Frame-Klassifizierung von Sprachsignalen mit Support-Vektor-Maschinen

Daniel Baumann

Semesterarbeit SA-2009-23

Herbstsemester 2009

Institut für Technische Informatik
und Kommunikationsnetze

Betreuer: T. Ewender und S. Hoffmann

Verantwortlicher: Prof. Dr. L. Thiele

Erklärung

Ich erkläre hiermit, das Merkblatt Plagiat[1] zur Kenntnis genommen, die vorliegende Arbeit selbständig verfasst und die im betroffenen Fachgebiet üblichen Zitiervorschriften eingehalten zu haben.

Zürich, 21. Dezember 2009

Daniel Baumann

[1] Merkblatt des Rektorats der ETH Zürich http://www.ethz.ch/students/semester/plagiarism_l_de.pdf

Inhaltsverzeichnis

Zusammenfassung

In dieser Arbeit wurde untersucht, inwieweit sich Support-Vektor-Maschinen (SVM) zur Klassifizierung von Sprachsignal-Frames eignen. SVM sind eine Methode zur Mustererkennung und Klassifizierung und sind grundsätzlich für binäre Klassifizierungen ausgelegt. Ausgehend vom sogenannten *one-against-one* Ansatz wurden verschiedene Strategien für die Multiclass-Klassifizierung untersucht. Jedoch haben die vorgeschlagenen Strategien keine Verbesserung gezeigt.

Weiter wurde das Verhalten der SVM mit unausgeglichenen und ausgeglichenen Trainingsdaten bezüglich der Multiclass-Strategien untersucht.
Als Vergleich zu den Resultaten wurden die Klassifizierungsergebnisse eines neuronalen Netzes betrachtet. Wie sich zeigte, erreicht man mit SVM in etwa gleich gute Resultate wie mit einem neuronalen Netz. Allerdings sind die mit SVM erreichten Resultate bei weitem nicht als abschliessend zu sehen, und eine Aussage darüber, welcher Klassifikator besser für die Frame-Klassifizierung geeignet ist, lässt sich nicht machen.

1 Einleitung

Die meisten heutigen Sprachsynthesesysteme basieren auf der Verkettung natürlicher Sprachabschnitte. Um eine gute Synthesequalität zu erreichen, müssen diese Signalabschnitte prosodisch verändert werden. Dazu wird detaillierte Information über die Signaleigenschaften benötigt. Die prosodische Veränderung kann beispielsweise mit der TD-PSOLA (*Time Domain Pitch Synchronous Overlap Add*) Methode geschehen. Die für dieses Verfahren benötigte Information über Sprachsignaleigenschaften wie etwa Stimmhaftigkeit und vorhandene Rauschanteile legt auch die gewählten Klassen der Klassifizierungsaufgabe fest.
Support-Vektor-Maschinen (SVM) sind eine Methode zur Mustererkennung und Klassifizierung. Im Bereich der Sprachklassifikation werden Support-Vektor-Maschinen unter anderem etwa zur Stimmhaft-/Stimmlos-Unterscheidung (siehe [8]) oder zur Sprache-/Nichtsprache-Unterscheidung (siehe [9]) eingesetzt.
Support-Vektor-Maschinen sind grundsätzlich binäre Klassifikatoren. Für die Anwendung auf Klassifizierungsprobleme mit mehreren Klassen gibt es verschiedene Ansätze, und wie SVM effizient auf solche Probleme übertragen werden können, ist noch Gegenstand der Forschung (siehe [5]).
Im Rahmen dieser Arbeit wurden leicht modifizierte Strategien für die Multiclass-Klassifizierung ausgehend vom sogenannten one-against-one Ansatz (siehe bspw. [5]) vorgeschlagen und untersucht.

Die Frame-Klassifizierung, wie sie in dieser Arbeit durchgeführt wurde, ist bereits mit einem neuronalen Netz realisiert worden (siehe [3]). Das Ziel dieser Semesterarbeit ist, unter Verwendung von Support-Vektor-Maschinen als Klassifikator, Sprachsignal-Frames bezüglich der benötigten Klassen zu unterscheiden. Dabei werden die in [3] beschriebenen und verwendeten Features übernommen.

Dieser Bericht ist wie folgt gegliedert:

Abschnitt 2 erläutert das Grundkonzept der SVM-Klassifizierung und wie sie auf Probleme mit mehr als zwei Klassen angewendet werden kann.

Abschnitt 3 erklärt kurz das erweiterte PSOLA-Verfahren als Motivation für die Frame-Klassifizierung. Weiter wird die Ausgangslage sowie das Vorgehen der Klassifizierung mit SVM dargelegt.

Abschnitt 4 stellt die Resultate der SVM-Klassifizierung vor. Weiter werden verschieden Strategien zur Verbesserung der Klassifizierungsergebnisse untersucht.

Abschnitt 5 macht einen kurzen Vergleich mit den Klassifizierungsergebnissen eines neuronalen Netzes und erläutert die wesentlichen Vor- und Nachteile der SVM-Klassifizierung gegenüber der Klassifikation mit einem neuronalen Netz.

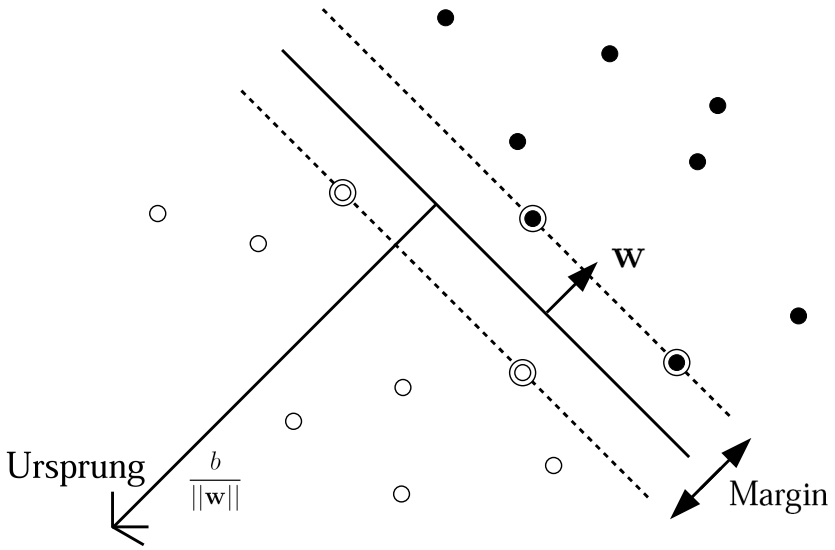
Abschnitt 6 zieht die Schlussfolgerungen und zeigt Möglichkeiten für weitere Verbesserungen auf.

2 Grundlagen: Support-Vektor-Machinen

Support-Vektor-Machinen (SVM) sind eine Technik zur Datenklassifizierung. Die Grundidee von SVM liegt in der Konstruktion einer optimalen Hyperebene, welche Objekte zweier Klassen mit maximal möglichem Abstand separiert (*maximum margine classifier*).
Eine Klassifizierungsaufgabe besteht grundsätzlich darin, aufgrund von Trainingsdaten, für welche die Zielklasse bekannt ist, ein Modell zu konstruieren, welches die Testdaten richtig klassifiziert. Zu jedem Trainingsdatum ist die zugehörige Klasse und eine Reihe von Attributen (die Features) gegeben. Das Ziel einer SVM ist nun, ein Modell zu erzeugen, welches die Testdaten aufgrund der Features richtig klassifiziert.

2.1 Lineare Support-Vektor-Maschinen

Im einfachsten Fall besteht das Klassifizierungsproblem aus Trainingsdaten zweier Klassen, die sich linear separieren lassen. Dies ist in Figur 1 illustriert.
Sei nun eine Menge von linear separierbaren Trainingsdaten $(\mathbf{x}_i, y_i), i = 1, \ldots, l$ mit $\mathbf{x}_i \in R^n$



Figur 1: *Linear separierbare Daten, Support-Vektoren eingekreist*

und $y_i \in \{1, -1\}$ gegeben, wobei $\mathbf{x}_i$ die Features als Vektor enthält und y_i das Klassenlabel ist. Für Punkte $\mathbf{x}$, die auf der trennenden Fläche liegen, gilt $\mathbf{x} \cdot \mathbf{w} + b = 0$, wobei $\mathbf{w}$ normal zur Fläche steht und $|b|/\,||\mathbf{x}||$ der senkrechte Abstand zum Ursprung ist. Linear separierbar heisst hier, dass ein $\mathbf{w}$ und b gefunden werden können, so dass für die Trainingsdaten folgende Einschränkung gilt

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \quad \text{für} \quad y_i = +1 \tag{1}$$
$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \quad \text{für} \quad y_i = -1 \tag{2}$$

welche folgendermassen zusammengefasst werden kann

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i. \tag{3}$$

Siehe dazu auch [1]. Ist eine Lösung des Problems gegeben (also ein bestimmtes $\mathbf{w}$ und b) lässt sich die Entscheidungsfunktion einfach beschreiben als

$$f(\mathbf{x}) = sgn(\mathbf{w} \cdot \mathbf{x} + b) \tag{4}$$

wobei $f(\mathbf{x}) \in \{\pm 1\}$ das Klassifizierungsergebnis ist und der Skalar $\mathbf{w} \cdot \mathbf{x} + b$ Entscheidungswert genannt wird. Die eigentliche Klassifizierung wird also dadurch bestimmt, auf welche Seite der Entscheidungsebene ein Punkt fällt.
Grundsätzlich kann es viele Lösungen geben, die die obigen Bedingungen erfüllen. Bei der SVM-Klassifizierung soll nun erreicht werden, dass der Abstand derjenigen Trainingsvektoren, die der Hyperebene am nächsten liegen, maximiert wird. Für diese Vektoren sind die Ungleichungen (1) und (2) mit Gleichheit erfüllt, und sie werden *Support-Vektoren* (SV) genannt (in Figur 1 eingekreiste Punkte). Der Abstand der Punkte, die der Entscheidungsebene am nächsten liegen, ergibt sich dann zu $2/\left|\left|\mathbf{w}\right|\right|$. Das Finden der separierenden Hyperebene mit maximalem Abstand (maximum margin) lässt sich also als Minimieren von $\left|\left|\mathbf{w}\right|\right|^2$ unter der Bedingung (3) verstehen:

$$\min_{\mathbf{w},b} \quad \frac{1}{2} \left|\left|\mathbf{w}\right|\right|^2 . \tag{5}$$

Dieses Problem lässt sich als duales Lagrange-Problem formulieren [1]. Für die Bedingungen (3) werden Lagrange-Multiplikatoren $\alpha_i, i = 1, \ldots, l$ eingeführt. Die primäre Lagrange-Funktion L_P ergibt sich gemäss [1] zu

$$L_P = \frac{1}{2} \left|\left|\mathbf{w}\right|\right|^2 - \sum_i \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_i \alpha_i \tag{6}$$

wobei nun L_P bezüglich $\mathbf{w}$ und b minimiert wird und $\alpha_i \geq 0$ gelten muss. Das duale Problem ist dann durch die Lagrange-Funktion L_D gegeben

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{7}$$

mit folgenden Nebenbedingungen

$$0 \leq \alpha_i \quad \forall i, \tag{8}$$

$$\sum_i \alpha_i y_i = 0, \tag{9}$$

wobei nun L_D bezüglich den α_i, unter den Bedingungen (8) und (9), maximiert wird. Es ist zu beachten, dass bei dieser Formulierung des Problems die Trainingsvektoren nur noch als Skalarprodukt in L_D erscheinen. Dieser Umstand erlaubt es, Kernelfunktionen direkt in das Optimierungsproblem einzusetzen, was im nächsten Abschnitt erläutert wird. Die Lösung des SVM-Problems ergibt sich dann als Linearkombination der Trainingsvektoren [1]:

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i \tag{10}$$

wobei für alle Trainingsvektoren, die am nächsten an der Entscheidungsgrenze liegen, d.h. die Support-Vektoren, $\alpha_i > 0$ gilt und für alle übrigen Trainingsvektoren $\alpha_i = 0$. Nur Trainingsvektoren, für welche $\alpha_i > 0$ gilt, fliessen in die Summe (10) mit ein. Die Support-Vektoren

legen also die Lösung des Optimierungsproblems fest, d.h. würden alle Trainingsvektoren bis auf die Support-Vektoren entfernt, bliebe sich das Resultat immer gleich.
Um auch Lösungen im Fall von nicht separierbaren Daten zuzulassen, wird das Optimierungsproblem (5) wie folgt verändert

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\left|\left|\mathbf{w}\right|\right|^2 + C\sum_i \xi_i \tag{11}$$

unter den Bedingungen

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) \geq 1 - \xi_i \quad \forall i, \tag{12}$$

$$\xi_i \geq 0 \quad \forall i. \tag{13}$$

Durch die Einführung der Variablen ξ_i wird die Bedingung (3) abgeschwächt, und es werden auch Trainingsfehler und Trainingspunkte innerhalb der Margin zugelassen. Über den Parameter C wird die Gewichtung von Trainingsfehlern festgelegt. Je grösser C gewählt wird, desto stärker fallen die Fehler ins Gewicht.
Auch dieses Problem lässt sich als duales Lagrange-Problem formulieren, und das sich daraus ergebende duale Problem ist nach [1]

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{14}$$

mit folgenden Nebenbedingungen

$$0 \leq \alpha_i \leq C, \tag{15}$$

$$\sum_i \alpha_i y_i = 0. \tag{16}$$

Die Lösung ist gegeben durch

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i. \tag{17}$$

Der einzige Unterschied zum Fall mit linear separierbaren Daten ist die obere Begrenzung der α_i durch C.
Eine weitere mögliche Modifizierung des Optimierungsproblems (11) ist die gewichtete SVM, wie sie beispielsweise in [2] beschrieben wird. Dabei werden die Trainingsfehler der beiden Klassen unterschiedlich gewichtet und das Optimierungsproblem lässt sich wiefolgt neu fomulieren [2]

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\left|\left|\mathbf{w}\right|\right|^2 + C_+ \sum_{i:y_i=+1} \xi_i + C_- \sum_{i:y_i=-1} \xi_i \tag{18}$$

wobei die Bedigungen (12) und (13) gleich bleiben.

2.2 Feature-Space und Kernels

Der bisher beschriebene Klassifikator erlaubt nur lineare Entscheidungsfunktionen. Um auch nicht lineare Entscheidungsfunktionen zu ermöglichen, werden die Trainingsdaten durch eine

Funktion ϕ in einen höher dimensionalen (evlt. unendlich-dimensionalen) euklidschen Raum (den *Feature-Space*) abgebildet und in diesem Raum ein linearer Klassifikator gesucht, wie dies im vorhergehenden Abschnitt beschrieben wurde. Dadurch werden linear nicht separierbare Daten möglicherweise linear separierbar, und es ergeben sich auch nicht lineare Entscheidungsfunktionen.
Wie im vorhergehenden Abschnitt erwähnt, erscheinen die Trainingsvektoren im dualen Problem nur als Skalarprodukte, im Feature Space also als $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$. Falls nun eine Kernelfunktion K mit

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \tag{19}$$

existiert, lässt sich diese auch ohne explizite Kenntnis von ϕ einfach auf das Optimierungsproblem anwenden. Im Zusammenhang mit SVM werdem zumeist folgende Kernels verwendet:

- linearer Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$.
- Polynom-Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma\, \mathbf{x}_i \cdot \mathbf{x}_j + r)^d, \gamma > 0$.
- Gauss-Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\ ||\mathbf{x}_i - \mathbf{x}_j||^2\right), \gamma > 0$.
- Sigmoid-Kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\gamma\, \mathbf{x}_i \cdot \mathbf{x}_j + r\right)$.

Damit eine bestimmte Funktion K von der Form (19) ist, müssen bestimmte Bedingungen erfüllt sein (siehe [1]).

Für einen Kernel der Form (19) wird das duale Promlem (14) zu

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{20}$$

wobei die Nebenbedingungen (15) und (16) gleich bleiben. Um diesen ”kernel trick” auf das SVM-Problem anzuwenden, muss nun aber auch $\mathbf{w}$, und somit die Entscheidungsfunktion, ohne explizite Kenntnis von ϕ bestimmt werden können. Sei $\mathbf{x}$ ein Testpunkt, dann folgt aus (17)

$$\mathbf{w} \cdot \phi(\mathbf{x}) = \sum_i \alpha_i y_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \tag{21}$$

und die Entscheidungsfunktion ist gegeben durch

$$f(\mathbf{x}) = sgn\left(\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{22}$$

Somit kann auch ein Testpunkt ohne Kenntnis von ϕ klassifiziert werden.

2.3 Multiclass-Klassifizierung

Grundsätzlich sind SVM für binäre Klassifizierungen ausgelegt. Wie sie effizient auf Probleme mit mehr als zwei Klassen angewendet werden können, ist noch Gegenstand der Forschung [5]. Im Folgenden werden die für diese Arbeit relevanten Ansätze kurz vorgestellt.
Ein einfacher und naheliegender Ansatz ist die sogenannte *one-against-all* Methode. Dabei werden K verschiedene SVM trainiert, wobei K die Anzahl Klassen ist. Die i-te SVM wird mit allen Trainingsdaten der i-ten Klasse mit positiven Labels und alle übrigen Trainingsdaten mit negativen Labels trainiert und sei durch $(\mathbf{w}^i, b^i)$ gegeben. Für einen Testpunkt $\mathbf{x}$ gibt es somit K Entscheidungsfunktionen

$$\begin{matrix} \mathbf{w}^1\phi(\mathbf{x}) + b^1 \\ \vdots \\ \mathbf{w}^K\phi(\mathbf{x}) + b^K \end{matrix} \tag{23}$$

und $\mathbf{x}$ ghört dann zu derjenigen Klasse, für welche die entsprechende SVM den grössten Entscheidungswert liefert. Dies resultiert in folgender Entscheidungsfunktion

$$\text{Klasse von} \quad \mathbf{x} = \underset{i=1,\ldots,K}{\arg\max}(\mathbf{w}^i \cdot \mathbf{x} + b^i). \tag{24}$$

Das Multiclass-Problem wird also mit K binären Problemen beschrieben.

Eine weitere Möglichkeit ist die sogenannte *one-against-one* Methode. Bei diesem Ansatz werden $K(K-1)/2$ Modelle trainiert, wobei jedes Modell jeweils mit Daten von genau zwei Klassen trainiert wird. Dadurch ist jedoch noch keine Entscheidungsfunktion festgelegt, und für die Klassifizierung wird eine Wahlstrategie verwendet, wie sie im Folgenden beschrieben wird. Sei nun die Entscheidungsfunktion der SVM, trainiert mit den Daten der i-ten und j-ten Klasse, gegeben durch $\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}$. Die Klassifizierung von $\mathbf{x}$ wird nun durch folgende Wahlstrategie bestimmt:

> Ist $\mathbf{x}$ gemäss $\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}$ in der i-ten Klasse, erhält diese Klasse eine Stimme, ansonsten erhält die j-te Klasse eine Stimme. Dieses Verfahren wird mit allen $K(K-1)/2$ binären SVM durchgeführt und $\mathbf{x}$ wird als die Klasse mit den meisten Stimmen klassifiziert.

Diese Strategie wird im Folgenden als MaxWins bezeichnet. Ein Vergleich dieser beiden sowie noch weiterer Methoden für Multiclass-Klassifizierung ist in [5] zu finden.

3 Frame-Klassifizierung mit SVM

In diesem Kapitel wird als Motivation für die Frame-Klassifizierung eine Methode zur Prosodiesteuerung vorgestellt, das sogenannte TD-PSOLA-Verfahren. Im weiteren werden die verwendeten Klassen und Features eingeführt und das Vorgehen der Frame-Klassifizierung mit SVM dargelegt.

3.1 Motivation: Prosodiesteuerung mit TD-PSOLA

Ein verbreiteter Ansatz bei heutigen Sprachsynthesesystemen ist der sogenannte Verkettungsansatz. Dabei wird ein Sprachsignal durch Aneinanderfügen vorhandener Grundelemente (z.B Diphone) erzeugt, siehe dazu [4]. Um eine gute Synthesequalität zu erreichen, müssen die verwendeten Grundelemente bezüglich Intensität, Dauer und Grundgrequenz angepasst werden. Diese im Sprachsignal messbaren physikalischen Grössen werden prosodische Grössen genannt, und die Anpassung dieser Grössen wird in der Sprachsynthese als Prosodiesteuerung bezeichnet.
Eine prosodische Veränderung von Signalen kann beispielweise mit dem TD-PSOLA Verfahren durchgeführt werden. Wie ein Signal im einzelnen verändert wird, um die gewünschte Anpassung zu erreichen, hängt bei TD-PSOLA wesentlich von Signaleigenschaften wie Stimmhaftigkeit und Rauschanteil ab; also ob ein Signal als quasi stationär oder rauschartig betrachtet werden kann.
Das TD-PSOLA-Verfahren, wie es in [7] beschrieben wird, unterscheidet bei seiner Anwendung zwischen stimmhaften und stimmlosen Partien. Diese Unterscheidung ist notwendig, da bei einer Verlängerung der Dauer eines Signalabschnittes bei einem stimmlosen Signalabschnitt das zu verdoppelnde Segment mit umgekehrter Zeitachse wiederholt werden muss, ansonsten entsteht eine künstliche, wahrnehmungsmässig stark störende Periodizität.

Das im Folgenden erläuterte erweiterte TD-PSOLA-Verfahren unterscheidet nicht nur zwischen stimmhaften und stimmlosen Partien, sondern basiert auf einer detaillierten Frame-Klassifizierung mit fünf verschiedenen Klassen: **stimmhaft**, **stimmlos**, **gemischt**, **unregelmässig** und **silence**. Mit Hilfe dieser detaillierten Unterscheidung kann eine differenziertere Modifikation der Grundfrequenz erfolgen, um möglicherweise dadurch entstehende Artefakte zu vermeiden. Eine Definition dieser verschiedenen Klassen findet sich in Abschnitt 3.2.

Ähnlich wie im Standardverfahren wird in den Signalabschnitten der Klassen **stimmhaft**, **gemischt** und **unregelmässig** der Anfang jeder Periode markiert, Signalabschnitte der Klassen **stimmlos** und **silence** werden in Intervalle fester Länge unterteilt. Je zwei benachbarte Abschnitte werden sodann mit einer Hanning-Fensterfunktion multipliziert. Aus der so entstandenen Folge von Doppelperiodensegmenten erzeugt nun das TD-PSOLA-Verfahren prosodisch veränderte Sprachsignale. Das erweiterte TD-PSOLA-Verfahren verfährt mit Partien der Klassen **stimmhaft**, **stimmlos** und **silence** wie das Standardverfahren (siehe [7]) und unterscheidet sich nur in der ergänzenden Behandlung von Partien der Klassen **gemischt** und **unregelmässig**:

- **gemischt**: Die Dauer von Signalabschnitten dieser Klasse kann nur verkürzt und nicht verlängert werden. Eine Verlängerung des Signals durch Wiederholung von Doppelperiodensegmenten ohne Spiegelung des wiederholten Doppelperiodensegments würde wie

bei rein stimmlosen Partien zu Entstehung von künstlicher Periodizität führen. Eine Spiegelung des wiederholten Doppelperiodensegments würde jedoch die vorhandene Periodizität verändern. Aus diesem Grund muss bei Signalen dieser Klasse auf eine Verlängerung des Signal verzichtet werden. Die Modifikation der Grundfrequenz geschieht wie im Standardverfahren.

- **unregelmässig**: Signalabschnitte dieser Klasse können verkürzt und verlängert werden. Eine Grundfrequenz ist für Signalabschnitte dieser Klasse jedoch nicht definiert und somit auch nicht notwendig. Eine Modifikation der Grundwelle wäre aber auch nicht möglich: Die Perioden variieren in diesen Signalabschnitten sehr stark, so dass die gewonnenen Doppelperiodensegmente stark asymmetrisch sein können, wobei der Glottispuls durch die Multiplikation mit dem Hanning-Fenster unregelmässig gedämpft wird.

Durch die Frame-Klassifizierung sollen Sprachsignale nach vorig genannten Gesichtspunkten aufgeteilt werden. Die Frame-Klassifizierung liefert also Information wie ein Signal mit dem TD-PSOLA Verfahren prosodisch zu verändern ist.

3.2 Klassen und Features

Bei der Klassifizierung wurden die fünf unten beschriebenen Klassen verwendet. Motiviert ist die Aufteilung in diese fünf Klassen dadurch, dass sie bei der prosodischen Veränderung verschieden zu behandeln sind, wie dies im vorhergehenden Abschnitt erläutert wurde.
Die fünf Klssen sind:

stimmhaft: Klar stimmhafte Sprache, harmonisches Signal, keine Rauschanteile, niedrige Frequenzen dominieren (typische Phoneme: Vokale, Nasale)

stimmlos: Stimmhaftigkeit nicht erkennbar, rauschartig, hohe Frequenzen über 2kHz sind dominierend (typische Phoneme: Frikative, stimmlose Plosive)

gemischt: Sprache mit stimmhaften Anteilen und Rauschanteil; nur die niedrigsten Harmonischen im Spektrum erkennbar, höhere spektrale Komponenten rauschhaft (typische Phoneme: stimmhafte Frikative, häufig auch in Übergängen zwischen stimmhaft und stimmlos)

unregelmässig: Sprache mit unregelmässigen Glottisschlägen, keine bedeutenden Frikativanteile, niedrige Frequenzen domonierend; auch bekannt als Knarrstimme

silence: Signalabschnitte mit sehr niedriger Energie, nicht hörbar

Für die Klassifizierung wurden 10 Features pro Frame verwendet, wobei ein Frame eine Dauer von 50 ms hat und um jweils 5 ms verschoben wird. Die Herleitung und Beschreibung der Features kann [3] entnommen werden. Für diese Arbeit wurden die Features ohne weitere Veränderung übernommen.

Die verwendeten Features sind:

1. Zero crossing rate
2. Sprachsignalleistung (logarithmisch)
3. erster MFCC
4. zweiter MFCC
5. Periodizität

wobei MFCC für die Koeffizienten des Mel-Cepstrums steht (engl. *mel frequency cepstral coefficients*). Die nicht namentlich aufgelisteten Features Nummer 6 bis 10 beschreiben die Dynamik der Grundwelle (siehe [3]).

3.3 Verwendetes SVM-Modell und Vorgehen

Für die Klassifizierung der Frames mittels SVM als Klassifikator wurde LIBSVM [2] verwendet. Diese Software implementiert die in Abschnitt 2 beschriebene SVM.
Bei der Wahl des SVM-Modells muss entschieden werden, welche Kernelfunktion benutzt wird, und abhängig von der Kernelfunktion müssen die Kernelparameter gewählt werden. In dieser Arbeit wurde nur der Gauss-Kernel verwendet (siehe Abschnitt 2.2). Die Gründe dafür sind unter anderem, dass dadurch nur ein weiterer Parameter eingeführt wird, der die Komplexität des Modells beeinflusst und der Gauss-Kernel gute numerische Eigenschaften zeigt [6].
Für die Multiclass-Klassifizierung wurde der one-against-one Ansatz verwendet. Da sich nach ersten Versuchen mit dem one-against-all Ansatz zeigte, dass mit diesem keine besseren Resultate erreicht werden, wurde entschieden, sich auf die one-against-one Methode zu konzentrieren. Diese Methode zeigt auch wesentlich kürzere Trainingszeiten [5]. Ausgehend von der one-against-one Methode und der Wahlstrategie, wie sie in Abschnitt 2.3 beschrieben wird, wurden in dieser Arbeit noch zwei weitere Wahlstrategien verwendet. Diese werden im Abschnitt 4.4 erläutert.
Für die gewichtete SVM (siehe Abschnitt 2.1) wurde eine leicht veränderte Formulierung verwendet:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\,||\mathbf{w}||^2 + w_{+1}C \sum_{i:y_i=+1} \xi_i + w_{-1}C \sum_{i:y_i=-1} \xi_i. \tag{25}$$

Dadurch bleibt der Parameter C für alle binären SVM gleich.

Um die Verwendung von SVM für die hier gestellte Klassifizierungsaufgabe zu untersuchen, wurde im wesentlichen in zwei Schritten vorgegangen.
In einem ersten Schritt wurden für die Untersuchung der Wahlstrategien und der Verwendung von gewichteten SVM die Resultate durch Kreuzvalidierung ermittelt, wobei der Parameter C und der Kernelparmeter γ jeweils fixe Werte haben. Bei der n-fachen Kreuzvalidierung werden die Trainingsdaten in n Teilmengen aufgeteilt. Dann wird der Reihe nach jeweils eine Teilmenge zum Testen des mit den übrigen $n-1$ Teilmengen trainierten Klassifikators benutzt. Dieser Schritt umfasst den grössten Teil dieser Arbeit.

In einem zweiten Schritt müssen die im ersten Schritt fix gehaltenen Parameter möglichst optimal gewählt werden. Unter Parameterselektion wird die Wahl des Parameterpaars (C, γ) verstanden. In dieser Arbeit wurde das in [6] beschriebene "grid-search"-Verfahren verwendet: In einem bestimmten vorgegebenen Suchraum für das Parameterpaar (C, γ) wird für jede mögliche Kombination der Parameter mittels Kreuzvalidierung die Klassifizierungsgenauigkeit bestimmt. Es werden diejenigen Parameter (C, γ) gewählt, für welche die Genauigkeit am höchsten ist.

Weiter ist zum Vorgehen zu bemerken, dass zumeist mit skalierten Daten gearbeitet wurde. Dabei werden die einzelnen Features auf den gleichen numerischen Bereich skaliert. Hauptsächlich wird die Skalierung durchgeführt, um zu verhindern, dass Features in einem grösseren numerischen Bereich gegenüber solchen in einem kleinen numerischen Bereich dominieren [6].

4 Auswertung und Diskussion

4.1 Verwendetes Sprachmaterial

Das verwendete Sprachmaterial besteht aus Sprachsignalen von 20 verschiedenen Stimmen aus 12 europäischen und asiatischen Sprachen. Pro Stimme stehen jweils drei Sätze mit Referenzklassifizierung zur Verfügung.
Das Sprachmaterial wurde wie folgt aufgeteilt: 16 der 20 Stimmen wurden verwendet, um mittels Kreuzvalidierung die verschiedenen Wahlstrategien und Gewichtungen zu testen.
Mit den Daten der übrgien vier Stimmen wurden Modelle getestet, die nach der Parameterselektion mit den Daten aller 16 Stimmen trainiert wurden.
In dieser Arbeit wurde das Datenmaterial für die Kreuzvalidierung nach Stimmen aufgeteilt, um möglichst stimmenunabhängige Resultate zu erreichen.
Sofern dies nicht anders angegeben ist, wurden die Features auf den Bereich $[-1, +1]$ skaliert.

4.2 Klassifizierung mit MaxWins-Strategie

In diesem Abschnitt wurde die in Abschnitt 2.3 beschriebene Wahlstrategie MaxWins verwendet. Für die n-fache Kreuzvalidierung wurden die 16 Stimmen in n Teilgruppen aufgeteilt. Daraus ergibt sich, dass n 16 teilen muss und jeweils mit $((n-1)/n)16$ Stimmen trainiert wird.
Die Parameter C und γ haben die fixen Werte $C = 1$ und $\gamma = 1/10$.
In einem ersten Versuch wurden die Genauigkeiten für alle möglichen Werte von n einmal mit skalierten und einmal mit unskalierten Daten ermittelt. Da die verwendeten Daten nach Klassen unausgeglichen sind, werden in den Tabellen 1 und 2 , und allen weiteren Angaben von Gesamtgenauigkeiten, die über die Genauigkeiten der einzelnen Klassen gemittelten Genauigkeiten angegeben.

n	Genauigkeit in %
2	74.34
4	75.32
8	75.96
16	75.99

Tabelle 1: *n-fache Kreuzvalidierungsgenauigkeit, unskalierte Daten*

n	Genauigkeit in %
2	78.01
4	78.70
8	79.02
16	79.00

Tabelle 2: *n-fache Kreuzvalidierungsgenauigkeit, skalierte Daten*

Wie den Resultaten zu entnehmen ist, kann durch die Skalierung eine Verbesserung von 3 bis 4 % erreicht werden. Um den Einfluss der Anzahl Stimmen, die in den Trainingsdaten vorhanden sind, zu untersuchen, wurde die Trainings- und Testmenge vertauscht. Das Training wird also nur mit einer von n Teilgruppen durchgeführt (d.h. $16/n$ Stimmen), und die übrigen $n - 1$ Teilgruppen werden als Testdaten verwendet. Die Resultate sind in den Tabellen 3 und 4 angegeben.

Es zeigt sich, dass für $n = 16$, d.h. es wird nur eine Stimme zum Trainieren verwendet, die Genauigkeit deutlich tiefer ausfällt. Allerdings fallen die Resultate nicht extrem schlechter

n	Genauigkeit in %
4	72.97
8	71.11
16	69.34

Tabelle 3: *n-fache Kreuzvalidierungsgenauigkeit mit vertauschten Trainings- und Testdaten, unskalierte Daten*

n	Genauigkeit in %
4	76.57
8	72.57
16	70.85

Tabelle 4: *n-fache Kreuzvalidierungsgenauigkeit mit vertauschten Trainings- und Testdaten, skalierte Daten*

aus, wenn nur wenige Stimmen zum Trainieren verwendet werden. Dies zeigt, dass SVM relativ stabile Resultate liefern, auch wenn nur wenige Stimmen für das Training verwendet werden.

Als nächstes wird nun die Klassifizierung der einzelnen Klassen betrachtet. Tabelle 5 zeigt die Klassifizierungsgenauigkeit je Klasse in der Konfusionsmatrix.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	91.87	3.19	0.14	3.17	1.63
silence	1.66	96.96	0.40	0.11	0.88
stimmhaft	0.04	0.22	98.27	0.52	0.95
gemischt	23.41	4.35	17.50	48.84	5.91
irregulär	3.56	10.47	23.31	3.52	59.15

Tabelle 5: *Konfusionsmatrix, $n = 8$, skalierte Daten, Angaben in %*

Es ist zu sehen, dass die Klassen **stimmlos**, **silence** und **stimmhaft** gut klassifiziert werden, wobei das Resulttat für **stimmhaft** mit 98.27 % am besten ausfällt. Für die Klassen **gemischt** und **irregulär** fällt das Resultat am schlechtesten aus. Die Klasse **gemischt** wird oft mit **stimmlos** (zu 23.41 %) und **stimmhaft** (zu 17.84 %) verwechselt. Dies ist nicht erstaunlich, da die Klasse **gemischt** Eigenschaften mit beiden anderen Klassen teilt und eine Unterscheidung nicht immer eindeutig ist. Die Klasse **irregulär** wird häufig (zu 23.31 %) als **stimmhaft** klassifiziert. Diese beiden Klassen haben ebenfalls gemeinsame Eigenschaften, wie etwa die Dominanz niedriger Frequenzen und im Gegensatz zu **stimmlos** und **gemischt** geringe Frikativanteile.
Es ist also davon auszugehen, dass sich gewisse Klassen besser voneinander unterscheiden lassen als andere. Um nun eine Aussage über die Separierbarkeit der Klassen zu machen, wird die Anzahl SV untersucht. Die Anzahl SV der einzelnen binären Klassifikatoren gibt einen Hinweis über die gegenseitige Separierbarkeit der einzelnen Klassen. Tabelle 7 zeigt die Anzahl SV der binären SVM. Die angegebenen Werte sind nicht ganzzahlig, da sie dem Mittelwert der 8-fachen Kreuzvalidierung entsprechen. Der Übersichtlichkeit halber wird dabei folgende Notation verwendet: Eine binäre SVM, trainiert mit den Daten zweier Klassen, wird mit $i \,\mathrm{v.}\, j$ (i versus j) bezeichnet, wobei i und j die Labels der entsprechenden Klassen sind. Im Folgenden wird, falls die Klassen nicht explizit angegeben werden, die Zuordnung von Klassen und Klassenlabels gemäss Tabelle 6 verwendet.
Aus den Resultaten in Tabelle 7 ist zu sehen, dass die SVM $-1 \,\mathrm{v.}\, 1$ am wenigsten und die SVM $-1 \,\mathrm{v.}\, 2$ am meistem SV hat. Dies deckt sich auch mit den Resultaten aus Tabelle 5, die zeigen, dass sich die Klassen **stimmlos** und **stimmhaft** gut unterscheiden lassen und **gemischt** häufig als **stimmlos** fehlklassifiziert wird. Allerdings zeigt sich auch, dass eine grosse Anzahl

Klasse	Klassenlabel
stimmlos	-1
silence	0
stimmhaft	1
gemischt	2
irregulär	3

Tabelle 6: *Zuordnung von Klassen und Klassenlabels*

SVM	-1 v. 0	-1 v. 1	-1 v. 2	-1 v. 3	0 v. 1	0 v. 2	0 v. 3	1 v. 2	1 v. 3	2 v. 3
# SV	1376.4	543.0	2360.9	944.8	722.9	832.0	1165.4	1952.8	2183.0	1388.6

Tabelle 7: *Anzahl SV der einzelnen binären SVM,* $n = 8$

SV nicht zwingend bedeutet, dass beide Klassen schlecht klassifiziert werden. So wird zwar **gemischt** häufig als **stimmlos** fehlklassifiziert, nicht jedoch **stimmlos** als **gemischt**. Dasselbe gilt etwa für **irregulär** und **stimmhaft**, obwohl die beiden SVM -1 v. 2 und 1 v. 3 über 2000 SV haben.
Gründe für dieses Verhalten liegen nicht nur in der schlechten Separierbarkeit gewisser Klassen sondern auch darin, wie die eigentliche Klassifizierung mit der MaxWins-Strategie zustande kommt.

Im weiteren wird untersucht, wie die einzelnen SVM ihre Stimmen gemäss der MaxWins-Strategie vergeben. Tabelle 8 zeigt den Anteil positiver Entscheidungen der einzelnen SVM in Abhängigkeit der Zielklasse. Eine positive Entscheidung bedeuted hier, dass sich die SVM für die erst genannte Klasse entscheidet, bspw. entscheidet die SVM -1 v. 0 zu 96.47 % auf **stimmlos**, wenn die Zielklasse des Testpunktes **stimmlos** ist.
Wird nun das Entscheidungsverhalten der einzelnen SVM bezüglich Daten der Klasse

	-1 v. 0	-1 v. 1	-1 v. 2	-1 v. 3	0 v. 1	0 v. 2	0 v. 3	1 v. 2	1 v. 3	2 v. 3
-1	96.47	99.05	95.90	97.44	69.49	13.85	61.51	0.39	4.03	96.15
0	1.86	91.30	80.48	76.57	99.48	99.56	98.84	7.95	12.01	61.69
1	95.75	0.20	0.12	0.64	0.37	1.12	0.50	99.31	98.90	72.47
2	93.59	57.75	26.70	69.60	19.20	6.99	20.25	20.76	57.32	87.93
3	66.27	28.36	22.56	5.94	26.25	29.20	11.43	70.71	26.17	9.19

Tabelle 8: *Positive Entscheidungen der einzelnen binären SVM in % ,* $n = 8$

gemischt betrachtet, lässt sich Folgendes beobachten: Die vier SVM, die mit Daten dieser Klasse trainiert wurden, entscheiden sich alle mit über 70 % auch für die Klasse **gemischt**. Die SVM -1 v. 0 stimmt zu 93.59 % und die SVM -1 v. 3 zu 69.60 % für die Klasse **stimmlos**.
Ein Grund für die schlechte Klassifizierung der Klassen **gemischt** und **irregulär** scheint also auch die Wahlstrategie MaxWins zu sein. Die binären SVM, die nicht mit Daten dieser Klassen trainiert wurden, können in ihrem Entscheidungsverhalten stark zu einer bestimmten Klasse tendieren und somit die Klassifizierung der anderen SVM überstimmen. Um dieses Verhalten abzuschwächen, wurden andere Wahlstrategien untersucht. Diese werden in Abschnitt 4.4 erläutert.

Ein weiterer zu beachtender Punkt ist die Verteilung der Klassen in den Testdaten. Tabelle 9 zeigt die Verteilung der Klassen in den Trainingsdaten (die Daten der ersten 16 Stimmen).
Durch die stark ungleiche Verteilung der Klassen wird in der Entscheidung der binären SVM

Klasse	*stimmlos*	*silence*	*stimmhaft*	*gemischt*	*irreguär*	Total
# Daten	7921	12611	29080	2760	2274	54646

Tabelle 9: *Anzahl Daten nach Klassen*

eine a priori Wahrscheinlichkeit zu Gunsten der Klasse mit mehr Trainingsdaten bewirkt. Es zeigt sich hier auch, dass für die Klasse **stimmlos** trotz weniger Trainingsdaten immer noch eine gute Vorhersage möglich ist. Die schlechten Resultate müssen also nicht zwingend daher rühren, dass zu wenig Trainingsdaten vorhanden sind. Einen wesentlichen Einfluss hat grundsätzlich sicher auch die Separierbarkeit von den anderen Klassen.

Die Probleme des verwendeten Klassifikators und Multiclass-Ansatzes liegen zum einen also darin, dass sich gewisse Klassen schlecht separieren lassen und die Wahlstrategie unerwünschte Effekte zeigt, und zum anderen darin, dass die Trainingsdaten nach Klassen unausgeglichen sind.
Im nächsten Abschnitt wird nun mit gewichteten SVM versucht, bessere Resultate zu erreichen, ohne die Trainingsdaten auszugleichen. Und anschliessend wird im Abschnitt 4.4 versucht, mittels modifizierter Wahlstrategien eine Verbesserung der Multiclass-Klassifizierung zu erreichen. Im Abschnitt 4.5 wird versucht, das Problen der unausgeglichenen Daten durch Ausgleichen der Trainingsdaten zu lösen.

4.3 Klassifizierung mit gewichteter SVM

Um die ungleiche Verteilung in den Trainingsdaten auszugleichen, können die Trainingsfehler für jede Klasse anders gewichtet werden. Für die gewichtete SVM wurde die Formulierung (25) verwendet. Die gewichtete SVM kann zum Einsatz kommen, wenn die Trainingsdaten bezüglich Klassen unausgeglichen sind oder auch wenn die Entscheidung in Richtung einer Klasse verschoben werden soll.
Im Folgenden wurde diese Gewichtung für drei Varianten durchgeführt. Die Werte der einzelnen Gewichte sind in Tabelle 10 angegeben, wobei w_i das Gewicht der Klasse mit Klassenlabel i nach Tabelle 6 ist.

Gewichtung	w_{-1}	w_0	w_1	w_2	w_3
prop	4	2	1	11	13
rez	0.27	0.43	1	0.09	0.08
quad	14	6	1	112	164

Tabelle 10: *Parameter w_i der drei verwendeten Gewichtungen*

Die Gewichtung *prop* heisst, dass die Gewichte umgekehrtproportional zur Anzahl Trainingsdaten gewählt wurden. Bei der Gewichtung *rez* wurden die Kehrwerte und bei der Gewichtung *quad* die quadrierten Werte der Gewichtung *prop* verwendet. Für *prop* und

quad wurden die Gewichte auf den nächsthöheren ganzzahligen Wert gerundet, und alle Gewichtungen wurden bezüglich w_2 normiert. Die Resultate wurden wie im vorhergehenden Abschnitt für $n = 8$ mit Kreuzvalidierung ermittelt, und es gilt weiterhin $C = 1$ und $\gamma = 1/10$. Die über Klassen gemittelte Klassifizierungsgenauigkeiten der drei Gewichtungen sind der Tabelle 11 zu entnehmen.

Gewichtung	Genauigkeit in %
prop	86.23
rez	60.08
quad	84.19

Tabelle 11: *Kreuzvalidierungsgenauigkeit, $n = 8$, mit Gewichtung*

Es ist klar, dass die Gewichte so gewählt werden müssen, dass die Trainingsfehler derjenigen Klassen, die in den Trainingsdaten untervertreten sind, stärker gewichtet werden. Es ist desshalb auch nicht überraschend, dass das Ergebnis für die Gewichtung *rez* am schlechtesten ausfällt. Diese Gewichtung wurde gewählt, um einen besseren Eindruck des Einflusses der Gewichte zu bekommen. Die Tabellen 12 bis 14 zeigen das Klassifizierungsergebnis der drei Gewichtungen als Konfusionsmatrix.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	86.39	1.64	0.00	9.28	2.69
silence	2.13	92.52	0.02	0.81	4.52
stimmhaft	0.00	0.06	90.81	3.65	5.48
gemischt	10.07	1.16	3.04	74.78	10.94
irregulär	1.10	2.73	3.03	6.51	86.63

Tabelle 12: *Konfusionsmatrix, mit Gewichtung prop, Angaben in %*

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	93.65	4.13	2.12	0.05	0.05
silence	1.30	97.51	1.19	0.00	0.00
stimmhaft	0.05	0.18	99.77	0.00	0.01
gemischt	39.60	5.07	47.32	7.83	0.18
irregulär	8.40	17.33	72.65	0.00	1.63

Tabelle 13: *Konfusionsmatrix, mit Gewichtung rez, Angaben in %*

Für die Klassen **gemischt** und **irregulär** ist mit der Gewichtung *prop* eine klare Verbesserung der Klassifizierungsgenauigkeit zu erreichen. Allerdings nimmt die Genauigkeit der anderen Klassen auch ab. Die Resultate der Gewichtung *rez* zeigen, dass die Wahl der Gewichte für die in den Trainingsdaten am schwächsten vertretenen Klassen **gemischt** und **irregulär** einen deutlichen Einfluss auf das Klassifizierungsresultat hat. Die Daten dieser Klassen lassen sich bei dieser Gewichtung offenbar nicht mehr klassifizieren.
Für die Gewichtung *quad* zeigt sich ein ähnliches Verhalten wie bei *prop*. Es ist aber eine

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	78.02	1.20	0.00	16.78	4.00
silence	1.93	88.36	0.00	1.40	8.31
stimmhaft	0.00	0.01	77.67	9.53	12.80
gemischt	2.97	0.36	0.43	84.28	11.96
irregulär	0.26	0.75	0.09	6.29	92.61

Tabelle 14: *Konfusionsmatrix, mit Gewichtung quad, Angaben in %*

deutliche Verschiebung der Genauigkeit zugunsten der am stärksten gewichteten Klassen **gemischt** und **irregulär** festzustellen. Die Klassifizierungsgenauigkeit der übrigen Klassen nimmt aber deutlich ab.
Von den untersuchten Gewichtungen erreicht *prop* die besten Resultate hinsichtlich der über Klassen gemittelten Genauigkeit, als auch der Ausgeglichenheit der Genauigkeit der einzelnen Klassen.
Wie die Resultate zeigen, kann durch die Gewichtung die Entscheidung zu Gunsten gewisser Klassen verschoben werden. Dies kann sinnvoll sein, wenn die Fehlklassifizierung gewisser Klassen schwerer wiegt als die Fehlklassifizierung anderer.

4.4 Modifizierte Wahlstrategien

Da die Entscheidung der SVM grundsätzlich eine Vorzeichenentscheidung ist, werden klare und knappe Entscheidungen gleich gewichtet. Es spielt also keine Rolle, ob ein Testpunkt nahe oder weit entfernt von der Entscheidungsebene liegt. Um die Entscheidungen der binären SVM entsprechend dieser Entfernung von der Entscheidungsebene zu gewichten, werden in diesem Abschnitt zwei weitere Wahlstrategien betrachtet. Diese Strategien gewichten die Stimmen der binären SVM entsprechend den Entscheidungswerten. Ausgehend von der one-against-one Methode wird die MaxWins-Wahlstrategie (siehe Abschnitt 2.3) für die Klassifizierung eines Testpunktes $\mathbf{x}$ wiefolgt modifiziert:

> Ist $\mathbf{x}$ gemäss $\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}$ in der i-ten Klasse, wird zu dieser Klasse eine Stimme von $|\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}|$ addiert, ansonsten erhält die j-te Klasse eine Stimme von $|\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}|$. Dieses Verfahren wird mit allen $K(K-1)/2$ binären SVM durchgeführt, und $\mathbf{x}$ wird als die Klasse mit der maximalen Stimme klassifiziert.

Es werden also die Entscheidungswerte $\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}$ je Klasse betragsmässig aufsummiert und $\mathbf{x}$ als diejenige Klasse mit der grössten Summe klassifiziert. Diese Wahlstrategie wird im Folgenden als SumDec bezeichnet.
Ein Nachteil der SumDec-Strategie ist jedoch, dass einzelne grosse Entscheidungswerte mehre kleine Entscheidungswerte überstimmen können. Deshalb wurde eine weiter modifizierte Wahlstrategie betrachtet:

> Ist $\mathbf{x}$ gemäss $\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}$ in der i-ten Klasse und $|\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}| \geq 1$, wird eine Stimme von 1 zu dieser Klasse addiert; falls $|\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}| < 1$ wird eine Stimme von $|\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}|$ zu dieser Klasse addiert. Entsprechend wird verfahren,

falls $\mathbf{x}$ gemäss $\mathbf{w}^{ij}\phi(\mathbf{x}) + b^{ij}$ in der j-ten Klasse ist. Dieses Verfahren wird mit allen $K(K-1)/2$ binären SVM durchgeführt, und $\mathbf{x}$ wird als die Klasse mit der maximalen Stimme klassifiziert.

Es wird also eine obere Grenze von 1 für eine einzelne Stimme festgelegt und ansonsten wie bei der SumDec-Strategie die Entscheidungswerte je Klasse betragsmässig aufsummiert. Diese Wahlstrategie wird im Folgenden als SumMax bezeichnet.

Für die Wahlstrategien SumDec und SumMax wurden die Ergebnisse mit 8-facher Kreuzvalidierung, einmal ohne Gewichtung und einmal mit der Gewichtung *prop* (siehe Abschnitt 4.3) berechnet, um zu ermitteln wie sich die modifizierten Wahlstrategien in Kombination mit der Gewichtung verhalten. Weiter gilt $C = 1$ und $\gamma = 1/10$.

4.4.1 SumDec-Strategie

Die Resultate der Klassifizierung mit der SumDec-Strategie sind in der Tabelle 15 zu sehen. Die über Klassen gemittelte Genauigkeit fällt mit 81.34 % etwas besser aus als für die

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	79.26	1.53	0.08	17.31	1.83
silence	2.83	94.35	0.40	0.66	1.76
stimmhaft	0.04	0.13	96.52	1.69	1.63
gemischt	11.38	2.68	10.58	67.90	7.46
irregulär	2.68	7.26	15.92	5.50	68.65

Tabelle 15: *Konfusionmatrix für die Wahlstrategie SumDec, Angaben in %*

MaxWins-Strategie (79.02 %). Wie in der Konfusionsmatrix zu sehen ist die Genauigkeit der einzelnen Klassen ausgeglichener. Während die Genauigkeit der Klassen **gemischt** und **irregulär** um etwa 10 % besser wird, zeigt sich für die Klassen **silence** und **stimmhaft** eine Verschlechterung von etwa 2 %.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	61.24	0.67	0.00	35.40	2.69
silence	2.77	89.49	0.00	2.60	5.15
stimmhaft	0.00	0.04	77.56	13.45	8.94
gemischt	2.03	0.25	0.58	86.74	10.40
irregulär	0.66	1.93	0.75	9.37	87.29

Tabelle 16: *Konfusionmatrix für die Wahlstrategie SumDec mit Klassengewichtung prop, Angaben in %*

Die Tabelle 16 zeigt die Resultate der Klassifizierung mit der SumDec-Strategie und der Gewichtung *prop*. Die über Klassen gemittelte Genauigkeit fällt mit 80.46 % schlechter aus als für die MaxWins-Strategie mit Gewichtung *prop* (86.23 %). Durch die veränderte

Wahlstrategie können die Klassengenauigkeiten etwas ausgeglichen werden. Allerdings lässt sich die Wahlstrategie nicht gut mit der Verwendung gewichteter SVM kombinieren, da die Verschlechterung der weniger gewichteten Klassen zu stark ausfällt.

4.4.2 SumMax-Strategie

Die Resultate der Klassifizierung mit der SumMax-Strategie sind in der Tabelle 17 zu sehen, wobei die über Klassen gemittelte Genauigkeit 79.28 % beträgt.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	91.95	2.99	0.11	3.37	1.58
silence	1.78	96.50	0.45	0.20	1.08
stimmhaft	0.04	0.16	98.22	0.58	1.00
gemischt	23.12	3.33	17.17	50.33	6.05
irregulär	3.56	9.06	23.66	4.31	59.41

Tabelle 17: *Konfusionsmatrix für die Wahlstrategie SumMax, Angaben in %*

Die erzielten Resultate liegen im gleichen Bereich wie jene der MaxWins-Strategie. Durch die Strategie SumMax lassen sich also keine Verbesserungen gegenüber der MaxWins-Strategie erzielen.

Die Tabelle 18 zeigt die Resultate der Klassifizierung mit der SumMax-Strategie und der Gewichtung *prop*.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	86.44	1.51	0.00	9.44	2.60
silence	2.25	92.36	0.00	0.90	4.48
stimmhaft	0.01	0.05	90.83	3.66	5.45
gemischt	9.89	0.87	3.15	75.36	10.72
irregulär	1.14	2.77	2.99	6.64	86.46

Tabelle 18: *Konfusionsmatrix der Wahlstrategie SumMax mit Klassengewichtung prop, Angaben in %*

Auch diese Resultate liegen im gleichen Bereich wie jene der MaxWins-Strategie mit der Gewichtung *prop*.

Wie die Resultate zeigen, lassen sich mit den Wahlstrategien SumDec und SumMax, die die Entscheidungswerte der einzelnen binären SVM berücksichtigen, keine besseren Ergebnisse erzielen.

4.5 Klassifizierung mit ausgeglichenen Trainingsdaten

In diesem Abschnitt wurde das Training und Testen mit Datensätzen durchgeführt, die nach Klassen ausgeglichen sind. Die Trainings- und Testmengen enthalten also jeweils von jeder Klasse gleichviele Instanzen. Sei m_i die Anzahl der Daten der Klasse i in einem bestimmten Datensatz. Es werden nun zufällig von jeder Klasse m_{min} Daten ausgewählt, wobei

$$m_{min} = \min_{i=1,\ldots,K} m_i$$

und K die Anzahl Klassen ist. Da die Daten stark unausgeglichen sind (siehe Tabelle 9) bedeutet dies, dass auf einen wesentlichen Teil der Trainingsdaten verzichtet wird.
Da sich die Gewichtungen, wie sie im Abschnitt 4.3 untersucht wurden, nach der Verteilung der Klassen in den Trainingsdaten richteten, wird die Verwendung gewichteter SVM in diesem Abschnitt nicht weiter untersucht. Für die drei Wahlstrategien MaxWins, SumDec und SumMax wurden die Klassifizierungsgenauigkeiten mit Kreuzvaliderung wie in den vorhergehenden Abschnitten ermittelt, und die Parameter C und γ haben die festen Werte $C = 1$ und $\gamma = 1/10$.

4.5.1 MaxWins-Strategie

Tabelle 19 zeigt die Resultate für die möglichen Werte von n. Gegenüber den Resultaten mit unausgeglichenen Trainingsdaten (siehe Tabelle 2) sind die Genauigkeiten etwa 5 % höher. Weiter zeigt sich auch, wie schon in Abschnitt 4.2, dass die Resultate keine allzu starke Abhängigkeit von n zeigen.

n	Genauigkeit in %
2	84.99
4	84.83
8	85.04
16	84.90

Tabelle 19: *n-fache Kreuzvalidierungsgenauigkeit*

Die Resultate für $n = 8$ sind in Tabelle 20 angegeben. Es zeigt sich, dass die Klassifizierung der Klassen **gemischt** und **irregulär** deutlich besser ausfällt als beim Training mit unausgeglichenen Daten. Mit dem Ausgleichen der Trainingsdaten wird die a priori Wahrscheinlichkeit zu gunsten der Klassen mit mehr Trainingsdaten aufgehoben. Allerdings werden die Klassen **gemischt** und **irregulär** immer noch am schlechtesten klassifiziert.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	84.70	1.85	0.00	9.46	3.99
silence	2.13	92.44	0.00	0.70	4.73
stimmhaft	0.00	0.00	90.82	3.62	5.56
gemischt	10.94	1.58	3.38	70.42	13.68
irregulär	0.97	2.74	3.06	5.84	87.39

Tabelle 20: *Konfusionsmatrix,*$n = 8$*, Angaben in %*

4.5.2 SumDec-Strategie

Die Tabellen 21 und 22 zeigen die Resultate der SumDec-Strategie. Die Ergebnisse fallen deutlich schlechter aus als für die MaxWins-Strategie. Auffällig ist, dass die Klassen **stimmhaft** und **stimmlos** am schlechtesten klassifiziert werden. Offenbar zeigt die SumDec-Strategie unerwünschte Effekte und liefert keine brauchbaren Ergebnisse.

n	Genauigkeit in %
2	69.60
4	71.15
8	70.97
16	71.07

Tabelle 21: *n-fache Kreuzvalidierungsgenauigkeit*

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	39.50	0.65	0.00	54.84	5.01
silence	1.95	85.67	0.00	3.94	8.44
stimmhaft	0.00	0.00	56.00	30.09	13.91
gemischt	0.74	0.09	0.00	86.46	12.70
irregulär	0.23	0.74	0.00	10.89	88.13

Tabelle 22: *Konfusionsmatrix,* $n = 8$*, Angaben in %*

4.5.3 SumMax-Strategie

Die Tabellen 23 und 24 zeigen die Resultate der SumMax-Strategie.

n	Genauigkeit in %
2	83.95
4	85.16
8	85.15
16	85.34

Tabelle 23: *n-fache Kreuzvalidierungsgenauigkeit*

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	87.11	1.16	0.00	8.76	2.97
silence	2.60	90.03	0.00	1.25	6.12
stimmhaft	0.00	0.00	90.17	3.43	6.40
gemischt	10.20	1.02	3.76	72.04	12.98
irregulär	0.83	2.36	3.25	6.72	86.83

Tabelle 24: *Komfusionsmatrix,* $n = 8$*, Angaben in %*

Wie zu sehen ist, sind die Reultate im gleichen Bereich wie für die MaxWins-Strategie, und es lassen sich mit der SumMax-Strategie keine wesentlichen Verbesserungen erreichen.

4.6 Parameterselektion

Für die Parameterselektion, also die Wahl des Parameterpaars (C, γ), wurde ein in LIBSVM [2] enthaltens Tool verwendet. Für die n-fache Kreuzvalidierung ist dabei zu beachten, dass die Daten in n gleich grosse Teilmengen aufgeteilt und nicht wie in den vorhergehenden Abschnitten nach Stimmen aufgeteilt werden. Weiter ist die für ein Parameterpaar (C, γ) berechnete Genauigkeit die durch Kreuzvalidierung mit unausgeglichenen Daten ermittelte Genauigkeit. Dieses Vorgehen ist nicht ideal, und aus Zeitgründen wurde auf eine Parameterselektion mit gewichteten SVM und ausgeglichenen Datensätzen verzichtet.
Da die Suche sehr zeitintensiv ist, wurden von allen Frames der 16 Stimmen 10000 ausgewählt und mit diesen die Parametersuche durchgeführt. Die Auswahl erfolgte dabei mit einem in LIBSVM [2] enthaltenen Tool, das die Daten der Teilmenge so wählt, dass die Verteilung der Klassen erhalten bleibt.
Die hier durchgeführte Parameterselektion für die drei Wahlstrategien MaxWins, SumDec und SumMax liefert also nur eine grobe Abschätzung über die mit der Parameterselektion erreichbare Verbesserung.

4.6.1 MaxWins-Strategie

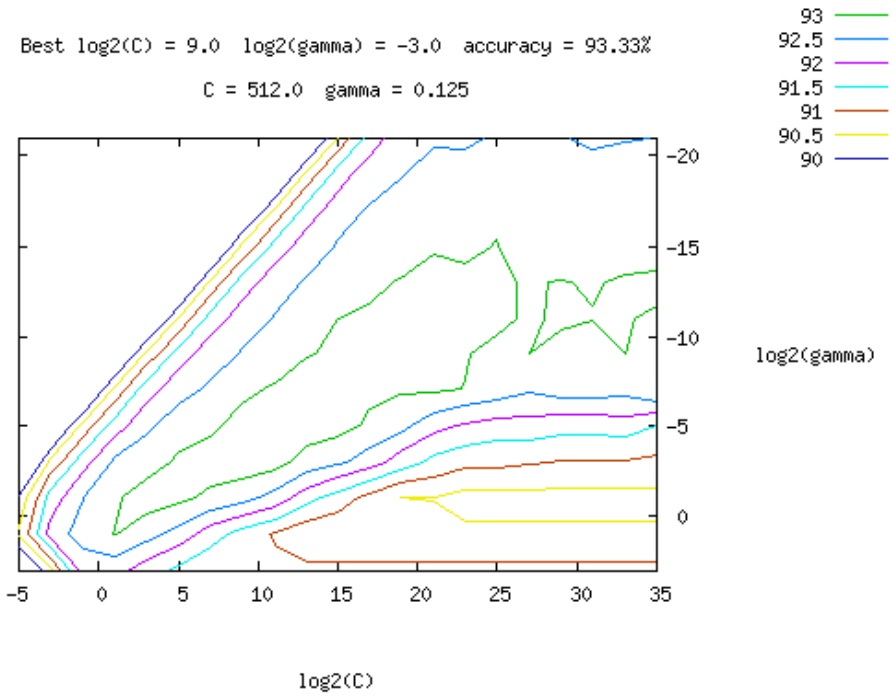
Für das Parameterpaar (C, γ) wurde für exponentiell steigende Werte im Bereich $C = 2^{-5}, 2^{-3}, \ldots, 2^{35}$ und $\gamma = 2^{-21}, 2^{-19}, \ldots, 2^{3}$ die Parametersuche auf einer Teilmenge von 10000 Datenpunkten mit 5-facher Kreuzvalidierung durchgeführt. Figur 2 zeigt einen Konturplot für die Parametersuche.

Wie in Figur 2 zu sehen ist, wird für $C = 512$ und $\gamma = 0.125$ eine Genauigkeit von 93.33 % erreicht. Wird nun mit diesen gefundenen Parametern ein SVM-Modell mit den Daten aller 16 Stimmen trainiert und anschliessend mit den Daten der übrigen vier Stimmen getestet, ergibt sich das in Tabelle 25 dargestellte Resultat.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	90.00	0.81	0.14	7.24	1.81
silence	1.28	96.51	0.59	0.59	1.03
stimmhaft	0.01	0.05	98.98	0.50	0.46
gemischt	4.98	0.75	20.21	69.98	4.07
irregulär	6.05	3.43	23.59	6.25	60.69

Tabelle 25: *Konfusionsmatrix,* $C = 512$, $\gamma = 0.125$, *Angaben in %*

Die über Klassen gemittelte Genauigkeit ergibt 83.23 %. Im Vergleich dazu ist die Genauigkeit mit $C = 1$ und $\gamma = 1/10$ und denselben Testdaten 80.75 %. Mit der Parameterselektion lässt sich die Genauigkeit also um knapp 2.5 % verbessern.



Figur 2: *Parametersuche mit 10000 Datenpunkten*

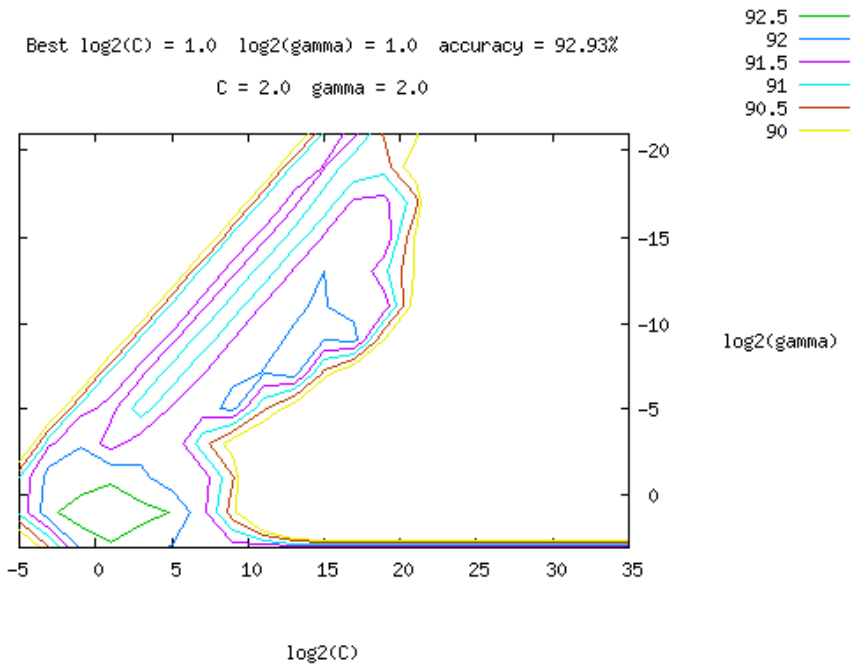
4.6.2 SumDec-Strategie

Unter Verwendung der SumDec-Strategie wurde für das Parameterpaar (C, γ) ebenfalls im Bereich $C = 2^{-5}, 2^{-3}, \ldots, 2^{35}$ und $\gamma = 2^{-21}, 2^{-19}, \ldots, 2^{3}$ die Parametersuche auf einer Teilmenge von 10000 Datenpunkten mit 5-facher Kreuzvalidierung durchgeführt. Figur 3 zeigt den Konturplot für die Parametersuche.

Es zeigt sich, dass bei der SumDec-Strategie die gefundenen Parameter mit $C = 2$ und $\gamma = 2$ in einem anderen Bereich liegen als bei MaxWins.
Tabelle 26 zeigt die Klassifizierungsergebnisse des Testes mit den Daten der letzten vier Stimmen. Die über Klassen gemittelte Genauigkeit ist 83.23 %. Mit $C = 1$ und $\gamma = 1/10$ ergibt die Genauigkeit 81.36 %.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	88.56	0.72	0.09	9.00	1.63
silence	1.68	95.60	0.59	0.72	1.40
stimmhaft	0.02	0.05	98.48	0.88	0.57
gemischt	5.73	0.60	16.89	72.25	4.52
irregulär	5.44	2.62	21.77	5.65	64.52

Tabelle 26: *Konfusionsmatrix, $C = 2$, $\gamma = 2$, Angaben in %*



Figur 3: *Parametersuche mit 10'000 Datenpunkten*

4.6.3 SumMax-Strategie

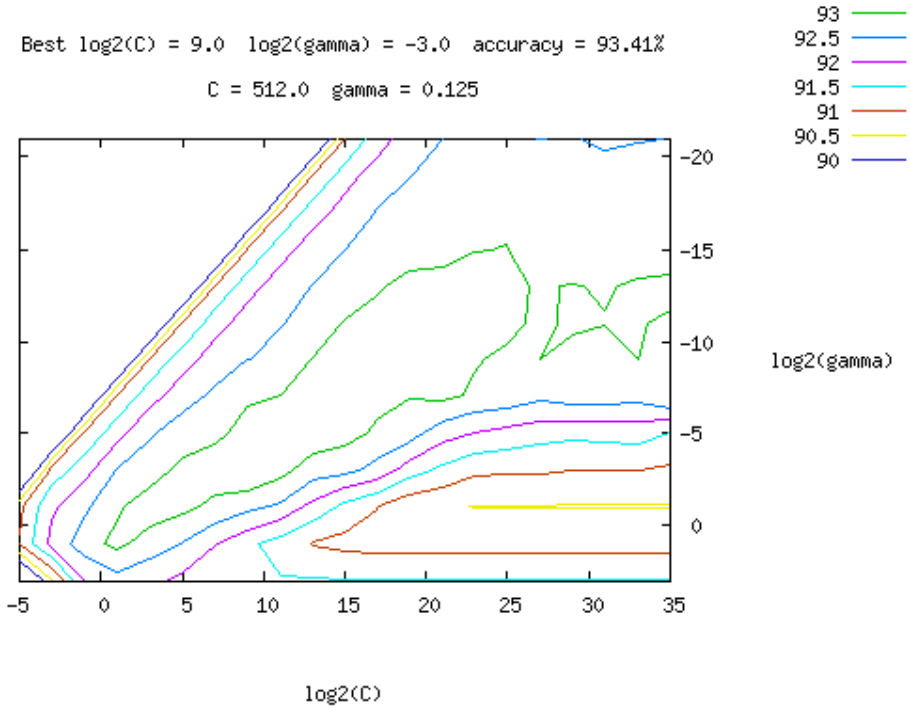
Den Konturplot der Parametersuche unter Verwendung der SumMax-Strategie zeigt Figur 4. Der Suchbereich für das Parameterpaar (C, γ) ist wiederum $C = 2^{-5}, 2^{-3}, \ldots, 2^{35}$ und $\gamma = 2^{-21}, 2^{-19}, \ldots, 2^{3}$.

Die Parametersuche liefert für die Wahlstrategie SumMax die gleichen Werte wie für MaxWins, also $C = 512$ und $\gamma = 0.125$.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	90.28	0.72	0.18	7.15	1.67
silence	1.40	96.26	0.62	0.59	1.12
stimmhaft	0.02	0.05	98.93	0.55	0.45
gemischt	5.13	0.60	19.91	70.29	4.07
irregulär	6.85	3.43	23.39	6.65	59.68

Tabelle 27: *Konfusionsmatrix,* $C = 512$, $\gamma = 0.125$, *Angaben in %*

Tabelle 27 zeigt die Klassifizierungsergebnisse des Testes mit den Daten der letzten vier Stimmen. Die über Klassen gemittelte Genauigkeit ist 83.09 %. Im Vergleich dazu ergibt sich mit $C = 1$ und $\gamma = 1/10$ eine Genauigkeit von 80.87 %.



Figur 4: *Parametersuche mit 10000 Datenpunkten*

5 Vergleich zu neuronalem Netz

In diesem Abschnitt wird die Frame-Klassifizierung, wie sie in [3] mit einem neuronalen Netz durchgeführt wurde, mit der SVM-Klassifizierung dieser Arbeit verglichen. Nebst einem kurzen Vergleich der mit dem jeweiligen Klassifikator erreichten Ergebnisse werden die Probleme und Vorteile der SVM-Klassifizierung, wie sie sich in dieser Arbeit gezeigt haben, im Gegensatz zur Klassifizierung mit einem neuronalen Netz erwähnt.

	stimmlos	*silence*	*stimmhaft*	*gemischt*	*irregulär*
stimmlos	87.90	1.99	0.00	7.42	2.69
silence	2.50	92.72	0.00	0.51	4.27
stimmhaft	0.00	0.05	90.73	3.66	5.56
gemischt	9.27	0.97	3.38	76.12	10.25
irregulär	0.97	2.69	3.20	4.96	88.18

Tabelle 28: *Klassifizierungsgenauigkeit in % des neuronalen Netzes*

Wie bei der SVM wurden die Daten von 16 Stimmen verwendet, die in 8 Teilmengen zu jeweils zwei Stimmen aufgeteilt und für die Kreuzvalidierung verwendet wurden. Tabelle 28 zeigt die Klassifizierungergebnisse für das neuronale Netz. Die über Klassen gemittelte Genauigkeit beträgt 87.13 %. Demgegenüber beträgt die Genauigkeit, die mit ausgeglichenen Trainingsdaten und der MaxWins- oder SumMax-Strategie erreicht wurde, 85.04 % bzw. 85.15 %. Die höchste Genauigkeit von 86.23 % mit SVM-Klassifizierung wurde durch unterschiedliche Gewichtung der Trainingsfehler nach Klasse erreicht, wobei mit unausgeglichenen Daten trainiert wurde.
Bezüglich der Klassifizierungsgenauigkeiten der einzelnen Klassen zeigen die beiden Klassifizierungen mit der Gewichtung *prop* und mit ausgeglichenen Trainingsdaten bei Verwendung der MaxWins-Strategie (siehe Tabelle 12 und 20) nahezu gleiches Verhalten wie das neuronale Netz.
Das neuronale Netz wird mit jeweils mit den Daten einer Klasse trainiert und kann somit jeweils die Klassifizierung einer einzelnen Klasse lernen. Da dabei ausgeglichene Traininsdaten verwendet werden, lassen sich die Ergebnisse der SVM mit unausgeglichenen Daten nur bedingt mit denen des neuronalen Netzes vergleichen. Auch lernt die SVM nicht die Klassifizierung einzelner Klassen, sondern nur binäre Entscheidungen zwischen zwei Klassen. Bei der Multiclass-Klassifizierung stellt sich dann das Problem, wie diese effizient umgesetzt werden kann.
Ein Vorteil der SVM liegt in der Möglichkeit, die Trainingsfehler je nach Klasse verschieden zu gewichten. Dadurch können auch brauchbare Resultate mit unausgeglichenen Trainingsdaten erreicht werden und einzelne Klassengenauigkeiten beeinflusst werden.

6 Schlussfolgerungen und Ausblick

Das Ziel dieser Semesterarbeit war es, Sprachsignalframes unter Verwendung von Support-Vektor-Maschinen als Klassifikator in fünf vorgegebene Klassen einzuteilen.
Dazu wurde für die Multiclass-Klassifizierung der one-against-one Ansatz mit der Wahlstrategie MaxWins verwendet. Wie sich zeigte, kann mit dieser Wahlstrategie keine zufriedenstellende Klassifizierung aller Klassen erreicht werden. Die Probleme liegen einerseits in der schlechten Separierbarkeit gewisser Klassen und andererseits in der unausgeglichenen Verteilung der Klassen in den Trainingsdaten.
Um die Klassifizierungsergebnisse zu verbessern, wurden zwei weitere modifizierte Wahlstrategien vorgeschlagen und getestet, die die Entscheidungswerte der einzelnen binären SVM berücksichtigen. Mit diesen konnte keine Verbesserung der Resultate erreicht werden.
Um das Problem der ungleichen Verteilung der Trainingsdaten zu lösen, wurden zum einen gewichtete SVM verwendet, die die Trainingsfehler je Klasse verschieden gewichten, zum anderen wurden die Trainingsdaten ausgeglichen.
Die besten Ergebnisse wurden mit gewichteten SVM bei unausgeglichenen Trainingsdaten und durch Ausgleichen der Daten erreicht, wobei man durch Ausgleichen der Trainingsdaten etwa gleich gute Resultate erreichen kann wie mit der Verwendung von gewichteten SVM.

Im Vergleich zu einem neuronalen Netz konnten in dieser Arbeit mit SVM etwa gleich gute Resultate erreicht werden. Da die Klassifizierung mit SVM noch weitere Möglichkeiten bietet, die in dieser Arbeit nicht weiter untersucht werden konnten, lässt sich keine Aussage machen, ob sich SVM oder neuronale Netze besser für die gestellte Klassifizierungsaufgabe eignen.

Wie erwähnt ist nicht klar, wie SVM effizient auf Probleme mit mehr als zwei Klassen angewendet werden können. Nebst dem in dieser Arbeit verwendeten Ansatz sind auch Multiclass-Klassifizierungen mit einem Entscheidungsbaum denkbar, oder es können SVM-Formulierungen für mehrere Klassen betrachtet werden, die nicht auf binären SVM beruhen (siehe [5]).
Für die Verwendung gewichteter SVM wurden lediglich drei Möglichkeiten untersucht, und es ist nicht klar, wie die Gewichte optimal zu wählen sind.
Die Parametersuche, wie sie in dieser Arbeit durchgeführt wurde, lässt keine abschliessende Aussage zu, inwieweit dadurch eine Verbesserung der Klassifizierung erreicht werden kann. Um verlässlichere Resultate zu erhalten, müsste die Parametersuche auch mit gewichteten SVM und ausgeglichenen Daten durchgeführt werden.

Literatur

[1] Christopher J.M. Burges. *A tutorial on support vector machines for pattern recognition*, 1998.

[2] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] T. Ewender, S. Hoffmann, and B. Pfister. Nearly perfect detection of continuous F0 contour and frame classification for TTS synthesis. In *Proceedings of Interspeech*, pages 100-103, Brighton, September 2009.

[4] B. Pfister and T. Kaufmann. *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer Verlag (ISBN: 978-3-540-75909-6), 2008. `http://www.springer.com/978-3-540-75909-6`.

[5] C.-W. Hsu and C.-J. Lin. *A comparision of methods for multi-class support vector machines. IEEE Transaction on Neural Networks*, 13(2):415-425, 2002.

[6] Hsu, Chih-Wei and Chang, Chih-Chung and Lin, Chih-Jen. *A Practical Guide to Support Vector Classification. Department of Computer Science*.

[7] Eric Moulines and Francis Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9(5-6): 453-467, December 1990. Journal article to PSOLA paper.

[8] Changchun Bao Fengyan Qi and Yan Liu. A novel two-step SVM classifier for voiced/ unvoiced/silence classification of speech. In *2004 International Symposium on Chinese Spoken Language Processing*, 2004. Rather low-quality paper about applying SVMs to voicing decisions.

[9] J.M. Gorriz C.G. Puntonet P. Yelamos, J. Ramirez and J.C. Segura. *Lecture Notes in Computer Science*, volume 3991/2006, chapter Speech Event Detection Using Support Vector Machines, pages 356-363. Springer Berlin / Heidelberg, 2006.

Anhang A

Vom Institut abgegebene Aufgabenstellung

Institut für
Technische Informatik und
Kommunikationsnetze

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo

Herbstsemester 2009
(SA-2009-23)

Aufgabenstellung

für

Herrn Daniel Baumann

Betreuer: T. Ewender, ETZ D97.7
S. Hoffmann, ETZ D97.5

Ausgabe: 23. September 2009
Abgabe: 18. Dezember 2009

Frame classification of speech using support vector machines

Einleitung

Die meisten Sprachsynthesesysteme basieren heute auf der Verkettung natürlicher Sprachabschnitte. Um gute Synthesequalität zu erreichen, werden neben einer genauen und robusten Schätzung der Grundfrequenzkontour detaillierte Informationen über die Signaleigenschaften benötigt.

Diese Information wird für die prosodische Veränderung von Signalen benötigt. Diese prosodische Veränderung kann beispielsweise mit *time domain PSOLA* geschehen. Die Art und Weise, wie die prosodische Veränderung im einzelnen durchgeführt wird, hängt von verschiedenen Signaleigenschaften ab, wie der Stimmhaftigkeit des Signals oder dem Vorhandensein von Rauschanteilen. Im Detail sind hierbei fünf Klassen sinnvoll, die wie folgt definiert sind:

stimmhaft: Klar stimmhafte Sprache, harmonisches Signal, keine Rauschen, niedrige Frequenzen dominierend (typische Phoneme: Vokale, Nasale)

stimmlos: Stimmhaftigkeit nicht erkennbar, rauschartig, hohe Frequenzen über 2 kHz sind dominierend (typische Phoneme: Frikative, stimmlose Plosive)

gemischt: Sprache mit stimmhaften Anteilen und Rauschanteilen; nur die niedrigsten Harmonischen im Spektrum erkennbar, höhere spektralen Komponenten rauschhaft (typische Phoneme: stimmhafte Frikative, häufig auch in Stimmhaft/Stimmlos-übergängen)

unregelmässig: Sprache mit unregelmässigen Glottisschlägen, keine bedeutenden Frikativanteile, niedrige Frequenzen dominierend; auch bekannt als Knarrstimme.

silence: Signalabschnitte mit sehr niedriger Energie, nicht hörbar;

Support Vector Machines (SVMs) sind eine Methode zur Mustererkennung und Klassifizierung aus dem Bereich Machine Learning. Support Vector Machines wurden in Bereichen wie handschriftliche Ziffern-Erkennung, Spracherkennung oder Sprecherverifikation eingesetzt. Seit kurzem werden Support Vector Machines auch im Bereich Sprachklassifikation wie etwa zur Stimmhaft/Stimmlos-Unterscheidung (siehe [1]) oder Sprache/Nicht-Sprache-Unterscheidung eingesetzt (siehe [2]).

Problemstellung

Das Ziel dieser Semesterarbeit ist die Anwendung von Support Vector Machines als Klassifikator, der eine Unterscheidung von Signalframes bezüglich der oben erwähnten fünf Klassen vornimmt. Als bisheriger Klassifikator wurde ein neuronales Netz verwendet. Aufgabe ist es nun, diesen durch eine Support Vector Machine zu ersetzen.

Vorgehen

Zum Thema Support Vector Machines ist einschlägige Literatur vorhanden (siehe z.B. [3]). Es wird ein Vorgehen wie folgt empfohlen:

1. Lesen Sie sich in die vorhandene Literatur zum Thema Support Vector Machines ein.
2. Schaffen Sie sich einen überblick über die vorhandenen Implementationen von Support Vector Machines und wählen Sie eine geeignete Implementation aus.
3. Überlegen Sie welche Konfiguration der SVM für die Aufgabe geeignet ist und welche Form die Merkmale haben müssen.
4. Für einen ersten Versuch können Sie die Merkmale verwenden, die für das neuronale Netz verwendet wurden, eventuell ist eine Anpassung dieser Merkmale notwendig.
5. Experimentieren Sie mit verschiedenen Konfiguration der SVM, bis zufriedenstellende Resultate erzielt werden.
6. Vergleichen Sie die Resultate mit denen des neuronalen Netzes und versuchen Sie, diese zu interpretieren.

Die ausgeführten Arbeiten und die erhaltenen Resultate sind in einem Bericht zu dokumentieren (siehe dazu [4]), der in gedruckter und in elektronischer Form (als PDF-Datei) abzugeben ist. Zusätzlich sind im Rahmen eines Kolloquiums zwei Präsentationen vorgesehen: etwa drei Wochen nach Beginn soll der Arbeitsplan und am Ende der Arbeit die Resultate vorgestellt werden. Die Termine werden später bekannt gegeben.

Optional

Eine mögliche Erweiterung der Aufgabe wäre, eine SVM zur Erkennung von Atemgeräuschen bei Sprachaufnahmen einzusetzen. Dazu müssten geeignete Merkmale ausgewählt und angepasst oder implementiert werden. Zusätzlich ist dazu die Aufbereitung von Trainings-Daten notwendig.

Literaturverzeichnis

[1] Changchun Bao Fengyan Qi and Yan Liu. A novel two-step SVM classifier for voiced/unvoiced/silence classification of speech. In *2004 International Symposium on Chinese Spoken Language Processing*, 2004. Rather low-quality paper about applying SVMs to voicing decisions.

[2] J.M. Górriz C.G. Puntonet P. Yélamos, J. Ramírez and J.C. Segura. *Lecture Notes in Computer Science*, volume 3991/2006, chapter Speech Event Detection Using Support Vector Machines, pages 356–363. Springer Berlin / Heidelberg, 2006.

[3] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition, 1998.

[4] B. Pfister. *Richtlinien für das Verfassen des Berichtes zu einer Semester- oder Diplomarbeit*. Institut TIK, ETH Zürich, März 2004.
(http://www.tik.ee.ethz.ch/~spr/SADA/richtlinien_bericht.pdf).

[5] B. Pfister. *Hinweise für die Präsentation der Semester- oder Diplomarbeit*. Institut TIK, ETH Zürich, März 2004.
(http://www.tik.ee.ethz.ch/~spr/SADA/hinweise_praesentation.pdf).

[6] B. Pfister and T. Kaufmann. *Sprachverarbeitung: Grundlagen und Methoden der Sprachsynthese und Spracherkennung*. Springer Verlag (ISBN: 978-3-540-75909-6), 2008. http://www.springer.com/978-3-540-75909-6.

Zürich, den 23. September 2009