Universität des Saarlandes Fachbereich Mathematik

Maschinelles Lernen in der Medizin

Anwendung von Support Vector Machines in der Ganganalyse

Diplomarbeit
zur Erlangung des akademischen Grades einer
Diplom-Mathematikerin
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes

von
Sabrina Bechtel

Saarbrücken
November 2008

Betreuer: Univ.-Prof. Dr. A. K. Louis

In Gedenken an meine Mutter
Annerose Bechtel
(03.11.1958 - 25.09.2000)

Inhaltsverzeichnis

Einleitung

Der Begriff *maschinelles Lernen* steht für die Generierung von „Wissen“ aus Erfahrungen mittels Algorithmen. Diese „lernen“ aus Beispielen und können nach Beendigung der Lernphase auch unbekannte Daten beurteilen. Einordnen kann man das maschinelle Lernen in das Gebiet der Künstlichen Intelligenz und des Data Mining.

In diesem Zusammenhang unterscheidet man drei Arten von Lernen: überwachtes Lernen, unüberwachtes Lernen und Bekräftigungslernen. Im ersten Fall lernt der Algorithmus aus gegebenen Ein- und Ausgabepaaren. Er berechnet eine Vorhersagefunktion, die zu den vorhandenen Eingaben die passende Ausgabe liefert und für neue Eingaben nach diesem Vorbild eine Ausgabe erzeugt. Zu diesem Gebiet gehören die Neuronalen Netze und auch die Support Vector Machines. Beim unüberwachten Lernen wird aus einer gegebenen Eingabemenge ein Modell entwickelt, um diese zu beschreiben und geeignete Vorhersagen zu treffen, eine Ausgabe ist hier nicht gegeben. Dazu gehören Clustering Verfahren, welche die Eingabedaten in Gruppen ähnlicher Struktur einteilen. Auch die Hauptkomponentenanalyse stellt ein unüberwachtes Lernverfahren dar. Die Trainingsdaten werden in einem Unterraum niedrigerer Dimension dargestellt. Man geht davon aus, dort auch neue Daten geeignet darstellen zu können, insofern erfolgt auch hier eine Vorhersage. Das Bekräftigungslernen findet Anwendung bei autonomen Agenten wie z.B. Robotern. Dort lernt der Algorithmus durch Belohnung und Bestrafung eine Taktik, wie in potenziell auftretenden Situationen zu handeln ist.

Zwei der genannten Methoden des maschinellen Lernens finden in dieser Arbeit Verwendung. Das Ziel ist die automatische Unterscheidung gesunder Personen von solchen, die an Parkinson erkrankt sind. Zunächst wird nach der Vorlage von Troje in [13] mit Hilfe der Hauptkomponentenanalyse eine geeignete Darstellung des Gangs entwickelt um dann mit einer Support Vector Machine die automatische Trennung durchführen zu können. Dabei stellt sich heraus, dass die Anwendung einer Support Vector Machine zur Trennung gewisse Vorteile gegenüber dem Ansatz von Troje mit sich bringt, wie zum Beispiel die Darstellung in Form einer trennenden Hyperebene.

Den Anfang der Arbeit bilden die mathematischen Grundlagen, es werden statistische Standardvokabeln für die Hauptkomponentenanalyse definiert und ein Überblick über konvexe Optimierung, die für Support Vector Machines wichtig ist, geschaffen. Im Anschluss daran wird näher auf die Hauptkomponentenanalyse eingegangen. Es folgt ein ausführliches Kapitel über Support Vector Machines. Dabei wird das auftre-

tende Optimierungsproblem genauer untersucht, um auf eine besondere Eigenschaft der Lösung in den dualen Variablen zu schließen. Im Hinblick auf die durchgeführte Programmierung einer Support Vector Machine wird genau angegeben, wie die Entscheidungsfunktion berechnet werden kann und wie man geeignete Parameter sucht. Damit wurden die Grundlagen für die Anwendung in der Ganganalyse geschaffen, die in dem darauffolgenden Kapitel behandelt wird. Zunächst werden die Daten in einem geeigneten Koordinatensystem dargestellt, durchlaufen anschließend eine bzw. zwei Hauptkomponentenanalysen und haben erst dann eine für die Klassifikation geeignete Form. Abschließend erfolgen die Tests mit den zur Verfügung stehenden Daten mit dem Ergebnis, dass man parkinsonkranke von gesunden Personen anhand der vorliegenden Gangbilder durch eine Support Vector Machine linear trennen kann.

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben. Mein besonderer Dank gilt Herrn Prof. Dr. Louis für die interessante und aktuelle Aufgabenstellung meiner Arbeit.
Ebenfalls richtet sich mein Dank an Frau Dipl.-Math. Yvonne Johann für die zahlreichen Stunden praktischer Ganganalyse und vieles mehr. Ein Dankeschön gebührt außerdem sämtlichen Mitarbeitern des Instituts, die stets überaus Hilfsbereit waren. Insbesondere danke ich Herrn Dipl.-Math. Thomas Weber und Herrn Dipl.-Math. Jochen Krebs fürs Korrekturlesen und die fachlichen Ratschläge. Ebenso Herrn Dr. Uwe Schmitt, der sich gerne für mich und meine Fragen Zeit genommen hat.
Ein großer Dank gilt auch meiner Familie und meinem Freund Klaus, die mich während meines gesamten Studiums unterstützt haben.

1 Mathematische Grundlagen

Dieses Kapitel teilt sich in zwei Unterabschnitte. Die *statistischen Grundlagen* werden für die Hauptkomponentenanalyse benötigt, die *Optimierung* ist für das Verständnis von Support Vector Machines notwendig.

1.1 Statistik

Die folgenden Grundlagen der Statistik findet man beispielsweise in [7]. Gegeben seien Punkte $\boldsymbol{x_1}, \ldots, \boldsymbol{x_m} \in \mathbb{R}^n$, mit denen wir eine Matrix $X = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_m})$ des $\mathbb{R}^{n \times m}$ definieren. Mit $\boldsymbol{x^{(1)}}, \ldots, \boldsymbol{x^{(n)}} \in \mathbb{R}^m$ werden die Zeilen der Matrix beschrieben. Mit $\mathbf{1_m}$ bezeichnen wir den m-dimensionalen Vektor $(1, \ldots, 1)^{\mathrm{T}}$.

1.1 Definition (Arithmetisches Mittel)
Das *arithmetische Mittel* $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ *von* X ist der Spaltenvektor

$$\overline{\boldsymbol{x}} = \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{x_j} \ .$$

1.2 Bemerkung
Die i-te Komponente $\overline{x}^{(i)}$ ist der Mittelwert der Punkte $\boldsymbol{x_1}, \ldots, \boldsymbol{x_m} \in \mathbb{R}^n$ entlang des i-ten kanonischen Einheitsvektors. $\overline{\boldsymbol{x}}$ kann als Mittelpunkt der Daten interpretiert werden, sagt jedoch nichts über deren Verteilung aus. Dies berücksichtigt die folgende Bezeichnung.

1.3 Definition (Varianz)
Die *Varianz* der i-ten Zeile von X ist definiert als

$$Var\big(\boldsymbol{x^{(i)}}\big) = \frac{1}{m-1} \left(\boldsymbol{x^{(i)}} - \overline{x}^{(i)} \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\right) \left(\boldsymbol{x^{(i)}} - \overline{x}^{(i)} \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\right)^{\mathrm{T}} .$$

Man beachte, dass $\left(\boldsymbol{x^{(i)}} - \overline{x}^{(i)} \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\right)$ ein Zeilenvektor ist und somit $Var\big(\boldsymbol{x^{(i)}}\big)$ ein Skalar.

1.4 Bemerkung
In der Berechnung der Varianz wird entlang des i-ten kanonischen Einheitsvektors $\boldsymbol{e_i}$ die durchschnittliche Distanz zwischen dem arithmetischen Mittel $\overline{x}^{(i)}$ und einem allgemeinen Datenpunkt betrachtet und dadurch die Streuung der Daten charakterisiert. Möchte man die Varianz von X entlang eines allgemeinen Vektors $\boldsymbol{r}$ berechnen, so geschieht dies durch $Var\big(\boldsymbol{r}^{\mathrm{T}} X\big)$, da $\boldsymbol{x^{(i)}} = \boldsymbol{e_i}^{\mathrm{T}} X$ und $\overline{x}^{(i)} = \boldsymbol{e_i}^{\mathrm{T}} \overline{\boldsymbol{x}}$.

1.5 Definition (Standardabweichung)
Die *Standardabweichung* der i-ten Zeile ist definiert als die Wurzel aus der Varianz, es gilt also:

$$\sigma\big(\boldsymbol{x^{(i)}}\big) = \sqrt{Var\big(\boldsymbol{x^{(i)}}\big)} \ .$$

Es folgt eine Verallgemeinerung der Varianz auf zwei Dimensionen, die Kovarianz. Hier berücksichtigt man ebenfalls, wie stark die Datenpunkte entlang des jeweiligen kanonischen Einheitsvektors durchschnittlich von ihrem Mittel abweichen. Durch Multiplikation der beiden Abweichungen werden die unterschiedlichen Komponenten in Bezug zueinander gesetzt.

1.6 Definition (Kovarianz)
Die *Kovarianz von* X zwischen der i-ten und j-ten Dimension ist definiert als

$$\mathrm{Cov}\big(\boldsymbol{x^{(i)}}, \boldsymbol{x^{(j)}}\big) = \frac{1}{m-1}\left(\boldsymbol{x^{(i)}} - \overline{x}^{(i)}\mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\right)\left(\boldsymbol{x^{(j)}} - \overline{x}^{(j)}\mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\right)^{\mathrm{T}} \ .$$

1.7 Bemerkung
Die Kovarianz ist ein Maß für den Zusammenhang zweier statistischer Merkmale, die durch die k-ten und l-ten Einträge der Punkte gegeben sind. Gilt $\mathrm{Cov}\big(\boldsymbol{x^{(k)}}, \boldsymbol{x^{(l)}}\big) > 0$, so besitzen die Merkmale eine ähnliche Tendenz, bei negativem Vorzeichen eine entgegengesetzte. Ist der Wert Null, so ist aus den Daten kein linearer Zusammenhang ersichtlich.

Die Kovarianz hängt, wie die Varianz auch, von den Maßeinheiten der Variablen ab. Deshalb gibt sie nur die Richtung einer Beziehung zwischen $\boldsymbol{x^{(k)}}$ und $\boldsymbol{x^{(l)}}$ an, nicht aber deren Stärke. Die Kovarianzen der verschiedenen Dimensionen von X werden in einer Matrix zusammengefasst.

1.8 Definition (Kovarianzmatrix)
Die *Kovarianzmatrix von* X ist die Matrix $C\big(X\big) \in \mathbb{R}^{n\times n}$ mit

$$C\big(X\big) = \frac{1}{m-1}\Big(X - \overline{\boldsymbol{x}}\,\mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\Big)\Big(X - \overline{\boldsymbol{x}}\,\mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\Big)^{\mathrm{T}} \ .$$

Ihre Einträge sind die Kovarianzen der entsprechenden Dimensionen:

$$C_{i,j}\big(X\big) = \mathrm{Cov}\big(\boldsymbol{x^{(i)}}, \boldsymbol{x^{(j)}}\big) \ .$$

Somit enthält die Kovarianzmatrix Informationen über den linearen Zusammenhang aller Dimensionen untereinander.

1.9 Satz
Die Kovarianzmatrix zu einer Datenmenge $X \in \mathbb{R}^{n\times m}$ *ist symmetrisch und positiv semidefinit.*

Beweis erfolgt durch einfaches Nachrechnen. □

1.2 Optimierung

Die Grundlagen zur Optimierung findet man im Buch von Geiger [6], ein Kapitel zur Optimierung im Rahmen von Support Vector Machines ist auch in [3] enthalten. Die Beweise zu den angegebenen Sätzen und Lemmata findet man dort ebenfalls.

1.2.1 Begriffsbildung

Zunächst werden die grundlegenden Definitionen zur Optimierung skizziert.

1.10 Definition (Allgemeines restringiertes Optimierungsproblem)
Ein *allgemeines restringiertes Optimierungsproblem* hat die Form

$$\min_{\boldsymbol{\omega}\in\Omega} f(\boldsymbol{\omega}), \quad \text{u. d. N.} \begin{cases} g_i(\boldsymbol{\omega}) \leq 0, & i \in \{1,\dots,m\} \\ h_j(\boldsymbol{\omega}) = 0, & j \in \{1,\dots,k\} \end{cases} \tag{1.1}$$

mit Abbildungen $f, g_i, h_j : \mathbb{R}^n \to \mathbb{R},\ \Omega \subset \mathbb{R}^n$.
Dabei bezeichnet man f als *die Zielfunktion*, g_i als *die Ungleichheitsrestriktionen* und h_j als *die Gleichheitsrestriktionen*. Den optimalen Wert der Zielfunktion nennt man *den Wert des Optimierungsproblems*.

1.11 Definition (Lagrange Funktion)
Die durch

$$L(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) := f(\boldsymbol{\omega}) + \sum_{i=1}^{m} \alpha_i\, g_i(\boldsymbol{\omega}) + \sum_{j=1}^{k} \beta_j\, h_j(\boldsymbol{\omega})$$

definierte Abbildung $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^k \to \mathbb{R}$ heißt *Lagrange Funktion* des restringierten Optimierungsproblems (1.1). Die Zahlenwerte α_i und β_j werden auch *Lagrange Multiplikatoren* genannt.

1.12 Bemerkung
Man beachte, dass die Restriktion $\boldsymbol{\omega} \in \Omega$ nicht in die Lagrange Funktion aufgenommen wird.

1.13 Definition (Lagrange Dualproblem)
Die Funktion

$$q(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \inf_{\boldsymbol{\omega}\in\Omega} L(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

heißt die *Lagrange Dualfunktion von* (1.1), das Optimierungsproblem

$$\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} q(\boldsymbol{\alpha}, \boldsymbol{\beta}), \quad \text{u. d. N.} \begin{cases} \boldsymbol{\alpha} \geq 0 \\ \boldsymbol{\beta} \in \mathbb{R}^k \end{cases} \tag{1.2}$$

heißt *Lagrange Dualproblem zu* (1.1). Das Minimierungsproblem (1.1) wird in diesem Zusammenhang als *Primalproblem* bezeichnet.

1.14 Bemerkung
Der hier eingeführte Begriff des Lagrange Dualproblems ist mit dem des Dualproblems für lineare Optimierungsprobleme konsistent. Die Lagrange Dualität kann demnach als eine Verallgemeinerung der bekannten Begriffe für lineare Optimierungsprobleme angesehen werden.

1.15 Definition (zulässig)
Ein Vektor heißt *zulässig* für ein Optimierungsproblem, wenn er sämtliche Nebenbedingungen erfüllt. Dazu zählt auch die Bedingung, die direkt unter „min“ bzw. „max“ steht.

1.16 Satz (Schwache Dualität)
Ist $\boldsymbol{\omega} \in \Omega$ zulässig für das Primalproblem (P) aus (1.1) und $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathbb{R}^m \times \mathbb{R}^k$ zulässig für das Lagrange Dualproblem (D) aus (1.2), so gilt:

$$q(\boldsymbol{\alpha}, \boldsymbol{\beta}) \leq f(\boldsymbol{\omega}) \ .$$

Bezeichnen

$$\begin{aligned} \inf(P) &:= \inf\left\{f(\boldsymbol{\omega}) \mid \boldsymbol{\omega} \in \Omega,\ \boldsymbol{g}(\boldsymbol{\omega}) \leq 0,\ \boldsymbol{h}(\boldsymbol{\omega}) = 0\right\}, \\ \sup(D) &:= \sup\left\{q(\boldsymbol{\alpha}, \boldsymbol{\beta}) \mid \boldsymbol{\alpha} \geq 0,\ \boldsymbol{\beta} \in \mathbb{R}^k\right\} \end{aligned}$$

die Optimalwerte des Primal- und des Dualproblems, so besteht die Ungleichung

$$\sup(D) \leq \inf(P) \ .$$

1.17 Bemerkung
Hat man zulässige Lösungen $\boldsymbol{\omega}^*$ und $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ gefunden und sind die Werte der entprechenden Zielfunktionen gleich, so sind die Lösungen optimal. Die Gleichheit der Lösungen ist nicht immer garantiert. Die Differenz zwischen den optimalen Werten des Optimierungsproblem (1.1) und (1.2) nennt man *Dualitätslücke*. Eine Möglichkeit herauszufinden, ob die Dualitätslücke den Wert Null besitzt, ist das Nachprüfen des Vorhandenseins eines Sattelpunktes.

1.18 Definition (Sattelpunkt der Lagrange Funktion)
Ein Vektor $(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^k$ mit $\boldsymbol{\alpha}^* \geq 0$ heißt *Sattelpunkt der Lagrange Funktion L*, wenn die Ungleichungen

$$L(\boldsymbol{\omega}^*, \boldsymbol{\alpha}, \boldsymbol{\beta}) \leq L(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \leq L(\boldsymbol{\omega}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$$

für alle $(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \in \Omega \times \mathbb{R}^m \times \mathbb{R}^k$ mit $\boldsymbol{\alpha} \geq 0$ gelten.

An $\boldsymbol{\omega}^*$ wurde hier nicht die Bedingung gestellt, dass es die Gleichheits- oder die Ungleichheitsrestriktionen erfüllen soll.

1.19 Satz
Der Vektor $(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ ist genau dann Sattelpunkt der Lagrange Funktion zu (1.1), wenn $\boldsymbol{\omega}^*$ und $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ jeweils optimale Lösungen zum Primalproblem (1.1) und zum Lagrange Dualproblem (1.2) sind und die Dualitätslücke den Wert Null hat. Es gilt dann: $f(\boldsymbol{\omega}^*) = q(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$.

1.2.2 Lineare Restriktionen

Von nun an betrachten wir linear restringierte Optimierungsprobleme mit konvexer Zielfunktion, da diese für die Theorie der Support Vector Machines von Bedeutung sind.

1.20 Definition (Linear restringiertes Optimierungsproblem)
Das *linear restringierte Optimierungsproblem* hat die Form:

$$\min f(\boldsymbol{\omega}), \quad \text{u. d. N.} \begin{cases} g_i(\boldsymbol{\omega}) \leq 0, & i \in \{1, \ldots, m\} \\ h_j(\boldsymbol{\omega}) = 0, & j \in \{1, \ldots, k\} \end{cases} \tag{1.3}$$

mit einer stetig differenzierbaren Funktion $f : \mathbb{R}^n \to \mathbb{R}$ und affin linearen Funktionen $g_i, h_j : \mathbb{R}^n \to \mathbb{R}$.

1.21 Bemerkung
Das Optimierungsproblem (1.3) bildet einen Spezialfall des allgemeinen Optimierungsproblems (1.1) mit $\Omega = \mathbb{R}^n$. Der Begriff lineares Optimierungsproblem hingegen beinhaltet auch die Linearität der Zielfunktion.

Konvexe Zielfunktionen besitzen in Bezug auf die Optimierung sehr angenehme Eigenschaften:

1.22 Lemma
Jedes lokale Minimum des Optimierungsproblems (1.3) ist auch ein globales Minimum.

1.23 Satz (Starke Dualität)
Gegeben sei das Optimierungsproblem (1.3), dann hat die Dualitätslücke den Wert Null. Es gilt also

$$f(\boldsymbol{\omega}^*) = q(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$$

für optimale Lösungen $\boldsymbol{\omega}^*$ des Primalproblems (1.3) und optimale Lösungen $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ des Lagrange Dualproblems (1.2).

1.24 Definition (KKT Bedingungen)
Betrachte das Optimierungsproblem (1.3).

a.) Die Bedingungen

$$\nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = 0, \tag{KKT 1}$$
$$\boldsymbol{g}(\boldsymbol{\omega}) \leq 0, \tag{KKT 2}$$
$$\boldsymbol{h}(\boldsymbol{\omega}) = 0, \tag{KKT 3}$$
$$\boldsymbol{\alpha} \geq 0, \tag{KKT 4}$$
$$\langle \boldsymbol{\alpha}, \boldsymbol{g}(\boldsymbol{\omega}) \rangle = 0 \tag{KKT 5}$$

heißen *Karush Kuhn Tucker Bedingungen (kurz: KKT Bedingungen)* des Optimierungsproblems (1.3), wobei

$$\nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \nabla f(\boldsymbol{\omega}) + \sum_{i=1}^{m} \alpha_i \nabla g_i(\boldsymbol{\omega}) + \sum_{i=1}^{k} \beta_j \nabla h_j(\boldsymbol{\omega})$$

den Gradienten der Lagrange Funktion L bezüglich der Variablen $\boldsymbol{\omega}$ bezeichnet.

b.) Jeder Vektor $(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^k$, der den KKT Bedingungen genügt, heißt *Karush Kuhn Tucker Punkt (kurz: KKT Punkt)* des Optimierungsproblems (1.3).

1.25 Bemerkung
Liegen keine Restriktionen vor, so reduzieren sich die KKT Bedingungen offensichtlich auf die Forderung $\nabla f(\boldsymbol{\omega}) = 0$. Sie verallgemeinern somit die übliche notwendige Optimalitätsbedingung für unrestringierte Minimierungsaufgaben. Der folgende Satz liefert hinreichende Bedingungen für eine optimale Lösung zu (1.3).

1.26 Satz (Sattelpunkt Theorem)
Betrachte das Optimierungsproblem (1.3). Das Tripel $(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^k$ *ist genau dann Sattelpunkt der Lagrange Funktion L, wenn* $(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ *ein KKT Punkt dieses Optimierungsproblems ist.*

Dies ist eine schöne Eigenschaft von linear restringierten Problemen, da dies im Allgemeinen nur unter Verwendung weiterer Regularitätsannahmen möglich ist.

1.27 Bemerkung
Sei $(\boldsymbol{\omega}^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ ein KKT Punkt, dann ist er nach Satz 1.26 auch ein Sattelpunkt der Lagrange Funktion. Mit Satz 1.19 folgt, dass $\boldsymbol{\omega}^*$ optimale Lösung für das Primalproblem (1.3) und $(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ optimale Lösung für das Lagrange Dualproblem (1.2) ist. Nach dem Satz über die Starke Dualität (Satz 1.23) folgt, dass die Dualitätslücke den Wert Null hat und somit $f(\boldsymbol{\omega}^*) = q(\boldsymbol{\alpha}^*, \boldsymbol{\beta}^*)$ gilt. Folglich ist es egal, ob man das Primalproblem oder das Lagrange Dualproblem löst.

2 Hauptkomponentenanalyse

Die Hauptkomponentenanalyse (engl. Principle Component Analysis, kurz: PCA) ist ein statistisches Verfahren, welches hauptsächlich zur Datenkomprimierung benutzt wird. Dazu werden die Daten so auf einen Unterraum projiziert, dass sie in diesem einen großen Anteil ihrer ursprünglichen Gesamtvarianz bilden. Es werden Orthonormalvektoren $\boldsymbol{v_i}$ konstruiert, bezüglich denen die Datenpunkte sukzessive am meisten streuen. Diese bilden die Achsen eines neuen Koordinatensystems, in dem nur wenige Basisvektoren ausreichen, um die Daten ohne großen Informationsverlust zu beschreiben. In dieser Arbeit wird die Hauptkomponentenanalyse gleich zwei Mal zur Datenreduktion verwendet.

2.1 Begriffsbildung

Gegeben seien Punkte $\boldsymbol{x_1}, \ldots, \boldsymbol{x_m} \in \mathbb{R}^n$, die in der Matrix $X \in \mathbb{R}^{n \times m}$ gespeichert werden.

Um zunächst alle Komponenten der Datenpunkte paarweise miteinander in Zusammenhang zu bringen, wird die Kovarianzmatrix $C \in \mathbb{R}^{n \times n}$ von X berechnet. Nach Satz 1.9 ist sie symmetrisch positiv semidefinit. Daher existieren n reelle Eigenwerte $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$ mit zugehörigen orthonormalen Eigenvektoren $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n} \in \mathbb{R}^n$. Es gilt also:

$$C\,\boldsymbol{v_j} = \lambda_j\,\boldsymbol{v_j}, \qquad \text{und} \qquad \boldsymbol{v_j}^{\mathrm{T}}\boldsymbol{v_k} = \begin{cases} 1 & \text{falls } j = k \\ 0 & \text{falls } j \neq k \end{cases}$$

für alle $j, k \in \{1, \ldots, n\}$. Die gefundenen Eigenvektoren werden die Basisvektoren des neuen Koordinatensystems bilden. Die Varianz der Datenpunkte ist in Richtung des ersten Basisvektors maximal und sinkt mit wachsendem Basisindex.

2.1 Satz
Gegeben seien die orthonormalen Eigenvektoren $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n} \in \mathbb{R}^n$ zu den Eigenwerten $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ der Kovarianzmatrix C. Es gilt:

a.) *Unter allen Vektoren $\boldsymbol{v}$ der Länge 1 ist $\boldsymbol{v_1}$ derjenige, entlang dessen die Varianz von X am größten ist.*

b.) *Für $j \in \{2, \ldots, n\}$ gilt: Unter allen Vektoren $\boldsymbol{v}$ der Länge 1, die zu $\boldsymbol{v_1}, \ldots, \boldsymbol{v_{j-1}}$ orthogonal sind, ist $\boldsymbol{v_j}$ derjenige, entlang dessen die Varianz von X am größten ist.*

Beweis
Sei $\boldsymbol{v} \in \mathbb{R}^n$ ein normierter Vektor. Dieser kann geschrieben werden als $\boldsymbol{v} = \sum_{i=1}^{n} a_i \, \boldsymbol{v_i}$, mit Koeffizienten $a_i \in \mathbb{R}$.
Es gilt:

$$1 = \langle \boldsymbol{v}, \boldsymbol{v} \rangle = \langle \sum_{i=1}^{n} a_i \, \boldsymbol{v_i}, \sum_{j=1}^{n} a_j \, \boldsymbol{v_j} \rangle \overset{a_i \in \mathbb{R}}{=} \sum_{i,j=1}^{n} a_i \, a_j \, \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle \overset{\boldsymbol{v_i}\text{ONB}}{=} \sum_{i=1}^{n} a_i^2 \; . \tag{2.1}$$

Die Varianz von X entlang der Richtung $\boldsymbol{v}$ ist:

$$\begin{aligned} Var(\boldsymbol{v}^{\mathrm{T}} X) &= \frac{1}{m-1} \left(\boldsymbol{v}^{\mathrm{T}} X - \overline{\boldsymbol{v}^{\mathrm{T}} X} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right) \left(\boldsymbol{v}^{\mathrm{T}} X - \overline{\boldsymbol{v}^{\mathrm{T}} X} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right)^{\mathrm{T}} \\ &= \frac{1}{m-1} \left(\boldsymbol{v}^{\mathrm{T}} X - \boldsymbol{v}^{\mathrm{T}} \, \overline{\boldsymbol{x}} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right) \left(\boldsymbol{v}^{\mathrm{T}} X - \boldsymbol{v}^{\mathrm{T}} \, \overline{\boldsymbol{x}} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right)^{\mathrm{T}} \\ &= \frac{1}{m-1} \left(\boldsymbol{v}^{\mathrm{T}} \left(X - \overline{\boldsymbol{x}} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right) \left(X - \overline{\boldsymbol{x}} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right)^{\mathrm{T}} \boldsymbol{v} \right) \\ &= \boldsymbol{v}^{\mathrm{T}} \left(\frac{1}{m-1} \left(X - \overline{\boldsymbol{x}} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right) \left(X - \overline{\boldsymbol{x}} \, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right)^{\mathrm{T}} \right) \boldsymbol{v} \\ &= \boldsymbol{v}^{\mathrm{T}} C \, \boldsymbol{v} \; . \end{aligned}$$

zu a.) Zu zeigen ist, dass für alle $\boldsymbol{v}$ mit $\|\boldsymbol{v}\| = 1$ gilt: $\boldsymbol{v}^{\mathrm{T}} C \, \boldsymbol{v} \leq \boldsymbol{v_1}^{\mathrm{T}} C \, \boldsymbol{v_1} = \lambda_1$.

Es ist:

$$\boldsymbol{v_1}^{\mathrm{T}} C \, \boldsymbol{v_1} \overset{\boldsymbol{v_1}\text{EV}}{=} \boldsymbol{v_1}^{\mathrm{T}} \lambda_1 \, \boldsymbol{v_1} = \lambda_1 \, \boldsymbol{v_1}^{\mathrm{T}} \boldsymbol{v_1} = \lambda_1$$

und

$$\begin{aligned} \boldsymbol{v}^{\mathrm{T}} C \, \boldsymbol{v} &\overset{\boldsymbol{v_j}\,\mathrm{EV}}{=} \langle \sum_{i=1}^{n} a_i \, \boldsymbol{v_i}, \sum_{j=1}^{n} a_j \, \lambda_j \, \boldsymbol{v_j} \rangle = \sum_{i,j=1}^{n} \lambda_j \, a_i \, a_j \langle \boldsymbol{v_i}, \boldsymbol{v_j} \rangle \\ &\overset{\boldsymbol{v_i}\text{ONB}}{=} \sum_{i=1}^{n} \lambda_i \, a_i^2 \leq \lambda_1 \sum_{i=1}^{n} a_i^2 \overset{(2.1)}{=} \lambda_1 \; . \end{aligned}$$

zu b.) Da der Vektor $\boldsymbol{v}$ orthogonal zu $\boldsymbol{v_1}, \ldots, \boldsymbol{v_{j-1}}$ sein soll, kann man ihn darstellen als $\boldsymbol{v} = \sum_{i=j}^{n} a_i \, \boldsymbol{v_i}$. Analog zu a.) gilt:

$$\boldsymbol{v}^{\mathrm{T}} C \, \boldsymbol{v} = \sum_{i=j}^{n} \lambda_i \, a_i^2 \leq \lambda_j \sum_{i=j}^{n} a_i^2 = \lambda_j \; . \qquad \square$$

2.2 Bezeichnung
Die im Satz 2.1 nachgewiesenen Eigenschaften legen die Sprechweise nahe, dass in $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ *„sukzessive“ die Streuung maximal* ist.

2.3 Bemerkung
Die Basisvektoren des neuen Koordinatensystems sind die Richtungen $\boldsymbol{v_i}$, die man aus der Eigenwertzerlegung von C erhalten hat. Der Ursprung des Systems ist $\overline{\boldsymbol{x}}$.

2.4 Definition (Hauptkomponenten, Hauptrichtungen)
Die durch die Eigenwertzerlegung der Kovarianzmatrix C berechneten, normierten und nach Eigenwert sortierten Eigenvektoren $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$ nennt man *Hauptrichtungen*. Durch die Abbildung

$$\boldsymbol{x_j} \longmapsto \begin{pmatrix} \boldsymbol{v_1}^{\mathrm{T}}(\boldsymbol{x_j} - \overline{\boldsymbol{x}}) \\ \vdots \\ \boldsymbol{v_n}^{\mathrm{T}}(\boldsymbol{x_j} - \overline{\boldsymbol{x}}) \end{pmatrix} =: \begin{pmatrix} {k_j}^{(1)} \\ \vdots \\ {k_j}^{(n)} \end{pmatrix} =: \boldsymbol{k_j} \,, \qquad j \in \{1, \ldots, m\} \,,$$

werden die ursprünglichen Daten $\boldsymbol{x_j}$ in das neue Koordinatensystem transformiert. Die Koeffizienten ${k_j}^{(1)}, \ldots, {k_j}^{(n)}$ in der neuen Basis $\{\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}\}$ heißen die *Hauptkomponenten* zu $\boldsymbol{x_j}$.

Im Folgenden sei die Matrix $K := (\boldsymbol{k_1}, \ldots, \boldsymbol{k_m}) \in \mathbb{R}^{n \times m}$ die Koeffizientenmatrix der Punkte im neuen Koordinatensystem. Mit $\boldsymbol{k^{(1)}}, \ldots, \boldsymbol{k^{(n)}}$ werden die Zeilen der Matrix bezeichnet, sie beziehen sich auf die Richtungen $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n}$.

2.5 Satz (Eigenschaften der Hauptkomponenten)
Es gelten die folgenden Beziehungen:

a.) $\dfrac{1}{m} \displaystyle\sum_{j=1}^{m} k_j^{(i)} = 0 \qquad \forall\, i \in \{1, \ldots, n\} \,,$

b.) $\dfrac{1}{m-1} \displaystyle\sum_{j=1}^{m} \left(k_j^{(i)}\right)^2 = \lambda_i \qquad \forall\, i \in \{1, \ldots, n\} \,,$

c.) $\dfrac{1}{m-1} \displaystyle\sum_{j=1}^{m} k_j^{(i)}\, k_j^{(l)} = 0 \qquad \forall\, i \neq l,\ i, l \in \{1, \ldots, n\} \,.$

Vor dem Beweis zunächst noch eine Bemerkung zu dem vorliegenden Satz. Diese begründet auch die Wahl der jeweiligen Vorfaktoren.

2.6 Bemerkung
zu a.) Das arithmetische Mittel der i-ten Hauptkomponente von X ist Null.

zu b.) Die Varianz der Daten entlang der i-ten Hauptrichtung ist gleich dem entsprechenden Eigenwert λ_i. Da die Hauptrichtungen nach absteigendem Eigenwert sortiert sind, nimmt auch die Varianz der Daten in diesen Richtungen ab.

zu c.) Da das arithmetische Mittel der Hauptkomponenten nach a) stets Null ist, beschreibt die in c) angegebene Summe die Kovarianz zwischen der i-ten und l-ten Hauptrichtung. Diese ist für $i \neq l$ immer Null. Es besteht also kein linearer Zusammenhang mehr zwischen den verschiedenen Dimensionen, die durch die Hauptrichtungen gegeben sind.

Beweis
Es ist $C \in \mathbb{R}^{n\times n}$ die Kovarianzmatrix zu $X = (\boldsymbol{x_1}, \ldots, \boldsymbol{x_m}) \in \mathbb{R}^{n\times m}$. Die reellen Eigenwerte zu C sind $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ mit zugehörigen orthonormalen Eigenvektoren $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n} \in \mathbb{R}^n$. Der Vektor $\overline{\boldsymbol{x}} \in \mathbb{R}^n$ bezeichne das arithmetische Mittel von X.

zu a.) Für alle $i \in \{1, \ldots, n\}$ gilt:

$$\begin{aligned}
\frac{1}{m}\sum_{j=1}^{m} k_j^{(i)} &\overset{\text{Def.}}{=} \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{v_i}^{\mathrm{T}}(\boldsymbol{x_j} - \overline{\boldsymbol{x}}) \\
&= \boldsymbol{v_i}^{\mathrm{T}}\Big(\frac{1}{m}\sum_{j=1}^{m} \boldsymbol{x_j}\Big) - \boldsymbol{v_i}^{\mathrm{T}}\Big(\frac{1}{m}\sum_{j=1}^{m} \overline{\boldsymbol{x}}\Big) \\
&\overset{\text{Def.}}{=} \boldsymbol{v_i}^{\mathrm{T}}\,\overline{\boldsymbol{x}} - \boldsymbol{v_i}^{\mathrm{T}}\,\overline{\boldsymbol{x}}\,\Big(\frac{1}{m}\sum_{j=1}^{m} 1\Big) \\
&= 0\ .
\end{aligned}$$

zu b.) Es ist für alle $i \in \{1, \ldots, n\}$:

$$\begin{aligned}
\frac{1}{m-1}\sum_{j=1}^{m} \big(k_j^{(i)}\big)^2 &\overset{\text{Def.}}{=} \frac{1}{m-1}\sum_{j=1}^{m} \Big(\boldsymbol{v_i}^{\mathrm{T}}(\boldsymbol{x_j} - \overline{\boldsymbol{x}})\Big)^2 \\
&= \frac{1}{m-1}\sum_{j=1}^{m} \boldsymbol{v_i}^{\mathrm{T}}\,(\boldsymbol{x_j} - \overline{\boldsymbol{x}})\,\boldsymbol{v_i}^{\mathrm{T}}\,(\boldsymbol{x_j} - \overline{\boldsymbol{x}}) \\
&= \boldsymbol{v_i}^{\mathrm{T}}\Big(\frac{1}{m-1}\sum_{j=1}^{m} (\boldsymbol{x_j} - \overline{\boldsymbol{x}})(\boldsymbol{x_j} - \overline{\boldsymbol{x}})^{\mathrm{T}}\Big)\boldsymbol{v_i} \\
&\overset{\text{Def.}}{=} \boldsymbol{v_i}^{\mathrm{T}} C\,\boldsymbol{v_i} \\
&= \lambda_i\ .
\end{aligned}$$

zu c.) Für alle $i \neq l,\ i, l \in \{1, \ldots, n\}$ gilt:

$$\begin{aligned}
\frac{1}{m}\sum_{j=1}^{m} k_j^{(i)} k_j^{(l)} &\overset{\text{Def.}}{=} \frac{1}{m}\sum_{j=1}^{m} \boldsymbol{v_i}^{\mathrm{T}}(\boldsymbol{x_j} - \overline{\boldsymbol{x}})\boldsymbol{v_l}^{\mathrm{T}}(\boldsymbol{x_j} - \overline{\boldsymbol{x}}) \\
&= \boldsymbol{v_i}^{\mathrm{T}}\Big(\frac{1}{m}\sum_{j=1}^{m} (\boldsymbol{x_j} - \overline{\boldsymbol{x}})(\boldsymbol{x_j} - \overline{\boldsymbol{x}})^{\mathrm{T}}\Big)\boldsymbol{v_l} \\
&= \boldsymbol{v_i}^{\mathrm{T}} C\,\boldsymbol{v_l} \\
&= 0\ .
\end{aligned}$$

□

2.2 Transformation der Daten

Die Koeffizientenmatrix zur Darstellung der Punkte $\boldsymbol{x_1}, \ldots, \boldsymbol{x_m}$ im ursprünglichen Koordinatensystem ist $X \in \mathbb{R}^{n \times m}$. Die Matrix $K \in \mathbb{R}^{n \times m}$ enthält die Koeffizienten der Punkte im neuen Koordinatensystem, sie berechnet sich durch:

$$K = V^{\mathrm{T}}\left(X - \overline{\boldsymbol{x}}\, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\right) \quad \in \mathbb{R}^{n \times m} .$$

Dies ist die Matrixschreibweise zur Definition 2.4, die Hauptrichtungen bilden die Spalten der *Transformationsmatrix* $V \in \mathbb{R}^{n \times n}$.

2.2.1 Datenreduktion

Eine Reduktion der Datendimension erfolgt durch Projektion der Daten in den Unterraum der ersten $l < n$ Hauptrichtungen:

$$K_l = V_l^{\mathrm{T}} \left(X - \overline{\boldsymbol{x}}\, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}\right) \quad \in \mathbb{R}^{l \times m} .$$

Zur Berechnung der auf l Dimensionen reduzierten Koeffizientenmatrix K_l genügt die entsprechend reduzierte Transformationsmatrix $V_l \in \mathbb{R}^{n \times l}$, welche nur noch die ersten l Hauptrichtungen als Spalten besitzt.

2.2.2 Wahl der Dimension

Um eine starke Reduktion der Daten zu erreichen soll die Anzahl l der Hauptrichtungen möglichst klein sein. Andererseits sollen die ursprünglichen Daten möglichst exakt dargestellt werden. Dies bedeutet dass der Varianzanteil der reduzierten Daten an der Gesamtvarianz der ursprünglichen Daten möglichst groß ist. Nach Satz 2.1 zeigt $\boldsymbol{v_1}$ in Richtung der größten Varianz und nach Satz 2.5 ist diese gerade λ_1. Die Richtung $\boldsymbol{v_2}$ hat unter allen zu $\boldsymbol{v_1}$ orthonormalen Richtungen die größte Varianz λ_2 usw. Für die Wahl von l betrachtet man daher den Anteil der ersten l Varianzen an der Gesamtvarianz und definiert:

$$Var_{\mathrm{ratio}}(l) = \left(\sum_{i=1}^{l} \lambda_i\right) \left(\sum_{i=1}^{n} \lambda_i\right)^{-1} . \tag{2.2}$$

Je größer dieser Wert ist, umso exakter können die Datenpunkte rekonstruiert werden. Das l wird so gewählt, dass ein bestimmter, zuvor festgelegter Varianzanteil, erfüllt ist.

2.2.3 Rücktransformation

Da die Transformationsmatrix V eine unitäre Matrix ist, lassen sich bei gegebener Darstellung K der Punkte im neuen System die Originaldaten wieder sehr leicht berechnen. Es gilt nämlich:

$$X = \overline{\boldsymbol{x}}\, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} + V\, K . \tag{2.3}$$

Einen Punkt $\boldsymbol{x_j}$ der Originaldaten kann man dann darstellen als Mittelwert der Daten plus Linearkombination der Hauptkomponenten $k_j^{(i)}$ von $\boldsymbol{x_j}$ mit den entsprechenden Hauptrichtungen $\boldsymbol{v_i}$:

$$\boldsymbol{x_j} = \overline{\boldsymbol{x}} + \sum_{i=1}^{n} k_j^{(i)} \; \boldsymbol{v_i} \; . \tag{2.4}$$

Wurden allerdings die Daten reduziert, existiert nur noch die Matrix $K_l \in \mathbb{R}^{l \times m}$. Durch Anhängen von Nullzeilen kann man eine Matrix $\widetilde{K} \in \mathbb{R}^{n \times m}$ konstruieren und so mit der Rücktransformation aus (2.3) eine Darstellung $\widetilde{X}$ der reduzierten Daten im ursprünglichen Koordinatensystem erreichen:

$$\begin{aligned} \widetilde{X} &= \overline{\boldsymbol{x}} \; \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} + V \; \widetilde{K} \\ &= \overline{\boldsymbol{x}} \; \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} + V_l \; K_l \; . \end{aligned} \tag{2.5}$$

Wurde l so gewählt, dass der Varianzanteil $Var_{ratio}(l)$ groß ist, kann ein Datenpunkt $\boldsymbol{x_j}$ ohne großen Fehler dargestellt werden als:

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{j}} = \overline{\boldsymbol{x}} + \sum_{i=1}^{l} k_j^{(i)} \; \boldsymbol{v_i} \; .$$

Gespeichert werden muss also der Mittelwert der Daten $\overline{\boldsymbol{x}}$, die ersten l Hauptrichtungen $\boldsymbol{v_1}, \ldots, \boldsymbol{v_l}$ sowie die Hauptkomponenten $\boldsymbol{k^{(1)}}, \ldots, \boldsymbol{k^{(l)}}$.

2.3 Durchführung

1. Berechne die Kovarianzmatrix C.
2. Berechne die Eigenwerte und Eigenvektoren der Kovarianzmatrix.
3. Sortiere die Eigenwerte und dazugehörige Eigenvektoren nach Größe der Eigenwerte in absteigender Reihenfolge. Die sortierten Eigenwerte seien $\lambda_1, \ldots, \lambda_n$, die zugehörigen Eigenvektoren $\boldsymbol{v_1}, \ldots, \boldsymbol{v_n} \in \mathbb{R}^n$.
4. Berechne für $l = 1, 2, \ldots$ den Varianzanteil

   $$Var_{\mathrm{ratio}}(l) = \Big(\sum_{i=1}^{l} \lambda_i\Big) \; \Big(\sum_{i=1}^{n} \lambda_i\Big)^{-1}$$

   so lange, bis er eine vorgegebene Größe hat (z.B. 90%) und speichere dieses l.
5. Speichere die Transformationsmatrix

   $$V_l = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_l}) \in \mathbb{R}^{n \times l} \; .$$

6. Die reduzierte Datenmenge K_l berechnet man als:

$$K_l = V_l^{\mathrm{T}} \left(X - \overline{\boldsymbol{x}}\, \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} \right) \quad \in \mathbb{R}^{l \times m} \ .$$

Die Matrix K_l enthält die komprimierten Daten. Sollen diese im ursprünglichen System dargestellt werden, so muss noch die Transformationsmatrix V_l und der Mittelwert $\overline{\boldsymbol{x}}$ abgespeichert werden, um mit Gleichung (2.5) die Rücktransformation durchzuführen.

Die Hauptrichtungen $\boldsymbol{v_i}$ geben an, wo die Daten liegen. Ihr Index l, und noch genauer ihr Varianzanteil $Var_{\mathrm{ratio}}(l)$, gibt Aufschluss über die Relevanz der Richtung.

2.7 Bemerkung

Da die Hauptkomponentenanalyse skalierungsabhängig ist (siehe Kovarianzmatrix), ist es sinnvoll, die Datenpunkte vorher durch ihre Standardabweichung zu dividieren

$$\boldsymbol{x^{(i)}} \longmapsto \frac{\boldsymbol{x^{(i)}}}{\sigma\left(\boldsymbol{x^{(i)}}\right)} \ ,$$

und dadurch zu reskalieren. Ein Beispiel dazu findet man in Kapitel 4.2.2.

3 Support Vector Machines

Support Vector Machines (SVM) wurden ursprünglich zur Einordnung von Daten in zwei Klassen entwickelt. Die beiden Klassen sollen durch einen möglichst breiten Streifen, d.h. mit maximalem Abstand (maximal margin) voneinander getrennt werden.

Ausgangsbasis für den Ansatz der Support Vector Machines ist eine Menge von Trainingsdaten $\boldsymbol{x_i} \in X \subseteq \mathbb{R}^n$ für die jeweils bekannt ist, welcher Klasse $y_i \in Y \subseteq \{-1; 1\}$ sie angehören. Im Folgenden gehen wir von m Trainingsdaten $\boldsymbol{x_i}$ mit entsprechenden Labels y_i aus; sie bilden die Trainingsmenge $S = \left\{(\boldsymbol{x_i}, y_i)\right\}_{i=1}^{m}$. Die Zuordnung neuer Daten $\boldsymbol{x}$ zu einer Klasse soll über eine Entscheidungsfunktion $f(\boldsymbol{x})$ und deren Vorzeichen erfolgen. Gesucht ist ein geeignetes f.

Der Einstieg in das Thema erfolgt unter der Annahme linear separierbarer Daten (Abschnitt 3.1). Das Problem der maximalen Margin wird als Optimierungsproblem formuliert. Durch Übergang zum Lagrange Dualproblem und unter besonderer Berücksichtigung der KKT Bedingungen wird eine Darstellung der Entscheidungsfunktion als Linearkombination von nur wenigen Trainingsobjekten, den sogenannten Support Vektoren, entwickelt. Durch die Einführung von Kernen in Abschnitt 3.2 kann man den linearen Ansatz auf nichtlinear separierbare Daten übertragen.

Nicht nur die perfekte Trennung der Trainingsdaten ist wichtig. Ein sinnvoller Klassifikator sollte auch zu fremden Daten $\boldsymbol{x}$ die richtige Klasse y finden, d.h. eine Generalisierungseigenschaft besitzen. Auf dieses Problem wird in Abschnitt 3.3 eingegangen, die Modifikationen C SVM und ν SVM berücksichtigen das Problem „perfekte Trennung kontra gute Generalisierung“.

SVM lassen sich auch gut zur Regression verwenden. In der Tat entstand die Idee der ν SVM ursprünglich aus dem Regressionsansatz und wurde anschließend auf die Klassifikation übertragen.

3.1 Linear separierbare Daten

In diesem Abschnitt werden die Grundlagen für die Support Vector Machines entwickelt. Der Übergang zu nichtlinear separierbaren Daten erfolgt in Abschnitt 3.2 durch die Einführung von Kernen. Dort bildet die lineare Trennbarkeit einen Spezialfall mit dem Skalarprodukt als Kern.

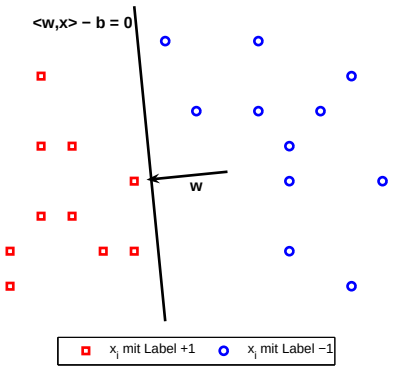


Abbildung 3.1: Trainingsdaten und trennende Hyperebene.

Trainingsdaten $\boldsymbol{x_i}$ mit positivem Label sollen von solchen mit negativem Label durch eine Hyperebene $H_{\boldsymbol{\omega},b}$ getrennt werden. Diese bestehe aus allen $\boldsymbol{x} \in \mathbb{R}^n$, welche der Gleichung

$$\langle \boldsymbol{\omega}, \boldsymbol{x} \rangle - b = 0$$

genügen. Hierbei ist $\boldsymbol{\omega} \in \mathbb{R}^n$ ein Vektor orthogonal zur Hyperebene. Ist dieser normiert, so gibt $\langle \boldsymbol{\omega}, \boldsymbol{x} \rangle$ die Länge der Orthogonalprojektion von $\boldsymbol{x}$ auf $\boldsymbol{\omega}$ an, $b \in \mathbb{R}$ ist dann der Abstand der Hyperebene zum Ursprung. Die *Entscheidungsfunktion* $f(\boldsymbol{x})$ für die Zuordnung neuer Daten zu einer Klasse wird mit Hilfe dieser Hyperebene definiert:

$$f(\boldsymbol{x}) := \langle \boldsymbol{\omega}, \boldsymbol{x} \rangle - b \ .$$

Je nach Vorzeichen der Entscheidungsfunktion wird $\boldsymbol{x}$ der Klasse $+1$ oder -1 zugeordnet. Schaut man sich die Abbildung 3.1 an, so erkennt man, dass es bei linear separierbaren Daten unendlich viele trennende Hyperebenen gibt; man kann die eingezeichnete Hyperebene verschieben und auch leicht drehen. Es stellt sich daher die Frage, welche der vielen möglichen Hyperebenen als optimal anzusehen sind.

3.1.1 Maximal Margin Hyperebene

Als *optimal trennende* Hyperebene bezeichnet man diejenige, die von beiden Klassen maximal weit entfernt ist. Dies hat den positiven Effekt, dass geringe Änderungen der Daten dann nicht direkt zu Fehlklassifikationen führen.

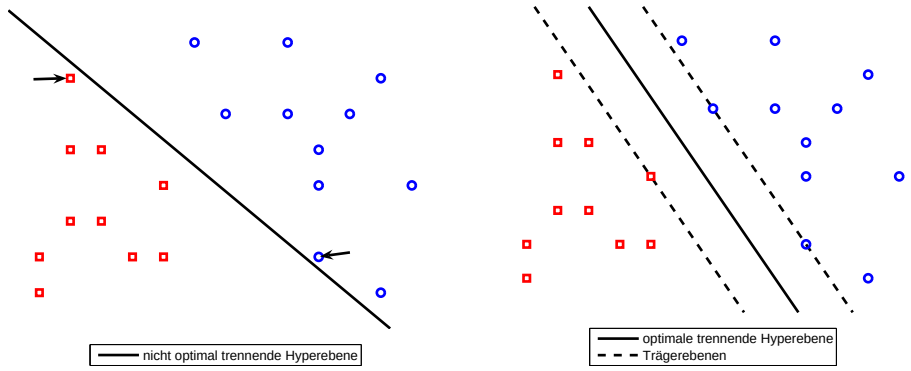


Abbildung 3.2: Veranschaulichung der optimal trennenden Hyperebene.

In der linken Abbildung sind zwei Trainingspunkte durch Pfeile gekennzeichnet, bei denen die Gefahr der Fehlklassifikation besteht. Ein neu zu klassifizierender Punkt, der einem dieser Trainingspunkte sehr ähnlich ist und ihm somit räumlich sehr nahe liegt, sollte auch der gleichen Klasse zugeordnet werden. Dies muss im linken Bild aber nicht der Fall sein, da der neue Punkt trotz seiner räumlichen Nähe auf der anderen Seite der Hyperebenen liegen kann. Bei der Hyperebene in der rechten Abbildung kann dieser Fall nicht eintreten, da dort die beiden Klassen durch einen breiten Streifen voneinander getrennt sind, kleine räumliche Abweichungen also nicht zu Fehlklassifikationen führen können. Eine optimale trennende Hyperebene bildet zusammen mit den beiden Trägerebenen einen maximal breiten Streifen zwischen den Klassen.

3.1 Definition (Margin bezüglich Hyperebene $H_{\boldsymbol{\omega},b}$)
Gegeben sei ein Punkt $(\boldsymbol{x_i}, y_i)$ und eine Hyperebene $H_{\boldsymbol{\omega},b}$. Mit $\widetilde{\boldsymbol{\omega}} = \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ und $\widetilde{b} = \frac{b}{\|\boldsymbol{\omega}\|}$ nennt man

$$\gamma_i = y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle - b \right) \qquad \textit{funktionale Margin}$$
$$\widetilde{\gamma}_i = y_i \left(\langle \widetilde{\boldsymbol{\omega}}, \boldsymbol{x_i} \rangle - \widetilde{b} \right) \qquad \textit{geometrische Margin}$$

eines Trainingspunktes $(\boldsymbol{x_i}, y_i)$ *bezüglich der Hyperebene* $H_{\boldsymbol{\omega},b}$.

3.2 Bemerkung
Ist $\gamma_i > 0$ so wurde $\boldsymbol{x_i}$ korrekt klassifiziert, da die Vorzeichen von $f(\boldsymbol{x_i})$ und y_i dann gleich sind. Da nur durch $\|\boldsymbol{\omega}\| > 0$ dividiert wurde, gilt dies auch für $\widetilde{\gamma}_i$. Der Wert $|\widetilde{\gamma}_i|$ ist der euklidische Abstand des Trainingspunktes $(\boldsymbol{x_i}, y_i)$ zur Hyperebene $H_{\boldsymbol{\omega},b}$.

3.3 Definition (Margin bezüglich Trainingsmenge S)
Die *geometrische Margin einer Hyperebene bezüglich einer Trainingsmenge S* ist definiert als

$$\min_{(\boldsymbol{x_i},y_i)\in S} \widetilde{\gamma}_i = \min_{(\boldsymbol{x_i},y_i)\in S} \left(y_i \left(\langle \widetilde{\boldsymbol{\omega}}, \boldsymbol{x_i} \rangle - \widetilde{b} \right) \right) .$$

Die *funktionale Margin einer Hyperebene bezüglich einer Trainingsmenge S* wird analog definiert.

3.4 Bemerkung
Treten Fehlklassifikationen auf, so ist das Minimum aus Definition 3.3 kleiner als Null. Die geometrische Margin der Hyperebene ist dann der negative Abstand zur größten Fehlklassifikation.

Treten keine Fehlklassifikationen auf, so ist dieses Minimum größer als Null. Die geometrische Margin der Hyperebene ist in diesem Fall der Abstand der Hyperebene zum nächstgelegenen Trainingspunkt. Die beiden Ebenen, die mit einem Abstand $\min\limits_{(\boldsymbol{x_i},y_i)\in S} \widetilde{\gamma}_i$ parallel zur Hyperebene $H_{\boldsymbol{\omega},b}$ liegen, werden auch *Trägerebenen* genannt.

3.5 Definition (Maximal Margin Hyperebene)
Die *Margin einer Trainingsmenge S* ist definiert als

$$\max_{\boldsymbol{\omega},b} \left(\min_{(\boldsymbol{x_i},y_i)\in S} \widetilde{\gamma}_i \right) = \max_{\boldsymbol{\omega},b} \left[\min_{(\boldsymbol{x_i},y_i)\in S} \left(y_i \left(\langle \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}, \boldsymbol{x_i} \rangle - \frac{b}{\|\boldsymbol{\omega}\|} \right) \right) \right] ,$$

d.h. als die größte aller geometrischen Margins über alle Hyperebenen. Eine Hyperebene, die dieses Bedingung erfüllt, nennt man *Maximal Margin Hyperebene*.

3.6 Bemerkung
Eine Maximal Margin Hyperebene ist diejenige mit größter geometrischer Margin bezüglich der Trainingsmenge. Bei linear trennbaren Daten ist dadurch schon sichergestellt, dass sie die Klassen voneinander trennt, siehe Bemerkung 3.4. Darüber hinaus ist sie eine optimal trennende Hyperebene, da sie größtmöglichen Abstand zu beiden Klassen hat und zusammen mit den beiden Trägerebenen einen Streifen größtmöglicher Breite zwischen die Klassen legt. Die Maximal Margin Hyperebene ist von beiden Klassen gleich weit entfernt.

3.1.2 Berechnung der Maximal Margin Hyperebene

Die Berechnung einer Maximal Margin Hyperebene $H_(\boldsymbol{\omega}, b)$ führt zu einem Optimierungsproblem über $(\boldsymbol{\omega}, b)$. Durch Lösung des Lagrange Dualproblems erhält man die Parameter für die Darstellung der trennenden Hyperebene.

Damit $(\boldsymbol{\omega}, b)$ die Daten $(\boldsymbol{x_i}, y_i) \in S$ entsprechend ihrer Klassen trennt, muss gelten:

$$\begin{aligned} \langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle - b < 0, &\quad \text{für } i \in \{1, \ldots, m\} \quad \text{mit } y_i = -1 \\ \langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle - b > 0, &\quad \text{für } i \in \{1, \ldots, m\} \quad \text{mit } y_i = +1 \ . \end{aligned}$$

Die beiden Fälle lassen sich zusammenfassen zu

$$y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle - b \right) > 0, \quad \text{für } i \in \{1, \ldots, m\}.$$

Ein solches $(\boldsymbol{\omega}, b)$ existiert, da die Daten als linear trennbar vorausgesetzt wurden.

Skalierung der Hyperebene

Hyperebenen sind invariant gegenüber Skalierungen, d.h. es beschreiben sowohl $(\boldsymbol{\omega}, b)$ als auch $(c\,\boldsymbol{\omega}, c\,b)$, $c \neq 0$ die gleiche Hyperebene. Aus diesem Grund werden $\boldsymbol{\omega}$ und b jetzt so reskaliert, dass Punkte $\boldsymbol{x_j}$, die am nächsten an der Hyperebene liegen, die Gleichung

$$\left| \langle \boldsymbol{\omega}, \boldsymbol{x_j} \rangle - b \right| = 1 \tag{3.1}$$

erfüllen. Diese Punkte sind nach Bemerkung 3.4 in der Trägerebene enthalten, wie in Abbildung 3.2 dargestellt. Über den tatsächlichen geometrischen Abstand der Trägerebenen von der trennenden Hyperebene und somit über die Breite der Margin sagt Gleichung (3.1) noch nichts aus. Sie liefert aber eine *kanonische Darstellung* der Hyperebene $H_{\boldsymbol{\omega},b}$ mit

$$y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle - b \right) \geq 1, \quad i \in \{1, \ldots, m\}$$

und gleichzeitig m lineare Ungleichheitsrestriktionen für das Optimierungsproblem.

Breite der Margin

Die Maximal Margin Hyperebene hat nach Bemerkung 3.6 von beiden Klassen die gleiche Entfernung. Daher existieren Punkte $\boldsymbol{x_1}$ mit Label $+1$ und $\boldsymbol{x_2}$ mit Label -1 die am nächsten an der Hyperebene liegen und die Gleichung (3.1) erfüllen. Die Breite des Streifens, der die beiden Klassen voneinander trennt (Größe der Margin), wird durch Projektion der Punkte $\boldsymbol{x_1}$ und $\boldsymbol{x_2}$ auf den Normalenvektor der Hyperebene $\frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}$ berechnet:

$$\begin{aligned} \langle \boldsymbol{x_1}, \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \rangle \; - \; \langle \boldsymbol{x_2}, \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \rangle &= \frac{1}{\|\boldsymbol{\omega}\|} \Big(\langle \boldsymbol{x_1}, \boldsymbol{\omega} \rangle - \langle \boldsymbol{x_2}, \boldsymbol{\omega} \rangle \Big) \\ &= \frac{1}{\|\boldsymbol{\omega}\|} \Big(1 + b - (b - 1) \Big) \\ &= \frac{2}{\|\boldsymbol{\omega}\|} \; . \end{aligned}$$

Die Größe der Margin soll maximal werden, damit kleine Störungen in den Daten nicht zu Fehlklassifikationen führen. Anstatt $\frac{2}{\|\boldsymbol{\omega}\|}$ zu maximieren kann auch $\frac{1}{2}\|\boldsymbol{\omega}\|^2$ minimiert werden, was zu einem konvexen Optimierungsproblem führt.

Das **primale Optimierungsproblem** lautet demnach:

$$\boxed{\begin{array}{ll} \min\limits_{\boldsymbol{\omega},b} & \frac{1}{2}\|\boldsymbol{\omega}\|^2 \\ \text{u. d. N.} & y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x_i}\rangle - b\right) \geq 1, \qquad i \in \{1,\dots,m\}\ . \end{array}} \tag{P}$$

Ein wichtiger Schritt bei Support Vector Machines ist der Übergang zum Lagrange Dualproblem

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \left(\inf_{\boldsymbol{\omega},b} L(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) \right) \\ &\text{u. d. N.} \quad \boldsymbol{\alpha} \geq 0 \end{aligned} \tag{3.2}$$

mit der Lagrange Funktion

$$L(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x_i}\rangle - b\right)\right)\ . \tag{3.3}$$

Die Komponenten des Primalproblems (P) sind $\boldsymbol{\omega}$ und b. Der Vektor $\boldsymbol{\alpha}$, mit den Lagrange Multiplikatoren α_i als Komponenten, bezieht sich auf das Lagrange Dualproblem. Da es keine Gleichheitsrestriktionen gibt, sind dies die einzigen Lagrange Multiplikatoren.

KKT Bedingungen

Existiert ein KKT Punkt $(\boldsymbol{\omega}^*, b^*, \boldsymbol{\alpha}^*)$, so ist es nach Bemerkung 1.27 egal, ob man das Primalproblem (P) über $(\boldsymbol{\omega}, b)$ minimiert oder das Lagrange Dualproblem (3.2) über $\boldsymbol{\alpha}$ maximiert. Die optimalen Werte stimmen dann überein und werden an der Stelle $(\boldsymbol{\omega}^*, b^*)$ bzw. $\boldsymbol{\alpha}^*$ angenommen. Aus diesem Grund werden nun die KKT Bedingungen für dieses spezielle Primalproblem und das entsprechende Lagrange Dualproblem formuliert. Betrachte hierzu auch Definition 1.24.

(KKT 1) wird in zwei Bedingungen gesplittet:

$$\begin{aligned} & \frac{\partial L}{\partial \boldsymbol{\omega}} = 0 && \wedge && \frac{\partial L}{\partial b} = 0 \\ \Leftrightarrow \quad & \boldsymbol{\omega} = \sum_{i=1}^{m} \alpha_i\, y_i\, \boldsymbol{x_i} && \wedge && \sum_{i=1}^{m} \alpha_i\, y_i = 0\ . \end{aligned} \tag{3.4}$$

Sie werden in das Dualproblem aufgenommen.

(KKT 2) liefert

$$1 \leq y_i\left(\langle \boldsymbol{\omega}, \boldsymbol{x_i}\rangle - b\right), \qquad i \in \{1,\dots,m\}\ . \tag{3.5}$$

Dies ist die Ungleichheitrestriktion für das Primalproblem (P).

(KKT 3) muss nicht berücksichtigt werden.

(KKT 4) kann man komponentenweise schreiben als

$$\alpha_i \geq 0, \qquad i \in \{1, \ldots, m\} \; . \tag{3.6}$$

Dies ist die Restriktion für das Lagrange Dualproblem.

(KKT 5) kann ebenfalls komponentenweise geschrieben werden:

$$\begin{aligned} & \sum_{i=1}^{m} \alpha_i \, g_i(\boldsymbol{\omega}, b) = 0 \\ \Leftrightarrow \qquad & \alpha_i \, g_i(\boldsymbol{\omega}, b) = 0, \qquad i \in \{1, \ldots, m\} \; . \end{aligned} \tag{3.7}$$

Mit $g_i(\boldsymbol{\omega}, b) = 1 - y_i \left(\langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle - b \right)$ folgt aus den Bedingungen (3.5) und (3.6) die Ungleichung $\alpha_i \, g_i(\boldsymbol{\omega}, b) \leq 0$ und damit die Äquivalenz.

Diese Bedingung stellt sicher, dass an einem KKT Punkt $(\boldsymbol{\omega}^*, b^*, \boldsymbol{\alpha}^*)$ die Lagrange Funktion den gleichen Wert wie die Zielfunktion des Primalproblems hat, siehe Gleichung (3.3).

3.7 Korollar
In einem KKT Punkt $(\boldsymbol{\omega}^*, b^*, \boldsymbol{\alpha}^*)$ gilt nach Gleichung (3.7) für alle $i \in \{1, \ldots, m\}$:

$$\alpha_i^* = 0 \qquad \vee \qquad g_i(\boldsymbol{\omega}^*, b^*) = 0 \; .$$

3.8 Bemerkung
Für Trainingsdaten $\boldsymbol{x_i}$ außerhalb des Marginbereiches gilt nach Bedingung (3.1) die Ungleichung $y_i \left(\langle \boldsymbol{\omega}^*, \boldsymbol{x_i} \rangle - b^* \right) > 1$ und somit $g_i(\boldsymbol{\omega}^*, b^*) \neq 0$. Nach Korollar 3.7 müssen die entsprechenden α_i^* gleich Null sein. Nur für die Trainingsdaten $\boldsymbol{x_j}$ auf den Trägerebenen gilt $g_j(\boldsymbol{\omega}^*, b^*) = 0$ und es besteht die Möglichkeit für $\alpha_j^* \neq 0$. Es können auch keine Daten innerhalb des Marginbereiches liegen, da die Nebenbedingungen von (P) dies verbieten. Schließlich kann man davon ausgehen, dass zahlreiche Trainingsdaten außerhalb der Margin liegen und somit die entsprechenden α_i^* den Wert Null annehmen. Die Lösung ist daher im Allgemeinen hinsichtlich $\boldsymbol{\alpha}$ dünn besetzt.

Lagrange Dualproblem

Einsetzen von Bedingung (3.4) in die Lagrange Funktion liefert:

$$\begin{aligned} L(\boldsymbol{\omega}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \sum_{i=1}^{m} \alpha_i\, y_i\, \langle \boldsymbol{\omega}, \boldsymbol{x_i} \rangle + \sum_{i=1}^{m} \alpha_i\, y_i\, b + \sum_{i=1}^{m} \alpha_i \\ &= \frac{1}{2} \langle \sum_{i=1}^{m} \alpha_i\, y_i\, \boldsymbol{x_i}, \sum_{j=1}^{m} \alpha_j\, y_j\, \boldsymbol{x_j} \rangle - \sum_{i=1}^{m} \alpha_i\, y_i \langle \sum_{j=1}^{m} \alpha_j\, y_j\, \boldsymbol{x_j}, \boldsymbol{x_i} \rangle + \sum_{i=1}^{m} \alpha_i \\ &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i\, \alpha_j\, y_i\, y_j\, \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle \ . \end{aligned}$$

Die Funktion hängt jetzt nur noch von $\boldsymbol{\alpha}$ ab, so dass das **Lagrange Dualproblem** folgende Form hat:

$$\boxed{\begin{array}{l} \max\limits_{\boldsymbol{\alpha} \in \mathbb{R}^m} \Big(\sum\limits_{i=1}^{m} \alpha_i - \frac{1}{2} \sum\limits_{i,j=1}^{m} \alpha_i\, \alpha_j\, y_i\, y_j\, \langle \boldsymbol{x_i}, \boldsymbol{x_j} \rangle \Big) \\ \\ \text{u. d. N.} \left\{ \begin{array}{rcll} \sum\limits_{i=1}^{m} \alpha_i\, y_i & = & 0 & \\ \alpha_i & \geq & 0, & i \in \{1, \ldots, m\} \ . \end{array} \right. \end{array}} \tag{D}$$

Hier entspricht die Dimension des Raumes, in dem optimiert wird, der Anzahl der Trainingsdaten m und ist dadurch kontrollierbar. Die Dimension des Primalproblems ist n, die Größe eines Trainingspunktes.

3.9 Bemerkung
Im Allgemeinen zieht man die Lösung des Lagrange Dualproblems der des Primalproblems vor, dies hat mehrere Gründe. Zum einen spielt die Dimension des Optimierungsroblems eine Rolle. Weiterhin lässt sich die Entscheidungsfunktion schnell berechnen, wenn sie nur noch in $\boldsymbol{\alpha}$ gegeben ist, da nur wenige $\alpha_i \neq 0$ sind. Der Hauptgrund liegt aber darin, dass in (D) die Trainingsdaten durch ein Skalarprodukt miteinander verbunden sind. Dies ebnet den Weg für die Einführung von Kernen.

3.1.3 Berechnung der Entscheidungsfunktion

Hat man eine optimale Lösung $\boldsymbol{\alpha}^*$ von (D) gefunden, wird $(\boldsymbol{\omega}^*, b^*)$ so berechnet, dass $(\boldsymbol{\omega}^*, b^*, \boldsymbol{\alpha}^*)$ ein KKT Punkt ist. Unter Verwendung der entsprechenden Bedingungen kann man $(\boldsymbol{\omega}^*, b^*)$ und folglich das gesamte Problem in Abhängigkeit von $\boldsymbol{\alpha}^*$ formulieren. Im Folgenden sei $I = \{i : \alpha_i^* \neq 0\}$.

Der Normalenvektor $\boldsymbol{\omega}^*$ der trennenden Hyperebene berechnet sich durch (3.4):

$$\boldsymbol{\omega}^* = \sum_{j \in I} \alpha_j^*\, y_j\, \boldsymbol{x_j} \ .$$

Für die Berechnung von b^* benutzt man Korollar 3.7, welches besagt, dass $g_i(\boldsymbol{\omega}^*) = 0$ für alle $i \in I$ gelten muss. Zunächst wird aus dieser Bedingung für jedes $i \in I$ ein b_i^* berechnet:

$$\begin{aligned} & g_i(\boldsymbol{\omega}^*) = 0 \\ \Leftrightarrow \quad & y_i \left(\langle \boldsymbol{\omega}^*, \boldsymbol{x_i} \rangle - b_i^* \right) = 1 \\ \Leftrightarrow \quad & \left(\langle \boldsymbol{\omega}^*, \boldsymbol{x_i} \rangle - b_i^* \right) = \frac{1}{y_i} \\ \Leftrightarrow \quad & \langle \boldsymbol{\omega}^*, \boldsymbol{x_i} \rangle - y_i = b_i^* \end{aligned} \tag{3.8}$$

mit $\frac{1}{y_i} = y_i$, da $y_i \in \{-1; 1\}$.
Wie in [2] bildet man über diese b_i^* das arithmetische Mittel, und erhält das gesuchte b^*.

$$b^* = \frac{1}{|I|} \sum_{i \in I} b_i^* \ .$$

Hierbei bezeichnet $|I|$ die Mächtigkeit der Menge I. Setzt man $\boldsymbol{\omega}^*$ in die Gleichung (3.8) ein, erhält man eine Darstellung, die nur noch die dualen Variablen enthält:

$$\boxed{b^* = \frac{1}{|I|} \sum_{i \in I} \Big(\sum_{j \in I} \alpha_j^* \, y_j \, \langle \boldsymbol{x_j}, \boldsymbol{x_i} \rangle - y_i \Big) \ .} \tag{3.9}$$

3.10 Bemerkung
Durch Einsetzen von b^* in die Bedingungen (3.4) bis (3.7) kann man nachrechnen, dass $(\boldsymbol{\alpha}^*, w^*, b^*)$ tatsächlich ein KKT Punkt und somit die optimale Lösung für (P) und (D) ist. Dabei stellt sich heraus, dass $b^* = b_k^* = b_l^*$ für alle $k \neq l \in I$ gilt. Man kann also b^* mit einem einzigen Trainingspunkt $\boldsymbol{x_i}$, $i \in I$ berechnen.

Die **Entscheidungsfunktion** hat folgende Form:

$$\boxed{f(\boldsymbol{x}) = \sum_{i \in I} \alpha_i^* \, y_i \, \langle \boldsymbol{x_i}, \boldsymbol{x} \rangle - b^* \ .} \tag{3.10}$$

3.1.4 Durchführung

1. Training der SVM:
   - Löse das Lagrange Dualproblem (D), die Lösung sei α_i^*.
   - Berechne damit b^* nach Gleichung (3.9) .
   - Die Entscheidungsfunktion $f(\boldsymbol{x})$ ist gegeben durch Darstellung (3.10).
2. Klassifikation:
   - Ein neuer Punkt $\boldsymbol{x}$ wird der Klasse $y = \operatorname{sgn}\big(f(\boldsymbol{x})\big)$ zugeordnet.

3.2 Nichtlinear separierbare Daten

Bisher wurden nur linear separierbare Trainingsdaten betrachtet. Die positiven Traingspunkte ließen sich von solchen mit negativem Label durch eine lineare Entscheidungsfunktion $f(\boldsymbol{x})$ trennen.

Jetzt soll dieser Ansatz auf nichtlinear trennbare Probleme erweitert werden. Dazu werden die Trainingsdaten $\boldsymbol{x_i}$ in einen hochdimensionalen Raum abgebildet, in der Hoffnung, die Bilder dort linear trennen zu können. Ist dies möglich, so ist das Urbild der linearen trennenden Hyperebene im Allgemeinen nicht mehr linear. Es wurde eine nichtlineare Trennung im ursprünglichen Raum durchgeführt.

Eine Begründung für diese Vorgehensweise liefert das Theorem von Cover. Es besagt, dass die Anzahl der möglichen linearen Trennungen mit der Dimension des Raumes wächst.

3.2.1 Theorem von Cover

Das Theorem inklusive Beweis, ist in [4] unter dem Namen „Function Counting Theorem“ zu finden.

3.11 Definition (Dichotomie, Vektoren in allgemeiner Lage)

1. Unter *Dichotomie* versteht man die Trennung einer Menge in zwei Untermengen, die sich gegenseitig ausschließen.

2. Es befinden sich m Vektoren *in allgemeiner Lage* im n-dimensionalen Raum, wenn jede n-elementige Teilmenge von Vektoren linear unabhängig ist.

3.12 Korollar

Ist eine Menge von Vektoren in allgemeiner Lage, dann spannen jeweils n der m Vektoren den n-dimensionalen Raum auf. Jeweils $k \leq n$ viele Vektoren erzeugen einen k-dimensionalen Unterraum.

3.13 Satz (Theorem von Cover)

Die Anzahl der linearen Dichotomien von m Punkten in allgemeiner Lage im n-dimensionalen euklidischen Raum beträgt:

$$2\sum_{i=0}^{n-1} \binom{m-1}{i} .$$

Das Theorem umfasst auch den Fall $m \leq n$. Die angegebene Summe geht dann nur bis $m-1$, nach dem Binomischen Lehrsatz ist sie daher gleich 2^{m-1}. Es gibt somit 2^m homogene lineare Dichotomien, dies sind sämtliche Trennungen, die mit m vielen Punkten möglich sind.

3.2.2 Die implizite Abbildung in den Featureraum

Wie in der Einleitung dieses Kapitels erwähnt, werden bei nichtlinear trennbaren Problemen die Daten $\boldsymbol{x_i}$ durch eine Funktion $\Phi : X \subset \mathbb{R}^n \to \Phi(X)$ in einen meist höherdimensionalen Raum abgebildet. In diesem Zusammenhang benutzt man häufig die folgenden Begriffe.

3.14 Definition (Attribute, Eingaberaum, Feature, Featureraum)
Der Definitionsbereich X der Funktion Φ ist der *Eingaberaum*. In diesem befinden sich die Daten $\boldsymbol{x_i}$, welche auch *Attribute* genannt werden. Der Wertebereich $\Phi(X)$ wird als *Featureraum* bezeichnet, in ihm befinden sich die *Features* $\Phi(\boldsymbol{x_i})$.

Vorgehensweise

1. Transformiere die Attribute $\boldsymbol{x_i}$ in den Featureraum. Dies geschieht durch eine feste, nichtlineare Funktion $\Phi : X \to \Phi(X)$.

2. Trainiere die SVM mit den Bilddaten $\Phi(\boldsymbol{x_1}), \ldots \Phi(\boldsymbol{x_m})$; die linear trennende Hyperebene befindet sich im Featureraum.

3. Auch neu zu klassifizierende Punkte $\boldsymbol{x}$ müssen erst transformiert werden; die Klassifikation erfolgt durch das Vorzeichen von $f\big(\Phi(\boldsymbol{x})\big)$.

Um die Schritte 2. und 3. durchführen zu können, muss allerdings der Featureraum ein Skalarprodukt besitzen. Hat man darüberhinaus noch die Möglichkeit, dieses Skalarprodukt direkt als Funktion der Attribute $\boldsymbol{x}$ zu berechnen, kann man eine nichtlineare Entscheidungsfunktion im Eingaberaum konstruieren.

3.15 Definition (Kern)
Ein Kern ist eine Funktion $k : X \times X \to \mathbb{R}$, so dass für alle $\boldsymbol{u}, \boldsymbol{v} \in X \subset \mathbb{R}^n$ gilt:

$$k(\boldsymbol{u}, \boldsymbol{v}) = \langle \Phi(\boldsymbol{u}), \Phi(\boldsymbol{v}) \rangle \ .$$

Hierbei ist Φ die Abbildung von X in einen Featureraum mit Skalarprodukt.

3.16 Bemerkung
Durch die Kernfunktion ist es nun sehr einfach, eine SVM für nichtlinear trennbare Probleme zu formulieren. Man muss bloß im Optimierungsproblem (D) und in der Entscheidungsfunktion f aus Gleichung (3.10) das Skalarprodukt durch die Kernfunktion ersetzen und wie in Abschnitt 3.1.4 vorgehen. Eine Auswertung der Funktion Φ findet nicht mehr statt. Daher taucht die hohe Dimension des Featureraumes in der Berechnung nicht mehr auf, ein hoher Rechenaufwand wird vermieden.

Der Schlüssel zu diesem Ansatz ist die Wahl einer geeigneten Kernfunktion. Diese soll sich effizient berechnen lassen und für die jeweilige Anwendung hinsichtlich der Trennbarkeit geeignet sein.

3.17 Beispiel
Gegeben sind Trainingspunkte $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$, *die nicht linear trennbar sind. Durch die Funktion*

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3, \qquad (x_1, x_2) \mapsto (y_1, y_2, y_3)$$

mit $y_1 = x_1^2$, $y_2 = \sqrt{2}\, x_1\, x_2$ *und* $y_3 = x_2^2$, *werden die Daten in den dreidimensionalen Featureraum abgebildet. Dort ist eine lineare Trennung möglich. Der linearen Hyperebene im* $\mathbb{R}^3$ *entspricht eine Kreislinie im* $\mathbb{R}^2$.

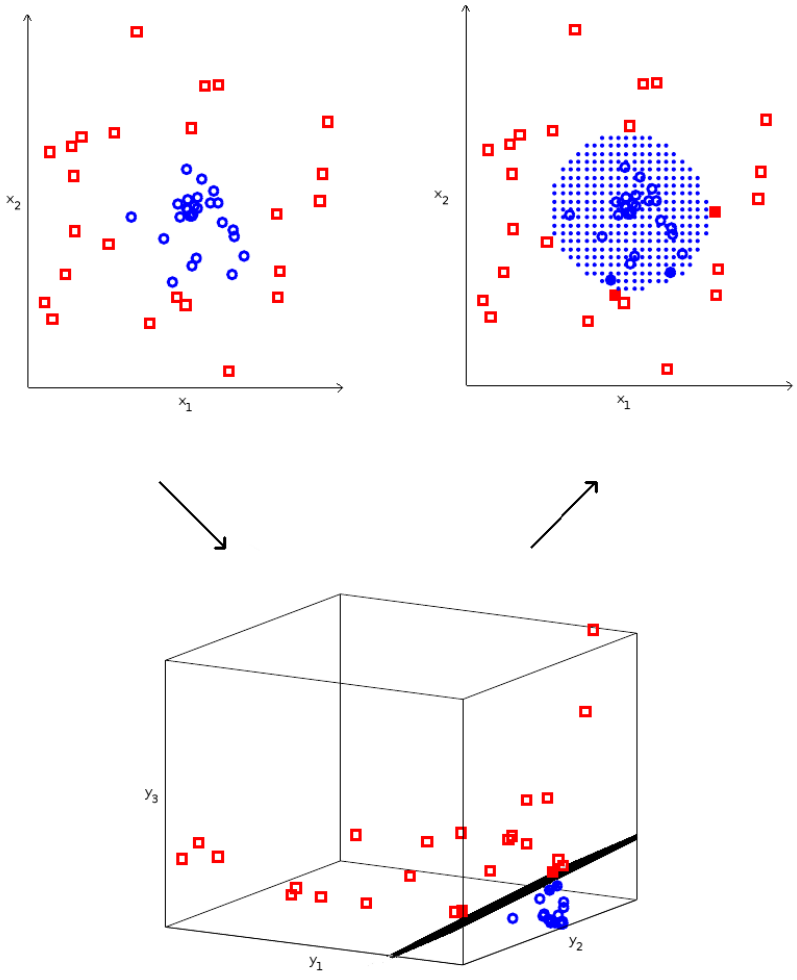


Abbildung 3.3: Funktionsweise einer SVM mit $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}\, x_1\, x_2, x_2^2)$

In Abbildung 3.3 wurde mit 50 Punkten trainiert. Nur vier davon sind Support Vektoren, dargestellt durch die beiden, ausgefüllten roten Quadrate und blauen Kreise. Außerdem ist

$$\langle \Phi(\boldsymbol{u}), \Phi(\boldsymbol{v}) \rangle_2 = u_1^2\, v_1^2 + 2\, u_1\, u_2\, v_1\, v_2 + u_2^2\, v_2^2 = \langle \boldsymbol{u}, \boldsymbol{v} \rangle_2^2 \ .$$

Der zu Φ gehörige Kern ist daher $k(\boldsymbol{u}, \boldsymbol{v}) = \langle \boldsymbol{u}, \boldsymbol{v} \rangle^2$. Den Umweg über die Funktion Φ kann man sich also sparen und direkt eine SVM mit dem angegebenen Kern verwenden.

3.18 Beispiel (Kerne)

Gauß RBF Kern	$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\big(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2}\big)$
Polynomialkern	$k(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + a)^n$
Sigmoid Kern	$k(\boldsymbol{x}, \boldsymbol{y}) = \tanh(b\, \langle \boldsymbol{x}, \boldsymbol{y} \rangle + c\,)$.

Hierbei sind $a, b, c, \sigma \in \mathbb{R}$ und $n \in \mathbb{N}$. In Anhang A findet man die Anwendung verschiedener Kerne auf zweidimensionale Daten.

Ist eine Abbildung Φ in einen Featureraum mit Skalarprodukt gegeben, lässt sich der entsprechende Kern mit Definition 3.15 berechnen. Nach Bemerkung 3.16 verwendet man sowohl für das Training als auch für die Klassifikation nichtlinear trennbarer Daten mit einer SVM nur noch den Kern. Aus diesem Grund möchte man den Kern direkt definieren, sodass die Abbildung Φ nicht mehr bekannt sein muss. Es stellt sich die Frage, wann eine Funktion k ein Kern ist. Eine Antwort liefert folgendes Theorem [3].

3.19 Satz (Theorem von Mercer)
Seien $X \subset \mathbb{R}^n$ kompakt und k eine stetige symmetrische Funktion, so dass

$$T_k : L_2(X) \to L_2(X) \ ; \qquad (T_k f)(\cdot) = \int_X k(\cdot, \boldsymbol{z})\, f(\boldsymbol{z})\, d\boldsymbol{z}$$

positiv ist, d.h. es gilt für alle $f \in L_2(X)$:

$$\int_{X \times X} k(\boldsymbol{x}, \boldsymbol{z})\, f(\boldsymbol{x})\, f(\boldsymbol{z})\, d\boldsymbol{x}\, d\boldsymbol{z} \geq 0 \ .$$

Dann kann $k(\boldsymbol{x}, \boldsymbol{z})$ auf $X \times X$ in eine gleichmäßig konvergente Reihe

$$k(\boldsymbol{x}, \boldsymbol{z}) = \sum_{i=1}^{\infty} \lambda_i\, \Phi_i(\boldsymbol{x})\, \Phi_i(\boldsymbol{z})$$

mit normalisierten Eigenfunktionen $\Phi_i \in L_2(X)$ und entsprechenden positiven Eigenwerten $\lambda_i \geq 0$ entwickelt werden.

3.20 Bemerkung
Das Theorem liefert die Featureabbildung

$$\Phi : \boldsymbol{x} \to \Big(\Phi_i(\boldsymbol{x})\Big)_{i=1}^{\infty}$$

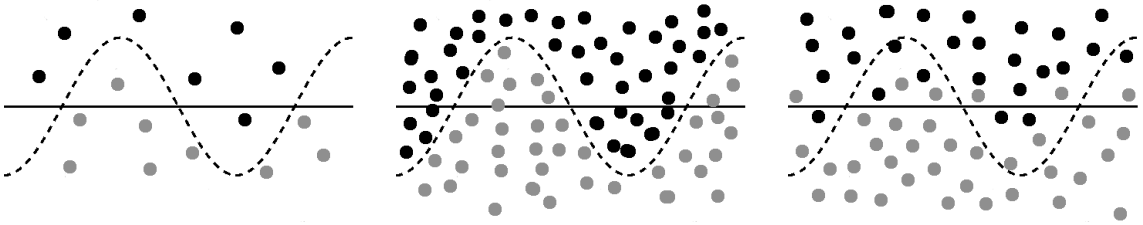
zum Kern k. Näheres zu Kernen, deren Featureabbildungen und dazu passenden Hilberträumen steht in [3] und [11].

3.3 Zulassen von Fehlklassifikationen

Nicht immer ist eine fehlerfreie Trennung der Trainingsdaten ein erstrebenswertes Ziel, besonders wenn man von verrauschten Daten ausgeht. Im schlimmsten Fall läuft eine exakte Trennung auf eine reine Fallunterscheidung hinaus, die Generalisierungseigenschaft geht dadurch völlig verloren. Aber auch der Rechenaufwand, beispielsweise für das Lösen des Optimierungsproblems, kann durch den Anspruch auf exakte Trennung sehr hoch werden. Aus diesem Grund möchte man Fehlklassifikationen zulassen, welche jedoch nicht zu schwerwiegend sein sollen.

3.3.1 Fehleranalyse

Durch die Einführung der Kerne ist es möglich, die Trainingsdaten nichtlinear zu trennen. Je nach Kern und damit verbundenem Featureraum sind sehr komplexe trennende Hyperebenen möglich. Man möchte jedoch nicht unter allen Umständen eine fehlerfreie Trennung erreichen. Sind die Trainingsdaten leicht verrauscht, kann eine perfekt trennende Hyperebene sehr komplex werden. Im Allgemeinen ist man an einer möglichst einfachen Hyperebene mit möglichst wenig Fehlklassifikationen interessiert, d.h. **niedrige Komplexität bei kleinem Fehler**.



(a) Wenige Trainingsdaten.

(b) Die gerade Linie verursacht Underfitting.

(c) Die sinusförmigen Linie verursacht Overfitting.

Die Abbildungen veranschaulichen die oben genannte Problematik. Es seien Trainingsdaten wie in (a) gegeben. Beide Trennlinien sind möglich. Die sinusförmige ist komplexer als die gerade Trennlinie, verursacht jedoch keinen Trainingsfehler. Sind die realen Daten wie in (b) verteilt, so ist eine höhere Komplexität sinnvoll. Die gerade Trennlinie trennt nicht gut genug. Bei einer Verteilung nach (c) ist dies nicht der Fall. Die leichten Fehlklassifikationen bei der geraden Linie können durch Rauschen entstanden sein. Die sinusförmige Linie trennt nicht besser, eine höhere Komplexität bewirkt keine bessere Trennung.

Um die Fehlklassifikationen handhaben zu können, führen wir den folgenden Begriff ein.

3.21 Definition (Lossfunktion)
Gegeben sei ein Paar $(\boldsymbol{x}, y) \in X \times Y$ und eine Entscheidungsfunktion f. Eine stetige Funktion

$$V : f(X) \times Y \to \mathbb{R}_0^+ \quad \text{mit} \quad V(y, y) = 0$$

wird *Lossfunktion* genannt.

3.22 Beispiel (Lossfunktionen)

L_2 Loss:	$V(f(\boldsymbol{x}), y) = (f(\boldsymbol{x}) - y)^2$
L_1 Loss:	$V(f(\boldsymbol{x}), y) = \|f(\boldsymbol{x}) - y)\|_1$
0/1 Loss:	$V(f(\boldsymbol{x}), y) = \theta(-yf(\boldsymbol{x})), \qquad \theta(y) = \begin{cases} 1 & y \geq 0 \\ 0 & y < 0 \end{cases}$
SVM Loss:	$V(f(\boldsymbol{x}), y) = (1 - f(\boldsymbol{x})y)_+$
ε-intensive Loss:	$V(f(\boldsymbol{x}), y) = ((\|f(\boldsymbol{x}) - y)\| - \varepsilon)_+$

Dabei ist $(m)_+ := \max\{m, 0\}$ für $m \in \mathbb{R}$. Die SVM Lossfunktion ist der Idee der maximal trennenden Hyperebene angepasst. Sie beginnt ab der Trägerebene zu wirken. Je weiter man sich von ihr aus in die falsche Richtung bewegt, umso größer wird der Loss Wert. Befindet man sich innerhalb der Margin auf der richtigen Seite der trennenden Ebene, so liegt der Wert der Lossfunktion im Intervall $(0, 1)$. Bei Fehlklassifikationen ist der Wert größer als 1 und hängt von der Entfernung des entsprechenden Punktes $\boldsymbol{x}$ von der trennenden Hyperebene ab.

3.23 Definition (Generalisierungsfehler, Trainingsfehler)
Gegeben sei die Trainingsmenge $S = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^m$. Weiter existiere eine unbekannte Wahrscheinlichkeitsverteilung $P(\boldsymbol{x}, y)$, aus der die gegebenen Trainingsdaten stammen. Der *Generalisierungsfehler* des Klassifizierers ist gegeben durch

$$R_{\text{gen}}(f) = \int V(f(\boldsymbol{x}), y)\ dP(\boldsymbol{x}, y)\ ,$$

der *Trainingsfehler* ist der empirische Fehler

$$R_{\text{emp}}(f) = \frac{1}{m} \sum_{i=1}^{m} V(f(\boldsymbol{x_i}), y_i)\ .$$

Ein guter Klassifizierer soll den Generalisierungsfehler auf neuen Daten minimieren. Dieser kann jedoch nicht explizit berechnet werden, da die Wahrscheinlichkeitsverteilung $P(\boldsymbol{x}, y)$ nicht bekannt ist.

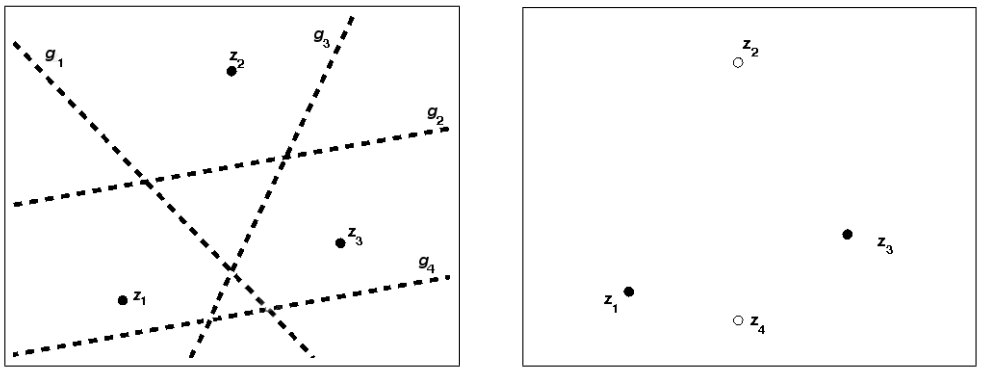
Bisher wurde nur die Darstellung des Fehlers berücksichtigt. Die von Vapnik in [14] entwickelte VC Theorie erfasst die Komplexität einer Funktionenklasse in der VC Dimension und liefert eine obere Schranke für den Generalisierungsfehler.

3.24 Definition (VC Dimension)
Gegeben sei eine Funktionenklasse F. Zu m vielen Vektoren gibt es 2^m verschiedene Möglichkeiten, Labels aus $\{-1, +1\}$ zuzuordnen. Entsprechend dieser Labels sollen die Vektoren durch Funktionen aus F getrennt werden.
Die *VC Dimension einer Funktionenklasse* F ist die maximale Anzahl von Vektoren, bei der sich alle der 2^m möglichen Zuweisungen von Labels durch Funktionen aus F trennen lassen. Existiert kein solches m, so ist die VC Dimension unendlich.

3.25 Beispiel (VC Dimension)
Betrachte den Raum $\mathbb{R}^2$ *und die Funktionenklasse* $F = \{H_{\boldsymbol{\omega},b} : \boldsymbol{\omega} \in \mathbb{R}^2, b \in \mathbb{R}\}$ *der Geraden in* $\mathbb{R}^2$. $H_{\boldsymbol{\omega},b}$ *sei die Menge aller Hyperebenen, vergleiche dazu Abschnitt 3.1.*

- *Für nur einen Punkt ergibt eine lineare Trennung keinen Sinn.*
- *Zwei Punkte sind linear trennbar, bei allen 4 möglichen Zuordnungen von Labels.*
- *Hat man drei Punkte, so veranschaulicht Abbildung 3.4(a) die Möglichkeiten der linearen Trennung, sofern die Punkte nicht kollinear sind. Gerade* g_1 *trennt, wenn* z_1 *ein anderes Label besitzt als* z_2 *und* z_3. *Für* g_2 *und* g_3 *gilt sinngemäß das gleiche. Haben alle Punkte gleiches Label, so trennt Gerade* g_4.
- *Vier Punkte, egal wie sie im* $\mathbb{R}^2$ *angeordnet sind, lassen sich nicht mehr allgemein trennen. Es gibt immer eine Zuordnung von Labels, welche die lineare Trennung unmöglich macht. Als Beispiel betrachte Abbildung 3.4(b)*

Die VC Dimension von F ist demnach 3.



(a) Lineare Trennung immer möglich. (b) Lineare Trennung unmöglich.

Abbildung 3.4: Bestimmung der VC Dimension von Geraden im $\mathbb{R}^2$.

3.26 Bemerkung
Bei den Support Vector Machines ist die Funktionenklasse die Menge aller Hyperebenen

$$H_{\boldsymbol{\omega},b,k} = \{\boldsymbol{x} \in X : k(\boldsymbol{\omega}, \boldsymbol{x}) - b = 0\} ,$$

mit $\boldsymbol{\omega} \in X, b \in \mathbb{R}$ und Kern $k : X \times X \to \mathbb{R}$. Die Wahl des Kernes beeinflusst die VC Dimension mit der durch ihn gegebenen Featureraumabbildung Φ. Für lineare Hyperebenen in einem hochdimensionalen Featureraum ist die VC Dimension im Allgemeinen größer als bei linearen Hyperebenen in einem Featureraum von niedrigerer Dimension.

Man kann nun eine obere Schranke für den Generalisierungsfehler angeben.

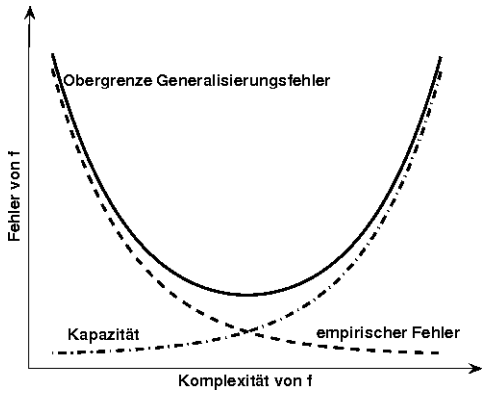
3.27 Satz (Obere Schranke für Generalisierungsfehler)
Für alle $\eta \geq 0, f \in F$ mit VC Dimension h gilt mit einer Wahrscheinlichkeit von mindestens $1 - \eta$

$$R_{\text{gen}}(f) \leq R_{\text{emp}}(f) + C(f, \eta, h)$$

mit einer Kapazitätsfunktion $C(f, \eta, h)$.

Beweis siehe [9], [14] und [15]. □

3.28 Bemerkung
Die Kapazitätsfuntion $C(f, \eta, h)$ ist ein Maß für die Ausdrucksfähigkeit des Klassifizierers f. Ein ausdrucksstarker Klassifizierer verursacht einen kleinen empirischen Fehler, die Kapazität ist jedoch groß. Bei einem schwachen Klassifizierer ist die Kapazität klein aber der empirische Fehler groß.



Die Idee der Kapazität hängt sehr eng mit der Komplexität zusammen. Tatsächlich kann man mit Hilfe der VC Theorie die Funktion $C(f, \eta, h)$ in geschlossener Form angeben. Die Komplexität wird durch die VC Dimension h dargestellt. Obwohl die Kapazität für die verschiedenen Loss Funktionen unterschiedlich ist, gilt stets, dass mit steigender Komplexität h des Klassifizierers f auch dessen Kapazität $C(f, \eta, h)$ wächst. Beispiele für Kapazitätsfunktionen findet man in [11].

Der empirische Fehler gibt an, wie gut die Trainingsdaten klassifiziert werden. Ein Klassifikator f mit hoher Komplexität führt im Allgemeinen zu einem kleinen empirischen Fehler. Es dürfen also weder der empirische Fehler noch der Kapazitätsterm zu groß werden. Daher ist es ratsam, schon in den Trainingsdaten Fehlklassifikationen zuzulassen, um die Kapazität und damit auch den Generalisierungsfehler klein zu halten. Eine Funktionenklasse F mit geeigneter Komplexität muss gefunden werden.

3.3.2 C Support Vector Machine

Der klassische Ansatz für SVM, der durch (P) realisiert wird, verbietet schon durch die Nebenbedingung

$$y_i \left(\langle \boldsymbol{\omega}, \Phi(\boldsymbol{x_i}) \rangle - b \right) \geq 1, \qquad \forall\, i \in \{1, \ldots, m\}$$

das Auftreten von Fehlklassifikationen. Um diese Enschränkung zu lockern, führt man *Schlupfvariablen* ξ_i ein :

$$y_i \left(\langle \boldsymbol{\omega}, \Phi(\boldsymbol{x_i}) \rangle - b \right) \geq 1 - \xi_i, \qquad \text{mit } \xi_i \geq 0,\ \forall\, i \in \{1, \ldots, m\}.$$

3.29 Bemerkung
Jede Schlupfvariable ξ_i bildet eine Obergrenze für die Größe der Überschreitung der Margin und damit auch eine Obergrenze für den SVM Loss $V(f(\boldsymbol{x_i}), y)$. Vergleiche dazu Beispiel 3.22. Ist $0 < \xi_i < 1$, so darf $\boldsymbol{x_i}$ innerhalb der Margin auf der richtigen Seite der trennenden Hyperebene sein. Ist $\xi_i > 1$, so darf die trennende Hyperebene überschritten werden.

Um das Maß an Fehlklassifikationen gering zu halten, addiert man zu der ursprünglichen Zielfunktion $\frac{1}{2}\|\boldsymbol{\omega}\|^2$ die Obergrenze $\sum_{i=1}^{m} \xi_i$ für den empirischen Fehler, und berücksichtigt bei der Minimierung die Variable $\boldsymbol{\xi}$. Als neues Optimierungsproblem ergibt sich:

$$\boxed{\begin{array}{l} \min\limits_{\boldsymbol{\omega}, b, \boldsymbol{\xi}} \quad \dfrac{1}{2}\|\boldsymbol{\omega}\|^2 + \dfrac{C}{m} \displaystyle\sum_{i=1}^{m} \xi_i \\ \\ \text{u. d. N.} \quad \left\{ \begin{array}{rcll} y_i \left(\langle \boldsymbol{\omega}, \Phi(\boldsymbol{x_i}) \rangle - b \right) & \geq & 1 - \xi_i, & \\ \xi_i & \geq & 0, & i \in \{1, \ldots, m\}\ . \end{array} \right. \end{array}} \qquad (\mathrm{P_C})$$

3.30 Bemerkung
Die Regularisierungskonstante C regelt das Zusammenspiel zwischen empirischem Fehler und Komplexitätsterm. Bei großem C haben Fehlklassifikationen ein starkes Gewicht, dem empirischen Fehler wird eine große Bedeutung zugewiesen. Ist C klein, so werden Fehlklassifikationen weniger stark gewichtet, die Nebenbedingungen spielen eine größere Rolle. In ihnen steckt die Featureabbildung und somit nach Bemerkung 3.26 auch implizit die VC Dimension.

Über die Lagrange Funktion

$$L(\boldsymbol{\omega}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r}) = \frac{1}{2}\|\boldsymbol{\omega}\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left(y_i\left(\langle\boldsymbol{\omega}, \Phi(\boldsymbol{x_i})\rangle - b\right) - 1 + \xi_i\right) - \sum_{i=1}^{m} r_i\,\xi_i$$

mit $\mathbf{r} = (r_1, \ldots, r_m)^{\mathrm{T}}$, erhält man analog zum Vorgehen in Abschnitt 3.1.2 das Lagrange Dualproblem zu ($\mathrm{P_C}$):

$$\boxed{\begin{array}{l}\displaystyle\max_{\boldsymbol{\alpha}\in\mathbb{R}^m}\left(\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\,\alpha_j\,y_i\,y_j\,k(\boldsymbol{x_i}, \boldsymbol{x_j})\right)\\[2ex] \text{u. d. N.}\ \left\{\begin{array}{rcll}\displaystyle\sum_{i=1}^{m}\alpha_i\,y_i & = & 0 & \\ \frac{C}{m} \geq \alpha_i & \geq & 0, & i \in \{1, \ldots, m\}\ .\end{array}\right.\end{array}} \qquad (\mathrm{D_C})$$

Die Entscheidungsfunktion berechnet sich wie bei der Support Vector Machine ohne Fehlklassifikation (3.10). Das Skalarprodukt muss nur durch die Kernfunktion zu ersetzt werden.

3.31 Bemerkung
Die Bedingung $\frac{\partial L}{\partial \xi_i} = 0$ aus (KKT 1) führt zu einer Darstellung der Lagrange Multiplikatoren r_i in Abhängigkeit von α_i:

$$r_i = \frac{C}{m} - \alpha_i, \qquad i \in \{1, \ldots, m\}\ . \qquad (3.11)$$

Zusammen mit $r_i \geq 0$ erscheint diese Abhängigkeit in der Nebenbedingung $\alpha_i \leq \frac{C}{m}$, der so genannten *Box Bedingung*.

Betrachtung von $\boldsymbol{\alpha}$

Im Folgenden wollen wir auch für die C Support Vector Machine zeigen, dass viele der α_i gleich Null sind. Die Bedingung $\boldsymbol{\xi} \geq 0$ aus ($\mathrm{P_C}$) führt zu m zusätzlichen Ungleichungen. Die Ungleichheitsrestriktionen lauten jetzt:

$$\text{und} \qquad \begin{aligned} g_{1,i}(\boldsymbol{\omega}^*, b^*, \boldsymbol{\xi}^*) &= 1 - \xi_i^* - y_i\left(\langle\boldsymbol{\omega}^*, \Phi(\boldsymbol{x_i})\rangle - b^*\right)\\ g_{2,i}(\boldsymbol{\omega}^*, b^*, \boldsymbol{\xi}^*) &= \quad -\xi_i^* \end{aligned}$$

für $i \in \{1, \ldots, m\}$.

Mit (KKT 5) ergibt sich eine Korollar 3.7 entsprechende Aussage.

3.32 Korollar
In einem KKT Punkt $(\boldsymbol{\omega}^*, b^*, \boldsymbol{\xi}^*, \boldsymbol{\alpha}^*, \mathbf{r}^*)$ *gelten mit* $f(\boldsymbol{x}) = \langle \boldsymbol{\omega}^*, \Phi(\boldsymbol{x}) \rangle - b^*$ *:*

$$\alpha_i^* = 0 \quad \vee \quad 1 - \xi_i^* = y_i\, f(\boldsymbol{x_i}) \tag{3.12}$$

und

$$r_i^* = 0 \quad \vee \quad \xi_i^* = 0 \tag{3.13}$$

für $i \in \{1, \ldots, m\}$.

Mit einer Fallunterscheidung bezüglich der Variablen α_i^* gewinnt man Erkenntnisse über die Lage der entsprechenden Punkte $\boldsymbol{x_i}$.

Fall 1: $\alpha_i^* = 0$

Mit Gleichung (3.11) gilt: $r_i^* = \frac{C}{m} \overset{(3.13)}{\Longrightarrow} \xi_i^* = 0$.

Daraus folgt mit ($\mathrm{P_C}$): $y_i\, f(\boldsymbol{x_i}) \geq 1$,

der Punkt $\boldsymbol{x_i}$ befindet sich außerhalb der Margin oder auf der Trägerebene und wurde korrekt klassifiziert.

Fall 2: $0 < \alpha_i^* < \frac{C}{m}$

Mit Gleichung (3.11) gilt: $r_i^* > 0 \overset{(3.13)}{\Longrightarrow} \xi_i^* = 0 \underset{\alpha_i \neq 0}{\overset{(3.12)}{\Longrightarrow}} y_i\, f(\boldsymbol{x_i}) = 1$,

der Punkt $\boldsymbol{x_i}$ befindet sich auf der Trägerebene und ist richtig klassifiziert.

Fall 3: $\alpha_i^* = \frac{C}{m}$

Mit Gleichung (3.11) gilt $r_i^* = 0$, was keine neuen Erkenntnisse über ξ_i^* liefert. Da $\alpha_i^* \neq 0$ gilt, muss nach Bedingung (3.12) die Gleichung $1 - \xi_i^* = y_i\, f(\boldsymbol{x_i})$ erfüllt sein. Gilt zusätzlich $\xi_i^* > 0$, so verursacht der Punkt $\boldsymbol{x_i}$ einen Marginfehler.

3.33 Definition (Marginfehler)
Einen *Marginfehler* verursachen alle Punkte, die innerhalb der Margin liegen oder falsch klassifiziert wurden.

3.34 Bemerkung
Die Schwere des Marginfehlers lässt sich an den ξ_i^* ablesen:

- Ist $0 < \xi_i^* < 1$, so wurde richtig klassifiziert, aber $\boldsymbol{x_i}$ liegt innerhalb der Margin.
- Ist $\xi_i^* = 1$, so liegt $\boldsymbol{x_i}$ auf der trennenden Hyperebene.
- Ist $\xi_i^* > 1$, so liegt eine echte Fehlklassifikation vor, mit der Größe von ξ_i^* wächst auch der Abstand des Punktes von der trennenden Hyperebene.

Auch hier ist zu erwarten, dass zahlreiche α_i^* Null werden. Dazu muss die Trainingsmenge im Featureraum jedoch auch die Gestalt haben, dass überhaupt eine Möglichkeit besteht, sie mit geringen Fehlern linear zu trennen.

3.3.3 ν Support Vector Machine

Diese Modifikation der klassischen Support Vector Machine wurde in [10] ursprünglich zur Anwendung in der Regression entwickelt. Eine Übertragung der gewonnenen Kenntnisse auf die Klassifikation erfolgt dort ebenfalls. Der neue Parameter ν ermöglicht unter gewissen Voraussetzungen die Kontrolle der Anzahl an Support Vektoren und Marginfehler.

Das primale Optimierungsproblem für eine ν Support Vector Machine ist:

$$\begin{array}{l} \min\limits_{\boldsymbol{\omega},b,\boldsymbol{\xi},\rho} \left(\frac{1}{2} \|\boldsymbol{\omega}\|^2 - \nu\rho + \frac{1}{m}\sum\limits_{i=1}^{m} \xi_i \right) \\ \\ \text{u. d. N.} \left\{ \begin{array}{rcll} y_i\big(\langle \boldsymbol{\omega}, \Phi(\boldsymbol{x_i})\rangle - b\big) & \geq & \rho - \xi_i, & \\ \xi_i & \geq & 0, & i \in \{1,\ldots,m\} \\ \rho & \geq & 0\,. & \end{array} \right. \end{array} \tag{P_ν}$$

3.35 Bemerkung
Das Einfügen des Termes $-\nu\rho$ ist durch den Regressionsansatz motiviert. Der Parameter ν wird im Voraus gewählt. Bei der Regression bestimmt das ρ, welches dort ε genannt wird, die Breite des Streifens, in dem sich die Punkte befinden sollen. Liegen Punkte $\boldsymbol{x_i}$ außerhalb, werden sie durch das entsprechende ξ_i bestraft. Auch dort wird der Streifen als Margin bezeichnet. Im Falle der Klassifikation werden Punkte bestraft, die innerhalb der Margin der Breite $\frac{2\rho}{\|\boldsymbol{\omega}\|}$ liegen. Dadurch ist das ν über das ρ an die Marginbreite gekoppelt.

Herleitung des Lagrange Dualproblems

Die Lagrangefunktion zu (P_ν) lautet:

$$\begin{aligned} L(\boldsymbol{\omega}, b, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta) = {} & \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \\ & - \sum_{i=1}^{m}\Big(\alpha_i\left[y_i\left(\langle\boldsymbol{\omega}, \Phi(\boldsymbol{x_i})\rangle - b\right) - \rho + \xi_i\right] + \beta_i\,\xi_i\Big) - \delta\,\rho \end{aligned} \tag{3.14}$$

mit Lagrange Multiplikatoren $\boldsymbol{\alpha}, \boldsymbol{\beta}, \delta$.

3.36 Bemerkung
Der Teil, welcher zu den Ungleichheitsrestriktionen von (P_ν) gehört, ist dem bei der Lagrangefunktion zur C SVM sehr ähnlich. Das r_i heißt hier β_i, statt $-1+\xi_i$ hat man jetzt $-\rho+\xi_i$ und der Term $-\delta\,\rho$ kommt dazu. Auch eine Korollar 3.32 entsprechende Aussage ist hier vorhanden. Wegen des zusätzlichen Lagrange Multiplikators δ kommt lediglich eine dritte Bedingung $\delta^* = 0 \ \vee \ \rho^* = 0$ hinzu.

Mit Hilfe der ersten KKT Bedingung

$$\nabla_{\boldsymbol{\omega},b,\boldsymbol{\xi},\rho}\, L(\boldsymbol{\omega}, b, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta) = 0$$

erhält man die folgenden vier Gleichungen:

$$\begin{aligned} \sum_{i=1}^{m} \alpha_i\, y_i\, \Phi(\boldsymbol{x_i}) &= \boldsymbol{\omega}\ , & \sum_{i=1}^{m} \alpha_i\, y_i &= 0\ , \\ \sum_{i=1}^{m} \alpha_i - \delta &= \nu\ , & \alpha_i + \beta_i &= \frac{1}{m}\ . \end{aligned} \tag{3.15}$$

Durch Einsetzen in Gleichung (3.14) erhält man die nur noch von $\boldsymbol{\alpha}$ abhängige Lagrangefunktion

$$L(\boldsymbol{\omega}, b, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}, \delta) = -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i\, \alpha_j\, y_i\, y_j\, k(\boldsymbol{x_i}, \boldsymbol{x_j})\ .$$

Das Lagrange Dualproblem zu (P_ν) hat die Form

$$\max_{\boldsymbol{\alpha}} \Big(-\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i\, \alpha_j\, y_i\, y_j\, k(\boldsymbol{x_i}, \boldsymbol{x_j}) \Big)$$

$$\text{u.d.N.} \quad \begin{cases} \sum\limits_{i=1}^{m} \alpha_i\, y_i & = 0 \\ \sum\limits_{i=1}^{m} \alpha_i & \geq \nu \\ \frac{1}{m} \geq \alpha_i & \geq 0\ , \qquad i \in \{1, \ldots, m\}\ . \end{cases} \tag{D_ν}$$

Die auftretenden Nebenbedingungen haben ihren Ursprung in (3.15). Die Entscheidungsfunktion berechnet sich wie zuvor, und auch hier ist die Lösung von (D_ν) dünn bezüglich $\boldsymbol{\alpha}^*$.

3.37 Bemerkung
Im Vergleich zum dualen Problem ($\mathrm{D_C}$) erkennt man, dass der lineare Term $\sum_{i=1}^{m} \alpha_i$ aus der Zielfunktion verschwunden ist. Diese wird dadurch quadratisch homogen in $\boldsymbol{\alpha}$, was die Verwendung spezieller Optimierungsverfahren ermöglicht. Außerdem tritt hier die zusätzliche Bedingung $\sum_{i=1}^{m} \alpha_i \geq \nu$ auf, welche zusammen mit der Beschränkung der α_i zu $\nu \in [\,0, 1]$ führt.

Berechnung von b^* und ρ^*

Es bezeichne wiederum $(\boldsymbol{\omega}^*, b^*, \boldsymbol{\xi}^*, \rho^*, \boldsymbol{\alpha}^*, \boldsymbol{\beta}^*, \delta^*)$ einen KKT Punkt zu (P_ν) und (D_ν). In diesem Zusammenhang wird eine neue Idee nach [10] zur Berechnung von b^* vorgeschlagen, welche man auch schon in Abschnitt 3.1.3 verwenden kann. Bilde dafür die Mengen

$$S_+ = \{\boldsymbol{x_i} \mid 0<\alpha_i^*<\tfrac{1}{m} \ \wedge \ y_i=1\} \quad \text{und} \quad S_- = \{\boldsymbol{x_i} \mid 0<\alpha_i^*<\tfrac{1}{m} \ \wedge \ y_i=-1\} \ ,$$

sie sollen identische Mächtigkeit s haben. Um dies zu erfüllen, kann es vorkommen, dass manche Support Vektoren nicht benutzt werden. In der Regel lässt man solche $\boldsymbol{x_j}$, die zu sehr kleinen $\alpha_j^* \neq 0$ gehören, wegfallen.

Die Elemente der beiden Mengen werden in $\boldsymbol{x_i^+}$ und $\boldsymbol{x_i^-}$ für $i \in \{1, \ldots, s\}$, umbenannt. Für diese Punkte gilt $0 < \alpha_i^* < \frac{1}{m}$ und mit den Gleichungen aus (3.15) folgt, dass $\beta_i^* > 0$ ist. Korollar 3.32 und Bemerkung 3.36 liefern $\xi_i^* = 0$, und zusammen mit $\alpha_i^* > 0$ die Gleichung $y_i\, f(\boldsymbol{x_i}) = \rho^*$. Es gilt also:

$$y_i^+\big(\langle \boldsymbol{\omega}^*, \Phi(\boldsymbol{x_i^+})\rangle - b^*\big) = \rho^* \qquad \text{und} \qquad y_i^-\big(\langle \boldsymbol{\omega}^*, \Phi(\boldsymbol{x_i^-})\rangle - b^*\big) = \rho^*$$

für alle $\boldsymbol{x_i^+} \in S_+$ und $\boldsymbol{x_i^-} \in S_-$ und damit

$$\sum_{i=1}^{s} y_i^+\big(\langle \boldsymbol{\omega}^*, \Phi(\boldsymbol{x_i^+})\rangle - b^*\big) \ = \ \sum_{i=1}^{s} y_i^-\big(\langle \boldsymbol{\omega}^*, \Phi(\boldsymbol{x_i^-})\rangle - b^*\big) \ , \tag{3.16}$$

sowie

$$2s\rho^* \ = \ \sum_{i=1}^{s} y_i^+\big(\langle \boldsymbol{\omega}^*, \Phi(\boldsymbol{x_i^+})\rangle - b^*\big) + \sum_{i=1}^{s} y_i^-\big(\langle \boldsymbol{\omega}^*, \Phi(\boldsymbol{x_i^-})\rangle - b^*\big) \ . \tag{3.17}$$

Aus Gleichung (3.16) erhält man mit den Gleichungen aus (3.15):

$$\boxed{b^* \ = \ \frac{1}{2s} \sum_{\boldsymbol{x} \in S_+ \cup S_-} \ \sum_{j=1}^{m} \alpha_j^*\, y_j\, k(\boldsymbol{x_j}, \boldsymbol{x})} \tag{3.18}$$

und aus Gleichung (3.17):

$$\boxed{\rho^* \ = \ \frac{1}{2s}\Big(\sum_{\boldsymbol{x} \in S^+} \sum_{j=1}^{m} \alpha_j^*\, y_j\, k(\boldsymbol{x_j}, \boldsymbol{x}) - \sum_{\boldsymbol{x} \in S^-} \sum_{j=1}^{m} \alpha_j^*\, y_j\, k(\boldsymbol{x_j}, \boldsymbol{x})\Big) \ .} \tag{3.19}$$

Dabei sind $y_i^+ = 1$ und $y_i^- = -1$ die zu $\boldsymbol{x_i^+}$ und $\boldsymbol{x_i^-}$ gehörenden y Werte.

Eigenschaften von ν SVM

Wie bei der C SVM besprochen wurde, verursacht ein Punkt $\boldsymbol{x_i}$ einen Marginfehler, wenn das entsprechende $\xi_i^* > 0$ ist und dadurch der Punkt innerhalb der Margin oder auf der falschen Seite der Hyperebene liegt. Diese Behauptung gilt auch hier und motiviert die folgende Definition.

3.38 Definition (Anteil an Marginfehlern, Anteil an Support Vektoren)
Der *Anteil an Marginfehlern* ist gegeben durch:

$$\frac{1}{m}\left|\{i : \xi_i^* > 0\}\right|,$$

der *Anteil an Support Vektoren* ist:

$$\frac{1}{m}\left|\{i : \alpha_i^* > 0\}\right| .$$

In den nachfolgenden Sätzen gilt stets die Voraussetzung $\rho^* > 0$.

3.39 Satz
Sei k eine Kernfunktion und der ν SVM Algorithmus wurde mit diesem Kern auf eine Trainingsmenge S angewendet, mit dem Ergebnis, dass $\rho^* > 0$ ist. Dann gilt:

a.) ν ist eine obere Grenze für den Anteil an Marginfehlern.

b.) ν ist eine untere Grenze für den Anteil an Support Vektoren.

c.) Seien die Trainingsdaten unabhängig und identisch gemäß $P(\boldsymbol{x}, y) = P(x)P(y|\boldsymbol{x})$ verteilt. Weder $P(\boldsymbol{x}, 1)$ noch $P(\boldsymbol{x}, -1)$ seien diskret. Weiterhin sei die Kernfunktion analytisch und nicht konstant.

Mit einer asymptotischen Wahrscheinlichkeit von 1 ist ν gleich dem Anteil an Support Vektoren und gleich dem Anteil an Marginfehlern.

Beweis
Im Folgenden wird nur den Beweis zu den ersten beiden Aussagen aufgeführt. Der Beweis zur letzten Behauptung findet man in [10].

zu a.) Seien $(\boldsymbol{x_i}, y_i)$ zusammen mit α_i^*, β_i^* und ξ_i^* so umnummeriert, dass $\xi_i^* > 0$ für alle $i \in \{1, \ldots, l\}$ und $\xi_j^* = 0$ für $j \in \{l+1, \ldots, m\}$. Dies bedeutet, dass nur die ersten l Trainingsdaten einen Marginfehler verursachen. Folgende Betrachtungen gelten wegen Korollar 3.32 und Bemerkung 3.36 für alle $i \in \{1, \ldots, l\}$.

Da $\xi_i^* > 0$ ist, gilt $\beta_i^* = 0$ und wegen Gleichung (3.15) ist $\alpha_i^* = \frac{1}{m}$. Nach Voraussetzung ist $\rho^* > 0$, was zu $\delta^* = 0$ führt. Somit ist mit $\boldsymbol{\alpha^*} \geq 0$:

$$\nu \overset{(3.15)}{=} \sum_{i=1}^{m} \alpha_i^* = \sum_{i=1}^{l} \alpha_i^* + \sum_{i=l+1}^{m} \alpha_i^* \geq \sum_{i=1}^{l} \alpha_i^* = \frac{l}{m} ,$$

was gerade dem Bruchteil an Marginfehlern entspricht.

zu b.) Seien $(\boldsymbol{x_i}, y_i)$ zusammen mit α_i^*, β_i^* und ξ_i^* so umnummeriert, dass $\alpha_i^* > 0$ für alle $i \in \{1, \ldots, l\}$ und $\alpha_j^* = 0$ für $j \in \{l+1, \ldots, m\}$. Dies bedeutet, dass die ersten l Trainingsdaten die Support Vektoren sind.

Wegen den Nebenbedingungen von (D_ν) können die entsprechenden α_i^* höchstens den Beitrag $\alpha_i^* = \frac{1}{m}$ zur Bedingung $\sum_{i=1}^m \alpha_i \geq \nu$ liefern. Es folgt:

$$\frac{l}{m} \;\geq\; \sum_{i=1}^{l} \alpha_i^* \;=\; \sum_{i=1}^{l} \alpha_i^* + \sum_{i=l+1}^{m} \alpha_i^* \;=\; \sum_{i=1}^{m} \alpha_i^* \;\geq\; \nu \;,$$

und damit die Behauptung.

□

3.40 Bemerkung
Die erste Nebenbedingung von (D_ν) ist $\sum_{i=1}^m \alpha_i \, y_i = 0$. Aus ihr kann man mit $I_+ = \{i : \alpha_i^* > 0 \;\wedge\; y_i = +1\}$ und $I_- = \{i : \alpha_i^* > 0 \;\wedge\; y_i = -1\}$ die Gleichung

$$\sum_{i \in I_+} \alpha_i^* = \sum_{i \in I_-} \alpha_i^*$$

folgern. Über

$$\sum_{i=1}^{m} \alpha_i^* = 2 \sum_{i \in I_+} \alpha_i^* = 2 \sum_{i \in I_-} \alpha_i^*$$

erhält man mit der zweiten Nebenbedingung von (D_ν):

$$\sum_{i \in I_+} \alpha_i^* \geq \frac{\nu}{2} \quad \text{und} \quad \sum_{i \in I_-} \alpha_i^* \geq \frac{\nu}{2} \;.$$

Führt man nun den Beweis zum Satz 3.39 für die Mengen I_+ und I_- durch, so erkennt man, dass dieser auch für beide Klassen separat gilt. Es ist lediglich $\frac{\nu}{2}$ statt ν zu setzen.

3.41 Satz
Wenn ν SVM Klassifikation ein $\rho^* > 0$ liefert, dann führt C SVM Klassifikation mit $C = \frac{1}{\rho^*}$ zur gleichen Entscheidungsfunktion $f(\boldsymbol{x})$.

Beweis siehe [10]. □

3.42 Bemerkung
Nach Bemerkung 3.37 ist $\nu \in [\,0, 1]$. Mit den Erkenntnissen aus Satz 3.39 ist jedoch weder $\nu = 0$, noch $\nu = 1$ wünschenswert. Im ersten Fall wären keine Support Vektoren vorhanden, im zweiten Fall würden alle Punkte einen Marginfehler verursachen. Dies gilt natürlich nur wenn $\rho^* > 0$. Man betrachtet daher nur $\nu \in (\,0, 1)$.

Strategie zur Wahl von ν

Bisher wurde noch keine Vorgehensweise zur Wahl eines geeigneten Parameters entwickelt, welche die Eigenschaften der ν SVM ausnutzt. Auch Schölkopf und Smola verweisen in [10] auf schon vorhandene Algorithmen für C SVM. Für die Anwendung in dieser Arbeit ist das folgende Prinzip ausreichend.

- Starte mit dem Intervall $I_1 = (0, 1)$ und der Schrittweite $h = 10^{-1}$.
- Für $i = 1, \ldots, 9$:
  - Trainiere eine ν SVM mit $\nu_i := i\,h$.
  - Zähle die Anzahl $f(i)$ an Fehlklassifikationen der Trainingsdaten.
- l sei dasjenige i, für welches die ν_i SVM die wenigsten Fehlklassifikationen produziert:
$$l := \arg\min_i f(i) \ .$$
- Setze $l_1 := l$.
- Für $r = 2, \ldots, N$:
  - Setze $I_r := \left[\frac{10\,l_{r-1}-9}{10^r}\ , \frac{10\,l_{r-1}+9}{10^r}\right]$ und $h := 10^{-r}$.
  - Für $i = 0, \ldots, 18$:
    - Trainiere eine ν SVM mit
$$\nu_i := \frac{10\,l_{r-1} - 9}{10^r} + i\,h = (10\,l_{r-1} - 9 + i)\,h \ .$$
    - Zähle die Anzahl $f(i)$ an Fehlklassifikationen der Trainingsdaten.
  - l sei dasjenige i, für welches die ν_i SVM die wenigsten Fehlklassifikationen produziert:
$$l := \arg\min_i f(i) \ .$$
  - Setze $l_r := 10\,l_{r-1} - 9 + l$.
- Setze $\nu = \frac{l_N}{10^N}$.

3.43 Bemerkung
Das Training der jeweiligen ν SVM findet mit einem festen Kern k und allen vorhandenen Trainingsdaten statt. Ist l nicht eindeutig, so bildet man mehrere Intervalle I_r, abhängig vom jeweiligen l, und sucht in jedem einzelnen nach dem Minimum an Fehlklassifikationen. Im Anschluss wird darüber das globale Minimum gebildet und entsprechend weiterverwendet. Die Wahl von N hängt von dem verwendeten Abbruchkriterium ab. Dafür gibt es verschiedene Möglichkeiten, bei denen auch die Anzahl an Fehlklassifikationen eine Rolle spielen kann. Ist auch das l für $r = N$ nicht eindeutig, so kann man zusätztlich mit Kreuzvalidierung testen, für welches der in Frage kommenden ν der Generalisierungsfehler am geringsten ist.

3.4 Test der Generalisierungsfähigkeit einer SVM

Das folgende Verfahren lässt sich auf alle Varianten einer Support Vector Machine anwenden. Man geht davon aus, dass eine SVM gegeben ist mit festem Kern k und festem C bzw. ν. Es soll nun getestet werden wie gut das vorhandene Modell für neue, unbekannte Daten funktioniert. Dies geschieht mit *Kreuzvalidierung* [12]. Obwohl es viele verschiedene Arten der Kreuzvalidierung gibt, arbeiten alle nach dem gleichen Prinzip.

Prinzip der Kreuzvalidierung

Gegeben seien Daten $D = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{m}$.
Sie stellen Punkte $\boldsymbol{x_i} \in \mathbb{R}^n$ dar, deren Klasse $y_i \in \{-1, +1\}$ bekannt ist.

- Teile die Daten in eine Trainingsmenge S und eine Testmenge T auf. Diese beiden Mengen seien disjunkt
- Trainiere die gegebene Support Vector Machine mit S. Die Entscheidungsfunktion $f(\boldsymbol{x})$ wird dadurch definiert.
- Bilde für alle Elemente $(\boldsymbol{x_t}, y_t)$ der Testmenge den Wert $f(\boldsymbol{x_t})$ und vergleiche ihn mit y_t. Man erhält eine Fehlerquote pro Testdurchlauf. Diese ist der Quotient aus der Anzahl an Fehlklassifikationen und der Anzahl der Tests.
- Dieser Prozess wird mit verschiedenen Untermengen so lange wiederholt, bis jedes Objekt der Datenmenge einmal für die Testmenge verwendet wurde.
- Die Gesamtfehlerquote ist der Mittelwert der einzelnen Fehlerquoten pro Testdurchlauf.

k-fache Kreuzvalidierung

Teile D in l Teilmengen $T_1, \ldots, T_l$ auf. Für $i = 1, \ldots, l$ sei T_i die Testmenge und die verbleibenden $l - 1$ Teilmengen die Trainingsmenge.

Die *l-fach stratifizierte Kreuzvalidierung* achtet darauf, dass jede der l Teilmengen eine annähernd gleiche Verteilung besitzt.

Leave One Out Kreuzvalidierung

Dies ist ein Spezialfall der l-fachen Kreuzvalidierung, bei der jede Teilmenge aus genau einem Element besteht. Die stratifizierte Variante ist hier nicht mehr möglich.

4 Ganganalyse

Der Gang einer Person enthält zahlreiche Informationen, die von direkten visuellen Eindrücken, z.B. der Kleidung, unabhängig sind. Um den Gang von solchen Eindrücken zu trennen, abstrahiert man die Person, indem man sie als Strichmännchen darstellt. Lässt man ein solches Strichmännchen laufen, kann man immer noch eventuelle körperliche Gebrechen erkennen und auf das Geschlecht der Person schließen.

Aufgabe der Ganganalyse ist es, den „reinen“ Gang einer Person darzustellen, um unabhängig von sonstigen visuellen Eindrücken neue Erkenntnisse zu gewinnen.

Anwendungen findet man beispielsweise im sportwissenschaftlichen Bereich, um Bewegungsabläufe zu analysieren oder auch in der Orthopädie, um die Funktionalität von Prothesen zu testen.

In dieser Arbeit soll mit Hilfe der Ganganalyse eine Klassifikation in parkinsonkranke und gesunde Personen erfolgen. Der hier verwendete Ansatz folgt dem von Troje [13], in dem das Geschlecht einer Person aus deren Gang bestimmt werden konnte.

Zunächst werden die Grundlagen, die für die Ganganalyse benötigt werden, erläutert. Es wird aufgezeigt, welche Daten für die Darstellung des Gangs oder gewisser Körperhaltungen einen Sinn ergeben und wie diese gemessen bzw. berechnet werden. Dann werden diese Daten durch zwei Hauptkomponentenanalysen mathematisch so bearbeitet, dass sie für die folgende Klassifikation eine geeignete Form haben.

4.1 Grundlagen

Als erstes werden die Daten betrachtet, die man benötigt, um einen Gang zu modellieren. Dies sind im Wesentlichen die Raumkoordinaten der Gelenke und nach Bedarf zusätzlich deren Winkeldaten. Im Anschluss daran wird erläutert, was genau gemessen wird und wie man daraus die benötigten Parameter berechnet.

4.1.1 Kinematische Standardparameter

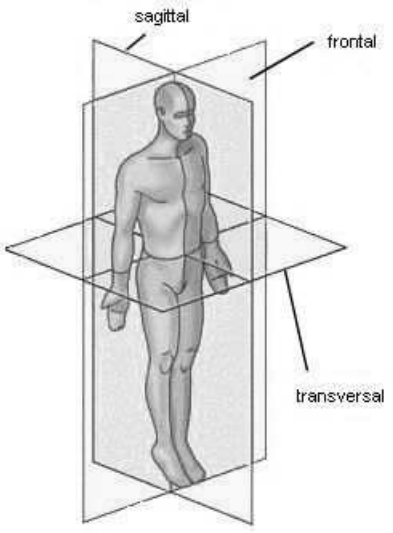
Im Folgenden werden die zwei wesentlichen kinematischen Standardparameter erläutert, die man für die Ganganalyse benötigt. Davon finden in dieser Arbeit nur die Raumkoordinaten der Gelenke Verwendung.

Raumkoordinaten der Gelenke

Zur Beschreibung des Gangs betrachtet man feste Punkte am Körper und deren Bewegung im Raum. Als Punkte eignen sich hierfür besonders gut die Gelenke. Sie sind durch steife Segmente, den Knochen, miteinander verbunden. Die Konstruktion eines Strichmännchens erfolgt durch Verbinden der entsprechenden Gelenkspunkte.

Winkeldaten der Gelenke

Beugung und Streckung, z.B. der Arme, sind bereits indirekt durch die Raumkoordinaten gegeben. Für gewisse Anwendungen ist jedoch eine explizite Darstellung in Form von Winkeldaten sinnvoll. Diese können, wie in [8] beschrieben, aus den Raumkoordinaten berechnet werden. Für die genaue Bezeichnung der Winkeldaten aller Körperstellungen definiert man drei Hauptebenen durch den Körper.



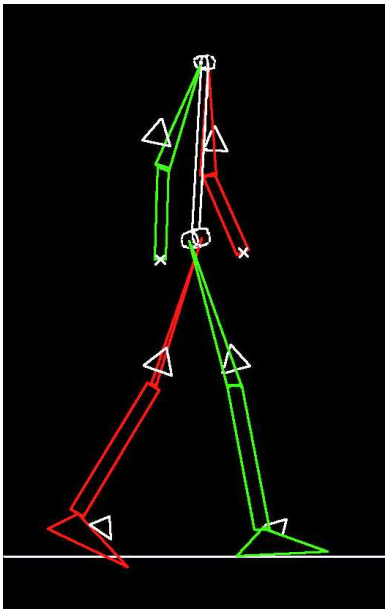
Die *Sagittalebene* beschreibt die Seitenansicht des Körpers, sie erfasst die Flexion (Beugung) und Extension (Streckung) von Arm-, Hüft-, Knie- und Sprunggelenken. Die *Frontalebene* ist die Ebene, auf die eine Person zu- oder weggeht, hier können Abduktion (Bewegung vom Körper weg) und Adduktion (Bewegung zum Körper hin) der Gliedmaße sowie die Beckenneigung dargestellt werden. Die Ebene parallel zum Boden ist die *Transversalebene*, sie zeigt die Rotation der Füße und die Drehung des Beckens.

Abbildung 4.1: Körperebenen [1].

4.1.2 Technische Umsetzung

Die hier verwendeten Gangparameter stammen von der Neurologie des Universitätsklinikums des Saarlandes in Homburg. Die Aufnahme erfolgte über ein System der zebris Medical GmbH.

An Oberarmen, Oberschenkel, Handgelenken sowie auf den Fußrücken werden Ultraschallmarker in Form von Einzel- bzw. Dreifachmarkern ($\times$ bzw. $\triangle$) angebracht. Die Person bewegt sich auf einem Laufband, welches sich zwischen zwei Messaufnehmern befindet. Über Laufzeitmessung von Ultraschallimpulsen, die von kleinen Sendern im Messaufnehmer zu den Mikrofonen der Marker abgegeben werden, erfolgt die exakte Bestimmung der Raumpositionen dieser und die Rekonstruktion einer Strichfigur, die in nebenstehender Abbildung zu sehen ist. Die Software „WinGait“ liefert die Raumkoordinaten der Marker und berechnet die Winkeldaten der Arm-, Ellbogen-, Hüft-, Knie- und Fußgelenke. Darüber hinaus werden für die Gelenke feste Koordinaten angegeben, welche deren Lage bezüglich geeigneter Referenzmarker definieren.



Durch dieses Messsystem erhält man eine objektive kinematische Analyse des menschlichen Gangs durch die dreidimensionale Verfolgung der Oberflächenmarker.

4.1.3 Daten

Zunächst einmal liegen die Daten in Form der Raumkoordinaten der Marker vor. Diese ergeben für sich gesehen keine sinnvolle Darstellung des Gangs vergleichbar mit dem Modell des Strichmännchens. Um dies zu erreichen werden die Raumkoordinaten der Gelenke benötigt und, für die Beschreibung des Fußes, die von großer Zehe und Ferse. Zur Berechnung der fehlenden Raumkoordinaten werden vor Aufnahme der eigentlichen Messung noch zusätzliche Messungen durchgeführt, welche die Lage der Gelenke bezüglich der Marker beschreiben.

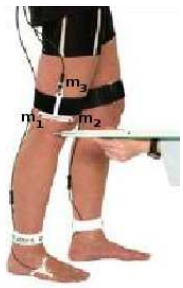
Bevor die Person zu laufen beginnt, werden die beiden Messaufnehmer mittels eines Kalibrierrahmens, der sich auf dem Boden des Laufbandes befindet, auf die Laufbandebene geeicht. Anschließend stellt sich die Person auf das Band und die Marker werden angebracht. Mit einem Taststift erfolgt die Vermessung der Gelenke an Schulter, Ellenbogen innen/außen, Hüfte, Knie innen/außen sowie am Fußgelenk innen/außen, an der großen Zehe und an der Ferse auf beiden Körperseiten. Diese Angaben sind essentiell für die Berechnung der Gelenkkoordinaten aus denen der Marker.

Berechnung der Gelenkkoordinaten

Durch die Eichung der Messaufnehmer wird ein *globales Koordinatensystem* festgelegt. Die x-Achse zeigt in Laufrichtung des Laufbandes, die y-Achse vertikal nach oben und die z-Achse in Laufrichtung rechts. In diesem Koordinatensystem sind die Daten der Marker beschrieben. Hier sollen auch die Daten der Gelenke, der Zehe und der Ferse gegeben sein.

Jeder einzelne Dreifachmarker wiederum beschreibt ein individuelles *lokales Koordinatensystem*, das *Markersystem*. Vor Beginn der Gangaufzeichnung werden die Ortsvektoren benachbarter Gelenke in diesem System festgelegt, dies erfolgt durch die Messung mit dem Taststift. Mit Hilfe dieser Vektoren kann man später die globalen Gelenkkoordinaten berechnen. Beispielsweise definiert der Dreifachmarker am Oberarm ein lokales System, auf das sich die Koordinaten von Schulter und Ellbogen beziehen. Anschaulich wird dadurch die Verbindung von Marker zu Schulter und von Marker zu Ellenbogen hergestellt, der Oberarmknochen ist entstanden.

Ein Markersystem wird wie folgt aufgebaut, dabei bezeichnen $\boldsymbol{m_1^g}$, $\boldsymbol{m_2^g}$ und $\boldsymbol{m_3^g}$ die Mikrofone eines Dreifachmarkers, jeweils dargestellt im globalen Koordinatensystem:



Ursprung $\boldsymbol{u^g}$	:	$\frac{1}{2}(\boldsymbol{m_2^g} - \boldsymbol{m_1^g})$
Richtung von $\boldsymbol{e_1^m}$	:	$\boldsymbol{m_2^g} - \boldsymbol{m_1^g}$
Richtung von $\boldsymbol{e_2^m}$	:	$\boldsymbol{m_3^g} - \boldsymbol{u^g}$
Richtung von $\boldsymbol{e_3^m}$	:	($\boldsymbol{e_1^m}$ - Richtung) $\times$ ($\boldsymbol{e_2^m}$ - Richtung)

Eine Normierung der Richtungen ist noch erforderlich.

Der Bildausschnitt [16] zeigt die drei Mikrofone des Oberschenkelmarkers, der Taststift zeigt auf das Kniegelenk. Da sich der Marker $\boldsymbol{m_3^g}$ auf der Mittelsenkrechten zwischen $\boldsymbol{m_1^g}$ und $\boldsymbol{m_2^g}$ befindet, kann die Richtung von $\boldsymbol{e_2^m}$ wie oben angegeben werden. Ansonsten wäre die Orthogonalität des entstehenden Systems nicht garantiert. Da sich die Dreifachmarker im Raum bewegen, und somit zu jedem Zeitpunkt neue Koordinaten im globalen Koordinatensystem besitzen, werden zeitabhängig neue Markerysteme definiert. In diesen ändern sich die Ortsvektoren der Gelenke, und damit die Gestalt der Knochen, nicht.

Die *Berechnung der Gelenkkoordinaten im globalen System* erfolgt durch eine gewöhnliche Koordinatentransformation:

$$\boldsymbol{x^g} = \boldsymbol{u^g} + \sum_{i=1}^{3} x_i^m \, \boldsymbol{e_i^m} \qquad \text{mit } \boldsymbol{x^m} = (x_i^m)_{i=1}^3$$

Hierbei gelten die folgenden Variablenbezeichnungen:

- $\boldsymbol{e_i^m}$: Einheitsvektoren des Markersystems
- $\boldsymbol{u^g}$: Ursprung des Markersystems
- $\boldsymbol{x^m}$: Gelenk im Markersystem
- $\boldsymbol{x^g}$: Gelenk im globalen System

4.2 Mathematische Bearbeitung der Daten

Den Gang einer Person kann man als Zeitreihe von Körperhaltungen betrachten. Bei der Messung werden mit einer Frequenz von 30 Hz die globalen Werte der sechs Dreifachmarker und der beiden Einfachmarker gemessen. Für jedes Mikrofon wird die x-, y- und z-Koordinate gemessen. Insgesamt erhält man zu jedem Zeitpunkt 60 Daten, welche die Haltung einer Person charakterisieren.

Aus diesen Daten berechnet man, wie in Abschnitt 4.1.3 beschriebenen, die globalen Gelenkkoordinaten sowie die der Füße. Die innen/außen Werte von Ellenbogen-, Knie- und Fußgelenk sind nur für die Berechnung der Winkeldaten bedeutsam, hier werden diese Werte gemittelt. Darüber hinaus wird der Fuß lediglich durch den Mittelwert der Fußgelenkdaten dargestellt. Pro Zeitpunkt entstehen so aus den Markerkoordinaten 36 Gelenkdaten für die Darstellung einer Haltung als dreidimensionales Photo eines Strichmännchens. Eine Laufzeit von 20 Sekunden wird untersucht. Dies entspricht 600 Haltungen bzw. 21600 Daten.

Im ersten Schritt wird *der Gang einer einzelnen Person modelliert.* Eine reine Auflistung der Gelenkkoordinaten in Abhängigkeit von der Zeit ist nicht sinnvoll. Die Größe der Datenmenge und somit auch die Darstellung des Gangs wäre abhängig von der Dauer der Aufnahme. Der zeitliche Zusammenhang der einzelnen Werte, sowie die Periodizität des Gangs wird damit ebenfalls nicht erfasst. Weiterhin sind die Daten hochgradig redundant, eine Folge der Periodizität. Ihr Informationsgehalt ist im Vergleich zur Größe sehr gering. Um diese Redundanz zu beseitigen, führt man eine erste Hauptkomponentenanalyse durch. Ergebnis dieser ist eine wesentlich kompaktere Darstellung des Gangbildes einer Person. Die Größe der Daten ist unabhängig von der Dauer der Messung.

4.2.1 Erste Hauptkomponentenanalyse

Eine 20 Sekunden lange Aufnahme des Gangs besteht aus 600 Haltungsbildern, die in einem zeitlichen Zusammenhang stehen. Die Körperhaltung (engl. posture) zum Zeitpunkt $t_k = (k-1)\Delta t$ wird durch einen 36-dimensionalen Vektor $\boldsymbol{p}(t)$ dargestellt. Es ist dabei $\Delta t = \frac{1}{30}s$ und $k \in \{1, \ldots, 600\}$. Dieser Vektor enthält die Raumkoordinaten von Schulter, Ellenbogen, Hand, Hüfte, Knie und Fuß der linken und rechten Körperseite zu diesem Zeitpunkt.

Die Darstellung von $\boldsymbol{p}(t)$ hängt von dem verwendeten Messsystem ab. Andere Systeme nehmen noch zusätzliche Daten am Kopf auf oder stellen die Füße genauer dar. Die Dimension einer einzelnen Haltung $\boldsymbol{p}(t_k)$ wird daher als N angenommen. Im Folgenden wird davon ausgegangen, dass die Messung bei einer Frequenz von f Hertz über einen Zeitraum von T Sekunden erfolgt, es enstehen $H = f \cdot T$ viele Haltungen.

Ursprüngliche Darstellung des Gangs

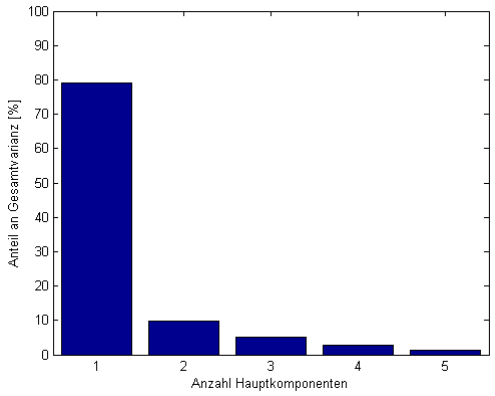
Da die einzelnen Haltungen in einem zeitlichen Zusammenhang stehen, ist es sinnvoll, sie in der gegebenen Reihenfolge in einer Matrix P zu speichern. Diese ist wie folgt definiert

$$P = \big(\boldsymbol{p}(t_1), \ldots, \boldsymbol{p}(t_H)\big) \in \mathbb{R}^{N \times H} \ ,$$

wobei $\boldsymbol{p}(t_k) \in \mathbb{R}^N$ die k-te Haltung darstellt. Von dieser Matrix wird eine Hauptkomponentenanalyse berechnet. Man erhält daraus den Mittelwert $\boldsymbol{p_0} \in \mathbb{R}^N$ über alle Haltungen, die Hauptrichtungen $\boldsymbol{p_1}, \ldots, \boldsymbol{p_N} \in \mathbb{R}^N$ und die haltungsspezifischen Hauptkomponenten $\boldsymbol{c}(t_1), \ldots, \boldsymbol{c}(t_H) \in \mathbb{R}^N$. Analog zu (2.4) kann man eine einzelne Haltung $\boldsymbol{p}(t_k)$ vollständig darstellen als:

$$\boldsymbol{p}(t_k) = \boldsymbol{p_0} + \sum_{i=1}^{N} c_i(t_k)\, \boldsymbol{p_i} \ ,$$

mit $\boldsymbol{c}(t_k) = (c_1(t_k), \ldots, c_N(t_k))^{\mathrm{T}}$ für alle $k \in \{1, \ldots, H\}$ und $c_i(t_k) \in \mathbb{R}$. Die Hauptrichtungen aus dieser ersten PCA bezeichnet man auch als *Eigenhaltungen* (engl. *eigenposture*), sie sind spezifisch für den Gang einer jeden Person. Mittelt man über alle Personen, so erkennt man, dass die erste Hauptrichtung einen Varianzanteil von 79.1% besitzt, siehe dazu Gleichung (2.2). Die ersten vier Hauptrichtungen zusammen decken schon 96.6% der Gesamtvarianz ab.



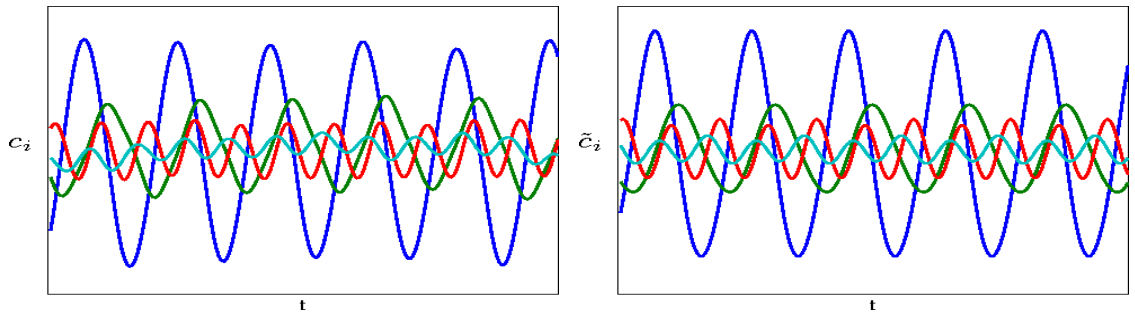
Somit reicht eine Darstellung mit nur vier Summanden aus, um die Haltung ohne großen Informationsverlust darstellen zu können:

$$\widetilde{\boldsymbol{p}}(t_k) = \boldsymbol{p_0} + \sum_{i=1}^{4} c_i(t_k)\, \boldsymbol{p_i} \ . \tag{4.1}$$

Die erste Hauptrichtung enthält im Wesentlichen die Vorwärtsbewegung. Mit der zweiten Hauptrichtung kommt auch die Bewegung nach oben dazu. Durch Hinzunahme der dritten und vierten Hauptrichtung wird der Gang im Ganzen runder und ist mit dem bloßen Auge nicht mehr vom Originalgang zu unterscheiden.

Zeitlicher Verlauf der Hauptkomponenten

Betrachtet man die Hauptkomponenten $c_1(t), \ldots, c_4(t)$ in Abhängigkeit von der Zeit, so erkennt man darin eine gewisse Periodizität. Dies ist nicht verwunderlich, da die Hauptkomponenten die haltungsspezifischen Daten enthalten und der Gang periodisch ist.



(a) Komponenten c_i für $i = 1, \ldots, 4$.

(b) Näherung der c_i durch $\widetilde{c}_i$.

Man nähert sie daher durch geeignete Sinusfunktionen. Die Frequenz der ersten beiden Hauptkomponenten entspricht der des eigentlichen Gangs, die der dritten und vierten Hauptkomponente ist doppelt so groß. Für die Gangfrequenz betrachtet man die Zeit, die man für zwei Schritte benötigt und bildet davon den Kehrwert. Man kann die ersten vier Hauptkomponenten nähern durch:

$$\begin{aligned} \widetilde{c}_1(t) &= a_1 \sin(\omega\, t + \varphi_1), & \widetilde{c}_3(t) &= a_3 \sin(2\,\omega\, t + \varphi_3), \\ \widetilde{c}_2(t) &= a_2 \sin(\omega\, t + \varphi_2), & \widetilde{c}_4(t) &= a_4 \sin(2\,\omega\, t + \varphi_4)\ , \end{aligned} \tag{4.2}$$

mit $\omega, a_1, \ldots, a_4, \varphi_1, \ldots, \varphi_4 \in \mathbb{R}$. Den Gang $\boldsymbol{g}(t)$ kann man nach den zwei Abstraktionen (4.1) und (4.2) sehr einfach als diskrete Zeitreihe von Haltungen beschreiben

$$\begin{aligned} \boldsymbol{g}(t) = \boldsymbol{p_0} &+ a_1 \sin(\omega\, t + \varphi_1)\, \boldsymbol{p_1} + a_2 \sin(\omega\, t + \varphi_2)\, \boldsymbol{p_2} \\ &+ a_3 \sin(2\,\omega\, t + \varphi_3)\, \boldsymbol{p_3} + a_4 \sin(2\,\omega\, t + \varphi_4)\, \boldsymbol{p_4}\ , \end{aligned} \tag{4.3}$$

mit $\omega, \varphi_1, \ldots, \varphi_4 \in \mathbb{R}$ und $\boldsymbol{p_i} \in \mathbb{R}^N$ für $i \in \{0, \ldots, 4\}$.

Neue Darstellung des Gangs

Mit dem Modell (4.3) genügt die Angabe eines $(5N{+}9)$-dimensionalen Spaltenvektors $\boldsymbol{w_j}$ (von engl. walker) mit

$$\boldsymbol{w_j} = (\boldsymbol{p_0}^{\mathrm{T}}, \boldsymbol{p_1}^{\mathrm{T}}, \ldots, \boldsymbol{p_4}^{\mathrm{T}}, \omega, a_1, \ldots, a_4, \varphi_1, \ldots, \varphi_4)^{\mathrm{T}} \in \mathbb{R}^{5N+9}\ , \tag{4.4}$$

um den Gang einer Person j ausreichend genau darzustellen. Die Länge der Messung spielt in dieser Darstellung keine Rolle mehr. Auf diese Weise konnte in dieser Anwendung eine Datengröße von 21600 auf 189 reduziert werden.

Als nächstes werden *die Gangbilder der Personen untereinander* verglichen. Durch eine zweite Hauptkomponentenanalyse kann man die Unterschiede und Gemeinsamkeiten der Personen hervorheben.

4.2.2 Zweite Hauptkomponentenanalyse

Die erste PCA wurde für jede Person seperat über alle Haltungen durchgeführt. Die zweite PCA erfolgt nun über alle Personen, die durch $\boldsymbol{w_j}$ dargestellt werden. Die Personenmatrix W ist eine $(n \times m)$-Matrix, deren Spalten die Gangbilder der verschiedenen Personen enthält:

$$W = (\boldsymbol{w_1}, \ldots, \boldsymbol{w_m}) \in \mathbb{R}^{n \times m} \ .$$

Man geht also von m Personen mit entsprechenden $\boldsymbol{w_j} \in \mathbb{R}^n$ aus. Über dieser Matrix soll die zweite Hauptkomponentenanalyse berechnet werden.

4.1 Bemerkung
Im Gegensatz zur PCA über die Haltungen taucht hier das Problem auf, dass die unterschiedlichen Einheiten nicht homogen sind. Die Eigenhaltungen sind in Millimeter, die Frequenz in Hertz und die Phasen in Grad gegeben. Da die Hauptkomponentenanalyse sehr sensibel bezüglich relativer Skalierung ist, werden zunächst die Daten reskaliert, so dass sie unabhängig von den Einheiten dargestellt sind.

4.2 Beispiel (Reskalierung der Daten)
φ_j: Phasenwert φ_1 der j-ten Person im Gradmaß
ϕ_j: entsprechender Wert im Bogenmaß

Es gilt: $\varphi_j = \frac{180}{\pi}\ \phi_j \quad \forall\ j \in \{1, \ldots, m\}$.
Dieser Zusammenhang überträgt sich auf die Mittelwerte $\overline{\varphi}$ und $\overline{\phi}$ der ersten Phasenwerte über alle m Personen sowie auf die Differenzenquadrate

$$(\varphi_j - \overline{\varphi})^2 \ = \ (\tfrac{180}{\pi}\ \phi_j - \tfrac{180}{\pi}\ \overline{\phi})^2 = (\tfrac{180}{\pi})^2\,(\phi_j - \overline{\phi})^2$$

und schließlich auch auf die Standardabweichungen $\sigma(\varphi)$ und $\sigma(\phi)$. Für den Quotienten aus Phasenwert und entsprechender Standardabweichung ergibt sich:

$$\frac{\varphi}{\sigma(\varphi)} \ = \ \frac{\varphi_j}{\sqrt{\frac{\sum_{j=1}^{n}(\varphi_j - \overline{\varphi})^2}{n-1}}} \ = \ \frac{\varphi_j}{\sqrt{\frac{(\frac{180}{\pi})^2 \sum_{j=1}^{n}(\phi_j - \overline{\phi})^2}{n-1}}} \ = \ \frac{\frac{180}{\pi}\ \phi_j}{\frac{180}{\pi}\sqrt{\frac{\sum_{j=1}^{n}(\phi_j - \overline{\phi})^2}{n-1}}} \ = \ \frac{\phi_j}{\sigma(\phi)} \ .$$

Die Quotienten sind gleich, die Winkeldaten sind unabhängig von der Einheit dargestellt.

Berechnung der reskalierten Personenmatrix W'

Es sei $\boldsymbol{u} \in \mathbb{R}^n$ der Vektor aller Standardabweichungen der Zeilen von W. Man geht davon aus, dass er keine Nullen als Einträge besitzt.

$$u_i = \sigma\big(\boldsymbol{w^{(i)}}\big), \quad \forall\, i \in \{1, \ldots, n\}\ .$$

Hierbei bezeichne $\boldsymbol{w^{(i)}} \in \mathbb{R}^m$ die i-te Zeile von W. Die Matrix $\operatorname{diag}\left(\frac{1}{\boldsymbol{u}}\right) \in \mathbb{R}^{n\times n}$ ist dann:

$$\Big[\operatorname{diag}\Big(\frac{1}{\boldsymbol{u}}\Big)\Big]_{i,j} := \begin{cases} \frac{1}{u_i} & , \text{falls } i = j \\ 0 & , \text{falls } i \neq j\ . \end{cases} \tag{4.5}$$

Mit ihr bewirkt man durch Multiplikation von links an die Matrix W eine Division der i-ten Zeile von W durch $u_i = \sigma\big(\boldsymbol{w^{(i)}}\big)$ und damit eine Reskalierung der Daten wie in Bemerkung 2.7 beschrieben. Die Matrix W' ist dann:

$$W' \ = \ \operatorname{diag}\Big(\frac{1}{\boldsymbol{u}}\Big)\ W \in \mathbb{R}^{n\times m}\ .$$

Hauptkomponentenanalyse über W'

Da die unterschiedlichen Einheiten der Matrix W' jetzt homogen sind, kann man problemlos eine Hauptkomponentenanalyse darüber berechnen. Man erhält die Koeffizientenmatrix $K \in \mathbb{R}^{n\times m}$, die Matrix der Hauptrichtungen $V \in \mathbb{R}^{n\times n}$ und den Mittelwert $\boldsymbol{w_0} \in \mathbb{R}^n$. Es ist also wie in Gleichung (2.3)

$$W' = \boldsymbol{w_0}\ \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} + VK\ .$$

Die Spalten von K enthalten die individuellen Eigenschaften der entsprechenden Personen, deren Gemeinsamkeiten findet man in V und $\boldsymbol{w_0}$. Aus diesem Grund werden hier die Hauptkomponenten, d.h. die Spalten von K, als *Eigenwalker* bezeichnet.

Berechnung von W aus W'

Die Matrix $\operatorname{diag}(\boldsymbol{u}) \in \mathbb{R}^{n\times n}$ ist analog zu Definition (4.5) definiert als:

$$\big[\operatorname{diag}\left(\boldsymbol{u}\right)\big]_{i,j} = \begin{cases} u_i & , \text{falls } i = j \\ 0 & , \text{falls } i \neq j\ . \end{cases}$$

Multipliziert man sie von links an die Matrix W', so erhält man wieder die Darstellung W der Personenmatrix mit den ursprünglichen Einheiten zurück. Unter Verwendung der Assoziativität der Matrixmultiplikation ergibt sich für W die Darstellung

$$W = \widetilde{W}_0 + V\,\widetilde{K} \tag{4.6}$$

mit $\widetilde{W}_0 = \operatorname{diag}(\boldsymbol{u})\,\boldsymbol{w_0}\,\mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}}$ und $\widetilde{K} = \operatorname{diag}(\boldsymbol{u})\,K$. Um den Gang einer jeden Person geeignet darstellen zu können, benötigt man also den Vektor $\boldsymbol{u}$ der Standardabweichungen von W, den Mittelwert $\boldsymbol{w_0}$, sowie die Matrizen V und K.

Spezialfall $m < n$

In dieser Anwendung liegen die Daten von 21 gesunden und 16 erkrankten Personen vor, somit ist $m = 37$ und $n = 5N + 9 = 189$, d.h. es ist $m << n$. Es stellt sich heraus, dass die Koeffizientenmatrix K nur in den ersten $m-1$ Zeilen Einträge ungleich Null besitzt. Das bedeutet, dass auch nur die entsprechenden Hauptrichtungen für die Berechnung von W' benötigt werden. Wegen $m < n$ definieren die ersten $m-1$ Hauptrichtungen, zusammen mit dem Mittelwert $\boldsymbol{w_0}$ den m-dimensionalen Unterraum, in dem die Daten liegen. Folglich kann man W' ohne Informationsverlust auch schreiben als

$$W' = \boldsymbol{w_0}\ \mathbf{1}_{\boldsymbol{m}}^{\mathrm{T}} + V_{red}\, K_{red}$$

mit den reduzierten Matrizen $V_{red} \in \mathbb{R}^{n\times(m-1)}$ und $K_{red} \in \mathbb{R}^{(m-1)\times m}$. Die ursprüngliche Personenmatrix W kann man dann analog zu Gleichung (4.6) darstellen als

$$W = \widetilde{W_0} + V_{red}\, \widetilde{K}_{red}\ ,$$

mit $\widetilde{W_0}$ wie oben und $\widetilde{K}_{red} = \mathrm{diag}(\boldsymbol{u})\, K_{red}$.

4.3 Bemerkung
Die für den Gang $\boldsymbol{w_j}$ spezifischen Daten sind in der j-ten Spalte der Matrix $\widetilde{K}_{red}$ enthalten und haben somit nur noch Dimension $m-1$ statt wie zuvor n. Die Hauptkomponenten aus V_{red} zusammen mit dem Mittelwert $\widetilde{\boldsymbol{w}}_{\mathbf{0}} = \mathrm{diag}(\boldsymbol{u})\, \boldsymbol{w_0}$ definieren den Raum, in dem sich die Daten befinden. Für die anschließende Klassifikation werden daher nur die Spalten von $\widetilde{K}_{red}$ als Trainingsdaten benutzt.

Darstellung eines neuen Personenvektors w

Wie in Kapitel 2 erläutert wurde, bilden die Spalten von V_{red} die Achsen eines neuen Koordinatensystems. Die Spalten von $\widetilde{K}_{red}$ enthalten die Koeffizienten der Personenmatrix W in diesem System.

Gegeben sei der Personenvektor $\boldsymbol{w}$ in in der Form (4.4). Man möchte nun die Koeffizienten von $\boldsymbol{w}$ im neuen Koordinatensystem darstellen. Dies erfolgt durch Lösen des Gleichungssystems

$$V_{red}\ \boldsymbol{k}\ =\ \boldsymbol{w} - \widetilde{\boldsymbol{w}}_{\mathbf{0}}\ . \tag{4.7}$$

Der Vektor $\boldsymbol{k}$ ist die gesuchte Darstellung von $\boldsymbol{w}$. Da es sich um ein überbestimmtes Gleichungssystem handelt, benutzt man dazu die Methode der kleinsten Fehlerquadrate, um die Minumum Norm Lösung zu erhalten.

5 Durchführung der Klassifikation

In den anschließenden Tests wird die Klassifikation auf zwei verschiedene Arten durchgeführt. Die Erste folgt dem Vorbild von Troje in [13] und löst ein lineares Gleichungssystem, um mit dem Ergebnis die Entscheidungsfunktion zu definieren. Die zweite Klassifikation erfolgt über eine Support Vector Machine. Die Testdaten umfassen 21 gesunde und 16 erkrankte Personen. Ihr Gang wurde über einen Zeitraum von 20 Sekunden erfasst und liegt in der durch Gleichung (4.4) gegebenen Form $\boldsymbol{w_j}$, $j \in \{1, \ldots, 37\}$, vor. Die Länge der Vektoren $\boldsymbol{w_j}$ ist $n = 189$. Die erste Hauptkomponentenanalyse wurde bereits durchgeführt, der Gang ist nicht mehr abhängig von der Dauer der Messung.

5.1 Klassifikation nach Troje

Hier wird vor der Klassifikation zunächst die in Kapitel 4 beschriebene zweite Hauptkomponentenanalyse über m Personen durchgeführt. Diese liefert den Mittelwert $\widetilde{\boldsymbol{w}}_{\boldsymbol{0}} \in \mathbb{R}^n$, die Matrix $V_{red} \in \mathbb{R}^{n \times (m-1)}$ der Hauptrichtungen und die Koeffizientenmatrix $\widetilde{K}_{red} \in \mathbb{R}^{(m-1) \times m}$.

Berechnung von f

Die Berechnung der Entscheidungsfunktion f entspricht dem Training einer Support Vector Machine. Die „Trainingsmatrix“ ist gegeben durch $X = K_{red} \in \mathbb{R}^{(m-1) \times m}$, eine Person i wird durch den Spaltenvektor $\boldsymbol{x_i} \in \mathbb{R}^{m-1}$ repräsentiert. Der entsprechende Wert y_i ist -1, wenn die Person erkrankt und $+1$ wenn sie gesund ist. Die y_i bilden den Spaltenvektor $y \in \mathbb{R}^m$. Um die Koeffizienten $\boldsymbol{c} \in \mathbb{R}^{m-1}$ für die Entscheidungsfunktion f aus den Trainingsdaten $(\boldsymbol{x_i}, y_i)_{i=1}^m$ zu berechnen, löst man das Gleichungssystem

$$X^{\mathrm{T}} \boldsymbol{c} = \boldsymbol{y} \ .$$

Auch hier verwendet man das Prinzip der kleinsten Fehlerquadrate. Mit diesem $\boldsymbol{c}$ wird die Entscheidungsfunktion f definiert als

$$f(\boldsymbol{k}) = \boldsymbol{k}\, \boldsymbol{c} \ .$$

Hierbei ist $\boldsymbol{k} \in \mathbb{R}^{(m-1)}$ die Darstellung der Person $\boldsymbol{w}$ in der durch V_{red} aufgespannten Basis. Die Umrechnung von $\boldsymbol{w}$ in $\boldsymbol{k}$ erfolgt nach Gleichung (4.7). Wie auch bei den Support Vector Machines entscheidet das Vorzeichen von f über die Klasse.

Tests

Zunächst wird mit allen $m = 37$ Personen trainiert. Dies bedeutet, dass die zweite PCA und auch die Entscheidungsfunktion aus allen 37 Personen berechnet werden. Die so entstandene Funktion f ordnet jeder Person die richtige Klasse zu.

Um die Generalisierungsfähigkeit zu prüfen, wird in einem zweiten Test eine Leave One Out Kreuzvalidierung durchgeführt. Es ergeben sich 37 Testdurchläufe. Bei jedem Durchlauf wird eine andere Person aus dem Training ausgeschlossen, die zweite PCA daher nur mit $m = 36$ Personen durchgeführt und auch die Entscheidungsfunktion nur aus diesen Daten berechnet. Am Ende eines jeden Durchlaufes werden mit Hilfe der aktuellen Entscheidungsfunktion alle 37 Personen klassifiziert und die Fehlklassifikationen gezählt. Dabei entstehen in allen 37 Testdurchläufen, in denen jeweils alle 37 Personen getestet wurden, keine Fehlklassifikationen.

5.2 Klassifikation mit Support Vector Machine

Beim Klassifizieren mit einer SVM werden die Daten nach der ersten Hauptkomponentenanalyse verwendet. Ein Versuch der linearen Trennung mit $\widetilde{K}_{red}$ wie in Abschnitt 5.1 war nicht erfolgreich; ein Kern wurde nicht benutzt, da eine Darstellung in einem höherdimensionalen Raum durch die $\boldsymbol{w_i}$ schon gegeben ist. Die Trainingspunkte $\boldsymbol{x_i}$ sind also die $\boldsymbol{w_i}$, $i \in \{1, \ldots, 37\}$ aus Gleichung (4.4). Das entsprechende y_i ist gleich 1, wenn es sich um eine gesunde Person handelt und -1, wenn die Person erkrankt ist.

Berechnung von f

Zum Training wird eine ν SVM mit dem Skalarprodukt als Kern verwendet. Das ν wird dabei, wie in Abschnitt 3.3.3 beschrieben, bis auf zwei Nachkommastellen genau bestimmt. Die Trainingsdaten sind $(\boldsymbol{x_i}, y_i)_{i=1}^m$ und die Entscheidungsfunktion wird wie in Kapitel 3 durch

$$f(\boldsymbol{x}) = \sum_{i \in I} \alpha_i^* \, y_i \, \langle \boldsymbol{x_i}, \boldsymbol{x} \rangle - b^*$$

bestimmt. Die Bezeichnungen sind analog zu denen aus Kapitel 3. Eine Umwandlung des Personenvektors $\boldsymbol{w}$ wie bei Troje ist hier nicht erforderlich.

Tests

Zunächst einmal wird mit allen $m = 37$ Personen trainiert. Dabei stellt sich heraus, dass man gesunde und erkrankte Personen für zahlreiche Werte von ν linear trennen kann. Die Entscheidungsfunktion f ordnet jeder Person die richtige Klasse zu.

Um die Generalisierungsfähigkeit zu testen, wird auch hier in einem zweiten Test eine Leave One Out Kreuzvalidierung mit 37 Testdurchläufen durchgeführt. Bei jedem Durchlauf trainiert man die SVM mit $m = 36$ Personen. Anschließend werden aufgrund der entstehenden Entscheidungsfunktion alle 37 Personen klassifiziert und die Fehlklassifikationen gezählt.

Der Parameter ν ist optimal für $\nu = 0.22$ und $\nu = 0.23$, dabei entsteht in allen 37 Testdurchläufen, in denen jeweils alle 37 Personen getestet werden, eine einzige Fehlklassifikation. Im schlechtesten Fall sind $37^2 = 1369$ Fehlklassifikationen möglich.

Bei jedem der 37 Testdurchläufe wird eine andere Hyperebene, und damit auch eine andere Entscheidungsfunktion berechnet. Um beurteilen zu können, wie stark sich diese verändert, berechnet man jeweils den Winkel zwischen dem Normalenvektor des Testdurchlaufes und dem „idealen“ Normalenvektor einer Referenzebene. Diese entsteht aus einer ν SVM mit $\nu = 0.22$ bzw. $\nu = 0.23$ beim Training mit allen 37 Personen. Es stellt sich heraus, dass sich bei $\nu = 0.22$ die Winkel im Intervall $[1.22°, 20.26°]$ befinden mit einer Mittelwert von $4.85°$ und einer Standardabweichung von $5.67°$. Für $\nu = 0.23$ unterscheiden sich diese Angaben nur unwesentlich. Beim Vergleich der entsprechenden Normalenvektoren beim Training mit $\nu = 0.22$ und $\nu = 0.23$ enstehen Winkel aus dem Intervall $[1.23°,2.60°]$ mit einem Mittelwert von $1.92°$ und einer Standardabweichung von $0.24°$. Dies betrifft die Normalenvektoren der Kreuzvalidierung, die Normalenvektoren beim Training mit 37 Personen sind für $\nu = 0.22$ und $\nu = 0.23$ identisch. Es macht also wenig Unterschied, ob man $\nu = 0.22$ oder $\nu = 0.23$ verwendet.

5.3 Vergleich der beiden Ansätze

Beim Training mit allen vorhandenen Daten klassifiziert die SVM ebenso gut wie das Verfahren von Troje. Bei der Generalisierung entsteht bei der SVM eine und bei Troje keine Fehlklassifikation, hinsichtlich 1369 möglichen Fehlklassifikationen ist dies kein großer Unterschied.

Da mit einer wachsenden Anzahl von Beispielen auch eine stabilere Entscheidungsfunktion zu erwarten ist, stellt sich die Frage, was passiert, wenn die Anzahl der Personen die Dimension $n = 189$ des Personenvektors $\boldsymbol{w_i}$ überschreitet. Bei dem Ansatz mit einer SVM hat man dann immer noch die Möglichkeit, Kerne zu benutzen, falls eine lineare Trennung mit einer akzeptablen Anzahl an Fehlklassifikationen nicht mehr möglich ist. Der Ansatz von Troje führt zu einem unterbestimmten Gleichungssystem, mit unendlich vielen Lösungen. Ob dort die Minimum Norm Lösung sinnvoll ist, bleibt zu klären.

6 Fazit und Ausblick

Die Trennung von gesunden und kranken Personen ist sowohl mit dem Ansatz von Troje als auch mit einer linearen Support Vector Machine möglich. Bei der Generalisierung schneidet die Vorgehensweise nach Troje unwesentlich besser ab, aber auch die SVM generalisiert sehr gut. Bei den vorliegenden Ergebnissen muss man berücksichtigen, dass es sich um eine sehr geringe Anzahl von Trainingsdaten handelt. Die Vorteile einer SVM werden erst bei zahlreichen Trainingsdaten deutlich. So bietet sie beispielsweise die Möglichkeit, kontrolliert Fehlklassifikationen zuzulassen, um die Generalisierungsfähigkeit zu erhalten. Des Weiteren bietet sie die Option, Kerne zu benutzen, falls eine lineare Trennung mit wenigen Fehlklassifikationen nicht mehr möglich sein sollte.

Um eine stabilere Entscheidungsfunktion zu erhalten kann man neben einer größeren Anzahl an Daten auch deren Darstellung an das Ziel der Trennung gesund-krank anpassen. Typisch für den Gang einer an Parkinson erkrankten Person ist, dass eine Körperseite gewisse Lähmungserscheinungen aufweist. Daher kann man den Fokus auf den rechts-links Unterschied richten, indem man die erste PCA auf die rechte und linke Seite getrennt anwendet und statt des Vektors $\boldsymbol{x_j} = (\boldsymbol{p_0}^{\mathrm{T}}, \boldsymbol{p_1}^{\mathrm{T}}, \ldots, \varphi_4)^{\mathrm{T}}$ mit dem Differenzenvektor $\boldsymbol{x_j} = (\boldsymbol{p_{0,r}}^{\mathrm{T}} - \boldsymbol{p_{0,l}}^{\mathrm{T}}, \boldsymbol{p_{1,r}}^{\mathrm{T}} - \boldsymbol{p_{1,l}}^{\mathrm{T}}, \ldots, \varphi_{4,r} - \varphi_{4,l})^{\mathrm{T}}$ weiterarbeitet.

Wurde die Entscheidungsfunktion mit einer Support Vector Machine erfolgreich und stabil berechnet, so enthält die trennende Hyperebene Parameter, die für die Unterscheidung wichtig sind. Diese sind zunächst schwierig zu interpretieren, da die Trainingsdaten schon eine Hauptkomponentenanalyse durchlaufen haben und die Hyperebene nur implizit gegeben ist. Es ist zu erwarten, dass man durch Rücktransformieren des Normalenvektors durch die PCA gewisse Trennparameter identifizieren kann. Aber auch, wenn die Interpretation nicht möglich sein sollte, ist das Vorhandensein der trennenden Hyperebene aus dem folgenden Grund nützlich: Die Behandlung von Parkinson erfolgt unter anderem durch Dopamin-Agonisten. Sie ersetzen das bei erkrankten Personen fehlende Dopamin und beheben die Störungen, die durch den Mangel entstehen. Unter anderem wird die Verlangsamung der Bewegungen bzw. die Muskelsteifheit behoben, das Ruhezittern wird gedämpft und es treten weniger unwillkürliche Überbewegungen auf. Der Abstand eines Patienten von der Ebene vor und nach Dopamingabe könnte ein Maß für die Wirkung des Mittels darstellen.

Viele der beschriebenen Möglichkeiten bietet der Ansatz von Troje nicht, daher ist es für weiterführende Untersuchungen sinnvoll, die Support Vector Machine zu verwenden.

A Beispiele für Kerne

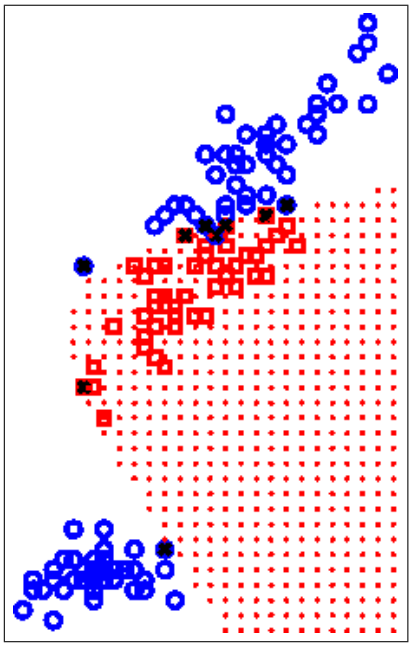
In allen Beispielen werden die vorhandenen Daten mit einer ν SVM trainiert. Die Bestimmung des Parameters ν erfolgt analog zu Abschnitt 3.3.3 bis auf vier Nachkommastellen genau. Daten mit positivem Label sind durch rote Quadrate, solche mit negativem Label durch blaue Kreise gekennzeichnet. Support Vektoren werden zusätzlich durch ein schwarzes Kreuz markiert. In dem durch rote Punkte markierten Bereich werden neuen Daten positive Label zugeordnet. Der Rand dieses Bereiches stellt die trennende Hyperebene dar.

Man beachte, dass sich die exakten Datenpunkte jeweils in der Mitte der blauen Kreise bzw. roten Quadrate befinden. Scheinbare Fehlklassifikationen, wie sie in den Abbildungen A.1(a), A.1(c) und A.2(a) auftreten, sind in Wirklichkeit nicht vorhanden. Welche Punkte fehlklassifiziert sind, wird durch die Entscheidungsfunktion berechnet und nicht anhand der Abbildungen ausgewertet.

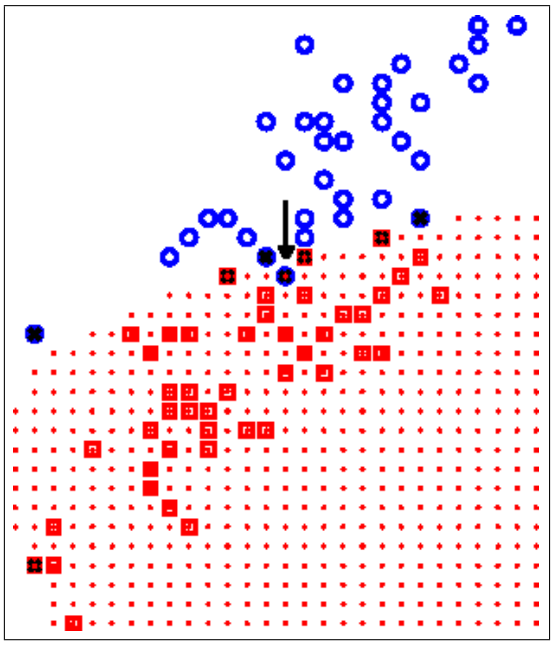
A.1 Irisdaten von R. A. Fisher

Dies ist ein klassischer Datensatz für das Testen von Lernalgorithmen. Er enthält die Maße der Blüten von drei verschiedenen Arten der Irispflanze. Die Originaldaten [5] sind vierdimensionale Vektoren und bestehen aus drei Klassen mit jeweils 50 Instanzen. Für diese Anwendung wurden sie jedoch leicht verändert. Zur besseren Veranschaulichung wurden zwei Klassen vereiningt und nur zwei Dimensionen betrachtet. Auch wurden drei Punkte mit Doppelfärbung entfernt, um eine fehlerfreie Trennung zu ermöglichen. Zur Verfügung stehen 49 positive und 98 negative Trainingspunkte im Bereich $[4.3, 7.9] \times [1.0, 6.9]$.

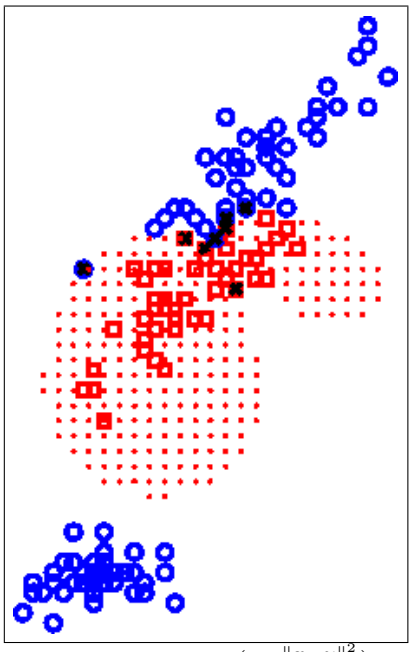
Auf diese Daten wird zum einen eine SVM mit Polynomialkern, zum anderen eine SVM mit Gaußkern angewendet. Es stellt sich heraus, dass eine fehlerfreie Trennung nur mit dem Gaußkern möglich ist, auch die Anzahl an Support Vektoren ist dort geringer als beim Polynomialkern.



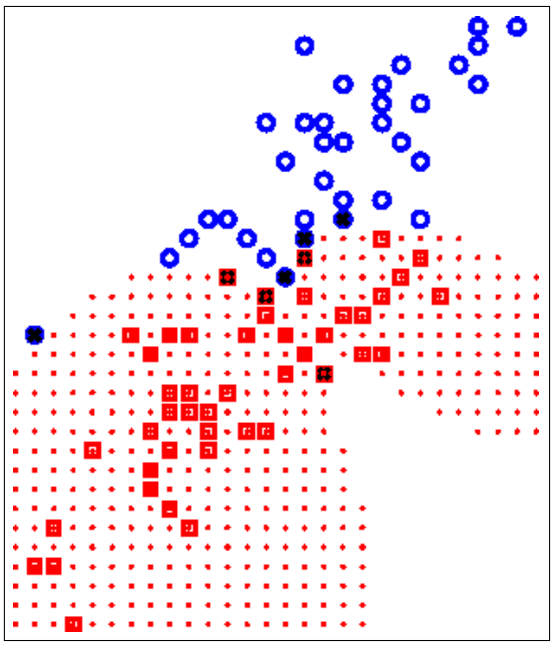
(a) $k(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + 5)^3$



(b) Ausschnitt: $\nu_{\text{opt}} = 0.033$, 1 Fehler, 9 SV'en.



(c) $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2}\right)$

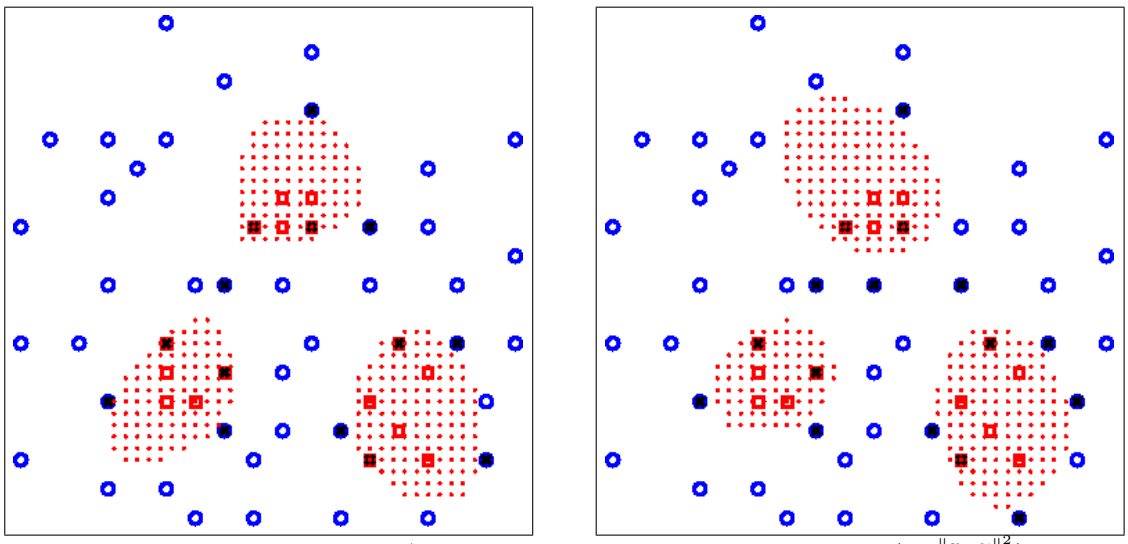


(d) Ausschnitt: $\nu_{\text{opt}} = 0.01$, 0 Fehler, 8 SV'en.

Abbildung A.1: Irisdaten mit Polynomial- und Gaußkern trainiert.

A.2 Inseldaten

Die Daten mit positivem Label bilden kleine „Inseln“ innerhalb der Daten mit negativem Label. Diese können schon durch ein Polynom vierten Grades erfasst werden. Der betrachtete Bereich ist $[0.75, 5.00] \times [0.75, 5.00]$. Insgesamt sind 58 Trainingspunkte vorhanden, davon sind 16 positiv und 42 negativ.



(a) $k(\boldsymbol{x}, \boldsymbol{y}) = (\langle \boldsymbol{x}, \boldsymbol{y} \rangle + 3)^4$ (b) $k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2}\right)$

Abbildung A.2: Inseldaten mit Polynomial- und Gaußkern trainiert.

Man kann erkennen, dass kein Trainingspunkt fehlklassifiziert wird, egal welcher der beiden Kerne verwendet wird. Allerdings unterscheiden sie sich in der Anzahl ihrer Support Vektoren. Sie beträgt beim Polynomialkern 14 und beim Gaußkern 17.

Prinzipiell schwankt der Anteil der Support Vektoren an den Gesamtdaten je nach Datenlage sehr stark. Liegt er im Falle der Irisdaten bei 6.12% (Polynomialkern) und 5.44% (Gaußkern), so beträgt er bei den Inseldaten 24.14% und 29.31%.

Literaturverzeichnis

[1] *Planes of the body.* – http://training.seer.cancer.gov/module_anatomy/unit1_3_-terminology2_planes.html - 07.11.2008

[2] Bennett, K. P. ; Campbell, C. : Support Vector Machines: Hype or Hallelujah? In: *SIGKDD Explorations* 2 (2000), Nr. 2, S. 1–13

[3] Christianini, N. ; Shawe-Taylor, J. : *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000

[4] Cover, T. M.: Geometrical and statistical properties of linear threshold devices. (1964)

[5] Fisher, R. A.: The use of multiple measurements in taxonomic problems. In: *Annual Eugenics* 7 (1936), Nr. 2, S. 179–188. – http://archive.ics.uci.edu/ml/datasets/Iris.html - 06.11.2008

[6] Geiger, C. : *Theorie und Numerik Restringierter Optimierungsaufgaben.* Springer, 2002

[7] Irle, A. : *Wahrscheinlichkeitstheorie und Statistik: Grundlagen – Resultate – Anwendungen.* Teubner, 2005

[8] Kiss, R. M. ; Kocsis, L. ; Knoll, Z. : Joint kinematics and spatial–temporal parameters of gait measured by an ultrasound-based system. In: *Medical Engineering and Physics* 26 (2004), Nr. 7, S. 611–620

[9] Müller, K. R. ; Mika, S. ; Rätsch, G. ; Tsuda, K. ; Schölkopf, B. : An introduction to kernel-based learning algorithms. In: *IEEE Neural Networks*, 181–201

[10] Schölkopf, B. ; Smola, A. J. ; Williamson, R. C. ; Bartlett, P. L.: New Support Vector Algorithms. In: *Neural Computation* 12 (2000), Nr. 5, S. 1207–1245

[11] Schölkopf, B. ; Smola, A. J.: *Learning with Kernels.* MIT Press, 2002

[12] Stahel, W. A.: *Statistische Datenanalyse: Eine Einführung für Naturwissenschaftler.* Vieweg, 2002

[13] Troje, N. F.: Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. In: *Journal of Vision* 2 (2002), Nr. 5, S. 371–387

[14] Vapnik, V. N.: *Statistical Learning Theory.* Wiley, 1998

[15] Vapnik, V. N.: *The Nature of Statistical Learning Theory.* 2. Springer, 1999

[16] zebris Medical GmbH: 3D Echtzeit Ganganalyse auf Laufband und Gehstrecke. In: *Produktinformation WinGait* (2005), S. 4

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit eigenständig und ausschließlich mit den angegebenen Quellen und Hilfsmitteln angefertigt habe.

Saarbrücken, den 28.11.2008

Sabrina Bechtel