vergleich der Abstandsmaße

In Abbildung 5.6 werden die drei Typen von Abstandsmaßen anhand des zweiten Experiments (Daten10) für die Gütemaße Prozent-Korrekt (Abb. 5.6a) und wechselseitiger Information (Abb. 5.6b) verglichen. Neben der über die zehn verwendeten Zellen gemittelte Güte des stimulusabhängigen Clusters ist das Zweifache des Standardfehlers als Balken abgetragen, um so einen Eindruck für die Güte der Mittelwerte zu bekommen. Bei einer Anzahl von neun Stimulusklassen würde ein zufälliges Clustern bedeuten, dass $1/9$, also 11%, der neuronalen Antworten richtig zugeordnet würden. Das Zufallslevel von 0 für die wechselseitige Information ist nur geringfügig verfälscht. Diese Verfälschung wird durch die sich überlagernden gestrichelten Linien dargestellt. Hierbei handelt es sich je

um das gemittelte empirische Zufallslevel für ein Abstandsmaß und somit eine empirische Abschätzungen des Bias (Abb. 5.6b).

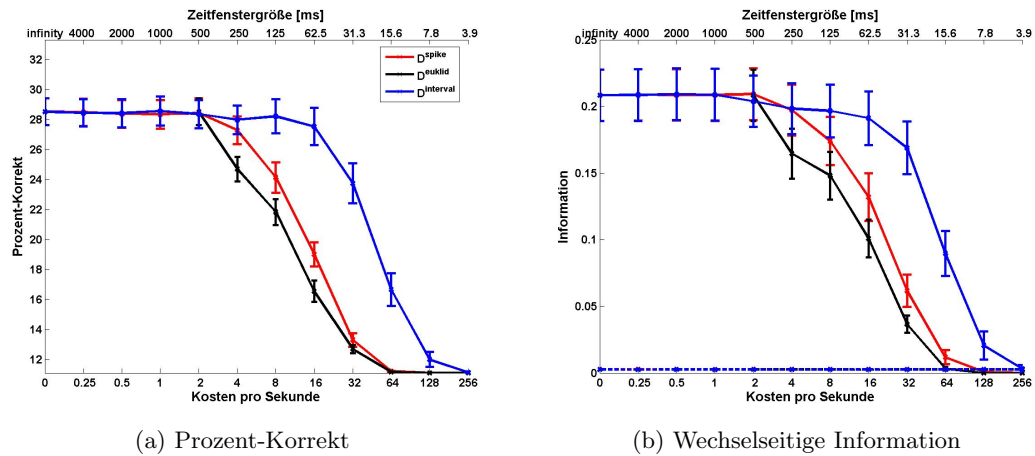
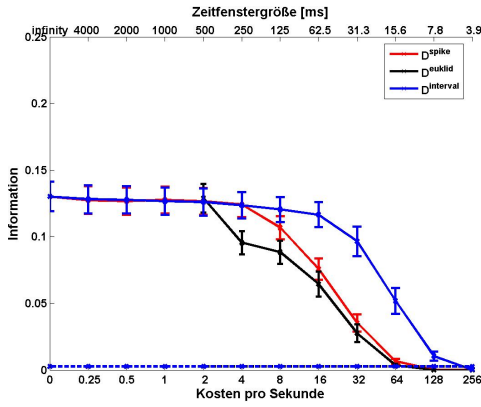


Abbildung 5.6: Mittlere Güte des stimulusabhängigen Clusters mit $z = 1$ für die zehn Zellen von Daten10. Balken zeigen das Zweifache des Standardfehlers an. 5.6a Zufallslevel von Prozent-Korrekt liegt bei 11%. 5.6b Mit $\log_2 9$ normierte wechselseitige Information (vgl. 3.2.2). Gestrichelte Linie zeigt empirisches Zufallslevel an.

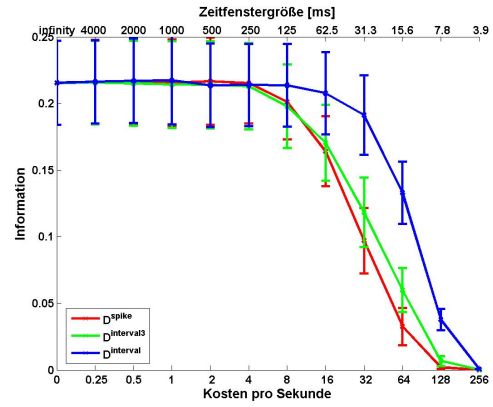
Wie man der Abbildung 5.6 entnehmen kann, bewerten Prozent-Korrekt und die wechselseitige Information die Güte des stimulusabhängigen Clusters mit einem Exponenten von $z = 1$ qualitativ gleichwertig. Für einen großen Bereich von q führen D^{spike} und $D^{interval}$ zu einer gleich guten Unterscheidung der Antworten, die deutlich über dem Zufallsleveln liegt. Stimulusabhängiges Clusters wird ab $q = 4$ für D^{spike} schlechter, wohingegen bei $D^{interval}$ die Werte von Prozent-Korrekt und wechselseitiger Information erst ab $q = 16$ fallen und mit $q = 64$ noch klar über dem jeweiligen Zufallslevel liegen. D^{euklid} ist qualitativ D^{spike} sehr ähnlich, es werden allerdings etwas weniger Antworten richtig zugeordnet. Mit diesen beiden Abstandsmaßen wird bis zu $\tau = 62.5ms$ beziehungsweise $q = 16$ Werte ermittelt, die noch merklich über dem Zufallslevel liegen.

Die beiden weiteren Datensätze weisen große Gemeinsamkeiten mit Daten10 auf. Für Daten10 und Daten14 (vgl. Abbildung 5.7b) ist nicht nur der qualitative Verlauf der wechselseitigen Information sehr ähnlich, sondern auch der quantitative. Dagegen werden für Daten02 vergleichsweise immer etwas weniger Antworten im Mittel stimulusabhängig eingeteilt (Abb. 5.7a).

In Abbildung 5.7b ist das Ergebnis des stimulusabhängigen Clusters mit $D^{interval3}$ (vgl. vorhergehende Kapitel) im Vergleich zu D^{spike} und $D^{interval}$ für Daten14 dargestellt. Da auch hier die Kurven von Prozent-Korrekt und wechselseitiger Information den gleichen Verlauf zeigen, ist nur die Information abgebildet. Der Verlauf von $D^{interval3}$ ist dem von D^{spike} ähnlicher als dem von $D^{interval}$. Auf Grund der Ähnlichkeit der bisherigen Ergebnisse ist davon auszugehen, dass die aufgrund der begrenzten zur Verfügung stehenden Rechenzeit nicht durchgeführten Berechnungen für Daten02 und Daten10 die gleiche Wirkung von $D^{interval3}$ zeigen würden. Offensichtlich ermöglicht die Information über den zeitlichen Abstand zwischen Reizbeginn und erstem Spike und dem zwischen letzten Spike



(a) Daten02



(b) Daten14

Abbildung 5.7: Mittlere Güte des stimulusabhängigen Clusters mit $z = 1$ für die zehn Zellen von Daten02 (5.7a) und die zehn Zellen von Daten14 (5.7b). Balken zeigen je den zweifachen Standardfehler an. Wechselseitige Information ist mit $\log_2 9$ normiert.

und Reizende eine bessere Unterscheidung der Spiketrains in Form von Interspikeintervall-Sequenzen bei einer zeitlichen Auflösung von $1/16 = 62.5$ bis $1/64 = 15.6$ Millisekunden im Gegensatz zu Befunden aus den Simulationen (vgl. Kapitel 4.2).

Die Abbildungen 5.6 und 5.7 zeigen also, dass die Antworten der Ganglienzellen für einen großen Bereich zeitlicher Auflösung für D^{spike} und $D^{interval}$ in etwa gleich gut unterschieden werden können. Da aber nicht wie in den simulierten Fällen davon ausgegangen werden kann, dass allen Zellen das gleiche Antwortverhalten zugrundeliegt, sollte dies etwas besser charakterisiert werden. Dazu erfolgt die Bestimmung eines größten Parameters q beziehungsweise kleinsten τ , für den die Güte des stimulusabhängigen Clusters in etwa genauso gut ist, wie die Unterscheidung auf einer größeren Zeitskala. Diese Stelle des Abbruchs wird beschrieben durch denjenigen Parameter, bei dem die Güte gemessen in Prozent-Korrekt oder wechselseitiger Information zum folgenden Parameter um mehr als zehn Prozent absinkt. Bei der Bestimmung wird für die wechselseitige Information jeweils der erste Summand nach Gleichung 3.2 abgezogen. Damit soll erreicht werden, dass die jeweiligen Verhältnisse der Werte möglichst korrekt bestimmt werden (siehe auch Diskussion).

Abbildung 5.8 zeigt die nach diesem Kriterium ermittelte Verteilung der 25 Neurone. Um die Statistik nicht zu verfälschen, wurde eine Zelle aus Daten10 entnommen, die auch aus dem letzten Datensatz, der ja auf der gleichen Retina beruht, zur Untersuchung ausgewählt wurde. Insgesamt erhält man für die Gütemaße Prozent-Korrekt und die wechselseitige Information wieder qualitativ ähnliche Ergebnisse. Am meisten unterschieden sie sich für $D^{euclid}[\tau]$. So ist die Stelle des Abbruchs für die Information stets bei einem Zeitfenster, also einer zeitlichen Auflösung, von $\tau = 500ms$ zu verzeichnen (Abbildung 5.8b). Dagegen nimmt die Anzahl an korrekten Zuordnungen bei einigen Zellen für Prozent-Korrekt erst nach $\tau = 125ms$ deutlich ab. Für D^{spike} und $D^{interval}$ spiegeln die Histogramme die Unterschiede wie bereits in Abbildung 5.6 festgestellt wieder. So ist eine starke Verschlechterung

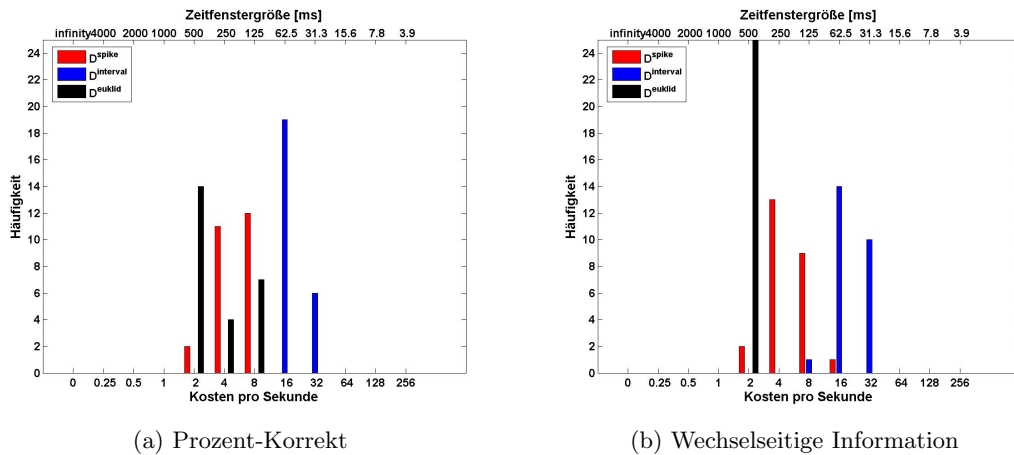


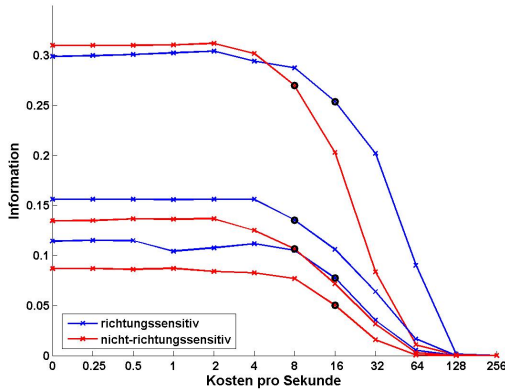
Abbildung 5.8: Anzahl der 25 Zellen, bei denen die Güte des stimulusabhängigen Clusters für die entsprechende Familie des Abstandsmaßes beim Parameter q beziehungsweise τ zum folgenden Wert um mehr als zehn Prozent abnimmt.

der Güte des stimulusabhängigen Clusters bei D^{spike} meistens bereits bei einem q von 8 oder 16, also bei einer zeitlichen Auflösung von $125ms$ oder $62.5ms$, zu verzeichnen, wohingegen dies bei $D^{interval}$ erst bei 32 oder 64 zu beobachten ist.

Vergleich der nicht- und richtungssensitiven Zellen

Die betrachteten Ganglienzellen weisen, wie bereits erwähnt, eine unterschiedliche Sensitivität für die Richtung eines Stimulus auf. Es stellt sich daher die Frage, ob auch ein Unterschied nach dem 10%-igem Abfallkriterium zu verzeichnen ist. In Tabelle 5.9b sind daher der prozentuale Anteil an danach getrennten Neuronen, für die die Stelle des Abbruchs bei den am größten ermittelten Parameter q beziehungsweise am kleinsten Parameter τ verzeichnet wurde, zusammengefasst. Dabei wurden von den 25 Ganglienzellen auf Grund ihrer Tuningkurven fünf Zellen als deutlich richtungssensitiv (zwei von Daten02 und drei von Daten14) und zwölf als deutlich nicht-richtungssensitiv eingeteilt (vier von Daten02, sechs von Daten10, zwei von Daten14). Die restlichen acht Zellen konnten nicht eindeutig zugeordnet werden (vier von Daten02, drei von Daten10, eine von Daten14, siehe Anhang).

Abbildung 5.9a zeigt am Beispiel von sechs Zellen aus den drei Datensätzen, dass die Bewertung der Güte des stimuluabhängigen Clusters für D^{spike} mit der wechselseitigen Information für richtungssensitive und nicht-richtungssensitive Zellen sowohl quantitativ als auch qualitativ sehr ähnlich verlaufen kann. Doch Tabelle 5.9b deutet darauf hin, dass die Unterscheidung der Antworten nach Geschwindigkeit für richtungssensitive Zellen oft auf einer feineren Zeitskala als für nicht-richtungssensitive Zellen deutlich schlechter wird. So ist der prozentuale Anteil an Zellen mit maximaler Stelle des Abbruchs sowohl für Prozent-Korrekt als auch wechselseitige Information stets etwas höher.



(a) Beispiele für nicht-/richtungssensitive Zellen

		D^{spike} $q \geq 8$	$D^{interval}$ $q = 32$	D^{euklid} $\tau = 125$
R	Info	60%	80%	0%
	Pro	60%	40%	40%
NR	Info	25%	25%	0%
	Pro	41.6%	16.6%	16.6%
S	Info	50%	37.5%	0%
	Pro	50%	25%	50%

(b) Prozentualer Anteil an Zellen mit maximaler Stelle des Abbruchs.

Abbildung 5.9: Vergleich zwischen richtungssensitiven und nicht-richtungssensitiven Zellen. 5.9a Güte des stimulusabhängigen Clusters einiger Zellen mit D^{spike} . Schwarzer Kreis markiert die Stelle nach einem erstmalig 10%-igen Abfall. 5.9b Prozentualer Anteil an allen 25 Zellen mit maximaler Stelle des Abbruchs nach Prozent-Korrekt (Pro) und wechselseitiger Information (Info) getrennt nach richtungssensitiven (R), nicht-richtungssensitiven (NR) und sonstigen (S) Zellen.

Auswirkung des Clusterexponenten im Vergleich mit künstlichen Daten

Die Güte des stimuluabhängigen Clusters der betrachteten retinalen Daten erinnert in der Abhängigkeit von q beziehungsweise τ an die mittels eines homogenen Poisson-Prozesses erzeugten künstlichen Daten in Kapitel 4.2.1. Betrachtet man jedoch die Histogramme der Spikezeitpunkte so scheinen diese nicht zufällig zu sein. So kann man eine von der Geschwindigkeit abhängig unterschiedlich stark ausgeprägte und verschobene Häufung der Spikezeitpunkte innerhalb der etwa ersten 150ms feststellen. Dies motiviert die Frage, ob bei gleichbleibender mittlerer Spikeanzahl eine derartige unterschiedliche Verteilung der Spikezeitpunkte nicht auch Unterschiede in der Diskriminierung der neuronalen Daten mit Hilfe der drei Typen von Abstandsmaßen nach sich zieht.

Dazu wurden zunächst nicht-richtungssensitive Zellantworten simuliert, so dass die Summe der Spikeanzahlen einen inhomogenen Poisson-Prozess mit lokaler Rate

$$\lambda(u) = h \cdot \frac{\alpha^\beta}{\Gamma(\beta)} u^{\beta-1} \exp(-\alpha u) + v \cdot u \quad (5.1)$$

bilden. Dabei wurde stets $\beta = 100$ gesetzt und für die Erzeugung der neun verschiedenen Stimulusklassen die verbleibenden Parameter nach Tabelle 5.1 gewählt.

Geschwindigkeit	-2.5	-1.875	-1.25	-0.625	0	0.625	1.25	1.875	2.5
α	6	7	8	9	10	9	8	7	6
h	1.8	1.4	1	0.6	0.2	0.6	1	1.4	1.8
v	8	6	4	2	0	2	4	6	8

Tabelle 5.1: Parameter zur Simulation von neuronalen Antworten auf neun Geschwindigkeiten (in mm/s) nach Gleichung 5.1.

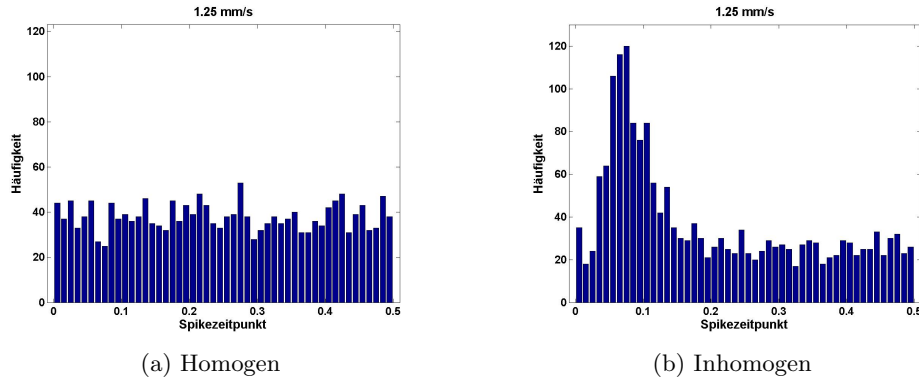


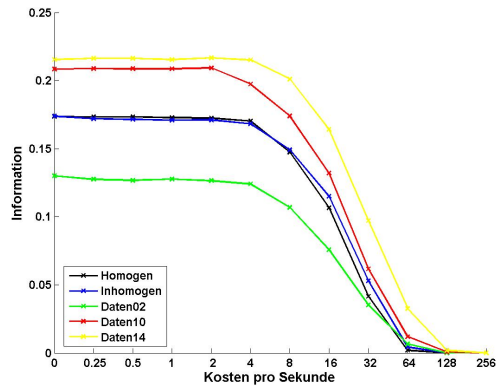
Abbildung 5.10: Histogramm der Spikezeitpunkte je einer simulierten Zelle als Reaktion auf die Geschwindigkeit 1.25mm/s .

Diese Wahl bewirkt, dass die Anzahl der Aktionspotentiale und der Peak, wie in Abbildung 5.10b zu sehen, mit betragsmäßig größerer Geschwindigkeit zunehmen. Ausserdem liegt die Maximumsstelle $((\alpha - 1)/\beta)$ der Spikeanzahlen pro Spikezeitpunkt für einen Betrag von 2.5mm/s , 1.875mm/s , 1.25mm/s , 0.625mm/s und 0mm/s bei 50ms , 60ms , 70ms , 80ms und 90ms vor. Wie bei den vorhergehenden Simulationen wurden Antworten von vierzig Zellen simuliert, denen jeder Stimulus wie bei den vorliegenden retinalen Daten 640 mal präsentiert wurden. Der entsprechende homogene Poisson-Prozess wurde mit den über die vierzig Zellen gemittelten Spikeraten von 11.6, 8.8, 6, 3.2, 0.4, 3.2, 6, 8.8 und 11.6 Spikes pro Sekunde für die neun Stimuli erzeugt.

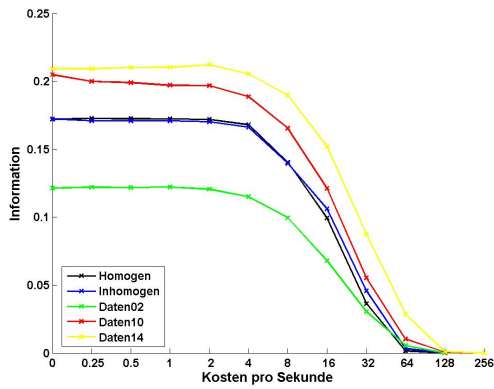
Im Folgenden werden die über die entsprechenden (simulierten) Zellen gemittelten Ergebnisse des stimulusabhängigen Clusters nur für die wechselseitige Information dargestellt (siehe Abbildung 5.11), da sie denen mit Prozent-Korrekt sehr ähneln. Ausserdem wird nur D^{spike} betrachtet, da für die beiden weiteren Distanzmaße qualitativ die gleichen Unterschiede und Gemeinsamkeiten zwischen den Daten festzustellen sind und die Wirkung der Clusterexponenten unabhängig vom Abstandsmaß war. Das empirische Zufallslevel ist auch nicht abgetragen, da es für alle (für die künstlichen Daten wurde dies nur für einen kleinen Stichprobenumfang geprüft) in etwa gleich war und die Güte des stimulusabhängigen Clusters stets deutlich über dem empirischen Zufallslevel liegt. Aus Gründen der Übersichtlichkeit wurde auf die Darstellung der Standardfehler verzichtet.

Wie man Abbildung 5.11a entnehmen kann, führt die Häufung der Spikezeitpunkte (Inhomogen) zu einer etwas besseren Zuordnung als bei zufälligen Spikezeitpunkten (Homogen), sofern bei der Abstandsbestimmung die zeitliche Präzision der Spikezeitpunkte zwischen $1/16 = 62.5\text{ms}$ und $1/64 = 15.6\text{ms}$ betrachtet wird. Dies entspricht der Tatsache, dass die Stelle der Peaks zwischen den Stimulusklassen maximal um 40ms und minimal um 10ms verschoben sind und eine dementsprechend gewählte feine Zeitskala die Antworten besser unterscheidbar macht, die Spiketrains aber ansonsten keine zeitlichen Strukturen enthalten, die zur Unterscheidung beitragen könnten.

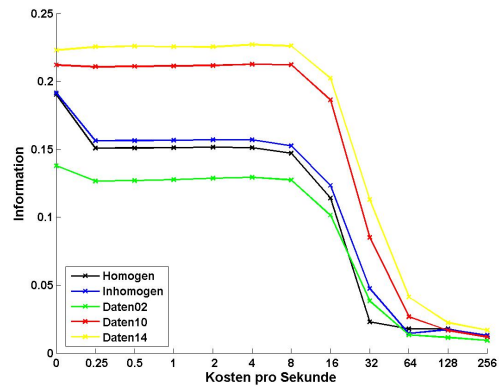
Verglichen mit den künstlichen Daten fällt für die retinalen Daten die Güte des stimulusabhängigen Clusters bei der etwa gleichen zeitlichen Auflösung ab. Auf einer sehr feinen



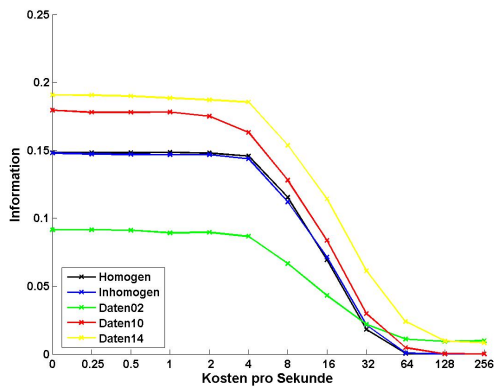
(a) $z=1$



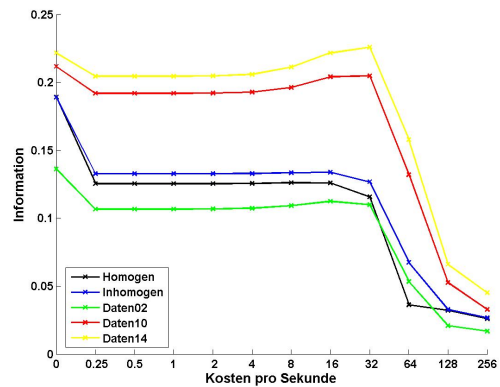
(b) $z=2$



(c) $z=-2$



(d) $z=8$



(e) $z=-8$

Abbildung 5.11: Güte des stimulusabhängigen Clusters mit D^{spike} der künstlichen und retinalen Daten, bei jeweiliger Mittelung über die entsprechenden (simulierten) Zellen, im Vergleich für verschiedene Clusterexponenten. Wechselseitige Information ist mit $\log_2 9$ normiert.

Zeitskala, $q = 64$, kann für alle neuronalen Daten noch eine ganz geringfügige systematisch stimulusabhängige Zuordnung verzeichnet werden. Der Wert der wechselseitigen Information liegt dabei knapp über denen der simulierten homogenen Daten, auch wenn, wie im Falle von Daten02, die Werte bei einem kleineren Preis fürs Verschieben eines Spikes deutlich unter diesen liegen.

Für einen Clusterexponenten von $z = 2$ lässt sich feststellen, dass die Güte des stimulusabhängigen Clusters insgesamt ein klein wenig schlechter als im Fall von $z = 1$ geworden ist. Werden die Abstände zu den „entferntliegenden“ Spiketrains noch stärker betont, wird die systematisch stimulusabhängige Einteilung der neuronalen Antworten noch schlechter (siehe Abbildung 5.11d). Der von q abhängige Verlauf der wechselseitigen Information ist dabei für die beiden künstlichen Datensätze nahezu gleich und für alle Daten insgesamt dem mit $z = 1$ sehr ähnlich.

Werden dagegen die kürzeren Abstände stärker gewichtet, liegt der maximale Wert der wechselseitigen Information über dem beim Clustern mit $z = 1$ ermittelten (5.11c). Zudem zeigt sich eine unterschiedliche Wirkung eines negativen Clusterexponenten auf die retinalen und künstlichen Daten. So ist für die mittels eines homogenen und die mittels eines inhomogenen Poisson-Prozesses erzeugten Zellantworten die beste Unterscheidung mit D^{count} mit $z = -2$ zu verzeichnen. Dies deutet sich bei Daten02 nur an und ist bei den Neuronen der anderen Retina nicht festzustellen. Gemein ist wiederum allen Daten, dass sich die rechte Grenze des Intervalls, aus dem der Parameter q gewählt werden kann, um eine möglichst und in etwa gleich gute stimulusabhängige Diskriminierung zu erreichen, auf den Wert 8 bis 16 vergrößert.

Dieser Wert erhöht sich für $z = -8$ sogar auf $q = 32$ (siehe Abbildung 5.11e). Die Überlegenheit von D^{count} gegenüber $D^{spike}[q]$ mit einem $q > 0$ zeichnet sich für die künstlichen Daten noch stärker ab. Daneben wird der Unterschied der beiden Datensätze hinsichtlich ihrer zeitlichen Struktur deutlicher, da die Güte des stimulusabhängigen Clusters auf einer feinen Zeitskala von $q = 64$ ausgeprägter differiert als es noch für $z = 1$ bei $q = 32$ für D^{spike} der Fall war, wohingegen für $D^{interval}$ ein derartiger Effekt nicht zu beobachten ist. Bei einer so starken Gewichtung der kleinen Abstände mit $z = -8$, also einer Betonung der sich am stärksten ähnelnden neuronalen Antworten, wird die Güte des stimulusabhängigen Clusters auf Basis der Spikeanzahlen auch für Daten10 und Daten14 vergleichsweise besser. Die wechselseitige Information ist aber bei $q = 32$, wo sich ein kleines lokales Maximum abzeichnet, in etwa gleich groß.

5.2.2 Dekodierung der Geschwindigkeitsübergänge

Bisher wurde mit Hilfe Metrik-basierter Verfahren untersucht, inwieweit die neuronalen Antworten unterscheidbar sind, sofern man die Stimuluseigenschaften absolute Geschwindigkeit in Kombination mit der Richtung betrachtete. Jedoch ist nicht klar, was für Informationen über den Reiz durch die Ganglienzellen tatsächlich ans Gehirn weitergeleitet

werden. So könnte es zum Beispiel auch sein, dass eine Stimulusänderung von einer Geschwindigkeit von 0mm/s auf eine Geschwindigkeit von -2.5mm/s in einem Spiketrain verschlüsselt wird. Daher wird im Folgenden stimulusabhängiges Clustern angewendet auf neuronale Antworten, die nach diesen Geschwindigkeitsübergängen in Stimulusklassen eingeteilt werden. Unter Berücksichtigung aller Möglichkeiten, also den Permutationen der Geschwindigkeiten ohne Wiederholung, erhält man hiervon $9 \cdot 8 = 72$ Stück. Dabei wurde jeder der zugehörigen Stimuli mindestens 10-mal in jeder der acht (Daten02 und Daten10) beziehungsweise sieben Wiederholungen (Daten14) präsentiert, so dass im Folgenden wieder die je ersten 10 entsprechenden Spiketrains betrachtet werden. Insgesamt erhält man so 80 beziehungsweise 70 Antworten pro Stimulus. Dabei werden die gleiche 26 Zellen wie im vorhergehenden Kapitel verwendet.

Bias

Bevor die eigentlichen Ergebnisse dargestellt werden, sei wieder kurz auf den Bias verwiesen. Wie im vorhergehenden Kapitel wurde das empirische Zufallslevel, also eine empirische Abschätzung des Bias bei Vorliegen stimulusunabhängiger Antworten, bestimmt. Die empirische Abschätzung des Bias auf Basis der Distanzmatrizen wurde erst überlegt, nachdem die Matrizen für die Geschwindigkeitsübergänge schon gelöscht waren und die Zeit zur erneuten Berechnung der Distanzmatrizen nicht mehr ausreichte. Daher musste für das Permutieren der Matrixeinträge und das stimulusabhängige Clustern in 72 Antwortklassen auf die Matrizen zurückgegriffen werden, die die Abstände der Spiketrains enthalten, die nach Geschwindigkeit in neun Klassen eingeteilt waren. Es sei daran erinnert, dass nie alle neuronalen Antworten auf einen Stimulus verwendet wurden, sondern nur die je ersten zehn (siehe oben) beziehungsweise ersten achtzig (siehe Kapitel 5.2.1) der sieben bis acht Wiederholungen. Daraus resultiert, dass die Menge der Einträge der verwendeten Matrix nicht identisch ist zur Menge der Matrixeinträge, die die Distanzen zwischen den Spiketrains eingeteilt nach Geschwindigkeitsübergängen darstellen. Allerdings unterscheiden sich die beiden Mengen der verwendeten Spiketrains maximal um 80 (Daten02 und Daten10) beziehungsweise 70 (Daten14) Elemente. Dies bedeutet für die benutzte Distanzmatrix, dass sie maximal um $80 \cdot (9 \cdot 640) + 80 \cdot (9 \cdot 640) - 80 \cdot 80 = 915200$ (bzw. 700700) Einträge von der eigentlich zu betrachtenden differiert. Dies entspricht aber lediglich einem Anteil von $915200 / (9 \cdot 640)^2$ (bzw. $700700 / (9 \cdot 560)^2$), also gerade mal 2.76%. Somit kann davon ausgegangen werden, dass das so ermittelte empirische Zufallslevel repräsentativ ist für das auf der eigentlichen Distanzmatrix basierende.

Abbildung 5.12 zeigt den mit D^{spike} empirisch bestimmten mittleren Bias bei stimulusunabhängigen Spiketrains für die zehn Zellen von Daten10 unter Verwendung verschiedener Clusterexponenten. Sowohl für Prozent-Korrekt als auch die wechselseitige Information lassen sich viele Übereinstimmung zu Abbildung 5.5 feststellen. Auch hinsichtlich Abstandsmaß und Datensatz ähneln diese Ergebnisse zum Bias denen im vorigen Kapitel. Mit der Absicht den Leser nicht mit wiederholenden Ausführungen zum Bias zu ermüden,

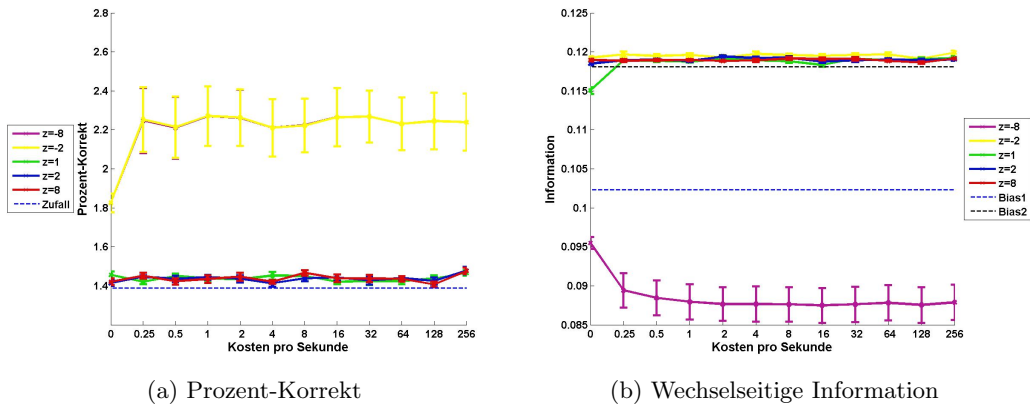


Abbildung 5.12: Güte des Clusters bei Vorliegen stimulusunabhängiger Antworten und analytische Abschätzung des entsprechenden Bias. Mittelung über zehn Zellen von Daten10 für D^{spike} . Balken zeigen den Zweifachen des Standardfehlers an. Wechselseitige Information ist mit $\log_2 72$ normiert.

wird im Folgenden nur auf abweichende Ergebnisse eingegangen. So ist das empirische Zufallslevel für einen Clusterexponenten von -8 am Geringsten (5.12b) und liegt unter der analytischen Abschätzung des Bias über den ersten Summanden aus Gleichung 3.2 (Bias1). Für alle anderen Clusterexponenten werden in etwa gleich große Werte der wechselseitigen Information angenommen, die deutlich größer als Bias1 sind. Wird in die Abschätzung des Bias die zugrundeliegende Verteilung einbezogen (Bias2), ist nur eine kleine Differenz zu verzeichnen. Das Verhältnis der Stimulusklassen zur Anzahl der Spiketrains pro Stimulus bedingt, dass sich für die Zellen von Daten14 das empirische Zufallslevel deutlicher von denen der beiden anderen Datensätzen nach oben hin abhebt als es im vorhergehenden Kapitel der Fall war (nicht gezeigt).

Vergleich der Abstandsmaße

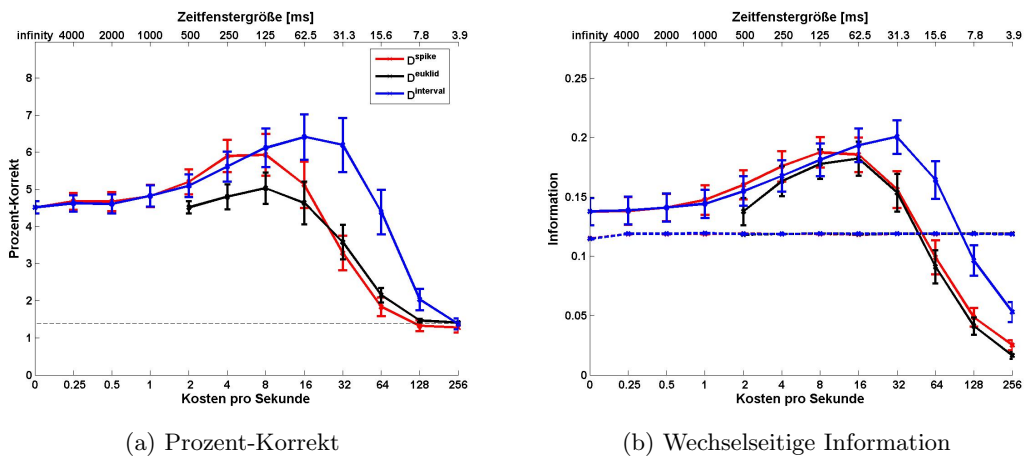


Abbildung 5.13: Mittlere Güte des stimulusabhängigen Clusters mit $z=1$ für zehn Zellen von Daten10. Balken zeigen den Zweifachen des Standardfehlers an. 5.13a Zufallslevel liegt bei 1.39% (gestrichelte Linie). 5.13b Empirische Zufallslevel der drei Abstandsmaße überlagern sich (gestrichelte Linien in entsprechender Farbe). Wechselseitige Information ist mit $\log_2 72$ normiert.

In Abbildung 5.13 ist die mit Prozent-Korrekt und wechselseitiger Information bestimmte mittlere Güte des stimulusabhängigen Clusters mit $z = 1$ für die drei Typen von Abstandsmaßen für den zweiten retinalen Datensatz dargestellt. Der Verlauf der Güte zeigt für die beiden Fehlermaße und alle drei Abstandsmaße ein Maximum, wobei sich D^{spike} und D^{euclid} wieder recht ähnlich sind. So befindet sich das Maximum an ähnlicher Stelle und die Werte für D^{euclid} liegen unter oder sehr nahe bei D^{spike} . Dabei zeigt sich für Prozent-Korrekt eine größere Differenz. Ebenfalls beiden Fehlermaßen gemein ist, dass für $D^{interval}$ im Mittel eine höhere Güte erreicht wird und sich die Maximumsstelle um ein bis zwei Werte für q auf etwa 16 beziehungsweise 32 verglichen mit D^{spike} nach rechts verschiebt.

Als Unterschied zwischen den Fehlermaßen lässt sich feststellen, dass das jeweilige Maximum für die drei Abstandsmaße für die wechselseitige Information auf einer feineren Zeitskala als für Prozent-Korrekt angenommen wird. Des Weiteren sei angemerkt, dass für Prozent-Korrekt die Werte ab etwa $q = 128$ und $\tau = 7.8$ auf das Zufallslevel abfallen, wohingegen die wechselseitige Information ab $q = 64$ beziehungsweise $q = 128$ und $\tau = 15.6$ unter das empirischen Zufallslevel fallen.

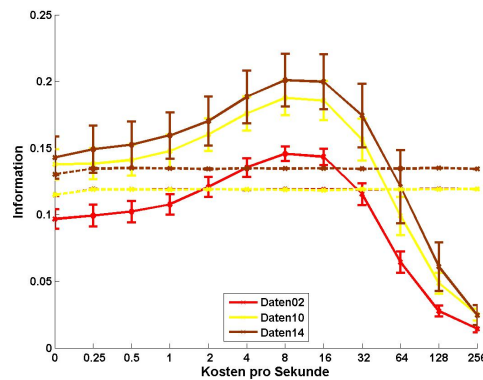


Abbildung 5.14: Vergleich der Güte des stimulusabhängigen Clusters mit $z=1$ und D^{spike} für die drei Datensätze. Dabei wurde über die verwendeten Zellen der jeweiligen Datensätze gemittelt. Balken zeigen das Zweifache des Standardfehlers an. Mittleres empirisches Zufallslevel (gestrichelte Linie in entsprechender Farbe) für die Zellen von Daten02 und die von Daten10 überlagern sich. Wechselseitige Information ist mit $\log_2 72$ normiert.

Die drei Datensätze im Vergleich weisen in ihren Ergebnissen der Güte des stimulusabhängigen Clusters die gleichen Gemeinsamkeiten und Unterschiede wie im vorhergehenden Kapitel auf. Abbildung 5.14 möge dies beispielhaft veranschaulichen. So ist hier die wechselseitige Information für D^{spike} für alle drei Datensätze dargestellt. Die Kurven verlaufen nahezu parallel, wobei sich zudem Daten10 und Daten14 quantitativ sehr ähnlich sind und Daten02 dagegen quantitativ schlechter ist.

Es lässt sich also zusammenfassen, dass sich für alle Abstandsmaße, Fehlermaße und Datensätze ein Maximum der Unterscheidbarkeit verzeichnen lässt. Zur bestmöglichen Unterscheidung der Spiketrains gibt es demnach eine jeweilige optimale Zeitskala. Diese wurde für jedes Abstandsmaß für jede Zelle ermittelt und das entsprechende Histogramm über alle 25 Zellen für Prozent-Korrekt und wechselseitige Information (Abbildung 5.15)

erstellt. Bei der Bestimmung der Maximalstelle wurde diese nur in die Statistik aufgenommen, sofern der Maximalwert über dem empirischen Zufallslevel plus dem Zweifachen der Standardabweichung lag, was für Prozent-Korrekt immer der Fall war. Dagegen sind bei Verwendung der wechselseitigen Information auf Grund dieses Kriteriums für Daten02 eine nicht-richtungssensitive und eine sonstige Zelle sowohl für D^{spike} als auch D^{euklid} und für Daten14 eine richtungssensitive Zelle für alle Abstandsmaße entfallen.

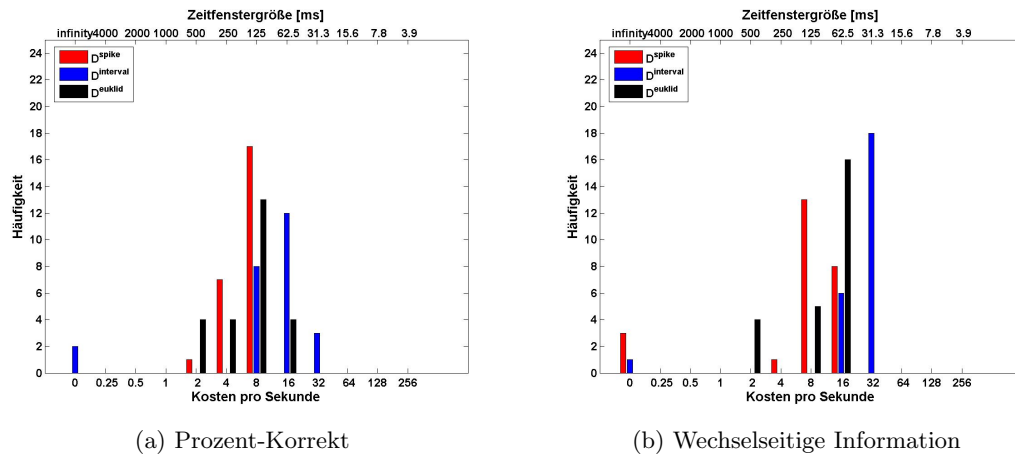


Abbildung 5.15: Histogramm der Maximumsstellen für die drei Typen von Abstandsmaßen über alle 25 Zellen

Die Histogramme spiegeln einige der Eigenschaften wie sie bei der Mittelung über den Zellen in Abbildung 5.13 verzeichnet wurden wieder. So liegt das Maximum tendenziell häufiger für die wechselseitige Information für alle drei Abstandsmaße auf einer feineren Zeitskala als bei Prozent-Korrekt vor. Für letzteres Fehlermaß sind die Maximalstellen für jedes Abstandsmaß etwas breiter gestreut und die Zellanzahlen legen zwar ein optimalen Bereich für q beziehungsweise τ nahe, doch ist dieser für die wechselseitige Information stärker betont. Für die Abstandsmaße selbst lässt sich feststellen, dass ein bestmögliches Unterscheiden der Spiketrains mit D^{euklid} und D^{spike} einer ähnlichen Zeitskala von $125ms$ beziehungsweise $125ms$ bis $62.5ms$ bedarf. Dabei wird für D^{euklid} tendenziell das Maximum der Fehlermaße häufiger auf einer etwas feineren Zeitskala angenommen. Für $D^{interval}$ erhöht sich diese auf $q = 16$ beziehungsweise $q = 32$, also $62.5ms$ beziehungsweise $31.5ms$.

Vergleich der nicht- und richtungssensitiven Zellen

Mit Hinblick auf das vorhergehende Kapitel stellt sich die Frage, ob die Zeitskala, für die eine bestmögliche Unterscheidung der Spiketrains hinsichtlich der Geschwindigkeitsübergänge erreicht wird, eine Abhängigkeit von der Richtungssensitivität einer Zelle zeigt. Daher wurden die Mittelwerte der nach dem vorhergehenden Kriterium ermittelten Maximalwerte getrennt nach richtungssensitiven, nicht-richtungssensitiven und sonstigen Zellen für jedes Abstandsmaß und Fehlermaß bestimmt. Gemittelt wurde dabei über die jeweilige

Anzahl Zellen für die eine Maximumsstelle für das entsprechende Abstandsmaß bestimmt werden konnten.

Wie man Tabelle 5.2a entnehmen kann, ist keine systematische Abhängigkeit von der Richtungssensitivität zu erkennen. Einige Werte, wie etwa Prozent-Korrekt für D^{spike} und wechselseitige Information für $D^{interval}$, sind sogar für alle drei Zellklassen sehr ähnlich. Auch für alle anderen Kombinationen an Abstands- und Fehlermaßen lassen sich keine Auffälligkeiten hinsichtlich der Richtungssensitivität feststellen.

		D^{spike}	$D^{interval}$	D^{euclid}
R	Info	9.25	28	62.5
	Pro	6	9.6	225
NR	Info	10.55	28	81.25
	Pro	6	12.67	161.45
S	Info	11.43	28	80.36
	Pro	8	19	164.06

(a) Mittlere Maximumsstelle

	W. Information	Prozent-K.
R	60%	20%
NR	16.6%	41.67%
S	25%	37.5%

(b) Prozentualer Anteil an Zellen, für die das Maximum von D^{spike} größer ist als von $D^{interval}$

Tabelle 5.2: Charakterisierung der richtungssensitiven (R), nicht-richtungssensitiven (NR) und sonstigen (S) der insgesamt 25 Zellen auf Basis der unter 5.2a und 5.2b beschriebenen Aspekte für Prozent-Korrekt (Pro) und wechselseitige Information (Info).

Abbildung 5.13 legte nahe, dass $D^{interval}$ quantitativ zu einer besseren Unterscheidung der neuronalen Antworten führt als D^{spike} . Bei Betrachtung der Güte des stimulusabhängigen Clusters für einzelne Zellen findet man aber für manche Zellen auch den umgekehrten Fall vor. Bestimmt man nun den prozentualen Anteil an Zellen für die dies festgestellt werden kann, erhält man keine klare Aussage über die Abhängigkeit von der Richtungssensitivität. So ist Tabelle 5.2b zu entnehmen, dass für die wechselseitige Information zwar ein größerer Anteil an richtungssensitiven Zellen diese Eigenschaft aufweisen, diese Abhängigkeit mittels Prozent-Korrekt aber nicht zu verzeichnen ist.

Auswirkungen der Clusterexponenten

Eine Änderung des Clusterexponenten z von eins auf die Werte $-8, -2, 2, 8$ hat auf alle drei Datensätze und für alle Abstandsmaße den qualitativ gleichen Einfluss. Daher ist die Güte des stimulusabhängigen Clusters für diese Exponenten in Abbildung 5.16 nur mit D^{spike} für Daten10 abgebildet. Zur besseren Übersichtlichkeit wurde dabei auf die Darstellung der Standardfehler verzichtet.

Wie im vorhergehenden Kapitel kann man feststellen, dass sowohl nach Prozent-Korrekt (Abbildung 5.16a) als auch nach wechselseitiger Information (Abbildung 5.16b) die Unterscheidung der neuronalen Antworten quantitativ besser werden je kleiner der Clusterexponent wird. Ebenfalls gemein ist ihnen, dass mit betragsmäßig größer werdenden negativen Werten für z die Güte des stimulusabhängigen Clusters bei größer werdendem Parameter q abnimmt. So verschlechtert sich die Güte für $z = -2$ für beide Fehlermaße deutlich für

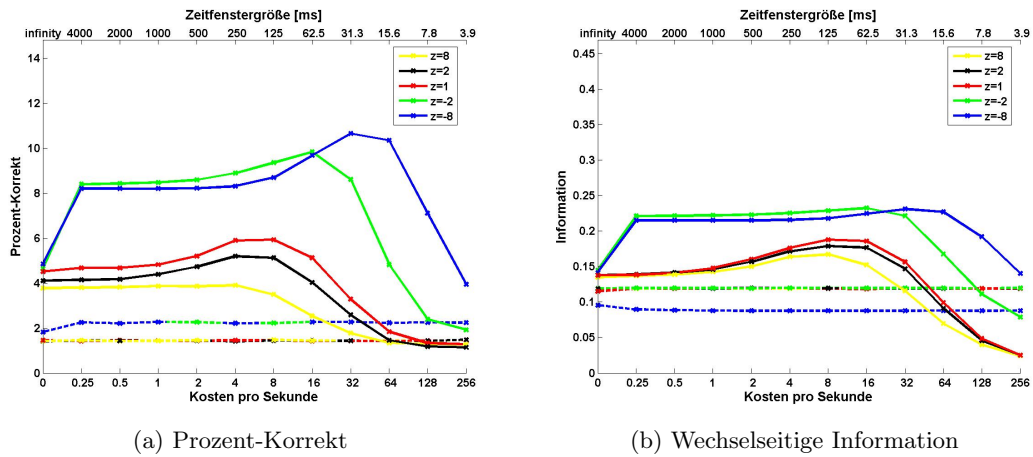


Abbildung 5.16: Mittlere Güte des stimulusabhängigen Clusters mit D^{spike} für Daten10 und verschiedene Clusterexponenten. Gestrichelte Linien zeigen empirische Zufallslevel (farblich entsprechend). Wechselseitige Information ist mit $\log_2 72$ normiert.

ein q größer als 32 und für $z = -8$ für ein q größer als 64. Die Werte für D^{count} , also bei einer Differenzierung allein auf Basis der Spikeanzahlen, sind aber nahezu unabhängig von z . Kleine Unterschiede in der Auswirkung des Clusterexponenten auf die mittels der beiden Fehlermaße bemessenen Güte des stimulusabhängigen Clusters zeigt sich in der Ausprägung des Maximums. So nimmt diese für Prozent-Korrekt mit kleiner werdendem z zu, wohingegen für die wechselseitige Information ein umgekehrter Effekt beobachtet werden kann.

5.2.3 Multineuronale Dekodierung der Geschwindigkeit

Zur Analyse mit dem Abstandsmaß $D^{multi}[q, k]$ (siehe Kapitel 2.3) werden im Folgenden nur Kombinationen aus zwei von vier Zellen betrachtet. Dazu wurden zwei nicht-richtungssensitive Zellen NR1 und NR2 (Elektrode 26, Prototyp 3 und Elektrode 35, Prototyp 1, siehe Anhang) und zwei für die gleiche Richtung sensitive Zellen R1 und R2 (Elektrode 54, Prototyp 1 und Elektrode 56, Prototyp 2) von Daten14 ausgewählt und für alle Kombinationen stimulusabhängiges Clusters durchgeführt. Dabei wurden die gekennzeichneten Spiketrains wieder in neun nach Geschwindigkeit getrennte Stimulusklassen eingeteilt. Es wurde aber nicht mit den neuronalen Antworten auf die insgesamt 560 Präsentationen eines Stimulus, sondern nur mit je 100 zufällig ausgewählten Antworten gerechnet. Der Parameter q wurde dabei auf 0, 1, 2, 4, 8, 16, 32, 64, 128 und 256 gesetzt und je mit einem k von 0, 0.2, 0.6, 1, 1.5 und 2 kombiniert. Der geringe Stichprobenumfang und die heruntergesetzte Anzahl an verwendeten Präsentationen eines Stimulus sind bedingt durch die hohe Rechenzeit (siehe auch Diskussion).

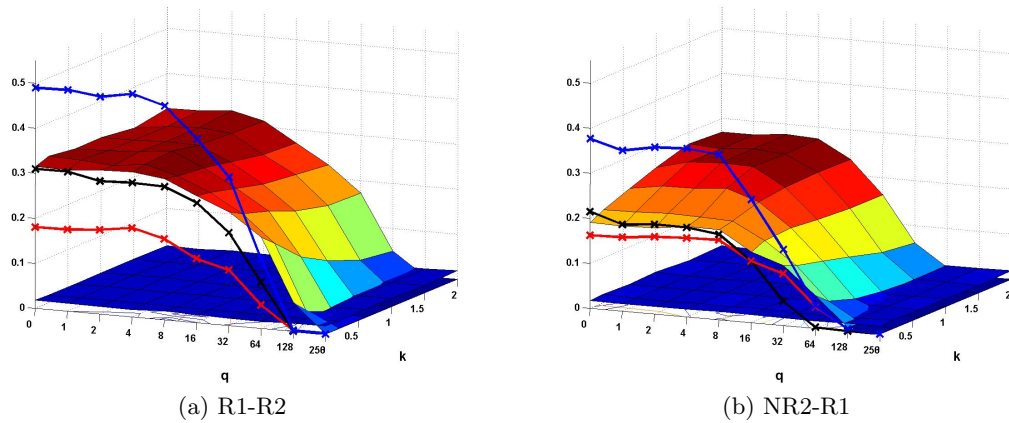


Abbildung 5.17: Güte des stimulusabhängigen Clusters mit $D^{multi}[q, k]$ mit $z=1$ (bunte Fläche) für die Kombination zweier richtungssensitiver Zellen (rote (R1) und schwarze Linie (R2) in 5.17a) und einer richtungssensitiven (R1, schwarze Linie) und nicht-richtungssensitiven (NR2, rote Linie) Zelle (5.17b). Die blaue Fläche zeigt das empirische Zufallslevel mit $D^{multi}[q, k]$ und die blaue Linie stellt je die Summe der einzelnen Zellen (für $D^{spike}[q]$) dar. Wechselseitige Information ist mit $\log_2 9$ normiert.

Abhängigkeit von den Parametern q und k

In Abbildung 5.17 ist die Güte des stimulusabhängigen Clusters mit $D^{multi}[q, k]$ und $z = 1$ dargestellt. Dabei sind nur die mittels wechselseitiger Information bestimmten Ergebnisse abgebildet, da die mittels Prozent-Korrekt bestimmten ähnlich sind. Neben den Werten für die multineuronale Betrachtung ist zu sehen, wie gut die entsprechenden hundert Spiketrains der einzelnen Zellen unterschieden werden konnten (rote und schwarze Linie) und die summierte Güte des stimulusabhängigen Clusters für die beiden Zellen (blaue Linie). In Abhängigkeit von q zeigt sich für die Kombination sowohl zweier richtungssensitiver Zellen (Abbildung 5.17a) als auch einer nicht- und einer richtungssensitiven Zelle (Abbildung 5.17b) ein qualitativ sehr ähnlicher Verlauf zu den entsprechenden einzelnen Neuronen.

Die Abhängigkeit von k ist jedoch nicht für alle Zellkombinationen so ähnlich. Die Güte des stimulusabhängigen Clusters ändert sich im Fall zweier richtungssensitiver Zellen mit k kaum. So scheint sich lediglich die Unterscheidung der Neurone auf einer feineren Zeitskala etwa zwischen $q = 16$ und $q = 128$ ganz gering mit größer werdendem k zu verschlechtern. Bei der Betrachtung der beiden nicht-richtungssensitiven Ganglienzellen sind hierzu keine Unterschiede zu erkennen (nicht gezeigt). Dagegen kann man für beide Kombinationen von nicht- und richtungssensitiver Zelle eine deutlichere Abhängigkeit von k sehen (Abbildung 5.17b). So nimmt die wechselseitige Information von $k = 0$ bis etwa $k = 1$ in einem Bereich von q zwischen 0 und 8 zu und nähert sich dem Niveau der Summe an. Der Verlauf der Güte des stimulusabhängigen Clusters mit D^{multi} zeigt wieder die gleiche Abhängigkeit von q wie bei den einzelnen Zellen mit D^{spike} .

Es zeigt sich also, dass mit $D^{multi}[q, k]$ für einen großen Bereich von q , also zeitlicher Auflösung, die Güte des stimulusabhängigen Clusters in etwa gleich gut ist. Dies, und

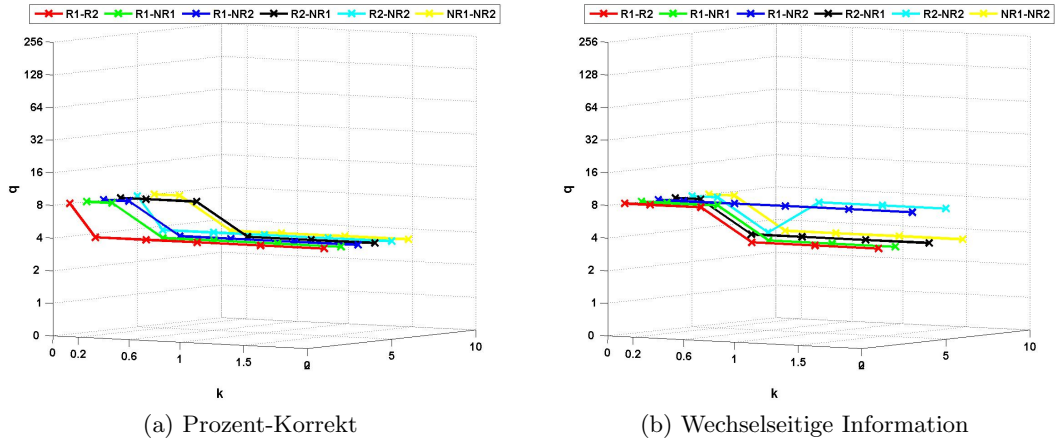


Abbildung 5.18: Stelle (q) des mindestens 10%-igen Abfalls in Abhängigkeit von k für alle betrachteten Zellkombinationen.

den Eindruck den Abbildung 5.17a vermittelt, legt die Frage nahe, ob es einen Wert für k gibt, für den dieses Intervall am größten wird und sich eine allgemeine Abhängigkeit für k feststellen lässt. Daher wird in Abhängigkeit von k das q bestimmt, für das die wechselseitige Information beziehungsweise Prozent-Korrekt darauffolgend um mehr als 10% abfällt. Dabei wurde wieder für die wechselseitige Information als Bias der erste Summand von Gleichung 3.2 abgezogen. Die entsprechenden Ergebnisse sind in Abbildung 5.18 dargestellt. Für Prozent-Korrekt lässt sich feststellen, dass die größten Werte von q , für die die Güte des stimulusabhängigen Clusters abnimmt, für besonders kleine Werte für k von 0 oder 0.2 erreicht werden. Eine ähnliche Tendenz ist mit der wechselseitigen Information zu verzeichnen, doch lässt sich hiermit weniger eindeutig sagen, dass der Bereich von q für gleichbleibend gute Unterscheidung der Antworten mit größer werdendem k kleiner wird. Die Ergebnisse geben ausserdem keinen klaren Hinweis darauf, dass die betrachtete Abhängigkeit von q und k mit der Kombination der Zellen hinsichtlich ihrer Richtungssensitivität variiert.

Redundanz

Eine wichtiger Aspekt des multineuronalen Zusammenwirkens ist, ob und inwieweit mehrere Zellen gegenüber einem Neuron zusätzliche Informationen über einen Stimulus liefern und somit zu einer besseren Güte des stimulusabhängigen Clusters führen. Um dies zu analysieren wird ein Redundanzindex nach [A03] beziehungsweise [RMV01] verwendet. Sei G_i die maximale Güte, die mit einem der beiden Fehlermaße und D^{spike} für die i -te Zelle ermittelt wurde. Bezeichne G_z die maximale Güte bei Betrachtung der Zellen zusammen mit D^{multi} und festem k . Dann wird der Redundanzindex wie folgt definiert:

$$RI = \left(1 - \frac{G_z}{G_1 + G_2}\right) / \left(1 - \frac{\max\{G_1, G_2\}}{G_1 + G_2}\right) \quad (5.2)$$

Sind die Zellen redundant und somit $G_z = \max\{G_1, G_2\}$, so ist der Redundanzindex 1. Wenn die Zellen dagegen vollständig unabhängige Information beitragen, also $G_z = G_1 + G_2$ gilt, ist der Redundanzindex 0. Sind die stimulusbezogenen Beiträge der einzelnen Zellen widersprüchlich oder verwirrend, kann der Index einen Wert größer als Eins annehmen. Ebenfalls ist es möglich, dass er kleiner als Null wird. Dies ist der Fall, wenn die Zellen synergetisch einen Reiz verschlüsseln.

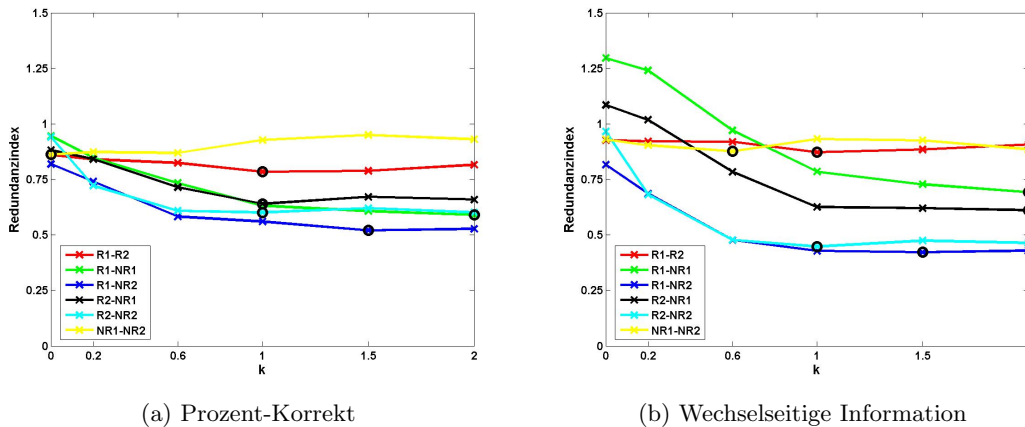


Abbildung 5.19: Redundanzindex bestimmt nach 5.2 in Abhängigkeit von k . Schwarze Kreise zeigen für welches k die Güte G_z maximal war.

In Abbildung 5.19 sind die Redundanzindizes in Abhängigkeit von k für die betrachteten Zellkombinationen dargestellt. Dabei wurde für die wechselseitige Information wieder die errechneten Werte abzüglich einem Bias bestehend aus dem ersten Summanden von Gleichung 3.2 verwendet. Sowohl nach Prozent-Korrekt (Abbildung 5.19a) als auch wechselseitiger Information (Abbildung 5.19b) liegt für die Kombination zweier Zellen mit ähnlicher Tuningkurve der Redundanzindex nicht deutlich variierend etwas unter 1. Bei Betrachtung der beiden richtungssensitiven Zellen wird die Redundanz zwar für $k = 1$ mit etwa 0.88 minimal und die Güte G_z hier maximal, doch sind diese Extrema kaum ausgeprägt. Ebenso wenig aussagekräftig erscheint der leichte Anstieg der Redundanz mit wachsendem k für die beiden nicht-richtungssensitiven Zellen.

Dagegen kann für die Kombination einer nicht- und einer richtungssensitiven Zelle in allen Fällen festgestellt werden, dass die Redundanz mit einem bis 1 wachsendem k deutlich abnimmt und dann in etwa konstant bleibt. Zwar differieren die Ergebnisse nach Prozent-Korrekt und wechselseitiger Information quantitativ, doch legen beide den Schluss nahe, dass die Betrachtung welches Aktionspotential durch welches Neuron ausgelöst wurde, eine bessere Zuordnung der neuronalen Antworten zu den verschiedenen Stimuli ermöglicht.

6 Diskussion

In den vorhergehenden Kapiteln wurden die verschiedenen Abstands- und Gütemaße als Voruntersuchungen für zukünftige Analysen retinaler Spiketrains verglichen. Dabei wurde keine eindeutig beste Methode gefunden. Des Weiteren wurden erste Ergebnisse zur Zeitstruktur der Kodierung in den vorliegenden Daten gewonnen, wobei diese publizierten Resultaten entsprechen. Im Folgenden werden die Ergebnisse der durchgeführten Untersuchungen unter methodischen und biologischen Aspekten diskutiert.

6.1 Methodische Aspekte

6.1.1 Abstandsmaße

Die Anwendung der beiden Typen von Abstandsmaßen D^{spike} und D^{euclid} auf simulierte und retinale Daten zeigt im Allgemeinen ähnliche Ergebnisse. Zu qualitativ gleichen Resultaten führten beide Methoden auch in Grewe et al. ([GKWE03]). Dies lässt für zukünftige Untersuchungen eine Beschränkung auf eines der beiden Familien von Abstandsmaße als ausreichend erscheinen. Bei der Wahl sei dann empfohlen die Rechenzeit zu berücksichtigen, die mit der mittleren Anzahl an Spikes pro Spiketrain stark variiert. Handelt es sich hierbei durchschnittlich um N Spikes, so müssen für eine Distanz nach $D^{spike}[q]$ (wie auch nach $D^{interval}[q]$, siehe 2.2.7) N^2 Berechnungen durchgeführt werden. Dagegen kann $D^{euclid}[\tau]$ so implementiert werden, dass es sich hierbei um $2N$ Schleifendurchläufe handelt. In die Rechenzeit geht aber auch ein, wie fein gefiltert wird (und somit auch die Länge eines Spiketrains), also um wieviele Sekunden das Zeitfenster τ immer verschoben wird.

Auf eine Untersuchung mit $D^{interval}$ sollte allerdings nicht verzichtet werden. Zwar zeigen die Ergebnisse der retinalen Daten kaum Unterschiede, doch haben die Simulationen gezeigt, dass es sich bei D^{spike} und $D^{interval}$ um qualitativ unterschiedliche Typen von Abstandsmaßen handelt. So führt nur D^{spike} mit einer entsprechend hohen Auflösung zu einer guten Unterscheidbarkeit der Spiketrains, wenn eine Stimulusklasse durch die Aktionspotentialzeitpunkte aber nicht durch die Länge der Interspikeintervalle charakterisiert ist (siehe Kapitel 4.2.2). Ein umgekehrter Fall ist in [VP97] dargestellt, wo die Güte des stimulusabhängigen Clusters im Fall von $D^{interval}$ weitaus über der von D^{spike} liegt. Hierbei waren die Stimulusklassen so erzeugt, dass sie hinsichtlich der Anordnung der Interspikeintervalle differieren, aber nicht hinsichtlich der absoluten Spikezeitpunkte. Natürlich kann

prinzipiell nicht ausgeschlossen werden, dass die neuronalen Antworten einem dieser Fälle entsprechen.

Bei der Berechnung von $D^{interval}$ wurde eine Motivation der Setzung je des gleichen ersten und gleichen letzten Spikezeitpunkts bereits in Kapitel 2.2.1 angesprochen. Neben dieser allgemeinen Begründung wird die Wahl einer solchen Setzung für die vorliegenden retinalen Daten untermauert durch die Ergebnisse in [W07]. Dort wird auf Grundlage des gleichen Datensatzes, der in dieser Arbeit verwendet wurden, gezeigt, dass eine Reizrekonstruktion allein auf Basis der Unterscheidung, ob ein Neuron überhaupt ein Aktionspotential ausgelöst hat oder nicht, zu einer guten Schätzung führt. Die Bedeutsamkeit der Unterscheidung zwischen einem Spike gegenüber einem leeren Spiketrain sollte also nicht unterbewertet werden.

Die Güte des stimulusabhängigen Clusters bei Vorliegen einer regulären Verteilung der Interspikeintervalle (siehe Kapitel 4.2.1) ist mit $D^{interval}$ und $D^{interval3}$ für alle Werte von q gleich gut. Bei den vorliegenden Daten (Abbildung 5.7b in Kapitel 5.2.1) führt die Information über den zeitlichen Abstand zwischen Reizbeginn und erstem Spikezeitpunkt und zwischen Reizende und letztem Spikezeitpunkt dazu, dass die Unterscheidbarkeit der Folgen der Interspikeintervalle mit $D^{interval}$ aber auf einer feineren Zeitskala besser möglich ist als mit $D^{interval3}$. Neben diesen Möglichkeiten die ersten und letzten Interspikeintervalle aus einem Spiketrain zu bestimmen, nennen Victor et al. ([VP97]) noch das als $D^{interval:min}$ bezeichnete Abstandsmaß. Hierbei werden der erste und letzte Spikezeitpunkt je neuronaler Antwort bei der Abstandsberechnung so gewählt, dass der Abstand zwischen den Interspikeintervallfolgen minimal ist. Eine Anwendung auf künstliche Daten in [VP97] zeigt, dass dieses Abstandsmaß sehr ähnliche Ergebnisse liefert wie $D^{interval}$. Auf Grund einer einfacheren Implementier- und Interpretierbarkeit ist daher Letzteres zu bevorzugen.

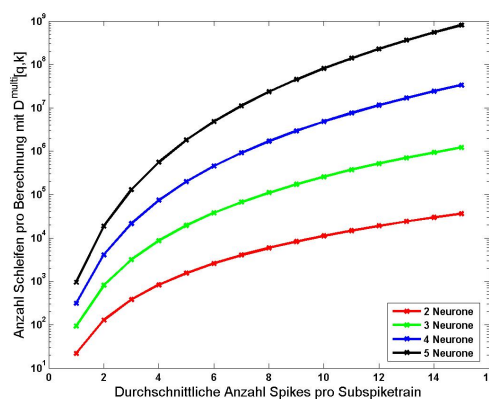


Abbildung 6.1: Anzahl Schleifendurchläufe, die im Algorithmus zur Berechnung eines Abstandes mit $D^{multi}[q, k]$ durchgeführt werden müssen, in Abhängigkeit von der durchschnittlichen Anzahl an Spikes eines Subspiketrain bei Einbeziehung unterschiedlich vieler Neurone (nach Gleichung 2.2)

Eine Abstandsberechnung mit $D^{multi}[q, k]$ ist besonders zeitaufwendig. Dieses Maß lässt aber auch Rückschlüsse über neuronale Kodierung durch mehrere Zellen zu, die über die anderen Abstandsmaße qualitativ hinausgehen. Während die Erstellung einer Distanzma-

trix mit dem Format 5760x5760 für die vorliegenden Daten durchschnittlich eine halbe Stunde Rechenzeit mit $D^{spike}[q]$ bedurfte, variierte die Rechenzeit für die multineuronale Erweiterung zwischen zwei und zehn Stunden für eine Distanzmatrix mit nur 900 Zeilen und Spalten. In Abbildung 6.1 ist nach Gleichung 2.2 für die Betrachtung unterschiedlich vieler Neurone die Anzahl an Schleifen, die der Algorithmus zur Berechnung des Abstandes $D^{multi}[q, k]$ zweier gekennzeichneten Spiketrains benötigt, in Abhängigkeit von der durchschnittlichen Anzahl an Spikes, die von einem Neuron ausgelöst werden, dargestellt. Wie man diesem Bild entnehmen kann, empfiehlt es sich, den Algorithmus in einer Programmiersprache zu schreiben, bei der das Durchlaufen von Schleifen nicht so zeitaufwendig ist wie etwa in *Matlab*. Dann ist es vielleicht auch möglich das Zusammenwirken von mehr als zwei Neuronen in einer hinreichend großen Stichprobe zu untersuchen.

Die verwendeten Abstandsmaße können alle modifiziert beziehungsweise erweitert werden. So kann man zur Filterung der Spiketrains anstelle eines rechteckigen Kerns, wie er für $D^{euklid}[\tau]$ verwendet wird, zum Beispiel auch einen exponentiellen wählen ([VR01]). Die Abstandsmaße $D^{spike}[q]$, $D^{interval}[q]$ und $D^{multi}[q, k]$ können verändert werden, indem man die Kosten zum Verschieben eines Spikes beziehungsweise der Längenänderung eines interspikeintervalls nicht mit $q|\Delta t|$ sondern zum Beispiel mit $1 - \exp(-q|\Delta t|)$ belegt. Dabei sind diese Änderungen motiviert durch die Tatsache, dass die Wirksamkeit zweier Spikes, die einen Koinzidenzdetektor (siehe Einleitung) erreichen, exponentiell abnimmt mit dem zeitlichen Abstand ([VP97]). Für das multineuronale Abstandsmaß könnte man dann noch den Preis für das Ändern der Kennzeichnung eines Spikes abhängig machen von den zugrundeliegenden Neuronen, also für alle möglichen Ummarkierungen j in i die Kosten beliebig festsetzen. Des Weiteren könnte man die elementaren Umformungen nach $D^{spike}[q]$ und $D^{interval}[q]$ kombinieren, wofür bisher jedoch kein effizienter Algorithmus gefunden werden konnte. Dies gestaltet sich auch schwer für die Idee zeitliche Strukturen von Teilfolgen eines Spiketrains zu betrachten, so dass anstelle eines Spikes beliebig viele Spikes verschoben werden dürfen.([VP97], [V05])

6.1.2 Gruppenbildung

Die Wahl eines negativen Exponenten im Algorithmus zum stimulusabhängigen Clustern führt zu einer quantitativen Verbesserung der Güte der Stimulusabhängigkeit der Einteilung, wie die hier und die in [VP97] und [VP96] dargestellten Ergebnisse zeigen. Die Variation des Exponenten kann sicherlich helfen die zugrundeliegende Datenstruktur zu charakterisieren. Eine neurobiologische Interpretation der Ergebnisse hinsichtlich einer möglicherweise zugrundeliegenden Kodierungsstrategie beziehungsweise Zeitskala ist bei Verwendung eines Exponenten von $z = 1$ sicher einfacher. Methodisch besteht ausserdem das Problem einer Setzung von 0^z für negative Werte von z . Hierbei könnte man sich zum Beispiel an der Genauigkeit der vorliegenden Spikezeitpunkte orientieren. Dies stellt ein Kriterium dar, aufgrund derer die Wahl vernünftig erfolgen kann, doch es existiert eben kein optimales und eindeutig richtiges für eine Setzung von 0^z mit negativem Wert von

z . Eine angemessene Wahl für 0^z ist umso bedeutsamer, wenn es sich bei Abständen von Null um keine Ausnahmen handelt. Dies ist zum Beispiel der Fall, wenn die neuronale Aktivität gering ist und somit mehr Abstände auf Grund leerer Spiketrains Null sind. Für die vorliegenden Daten traf gerade dies zu, denn die betrachteten Zellen reagierten insbesondere auf betragsmäßig kleine Geschwindigkeiten mit wenigen, oft gar keinen, Spikes. Diese Aspekte begründen die ausführlicheren Analysen in den vorhergehenden Kapitel und die im folgenden behandelte Interpretation auf Basis eines Exponenten von $z = 1$.

Zur Bildung der Antwortklassen hätte man auf Basis der Abstandsmaße auch einen (anderen) Clusteralgorithmus verwenden können. Victor et al. begründen aber die Wahl des von ihnen verwendeten Algorithmus zum stimulusabhängigen Clustern in [VP97]. Unter Benutzung metrischer multidimensionaler Skalierung zeigen sie mit Hilfe simulierter Daten, dass die geometrischen Zentren zweier Stimulusklassen gleich sein können. Ein Clusteralgorithmus, der darauf basiert, dass die Spiketrains in konvexen Wolken liegen, wäre in diesem Fall ungünstig. Dies legt somit die Wahl eines Algorithmus nahe, der keine Annahmen über die Form der Cluster macht, wie etwa der Algorithmus von Victor et al.

6.1.3 Prozent-Korrekt und wechselseitige Information

Prozent-Korrekt stellt eine anschauliche Möglichkeit dar, um die Güte des stimulusabhängigen Clusters zu bestimmen und der Schätzer ist erwartungstreu im Gegensatz zu dem der wechselseitigen Information. Allerdings bezieht sich Prozent-Korrekt nur auf die Diagonale der aus dem Algorithmus 3.1.1 ermittelten Matrix N , wohingegen die wechselseitige Information die gesamte Matrix einbezieht um die Stimulusabhängigkeit der Klasseneinteilung zu bewerten. Trotz dieser Unterschiede sind die Ergebnisse basierend auf simulierten und retinalen Daten mit beiden Gütemaßen qualitativ sehr ähnlich.

Bei der Anwendung der wechselseitigen Information ist die Verfälschtheit des Schätzers zu beachten. Mit dieser Problematik gehen Victor et al. in ihren ersten Arbeiten ([VP96], [VP97]) um, indem sie ein empirisches Zufallslevel bestimmen (siehe Kapitel 5.2.1) und Ergebnisse, die über diesem liegen, als signifikante Güte stimulusabhängigen Clusters werten. In späteren Veröffentlichungen ([VP98], [ARMV03]) wird allerdings dieses empirische Zufallslevel als Bias von den ermittelten Werten abgezogen. In den vorhergehenden Kapiteln wurde der erste Zugang bevorzugt. Zwar gibt dieses empirische Zufallslevel einen empirisch geschätzten Bias an, aber eben nur den bei Vorliegen gleichverteilter stochastisch unabhängiger Zufallsvariablen.

Abbildung 5.5b zeigt, dass bei unseren retinalen Daten, bei denen als Stimuli die unterschiedlichen Geschwindigkeiten betrachtet wurden, dieser empirisch bestimmte Bias, der analytisch bestimmte Bias ohne eine konkrete Verteilungsannahme (erster Summand nach Gleichung 3.2) und der analytisch bestimmte Bias unter Annahme gleichverteilter stochastisch unabhängiger Zufallsvariablen (ersten beiden Summanden nach Gleichung 3.2) sehr dicht beieinander liegen. In solchen Fällen erscheint es legitim den empirisch ermittelten Bias abzuziehen. Die Situation bei Betrachtung der Geschwindigkeitsübergänge stellt sich

in Abbildung 5.12b aber anders dar. So zeigt sich, wie stark eine Schätzung des Bias mit dem Zugrundelegen einer konkreten Verteilung variieren kann.

Dies und das Anliegen einer einheitlichen Vorgehensweise begründete, stets mit den Ergebnisse in Anlehnung an die ersten Arbeiten von Victor et al. umzugehen. Dabei wurde Bezug genommen auf das empirische Zufallslevel, auch wenn diese Verfälschung vom Wert 0 in den betrachteten Fällen durch eine analytische Berechnung gut abgeschätzt wurde. Prinzipiell sollte auch stets zunächst das empirische Zufallslevel bestimmt werden, da nicht auszuschließen ist, dass das Abstandsmaß beziehungsweise der Parameter q Einfluss auf dieses haben und somit eine rein analytische Abschätzung keine gute Näherung darstellen könnte. Entsprechende Beispiele sind hierfür in [VP97] gegeben. In den hier betrachteten Fällen wurden aber keine solche Abhängigkeiten festgestellt. Mit der Absicht der Verfälschung gerecht zu werden, wurde daher bei Betrachtung von Verhältnissen (wie beim Abbruchkriterium und Redundanzindex) ein Bias in Form des ersten Summanden nach Gleichung 3.2 abgezogen.

Der Leser möge sich nicht wundern, dass insbesondere bei Betrachtung der Geschwindigkeitsübergänge viele Ergebnisse unter das empirische Zufallslevel fallen. Bei großen Werten von q mit $D^{spike}[q]$ zum Beispiel werden die Spiketrains ja nicht zufällig in Gruppen eingeteilt. So ist es in diesen Fällen oft teurer einen Spike zu verschieben als einen zu löschen und zu einem anderen Zeitpunkt wieder einzufügen, weshalb die Abstände immer öfter zu den Stimulusklassen mit vielen leeren Spiketrains minimal sind und somit alle Spiketrains in eine oder wenige Klassen eingeteilt werden.

Dies zeigt auch, wie ungünstig es wäre, das empirische Zufallslevel als Bias zu verwenden, der abgezogen werden sollte. Es muss in diesen Fällen nach obigen Erläuterungen von einer zugrundeliegenden Verteilung ausgegangen werden, bei der viele der p_{ij} gleich Null sind. Diese unterscheidet sich somit sehr stark von einer stochastisch unabhängigen Gleichverteilung zweier Zufallsvariablen. Zudem müssen nach Bemerkung 3.2.20 dementsprechende Modifikationen für die Berechnung eines Bias in Form des ersten Summanden von Gleichung 3.2 vorgenommen werden, weil der Bias sonst überschätzt würde.

Auf Grund des sehr hohen Bias der wechselseitigen Information sind die Ergebnisse hiermit für die Geschwindigkeitsübergänge etwas vorsichtiger zu interpretieren. Damit zukünftig sicherere Aussagen über den retinalen Kode getroffen werden können, sollte daher bei der nächsten Experimentplanung der Bias bedacht werden. Eine geeignete Anzahl der Stimuli und der jeweiligen Anzahl an Präsentationen könnte dann mit Hilfe von Gleichung 3.2 und einem Vergleich dieser Biasabschätzung mit den bisher gewonnenen und damit eventuell zu erwartenden Ergebnissen bestimmt werden.

6.2 Biologische Aspekte

6.2.1 Neuronale Kodierung

Geschwindigkeit

Die Ergebnisse in Kapitel 5.2.1 variieren zwar quantitativ für die verschiedenen Datensätze, zeigen qualitativ aber keine Unterschiede. So liegt die normierte wechselseitige Information als Güte des stimulusabhängigen Clusters im Mittel für die Datensätze zwischen 0.13 und 0.21. Dabei lässt sich dieser Unterschied nicht über einen unterschiedlichen Anteil bestimmter Zelltypen hinsichtlich der Richtungssensitivität in einem Datensatz erklären. Die Ursachen liegen somit vielleicht in der Experimentdurchführung oder die Auswahl war einfach zufällig schlechter.

Hinsichtlich der Quantität sei angemerkt, dass wir ja als Reiz die Geschwindigkeit inklusive der Richtung betrachten. Letztere Eigenschaft des Reizes wird aber hauptsächlich von den richtungssensitiven Zellen kodiert, wie in [Th07] gezeigt wird. Abbildung 6.2 legt ebenfalls nahe, dass nicht alle Ganglienzellen die Richtungsinformation verschlüsseln. So nimmt die Güte des stimulusabhängigen Clusters zu, wenn man die neuronalen Antworten nur nach den absoluten Geschwindigkeiten und somit in fünf Klassen einteilt. Qualitativ zeigt sich aber kein Unterschied.

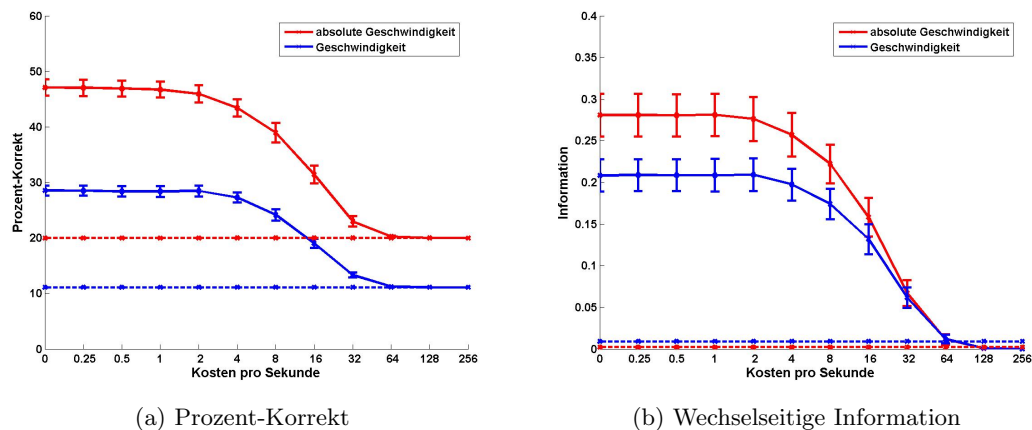


Abbildung 6.2: Mittlere Güte des stimulusabhängigen Clusters mit $z = 1$ für die zehn von Daten10 betrachteten Zellen, wenn die Stimuli die fünf absoluten Geschwindigkeiten sind im Vergleich mit der entsprechenden Güte mit zuvor betrachteten neun Geschwindigkeiten. Balken stellen je das Zweifache des Standardfehlers dar.

Die Ergebnisse legen insgesamt nahe, dass der Verschlüsselung der Geschwindigkeit eines visuellen Reizes auf retinaler Ebene ein Ratenkode zugrunde liegt. So zeigt 5.11 zwar, dass die retinalen Spiketrains verglichen mit den auf Basis eines homogenen Poisson-Prozesses simulierten Spiketrains eine gewisse zeitliche Struktur aufweisen (siehe negative Exponenten), doch ähneln sich die Ergebnisse für $z = 1$. Wäre die Information über einen Stimulus also über die Anzahl der Spikes kodiert (wie beim homogenen Poisson-Prozess),

so würde die Güte des stimulusabhängigen Clusters auch die Abhängigkeit von der zeitlichen Präzision zeigen, wie es sich in den retinalen Ergebnissen darstellt. Auf Basis der Spikeanzahlen können die Geschwindigkeiten genauso gut unterschieden werden wie auf Basis der Spikezeitpunkte bis zu einer zeitlichen Auflösung von $250 - 125ms$ und auf Basis der Interspikeintervalllängen bis zu einer Auflösung von $62.5 - 31.3ms$.

Es besteht die Möglichkeit, dass bei Stimulation der Ganglienzellen mit feiner abgestuften absoluten Geschwindigkeiten, die Ergebnisse auf eine andere der Kodierung zugrundeliegenden Zeitskala hinweisen könnten. So fanden Samonds et al. ([SB04]) mit dem auf den Metriken $D^{spike}[q]$ und $D^{interval}[q]$ basierenden Verfahren nach Victor et al. ([VP96]), dass sich die kortikale Kodierung zur Unterscheidung von Orientierungen mit dem Anspruch der Aufgabe ändert. So scheint bei groben Orientierungsdifferenzen die Stimulusinformation vor allem in der Spikerate verschlüsselt zu sein, wohingegen die zeitliche Präzision von Spikezeitpunkten und Interspikeintervallen die Information zur Unterscheidung feiner Differenzen gewährleistet.

Geschwindigkeitsübergänge

Die Ergebnisse der Geschwindigkeitsübergänge zeigten hinsichtlich der drei Datensätze die gleichen Unterschiede und Gemeinsamkeiten wie die der Geschwindigkeit. So war die Güte des stimulusabhängigen Clusters der Zellen von Daten02 im Mittel quantitativ schlechter als die der von Daten10 und Daten14, zeigte aber die gleiche Abhängigkeit von der zeitlichen Präzision. Die maximalen Werte für Prozent-Korrekt lagen im Durchschnitt bei Betrachtung der Geschwindigkeitsübergänge deutlich unter denen der Geschwindigkeit, allerdings liegt in ersterem Fall das Zufallslevel auch deutlich niedriger, weshalb sich diese Tatsache relativiert. Dagegen waren die Werte der wechselseitigen Information denen bei Betrachtung der Geschwindigkeit ähnlich, allerdings muss auf Grund des hohen empirischen Zufallslevels und eines schätzungsweise hohen Bias von eigentlich schlechteren Werten ausgegangen werden.

Insgesamt ist die Güte des stimulusabhängigen Clusters aber so gut, dass darauf geschlossen werden kann, dass in den neuronalen Signalen der Ganglienzellen auch eine Information über die Geschwindigkeitsübergänge verschlüsselt ist. Dabei zeichnet sich für beide Fehlermaße ab, dass diese Information durch eine zeitliche Präzision von Spikezeitpunkten (etwa $125 - 62.5ms$) und Interspikeintervalle (etwa $62.5 - 31.3ms$) gewährleistet wird. Eine ähnliche Zeitskala wurde auch bei einer anderen Arbeit zu retinalen Ganglienzellen gefunden. So erwies sich eine zeitliche Auflösung von $100ms$ bei einer Anwendung von D^{spike} bei Ganglienzellantworten eines Tigersalamanders in [CR05] als bedeutsam.

Wie bereits angesprochen, ist die Datenlage nicht sehr gut. Mit dem Ziel zuverlässigere Aussagen über die der Kodierung zugrundeliegenden Zeitskala machen zu können, sollte daher ein weiteres Experiment betrachtet werden, bei dem die jeweiligen Geschwindigkeitsübergänge öfter vorkommen. Sollte dann wieder eine Differenz der optimalen zeitlichen Auflösung nach Prozent-Korrekt und wechselseitiger Information festzustellen sein,

könnte es lohnenswert sein, die entsprechend zugrundeliegenden Gruppeneinteilungen zu vergleichen.

Symmetrische und richtungssensitive Zellen

Die Einteilung der Zellen auf Grund ihrer Richtungssensitivität erfolgte auf Basis der Betrachtung ihrer Tuningkurven. Dies wird durch den kleinen Stichprobenumfang legitimiert, ein größerer hätte dagegen ein systematisches Verfahren, wie etwa in [Th07], erfordert.

Für beide Zelltypen konnte nachgewiesen werden, dass sie fähig sind sowohl Geschwindigkeit als auch Geschwindigkeitsübergänge zu kodieren. Dies entspricht den Resultaten in [Th07], wo mit Hilfe Bayes'scher Schätztheorie die Stimuli rekonstruiert wurden.

Die Ergebnisse lassen nicht darauf schließen, dass richtungssensitive und nicht-richtungssensitive Ganglienzellen unterschiedliche Kodierungsstrategien verwenden. So lag die maximale zeitliche Auflösung, bei der die Güte des stimulusabhängigen Clusters in etwa so hoch war wie die Güte auf Basis der Spikeanzahlen, zwar bei den richtungssensitiven Zellen im Durchschnitt höher als bei den symmetrischen, doch ist dies ein eher schwacher Unterschied. Zudem bedeutet eine gute Unterscheidbarkeit bei einer höheren zeitlichen Auflösung ja nicht, dass die Stimulusinformation auch auf dieser feinen Zeitskala entschlüsselt wird, sondern nur, dass die Information auf dieser verschlüsselt ist.

Bei den Geschwindigkeitsübergängen gaben die betrachteten Aspekte keinen Hinweis darauf, dass verschiedene Zelltypen hinsichtlich ihrer Richtungssensitivität quantitativ oder qualitativ die Stimulusinformation unterschiedlich verschlüsseln.

Transient oder sustained

Die Histogramme der Spikezeitpunkte für die verschiedenen Geschwindigkeiten zeigten für die Zellen stets einen Peak zu Beginn der 500ms und dann ein Plateau, wie zum Beispiel in Abbildung 6.3 zu sehen ist. Zusammen mit den Ergebnissen legt dies die Vermutung nahe, dass die Information über die Geschwindigkeit in den 500ms durchgehend kodiert ist (*sustained*), wohingegen die Information über die Geschwindigkeitsübergänge in einer transienten Aktivität der Zellen niedergelegt ist.

Diese Hypothese muss durch weitere Untersuchungen überprüft werden. So sollte die Güte des stimulusabhängigen Clusters bei Betrachtung der Geschwindigkeiten quantitativ schlechter werden, wenn bei der

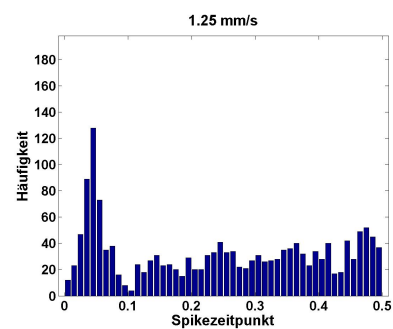


Abbildung 6.3: Histogramm der Spikezeitpunkte einer Zelle (Daten14, Elektrode 26 Prototyp 3) bei der Geschwindigkeit 1.25mm/s

Abstandsbestimmung Spiketrams mit kürzerer Dauer verwendet werden. Dagegen sollte dieser Effekt für die Geschwindigkeitsübergänge nicht beziehungsweise in nur geringem Ausmaß vorgefunden werden.

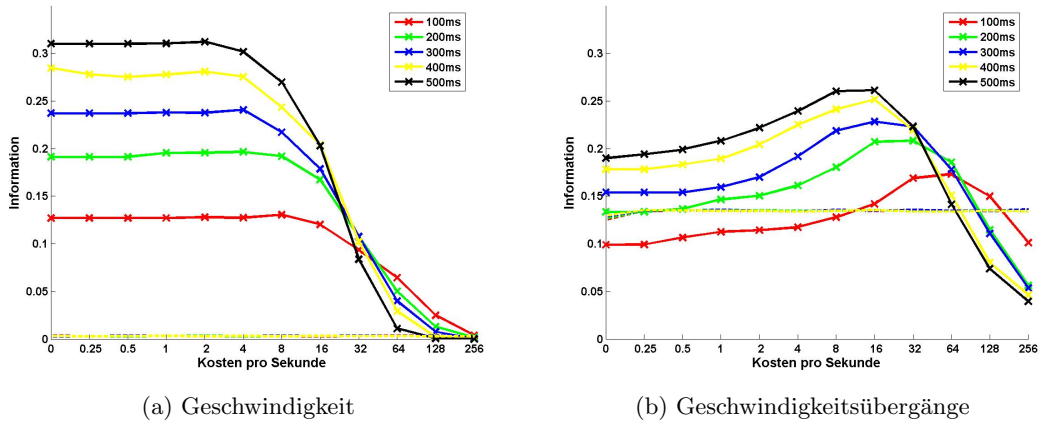


Abbildung 6.4: Güte des stimulusabhängigen Clusters mit D^{spike} , wenn je die ersten 100, 200, 300, 400ms und 500ms der neuronalen Antworten einer Ganglienzelle (Daten14, Elektrode 26 Prototyp 3) verwendet werden. Wechselseitige Information ist je mit $\log_2 9$ (6.4a) bzw. $\log_2 72$ (6.4b) normiert. Gestrichelte Linien zeigen (je farblich entsprechend) das empirische Zufallslevel an.

Dies wurde beispielhaft für eine Ganglienzelle mit D^{spike} durchgeführt und in Abbildung 6.4 sind die entsprechenden Ergebnisse für die wechselseitige Information zu sehen. Wie man Abbildung 6.4a entnehmen kann, verschlechtert sich die Güte des stimulusabhängigen Clusters deutlich mit kürzerer Dauer. Auch bei den Geschwindigkeitsübergängen wird diese schlechter, wobei die Spannweite der maximalen Information (0.5431) etwas unter der der Geschwindigkeit (0.5756) liegt. Während sich die Abhängigkeit von der zeitlichen Präzision der Spikezeitpunkte mit unterschiedlicher Länge der betrachteten neuronalen Antworten für die Geschwindigkeit so gut wie nicht ändert, verschiebt sich die optimale zeitliche Auflösung für die Geschwindigkeitsübergänge von etwa 125 – 62.5ms auf 31.3 – 15.6ms.

Dieses Beispiel deutet also daraufhin, dass die Information über die Geschwindigkeitsübergänge durch die Peaks mit einer zeitlichen Präzision der Spikezeitpunkte gewährleistet wird. Ob hier wirklich ein transienter Effekt vorliegt, die meiste Information also in diesem Peak verschlüsselt ist, sollte ausführlicher untersucht werden.

6.2.2 Multineuronale Kodierung

Die Ergebnisse mit $D^{multi}[q, k]$ geben auf Grund des geringen Stichprobenumfangs nur erste Hinweise auf zugrundeliegende Kodierungsstrategien. Der enorme Rechenaufwand (siehe Kapitel 6.1.1) erforderte die Verwendung einer geringeren als der zur Verfügung stehenden Anzahl an Spiketrains pro Stimulus. Die Abbildung 6.5 zeigt, dass die Ergebnisse sehr wahrscheinlich aber durchaus repräsentativ sind und mehr Stimuluswiederholungen nicht quantitativ bessere oder gar qualitativ andersartige Resultate liefern würde. So wurde für eine Ganglienzelle für unterschiedlich viele zufällig gewählten Spiketrains pro Geschwindigkeit zehnmal die Güte des stimulusabhängigen Clusters mit D^{spike} bestimmt und der Mittelwert zusammen mit dem Zweifachen des Standardfehlers in Abbildung 6.5

dargestellt. Dabei wurden von den gemittelten Werten der wechselseitigen Information je der entsprechende erste Summand nach Gleichung 3.2 abgezogen. Hiernach werden im Mittel mit einer Stimulusklassengröße von 100 sowohl mit Prozent-Korrekt als auch wechselseitiger Information nahezu die gleichen Ergebnisse erzielt wie mit der vorhandenen Klassengröße von 560. Das Verhältnis der Anzahl an Stimulusklassen zu ihrer jeweiligen Größe sollte aber nicht größer werden, da dann die Güte des stimulusabhängigen Clusters abnimmt, auch wenn sie noch die gleiche Abhängigkeit von q zeigt.

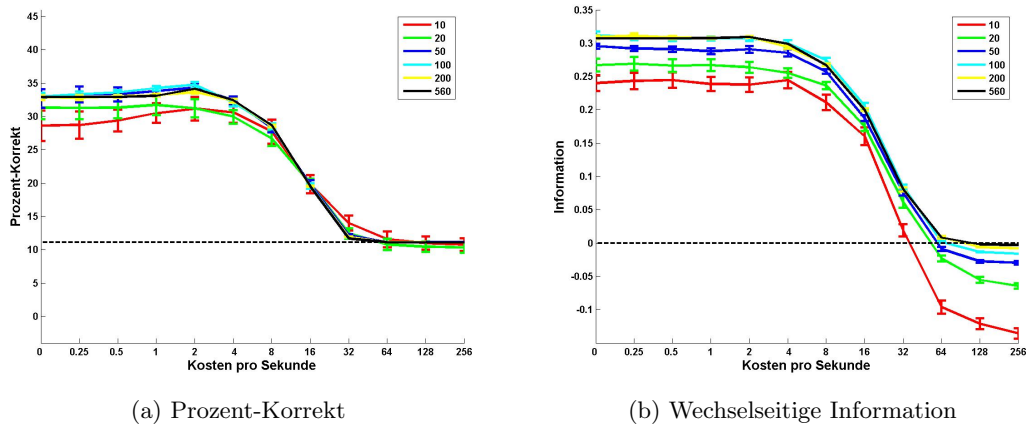


Abbildung 6.5: Güte des stimulusabhängigen Clusters einer Zelle (Daten14, Elektrode 26 Prototyp 3) für unterschiedlich große Stimulusklassen. Mittelung über zehnfaches zufälliges Herausgreifen der jeweiligen Anzahl an Spiketrains pro Stimulus. Balken zeigen zweifaches des Standardfehlers an. 6.5b Der erste Summand nach Gleichung 3.2 wurde von der gemittelten wechselseitigen Information abgezogen. Die wechselseitige Information ist mit $\log_2 9$ normiert.

Es sei angemerkt, dass diese Resultate nahelegen, dass die Ergebnisse für die Geschwindigkeitsübergänge wahrscheinlich zu qualitativ richtigen Schlüssen führen. Ausserdem lassen sie vermuten, dass eine Erhöhung der Anzahl an Stimuluspräsentationen die Güte des stimulusabhängigen Clusters bei Betrachtung der Geschwindigkeitsübergänge quantitativ verbessern könnte.

Mit $D^{multi}[q, k]$ konnte der gleiche Bereich zeitlicher Auflösung, bei der die Güte des stimulusabhängigen Clusters am besten ist, wie mit $D^{spike}[q]$ ermittelt werden. Über die Lokaltät einer Populationsmittelung und eine bessere Unterscheidbarkeit von Stimuli lässt sich auf Grund des Redundanzindex und seiner Abhängigkeit vom Parameter k vermuten, dass zwei Ganglienzellen mit der gleichen Richtungssensitivität zusammen kaum mehr Information über einen Reiz verschlüsseln als in den Spiketrains einer der beiden Zellen maximal enthalten ist. Da hiervon nur zwei Kombinationen betrachtet wurden, müssen natürlich weitere Untersuchungen erfolgen. Dabei sollten dann auch Zellen gewählt werden, deren Tuningkurven hinsichtlich der mittleren Spikeanzahl pro Stimulus nicht so ähnlich sind wie in dem hier betrachteten Beispiel. Die Unterscheidbarkeit der Geschwindigkeiten kann dagegen verbessert werden, wenn die Information über den Stimulus aus Spiketrains einer symmetrischen und einer richtungssensitiven Zelle gewonnen wird und dabei auch unterschieden wird, welcher Spike von welchem Neuron stammt. Es sollte auch noch die

Kombinationen zweier Zellen analysiert werden, die für unterschiedliche Richtungen sensitiv sind.

Die bisherigen Befunde sind in Einklang mit denen von Thiel et al. in [Th07]. Hier wurde gezeigt, dass eine Kombination der beiden Gruppen richtungssensitiver und nicht-richtungssensitiver Zellen zu einer präziseren Rekonstruktion der Geschwindigkeit, also Richtung und absolute Geschwindigkeit, führte.

Eine Analyse mit $D^{multi}[q, k]$ bei der als Stimuli die verschiedenen Geschwindigkeitsübergänge betrachtet werden, wäre sicherlich auch interessant, erscheint mit den vorhandenen Daten aber hinsichtlich des Rechenaufwands kaum durchführbar.

6.3 Schlusswort

Eine zuverlässige Schätzung über die Richtung, absolute Geschwindigkeit und Beschleunigung eines bewegten Gegenstandes ist wichtig, um die Augenbewegungen so fein abzustimmen, dass das Objekt auf der Retina fixiert werden kann. Es konnte in Einklang mit den Resultaten in [Th07] und [W07] gezeigt werden, dass Information hierüber in den Spike-trains retinaler Ganglienzellen einer Schildkröte verschlüsselt ist. Die Information über die Geschwindigkeit, inklusive der Richtung, ist nach dem verwendeten Metrik-basierten Verfahren in der Anzahl der Spikes verschlüsselt. Dagegen legen die Ergebnisse nahe, dass die Information über die Geschwindigkeitsübergänge, also die Beschleunigung, sowohl in den Spikezeitpunkten als auch den Interspikeintervallen auf einer feineren Zeitskala kodiert werden. Für das Zusammenwirken zweier Ganglienzellen scheint eine unterschiedliche Richtungssensitivität und eine Unterscheidung der Spikes nach Neuron die Information über die Geschwindigkeit eines Objekts zu erhöhen.

Mit Hilfe des Metrik-basierten Verfahrens konnten also Aufschlüsse über zugrundeliegende Kodierungsstrategien gewonnen werden, auch wenn sich dieses hinsichtlich der Rechenzeit und des Bias der wechselseitigen Information als nicht unproblematisch erwies. Dabei liegen die Ergebnisse für die Güte des stimulusabhängigen Clusters quantitativ im Bereich bisheriger Anwendungen (zum Beispiel [VP96], [SB04]).

Weitere Untersuchungen zeigen sich durchaus als empfehlenswert. So sollte zum einen der Stichprobenumfang bei den multineuronalen Betrachtungen erhöht werden. Zum anderen sollte ein weiteres Experiment erfolgen und analysiert werden, bei dem die jeweiligen Geschwindigkeitsübergänge öfter vorkommen, um so unbestreitbarere und präzisere Angaben über eine optimale zeitliche Auflösung machen zu können. Da eine schnelle Informationsübertragung von besonderem Interesse ist, sollte auch der Vermutung eines transienten Zeitkodes für die Beschleunigung nachgegangen werden.

A Anhang

A.1 Beschreibung der Matlab-Programme

A.1.1 Datenaufbereitung

Zunächst werden die Funktionen und Skripte aufgeführt, um die Daten, wie sie in Kapitel 5 verwendet werden, aus einer txt-Datei in eine Struktur in *Matlab* umzuwandeln.

```
% function einlesen(daten,name,lat,reiz,anzd,anzst)
% -----
%
% Dieses Programm dient dazu einen Datensatz 'daten' (txt. - Format)
% einzulesen und unter dem Dateinamen 'name' abzuspeichern.
%
% -----Eingabeargumente-----
% daten: Datensatz, der in folgender Spaltenform vorliegt: Eingerahmt von
%       Rauten steht zuerst die Angabe der Wiederholung (etwa s=1), dann
%       Elektrodennummer(e=13) und dann Nummer des Prototypen an dieser
%       Elektrode (p=2) und dann die beim Versuch gewonnenen Zeitpunkte
%       der Aktionspotentiale. Zwischen den Daten der verschiedenen Zellen
%       steht also wiederum mindestens eine Raute. In zwei aufeinander-
%       folgenden Zeilen sollten Rauten stehen, wenn sich die Nummer der
%       Wiederholung ändert.
% name: Name, unter dem der eingelesene Datensatz abgespeichert wird.
% lat: Zeit in Sekunden (Latenz), um die die einzulesenden Zeitpunkte
%      verschoben werden sollen [Daten02: 53 ms, Daten10: 54 ms,
%      Daten14: 56 ms]
% reiz: Matrix, in der im (s,n)-ten Eintrag der Startzeitpunkt in Sekunden
%      des s-ten Stimulus (z.B. Geschwindigkeit) im n-ten Durchlauf einer
%      beliebigen Wiederholung/Stimulussequenz steht
% anzd: Anzahl Durchlaeufe/Praesentationen eines Stimulus innerhalb einer
%      Stimulussequenz
% anzst: Anzahl Stimuli (z.B. Geschwindigkeiten)
%
% -----Ausgabeargumente-----
% Unter neuro(nr).p(j).st(s).w(i*d-d+n).ts befinden sich dann die
% Spike-Zeitpunkte des j-ten Prototypens an Elektrode nr des n-ten von d
% Durchlaeufen der i-ten Wiederholung/Replikation für den Stimulus s.
```

Die Matrix 'reiz' die für die Funktion *einlesen* benötigt wird, wird mit den folgenden beiden Skripten für die Geschwindigkeiten als Stimuli und die Geschwindigkeitsübergänge als Stimuli erstellt.

```
% reiz_geschw
% -----
%
% Erstellen der Matrix 'reiz', die im (s,n)-ten Eintrag die Startzeitpunkte
% in Sekunden der s-ten Geschwindigkeit für den n-ten von 80
% Durchlaefen/Praesentationen enthaelt. Diese Matrix wird unter 'reiz'
% abgespeichert.
```

```
% reiz_beschl
% -----
%
% Erstellen der Matrix 'reiz', die im (s,n)-ten Eintrag die Startzeitpunkte
% in Sekunden der s-ten Beschleunigung/Geschwindigkeitsuebergang fuer den
% n-ten von 10 Durchlaefen/Praesentationen enthaelt. Die Matrix 'reiz'
% wird abgespeichert unter dem Namen 'reiz_beschl'.
```

Im Folgenden sind alle *Matlab*-Funktionen und -Skripte geschrieben für Strukturen, wie sie mit dem Programm *einlesen* festgesetzt werden.

A.1.2 Charakterisierung des Antwortverhaltens

Stabilität des Antwortverhalten

Für die Datenselektion auf Basis der Stabilität des Antwortverhaltens in Kapitel 5.1 wurde folgende *Matlab*-Funktion genutzt.

```
% function nett(daten,anzst,anzr,anzp,stv)
% -----
%
% Dieses Programm soll helfen "nette" Neurone, also mit moeglichst
% stabilem Antwortverhalten, aus einem Datensatz 'daten' mittels
% Durchklicken herauszusuchen und gleichzeitig ihr Antwortverhalten zu
% charakterisieren.
% Dazu werden fuer jede Zelle und jede der 'anzst' Stimuli die
% standardisierten mittleren Spikezahlen der 'anzr' Replikate/Wieder-
% holungen einer Stimulussequenz ermittelt und grafisch dargestellt.
% Ausserdem wird die mittlere Anzahl Spikes pro Stimulus veranschaulicht
% (Tuningkurve).
%
% -----Eingabeargumente-----
```

```

% daten: Datensatz, der durchgesehen werden soll
% anzst: Anzahl Stimuli (d.h. 9 bei den Geschwindigkeiten)
% anzr: Anzahl der Wiederholungen/Replikate (d.h. 7 bzw. 8 bei den
%       Geschwindigkeiten)
% anzp: Anzahl der Praesentationen eines Stimulus innerhalb einer
%       Wiederholung (d.h. 80 bei den Geschwindigen)
% stv: Vektor, in dem Bezeichnungen (etwa 2.5 fuer 2.5 mm/s) der Stimuli
%       von 1 bis 'anzst' stehen

```

Histogramme

Mit folgenden Programmen können Histogramme der Spikeanzahlen, der Spikezeitpunkte und der Interspikeintervalle erstellt werden.

```

% function histoo(daten,elek,proto,anzst)
% -----
%
% Dieses Programm dient der Erstellung von Histogrammen der
% Spikeanzahlen ueber alle Wiederholungen/Praesentationen (d.h 640 bzw. 560
% bei den Geschwindigkeiten fuer jeden der 'anzst' Stimuli.
%
% -----Eingabeargumente-----
% daten: Datensatz, in dem die Antworten abgespeichert sind
% elek: Elektrodennummer
% proto: Prototypnummer
% anzst: Anzahl Stimuli

```

```

% function histts(daten,elek,proto,laeng, anzst)
% -----
%
% Dieses Programm dient der Erstellung von Histogrammen der Spikezeit-
% punkten ueber alle Wiederholungen/Praesentationen fuer jeden der 'anzst'
% Stimuli.
%
% -----Eingabeargumente-----
% daten: Datensatz, unter dem die Spiketrains abgespeichert sind
% elek: Elektrodennummer
% proto: Prototypnummer
% laeng: Laenge jedes einzelnen Stimulus in Sekunden (d.h. 0.5 bei den
%       Geschwindigkeiten)
% anzst: Anzahl Stimuli

```

```

% function histISI(daten,elek,proto,laeng,anzst)

```

```

% -----
%
% Dieses Programm dient der Erstellung von Histogrammen der Interspike-
% intervalle (ISI's) ueber alle Wiederholungen/Praesentationen fuer jeden
% der 'anzst' Stimuli.
%
% -----Eingabeargumente-----
% daten: Datensatz, unter dem die Spiketrains abgespeichert sind
% elek: Elektrodennummer
% proto: Prototypnummer
% laeng: Laenge jedes einzelnen Stimulus in Sekunden
% anzst: Anzahl Stimuli

```

A.1.3 Erzeugung künstlicher Daten

Die folgenden drei Funktionen dienen der Simulation von Spiketrains wie sie in Kapitel 4.2 (vgl. [VP97]) verwendet wurden.

```

% function simulation_h(anzst,anzw,k,laeng)
% -----
%
% Dieses Programm dient der Simulation von 'anzst'*'anzw' verschiedenen
% Spiketrains mit Spikezeitpunkten in Sekunden mit einer Laenge von 'laeng'
% Sekunden, so dass deren Summe der Spikeanzahlen einen homogenen Poisson-
% Prozess mit Rate 2*s fuer den s-ten von 'anzst' Stimuli bilden
% (siehe Victor 1997).
%
% -----Eingabeargumente-----
% anzst: Anzahl Stimuli
% anzw: Anzahl Wiederholungen
% k: Zahl fuer Namen (Abspeichern der Spiketrains unter simvic1_k)
% laeng: simulierte Stimuluslaenge
%
% -----Ausgabeargumente-----
% Spiketrains werden unter dem Namen 'simvic1_k' abgespeichert. Unter der
% Struktur neuro(1).p(1).st(s).w(i).ts befinden sich dann die
% Spikezeitpunkte der Antwort auf die i-te Präsntation des s-ten Stimulus.

% function simulation_it(anzst,anzw,n,laeng)
% -----
%
% Dieses Programm dient der Simulation von 'anzw'*'anzst' verschiedenen
% Spiketrains mit Spikezeitpunkten in Sekunden mit einer Laenge von 'laeng'
% Sekunden, so dass deren Summe der Spikeanzahlen einen iterierten Poisson-
% Prozess der Ordnung k=64 mit Rate 2*s fuer den s-ten von 'anzst' Stimuli
% nach Victor et al. 1997 bilden.

```

```

%
% -----Eingabeargumente-----
% anzst: Anzahl Stimuli
% anzw:  Anzahl Wiederholungen
% n:     Zahl fuer Namen (Abspeichern der Spiketrains unter simvic1_n)
% laeng: simulierte Stimuluslaenge
%
% -----Ausgabeargumente-----
% Spiketrains werden unter dem Namen 'simvic2_n' abgespeichert. Unter der
% Struktur neuro(1).p(1).st(s).w(i).ts befinden sich dann die
% Spikezeitpunkte der Antwort auf die i-te Präsensation des s-ten Stimulus.

```

```

% function simulation_inh(anzw,f,k)
% -----
%
% Dieses Programm dient der Simulation von 4*'anzw' Spiketrains mit
% Spikezeitpunkten in Sekunden und einer Laenge von
% einer Sekunde, so dass deren Summe der Spikeanzahlen einen inhomogenen
% Poisson-Prozess mit Rate  $20*(1+0.5*\cos(2*\pi*f*t+\phi))$  bilden. Dabei wird
% fuer die vier Stimulusklassen 'phi' auf jeweils 0, pi/2, pi und 3*pi/2
% gesetzt (siehe Victor et al. 1997).
%
% -----Eingabeargumente-----
% anzw: Anzahl Wiederholungen eines Stimulus
% f:    Frequenz (in Hertz)
% k:    Zahl fuer Namen (Abspeichern der Spiketrains unter simvic1_k)
%
% -----Ausgabeargumente-----
% Spiketrains werden unter dem Namen 'simvic3_k' abgespeichert. Unter der
% Struktur neuro(1).p(1).st(s).w(i).ts befinden sich dann die
% Spikezeitpunkte der Antwort auf die i-te Präsensation des s-ten Stimulus.

```

Die in Kapitel 5.2.1 verwendeten künstlichen Daten basieren auf folgenden zwei *Matlab*-Funktionen.

```

% function simulation_simdat_h(anzst,anzw,k,laeng)
% -----
%
% Simulation von anzst*anzw verschiedenen Spiketrains mit Spikezeitpunkten
% in Sekunden mit einer Laenge von laeng Sekunden, so dass deren Summe der
% Spikeanzahlen einen homogenen Poisson-Prozess mit Rate a(s) fuer den
% s-ten von 'anzst' Stimuli und a=[11.6 8.8 6 3.2 0.4 3.2 6 8.8 11.6]
% bilden.
%
% -----Eingabeargumente-----
% anzst: Anzahl Stimuli

```

```

% anzw: Anzahl Wiederholungen
% k:     Zahl fuer Namen (Abspeichern der Spiketrains unter
%       simdath_k_anzw640)
% laeng: simulierte Stimuluslaenge
%
% -----Ausgabeargumente-----
% Spiketrains werden unter dem Namen 'simdath_k_anzw640' abgespeichert.
% Unter der Struktur neuro(1).p(1).st(s).w(i).ts befinden sich dann die
% Spikezeitpunkte der Antwort auf die i-te Praesentation des s-ten
% Stimulus.

% function simulation_simdat_inh(anzw,k)
% -----
%
% Dieses Programm dient der Simulation von 9*'anzw' Spiketrains mit
% Spikezeitpunkten in Sekunden und einer Laenge von einer halben Sekunde,
% so dass deren Summe der Spikeanzahlen einen inhomogenen Poisson-Prozess
% mit Rate  $\lambda(u) = h \cdot \alpha^\beta / \Gamma(\beta) \cdot u^{(\beta-1)} \cdot$ 
%  $\exp(-\alpha \cdot u) + v \cdot u$  bilden. Die Setzungen fuer die neun Stimulusklassen
% sind dem Quellcode zu entnehmen.
%
% -----Eingabeargumente-----
% anzw: Anzahl Wiederholungen
% k:     Zahl fuer Namen (Abspeichern der Spiketrains unter
%       simdat_k_anzw640)
%
% -----Ausgabeargumente-----
% Spiketrains werden unter dem Namen 'simdat_k_anzw640' abgespeichert.
% Unter der Struktur neuro(1).p(1).st(s).w(i).ts befinden sich dann die
% Spikezeitpunkte der Antwort auf die i-te Präsntation des s-ten Stimulus.

```

A.1.4 Abstandsberechnung

Die in den vorhergehenden Kapiteln verwendeten Abstände zweier (gekennzeichnet) Spiketrains wurden mit folgenden Programmen berechnet.

```

% function dist=Dspike(q,neuron1,neuron2)
% -----
%
% Dieses Programm dient der Berechnung des Abstands  $D^{(spike)}[q]$  (nach
% Victor J.D., Purpura K.P.: "Metric-space analysis of spike trains:
% theory, algorithms and application." Network: Comput. Neural Syst. 1997,
% 8:127-164 fuer  $0 < q < \infty$  zwischen zwei Spiketrains.
%
% -----Eingabeargumente-----
% q:     Parameter in  $\text{sec}^{-1}$  (zeitliche Aufloesung)

```

```

% neuron1: erster Spiketrain
% neuron2: zweiter Spiketrain
%
% -----Ausgabeargumente-----
% dist: Abstand  $D^{\text{spike}}[q]$  (neuron1,neuron2)

% function dist=Dspike_k(l,q,neuron1, neuron2)
% -----
%
% Dieses Programm dient der Berechnung des Abstands  $D^{\text{spike}}[q]$  (nach
% Victor J.D., Purpura K.P.: "Metric-space analysis of spike trains:
% theory,algorithms and application." Network: Comput. Neural Syst. 1997,
% 8:127-164 fuer  $0 \leq q \leq \infty$  zwischen zwei Spiketrains neuron1 und
% neuron2, von denen je nur die Spikezeitpunkte kleiner oder gleich dem
% Zeitpunkt 'l' betrachtet werden.
%
% -----Eingabeargumente-----
% l:      Zeitpunkt in Sekunden
% q:      Parameter in  $\text{sec}^{-1}$  (zeitliche Aufloesung)
% neuron1: erster Spiketrain
% neuron2: zweiter Spiketrain
%
% -----Ausgabeargumente-----
% dist: Abstand  $D^{\text{spike}}[q]$  bei Betrachtung der Spikezeitpunkte von
%      neuron1 und neuron2 kleiner oder gleich dem Zeitpunkt 'l'

% function dist=Dinterval(q,neuron1, neuron2,T1,T2)
% -----
%
% Dieses Programm dient der Berechnung des Abstands  $D^{\text{interval}}[q]$  fuer
%  $0 \leq q \leq \infty$  zwischen zwei Spiketrains.
% Dabei wird  $D^{\text{interval:fix}}[q]$  nach Victor J.D., Purpura K.P.: "Metric-
% space analysis of spike trains: theory, algorithms and application."
% Network: Comput. Neural Syst. 1997, 8:127-164 berechnet. Das erste
% Interspikeintervall hat somit die Laenge T1-(Zeitpunkt des ersten Spikes)
% und das letzte die Laenge T2-(Zeitpunkt des letzten Spikes)
%
% -----Eingabeargumente-----
% q:      Parameter in  $\text{Sekunden}^{-1}$  (zeitliche Aufloesung)
% neuron1: erster Spiketrain
% neuron2: zweiter Spiketrain
% T1:     Anfangsspikezeitpunkt in Sekunden
% T2:     Endspikezeitpunkt in Sekunden
%

```



```

% -----Ausgabeargumente-----
% dist: Abstand  $D^{\text{interval}}[q]$  (neuron1,neuron2)

% function dist=Dinterval2(q,neuron1, neuron2)
% -----
%
% Dieses Programm dient der Berechnung des Abstands  $D(\text{interval})[q]$  fuer
%  $0 \leq q < \infty$  zwischen zwei Spiketrains neuron1 und neuron2.
% Dabei entspricht der Zeitpunkt des ersten Spikes gerade der Laenge des
% ersten Interspikeintervalls und das letzte Interspikeintervall hat gerade
% die Laenge (Zeitpunkt des letzten Spikes - Zeitpunkt des vorletzten
% Spikes)
%
% -----Eingabeargumente-----
% q:      Parameter in  $\text{sec}^{-1}$ 
% neuron1: erster Spiketrain
% neuron2: zweiter Spiketrain
%
% -----Ausgabeargumente-----
% dist: Abstand  $D^{\text{interval2}}[q]$  (neuron1,neuron2)

% function dist=Dinterval3(q,neuron1, neuron2,T)
% -----
%
% Dieses Programm dient der Berechnung des Abstands  $D(\text{interval})[q]$  für
%  $0 \leq q < \infty$  zwischen zwei Spiketrains neuron1 und neuron2.
% Dabei hat das ersten Interspikeintervall die Laenge (Zeitpunkt des zweiten
% Spikes- Zeitpunkt des ersten Spikes)und das letzte Interspikeintervall
% hat die Laenge (T-Zeitpunkt des letzten Spikes)
%
% -----Eingabeargumente-----
% q:      Parameter in  $\text{sec}^{-1}$ 
% neuron1: erster Spiketrain
% neuron2: zweiter Spiketrain
% T:      Endspikezeitpunkt in Sekunden
%
% -----Ausgabeargumente-----
% dist: Abstand  $D^{\text{interval2}}[q]$  (neuron1,neuron2)

% function dist=DistanzE(tau,neuro1,neuro2,laeng)
% -----
%
% Dieses Programm dient dazu, den Abstand zwischen zwei Spiketrains nach
% Kretzberg J., Warzecha A.-K., Egelhaaf M.: "Neural Coding with

```

```

% Graded Membrane Potential Changes and Spikes" Journal of Computational
% Neuroscience 2001, 11:153-164 zu berechnen.
% Filterung: Rechteckfilter der Laenge 'tau' wird um 0.1 ms verschoben;
% mittlere Spikeanzahlen pro 0.1 ms und nicht pro 1 sec und ausserdem
% keine Nomierung mit Anzahl Datenpunkte wie in Kretzberg et al.
%
% -----Eingabeargumente-----
% tau:    Zeitfenstergroesse in ms * 10 des Rechteckkerns
% neuro1: erster Spiketrain
% neuro2: zweiter Spiketrain
% laeng:  Laenge je eines Stimulus in Sekunden
%
% -----Ausgabeargumente-----
% dist: Abstand DistanzE[tau] (neuron1,neuron2)

```

Der Algorithmus zur Abstandsberechnung mit $D^{multi}[q, k]$ wurde online unter http://www-users.med.cornell.edu/~jdvicto/labdist_fast.html (am 26.4.2007) entnommen. Es sei angemerkt, dass hier der Fall zweier leerer gekennzeichneteter Spiketrains nicht erfasst wird.

```

% LABDIST_FAST(SA,LA,SB,LB,Q,K)
%   Calculates the multi-unit metric distance between two spike trains
%   Uses a fast version of the algorithm
%   SA, SB - spike times on the two spike trains
%   LA, LB - spike labels (positive integers)
%   Q - timing precision parameter
%   K - label reassigning parameter
%
%   Dmitriy Aronov, 6/20/01

```

Die paarweisen Abstände wurden stets in Distanzmatrizen erfasst, wie hier beispielhaft für $D^{spike}[q]$ aufgeführt.

```

% function Distanzmatrix_Spi(daten,distanz,elek,proto,q,anzst,anzw)
% -----
%
% Dieses Programm dient der Erstellung einer Abstandsmatrix 'M' für das
% Abstandsmass  $D^{(spike)}[q]$ .
%
% -----Eingabeargumente-----
% daten:  Datensatz unter dem die Spiketrains abgespeichert sind
% distanz: Name, unter dem die Matrix abgespeichert wird
% elek:   Elektrodennummer
% proto:  Prototypnummer an dieser Elektrode
% q:      Parameter in  $\text{sec}^{-1}$  (zeitliche Aufloesung)
% anzst:  Anzahl Stimuli
% anzw:   Anzahl Stimuluspraesentationen/-wiederholungen insgesamt

```

```

% laeng:   Laenge je eines Stimulus
%
% -----Ausgabeargumente-----
% In dem Matrixeintrag  $M((i-1)*anzw+k,(j-1)*anzw+1)$  befindet sich dann der
% Abstand zwischen der Nervenzellantwort auf die k-te Praesentation des
% i-ten Stimulus und der Antwort auf die l-te Praesentation des j-ten
% Stimulus bei insgesamt anzw Praesentationen eines Stimulus.

```

Die Berechnung der Abstände mit vielen Werten für q beziehungsweise τ ermöglicht *alldistmat*.

```

% function alldistmat(daten, elek, proto,anzst,anzw,typ,laeng)
% -----
%
% Dieses Programm dient dazu fuer die verschiedenen Abstandsmaße
% ( $D^{(spike)}[q]$ ,  $D^{\{spike\_k\}}[q]$ ,  $D^{(interval)}[q]$ ,  $D^{(interval2)}[q]$ ,
%  $D^{(interval3)}[q]$ ,  $D^{(euklid)}[\tau]$ , das entspricht den Typen 'Spi',
% 'Spi_k', 'Int', 'Int2', 'Int3', 'E') eine ausgewaehlte Reihe an Distanz-
% matrizen zu berechnen.
%
% -----Eingabeargumente-----
% daten: Datensatz, in dem die Spiketrains abgespeichert sind
% elek:  Elektrodennummer
% proto: Prototyp an Elektrodennummer elek
% anzst: Anzahl Stimuli
% anzw:  Anzahl Wiederholungen/Praesentationen eines Stimulus insgesamt
% typ:   Art des Abstandmaßes, für das die Distanzen berechnet werden
%        sollen (Spi, Spi_k, Int, Int2, Int3 oder E)
% laeng: Laenge je eines Stimulus

```

Für gekennzeichnete Spiketrains, die auf den Aktionspotentialen von zwei Nervenzellen beruhen, wurde folgendes Programm zur Erstellung der Distanzmatrizen verwendet,

```

% function Distanzmatrix_Multi2(daten,distanz,elek1,proto1,elek2,proto2,q,
%                               pk,anzst,anzw,wdhlg)
% -----
%
% Dieses Programm dient der Erstellung einer Abstandsmatrix 'M' fuer das
% Abstandsmass  $D^{(multi)}[q,k]$ .
%
% -----Eingabeargumente-----
% daten:  Datensatz unter dem die Spiketrains abgespeichert sind
% distanz: Name, unter dem die Matrix abgespeichert wird
% elek:   Elektrodennummer
% proto:  Prototypnummer an dieser Elektrode
% q:     Parameter in  $sec^{-1}$  (zeitliche Aufloesung)

```

```

% anzst:   Anzahl Stimuli
% anzw:   Anzahl Stimuluspraesentationen/-wiederholungen insgesamt
% laeng:  Laenge je eines Stimulus
% wdhlg:  Dateiname, unter dem eine Matrix 'A' vom Format 'anzst' x 'anzw'
%         abgespeichert ist. Diese enthaelt in der j-ten Zeile die Wieder-
%         holungen, die vom j-ten Stimulus betrachtet werden sollen.
%
% -----Ausgabeargumente-----
% In dem Matrixeintrag M((i-1)*anzw+k,(j-1)*anzw+1) befindet sich dann der
% Abstand zwischen der Nervenzellantwort auf die A(i,k)-te Repraesentation
% des i-ten Stimulus und der Antwort auf die l-te Repraesentation des
% A(j,l)-ten Stimulus bei insgesamt 'anzw' Repraesentationen eines Stimulus.

```

A.1.5 Stimulusabhängiges Clustern

Das stimulusabhängige Clustern wurde mit folgenden *Matlab*-Programmen durchgeführt.

```

% function [N,mit,Clu]=clustern(daten,anzst,anzw,z)
% -----
%
% Mit diesem Programm kann eine Distanzmatrix nach Victor et al. (1997)
% stimulusabhaengig geclustert und das Ergebnis mit Prozent-Korrekt und der
% wechselseitigen Information bewertet werden (siehe Victor 97).
% Ausserdem werden die Stimulus-/Antwortklassenmatrix 'N' und die
% Zuordnungen der Spiketrains in die Antwortklassen ermittelt.
%
% -----Eingabeargumente-----
% daten: Distanzmatrix, die zum Clustern herangezogen werden soll
% anzst: Anzahl der Stimuli
% anzw:  Anzahl der Wiederholungen/Praesentationen eines Stimulus insgesamt
% z:     Clusterexponent z
%
% -----Ausgabeargumente-----
% N:    Stimulus-/Antwortklassenmatrix, im (i,j)-ten Eintrag steht die
%       Anzahl an Spiketrains, die aus Stimulusklasse s_i in Antwort-
%       klasse r_j eingeteilt wurden
% mit:  Vektor, der die Bewertung des stimulusabhaengigen Clusters be-
%       inhaltet: mit(1) Prozent-Korrekt und mit(2) wechselseitige
%       Information
% Clu:  Struktur, in der die Zuordnung der jeweiligen Spiketrains in die
%       Antwortklassen steht

% function [N,mit,Clu]=clustern5(daten,anzst,anzw,z)
% -----
%

```

```

% Mit diesem Programm kann eine Distanzmatrix mit den neun Geschwindig-
% keiten als Stimuli nach Victor et al. (1997) stimulusabhaengig geclustert
% werden,wobei die Spiketrains, die durch die betragsmaeßig gleiche
% Geschwindigkeiten ausgeloeset wurden eine Stimulusklasse bilden. Die Guete
% des stimuluabhaengigen Clusters wird anschließend mit Prozent-Korrekt
% und der wechselseitigen Information bewertet (siehe Victor 1997).
% Ausserdem werden die Stimulus-/Antwortklassenmatrix 'N' und die
% Zuordnungen der Spiketrains in die Antwortklassen ermittelt.
%
% -----Eingabeargumente-----
% daten: Distanzmatrix, die zum Clustern herangezogen werden soll
% anzst: Anzahl der Stimuli
% anzw: Anzahl der Wiederholungen/Praesentationen eines Stimulus insgesamt
% z: Clusterexponent z
%
% -----Ausgabeargumente-----
% N: Stimulus-/Antwortklassenmatrix, im (i,j)-ten Eintrag steht die
% Anzahl an Spiketrains, die aus Stimulusklasse s_i in Antwortklasse
% r_j eingeteilt wurden
% mit: Vektor, der die Bewertung des stimulusabhaengigen Clusters bein-
% haltet: mit(1) Prozent-Korrekt und mit(2) wechselseitige Information
% Clu: Struktur, in der die Zuordnung der jeweiligen Spiketrains in die
% Antwortklassen steht

```

Analog zur Funktion *alldistmat* gibt es ein Programm für das stimulusabhängige Clustern einer Reihe von Distanzmatrizen.

```

% function mit=allclust(struc,anzst,anzw,z,name,typ,bild,laeng)
% -----
%
% Dieses Programm dient zum stimulusabhaengigen Clustern von Distanz-
% matrizen, die mit alldistmat erstellt worden sind. Die erhaltenen
% Stimulus-/Antwortklassen-Matrizen koennen auf Wunsch mit imagesc visuell
% dargestellt werden.
%
% -----Eingabeargumente-----
% struc: die allgemeine Struktur des Namen, unter der die Distanzmatrizen
% abgespeichert sind
% anzst: Anzahl Stimuli
% anzw: Anzahl Wiederholungen/Repraesentationen eines Stimulus insgesamt
% z: Clusterexponent
% name: Name unter dem die Stimulus-/Antwortklassenmatrizen, die
% Zuordnungen der Zellen und die Werte von Prozent-Korrekt und der
% wechselseitigen Information gespeichert werden
% typ: Art des Abstandsmasses, also 'Spi', 'Spi_k','Int','Int3','Int2'
% oder 'E'
% bild: boolean, 1 erzeugt Bild, 0 erzeugt kein Bild

```

```

% laeng: Stimuluslaenge
%
% -----Ausgabeargumente-----
% mit: Matrix, in der im (i,1)-ten Eintrag Prozent-Korrekt und im
%      (i,2)-ten Eintrag wechselseitige Information des stimulusabhaengigen
%      Clusters basierend auf der i-ten Distanzmatrix eingetragen sind

```

Bias

Das empirische Zufallslevel von Prozent-Korrekt und wechselseitiger Information, also ein empirisch abgeschätzter Bias der wechselseitigen Information bei Vorliegen stochastisch unabhängiger gleichverteilter Zufallsvariablen, wurde mit folgendem Programm ermittelt.

```

% function mit=allclust_bias(struc,anzst,anzw,name,typ,bild,laeng,mat)
% -----
%
% Dieses Programm dient dazu auf Basis von Distanzmatrizen, die mit dem
% Programm 'alldistmat' erstellt worden sind, das empirische Zufallslevel
% fuer Prozent-Korrekt und wechselseitige Information zu bestimmen, also
% den Bias der wechselseitigen Information bei Vorliegen stochastisch
% unabhangiger gleichverteilter Zufallsvariablen fuer die Cluster-
% exponenten -8, -2, 1, 2 und 8 nach Victor et al. (1997) abzuschatzen.
%
% -----Eingabeargumente-----
% struc: die allgemeine Struktur des Namens, unter der die Distanzmatrizen
%        abgespeichert sind
% anzst: Anzahl Stimuli
% anzw:  Anzahl Wiederholungen/Repraesentationen eines Stimulus insgesamt
% z:     Clusterexponent
% name:  Name unter dem die Stimulus-/Antwortklassenmatrizen, die
%        Zuordnungen der Zellen und die Werte von Prozent-Korrekt und der
%        wechselseitigen Information gespeichert werden
% typ:   Art des Abstandsmasses, also 'Spi', 'Spi_k', 'Int', 'Int3', 'Int2'
%        oder 'E'
% bild:  boolean, 1 erzeugt Bild, 0 erzeugt kein Bild
% laeng: Stimuluslaenge
% mat:   Matrix, die die Postionen der oberen Dreiecksmatrix der
%        Distanzmatrix enthalt
%
% -----Ausgabeargumente-----
% Matrizen fuer die fuef verschiedenen Clusterexponenten, in denen je
% im (i,1)-ten Eintrag Prozent-Korrekt und im (i,2)-ten Eintrag wechsel-
% seitige Information des stimulusabhaengigen Clusters basierend auf der
% i-ten Distanzmatrix eingetragen sind.

```

Eine analytische Abschätzung des Bias der wechselseitigen Information ermöglicht folgende Funktion.

```
% function erg=bias(K,N,t)
% -----
%
% Dieses Programm dient der Abschaetzung des Bias nach Miller 1955 bei
% Vorliegen gleichgroßer Stimulusklassen und gleicher Anzahl an Stimulus-
% und Antwortklassen.
%
% -----Eingabeargumente-----
% K: Anzahl der Stimulus- bzw. Antwortklassen
% N: Anzahl der Spiketrains insgesamt
% t: Belegung mit 1 oder 2. 1: Abschaetzung des Bias ueber ersten Summand
%   nach Miller 1955. 2: Abschaetzung des Bias bei Vorliegen stochastisch
%   unabhængiger gleichverteilter Zufallsvariablen ueber Summe der ersten
%   beiden Summanden nach Miller 1955.
%
% -----Ausgabeargumente-----
% erg: Biasabschaetzung
```

Bilder

Die Ergebnisse des stimulusabhängigen Clusters lassen sich visuell mit den folgenden beiden Funktionen veranschaulichen.

```
% bild_guete(proinfo,typ,anzst,name,z,laeng)
%
% Dieses Programm dient der grafischen Darstellung der mit Hilfe des
% Programms 'clustern' ermittelten und unter dem Namen 'proinfo' abge-
% speicherten Werte Prozent-Korrekt und wechselseitigen Information.
%
% -----Eingabeargumente-----
% protra: Name unter der eine Matrix, die die Werte Prozent-Korrekt
%         (mit(:,1)) und wechselseitige Information (mit(:,2)) enthaelt,
%         abgespeichert ist
% typ:    Art des Abstandsmaßes, also 'Spi', 'Spi_k', 'Int', 'Int2', 'Int3'
%         oder 'E'
% anzst:  Anzahl der Stimuli
% name:   Name zum Abspeichern des Bildes unter proinfo_'name'_'z'.jpg
% z:     Clusterexponent
% laeng:  Laenge je eines Stimulus in Sekunden

% function bild_multi(elek1,proto1,elek2,proto2,z)
% -----
```

```

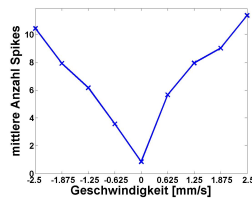
%
% Dieses Programm dient der visuellen Darstellung der Ergebnisse des
% stimulusabhaengigen Clusters mit  $D^{\text{multi}}[q,k]$  zweier Zellen aus
% Daten14 mit den Elektrodennummern 'elek1', 'elek2' und den zugehoerigen
% Prototypnummern 'proto1', 'proto2'. Ausserdem werden die Ergebnisse der
% einzelnen Zellen mit  $D^{\text{spike}}[q]$  und die Summe dieser abgebildet.
% Des Weiteren werden die Redundanzindizes nach Prozent-Korrekt und
% wechselseitiger Information berechnet.
%
% -----Eingabeargumente-----
% elek1: Elektrodennummer der ersten Zelle
% proto1: Prototypnummer der ersten Zelle
% elek2: Elektrodennummer der zweiten Zelle
% proto2: Prototypnummer der zweiten Zelle
%
% -----Ausgabeargumente-----
% Redundanzindizes nach Prozent-Korrekt und wechselseitiger Information
% werden abgespeichert.

```

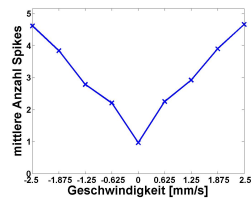

A.2 Tuningkurven der verwendeten Zellen

Im Folgenden sind die Tuningkurven aller verwendeten Zellen getrennt nach Datensatz abgebildet. Dabei sind jeweils die Elektrodennummer (E), an welcher die elektrischen Signale aufgezeichnet wurden, und der Prototyp (P), zu welchem diese sortiert wurden, angegeben. Ausserdem ist die entsprechende Zuordnung zu nicht-richtungssensitiv (NR), richtungssensitiv (R) und, als nicht eindeutig zuteilbar, sonstige (S) verzeichnet. Die Zelle, die bei Betrachtung aller Datensätze entnommen wurde, ist mit X markiert.

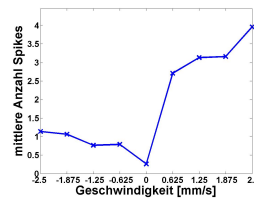
Daten02



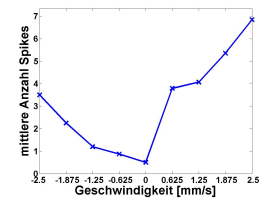
(a) E.14 P.3 (NR)



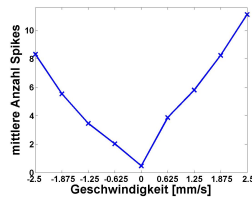
(b) E.15 P.2 (NR)



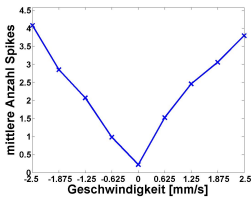
(c) E.16 P.1 (R)



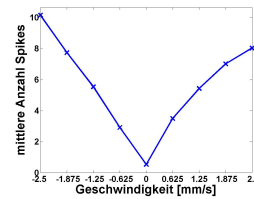
(d) E.19. P.1 (R)



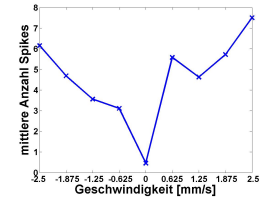
(e) E.19 P.3 (S)



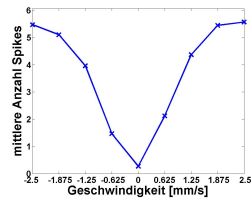
(f) E.22 P.1 (NR)



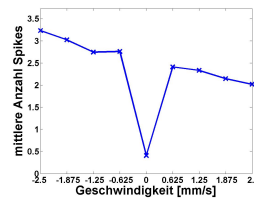
(g) E.22 P.2 (S)



(h) E.56 P.3 (S)

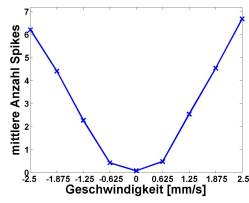


(i) E.83 P.4 (NR)

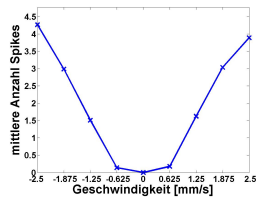


(j) E.87 P.1 (S)

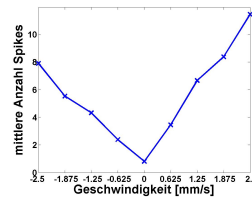
Daten10



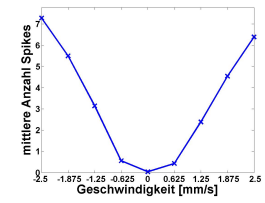
(a) E.26 P.3 (X)



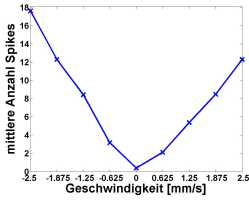
(b) E.52 P.1 (NR)



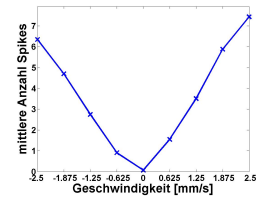
(c) E.52 P.3 (S)



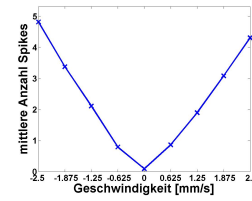
(d) E.59 P.1 (NR)



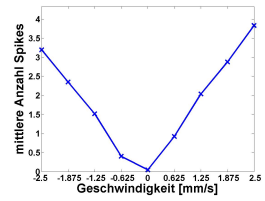
(e) E.59 P.2 (S)



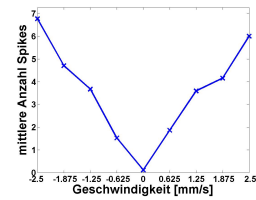
(f) E.63 P.1 (NR)



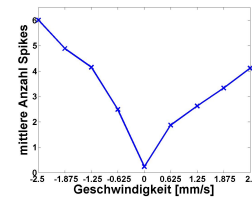
(g) E.63 P.2 (NR)



(h) E.64 P.2 (NR)

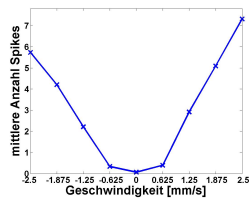


(i) E.64 P.3 (NR)

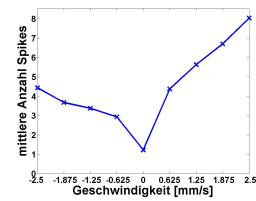


(j) E.65 P.3 (S)

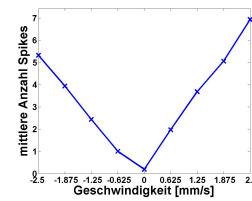
Daten14



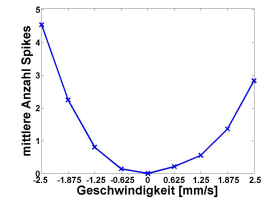
(a) E.26 P.3 (NR)



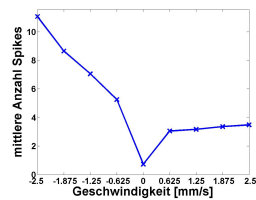
(b) E.27 P.3 (R)



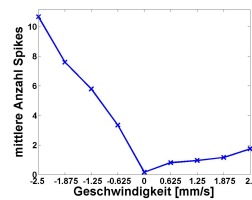
(c) E.35 P.1 (NR)



(d) E.44 P.1 (S)



(e) E.54 P.1 (R)



(f) E.56 P.2 (R)

Literaturverzeichnis

- [AMK95] Ammermüller J., Muller J.F., Kolb H.: *The organization of the turtle inner retina: II. Analysis of color-coded and directionally selective cells.* Journal of Comparative Neurology 1995, 358:35-62
- [A03] Aronov D.: *Fast algorithm for the metric-space analysis simultaneous responses of multiple single neurons.* Journal of Neuroscience Methods 2003, 124:175-179.
- [ARMV03] Aronov D., Reich D.S., Mechler F., Victor D.: *Neural Coding of Spatial Phase in V1 of the Macaque Monkey* Journal of Neurophysiology 2003, 89:3304-3327
- [AV04] Aronov D., Victor J.D.: *Non-Euclidean properties of spike train metric spaces.* Physical Review E Stat Nonlin Soft Matter Phys 2004, 69:061905.
- [B70] Blumenthal L.M.: *Theory and Application of Distance Geometry.* New York: Chelsea Publishing Company, 2nd Edition, 1970
- [C69] Carlton A.G.: *On the Bias of Information Estimates.* Psychological Bulletin, 1969: 71:108-109
- [CR05] Chichilnisky E.J., Rieke F.: *Detection Sensitivity and Temporal Resolution of Visual Signals near Absolute Threshold in the Salamander Retina.* The Journal of Neuroscience, 2005, 25(2):318-330
- [CT91] Cover T.M., Thomas, Joy A.: *Elements of Information Theory.* New York: J. Wiley and Sons, 1991
- [DA01] Dayan P., Abbott L.F.: *Theoretical neuroscience: computational and mathematical modeling of neural systems.* Massachusetts: The MIT Press, 2001
- [ES00] Ester M., Sander J.: *Knowledge Discovery in Databases: Techniken und Anwendungen* Berlin: Springer, 2000
- [G64] Gaal S.: *Point Set Topology.* New York: Academic Press, 1964.
- [GKWE03] Grewe J., Kretzberg J., Warzecha A.-K., Egelhaaf M.: *Impact of Photon Noise on the Reliability of a Motion-Sensitive Neuron in the Fly's Visual System.* Journal of Neuroscience 2003, 23(34):10776-10783
- [GB03] Grünbaum, B.: *Convex Polytopes.* New York: Springer, 2nd Edition, 2003
- [HKC83] Havel T.F., Kuntz I.D., Crippen G.M.: *The Theory and Practice of Distance Geometry.* Bulletin of Mathematical Biology, 1983, Vol.45, No.5: 665-720

- [H00] Herwig R.: *Large-scale information theoretic clustering with application to the analysis of genetic fingerprinting data*. Doctoral Dissertation. Berlin, 2000
- [HR05] Hohage T., Rapin G.: *Numerische Behandlung partieller Differentialgleichungen*. Skript, Universität Göttingen, Sommersemester 2005, verfügbar (am 10.8.2007) unter <http://www.num.math.uni-goettingen.de/hohage/numpde/einleitung.pdf>
- [KS99] Koch C., Segev I.: *Methods in neuronal modeling: from ions to network*. Massachusetts: The MIT Press, 2nd Edition, 1999.
- [K01] Kretzberg J.: *Reliability of Neural Coding. A Model Study*. Doctoral Dissertation. Bielefeld, 2001.
- [KWE01] Kretzberg J., Warzecha A.-K., Egelhaaf M.: *Neural Coding with Graded Membrane Potential Changes and Spikes*. *Journal of Computational Neuroscience* 2001, 11:153-164.
- [M83] Maehara, H.: *Regular Embeddings of a Graph*. *Pacific Journal of Mathematics* 1983, 107.2:393-402.
- [MB99] Meister M., Berry M.J.: *The Neural Code of the Retina*. *Neuron* 1999, 22:435-450
- [MS05] Meintrup D., Schäffler S.: *Stochastik. Theorie und Anwendung*. Berlin: Springer, 2005.
- [M55] Miller G.A.: Note on the bias of information estimates in Quastler H.: *Information Theory in Psychology. Problems and Methods*. Illinois: The Free Press, 1955
- [NMWF01] Nicholls J.G., Martin A.R., Wallace B.G., Fuchs P.A. *From Neuron To Brain*. Massachusetts: Sinauer Associates, 4th Edition, 2001
- [PT96] Panzeri S., Treves, A.: *Analytical estimates of limited sampling bias in different measures*. *Network: Computation in Neural Systems* 1996, 7:87-107
- [PS07] Pfeifer D., Straßburger D.: *Einführung in die Stochastik*. Skript zur Vorlesung, Universität Oldenburg, Stand 4.2.2007, verfügbar unter <http://www.mathematik.uni-oldenburg.de/personen/pfeifer/ESToch.pdf>
- [RMV01] Reich D.S., Mechler F., Victor J.D.: *Independent and redundant information in nearby cortical neurons*. *Science* 2001b, 294:2566-2568
- [RWRB97] Rieke F., Warland D., de Ruyter van Steveninck R., Bialek W.: *Spikes. Exploring the neural code*. Massachusetts: MIT Press, 1997.
- [RG55] Rogers M.S., Green B.F.: The Moments of Sample Information when the Alternatives are equally likely in Quastler H.: *Information Theory in Psychology. Problems and Methods.*, Illinois: The Free Press, 1955

- [SB04] Samonds J.M., Bonds A.B.: *From Another Angle: Differences in Cortical Coding Between Fine and Coarse Discrimination of Orientation*. Journal of Neurophysiology, 2004, 91:1193-1202
- [SS00] Schmidt, Schaible (Hrsg.) *Neuro- und Sinnesphysiologie*. Berlin: Springer Verlag, 4. Auflage, 2000.
- [Se74] Sellers P.H.: *On the Theory and Computation of Evolutionary Distances*. SIAM Journal on Applied Mathematics 1974, Vol.26, No.4:787-793.
- [Sp83] Späth H.: *Cluster-Formation und -Analyse: Theorie, FORTRAN-Programme, Beispiele*. München: Oldenbourg, 1983
- [Th07] Thiel A., Greschner M., Eurich C., Ammermüller J., Kretzberg J.: *Contribution of individual retinal ganglion cell responses to velocity and acceleration encoding*. Journal of Neurophysiology, in Druck.
- [Tr02] Trappenberg, Thomas P. *Fundamentals of Computational Neuroscience.*, New York: Oxford University Press, 2002
- [VR01] van Rossum M.: *A novel spike distance*. Neural Computation 2001, 13(4):751-763
- [VP96] Victor J.D., Purpura K.P.: *Nature and precision of temporal coding in visual cortex: a metric-space analysis*. Journal of Neurophysiology 1996, 76:1310-1326.
- [VP97] Victor J.D., Purpura K.P.: *Metric-space analysis of spike trains: theory, algorithms and application*. Network: Comput. Neural Syst. 1997, 8:127-164.
- [VP98] Victor J.D., Purpura K.P.: *Spatial Phase and the Temporal Structure of the Response to Gratings in V1*. Journal of Neurophysiology 1998, 80: 554-571
- [V99] Victor J.D.: *Temporal aspects of neural coding in retina and lateral geniculate*. Network: Comp. Neural Syst. 1999, 10:R1-R66.
- [V05] Victor J.D.: *Spike train metrics* Current Opinion in Neurobiology 2005, 15:585-592.
- [W07] Winzenborg I.: *Bayes'sche Schätztheorie und ihre Anwendung auf neuronale Daten zur Reizrekonstruktion* Diplomarbeit. Oldenburg, 2007

Erklärung

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe.

Oldenburg, den 23.08.2007