

Durchgeführt im Rahmen des

BMBF-Projekt: ELISE

Dokumentation der Projektarbeit

Entwurf eines kompakten mikrocontrollergestützten Systems zur
Emotionserkennung in einer Virtual-Reality-Umgebung

WiSe 2017/2018 und SoSe 2018

Projektbetreuer:

Medizinische Informatik und Mikrosystementwurf

Prof. Dr. rer. nat. Rainer Brück

Dr.-Ing. Armin Grünewald

M.Sc. David Krönert

Forschungsgruppe für Mustererkennung

Prof. Dr.-Ing. Marcin Grzegorzek

M.Sc. Frédéric Li

Projektteilnehmer:

Artur Piet (Sprecher der Projektgruppe)

Jonas Pöhler (Stellv. Sprecher der Projektgruppe)

Arnaud Eric Toham Waffo

Boris Kamdem

Kevin Orth

Meryem Dural

Minas Michail

Inhaltsverzeichnis

1	Einleitung	8
1.1	Hintergrund und Motivation	8
1.2	ELISE Projektbeschreibung	8
1.3	Gliederung dieser Dokumentation	8
1.4	Anhang	8
2	Organisation	9
3	Grundlagen	10
3.1	Definition von Emotionen	10
3.2	Virtual Reality (VR)	10
3.3	Sensoren und biophysiological Signale zur Emotionserkennung . .	10
3.3.1	Körpertemperatur-Sensor	10
3.3.2	Blood Volume Pulse-Sensor (BVP)	10
3.3.3	Messen der Sauerstoffsättigung (SpO2)	10
3.3.4	Galvanic Skin Response (GSR)	10
3.3.5	Elektroenzephalografie (EEG)	10
3.3.6	Elektrookulografie (EOG)	10
3.3.7	Analog/Digital-Wandler	10
3.4	Kommunikation	10
3.5	Grundlagen der Mustererkennung	10
3.6	Emotion Recognition Chain	12
3.6.1	Datenerfassung	12
3.6.2	Vorverarbeitung	12
3.6.3	Segmentation	13
3.6.4	Merkmalsextraktion	14
3.6.5	Klassifikation	14
4	State-of-the-Art Analyse	19
I	Erster Prototype	20
<hr/>		
5	Systementwurf und Konzept	20
5.1	Anforderungen	20
5.2	Konzept	20
5.3	Hardwareauswahl	20
5.3.1	Auswahlkriterien	20

5.3.2	Festlegung der genutzten Hardware	20
5.4	Hardwarearchitektur	20
5.4.1	GSR-Sensor	20
5.4.2	Temperatur-Senosr	20
5.4.3	Pulsoximeter	20
5.4.4	EEG	20
5.4.5	EOG	20
5.4.6	Datenübertragung	20
5.5	Programmierung	20
5.6	Aufnahme der übertragenen Daten	20
6	Realisierung	21
7	Emotionsinduktion	22
7.1	Ablauf	22
7.2	Fragebogen	22
7.3	Szenarien	22
7.3.1	Glück	22
7.3.2	Langeweile	22
7.3.3	Frustration	22
8	Messreihe	23
9	Mustererkennung	24
9.1	Merkmalsextraktion für Emotionserkennung	24
9.1.1	Hand-gefertigte Merkmale	24
9.1.2	Codebook Approach	24
10	Ergebnisse	25
10.1	Ergebnisse der hand-gefertigten Merkmale	25
10.2	Ergebnisse des Codebook Approach	25
10.3	Analyse der Ergebnisse	25
II	Zweiter Prototype	26
11	Systementwurf und Konzept	26
11.1	Anforderungen	26
11.2	Konzept	26
11.3	Hardwareauswahl	26
11.3.1	Auswahlkriterien	26

11.3.2	Festlegung der genutzten Hardware	26
11.4	Hardwarearchitektur	26
11.4.1	GSR-Sensor	26
11.4.2	Temperatur-Senosr	26
11.4.3	Pulsoximeter	26
11.4.4	EEG	26
11.4.5	EOG	26
11.4.6	Datenübertragung	26
11.5	Programmierung	26
11.6	Aufnahme der übertragenen Daten	26
12	Realisierung	27
13	Emotionsinduktion	28
13.1	Ablauf	28
13.2	Fragebogen	28
13.3	Szenarien	28
13.3.1	Glück	28
13.3.2	Langeweile	28
13.3.3	Frustration	28
14	Messreihe	29
15	Mustererkennung	30
15.1	Merkmalsextraktion für Emotionserkennung	30
15.1.1	Hand-gefertigte Merkmale	30
15.1.2	Codebook Approach	30
16	Ergebnisse	31
16.1	Ergebnisse der hand-gefertigten Merkmale	31
16.2	Ergebnisse des Codebook Approach	31
16.3	Analyse der Ergebnisse	31
III	Dritter Prototype	32
<hr/>		
17	Systementwurf und Konzept	32
17.1	Anforderungen	32
17.2	Konzept	32
17.3	Hardwareauswahl	32
17.3.1	Auswahlkriterien	32

17.3.2	Festlegung der genutzten Hardware	32
17.4	Hardwarearchitektur	32
17.4.1	GSR-Sensor	32
17.4.2	Temperatur-Senosr	32
17.4.3	Pulsoximeter	32
17.4.4	EEG	32
17.4.5	EOG	32
17.4.6	Datenübertragung	32
17.5	Programmierung	32
17.6	Aufnahme der übertragenen Daten	32
18	Realisierung	33
19	Emotionsinduktion	34
19.1	Ablauf	34
19.2	Fragebogen	34
19.3	Szenarien	34
19.3.1	Glück	34
19.3.2	Langeweile	34
19.3.3	Frustration	34
20	Messreihe	35
21	Mustererkennung	36
21.1	Merkmalsextraktion für Emotionserkennung	36
21.1.1	Hand-gefertigte Merkmale	36
21.1.2	Codebook Approach	36
22	Ergebnisse	37
22.1	Ergebnisse der hand-gefertigten Merkmale	37
22.2	Ergebnisse des Codebook Approach	37
22.3	Analyse der Ergebnisse	37
IV	Vierter Prototype	38
<hr/>		
23	Systementwurf und Konzept	38
23.1	Anforderungen	38
23.2	Konzept	38
23.3	Hardwareauswahl	38
23.3.1	Auswahlkriterien	38

23.3.2 Festlegung der genutzten Hardware	38
23.4 Hardwarearchitektur	38
23.4.1 GSR-Sensor	38
23.4.2 Temperatur-Senosr	38
23.4.3 Pulsoximeter	38
23.4.4 EEG	38
23.4.5 EOG	38
23.4.6 Datenübertragung	38
23.5 Programmierung	38
23.6 Aufnahme der übertragenen Daten	38
24 Realisierung	39
25 Emotionsinduktion	40
25.1 Ablauf	40
25.2 Fragebogen	40
25.3 Szenarien	40
25.3.1 Glück	40
25.3.2 Langeweile	40
25.3.3 Frustration	40
26 Messreihe	41
27 Mustererkennung	42
27.1 Merkmalsextraktion für Emotionserkennung	42
27.1.1 Hand-gefertigte Merkmale	42
27.1.2 Codebook Approach	42
28 Ergebnisse	43
28.1 Ergebnisse der hand-gefertigten Merkmale	43
28.2 Ergebnisse des Codebook Approach	43
28.3 Analyse der Ergebnisse	43
29 Zusammenfassung und Ausblick	44
29.1 Zusammenfassung	44
29.2 Fazit	44
29.3 Ausblick	44
Abbildungsverzeichnis	45
Tabellenverzeichnis	46

Abkürzungen	47
Anhang	48

1 Einleitung

Verantwortlich: Minas

- Next Step: Bitte selbst bei github hochladen oder Artur zuschicken

1.1 Hintergrund und Motivation

1.2 ELISE Projektbeschreibung

1.3 Gliederung dieser Dokumentation

1.4 Anhang

2 Organisation

Verantwortlich: Artur
- Next Step: Gliederung erstellen

3 Grundlagen

Verantwortlich: Arnaud

3.1 Definition von Emotionen

Verantwortlich: Arnaud

3.2 Virtual Reality (VR)

Verantwortlich: Arnaud, Boris

3.3 Sensoren und biophysiological Signale zur Emotionserkennung

Verantwortlich: Arnaud, Kevin

3.3.1 Körpertemperatur-Sensor

3.3.2 Blood Volume Pulse-Sensor (BVP)

3.3.3 Messen der Sauerstoffsättigung (SpO2)

3.3.4 Galvanic Skin Response (GSR)

3.3.5 Elektroenzephalografie (EEG)

3.3.6 Elektrookulografie (EOG)

3.3.7 Analog/Digital-Wandler

3.4 Kommunikation

Verantwortlich: Kevin, Jonas

3.5 Grundlagen der Mustererkennung

Verantwortlich: Artur

- Bereit zum Korrekturlesen.

Mustererkennung (enlg. "pattern recognition") ist ein Unterthema des maschinellen Lernens. Das Ziel besteht darin, automatisierte Systeme zu entwerfen, die hoch abstrakte Muster in Daten erkennen können. Konkret heißt dies, dass man Maschinen beibringen möchte komplexer Aufgaben zu lösen, welche vom Menschen nahezu mühelos und natürlich erledigt werden können. Typische Beispiele für die zahlreichen Anwendungsbereiche sind die Objekterkennung, Spracherkennung sowie die

Erkennung und Verfolgung in Bildern. Die Emotionserkennung ist ein Anwendungsbereich der Mustererkennung. Die Hauptidee hinter der Lösung eines Mustererkennungs-Problems ist es, dieses als Klassifikationsproblem zu übersetzen, wobei die zu erkennende Mustern die unterschiedliche Klassen bilden. Die vom Mustererkennungs-System eingegebenen Daten werden dann verarbeitet und der “am nächsten liegenden” Klasse zugeordnet. Beispielsweise können bei der Emotionserkennung die Eingangsdaten Bilder oder physiologische Signale sein, die in verschiedene Klassen eingeteilt werden, welche jeweils einer Emotion entsprechen.

Ein wichtiger Teil eines jeden Mustererkennung-Problems ist der Lernansatz, mit welchem die Maschine lernen soll die Muster in den Daten zu erkennen. Traditionell werden zwei Ansätze verwendet:

- Überwachter Lernansatz: Dieser Ansatz kann nur verwendet werden, wenn vor der Verarbeitung der Daten ein Datenbeschriftungsschritt durchgeführt wurde. In diesem Schritt wird jedem Element des Datensatzes ein Etikett (engl. “label”) zugewiesen, das angibt, welcher Klasse der jeweilige Datenpunkt zugeordnet werden kann. Die zusätzlichen Informationen, die die Etiketten liefern, werden als Grundlage verwendet, um sie mit der Vorhersage des Systems zu vergleichen und zu korrigieren, wenn sie nicht gleich sind.
- Unüberwachter Lernansatz: Dieser Ansatz wird verwendet, wenn keine Etiketten für die Daten vorhanden sind. Unüberwachte Lerntechniken zielen darauf ab, der Maschine beizubringen, Muster in den Daten selbst zu finden. Sie werden meist verwendet, um Einblicke in Daten zu erhalten, deren Struktur unbekannt ist.

Überwachtes Lernen liefert aktuell weit bessere Ergebnisse, jedoch ist die Beschriftung mit Etiketten der Daten nicht immer einfach oder teilweise sogar gar nicht möglich (z.B. wenn die Datenmenge sehr groß ist oder wenn Unsicherheit über die Vergabe der Etiketete besteht). Aus diesem Grund wächst das Interesse an unüberwachten Lernansätzen. Diese Ansätze sind jedoch schwierig zu benutzen, da sie eine große Menge an Daten voraussetzen. Kompromisse sind mit semi-überwachten Lernansätzen möglich, bei denen die Daten für einen Teil des Datensatzes (aber nicht für den ganzen Datensatz) mit Etiketten beschriftet und damit bekannt sind. In diesem Fall kann eine Mischung aus überwachten und Unüberwachten Techniken angewendet werden [1].

Im Rahmen des ELISE-Projekts werden mit Hilfe von Mustererkennungsverfahren eindimensionale Zeitsignale von physiologischen Sensoren in Echtzeit für die Erkennung von drei Emotionen verarbeitet: Glück, Frustration und Langeweile. Um den Emotionsklassifizierer aufzubauen, wird ein standardmäßiger, überwachter Ler-

ansatz namens Emotion Recognition Chain verwendet, der im folgendem Kapitel beschrieben wird.

3.6 Emotion Recognition Chain

Verantwortlich: Artur
- ERC Bild ändern (5 statt 4 Schritte)

Die Emotion Recognition Chain (ERC) besteht aus fünf Hauptschritten: Datenerfassung, Vorverarbeitung, Segmentierung, Merkmalsextraktion und Klassifizierung (vgl. Abb. 1). In den folgenden Unterkapiteln wird für jeden Schritt eine allgemeine Erklärung gegeben.

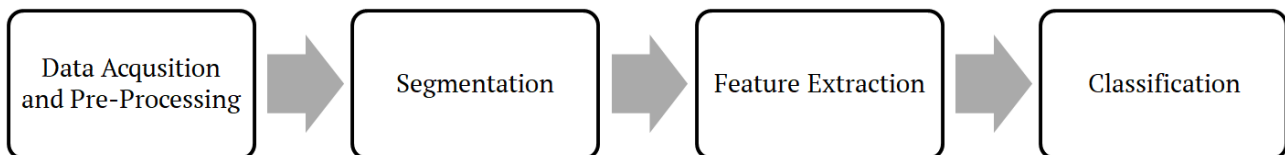


Abbildung 1: Emotion Recognition Chain: Zeitreihen-Datensätze werden von tragbaren Sensoren aufgenommen (Datenerfassung) und vorverarbeitet (Vorverarbeitung). Die Daten werden dann in Segmente unterteilt (Segmentierung), aus denen Merkmale extrahiert werden (Merkmalsextraktion). Mit den gewonnenen Merkmalen wird schließlich ein Klassifikator trainiert und anschließend dessen Ergebnisse bewertet (Klassifikation).

3.6.1 Datenerfassung

Verantwortlich: Artur
@Kevin, bitte Korrekturlesen und mit Deinem Teil abstimmen.

Dieser Schritt der ERC beinhaltet die Auswahl und den Aufbau der Sensoren, die Messreihendurchführung (um Daten zu erhalten) und Etikett-Beschriftungstechniken. Ziel ist es relevante und möglichst fehlerfreie Daten von Versuchspersonen für die verschiedenen emotionalen Zustände zu gewinnen. Der Datenerfassungsschritt ist besonders wichtig, da er der erste in der ERC ist und die Ergebnisse aller folgenden Schritte direkt von der Qualität des Datensatzes abhängen.

3.6.2 Vorverarbeitung

Verantwortlich: Artur
- Bereit zum Korrekturlesen.

Das Ziel der Vorverarbeitung ist die "Verbesserung" der Daten für die nachfolgenden Schritte der ERC. In der Regel ist es dadurch besser möglich Muster in Daten erkennen zu können. Vorverarbeitete Daten erreicht man durch Anwendung von z.B. Fil-

terung (Rauschunterdrückung), Normierung oder Reduzierung von unerwünschten oder unbedeutenden Datenteilen.

Fehl am Platz, das kommt erst später.

Normalisierungstechniken wurden auf dem gesamten Datensatz angewendet. Wir haben insbesondere die Standardnormalisierung verwendet, welche den Mittelwert der Daten auf Null setzt und die Einheitsvarianz ergibt [2]. Die Formel für die Standardnormierung lautet:

$$x' = \frac{x - \bar{x}}{\sigma} \quad (1)$$

wobei x ein Datenpunkt eines Sensorkanales, \bar{x} ist der Durchschnitt der Gesamtheit für diesen Sensorkanal und σ ist die entsprechende Standardabweichung.

3.6.3 Segmentation

Verantwortlich: Artur
- Bereit zum Korrekturlesen.

Ziel dieses Schrittes ist es, Teile von Daten zu identifizieren, welche wichtige Informationen über die zu erkennenden Emotionen enthalten. Dies geschieht durch Filtern der Daten und Ausschließen von Segmenten, die für das Klassifizierungsproblem nicht relevant sind. Zusätzlich wird die zu verarbeitende Datenmenge reduziert, indem Segmente eines Zeitfensters fester Größe aus den Daten extrahiert werden. Diese Vorgehensweise ist heute in der Praxis besonders wichtig, da sonst hardwarebedingte Einschränkungen die zu verarbeitende Datenmenge begrenzen könnten.

Eventuell fehl am Platz, wenn hier nur Grundlagen und weitere Details erst später.
- Abklären.

In dieser Projektarbeit wurde ein Schiebefensteransatz (engl. “sliding window approach”) verwendet. Ziel der Methode ist die Segmentierung der vorhandenen Daten in kleinere Einheiten, um die Merkmalsextraktion sowie die anschließende Klassifizierung zu vereinfachen oder gar erst zu ermöglichen. Die Länge des Zeitfensters (engl. “time window”) und des Gleitschritts (engl. “sliding stride”) sind zu bestimmende Parameter (und werden auch als “Hyperparameter” bezeichnet), wobei sich das Zeitfenster auf die feste Größe pro extrahiertem Segment und der Gleitschritt auf den Abstand zu dem Beginn des darauf folgenden Zeitfensters bezieht. Es ist zu beachten, dass sich aufeinanderfolgende Zeitfenster überlappen können, sobald der definierte Gleitschritt kleiner als das Zeitfenster ist.

Die Daten werden auf Zeitstempel-Ebene mit Etiketten beschriftet, basierend auf den von der jeweiligen Versuchsperson ausgefüllten Fragebögen. Jedem Zeitfenster

wird ein Etikett zugeordnet, welches das dominante (d.h. am meisten vorhandene) Etikett der im entsprechenden Fenster enthaltenen Zeitstempel basiert. Es wird davon ausgegangen, dass jedes Zeitfenster nur von einer Emotion belegt ist.

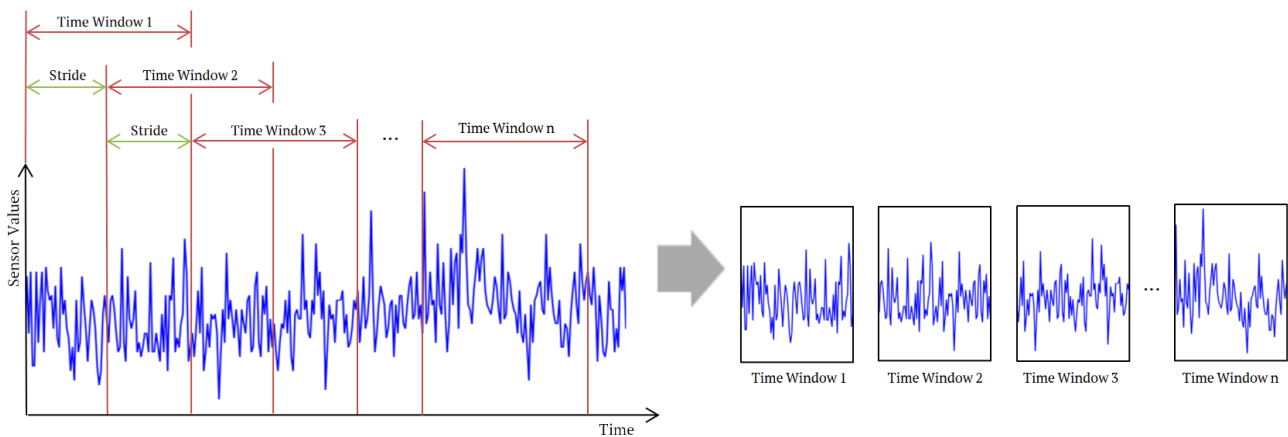


Abbildung 2: Schiebefenster-Segmentierung: Die Daten werden durch ein Zeitfenster fester Größe in kleinere Segmente aufgeteilt. Das Fenster wird mit einem festen Gleichschritt geschoben, um den aufeinanderfolgend Daten-Zeitfenster zu erhalten.

3.6.4 Merkmalsextraktion

Verantwortlich: Artur
- Bereit zum Korrekturlesen.

Hier werden Charakteristiken und Merkmale in den Daten gesucht, die für das Klassifizierungsproblem von möglichst hoher Relevanz sind. Alle nach dem Segmentierungsschritt extrahierte Daten-Zeitfenster kann durch einen Merkmalsvektor (engl. "feature vector") dargestellt werden. Mit Hilfe von Merkmalsvektoren kann ein Klassifikator dann einfacher trainiert werden als nur mit den Rohdaten. Unser Fokus in der Mustererkennung lag vor allem auf der Merkmalsextraktion, da unsere Erfahrungen und frühere Forschungsarbeiten gezeigt haben, dass die Wahl der Merkmale sehr wichtig für die endgültigen Klassifizierungsergebnisse sind. Darüber hinaus wurden noch keine state-of-the-art Merkmale für die Emotionserkennung mit dieser spezifischen Assoziation von eindimensionalen physiologischen Signalen gefunden.

3.6.5 Klassifikation

Verantwortlich: Artur
- LOSOCV im späteren Kapitel erklären.
- Bereit zum Korrekturlesen.

Ziel des Klassifizierungsschritts ist es, ein Klassifizierungsmodell zu trainieren, das in der Lage ist, Objekte in den Daten (dargestellt durch ihren Merkmalsvektor) in

die entsprechende Klasse zuzuordnen.

Der Datensatz der Merkmalsvektoren, der im vorherigen Schritt des ERC erhalten wurde, wird in einen Trainingsset (engl. "training set") und einen Testset (engl. "testing set") unterteilt, so dass alle Klassen in beiden Sets vorhanden sind. Mit dem Trainingsdaten wird ein Klassifikator erstellt und trainiert. Der so erhaltene Klassifikator wird dann anhand der Daten des Testsets ausgewertet. Es ist wichtig, dass die Trainings- und Testsets unterschiedlich sind (d.h. nicht die gleichen im Daten Trainings- und Testset verwenden), da es sonst in einer Überanpassung (engl. "overfitting") des Klassifikator resultieren kann. Eine Überpassung tritt auf, wenn ein Klassifikator zufällige Schwankungen oder Rauschen in den Trainingsdaten "zu gut" lernt und dann bei neuen, unbekannten Daten deutlich schlechter abschneidet. Der Grund hierfür ist, dass diese gelernten Schwankungen oder Rauschen in den Trainingsdaten keinerlei Relevanz für das eigentliche Klassifizierungsproblem haben.

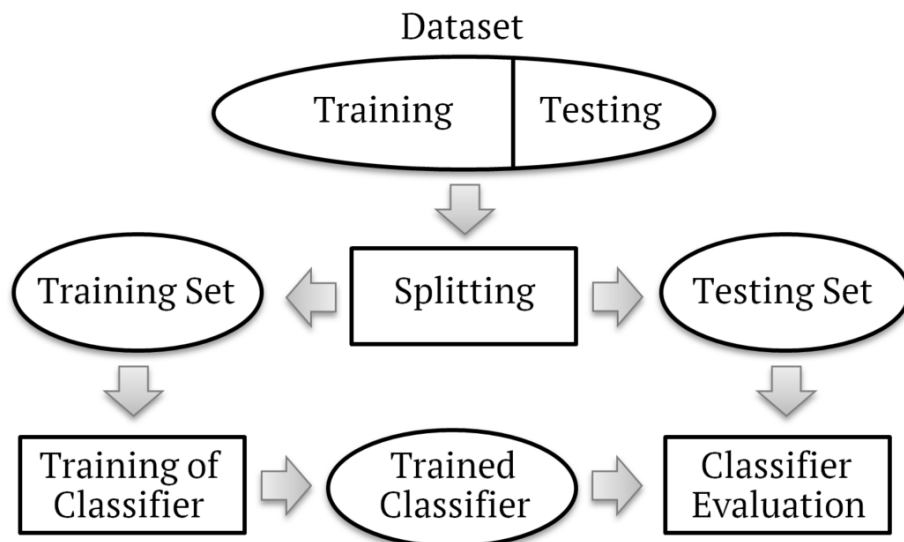


Abbildung 3: Aufteilen eines Datensets in ein Trainings- und ein Testset. Das Trainingsset dient zum Erstellen und Trainieren eines Klassifikators. Die Performance wird mithilfe des Testsets ermittelt und bewertet.

Für die Klassifizierung wurde der state-of-the-art Klassifikator Support-Vector-Machine (SVM) verwendet. SVM ist ein überwachtes Lernansatz, der für binäre Klassifikation oder Regressionszwecke genutzt werden kann. Das Ziel des SVM ist es eine Hyperebene (engl. "hyperplane") zu finden, die zwei Klassen im Raum der Merkmale trennt und gleichzeitig den Abstand (engl. "margin") zwischen der Hyperebene und den nächsten Datenpunkten jeder Klasse maximiert. Für jeden Datenpunkt (dargestellt durch den entsprechenden Vektor von Merkmalen) den wir klassifizieren möchten, wird ein Klassenetiket vergeben, je nachdem, zu welcher Seite der Hyperebene es gehört. Die Methode hat ihren Namen von den "Stützvektoren" (engl. "support vectors"), welche die nächstgelegenen Vektoren beider Klassen zur tren-

nenden Hyperebene sind. Cortes und Vapnik zeigten in [3], dass die Gleichung der optimalen Hyperebene nur von diesen spezifischen Vektoren abhängig ist. Abbildung 3.6.5 zeigt eine optimale Hyperebene in 2D, die beide Klassen perfekt teilt. Datenpunkte werden durch nicht-ausgefüllte blaue Dreiecke bzw. rote Kreise für beide Klassen dargestellt, während Unterstützungsvektoren durch ausgefüllte Punkte und Kreise hervorgehoben werden.

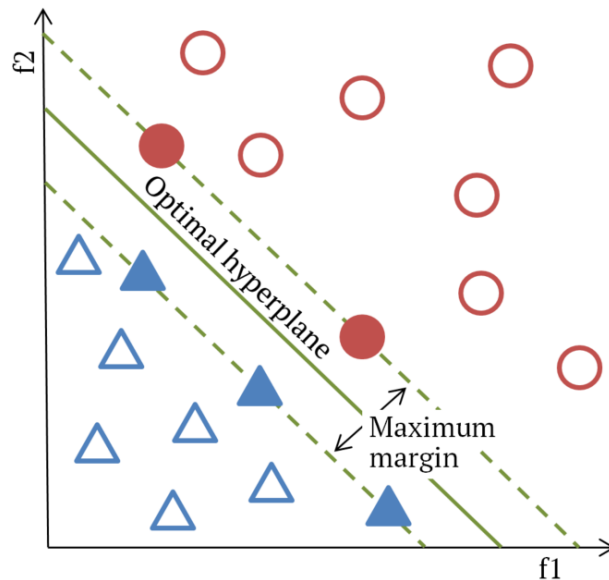


Abbildung 4: Beispiel eines SVM-Klassifikators im zweidimensionalen Merkmalsraum: Die Datenpunkte beider Klassen (dargestellt durch nicht-ausgefüllte blaue Dreiecke und rote Kreise) sind durch eine Hyperebene getrennt, welche die Margin maximiert. Stützvektoren werden als ausgefüllte Dreiecke und Kreise dargestellt.

Es ist anzumerken, dass SVM nur mit zwei Klassen funktioniert, aber sie können zu Z Klassen verallgemeinert werden. Es gibt zwei Ansätze zur Verallgemeinerung auf Z Klassen:

- 1 vs 1 besteht darin, Klassifikatoren für jedes Klassenpaar zu trainieren. Auf diese Weise werden $Z(Z - 1)/2$ Klassifikatoren trainiert, d.h. einen für jedes Klassenpaar.
- 1 vs all besteht darin, $Z - 1$ Klassen als eine Klasse zu betrachten und die letzte als zweite Klasse anzunehmen, mit der der Klassifikator trainiert wird. Dies wird für jede Klasse wiederholt, was zu Z Klassifikatoren führt, d.h. einen für jede Klasse.

In der Praxis sind die Daten durch normales SVM fast nie linear trennbar. Einer der Hauptgründe für die Beliebtheit von SVM ist jedoch die Möglichkeit es so genannten Kernel-Tricks. Es basiert auf der Annahme, dass nicht-linear trennbare Daten linear trennbar werden können, wenn sie in einen Raum höherer Dimension projiziert werden. In [3] zeigten Cortes und Vapnik, dass die SVM-Klassifikationsentscheidungsfunktion

als gewichtete Summe von Skalarprodukten zwischen Stützvektoren und dem Vektor der zu klassifizierenden Merkmale ausgedrückt werden kann. Der Kernel-Trick nutzt dies aus, indem er eine Kernelfunktion einführt, die ein skalares Produkt im hochdimensionalen Zielraum repräsentiert. Diese Kernelfunktion erübrigt die eigentliche Zuordnung zwischen dem ursprünglichen und dem hochdimensionalen Feature-Raum. Außerdem ist die Verwendung des Kernel-Tricks oft rechengünstiger als andere Alternativen. Die beiden beliebtesten Kernel sind der lineare und der Radial Basis Function (RBF) Kernel, definiert als:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|) \quad (2)$$

wobei \mathbf{x} und \mathbf{x}' Vektoren des Merkmalsraums bezeichnen und γ der Parameter ist, der die "Ausbreitung" (engl. "Spread") des Kernels definiert.

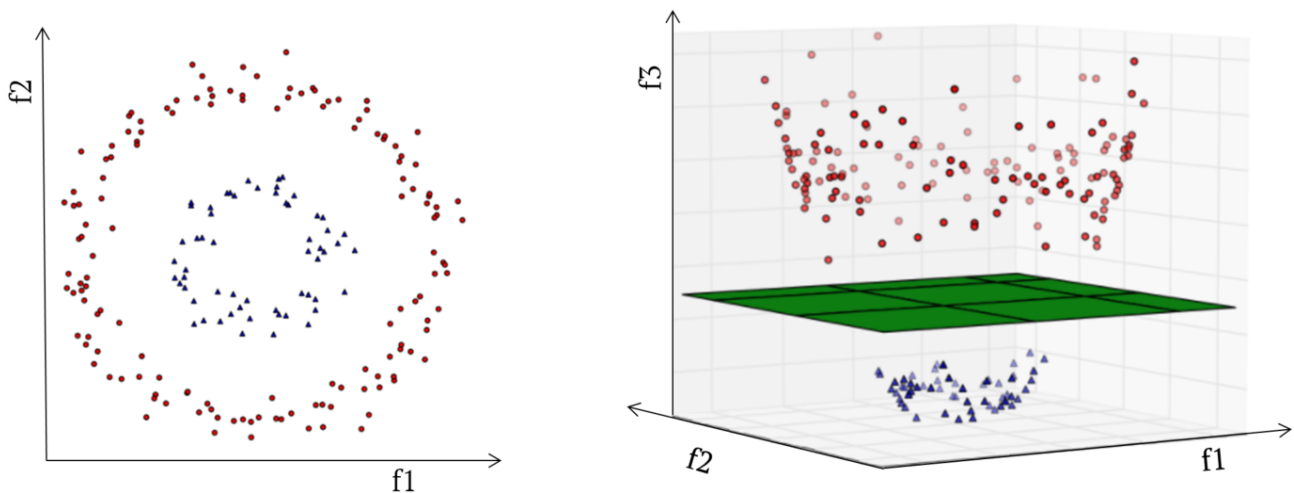


Abbildung 5: Beispiel für die Verwendung des Kernel-Tricks: Nicht-linear trennbare Daten in einem 2D-Merkmalraum (links) können linear trennbar gemacht werden, wenn sie auf 3D projiziert werden (rechts) [4].

Normales SVM verwendet sogenannte hard-margins, die versuchen, eine optimale Hyperebene zu schaffen, die keine Fehlklassifizierungen zulässt. Es ist jedoch oft besser, einige Klassifizierungsfehler zuzulassen, um eine Überanpassung zu verhindern und eine generalisierte Hyperebene zu erhalten. Diese allgemeinere Hyperebene liefert deutlich bessere und zuverlässigere Ergebnisse, wenn sie auf neue und unbekannte Datensätze angewendet wird. Hard-margin SVM kann zu einem Modell führen, das für die Trainingsdaten perfekt funktioniert, aber bei anderen Datensätzen sehr schlecht, weil es seine Trainingsdaten "zu gut" gelernt hat. Aus diesem Grund haben Cortes und Vapnik in [3] eine Variante des Standard-SVM-Klassifikators namens soft-margin SVM (oder auch C-SVM genannt) eingeführt, die eine Fehlklassifizierung von Beispielen beim Erstellen der trennenden Hyperebene toleriert. Der soft-margin Parameter C wird genutzt um die Anzahl der Fehlklassifikationen festzulegen. Je größer der Wert von C , desto weniger Fehleinstufungen

sind zulässig. Umgekehrt erlauben kleine Werte von C mehr Fehlklassifizierungen, um die Verallgemeinerungsfähigkeit des Klassifikators zu verbessern.

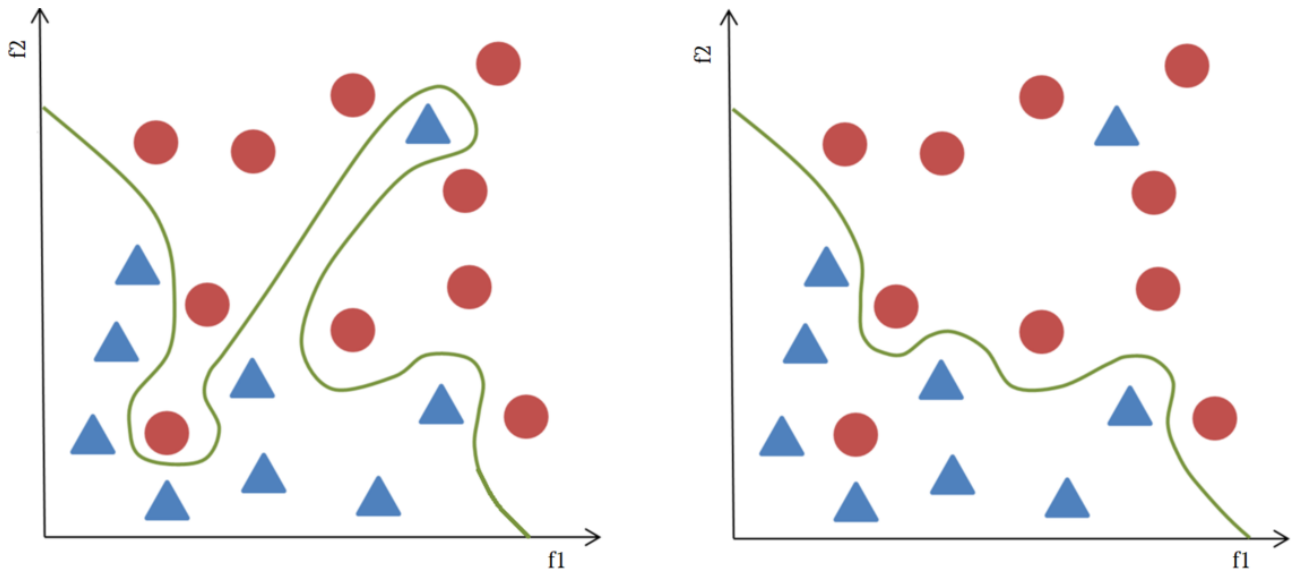


Abbildung 6: Beispiele für einen hard-margin SVM (links) und einen soft-margin SVM (rechts) in einem 2D-Merkmalraum: Der hard-margin SVM trennt die beiden Klassen perfekt im Gegensatz zum Soft-Margin SVM, der einige Fehlklassifikationen zulässt.

4 State-of-the-Art Analyse

Verantwortlich: Jonas

- Next Step: Gliederung erstellen

Erster Prototype

5 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

5.1 Anforderungen

5.2 Konzept

5.3 Hardwareauswahl

5.3.1 Auswahlkriterien

5.3.2 Festlegung der genutzten Hardware

5.4 Hardwarearchitektur

5.4.1 GSR-Sensor

5.4.2 Temperatur-Senosr

5.4.3 Pulsoximeter

5.4.4 EEG

5.4.5 EOG

5.4.6 Datenübertragung

5.5 Programmierung

5.6 Aufnahme der übertragenen Daten

6 Realisierung

Verantwortlich: Jonas, Kevin
- Next Step: Gliederung erstellen

7 Emotionsinduktion

Verantwortlich: Minas

7.1 Ablauf

Verantwortlich: Minas

7.2 Fragebogen

Verantwortlich: Boris

7.3 Szenarien

Verantwortlich: Meryem

7.3.1 Glück

Verantwortlich: Minas

7.3.2 Langeweile

Verantwortlich: Boris

7.3.3 Frustration

Verantwortlich: Meryem

8 Messreihe

Verantwortlich: Kevin, Artur

9 Mustererkennung

Verantwortlich: Artur

9.1 Merkmalsextraktion für Emotionserkennung

9.1.1 Hand-gefertigte Merkmale

9.1.2 Codebook Approach

10 Ergebnisse

Verantwortlich: Artur

10.1 Ergebnisse der hand-gefertigten Merkmale

10.2 Ergebnisse des Codebook Approach

10.3 Analyse der Ergebnisse

Zweiter Prototype

11 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

11.1 Anforderungen

11.2 Konzept

11.3 Hardwareauswahl

11.3.1 Auswahlkriterien

11.3.2 Festlegung der genutzten Hardware

11.4 Hardwarearchitektur

11.4.1 GSR-Sensor

11.4.2 Temperatur-Senosr

11.4.3 Pulsoximeter

11.4.4 EEG

11.4.5 EOG

11.4.6 Datenübertragung

11.5 Programmierung

11.6 Aufnahme der übertragenen Daten

12 Realisierung

Verantwortlich: Jonas, Kevin
- Next Step: Gliederung erstellen

13 Emotionsinduktion

Verantwortlich: Minas

13.1 Ablauf

Verantwortlich: Minas

13.2 Fragebogen

Verantwortlich: Boris

13.3 Szenarien

Verantwortlich: Meryem

13.3.1 Glück

Verantwortlich: Minas

13.3.2 Langeweile

Verantwortlich: Boris

13.3.3 Frustration

Verantwortlich: Meryem

14 Messreihe

Verantwortlich: Kevin, Artur

15 Mustererkennung

Verantwortlich: Artur

15.1 Merkmalsextraktion für Emotionserkennung

15.1.1 Hand-gefertigte Merkmale

15.1.2 Codebook Approach

16 Ergebnisse

Verantwortlich: Artur

16.1 Ergebnisse der hand-gefertigten Merkmale

16.2 Ergebnisse des Codebook Approach

16.3 Analyse der Ergebnisse

Dritter Prototype

17 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

17.1 Anforderungen

17.2 Konzept

17.3 Hardwareauswahl

17.3.1 Auswahlkriterien

17.3.2 Festlegung der genutzten Hardware

17.4 Hardwarearchitektur

17.4.1 GSR-Sensor

17.4.2 Temperatur-Sensor

17.4.3 Pulsoximeter

17.4.4 EEG

17.4.5 EOG

17.4.6 Datenübertragung

17.5 Programmierung

17.6 Aufnahme der übertragenen Daten

18 Realisierung

Verantwortlich: Jonas, Kevin
- Next Step: Gliederung erstellen

19 Emotionsinduktion

Verantwortlich: Minas

19.1 Ablauf

Verantwortlich: Minas

19.2 Fragebogen

Verantwortlich: Boris

19.3 Szenarien

Verantwortlich: Meryem

19.3.1 Glück

Verantwortlich: Minas

19.3.2 Langeweile

Verantwortlich: Boris

19.3.3 Frustration

Verantwortlich: Meryem

20 Messreihe

Verantwortlich: Kevin, Artur

21 Mustererkennung

Verantwortlich: Artur

21.1 Merkmalsextraktion für Emotionserkennung

21.1.1 Hand-gefertigte Merkmale

21.1.2 Codebook Approach

22 Ergebnisse

Verantwortlich: Artur

22.1 Ergebnisse der hand-gefertigten Merkmale

22.2 Ergebnisse des Codebook Approach

22.3 Analyse der Ergebnisse

Vierter Prototype

23 Systementwurf und Konzept

Verantwortlich: Kevin, Jonas

23.1 Anforderungen

23.2 Konzept

23.3 Hardwareauswahl

23.3.1 Auswahlkriterien

23.3.2 Festlegung der genutzten Hardware

23.4 Hardwarearchitektur

23.4.1 GSR-Sensor

23.4.2 Temperatur-Sensor

23.4.3 Pulsoximeter

23.4.4 EEG

23.4.5 EOG

23.4.6 Datenübertragung

23.5 Programmierung

23.6 Aufnahme der übertragenen Daten

24 Realisierung

Verantwortlich: Jonas, Kevin
- Next Step: Gliederung erstellen

25 Emotionsinduktion

Verantwortlich: Minas

25.1 Ablauf

Verantwortlich: Minas

25.2 Fragebogen

Verantwortlich: Boris

25.3 Szenarien

Verantwortlich: Meryem

25.3.1 Glück

Verantwortlich: Minas

25.3.2 Langeweile

Verantwortlich: Boris

25.3.3 Frustration

Verantwortlich: Meryem

26 Messreihe

Verantwortlich: Kevin, Artur

27 Mustererkennung

Verantwortlich: Artur

27.1 Merkmalsextraktion für Emotionserkennung

27.1.1 Hand-gefertigte Merkmale

27.1.2 Codebook Approach

28 Ergebnisse

Verantwortlich: Artur

28.1 Ergebnisse der hand-gefertigten Merkmale

28.2 Ergebnisse des Codebook Approach

28.3 Analyse der Ergebnisse

29 Zusammenfassung und Ausblick

Machen wir später.

29.1 Zusammenfassung

29.2 Fazit

29.3 Ausblick

Abbildungsverzeichnis

1	Emotion Recognition Chain	12
2	Schiebefenster-Segmentierung	14
3	Aufteilen eines Datensets in ein Trainings- und Testset	15
4	Beispiel eines SVM-Klassifikators im zweidimensionalen Merkmal- raum	16
5	Beispiel für die Verwendung des Kernel-Tricks	17
6	Beispiele für einen hard-margin SVM (links) und einen soft-margin SVM (rechts) in einem 2D-Merkmalsraum	18

Tabellenverzeichnis

Abkürzungen

Bitte alle verwendete Abkürzungen nochmals hier aufführen.

ANN	Artificial Neural Networks
BMBF	Bundesministerium für Bildung und Forschung
BVP	Blood Volume Pulse
CA	Codebook Approach
CRID	Center for Responsible Innovation & Design
C-SVM	Soft-margin Support-Vector-Machine
CSV	Comma-Separated-Values
EEG	Electroencephalography
EOG	Electrooculography
ERC	Emotion Recognition Chain
GSR	Galvanic Skin Response
HR	Heart Rate
LOSOVC	Leave-One-Subject-Out-Cross-Validation
PPG-ir	Photoplethysmography Infrared
PPG-red	Photoplethysmography Red
RBF	Radial Basis Function
SpO2	Pulse Oximetry
SVM	Support-Vector-Machine
VR	Virtual Reality

Anhang

Referenzen

- [1] X. Zhu. Semi-supervised learning literature survey, July 2008.
- [2] J. Grus. *Data Science from Scratch*. O'Reilly Media, April 2015.
- [3] C. Cortes and V. Vapnik. *Support-Vector Networks*, volume volume 20. Kluwer Academic Publishers, Boston, September 1995.
- [4] E. Kim. Everything you wanted to know about the kernel trick. September 2013.