Statistische Eigenschaften lokalisierter maschineller Lernverfahren

Von der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

von

Florian Dumpert

aus Bayreuth

1. Gutachter: Prof. Dr. Andreas Christmann
2. Gutachter: Prof. Dr. Ingo Steinwart

Tag der Einreichung: 02.09.2019
Tag des Kolloquiums: 24.01.2020

Zusammenfassung

Neben anderen Methoden des maschinellen Lernens spielen Support Vector Machines (SVMs) heute in vielen Wissenschaftsbereichen eine wichtige Rolle. In den letzten zwei Jahrzehnten wurde beträchtlich im Bereich statistischer Eigenschaften und der Berechenbarkeit von Support Vector Machines und verwandten kernbasierten Methoden geforscht. Auf der einen Seite ist man aus statistischer Sicht an der Konsistenz und Robustheit der Methode interessiert. Auf der anderen Seite, aus Sicht der Berechenbarkeit, ist man an einer Methode interessiert, die mit vielen Beobachtungen und vielen erklärenden Variablen umgehen kann. Da SVMs viel Rechenleistung und Speicherkapazität benötigen, wurden verschiedene Möglichkeiten zur Handhabung großer Datensätze vorgeschlagen. Eine davon, die als Regionalisierung bezeichnet wird, teilt den Raum der erklärenden Variablen datengesteuert in möglicherweise überlappende Bereiche auf und definiert den Prädiktor durch das Zusammenspiel lokal erlernter Support Vector Machines. Diese Arbeit zeigt, dass ein so erlernter Prädiktor Konsistenz und Robustheitseigenschaften unter Annahmen bewahrt, die vom Anwender dieser Methode geprüft werden können.

Abstract

Among different machine learning methods, support vector machines (SVMs) play an important role in many fields of science nowadays. A lot of research about statistical and computational properties of support vector machines and related kernel methods has been done during the last two decades up to now. On the one hand, from a statistical point of view, one is interested in consistency and robustness of the method. On the other hand, from a computational point of view, one is interested in a method that can deal with many observations and many features. As SVMs need a lot of computing power and storage capacity, different ways to handle big data sets were proposed. One of them, which is called regionalization, divides the space of the declaring variables into possibly overlapping regions in a data driven way and defines the output predicting function by composing locally learnt support vector machines. This thesis shows that a predictor learnt in this way conserves consistency and robustness results under assumptions that can be checked by the user of this method.

Inhalt

Zusammenfassung ii

Abstract iii

Symbolverzeichnis vi

Abbildungsverzeichnis viii

Tabellenverzeichnis ix

1 Einordnung 1

1.1 Maschinelles Lernen im Allgemeinen 1

1.2 Abgrenzung 6

1.3 Bayesianische Statistik 7

1.4 Support Vector Machines im Speziellen 8

1.5 Wünschenswerte Eigenschaften 16

2 Große Datenmengen und lokales Lernen 19

2.1 Problembeschreibung 19

2.2 Zerlegung des Datenraumes mittels eines Baumes für SVMs auf großen Datensätzen 21

2.3 Lokales Lernen 24

3 Konkretisierung der Regionalisierung 27

4 Statistische Eigenschaften 32

4.1 Konsistenz 32

4.2 Beweis der Konsistenz 34
4.3 Robustheit im Sinne des *maxbias* 43
4.4 Robustheit im Sinne der Influenzfunktion 45
4.5 Vergleich der Robustheitsbegriffe 51

5 Testrechnungen **54**

5.1 Simulationsbeispiel zur Klassifikation 54
5.2 Simulationsbeispiel zur Regression 57
5.3 Simulationsbeispiel zur Regression in höheren Dimensionen 61
5.4 Bayern 65
5.5 Klassifikation anhand des SUSY-Datensatzes 65

6 Zusammenfassung und Ausblick **69**

A Zum Einsatz geshifteter Verlustfunktionen **71**

Quellenverzeichnis **73**

Symbolverzeichnis

$\lvert M \rvert$	Anzahl der Elemente einer Menge M
Φ	eine feature map
$\mathfrak{B}_M$	Borel-σ-Algebra auf einer Menge M
δ_z	Dirac-Maß im Punkt z
$\mathcal{D}_n$	n unabhängige und identisch verteilte Beobachtungen von (X, Y)
D_n	empirische Verteilung basierend auf $\mathcal{D}_n$
$D_{n,b}$	renormierte empirische Verteilung in $\mathcal{X}_b \times \mathcal{Y}$ basierend auf $\mathcal{D}_n$
$f_{L,P,\lambda}$	Minimierer von $\mathcal{R}_{\mathcal{X},L,P,\lambda}(f)$
$f^{comp}_{L,P,\lambda}$	zusammengesetzter Prädiktor
L	eine Verlustfunktion, falls geshiftet mit L^* bezeichnet
n_b	Anzahl der Trainingsdatenpunkte, die in der Region $\mathcal{X}_b \times \mathcal{Y}$ liegen
$\mathcal{O}$	Landau-Symbol
P	die (X, Y) zugrundeliegende Verteilung auf $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$
$P_{\lvert \mathcal{X}_b \times \mathcal{Y}}$	die (X, Y) zugrundeliegende Verteilung P eingeschränkt auf $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b\times\mathcal{Y}})$
P_b	die renormierte (X, Y) zugrundeliegende Verteilung auf $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b\times\mathcal{Y}})$
$P^{\mathcal{X}}$	die Randverteilung von X auf $(\mathcal{X}, \mathfrak{B}_{\mathcal{X}})$
$P_b{}^{\mathcal{X}_b}$	die renormierte Randverteilung von X auf $(\mathcal{X}_b, \mathfrak{B}_{\mathcal{X}_b})$

$\mathcal{R}_{\mathcal{X},L,P}(f)$	Risiko über $\mathcal{X}$ eines Prädiktors f bezüglich einer Verlustfunktion L und einer Verteilung P
$\mathcal{R}_{\mathcal{X},L,P,\lambda}(f)$	regularisiertes Risiko über $\mathcal{X}$ eines Prädiktors f bezüglich einer Verlustfunktion L und einer Verteilung P
$\mathcal{R}^*_{\mathcal{X},L,P}$	Bayes-Risiko über $\mathcal{X}$ (und über alle messbaren Funktionen) bezüglich einer Verlustfunktion L und einer Verteilung P
$\mathcal{R}^*_{\mathcal{X},L,P,\mathcal{F}}$	Bayes-Risiko über $\mathcal{X}$ und über einer Funktionenklasse $\mathcal{F}$ bezüglich einer Verlustfunktion L und einer Verteilung P
w_b	Gewichtsfunktion, indiziert mit b
$\mathcal{X}$	Eingaberaum, mindestens als separabler metrischer Raum vorausgesetzt
$\mathcal{X}_b \times \mathcal{Y}$	Region mit Index b
$\mathcal{X}_I \times \mathcal{Y}$	„reiner“ Schnitt von Regionen
$\mathcal{Y}$	Ausgaberaum, stets als abgeschlossene Teilmenge der reellen Zahlen vorausgesetzt

Abbildungsverzeichnis

1.1 Vergleich von Polynominterpolation und linearer Regression 4

3.1 Gewichtsfunktionen 30

4.1 Illustration zur Robustheit 52

5.1 Wahre Verteilung der beiden Klassen (rot und blau) 55
5.2 Zusammenfassung der Resultate für 750 Trainingspunkte 56
5.3 Zusammenfassung der Resultate für 10000 Trainingspunkte 56
5.4 Zusammenfassung der Resultate für 50000 Trainingspunkte 57
5.5 Trainingsdaten und wahrer Zusammenhang 58
5.6 Testdaten, wahrer Zusammenhang und globale SVM 59
5.7 Testdaten, wahrer Zusammenhang und zusammengesetzter Prädiktor auf Basis lokaler SVMs ($n_{train} = 600$) 60
5.8 Testdaten, wahrer Zusammenhang und Prädiktoren ($n_{train} = 4800$) . 60
5.9 Testdaten, wahrer Zusammenhang und Prädiktoren ($n_{train} = 6000$) . 61
5.10 RMSE im Vergleich 63
5.11 Laufzeiten im Vergleich für 5000 Datenpunkte 63
5.12 Laufzeiten im Vergleich für 25000 Datenpunkte 64
5.13 Laufzeiten im Vergleich zur Größe des Trainingsdatensatzes 64
5.14 Betrachtungen der Regionen 1 65
5.15 Betrachtungen der Regionen 2 66
5.16 Laufzeiten im Vergleich 67
5.17 Genauigkeit (Accuracy – Acc) im Vergleich 68

Tabellenverzeichnis

1.1 Eigenschaften von supervised-Verlustfunktionen 11

Kapitel 1

Einordnung

1.1 Maschinelles Lernen im Allgemeinen

Der Versuch einer exakten Fassung des Begriffs *maschinelles Lernen*[1] ist nicht Gegenstand dieser Arbeit. Dennoch sollen ein paar Aspekte benannt werden, die im Kontext dieses Begriffs immer wieder zutage treten. Der Begriff des Lernens an sich ist bereits nicht einheitlich definiert, wenngleich Valiant (1984) als eine Art Standardreferenz der theoretischen Auseinandersetzung mit dieser Frage gesehen werden kann. Der Begriff des *Probably Approximately Correct (PAC) Learnings*, der das statistische Konzept der Konsistenz nutzt, wird dort erstmals erwähnt.[2] Simon (1983) schreibt zum Thema *Lernen* in seinem Aufsatz *Why should machines learn?*:

> Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time.

Häufig wird Samuel (1959) zur Charakterisierung maschinellen Lernens herangezogen:

> The studies reported here have been concerned with the programming of a digital computer to behave in a way which, if done by human beings or animals, would be described as involving the process of learning. [...] Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort.

[1] Der Ausdruck *maschinelles Lernen* wird für diese Arbeit stets im Sinne des statistischen maschinellen Lernens verstanden.

[2] Dabei wird die Idee/Heuristik genutzt, dass – sofern sich die zugrundeliegende Verteilung zwischen Training, Testen und späterer Anwendung nicht ändert – vollkommen falsche Zusammenhänge schnell erkannt werden; solche jedoch, die nicht frühzeitig zu schlechten Ergebnissen führen, können auch nicht vollkommen unzutreffend sein.

Maschinelles Lernen liegt demnach dann vor, wenn ein Computer basierend auf Erfahrung *lernt*, eine Aufgabe auszuführen, ohne dass die Lösungsstrategie oder mögliche Lösungswege explizit (hart codiert) vorgegeben werden. In vielen Fällen wäre das explizite Codieren auch sehr aufwändig, meist auch fehleranfällig, vielleicht aufgrund der Anzahl möglicher Kombinationen (man denke an ein Brettspiel wie Go) sogar niemals möglich. Wie genau das Lernen vonstatten geht, bleibt zunächst noch unbestimmt. Die Literatur unterscheidet hier im Wesentlichen drei Gruppen[3]: Supervised learning (im Deutschen meines Erachtens etwas unglücklich als *überwachtes Lernen* bezeichnet[4]), unsupervised learning (unüberwachtes Lernen) sowie reinforcement learning (manchmal auch als *bestärkendes Lernen* ins Deutsche übersetzt). Die drei Gruppen unterscheiden sich in den Voraussetzungen und in der Herangehensweise beim Lernen. Da diese Arbeit im Bereich des supervised learning anzusiedeln ist, werden die beiden anderen Konzepte nur oberflächlich beschrieben. Supervised und unsupervised learning zeichnen sich in ihrer typischen[5] Form dadurch aus, dass sie auf Basis eines zur Verfügung stehenden Datensatzes ein (wie auch immer geartetes) Modell erlernen, das anschließend zur Anwendung auf neue Datenpunkte aus der gleichen Verteilung herangezogen werden kann. Während beim supervised learning der Datensatz jedoch aus Informationen über erklärende und zu erklärende Variablen besteht (häufig als x- und y-Werte bezeichnet), stehen beim unsupervised learning nur Eingabewerte (x-Werte) zur Verfügung. Letzteres beschreibt daher Fragestellungen, bei denen der Begriff des Output-Wertes a priori unklar ist (so beispielsweise beim Clustering, der Schätzung des Trägers einer Verteilung oder bei der Auswahl „wichtiger“ Variablen in Form einer Dimensionsreduktion durch eine Hauptkomponentenanalyse). Supervised learning hingegen umfasst die wohlbekannten Aufgaben Klassifikation und Regression und ist dadurch charakterisiert, dass nicht nur x-Werte, sondern auch die zugehörigen y-Werte im Datensatz vorhanden sind. Daher sind in dieser Situation Überlegungen im Hinblick auf den Unterschied zwischen beobachtetem und erwartetem Wert sinnvoll. Eine dritte Gruppe bildet schließlich das reinforcement learning, das sich dadurch vom supervised learning unterscheidet, dass der Computer nicht passiv einen Trainingsdatensatz mit Daten-

[3] Manchmal wird darüber hinaus auch noch das sogenannte semi-supervised learning als eigene Gruppe angeführt, bei dem der Datensatz sowohl Datenpunkte mit Input- und Output-Werten enthält als auch solche, die nur Input-Werte aufweisen.

[4] Die Bezeichnung als *angeleitetes Lernen* wäre ggf. vorzuziehen. Eine analoge Anmerkung ist für das unsupervised learning vorzunehmen.

[5] Eine Ausnahme hiervon bietet das sogenannte Online-Learning, welches keine weitere Gruppe darstellt, sondern nur eine andere Datenlage. Während im typischen Fall der Datensatz vollständig zu Beginn des Lernvorgangs zur Verfügung steht, wird das Modell beim Eintreffen oder Zuführen neuer Datenpunkte beim Online-Learning immer wieder fortgeschrieben. In seiner Reinform beginnt das Lernen beim Online-Learning also mit dem ersten Datenpunkt, erzeugt auf dessen Basis ein Modell, zieht dann den zweiten Datenpunkt heran und erzeugt auf Basis des Vormodells und des neuen Datenpunktes ein fortgeschriebenes Modell usw.

punkten, die x- und y-Werte enthalten, erhält, sondern stattdessen Information über die Interaktion mit der Umgebung generiert. Es muss hier auch keine feste Verteilung geben, die die Datenpunkte generiert. Die Interaktion mit der Umgebung geschieht durch Aktionen (z. B. Spielzüge) und zwei Rückmeldungen: Einerseits verändert sich die Umgebung durch einen Spielzug; der neue Zustand wird dem Computer mitgeteilt. Andererseits wird bewertet, ob die Aktion des Computers positiv oder negativ (in der Regel auch wie positiv oder wie negativ) war; der Computer erhält also eine (positive oder negative) Auszahlung. Das Ziel des Computers besteht dann beispielsweise darin, die langfristige Summe der Auszahlungen zu maximieren. Dabei ist zu berücksichtigen, dass es einen Zielkonflikt zwischen Informationsgenerierung und Auszahlungsmaximierung gibt.

Bezugnehmend auf Vapnik (2000)[6] wird das folgende Schema betrachtet, das auch dem Rest dieser Arbeit gedanklich zugrunde liegt. Es beschreibt das allgemeine Modell des Lernens durch drei Komponenten:

(i) Es gibt einen Erzeuger der x-Werte (Input-Werte; Werte der erklärenden Variablen; Eingabewerte). Die Input-Werte werden in Vapniks Ausführungen unabhängig gezogen und entstammen einer festen, aber unbekannten (Rand-) Verteilung $P^{\mathcal{X}}$.[7] Im Rahmen dieser Arbeit bleibt diese Komponente – wie beschrieben – allgemeine gedankliche Voraussetzung.

(ii) Eine Instanz (von Vapnik als *Supervisor* bezeichnet), die jedem Input x einen Outputwert y (d. h. ein Label für eine Klassenzugehörigkeit oder einen Wert im Falle der Regression) gemäß einer festen, wenngleich ebenfalls unbekannten bedingten Verteilung $P(y|x)$ zuweist. $P^{\mathcal{X}}$ und $P(y|x)$ bilden zusammen die gemeinsame Verteilung P von Input- und Outputvariablen.

(iii) Schließlich braucht es die Lernmethode (*learning machine*), von Vapnik als dazu in der Lage beschrieben, eine Menge von Funktionen $f(x, \alpha)$, $\alpha \in \Lambda$, abbilden zu können, wobei Λ eine Menge von zunächst abstrakten Parametern ist.

Vapnik beschreibt das *Problem zu lernen* als die Aufgabe, diejenige Funktion aus jener Menge $\{f(x, \alpha) \mid \alpha \in \Lambda\}$ auszuwählen, die die Antwort des Supervisors zu gegebenen x auf Basis der gegebenen endlichen Stichprobe $((x_1, y_1), \ldots, (x_n, y_n))$ (unabhängig und identisch verteilt gemäß P) am besten approximiert. Vapnik selbst gibt

[6] und frühere Arbeiten, siehe Abschnitt 1.4

[7] Hierzu gibt es mindestens im Bereich der Support Vector Machines Erweiterungen, die die Unabhängigkeit und/oder die identische Verteilung nicht mehr voraussetzen, siehe beispielsweise Steinwart, Hush & Scovel (2009), Hang & Steinwart (2014), Hang (2015), Strohriegl & Hable (2016) und Strohriegl (2018).

an, wie diese *beste* Approximation greifbar werden soll: Verlustfunktionen $L(y, f(x, \alpha))$ haben die Aufgabe, den Unterschied zwischen beobachteter Antwort y und Prädiktion $f(x, \alpha)$ zu bemessen. Das Ziel besteht dann darin, das Risiko, d. h. den zu erwartenden Verlust bezüglich P über alle (x, y) zu minimieren. Gesucht wird also der Minimierer von $\int L(y, f(x, \alpha))\, dP(x, y)$ über alle Funktionen aus $\{f(x, \alpha) \mid \alpha \in \Lambda\}$. Die Verteilung P ist jedoch im Allgemeinen vollkommen unbekannt; allenfalls die Stichprobe $((x_1, y_1), \ldots, (x_n, y_n))$ liefert Informationen. Dies führt zur Minimierung des sogenannten empirischen Risikos $n^{-1} \sum_{i=1}^{n} L(y_i, f(x_i, \alpha))$ über die Funktionen aus $\{f(x, \alpha) \mid \alpha \in \Lambda\}$ auf Basis der Stichprobe in der Hoffnung, dass damit auch $\int L(y, f(x, \alpha))\, dP(x, y)$ minimiert wird. Im Englischen wird diese Herangehensweise als *empirical risk minimization (ERM)* bezeichnet. Das Vorgehen erscheint auf den ersten Blick plausibel, birgt aber die nicht zu unterschätzende Gefahr der Auswahl einer Funktion, die sich zu gut an die (endlich vielen) Daten aus der Stichprobe anpasst. Es ist wohlbekannt, dass für n paarweise verschiedene Datenpunkte ein eindeutiges Polynom vom Grad kleiner oder gleich $n - 1$ existiert, das diese n Datenpunkte interpoliert. Somit existiert immer eine Funktion, die das empirische Risiko auf 0 reduziert. Ein solches Polynom wird sich allerdings durch ein ständiges Auf- und Abschwingen schnell als unbrauchbar für die Statistik herausstellen. Als erster Ansatz würde daher stattdessen wohl eine lineare Regression gerechnet und dabei implizit eine Einschränkung auf ein Polynom ersten Grades vorgenommen. Eine Interpolation findet nun nicht mehr statt, vergleiche Abbildung 1.1.

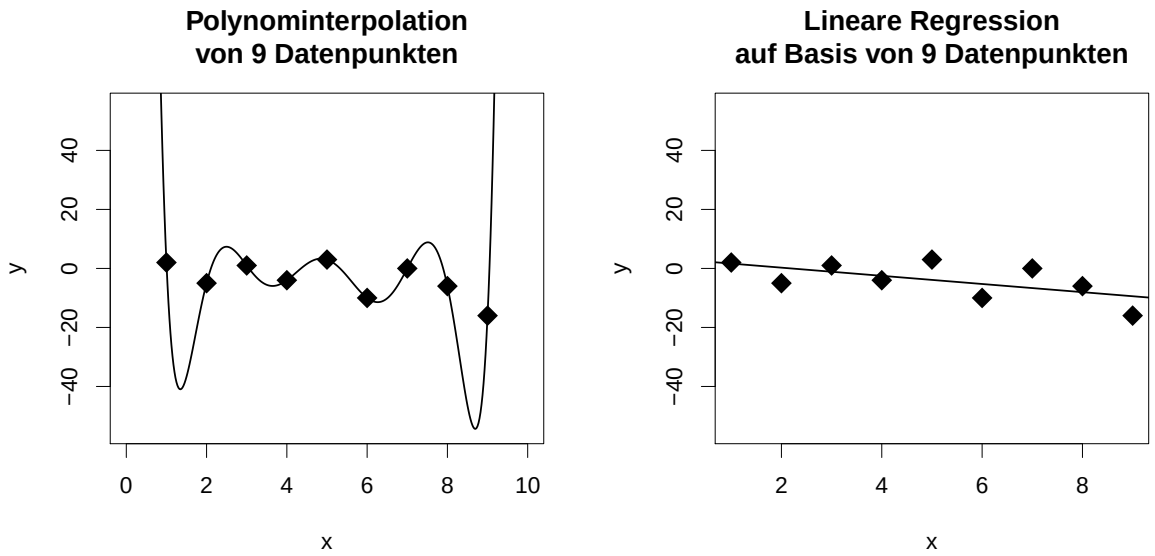


Abbildung 1.1: Vergleich von Polynominterpolation und linearer Regression

Die Einschränkung auf ein weniger kompliziertes Modell (in diesem Fall auf ein Polynom niedrigeren Grades) bietet augenscheinlich den Vorzug[8], auch für weitere Datenpunkte, die aus der gleichen Verteilung wie die bisherigen gezogen werden, gut geeignet zu sein, die zugrundeliegende Verteilung also besser zu approximieren als das Polynom achten Grades. Man spricht hier von der Verallgemeinerbarkeit (Generalisierbarkeit) bzw. von der Vermeidung einer Überanpassung (letztere wird in Anlehnung an das Englische auch im Deutschen häufig als *Overfitting* bezeichnet). Um ein solches Overfitting zu vermeiden, ist daher auch beim maschinellen Lernen darauf zu achten, dass das gelernte Modell nicht zu kompliziert, mithin verallgemeinerbar ist, der gelernte Prädiktor also eher ausgleichend als interpolierend arbeitet. Dies wird erreicht durch eine Einschränkung a priori auf eine kleinere Klasse von Funktionen $\{f(x,\alpha) \mid \alpha \in \Lambda_1\}$, $\Lambda_1 \subset \Lambda$, (z. B. sollen nur lineare Funktionen als Prädiktor in Frage kommen) oder durch einen explizit eingebauten Zielkonflikt zwischen Genauigkeit auf der Stichprobe und der „Komplexität“ des Prädiktors $f(x,\alpha)$. Der erste Fall wird in der Literatur auch als *inductive bias* bezeichnet, der zweite umfasst Regularisierung und *structural risk minimization*. Im Falle von SVMs wird vorwiegend der zweite Ansatz gewählt und dessen Umsetzung in Abschnitt 1.4 verdeutlicht.[9] Einen umfassenden Überblick über statistische maschinelle Lerntheorie liefert beispielsweise Shalev-Shwartz & Ben-David (2014).

Während maschinelles Lernen sich zunächst gedanklich an biologischen Lernvorgängen versuchte zu orientieren, wandelte sich diese Auffassung um die Jahrtausendwende. Vapnik (2000, S. 15) schreibt hierzu:

> Now a new methodological situation in the learning problem has developed where practical methods are the result of a deep theoretical analysis of the statistical bounds rather than the result on inventing new smart heuristics. This fact has in many respects changed the character of the learning problem.

Spätestens mit Ausarbeitung der mathematischen Theorie (im Wesentlichen aus den Bereichen der (Funktional-)Analysis und der Stochastik) kehrt die Thematik wieder in den Bereich der klassischen Statistik oder auch der Approximationstheorie[10] zurück und in der Tat erscheint eine scharfe Abgrenzung zwischen diesen Bereichen

[8] Diese Einsicht, bei im Wesentlichen gleicher Erklärungskraft das einfachere Modell zu wählen, ist in der Erkenntnistheorie wohlbekannt und firmiert dort häufig unter dem Label *Ockhams Rasiermesser* bzw. *Prinzip der Parsimonie*, siehe beispielsweise Mittelstraß (2004). Nichtsdestoweniger hat die Vermeidung von Interpolation hier auch handfeste statistische Gründe.

[9] Dass die Einschränkung der zur Verfügung stehenden Funktionen bei Support Vector Machines zwar ebenfalls vorliegt, das Ergebnis im Fall günstig gewählter Funktionenklassen jedoch nicht induktiv verzerrt, wird später noch deutlich: Geeignete reproduzierende Kern-Hilberträume sind groß genug, um mit Funktionen daraus jede messbare Funktion approximieren zu können.

[10] Als Referenzen seien hier beispielsweise Wendland (2005) und Cucker & Zhou (2007) genannt.

kaum möglich. Einzelne Methoden werden sowohl dem maschinellen Lernen als auch der klassischen Statistik zugeordnet, beispielsweise die Ridge-Regression, so Ghatak (2017) für maschinelles Lernen und Fahrmeir, Kneib, Lang & Marx (2009) für die klassische Statistik. Die Fähigkeit, aus Erfahrung (also aus Daten) zu lernen (d. h. ein Modell zu bilden), um später Entscheidungen (z. B. die Zuordnung eines neu erfassten Objekts zu einer Kategorie) ohne explizite (harte) Codierung treffen zu können, ist tatsächlich auch bereits im Konzept der klassischen Statistik enthalten. Viele der „neuen“ Methoden können aber erst mit zunehmender Leistungsfähigkeit der Computer brauchbar auf interessante Datensätze angewendet werden. Wohl aber ist festzuhalten, dass Methoden, die kaum bestritten dem maschinellen Lernen zuzuordnen sind, in der Regel mit einem deutlich höheren Rechen- und gegebenenfalls auch Speicheraufwand einhergehen als Methoden der klassischen Statistik. Außerdem orientieren sie sich häufig nicht mehr (stark) am Ziel, das Zustandekommen eines Outputs auf Basis des Inputs erklären zu können. Zielsetzung von supervised machine learning ist sehr häufig eine sehr gute Prädiktion, auch wenn dies zu Lasten der Interpretierbarkeit[11] geht. Auf das Vorliegen dieser zwei zum Teil konkurrierenden Zielsetzungen hat bereits Breiman (2001) hingewiesen; siehe auch Shmueli (2010).

1.2 Abgrenzung

Dieser Absatz enthält die Abgrenzung des Begriffs des maschinellen Lernens von zwei anderen Begriffen, die gegenwärtig inflationär gebraucht werden: Künstliche Intelligenz und Big Data.[12] Der Begriff der künstlichen Intelligenz umfasst nach Russell & Norvig (2016) die Komponenten

(i) Verarbeitung natürlicher Sprache (zur Kommunikation),

(ii) Wissensrepräsentation (Abspeichern und Organisieren vorhandener Informationen),

(iii) automatisches logisches Schließen (Schlussfolgerungen und Beantwortung von Fragen),

[11] Andererseits gibt es aus diesem Grund Bestrebungen, zumindest für konkrete Vorhersagen, die eine Methode des maschinellen Lernens ausgibt, Aussagen über ihr Zustandekommen zu treffen. Dies wird beispielsweise dadurch versucht, das Verhalten der Machine-Learning-Methode für den vorliegenden Eingabewert und für gegebenenfalls künstlich erzeugte weitere Eingabewerte „in dessen Nähe“ auszuwerten und durch ein einfach zu interpretierendes Modell (z. B. auf Basis einer linearen Regression oder eines Klassifikations- oder Regressionsbaumes) zu approximieren. Siehe hierzu beispielsweise Ribeiro, Singh & Guestrin (2016).

[12] In ähnlicher Form wurde dieser Abschnitt vom Autor dieser Arbeit bereits in Beck, Dumpert & Feuerhake (2018) eingebracht.

(iv) maschinelles Lernen (Anpassung an neue Umstände, Mustererkennung, Extrapolation),

(v) Computervision (Wahrnehmung von Objekten) und

(vi) Robotik (Manipulation und Bewegung von Objekten).

Die Fähigkeit zu maschinellem Lernen ist demzufolge notwendig für das Vorliegen von künstlicher Intelligenz, keinesfalls aber damit gleichzusetzen.

Auch Big Data beschreibt nicht das Gleiche wie maschinelles Lernen (schon semantisch nicht). Darüber hinaus indiziert weder Big Data maschinelles Lernen noch umgekehrt. Im Allgemeinen wird Big Data durch die drei Vs charakterisiert: Volume, Velocity und Variety (siehe beispielsweise die entsprechenden Aufsätze in König, Schröder & Wiegand (2017)). Suthaharan (2014) definiert Big Data als Datenlage: *Big Data meint einen Zeitpunkt, zu dem die Anzahl der Beobachtungen und Merkmale (Volume), das Fehlen oder zumindest die starke Verschiedenheit der Struktur der Daten (Variety) und die Geschwindigkeit des ständigen Nachströmens neuer Daten (Velocity) derart angestiegen sind, dass die aktuellen Techniken und Technologien nicht mehr in der Lage sind, die Speicherung und Verarbeitung der Daten zu bewältigen.*[13]

1.3 Bayesianische Statistik

Maschinelles Lernen, interpretiert als das Auffinden eines funktionalen Zusammenhangs, kann auch im Licht der Bayesianischen Statistik betrachtet werden, vgl. hierzu beispielsweise Vapnik (2000, Kapitel 4.11): Es wird die beste (z. B. im Sinne eines minimalen Risikos) Funktion gegeben die vorliegenden Daten gesucht. Wie üblich im Falle der Bayesianischen Statistik besteht auch hier die Notwendigkeit einer *a-priori-Dichtefunktion*, die für jede Funktion aus der Klasse, in der das Optimum gesucht wird, die Wahrscheinlichkeit angibt, die am besten geeignete Funktion zu sein. Sobald die Daten bekannt sind, kann man diese Wahrscheinlichkeit durch die durch die Daten gelieferte Information zur *a-posteriori-Wahrscheinlichkeit* einer Funktion, die beste zu sein, fortschreiben.[14] Die vorliegende Arbeit folgt jedoch keinem Bayesianischen Ansatz.

[13] Eigene, kommentierte Übersetzung des Autors der vorliegenden Arbeit aus Suthaharan (2014). Bemerkenswert an dieser Definition ist hierbei insbesondere ihre Zeitlosigkeit.

[14] Diese Sicht der Dinge motiviert die Bezeichnung des besten Risikos als Bayes-Risiko (bzgl. der Klasse der messbaren Funktionen) sowie die Bezeichnung der ggf. existierenden besten Entscheidungsfunktion (historisch im Bereich der Klassifikation entstanden) als Bayes-Entscheidungsfunktion.

1.4 Support Vector Machines im Speziellen

Support Vector Machines, letztlich zurückgehend auf Vapnik & Tscherwonenkis (1979), Boser, Guyon & Vapnik (1992) und Cortes & Vapnik (1995), haben im Bereich des supervised learnings das Ziel, den Einfluss einer Inputvariablen X, die zumeist (aber nicht notwendigerweise) multivariat, also aus $\mathbb{R}^d$ ist, auf eine univariate[15] Outputvariable Y zu untersuchen.[16] Wie in Abschnitt 1.1 bereits angedeutet, geht es darum, einen funktionalen Zusammenhang, der die bedingte Verteilung von Y gegeben eine Ausprägung von X beschreibt, zu erlernen. Um dies zu formalisieren wird ein Wahrscheinlichkeitsraum $(\Omega, \mathcal{A}, Q)$ betrachtet, der – wie üblich im Bereich der mathematischen Statistik – im Weiteren lediglich als abstrakter Urbildraum fungiert und darüber hinaus nicht von Interesse ist. Er ist jedoch erforderlich, um eine vollständige technische Beschreibung der Untersuchungsgegenstände liefern zu können. Hinsichtlich grundlegender Begrifflichkeiten und Standardaussagen (Wahrscheinlichkeitsraum, Zufallsvariable, Borel-σ-Algebra usw.) sei beispielsweise auf Hoffmann-Jørgensen (2003) verwiesen.

Es werden folgende grundlegende Annahmen getroffen und Notationen verwendet: $\mathfrak{B}_M$ steht für die Borel-σ-Algebra auf einer Menge M. Genutzt werden in dieser Arbeit ausschließlich Borel-σ-Algebren, d. h. eine messbare Menge ist eine Borel-messbare Menge und eine messbare Funktion ist messbar bezüglich der zuständigen Borel-σ-Algebren. Eine messbare Menge M ist stets vollständig messbar gedacht, d. h. $(M, \mathfrak{B}_M)$ ist vollständig für jedes Wahrscheinlichkeitsmaß, vgl. z. B. Ash & Doleans-Dade (2000, Definition 1.3.7). Betrachtet werden Zufallsvariablen $X : (\Omega, \mathcal{A}) \to (\mathcal{X}, \mathfrak{B}_\mathcal{X})$ und $Y : (\Omega, \mathcal{A}) \to (\mathcal{Y}, \mathfrak{B}_\mathcal{Y})$ mit gemeinsamer Verteilung $P := (X, Y) \circ Q$ auf $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$. Die Menge $\mathcal{X}$ (der Eingaberaum) wird generell als separabler metrischer Raum vorausgesetzt; einzelne Resultate benötigen stärkere Annahmen an $\mathcal{X}$. Hinsichtlich der Begriffe eines metrischen Raumes, der Separabilität, eines Polnischen Raumes usw. sei auf Dunford & Schwartz (1958) verwiesen. Der Ausgaberaum $\mathcal{Y}$ wird generell als abgeschlossene Teilmenge der reellen Zahlen $\mathbb{R}$ vorausgesetzt. Falls $\mathcal{Y}$ endlich ist (d. h. aus nur endlich vielen Elementen besteht), handelt es sich um Klassifikation, anderenfalls um Regression.

Betrachtet wird nun der in Abschnitt 1.1 beschriebene Prozess, dass die Natur zunächst eine Realisierung $x = X(\omega)$ erzeugt und anschließend das zugehörige $y = Y(\omega)$ durch den Supervisor gebildet wird. Wie erwähnt soll nun mindestens

[15] Erweiterungen für multivariaten Output sind möglich und wurden beispielsweise durch Micchelli & Pontil (2005) oder Caponnetto & De Vito (2007) besprochen; für die Betrachtung funktionaler Daten wird auf Kadri, Duflos, Preux, Canu & Davy (2010) und Kadri, Duflos, Preux, Canu, Rakotomamonjy & Audiffren (2016) verwiesen.

[16] Diese Einführung in Support Vector Machines ist eine Übersetzung ins Deutsche und gleichzeitige Erweiterung der einführenden Kapitel in Dumpert & Christmann (2018) und Dumpert (2019b).

ein Charakteristikum (z. B. ein Lagemaß) der bedingten Verteilung von Y gegeben X geschätzt werden. Da $\mathcal{Y}$ eine abgeschlossene Teilmenge von $\mathbb{R}$ ist, ist es ein Polnischer Raum. Daher gibt es eine eindeutige, reguläre bedingte Verteilung von Y gegeben $X = x$ und die gemeinsame Verteilung P kann in die Randverteilung $P^{\mathcal{X}}$ und die bedingte Verteilung $P(\cdot|x) := P(\cdot|X = x)$ aufgespalten werden, siehe Dudley (2004, Theorem 10.2.1, Theorem 10.2.2). Der Eingaberaum $\mathcal{X}$ muss nicht notwendigerweise ein Polnischer Raum sein, insbesondere wird zunächst keine Vollständigkeitsannahme[17] an $\mathcal{X}$ benötigt.

Datensätze (oder auch Stichproben, beobachtete Daten usw.) werden für $n \in \mathbb{N}$ als n-Tupel $\mathcal{D}_n$ von unabhängig und identisch verteilten Beobachtungen definiert:

$$\begin{aligned}\mathcal{D}_n &= ((x_1, y_1), \ldots, (x_n, y_n)) \\ &:= \mathfrak{D}_n(\omega) := ((X_1(\omega), Y_1(\omega)), \ldots, (X_n(\omega), Y_n(\omega))) \in (\mathcal{X} \times \mathcal{Y})^n,\end{aligned}$$

wobei $\mathfrak{D}_n : (\Omega, \mathcal{A}) \to (\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})^n$ die die Stichprobe erzeugende Zufallsvariable ist. Erlaubt werden soll auch der Fall $n \to \infty$, um asymptotische Eigenschaften zu untersuchen. Wohlwissend, dass es sich um ein Tupel handelt, werden die mengentheoretischen Operatoren $\in, \cap$ usw. genutzt; die Tupel werden insofern wie Mengen behandelt. Allerdings ist klar (und soll auch hier so gehandhabt werden), dass Tupel einen Datenpunkt mehr als einmal enthalten können.

Support Vector Machines (wie andere statistische Methoden auch) werden eingesetzt, um eine gute Vorhersage $f(x)$ von y gegeben einen Eingabewert x zu erhalten.[18] Support Vector Machines bieten auf vielerlei Fragestellungen eine Antwort, Beispiele dazu sind im Folgenden genannt. y steht dabei für das Label der Klasse (genauer: für dessen numerische Codierung) im Fall der Klassifikation (Christmann, 2002), einen Rang bei ordinaler Regression (Herbrich, Graepel & Obermayer, 1999), ein Quantil (Steinwart & Christmann, 2011), einen Erwartungswert (oder etwas, das diesen substituiert, Steinwart & Christmann (2009)) oder ein Expectile (Farooq & Steinwart, 2017) der bedingten Verteilung von Y gegeben ein spezifiziertes x.[19]

Für $n \in \mathbb{N}$ wird ein Operator $S : (\mathcal{X} \times \mathcal{Y})^n \to \{f : \mathcal{X} \to \mathbb{R} \mid f \text{ messbar}\}$, welcher einem vorliegenden Datensatz $\mathcal{D}_n$ einen Prädiktor $f_{\mathcal{D}_n}$ zuweist, statistische Lernmethode (*statistical learning method*) genannt. Selbstverständlich ist man an sinnvollen

[17] Vollständigkeit in dem Sinne, dass jede Cauchy-Folge in $\mathcal{X}$ einen Grenzwert in $\mathcal{X}$ hätte.

[18] Man unterstellt manchmal – jedoch nicht notwendigerweise für diese Arbeit – einen Zusammenhang der Art $y = f(x) + \varepsilon$, wobei ε eine Zufallsvariable darstellt, die für einen Zufallsfehler steht. Der hier implizierte additive Einfluss des Fehlers ist jedoch nicht Voraussetzung für die weiteren Untersuchungen.

[19] Auch denkbar sind Aufgaben im Bereich des Rankings (Clémençon, Lugosi & Vayatis, 2008; Agarwal & Niyogi, 2009), Metrik- und Ähnlichkeitslernen (Mukherjee & Zhou, 2006; Xing, Ng, Jordan & Russell, 2003; Cao, Guo & Ying, 2016) oder Minimum-Entropie-Lernen (Hu, Fan, Wu & Zhou, 2013; Fan, Hu, Wu & Zhou, 2016).

Operatoren interessiert, also an solchen, die letztlich zu guten Vorhersagen führen. Offensichtlich entsteht nun die Notwendigkeit zu präzisieren, was eine gute Vorhersage ist. In dieser Arbeit wird hierfür in Anlehnung an Vapnik (siehe Abschnitt 1.1) der Zugang über Verlustfunktionen und die sogenannten Risiken gewählt. Die Aufgabe einer Verlustfunktion besteht in dem Vergleich zwischen vorhergesagtem Wert und zugehörigem wahren (oder beobachteten) Wert. Je nach Fragestellung (auch innerhalb von Klassifikation und Regression) ist eine andere Verlustfunktion zu wählen, um das gewünschte Ergebnis zu erhalten, vgl. Rosasco, De Vito, Caponnetto, Piana & Verri (2004), Steinwart (2007) und Steinwart & Christmann (2008, Chapter 2, Chapter 3). Formal ist eine supervised-Verlustfunktion (im Folgenden auch kurz: eine Verlustfunktion) als messbare Funktion $L : \mathcal{Y} \times \mathbb{R} \to [0, \infty[$ definiert.[20] Aus technischen Gründen ist außerdem die geshiftete Version L^* einer Verlustfunktion L von Interesse, die durch $L^* : \mathcal{Y} \times \mathbb{R} \to \mathbb{R},\ L^*(y,t) := L(y,t) - L(y,0)$ definiert wird. Das wesentliche Ziel besteht darin, Annahmen an die Existenz von Momenten vermeiden zu können. Dieser Aspekt wird in Dumpert & Christmann (2018, Appendix B) anschaulich dargestellt, vgl. auch Anhang A; die Verwendung geht zurück auf Christmann, Van Messem & Steinwart (2009) im Bereich der Support Vector Machines und letztlich auf Huber (1967), vgl. auch Huber & Ronchetti (2009, S. 46f.). Wird exakt der wahre (oder beobachtete) Wert vorhergesagt, so soll die Verlustfunktion einen Wert von 0 liefern, d. h. $L(y,y) = 0$ für alle $y \in \mathcal{Y}$. Die meisten gängigen Verlustfunktionen erfüllen diese Voraussetzung. Eine Ausnahme stellt die logistische Verlustfunktion für Klassifikation dar. Gängige Verlustfunktionen im Umfeld dieser Arbeit sind für binäre Klassifikation, d. h. im Fall von $\mathcal{Y} = \{-1, 1\}$,

(a) $L_{LS}(y, f(x)) = (1 - yf(x))^2$,

(b) $L_{hinge}(y, f(x)) = \max\{0, 1 - yf(x)\}$,

und für (Quantils-)Regression, $\mathcal{Y} = \mathbb{R}$, beispielsweise

(c) $L_{LS}(y, f(x)) = (y - f(x))^2$,

(d) $L_{\varepsilon\text{-}ins}(y, f(x)) = \max\{0, |y - f(x)| - \varepsilon\}$,

(e) $L_{\alpha\text{-}Huber}(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{, falls } |y - f(x)| \leq \alpha \\ \alpha|y - f(x)| - \frac{\alpha^2}{2}, & \text{, sonst} \end{cases}, \quad \alpha > 0,$

(f) $L_{\tau\text{-}pinball}(y, f(x)) = \begin{cases} (\tau - 1)(y - f(x)) & \text{, falls } y - f(x) < 0 \\ \tau(y - f(x)), & \text{, sonst} \end{cases}, \quad \tau \in]0, 1[.$

mit den in Tabelle 1.1 dargestellten Eigenschaften.

[20] Verlustfunktionen in der unüberwachten Situation würden stattdessen mit $L : \mathcal{X} \times \mathbb{R} \to [0, \infty[$ definiert. Da in dieser Arbeit aber nur der Fall des supervised learning betrachtet wird, wird auf eine allgemeinere Darstellung verzichtet.

	Einsatzzweck	L	Lipschitzstetig	zweifach differenzierbar	resultierendes Problem[‡]
(a)	Klassifikation	L_{LS}	nein	ja	LP
(b)	Klassifikation	L_{hinge}	ja	nein	boxed QP
(c)	Regression	L_{LS}	nein	ja	LP
(d)	Regression	$L_{\varepsilon\text{-}ins}$	ja	nein	boxed QP
(e)	Regression	$L_{\alpha\text{-}Huber}$	ja	nein	boxed QP
(f)	Regression	$L_{\tau\text{-}pinball}$	ja	nein	boxed QP

[‡] LP steht für *Lineares Programm*, boxed QP für ein *Quadratisches Problem mit Box-Constraints.*

Tabelle 1.1: Eigenschaften von supervised-Verlustfunktionen

Offensichtlich gibt es keine „beste“ Verlustfunktion für Klassifikation oder Regression. Neben den bislang genannten kommen weitere Verlustfunktionen in Literatur und Praxis zum Einsatz, insbesondere die Lipschitz-stetigen und zweifach Fréchet-differenzierbaren logistischen Verlustfunktionen $L_{r\text{-}log}(y, f(x)) = -\ln\left(\frac{4e^{y-f(x)}}{(1+e^{y-f(x)})^2}\right)$ für Regression und $L_{c\text{-}log}(y, f(x)) = \ln(1+e^{-yf(x)})$ für Klassifikation mit resultierendem (lediglich) konvexen Optimierungsproblem.

Die einzige Information, die über die allem zugrundeliegende Verteilung P bekannt ist, ist durch die Stichprobe $\mathcal{D}_n$ gegeben. Es ist daher im Allgemeinen nicht zu erwarten, auf dieser Basis einen Prädiktor $f_{\mathcal{D}_n}$ bestimmen zu können, der $L(y, f_{\mathcal{D}_n}(x)) = 0$ für alle $x \in \mathcal{X}, y \in \mathcal{Y}$ erfüllt. Das mag, wie im Abschnitt 1.1 bereits dargestellt, für alle Datenpunkte (x_i, y_i) , $i = 1, \ldots, n$, der Stichprobe $\mathcal{D}_n$ möglich (wenngleich im Hinblick auf die Verallgemeinerbarkeit nicht sinnvoll) sein. Eine Methode, die die Stichprobenwerte interpoliert, ist höchst anfällig für das Phänomen des Overfittings, der Überanpassung an den vorhandenen Datensatz und büßt somit in der Regel die Fähigkeit zur Verallgemeinerung des Modells auf alle bezüglich P relevanten $x \in \mathcal{X}, y \in \mathcal{Y}$ ein. Dass die Generalisierbarkeit erwünscht ist, leuchtet ein, wenn man Prädiktionen (von y) auf Basis neuer, bislang nicht beobachteter Eingabewerte x und das Vorhandensein von Messfehlern in Betracht zieht. Günstiger ist es daher, den mittleren Verlust über alle möglichen $x \in \mathcal{X}, y \in \mathcal{Y}$ zu minimieren. Dieser durchschnittliche Verlust heißt dann das *(theoretische) Risiko über $\mathcal{X}$ eines messbaren Prädiktors f bezüglich einer Verlustfunktion L und der unbekannten zugrundeliegenden Verteilung P* und ist formal definiert als

$$\mathcal{R}_{\mathcal{X},L,P} : \{f : \mathcal{X} \to \mathbb{R} \mid f \text{ messbar}\} \to \mathbb{R},\ \mathcal{R}_{\mathcal{X},L,P}(f) := \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x))\ dP(x, y).$$

Wird die geshiftete Verlustfunktion von L genutzt, lautet die Definition analog

$$\mathcal{R}_{\mathcal{X},L^*,P}(f) := \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) - L(y, 0)\; dP(x, y).$$

Selbst dann, wenn alle Situationen, d. h. alle[21] gemäß P möglichen Kombinationen $(x, y) \in \mathcal{X} \times \mathcal{Y}$, bekannt wären, ist im Allgemeinen nicht zu erwarten, dass das Risiko eines messbaren Prädiktors bezüglich L und P gleich 0 sein wird. Dies liegt darin begründet, dass der wahre funktionale Zusammenhang zwischen x und y gegebenenfalls durch eine nicht messbare Funktion dargestellt wird.[22] Das Ziel besteht also darin, eine messbare Funktion f zu finden, deren Risiko dem kleinsten Risiko entspricht, das beim Einsatz eines messbaren Prädiktors erreichbar ist:

$$\mathcal{R}^*_{\mathcal{X},L,P} := \inf\left\{\mathcal{R}_{\mathcal{X},L,P}(f) \mid f : \mathcal{X} \to \mathbb{R} \text{ messbar}\right\},$$

das sogenannte *Bayes-Risiko auf $\mathcal{X}$ bezüglich L und P*. Das entsprechende Bayes-Risiko bei Verwendung der geshifteten Version einer Verlustfunktion L ist definiert als

$$\mathcal{R}^*_{\mathcal{X},L^*,P} := \inf\left\{\mathcal{R}_{\mathcal{X},L^*,P}(f) \mid f : \mathcal{X} \to \mathbb{R} \text{ messbar}\right\}.$$

Ohne weitere Annahmen ist das Optimierungsproblem NP-schwer (Höffgen, Simon & Van Horn, 1995). Hieraus folgt die Notwendigkeit, die Klasse der betrachteten Funktionen einzuschränken. Ist $\mathcal{F}$ eine Teilmenge der messbaren Funktionen von $\mathcal{X}$ nach $\mathbb{R}$, so sei

$$\mathcal{R}^*_{\mathcal{X},L,P,\mathcal{F}} := \inf\left\{\mathcal{R}_{\mathcal{X},L,P}(f) \mid f \in \mathcal{F}\right\} \quad \text{und} \quad \mathcal{R}^*_{\mathcal{X},L^*,P,\mathcal{F}} := \inf\left\{\mathcal{R}_{\mathcal{X},L^*,P}(f) \mid f \in \mathcal{F}\right\}.$$

Wenn die Integration nicht über $\mathcal{X}$, sondern nur über eine messbare Teilmenge $\Xi \subset \mathcal{X}$ stattfinden soll, wird eine entsprechende Notation verwendet:

$$\mathcal{R}_{\Xi,L,P}(f) := \int_{\Xi\times\mathcal{Y}} L(y, f(x))\; dP(x, y) \text{ bzw. } \mathcal{R}_{\Xi,L^*,P}(f) := \int_{\Xi\times\mathcal{Y}} L^*(y, f(x))\; dP(x, y).$$

Motiviert durch das Gesetz der großen Zahlen soll nun die in der Stichprobe enthaltene Information zum Lernen[23] eines Prädiktors genutzt werden, dessen Risiken den oben genannten Bayes-Risiken möglichst nahe kommen. Sei im Folgenden $D_n := n^{-1}\sum_{i=1}^n \delta_{(x_i,y_i)}$ die empirische Verteilung basierend auf $\mathcal{D}_n$, wobei $\delta_{(x_i,y_i)}$ das Dirac-Maß im Punkt $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ ist. Dieses empirische Maß ist natürlich selbst

[21] i. d. R. überabzählbar viele

[22] Es ist allerdings einzusehen, dass diese Einschränkung in der Praxis nur eine untergeordnete Rolle spielt.

[23] Hier und in der gesamten Arbeit bezeichnet der Ausdruck *Lernen des/eines Prädiktors* den Vorgang, dass der Prädiktor, also die Schätz- oder Klassifikationsfunktion, berechnet, mithin gelernt (oder erlernt) wird.

eine Zufallsgröße, da die Stichprobe $\mathcal{D}_n$ eine Realisierung von Zufallsvariablen ist. Darauf aufbauend kann nun das *empirische Risiko von f auf $\mathcal{X}$ bezüglich L* (und analog bezüglich L^*) definiert werden:

$$\mathcal{R}_{\mathcal{X},L,D_n}(f) := \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)).$$

Bei Betrachtung von messbaren Teilmengen Ξ von $\mathcal{X}$ sei analog

$$\mathcal{R}_{\Xi,L,D_n}(f) := \frac{1}{|\mathcal{D}_n \cap \Xi|}\sum_{(x_i,y_i)\in\mathcal{D}_n\cap\Xi} L(y_i, f(x_i)),$$

wobei $|M|$ die Anzahl der Elemente einer endlichen Menge M bezeichne.

Der Prädiktor wird nun derart gelernt, dass er das empirische Risiko minimiert. Um dabei eine Überanpassung zu vermeiden, wird die Komplexität des Prädiktors kontrolliert, indem ein Regularisierungsterm[24] $p(\lambda, f)$ additiv ergänzt wird. Dabei steht $\lambda > 0$ für den Einfluss dieses Strafterms im Minimierungsproblem. In dieser Arbeit wird $p(\lambda, f) := \lambda\|f\|_H^2$ verwendet. Die Literatur weist weitere Möglichkeiten, besonders für lineare Support Vector Machines, aus, darunter ℓ_1-Regularisierung, falls Sparsity ein besonderes Ziel darstellt (Zhu, Rosset, Hastie & Tibshirani, 2004), oder sogenannte elastic nets, vgl. Zou & Hastie (2005), Wang, Zhu & Zou (2006) und De Mol, De Vito & Rosasco (2009). Weitere Varianten, wie beispielsweise $\lambda\|f\|_H^q$ für $q \geq 1$, sind ebenfalls denkbar, treten allerdings in Theorie und Praxis kaum auf. Es leuchtet ein, dass λ von der Größe der Stichprobe abhängen sollte.

Im Fall der SVMs in dieser Arbeit ist H ein aus messbaren Funktionen bestehender sogenannter *reproduzierender Kern-Hilbertraum (RKHS)*. Weitere Anmerkungen hierzu folgen im Nachgang. Zunächst sei festzuhalten, dass das Ziel darin besteht, das folgende Problem zu lösen:

$$\text{minimiere}\quad \mathcal{R}_{\mathcal{X},L,D_n,\lambda_n}(f) := \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda_n\|f\|_H^2$$

oder

$$\text{minimiere}\quad \mathcal{R}_{\mathcal{X},L^*,D_n,\lambda_n}(f) := \frac{1}{n}\sum_{i=1}^{n} L^*(y_i, f(x_i)) + \lambda_n\|f\|_H^2,$$

über einer geeigneten Funktionenklasse und ausschließlich basierend auf einer Stichprobe $\mathcal{D}_n$ von Beobachtungen basierend auf P. Gefunden werden soll also die sogenannte *empirische Support Vector Machine*

$$f_{L^*,D_n,\lambda_n} := \arg\inf_{f\in H}\ \frac{1}{n}\sum_{i=1}^{n} L^*(y_i, f(x_i)) + \lambda_n\|f\|_H^2.$$

[24] manchmal auch als Strafterm bezeichnet

Die Ausdrücke $\mathcal{R}_{\mathcal{X},L,P,\lambda_n}(\cdot)$, $\mathcal{R}_{\mathcal{X},L^*,P,\lambda_n}(\cdot)$ bzw. f_{L^*,P,λ_n} sind analog für P statt für D_n zu verstehen; in diesem Fall wird das theoretische, regularisierte Risiko betrachtet bzw. minimiert.

In der Praxis wird die Wahl der Verlustfunktion durch die gewünschte Eigenschaft des Prädiktors festgelegt, d. h. anhand der Frage, was vorausgesagt werden soll. Die Wahl des „richtigen“ RKHS hingegen ist weniger offensichtlich. Dank des bijektiven Zusammenhangs zwischen sogenannten Kernen und deren reproduzierenden Kern-Hilberträumen kann diese Frage jedoch auf die Wahl eines geeigneten Kerns reduziert werden. Dieser Aspekt wird im Folgenden näher ausgeführt. Support Vector Machines und andere kernbasierte Methoden nutzen eine Theorie, die auf RKHS basiert. Aus mathematischer Sicht bestehen die Vorzüge darin, dass RKHS mächtig genug sein können, um Prädiktoren mit Risiken nahe der Bayes-Risiken zu enthalten; andererseits aber auch klein genug sein können, um darüber (nicht NP-schwere) Optimierungsprobleme lösen zu können. Eine Einführung und die allgemeine Theorie werden insbesondere in Aronszajn (1950), Schölkopf & Smola (2001), Berlinet & Thomas-Agnan (2001) und Paulsen & Raghupathi (2016) dargestellt. Einige grundlegende Definitionen und Eigenschaften daraus seien hier wiederholt. Ein Kern (auf $\mathcal{X}$) ist eine Funktion $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \quad (x, x') \mapsto k(x, x')$, die symmetrisch und positiv semi-definit ist, d. h. für alle $x, x' \in \mathcal{X}$ ist $k(x, x') = k(x', x)$ und für alle $n \in \mathbb{N}$ gilt $\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(x_i, x_j) \geq 0$ für alle $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ und alle $x_1, \ldots, x_n \in \mathcal{X}$. Kerne messen die Ähnlichkeit ihrer beiden Argumente zueinander.[25] Ein Kern k heißt *reproduzierender Kern eines Hilbertraumes* H, falls $k(\cdot, x) \in H$ für alle $x \in \mathcal{X}$ und $f(x) = \langle f, k(\cdot, x) \rangle_H$ für alle $x \in \mathcal{X}$ und alle $f \in H$. $\langle \cdot, \cdot \rangle_H$ steht hierbei für das innere Produkt (Skalarprodukt) auf H, $\| \cdot \|_H$ für die dadurch induzierte Norm auf H. In diesem Fall ist H der reproduzierende Kern-Hilbertraum (RKHS) von k.[26] Dass die in gängiger Software häufig als Standard voreingestellten Gaußkerne diesen (herausragenden) Status zurecht genießen, liegt an ihren hervorragenden analytischen Eigenschaften, siehe beispielsweise Christmann, Dumpert & Xiang (2016). Wichtige Eigenschaften von reproduzierenden Kern-Hilberträumen für diese Arbeit liefern die folgenden Propositionen (Steinwart & Christmann, 2008, Lemma 4.23, Lemma 4.28).

Proposition 1.1

Ein Kern k heißt beschränkt, wenn $\|k\|_\infty := \sup_{x \in \mathcal{X}} \sqrt{k(x, x)} < \infty$. Dann und nur dann, wenn der reproduzierende Kern k eines RKHS H beschränkt ist, ist je-

[25] Kerne können auch als $\mathbb{C}$-wertige Funktionen definiert werden. Dies kann nützlich sein, wenn bestimmte Eigenschaften von Kernen bewiesen werden sollen. Für die vorliegende Arbeit ist dies jedoch nicht notwendig.

[26] Details zur Bijektion zwischen Kernen und ihren RKHS finden sich beispielsweise in Berlinet & Thomas-Agnan (2001, Moore-Aronszajn Theorem, S. 19).

des $f \in H$ beschränkt und für jedes $f \in H, x \in \mathcal{X}$ gilt die Ungleichung $|f(x)| = |\langle f, k(\cdot, x)\rangle_H| \leq \|f\|_H \|k\|_\infty$. Insbesondere:

$$\|f\|_\infty \leq \|f\|_H \|k\|_\infty. \tag{1.1}$$

Proposition 1.2
Sei k ein Kern mit RKHS H. Dann ist k beschränkt mit $k(\cdot, x) : \mathcal{X} \to \mathbb{R}$ stetig für alle $x \in \mathcal{X}$, wenn und nur wenn alle $f \in H$ stetig und beschränkt sind. Offensichtlich gilt: Falls sogar $k(\cdot, \cdot)$ stetig ist, so ist $k(\cdot, x) : \mathcal{X} \to \mathbb{R}$ stetig für alle $x \in \mathcal{X}$.

Brauchbare statistische Methoden sollten zumindest konsistent in einem geeigneten Sinne sein, also mit zunehmender Information, d. h. zunehmendem (informativen) Stichprobenumfang, einem besten oder wahren Wert immer näher kommen und schließlich dagegen konvergieren, vgl. auch Abschnitt 1.5. Dies könnte man als Minimalanforderung an eine Methode auffassen. Dabei darf die asymptotische Eigenschaft der Konsistenz von der zugrundeliegenden (unbekannten) Verteilung P abhängen. Kann man sie für alle Verteilungen P zeigen, so heißt eine statistische Methode *universell konsistent*. Im Falle von Support Vector Machines wird universelle Risiko-Konsistenz gezeigt, d. h.

$$\mathcal{R}_{\mathcal{X},L^*,P}(f_{L^*,D_n,\lambda_n}) \xrightarrow[n\to\infty]{} \mathcal{R}^*_{\mathcal{X},L^*,P} \quad \text{in Wahrscheinlichkeit bzgl. } P.$$

SVMs erfüllen diese Eigenschaften unter schwachen Voraussetzungen. Für die Situation in dieser Arbeit sei auf Christmann, Van Messem & Steinwart (2009, Theorem 8) verwiesen, für andere Situationen beispielsweise auf Fan, Hu, Wu & Zhou (2016, minimum error entropy), Christmann & Hable (2012, additive Modelle) oder Strohriegl (2018, abhängige, nicht identisch verteilte Daten).

Aus technischer Sicht werden noch einige Definitionen und Eigenschaften benötigt. Bei Verlustfunktionen beziehen sich die Eigenschaften in dieser Arbeit stets auf das zweite Argument: Eine Verlustfunktion L heißt (strikt) konvex, falls $t \mapsto L(y,t)$ (strikt) konvex für alle $y \in \mathcal{Y}$ ist. Ihre geshiftete Version L^* heißt (strikt) konvex, falls $t \mapsto L^*(y,t)$ (strikt) konvex für alle $y \in \mathcal{Y}$ ist. L heißt Lipschitz-stetig, falls es eine Konstante $|L|_1 \in [0, \infty[$ gibt, sodass für alle $y \in \mathcal{Y}$ und alle $t, s \in \mathbb{R}$ gilt: $|L(y,t) - L(y,s)| \leq |L|_1 |t - s|$. Ebenso heißt L^* Lipschitz-stetig, falls es eine Konstante $|L^*|_1 \in [0, \infty[$ gibt, sodass für alle $y \in \mathcal{Y}$ und alle $t, s \in \mathbb{R}$ gilt: $|L^*(y,t) - L^*(y,s)| \leq |L^*|_1 |t - s|$. Wie oben beschrieben, ist die Betrachtung der geshifteten Verlustfunktion im Wesentlichen technischer Natur. In der Tat ist es so, dass $f_{\mathcal{X},L^*,P,\lambda} = f_{\mathcal{X},L,P,\lambda}$, falls $\mathcal{R}_{\mathcal{X},L,P}(0) < \infty$. In diesem Fall ist es nicht erforderlich, mit L^* statt mit L zu arbeiten; andere Algorithmen o. ä. sind somit nicht

notwendig (Christmann, Van Messem & Steinwart, 2009). Die folgenden Propositionen aus Christmann, Van Messem & Steinwart (2009) und Steinwart & Christmann (2008) wiederholen einige Eigenschaften zu geshifteten Verlustfunktionen und den zugehörigen Support Vector Machines.

Proposition 1.3
Wenn eine Verlustfunktion L (strikt) konvex ist, dann ist auch L^* (strikt) konvex. Wenn eine Verlustfunktion L Lipschitz-stetig ist, dann ist auch L^* Lipschitz-stetig mit der gleichen Lipschitz-Konstante. Wenn L eine Lipschitz-stetige Verlustfunktion ist und $f \in L^1(P^{\mathcal{X}})$, dann ist $-\infty < \mathcal{R}_{\mathcal{X},L^*,P}(f) < \infty$. Wenn L eine Lipschitz-stetige Verlustfunktion ist und $f \in L^1(P^{\mathcal{X}}) \cap H$, dann ist $\mathcal{R}_{\mathcal{X},L^*,P,\lambda}(f) > -\infty$ für alle $\lambda > 0$.[27]

Proposition 1.4
Die empirische SVM bezüglich $\mathcal{R}_{\mathcal{X},L,D_n,\lambda}$ und die empirische SVM bezüglich $\mathcal{R}_{\mathcal{X},L^*,D_n,\lambda}$ existieren und sind eindeutig für jedes $\lambda \in]0,\infty[$ und jede Stichprobe $\mathcal{D}_n \in (\mathcal{X}\times\mathcal{Y})^n$, falls L konvex ist.[28] *Die theoretischen SVMs existieren und sind eindeutig für alle $\lambda \in]0,\infty[$, falls L eine Lipschitz-stetige und konvexe Verlustfunktion und $H \subset L^1(P^{\mathcal{X}})$ der RKHS eines beschränkten und messbaren Kerns ist.*

1.5 Wünschenswerte Eigenschaften

Es ist legitim, die Frage zu stellen, welche Eigenschaften eine statistische Methode aufweisen soll, um ein brauchbares Hilfsmittel für die Wissenschaft zu sein. Als Minimalanforderung hat sich für Methoden, die (in jeweils geeigneter Weise) Schätzungen vornehmen, die asymptotische Eigenschaft der Konsistenz herausgebildet. Die Konvergenz (in noch näher zu bestimmender Weise) eines Schätzers mit zunehmendem Stichprobenumfang gegen den wahren oder besten Wert wird als grundlegend für die Statistik angesehen. Support Vector Machines erfüllen – wie oben beschrieben – diese Anforderung im Sinne der Risiko-Konsistenz. Wünschenswert ist es, wenn man Konsistenz für alle zugrundeliegenden Verteilungen beweisen kann; in diesem Fall spricht man von universeller Konsistenz. Support Vector Machines sind unter geeigneten Voraussetzungen universell konsistent. Anders als beispielsweise bei der linearen Regression ist es bei Support Vector Machines nicht möglich, (in der zugrundeliegenden Verteilung) gleichmäßige Konvergenzgeschwindigkeiten (Lernraten) zu zeigen (*no free lunch theorem*, vgl. Devroye (1982)). Dies impliziert, dass

[27] Bei der letzten Aussage muss $f \in H$, also im zum Kern gehörigen RKHS sein, damit $\|f\|_H$ Sinn ergibt.
[28] Bezüglich $\mathcal{R}_{\mathcal{X},L^*,D_n,\lambda}$ ist als Argument zu ergänzen, dass zu gegebener Stichprobe $\mathcal{D}_n$ der Term $n^{-1}\sum_{i=1}^n L(y_i,0)$ endlich und konstant ist.

es für alle Anwendungsfälle zugleich (d. h. auch für Fälle ohne nachprüfbare Eigenschaften der zugrundeliegenden Verteilung und bei Vorliegen von nur endlich vielen Beobachtungen) nicht allgemein möglich ist, eine Aussage darüber zu treffen, welche statistische Methode die bessere oder gar die beste ist.[29] Trifft man zusätzliche Annahmen, was in der vorliegenden Arbeit jedoch vermieden werden soll, so können Lernraten hergeleitet werden, siehe unter anderem Eberts & Steinwart (2011), Eberts & Steinwart (2013), Eberts (2015), Blaschzyk & Steinwart (2018) und Farooq & Steinwart (2019).[30]

Häufig genannt werden außerdem *three principles of data science: predictability, stability, and computability*.[31] Die Begriffe, insbesondere *stability* und *predictability*, werden dabei nicht einheitlich verwendet. In letzter Zeit kommt auch die Interpretierbarkeit verstärkt hinzu.[32] Während die Konsistenz die *predictability* abdeckt[33], gilt es, genauere Begriffe für die Stabilität und die Berechenbarkeit zu finden. So erachten beispielsweise Shawe-Taylor & Cristianini (2004) in ihrem Kapitel 1.2 Algorithmen, die nicht in der Lage sind, mit großen Datensätzen umzugehen, deren Aufwand mehr als exponentiell in der Größe des Datensatzes ansteigt oder die nicht garantieren können, dass eine Lösung gefunden wird, als unzureichend. Algorithmen sollen demnach außerdem robust sein in dem Sinne, dass sie mit Daten sinnvoll umgehen können, die nicht direkt aus der zugrundeliegenden Verteilung stammen, sondern in irgendeiner Weise (beispielsweise durch Mess- oder Erhebungsfehler) überlagert sind, ohne sich dabei zu stark von der eigentlich zu lernenden Verteilung zu entfernen. Darüber hinaus soll das Verfahren insofern stabil sein, als es bei einer weiteren Stichprobe aus der zugrundeliegenden Verteilung ein ähnliches Ergebnis liefern soll.[34]

[29] Empirisch lassen sich natürlich Untersuchungen zu dieser Frage anstellen. Support Vector Machines schneiden dabei gemeinsam mit sogenannten Random Forests, siehe Breiman (2001) und Athey, Tibshirani & Wager (2019), meist sehr gut im Vergleich zu anderen Methoden ab, siehe Caruana & Niculescu-Mizil (2006), Kotsiantis (2007), Caruana, Karampatziakis & Yessenalina (2008), Fernández-Delgado, Cernadas, Barro & Amorim (2014) und Wainberg, Alipanahi & Frey (2016).

[30] Zur Forschung an Oracle-Ungleichungen und Lernraten von regularisierten Ansätzen mit Lipschitz-stetigen Verlustfunktionen, aber unter zusätzlichen Voraussetzungen, siehe jüngst Alquier, Cottet & Lecué (2019).

[31] So beispielsweise Bin Yu im Rahmen eines Keynote-Vortrages bei den Stochastik-Tagen 2018 in Freiburg.

[32] Support Vector Machines weisen im Vergleich zu anderen Methoden eine geringe Interpretierbarkeit auf. Dieser Aspekt wird in dieser Arbeit aber nicht weiter betrachtet.

[33] Bei SVMs geschieht dies in der Regel – wie beschrieben – im Sinne der Risiko-Konsistenz, d. h. der Minimierung des mittleren Verlusts. Soll ein Klassifikationsproblem gelöst werden, würde die mittlere Missklassifikationsrate minimiert; die Verfahren stellen in der Regel auf die Genauigkeit (accuracy) ab. Andere Gütemaße, siehe beispielsweise Pepe (2004), wären aber ebenfalls denkbar oder sogar angezeigt (z. B. in *imbalanced-data*-Situationen), z. B. geeignet gewählte Mittel aus Sensitivität und Spezifität oder das sogenannte F-Maß.

[34] Einen Zusammenhang zwischen *learnability*, Stabilität und (gleichmäßiger) Generalisierbarkeit stellen die Aufsätze von Bousquet & Elisseeff (2002) und Shalev-Shwartz, Shamir, Srebro & Sridharan (2010) her. Der Begriff der Stabilität vergleicht dort Eigenschaften der Prädiktoren basierend auf einer Stichprobe bzw. auf dieser Stichprobe weniger einem Punkt. Die Robust-

heitsbegriffe in der vorliegenden Arbeit sind insofern allgemeiner als dieses Konzept, als sie Prädiktoren basierend auf verschiedenen (ggf. empirischen) Maßen betrachten.

Kapitel 2

Große Datenmengen und lokales Lernen

2.1 Problembeschreibung

Während Support Vector Machines die Eigenschaften der Konsistenz (im Sinne der Risiko-Konsistenz) und der Robustheit (in geeignetem Sinne) gut erfüllen, weisen sie Schwierigkeiten im Bereich der Berechenbarkeit, der *computability*, auf. In einer theorienahen Implementierung benötigen sie eine Rechenzeit in der Größenordnung von $\mathcal{O}(n^3)$ und Speicher in der Größenordnung von $\mathcal{O}(n^2)$.

Es sind verschiedene Ansätze verfügbar, um dieses Problem der Berechenbarkeit oder Skalierbarkeit zu lösen. Einige werden im Folgenden explizit genannt. Hierbei ist n die Anzahl an Beobachtungen, d die Anzahl der erklärenden Variablen (features; input variables), die zum Lernen herangezogen werden.[35]

1. *Feature selection*, um d zu reduzieren. Einen allgemeinen Überblick hierzu liefern Guyon & Elisseeff (2003) und die dort aufgeführten Referenzen. Frühe Ansätze zur feature selection für SVMs liefern unter anderem Hermes & Buhmann (2000), Weston, Mukherjee, Chapelle, Pontil, Poggio & Vapnik (2001) und Claeskens, Croux & Kerckhoven (2008). Einen aktuellen Überblick und weitere theoretische Untersuchungen zu dieser Herangehensweise liefert Zhang, Wu, Wang & Li (2016).

2. *Low-rank approximations* der Kernmatrix, um n und d zu reduzieren (basierend auf der Idee, dass geeignet gewählte Teilmengen der Stichprobe bereits ausreichend Information enthalten) und Approximationen des Kerns selbst. Hierzu stehen viele mögliche Wege offen, beispielsweise Singulärwertzerlegung,

[35] Dieser Abschnitt basiert auf dem eingereichten, aber noch nicht erschienenen Aufsatz Dumpert (2019a).

CUR-Matrix-Zerlegung oder verschiedene Nyström-Methoden. Bach (2013) und Si, Hsieh & Dhillon (2017) stellen (neben den eigenen Resultaten) Übersichten über relevante Arbeiten in diesen Feldern bereit.

3. *Sequential learning* oder *online learning*, um n pro Zeiteinheit zu reduzieren, vgl. z. B. Smale & Yao (2006), Ying & Zhou (2006), Ying & Pontil (2008) und Guo, Ying & Zhou (2017). In diesem Fall sind die Daten zu Beginn des Lernens nicht vollständig verfügbar oder werden zumindest nicht vollständig genutzt. Das zu lernende Modell wird also immer weiter fortgeschrieben, indem weitere Daten berücksichtigt werden. Ein so gelernter Prädiktor wird immer wieder aktualisiert, wenn neue Datenpunkte verfügbar sind oder Berücksichtigung finden sollen.

4. *Distributed learning*, um n pro CPU/GPU zu reduzieren (wobei in Summe der gesamte Datenbestand genutzt wird), siehe beispielsweise Christmann, Steinwart & Hubert (2007), Duchi, Jordan, Wainwright & Zhang (2014), Lin, Guo & Zhou (2017), Mücke (2017a) und Guo, Lin & Zhou (2017). Der große Vorteil dieses Ansatzes besteht in der hohen Skalierbarkeit in dem Sinne, dass immer weitere Prozessoren herangezogen werden können, um Prädiktoren auf den Teilstichproben zu berechnen. Es ist jedoch denkbar, dass Strukturen, die (nur) in verschiedenen Bereichen des Datensatzes vorhanden sind, nicht erhalten bleiben oder nicht erkannt werden.

5. *Local learning* in der Spezifikation, dass immer dann, wenn eine Vorhersage für einen neuen Datenpunkt benötigt wird, nur das lokal um diesen neuen Datenpunkt vorhandene Trainingsmaterial zum Lernen eines lokalen Modells herangezogen wird, vgl. beispielsweise Zakai & Ritov (2009), Blanzieri & Bryl (2007), Blanzieri & Melgani (2008) oder Hable (2013). Dieser Ansatz benötigt somit keine Trainingszeit auf dem gesamten Datensatz, aber immer dann ein wenig Trainingszeit, wenn ein neuer Datenpunkt eine Vorhersage benötigt. Gibt es also (prospektiv) wenige neue Daten, für die eine Vorhersage zu treffen ist, erscheint dieses Vorgehen sehr vorteilhaft.

6. *Local learning* in der Spezifikation, dass der gesamte Eingaberaum auf Basis der Trainingsdaten vor dem Lernen des eigentlichen Prädiktors in Regionen aufgeteilt wird. Wird eine Prädiktion für einen neuen Datenpunkt benötigt, hängt diese lediglich von Prädiktoren ab, die auf den Regionen gelernt wurden, zu welchen der neue Datenpunkt gehört. Diese Herangehensweise wird für Support Vector Machines in dieser Arbeit näher untersucht.

Natürlich sind Kombinationen oder die sukzessive Anwendung dieser Ansätze ebenfalls denkbar, siehe beispielsweise Mücke (2017b). Es gilt außerdem zu beachten, dass

die sechs aufgeführten Klassen von Lösungen nicht alle Möglichkeiten umfassen, mit dem Problem der Berechenbarkeit (oder Skalierbarkeit) umzugehen. Andere Ansätze sind z. B. durch *gradient descent with early stopping regularization* oder *iterative regularization* gegeben, siehe Guo, Hu & Shi (2018), Lin, Rosasco & Zhou (2016) und die dort genannten Referenzen. Einen (bis dahin) zusammenfassenden Überblick liefert García-Pedrajas & de Haro-García (2012); speziell auf Implementierungen von Support Vector Machines zugeschnitten ist die Studie von Horn, Demircioğlu, Bischl, Glasmachers & Weihs (2018).[36]

2.2 Zerlegung des Datenraumes mittels eines Baumes für SVMs auf großen Datensätzen

Die Idee für die vorliegende Arbeit lieferte der Artikel *Tree Decomposition for Large-Scale SVM Problems* (Chang, Guo, Lin & Lu, 2010). Darin beschreiben die Autoren die Problematik, dass das Lernen von Support Vector Machines im Hinblick auf die Laufzeit und den verfügbaren Arbeitsspeicher aufwändig ist. Große Datensätze führen hier schnell zu unüberwindbaren Problemen bei gegebener Ausstattung. Die Autoren des Artikels schlagen vor, den Datenraum mittels eines Entscheidungsbaumes zu zerlegen und SVMs auf den dadurch entstehenden Regionen separat zu lernen. Dabei soll die Anzahl der Regionen so gewählt werden, dass die Anzahl der pro Region vorhandenen Datenpunkte zum Trainieren der SVMs mit der zur Verfügung stehenden Ausstattung an Rechnern handhabbar ist. Die Autoren nennen bereits einen weiteren Aspekt: Die Wahl der Hyperparameter kann nun pro Region erfolgen. Als dritter Vorzug wird die Möglichkeit benannt, eine obere Fehlerschranke für den Klassifikationsfehler anzugeben.

Herangezogen wird ein Entscheidungsbaum, der achsenparallele Aufteilungen des Eingaberaums vornimmt. Im Unterschied zu anderen Varianten von Entscheidungsbäumen ist die achsenparallele Aufteilung diejenige mit dem geringsten Rechenaufwand. Als Vorzug einer Aufteilung des Datenraums nennen die Autoren:

(i) Die so gefundenen Regionen sind homogener als der Gesamtraum, enthalten also hauptsächlich Datenpunkte, die der gleichen Klasse zugehörig sind. Ist eine Region bereits hinreichend rein, so braucht keine SVM mehr gelernt zu werden; neue Datenpunkte, die in diese Region fallen, werden anhand des Entscheidungsbaumes klassifiziert. Nur für heterogene Regionen, also solche mit nennenswerten Anteilen verschiedener Klassenzugehörigkeiten, wird ein aufwändigerer Klassifikator benötigt und daher eine SVM gelernt.

[36] Erstaunlicherweise enthält diese Studie (Horn, Demircioğlu, Bischl, Glasmachers & Weihs, 2018) nicht das für diese Arbeit eingesetzte R-Paket `liquidSVM` (Steinwart & Thomann, 2017).

(ii) Die Größe der Regionen, d. h. die Anzahl der darin enthaltenen Trainingsdatenpunkte, kann leicht kontrolliert werden, beispielsweise in Form eines zusätzlichen Hyperparameters.

(iii) Derselbe Datenraum kann schließlich auch auf Basis mehrerer Bäume in immer wieder unterschiedlicher Weise aufgeteilt werden, z. B. dadurch, dass die Splits nicht mehr gemäß eines Optimalitätskriteriums, sondern zufällig gesetzt werden. SVMs können dann auf allen gebildeten Regionen gelernt werden; zur Klassifikation eines Datenpunktes würde dann ggf. ein Mehrheitsentscheid durchgeführt werden.[37]

Zum Zwecke der Aufteilung des Datenbestandes (der Prozess, der in der vorliegenden Arbeit als *Regionalisierung* bezeichnet wird) wird hier ein binärer Entscheidungsbaum basierend auf der Entropie als Unreinheitsmaß verwendet. Als Abbruchkriterien kommen zwei Situationen infrage: (i) In einem Knoten befinden sich weniger Datenpunkte als durch einen Parameter (Mindestanzahl) vorgegeben. In diesem Fall wird nicht weiter aufgeteilt. Diese Mindestanzahl wird im Algorithmus datenabhängig gewählt. (ii) Es wird (überhaupt) kein Zuwachs an Reinheit mehr erlangt, egal wie der Datensatz weiter aufgeteilt würde.

Der beschriebene Algorithmus geht nun schrittweise vor. In einem ersten Schritt wird mit von vorneherein festgelegter Mindestanzahl σ_0 ein Entscheidungsbaum gelernt und lokal, d. h. in jeder Region, geprüft, ob sie homogen ist. Ist das der Fall, wird eine solche Region automatisch mit einer Klasse assoziiert. Falls nicht, werden für ein vorgegebenes Gitter von Hyperparametern SVMs in jeder Region gelernt und anhand eines Validierungsdatensatzes validiert. Das Verfahren wird anschließend für Mindestanzahlen größer σ_0 (z. B. $4\sigma_0$) wiederholt. Dabei werden jedoch nicht mehr alle ursprünglich verwendeten Kombinationen von Hyperparametern für die SVM genutzt, sondern nur noch solche, die sich auf den Regionen der Mindestgröße σ_0 als am besten geeignet herausgestellt haben (also nur die k Kombinationen je Region, die bei der Validierung am besten abgeschnitten haben). Hierbei wiederum werden nicht in jedem Schritt neue Bäume gelernt, sondern der bestehende (gewissermaßen maximale) Baum wird immer weiter zurückgeschnitten, bis die jeweilige Mindestgröße der Blätter erreicht ist. Dieses Vorgehen wird so lange wiederholt, bis keine hinreichende Verbesserung der Klassifikationsgenauigkeit mehr erreicht wird (oder die Größe des Trainingsdatensatzes selbst erreicht wurde). Die Hyperparameter (Mindestgröße und Hyperparameter der SVM) des insgesamt besten erreichten Ergebnisses auf dem Validierungsdatensatz bilden schließlich die Hyperparameter des Prädiktors. σ_0 wird im Artikel auf Basis von durchgeführten Simulationen auf

[37] Die Autoren des Artikels bemerken jedoch, dass die Ausnutzung dieser Variante zu keiner Verbesserung der Klassifikationsgenauigkeit führt.

1500 gesetzt, k auf 5. Die Simulationen zeigen, dass eine Veränderung der initialen Mindestgröße σ_0 keine bedeutenden Veränderungen in der Klassifikationsgenauigkeit hervorruft; verschiedene Werte für k spielen jedoch eine Rolle (zu kleine k sind ungünstig).

Untersucht wird der Algorithmus im Wesentlichen experimentell anhand von Beispieldatensätzen. Insbesondere bei großen Datensätzen (4,9 Mio. Beobachtungen; 16,6 Mio. Merkmale) zeigt sich die Überlegenheit dieser Methode gegenüber anderen Implementierungen wie beispielsweise LIBSVM bezüglich Laufzeit, resultierenden Support-Vektoren und Klassifikationsgenauigkeit. Allerdings wird auch eine Abschätzung für das Risiko, also den erwarteten Klassifikationsfehler, theoretisch hergeleitet. Unterschieden wird dabei noch nach der Eigenschaft des Trainingsdatensatzes, vollständig linear separierbar zu sein (hard margin) oder (prinzipiell oder aufgrund von Rauschen) nicht (soft margin). Diese Unterscheidung wird im Laufe dieser Arbeit nicht weiter getroffen, weshalb auch hier nur das allgemeinere Resultat (soft margin) zitiert wird.

Sei $\mathcal{D}_n := ((x_1, y_1), \ldots, (x_n, y_n))$ eine Zufallsstichprobe (der Trainingsdatensatz) und π ein binärer Baum (als Regionalisierungsmethode) auf $\mathbb{R}^d$, der $\mathcal{D}_n$ und damit $\mathbb{R}^d$ (genauer: $\mathbb{R}^d \times \{-1, +1\}$) in B Regionen aufteilt. In Region 1 seien dann n_1 Trainingsdatenpunkte, ..., in Region B seien dann n_B Trainingsdatenpunkte. Seien $f_1, \ldots, f_B$ lineare Funktionen von H nach $\mathbb{R}$ mit $\|f_b\| \leq \beta_b$ für $\beta_b > 0$ für alle $b \in \{1, \ldots, B\}$, wobei H ein Hilbertraum (der *feature space*) ist. Sei dann $\xi_{b,j} := \max\{0, \gamma_b - y_{b,j} f_b(\Phi(x_{b,j}))\}$ die Schlupfvariable von f_b zur Spanne (margin) $\gamma_b > 0$ zu $(x_{b,j}, y_{b,j})$, also zum j-ten Trainingsdatenpunkt in Regionen b, $b \in \{1, \ldots, B\}$, $j \in \{1, \ldots, n_b\}$. Φ steht hier für eine Abbildung von $\mathbb{R}^d$ in den Hilbertraum H, die sogenannte *feature map*. Es bezeichne weiter $\xi_b := (\xi_{b,1}, \ldots, \xi_{b,n_b})$ den Schlupfvektor von f_b bezüglich π und γ_b über $\mathcal{D}_n$, $b \in \{1, \ldots, B\}$.

Proposition 2.1 (Theorem 10 aus Chang, Guo, Lin & Lu (2010))
Sei $d \in \mathbb{N}$, P eine Wahrscheinlichkeitsverteilung auf $\left(\mathbb{R}^d \times \{-1, +1\}, \mathfrak{B}_{\mathbb{R}^d \times \{-1,+1\}}\right)$, $n \in \mathbb{N}$ hinreichend groß, $\mathcal{D}_n := ((x_1, y_1), \ldots, (x_n, y_n))$ eine Zufallsstichprobe (unabhängig und identisch verteilt gemäß P). Es gelte für ein $\rho > 0$, dass $\|\Phi(x)\| \leq \rho$ für alle $x \in \mathbb{R}^d$. Dann beträgt mit oben eingeführter Notation und mit Wahrscheinlichkeit $1 - \delta$ das theoretische Risiko (also die erwartete Missklassifikationsrate bzgl. P) auf Basis des Datensatzes $\mathcal{D}_n$ höchstens

$$\frac{c}{n}\left(\sum_{b=1}^{B}\left(\frac{\rho^2\beta_b^2 + \|\xi_b\|^2}{\gamma_b^2}\right)\log^2 n + B\log(dnB^2) + \log\left(\frac{1}{\delta}\right)\right) \tag{2.1}$$

für eine Konstante $c > 0$.

Die Autoren des Artikels kommentieren dieses Theorem wie folgt: Findet man einen Prädiktor auf Basis weniger Regionen (kleines B) und mit wenig Schlupf innerhalb dieser Regionen (kleine $\|\xi_b\|$), so sind die ersten beiden Summanden klein. Das bedeutet nicht, dass ein kleines B deswegen zwingend vorzuziehen ist, denn es besteht in der Regel ein Zielkonflikt zwischen B und $\|\xi_b\|$, der datenabhängig gelöst werden sollte. Verglichen mit der Situation von nur einer Region ($B = 1$) zeigt sich eine leichte Verschlechterung. Gemäß Cristianini & Shawe-Taylor (2000) betrüge das theoretische Risiko hier

$$\frac{c}{n}\left(\left(\frac{\rho^2\beta^2 + \|\xi\|^2}{\gamma^2}\right)\log^2 n + \log\left(\frac{1}{\delta}\right)\right).$$

Der Term $B\log(dnB^2)$, also $\log(dn)$ wegen $B = 1$, tritt in der Situation ohne Aufteilung des Eingaberaums nicht auf. Die numerischen Untersuchungen der Autoren zeigen jedoch, dass $B\log(dnB^2)$ in (2.1) durch den ersten Summanden dominiert wird, also keinen wesentlichen Bestandteil der oberen Schranke ausmacht. $B\log(dnB^2)$ ergibt sich aus dem sogenannten Shatter-Koeffizienten[38] der Aufteilung durch den Baum (Chang, Guo, Lin & Lu, 2010, Lemmata 4 und 6).

Der eben besprochene Artikel beschreibt einen Ansatz, um SVMs auch auf großen Datensätzen handhabbar zu machen. Betrachtet wird dabei nur (Multiclass-) Klassifikation (sowohl im one-vs-one- als auch im one-vs-all-Ansatz; stets und ausschließlich mit der hinge-Verlustfunktion), also die Fragestellung, die auch zu Beginn der Arbeiten Vapniks und dessen Co-Autoren stand. Regression wird nicht betrachtet. Bemerkenswert ist der Umstand, dass die Fehlerabschätzung explizit einen Term enthält, der durch die Regionalisierungsmethode zustande kommt.

2.3 Lokales Lernen

Die Idee, lokal zu lernen, ist (auch für SVMs) nicht neu. Frühe theoretische Überlegungen finden sich bereits in Bottou & Vapnik (1992) und Vapnik & Bottou (1993); einen Überblick über verschiedene Arten lokalisierten Lernens (mit anschließendem Kombinieren der lokalen Prädiktoren) liefert Collobert, Bengio & Bengio (2002). Im Rahmen eigener Untersuchungen führte Hable (2013) aus, dass die prinzipielle Notwendigkeit besteht, lokalisierte Ansätze im Bereich von SVMs und anderen kernbasierten Methoden zu untersuchen. Die schiere Menge an Daten, die heutzutage verfügbar ist und genutzt werden soll, stellt eine Herausforderung für diese Algorithmen in Bezug auf Laufzeit und (Arbeits-)Speicher dar. Lokale Ansätze bieten die Möglichkeit, diese Probleme zumindest abzumildern. Solche Ansätze wurden auch vorgeschlagen, siehe beispielsweise Bennett & Blue (1998), Wu, Bennett, Cristianini

[38] Vgl. hierzu beispielsweise Shalev-Shwartz & Ben-David (2014).

& Shawe-Taylor (1999) und Chang, Guo, Lin & Lu (2010), die hierzu Entscheidungsbäume verwenden. Dichtebasierte Zerlegungen des Eingaberaums werden von Rida, Labbi & Pellegrini (1999) propagiert. k-nearest neighbor (KNN) wurde beispielsweise von Zhang, Berg, Maire & Malik (2006), Blanzieri & Bryl (2007), Blanzieri & Melgani (2008) und Segata & Blanzieri (2010) vorgeschlagen; Cheng, Tan & Jin (2007), Cheng, Tan & Jin (2010) und Gu & Han (2013) nutzen KNN-Clustering-Methoden. Lokales Lernen ermöglicht unmittelbar die Parallelisierung der Berechnungen, was neben kleineren lokalen Datensätzen einen weiteren Grund für die zu erwartende Verbesserung der Laufzeiten darstellt.

Aus statistischer Sicht gibt es eine weitere Motivation, einen genaueren Blick auf lokale Ansätze zu werfen. Verschiedene Bereiche des Raumes $\mathcal{X} \times \mathcal{Y}$ haben gegebenenfalls verschiedene Anforderungen an die verwendete statistische Methode. Beispielsweise könnte es eine Region geben, die nur eine sehr einfache Funktion als Prädiktor benötigt; eine andere Region hingegen benötigt vielleicht eine sehr volatile Funktion, um die Grenze zwischen den Klassen oder gute Schätzungen im Rahmen einer Regression abbilden zu können. Statistische Methoden, die alle Datenpunkte berücksichtigen (also global lernen), bestimmen auch die dann optimalen Hyperparameter (z. B. die Bandbreite eines Kernes oder den Regularisierungsparameter λ) global. Diese Parameter haben Einfluss auf die Komplexität des Prädiktors, werden dem Datensatz in seiner lokal stark unterschiedlichen Struktur aber möglicherweise (bei fester Stichprobengröße) nicht gerecht; die lokalen Besonderheiten werden gegebenenfalls „ausgemittelt“, um ein global optimales Ergebnis zu erreichen. Lokales Lernen erlaubt die Verwendung verschiedener Hyperparameter (und sogar die Verwendung gänzlich verschiedener Kerne) in den verschiedenen Regionen. Um eben diesen statistischen Effekt zu erfassen, gibt es wenigstens zwei mögliche Ansätze.

(i) Der erste Ansatz wurde von Hable (2013) aus statistischer Sicht, von Blanzieri & Melgani (2008) numerisch untersucht. Die Prädiktion von y gegeben einen neuen Eingabedatenpunkt $x \in \mathcal{X}$ wird hierbei wie folgt gelernt: Um diesen neuen Datenpunkt wird eine Umgebung festgelegt (z. B. eine Kugel (Zakai & Ritov, 2009) oder durch Bestimmung der k nächsten Nachbarn im Trainingsdatensatz) und der Prädiktor auf Basis der Trainingsdatenpunkte innerhalb dieser Umgebung gelernt. Anschließend wird der Prädiktor für den neuen Datenpunkt ausgewertet, um eine Vorhersage für y zu erhalten.

(ii) Der zweite Ansatz besteht darin, den Eingaberaum anhand des Trainingsdatensatzes (also unberührt von neuen Datenpunkten) in (sich ggf. überlappende) Regionen aufzuteilen und lokale Prädiktoren zu lernen. Um eine Vorhersage für y eines neuen Datenpunktes x zu erhalten, werden die lokalen Prädiktoren, in deren Regionen der neue Datenpunkt liegt, eingesetzt.

Diese Arbeit umfasst Untersuchungen zu den statistischen Eigenschaften des zweiten Ansatzes im Falle von SVMs.

Die Tatsache, dass es notwendigerweise möglich ist, eine konsistente Methode zu lokalisieren – denn Konsistenz betrifft lokale Aspekte –, wird in Zakai & Ritov (2009) gezeigt. Es gibt weitere Arbeiten in diesem Bereich, die beispielsweise optimale Lernraten (und somit auch Konsistenz) zeigen und dabei Aufteilungen wie Voronoi-Partitionen (Aurenhammer, 1991), die Kleinste-Quadrate-Verlustfunktion oder die hinge-Verlustfunktion sowie einen Gaußkern verwenden und außerdem Annahmen an die Bayes-Entscheidungsfunktion und die zugrundeliegende Verteilung treffen (müssen), vgl. Eberts (2015), Meister & Steinwart (2016) und Thomann, Blaschzyk, Meister & Steinwart (2017). Diese Arbeit hingegen erlaubt allgemeine Regionalisierungsmethoden, überlappende Regionen, allgemeine Kerne und allgemeine (Lipschitz-stetige und konvexe) Verlustfunktionen und setzt nichts bezüglich der zugrundeliegenden Verteilung P voraus, was nicht überprüfbar wäre. Lernraten hingegen können aufgrund dieser Allgemeinheit – wie früher bereits beschrieben – nicht geliefert werden.[39]

[39] Ein nicht identischer, aber im Hinblick auf die Zielsetzung der Lokalisierung durchaus verwandter Ansatz ist bei den sogenannten lokalen Kleinste-Quadrate-Regressionen (Cleveland & Devlin, 1988; Ruppert & Wand, 1994) zu finden.

Kapitel 3

Konkretisierung der Regionalisierung

Die Regionalisierungsmethode, d. h. die Methode, die für die Bildung der Regionen zuständig ist, ist für die Resultate in dieser Arbeit beliebig wählbar, solange sie (je nach Resultat) mehrere der folgenden Eigenschaften aufweist.

(R1) Die Regionalisierungsmethode teilt den Eingaberaum $\mathcal{X}$ in (sich möglicherweise überlappende) Regionen auf, d. h. $\mathcal{X} = \bigcup_{b=1}^{B_n} \mathcal{X}_{(n,b)}$ oder $\mathcal{X} \times \mathcal{Y} = \bigcup_{b=1}^{B_n} \left(\mathcal{X}_{(n,b)} \times \mathcal{Y}\right)$. B_n ist die Anzahl der Regionen, die vom Anwender oder der Regionalisierungsmethode selbst gewählt wird und daher von zumindest einer Unterstichprobe des Traniningsdatensatzes abhängen kann. Für alle Schritte nach der Regionalisierung ist $B := B_n$ konstant, d. h. dann gilt $\mathcal{X} = \bigcup_{b=1}^{B} \mathcal{X}_b$ oder $\mathcal{X} \times \mathcal{Y} = \bigcup_{b=1}^{B} (\mathcal{X}_b \times \mathcal{Y})$.

(R2) Für alle $b \in \{1, \ldots, B\}$ ist $\mathcal{X}_b$ ein separabler metrischer Raum. (Diese Bedingung ist stets erfüllt, da Teilmengen separabler Mengen separabel und Teilmengen metrischer Räume metrische Räume sind, siehe Dunford & Schwartz (1958, I.6.4, I.6.12).) Zusätzlich wird gefordert, dass alle $\mathcal{X}_b$ vollständig messbare Räume sind, d. h. bezüglich aller Wahrscheinlichkeitsmaße ist $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b \times \mathcal{Y}})$ vollständig, wobei sich dieser Begriff von Vollständigkeit auf die Messbarkeit von Nullmengen bezieht, vgl. Ash & Doleans-Dade (2000, Definition 1.3.7).

(R3) Für $n \to \infty$ stellt die Regionalisierungsmethode sicher, dass $|\mathcal{D}_n \cap (\mathcal{X}_b \times \mathcal{Y})| \to \infty$ für alle $b \in \{1, \ldots, B\}$, d. h. $\lim_{n\to\infty} \min_{b\in\{1,\ldots,B\}} |\mathcal{D}_n \cap (\mathcal{X}_b \times \mathcal{Y})| = \infty$, wobei $|M|$ wiederum die Anzahl der Elemente einer Menge M darstellt.

(R4) Jede Region $\mathcal{X}_b$ ist vollständig, $b \in \{1, \ldots, B\}$, in dem Sinne, dass jede Cauchyfolge in $\mathcal{X}_b$ einen Grenzwert in $\mathcal{X}_b$ besitzt. (Dies ist garantiert, wenn man stets die jeweilige Vervollständigung einer durch die Regionalisierungsmethode gebildeten Region betrachtet; das ist möglich, da die Regionalisierung nicht disjunkt, also keine Partition sein muss.)

In einer Situation, in der der gesamte Eingaberaum $\mathcal{X}$ durch die Regionalisierungsmethode in (nicht notwendigerweise disjunkte) Regionen $\mathcal{X}_1, \ldots, \mathcal{X}_B$ aufgeteilt wurde, soll nun pro Region eine SVM gelernt werden. Diese lokal gelernten SVMs werden anschließend zu einem zusammengesetzten Prädiktor (Schätzer, Klassifizierer) zusammengesetzt. Der Einfluss der lokalen SVMs kann dabei punktweise über messbare Gewichtsfunktionen $w_b : \mathcal{X} \to [0,1], b \in \{1, \ldots, B\}$, gesteuert werden. Die Gewichtsfunktionen müssen dabei die folgenden beiden (für Gewichtsfunktionen üblichen) Bedingungen erfüllen: **(W1)** $\sum_{b=1}^{B} w_b(x) = 1$ für alle $x \in \mathcal{X}$ und **(W2)** $w_b(x) = 0$ für alle $x \notin \mathcal{X}_b$ und für alle $b \in \{1, \ldots, B\}$.

Die Arbeit folgt der bereits in Dumpert & Christmann (2018) verwendeten Notation und definiert die zusammengesetzten Prädiktoren wie folgt:

$$f^{comp}_{L^*,P,\lambda} : \mathcal{X} \to \mathbb{R}, \qquad f^{comp}_{L^*,P,\lambda}(x) := \sum_{b=1}^{B} w_b(x) f_{b,L^*,P_b,\lambda_b}(x), \tag{3.1}$$

$$f^{comp}_{L^*,D_n,\lambda} : \mathcal{X} \to \mathbb{R}, \qquad f^{comp}_{L^*,D_n,\lambda}(x) := \sum_{b=1}^{B} w_b(x) f_{b,L^*,D_{n,b},\lambda_b}(x), \tag{3.2}$$

wobei gilt:

- P ist die unbekannte Verteilung von (X,Y) auf $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$ und $D_n := n^{-1} \sum_{i=1}^{n} \delta_{(x_i,y_i)}$ ist die empirische Verteilung basierend auf einer Stichprobe oder einem anderweitig erzeugten Datensatz $\mathcal{D}_n := ((x_1,y_1), \ldots, (x_n,y_n))$ von n unabhängigen, identisch verteilten Realisationen von (X,Y).

- P_b ist die theoretische Verteilung auf $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b\times\mathcal{Y}})$, $D_{n,b}$ ihr empirisches Analogon. Beide sind in allen relevanten Fällen Wahrscheinlichkeitsverteilungen, d. h. falls $P(\mathcal{X}_b \times \mathcal{Y}) > 0$ beziehungsweise $D_n(\mathcal{X}_b \times \mathcal{Y}) > 0$, da sie aus P beziehungsweise D_n wie folgt gebildet werden:

$$P_b := \begin{cases} P(\mathcal{X}_b \times \mathcal{Y})^{-1} \, P_{|\mathcal{X}_b\times\mathcal{Y}} & , \text{ falls } \; P(\mathcal{X}_b \times \mathcal{Y}) > 0 \\ 0 & , \text{ sonst} \end{cases}$$

und

$$D_{n,b} := \begin{cases} D_n(\mathcal{X}_b \times \mathcal{Y})^{-1} \, D_{n|\mathcal{X}_b\times\mathcal{Y}} & , \text{ falls } \; D_n(\mathcal{X}_b \times \mathcal{Y}) > 0 \\ 0 & , \text{ sonst} \end{cases}.$$

Es ist also $D_n(\mathcal{X}_b \times \mathcal{Y}) = |\mathcal{D}_{n,b}| =: n_b$.

- Analog wird die regionale Randverteilung von X mit $P_b^{\mathcal{X}_b} := P^{\mathcal{X}}(\mathcal{X}_b)^{-1} P^{\mathcal{X}}_{|\mathcal{X}_b}$, falls $P^{\mathcal{X}}(\mathcal{X}_b) > 0$ und 0 sonst dargestellt.

- $\lambda := (\lambda_1, \ldots, \lambda_B) \subset]0, \infty[^B$ oder, falls die Anzahl der Datenpunkte ausgewiesen werden soll, $\lambda_n := \left(\lambda_{(n_1,1)}, \ldots, \lambda_{(n_B,B)}\right)$, $n = \sum_{b=1}^{B} n_b$, anstelle eines festen λ.

- Mit f_{b,L^*,P_b,λ_b} wird die theoretische lokale SVM auf $\mathcal{X}_b \times \mathcal{Y}$ bezüglich L^* und P_b bezeichnet, sofern P_b ein Wahrscheinlichkeitsmaß ist; falls P_b das Nullmaß ist, so stellt f_{b,L^*,P_b,λ_b} eine beliebige messbare Funktion (von $\mathcal{X}$ nach $\mathbb{R}$) dar. Mit $f_{b,L^*,D_{n,b},\lambda_b}$ wird die auf $\mathcal{X}_b \times \mathcal{Y}$ gelernte empirische lokale SVM bezüglich L^* und $D_{n,b}$ bezeichnet, sofern $D_{n,b}$ ein Wahrscheinlichkeitsmaß ist; ist $D_{n,b}$ das Nullmaß, so ist $f_{b,L^*,D_{n,b},\lambda_b}$ eine beliebige messbare Funktion (von $\mathcal{X}$ nach $\mathbb{R}$).

Es gilt zu beachten, dass die so gebildeten Prädiktoren im Falle von sich überlappenden Regionen $\mathcal{X}_b$ im Allgemeinen nicht mehr Elemente eines Hilbertraumes oder gar eines RKHS sind.[40] Der aus der Theorie der Support Vector Machines bekannte Ausdruck $\|f_{L^*,P,\lambda}\|_H$ ergibt somit für die zusammengesetzten Prädiktoren $f^{comp}_{L^*,P,\lambda}$ keinen Sinn.

Während durch die Verwendung eines beschränkten, stetigen Kerns im Falle der globalen Support Vector Machine sichergestellt wird, dass der zugehörige RKHS aus stetigen und beschränkten Funktionen besteht (Steinwart & Christmann, 2008, Lemma 4.28)[41] bzw. durch Verwendung eines m-fach differenzierbaren Kerns (in geeignetem Sinne) sichergestellt werden kann, dass jede Funktion im zugehörigen RKHS m-fach differenzierbar ist (Steinwart & Christmann, 2008, Corollar 4.36), ist dies im Fall zusammengesetzter Prädiktoren insbesondere aufgrund der verwendeten Gewichte fraglich. Eine naheliegende Wahl für die Gewichte sind auf Basis der Anforderungen **(W1)** und **(W2)** Indikatorfunktionen der jeweiligen Mengen, d. h.

$$w_b(x) = \frac{\mathbf{1}_{\mathcal{X}_b}(x)}{\sum_{\beta=1}^{B} \mathbf{1}_{\mathcal{X}_\beta}(x)}, \quad b \in \{1, \ldots, B\},\ x \in \mathcal{X}.$$

Sie garantieren, dass eine lokale SVM nur für Datenpunkte in derjenigen Region Einfluss auf den zusammengesetzten Prädiktor nimmt, in der sie auch gelernt wurde. Der offensichtliche Nachteil ist jedoch der Verlust der Stetigkeit (und somit natürlich auch der Differenzierbarkeit) des zusammengesetzten Prädiktors. Einen Ausweg bietet der Einsatz gegebenenfalls sogar beliebig glatter Abschneidefunktionen (mit

[40] Falls die Regionen disjunkt sind, ist eine solche Konstruktion möglich. Dieser Fall ist für die vorliegende Arbeit aber nicht von Interesse.

[41] vgl. Proposition 1.2

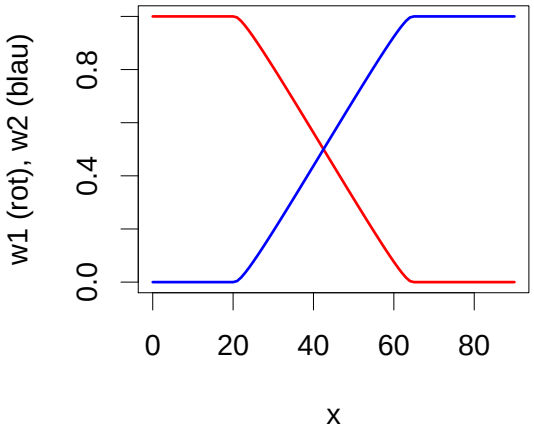


Abbildung 3.1: Gewichtsfunktionen

kompaktem Träger auf der jeweiligen Menge), welche die Indikatorfunktion beliebig genau approximieren. Dass es solche Abschneidefunktionen stets gibt, folgt aus Urysohn's Lemma, vgl. für den für die Anwendung relevanten Fall $\mathcal{X} = \mathbb{R}^d$ beispielsweise Lieb & Loss (2001, S. 4 und 38).

Zur Veranschaulichung sei $\mathcal{X} = [0, 90] \subset \mathbb{R}$ betrachtet. Seien

$$\varphi : \mathbb{R} \to \mathbb{R}, \quad \varphi(x) := \begin{cases} \exp(-x^{-1}) & , \text{ falls } x > 0 \\ 0 & , \text{ sonst} \end{cases},$$

$\zeta : \mathbb{R}^3 \to \mathbb{R}$, $\zeta(x, b, a) := \varphi(x - b)\,\varphi(a - x)$ für $b < a$, und sei schließlich

$$\chi : \mathbb{R}^4 \to \mathbb{R}, \quad \chi(x, b, a, c) := 1 - \frac{\int_{-\infty}^{|x-c|} \zeta(t, b, a)\ dt}{\int_{-\infty}^{\infty} \zeta(t, b, a)\ dt}$$

eine Abschneidefunktion. Die auf ganz $\mathcal{X}$ definierten Funktionen

$$w_1(x) := \frac{\chi(x, 20, 65, 0)}{\chi(x, 20, 65, 0) + \chi(x, 25, 70, 90)}$$

und

$$w_2(x) := \frac{\chi(x, 25, 70, 90)}{\chi(x, 20, 65, 0) + \chi(x, 25, 70, 90)}$$

erfüllen für $\mathcal{X} = [0, 90]$, $\mathcal{X}_1 = [0, 65]$, $\mathcal{X}_2 = [20, 90]$ die Anforderungen **(W1)** und **(W2)**, denn: Für alle $x \in \mathcal{X}_1 \backslash \mathcal{X}_2 = [0, 20[$ (und auch für alle $x \in \overline{\mathcal{X}_1 \backslash \mathcal{X}_2} = [0, 20]$) ist $w_1(x) = 1$ und $w_2(x) = 0$. Ebenso gilt für alle $x \in \mathcal{X}_2 \backslash \mathcal{X}_1 =]65, 90]$ (und auch für alle $x \in \overline{\mathcal{X}_2 \backslash \mathcal{X}_1} = [65, 90]$), dass $w_1(x) = 0$ und $w_2(x) = 1$. Für alle x aus dem Schnittbereich $\mathcal{X}_{\{1,2\}} = [20, 65]$ gilt offensichtlich: $w_1(x) + w_2(x) = 1$. Abbildung 3.1

veranschaulicht die beiden beliebig glatten Gewichtsfunktionen w_1 (in rot) und w_2 (in blau).

Sollen explizit lokale Normen betrachtet werden, so werde die Supremumsnorm auf $\mathcal{X}_b$ mit $\|\cdot\|_{\mathcal{X}_b\text{-}\infty}$ bezeichnet, d. h. für eine Funktion $f : \mathcal{X} \to \mathbb{R}$ sei $\|f\|_{\mathcal{X}_b\text{-}\infty} := \sup\{|f(x)| \mid x \in \mathcal{X}_b\}$ und für einen Kern $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ sei die Supremumsnorm definiert als $\|k\|_{\mathcal{X}_b\text{-}\infty} := \sup_{x \in \mathcal{X}_b} \sqrt{k(x,x)}$.

Kapitel 4

Statistische Eigenschaften

4.1 Konsistenz

Ein gemäß (3.2) aus lokal gelernten SVMs zusammengesetzter Prädiktor ist universell risikokonsistent[42], insbesondere führt die Verwendung des Regionalisierungsansatzes (anstelle einer globalen SVM) nicht zu einem Verlust dieser Eigenschaft. Die Mindestanforderung an eine statistische Methode (vgl. Abschnitt 1.5) ist damit gegeben.

Im Folgenden wird von der Notation

$$\mathcal{X}_I := \left(\bigcap_{b \in I} \mathcal{X}_b \right) \setminus \left(\bigcup_{b \notin I} \mathcal{X}_b \right), \quad I \subset \{1, \ldots, B\} ,$$

Gebrauch gemacht; sie beschreibt die „reinen“ Schnitte.

Theorem 4.1
*Sei $\mathcal{X}$ ein separabler metrischer Raum. Sei L eine konvexe, Lipschitz-stetige Verlustfunktion mit Lipschitzkonstante $|L|_1 \neq 0$ und L^* ihre geshiftete Version. Für alle $b \in \{1, \ldots, B\}$ sei k_b ein messbarer und beschränkter Kern auf $\mathcal{X}$; die zugehörigen RKHS H_b seien separabel. Die Regionalisierungsmethode erfülle die Voraussetzungen **(R1)**, **(R2)** und **(R3)**.*

Dann gilt für alle Verteilungen P auf $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$ mit H_b dicht in $L^1(P_b^{\mathcal{X}_b}), b \in \{1, \ldots, B\}$, und alle Folgen $\left(\lambda_{(n_1,1)}, \ldots, \lambda_{(n_B,B)}\right)$ mit $\lambda_{(n_b,b)} \to 0$ und $\lambda_{(n_b,b)}^2 n_b \to \infty$ für $n_b \to \infty$, $b \in \{1, \ldots, B\}$, dass

$$\mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,D_n,\lambda_n}) \xrightarrow[n\to\infty]{} \mathcal{R}^*_{\mathcal{X},L^*,P} \quad \text{in Wahrscheinlichkeit bzgl. } P.$$

[42] Die Abschnitte zur Konsistenz entsprechen einem Großteil der Veröffentlichung Dumpert & Christmann (2018).

Bemerkung 4.2 *(a) Die Voraussetzung $|L|_1 \neq 0$ an die Lipschitz-Konstante der Verlustfunktion ist rein technisch. Verlustfunktionen mit $|L|_1 = 0$ wären konstant und somit uninteressant für statistisches Lernen.*

(b) Die Dichtheitsvoraussetzung des RKHS ist leicht zu erfüllen. Beispielsweise erfüllt der Gaußkern diese Voraussetzung automatisch, siehe Steinwart & Christmann (2008, Theorem 4.63).

(c) Die Voraussetzung, dass die RKHS separabel sind, ist einfach zu erfüllen. Die Verwendung von stetigen Kernen ist hinreichend für die Separabilität der korrespondierenden RKHS, siehe Steinwart & Christmann (2008, Lemma 4.33).[43]

Lemma 4.3
Schwierigkeiten in Bezug auf unendliche Risiken können in der Situation des obigen Theorems nicht auftreten, denn für alle $n \in \mathbb{N}$ gilt:

$$
\begin{aligned}
\left|\mathcal{R}_{\mathcal{X},L^*,P}\left(f^{comp}_{L^*,D_n,\lambda_n}\right)\right| &= \left| \int_{\mathcal{X}\times\mathcal{Y}} L^*\left(y, f^{comp}_{L^*,D_n,\lambda_n}(x)\right)\ dP(x,y) \right| \\
&= \left| \int_{\mathcal{X}\times\mathcal{Y}} L^*\left(y, \sum_{b=1}^{B} w_b(x) f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right)\ dP(x,y) \right| \\
&= \left| \int_{\mathcal{X}\times\mathcal{Y}} L\left(y, \sum_{b=1}^{B} w_b(x) f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right) - L(y,0)\ dP(x,y) \right| \\
&\leq |L|_1 \int_{\mathcal{X}\times\mathcal{Y}} \left|\sum_{b=1}^{B} w_b(x) f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
&\leq |L|_1 \sum_{I\subset\{1,\dots,B\}} \int_{\mathcal{X}_I\times\mathcal{Y}} \left|\sum_{b=1}^{B} w_b(x) f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
&\leq |L|_1 \sum_{I\subset\{1,\dots,B\}} \sum_{b=1}^{B} \int_{\mathcal{X}_I\times\mathcal{Y}} \left|w_b(x) f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
&\leq |L|_1 \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} \int_{\mathcal{X}_I\times\mathcal{Y}} \left|w_b(x) f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
&\leq |L|_1 \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} \int_{\mathcal{X}_b\times\mathcal{Y}} \left|w_b(x) f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
&\leq |L|_1 \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} P(\mathcal{X}_b\times\mathcal{Y}) \int_{\mathcal{X}_b} \left|f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x)\right|\ dP_b^{\mathcal{X}_b}(x)\ < \infty.
\end{aligned}
$$

[43] Einen allgemeinen Zusammenhang zwischen Separabilität, Feature Maps und den zugehörigen RKHS liefert Owhadi & Scovel (2017).

Die Abschätzungen gelten auch, wenn der theoretische Fall betrachtet wird, d. h. wenn man D_n durch P ersetzt.

4.2 Beweis der Konsistenz

Es erscheint für die weiteren Schritte sinnvoll, die Ungleichung aus Proposition 1.1 für Funktionen f im reproduzierenden Kern-Hilbertraum eines beschränkten Kernes k, $\|f\|_\infty \leq \|f\|_H \|k\|_\infty$, in Erinnerung zu rufen. Sie spielt insbesondere deshalb eine so tragende Rolle, weil die zusammengesetzten Prädiktoren nun nicht mehr zwingend Elemente eines Hilbertraums sein müssen und man sich daher bei Abschätzungen auf die Supremumsnorm zurückziehen muss.[44]

Für den Beweis des Konsistenztheorems 4.1 werden außerdem die im Folgenden dargelegten Lemmata benötigt. Das erste zeigt auf, dass sich das Bayes-Risiko additiv aus den gewichteten Bayes-Risiken auf Teilmengen zusammensetzt, wenn diese eine Partition bilden.

Lemma 4.4
Sei M eine messbare Menge, $s \in \mathbb{N}$ beliebig, aber fest, und sei $(T_1, \ldots, T_s)$ eine messbare Partition von M, d. h. T_i messbar, $i = 1, \ldots, s$, $T_i \cap T_j = \emptyset, i, j = 1, \ldots, s$, $i \neq j$, und $M = \bigcup_{i=1}^{s} T_i$. Seien Q ein Wahrscheinlichkeitsmaß auf $(M \times \mathcal{Y}, \mathfrak{B}_{M\times\mathcal{Y}})$ für $\mathcal{Y} \subset \mathbb{R}$ abgeschlossen, L^* eine geshiftete Verlustfunktion und $\mathcal{R}^*_{M,L^*,Q} \in \mathbb{R}$. Dann gilt:

$$\mathcal{R}^*_{M,L^*,Q} = \sum_{i=1}^{s} Q(T_i \times \mathcal{Y})\ \mathcal{R}^*_{T_i,L^*,Q_i},$$

wobei

$$Q_i := \begin{cases} \frac{1}{Q(T_i\times\mathcal{Y})}\ Q_{|_{T_i\times\mathcal{Y}}} & ,\ \textit{falls}\ \ Q(T_i \times \mathcal{Y}) > 0 \\ 0 & ,\ \textit{sonst} \end{cases}, \quad i = 1, \ldots, s.$$

Beweis. Der Beweis des Lemmas basiert auf der Tatsache, dass in jedem Schritt die gleiche Menge von Funktionen betrachtet wird und dass $(T_1, \ldots, T_s)$ eine Partition von M bildet.

[44] Aus Sicht eines Anwenders mag eine solche Abschätzung sogar mehr Aussagekraft liefern als die reine Abschätzung einer hier schwer zu interpretierenden Hilbertraum-Norm.

$$
\begin{aligned}
\mathcal{R}^*_{M,L^*,Q} &= \inf\{\ \mathcal{R}_{M,L^*,Q}(f) |\ f : M \to \mathbb{R}, f \text{ messbar }\} \\
&= \inf\left\{\ \mathcal{R}_{M,L^*,Q}(f) \,\middle|\, f = \sum_{i=1}^{s} f_i \mathbf{1}_{T_i}, f_i : M \to \mathbb{R}, f_i \text{ messbar }, i = 1, \ldots, s \right\} \\
&= \inf\left\{\ \mathcal{R}_{M,L^*,Q}\left(\sum_{i=1}^{s} f_i \mathbf{1}_{T_i}\right) \,\middle|\, f_i : M \to \mathbb{R}, f_i \text{ messbar }, i = 1, \ldots, s \right\} \\
&= \inf\left\{\ \sum_{i=1}^{s} \mathcal{R}_{T_i,L^*,Q}\left(f_i\right) \,\middle|\, f_i : M \to \mathbb{R}, f_i \text{ messbar }, i = 1, \ldots, s \right\} \\
&= \sum_{i=1}^{s} \inf\{\mathcal{R}_{T_i,L^*,Q}\left(f_i\right) |\ f_i : M \to \mathbb{R}, f_i \text{ messbar }, i = 1, \ldots, s\} \\
&= \sum_{i=1}^{s} Q(T_i \times \mathcal{Y}) \inf\{\mathcal{R}_{T_i,L^*,Q_i}\left(f_i\right) |\ f_i : M \to \mathbb{R}, f_i \text{ messbar }, i = 1, \ldots, s\} \\
&= \sum_{i=1}^{s} Q(T_i \times \mathcal{Y})\ \mathcal{R}^*_{T_i,L^*,Q_i}.
\end{aligned}
$$

□

Das folgende Lemma liefert eine vergleichbare Aussage wie das vorangegangene, in diesem Fall aber für die Risiken von messbaren Funktionen.

Lemma 4.5
Sei M eine messbare Menge, $s \in \mathbb{N}$ beliebig, aber fest, und $(T_1, \ldots, T_s)$ eine messbare Partition von M, d. h. T_i messbar, $i = 1, \ldots, s$, $T_i \cap T_j = \emptyset, i, j = 1, \ldots, s$, $i \neq j$, und $M = \bigcup_{i=1}^{s} T_i$. Seien Q ein Wahrscheinlichkeitsmaß auf $(M \times \mathcal{Y}, \mathfrak{B}_{M\times\mathcal{Y}})$ für $\mathcal{Y} \subset \mathbb{R}$ abgeschlossen und L^* eine Lipschitz-stetige geshiftete Verlustfunktion. Dann gilt für alle messbaren Funktionen $f : M \to \mathbb{R}$ mit $\int_{M\times\mathcal{Y}} |f(x)| dQ(x, y) < \infty$ dass

$$\mathcal{R}_{M,L^*,Q}(f) = \sum_{i=1}^{s} Q(T_i \times \mathcal{Y})\ \mathcal{R}_{T_i,L^*,Q_i}(f),$$

wobei Q_i wie in Lemma 4.4 definiert sei, $i = 1, \ldots, s$.

Beweis.

$$\begin{aligned}\mathcal{R}_{M,L^*,Q}(f) &= \int\limits_{M\times\mathcal{Y}} L^*(y,f(x))\; dQ(x,y)\\ &= \sum_{i=1}^{s} \int\limits_{T_i\times\mathcal{Y}} L^*(y,f(x))\; dQ(x,y)\\ &= \sum_{i=1}^{s} Q(T_i\times\mathcal{Y}) \int\limits_{T_i\times\mathcal{Y}} L^*(y,f(x))\; dQ_i(x,y)\\ &= \sum_{i=1}^{s} Q(T_i\times\mathcal{Y})\; \mathcal{R}_{T_i,L^*,Q_i}(f).\end{aligned}$$

□

Das folgende Corollar zeigt, dass die Konvergenz des Risikos gegen das Bayes-Risiko auf einer Menge, die Konvergenz auf jeder ihrer Teilmengen impliziert.

Corollar 4.6
Seien M und T messbare Mengen mit $T \subset M$ und sei $\mathcal{Y} \subset \mathbb{R}$ abgeschlossen. Sei Q ein Wahrscheinlichkeitsmaß auf $(M\times\mathcal{Y}, \mathfrak{B}_{M\times\mathcal{Y}})$. Falls für eine Folge messbarer Funktionen $(f_n)_{n\in\mathbb{N}}$, $f_n : M \to \mathbb{R}$ für alle $n \in \mathbb{N}$, gilt, dass

$$\mathcal{R}_{M,L^*,Q}(f_n) \xrightarrow[n\to\infty]{} \mathcal{R}^*_{M,L^*,Q},$$

so folgt:

$$\mathcal{R}_{T,L^*,Q_T}(f_n) \xrightarrow[n\to\infty]{} \mathcal{R}^*_{T,L^*,Q_T},$$

wobei

$$Q_T := \begin{cases} \frac{1}{Q(T\times\mathcal{Y})}\; Q_{|_{T\times\mathcal{Y}}} & ,\ \textit{falls}\ \ Q(T\times\mathcal{Y})>0 \\ 0 & ,\ \textit{sonst}\end{cases}.$$

Beweis. Das Exzess-Risiko ist stets nichtnegativ, d. h. $\mathcal{R}_{M,L^*,Q}(f_n) - \mathcal{R}^*_{M,L^*,Q} \geq 0$ für alle $n \in \mathbb{N}$, denn $\mathcal{R}^*_{M,L^*,Q}$ ist als das Infimum von $\mathcal{R}_{M,L^*,Q}(\cdot)$ über alle messbaren Funktionen $f : M \to \mathbb{R}$ definiert. Daher und mit Lemmata 4.4 und 4.5 folgt:

$$\begin{aligned}0 &\leq \mathcal{R}_{M,L^*,Q}(f_n) - \mathcal{R}^*_{M,L^*,Q}\\ &= Q(T\times\mathcal{Y})\; \mathcal{R}_{T,L^*,Q_T}(f_n) + Q((M\setminus T)\times\mathcal{Y})\; \mathcal{R}_{M\setminus T,L^*,Q_{M\setminus T}}(f_n)\\ &\quad - \left[Q(T\times\mathcal{Y})\; \mathcal{R}^*_{T,L^*,Q_T} + Q((M\setminus T)\times\mathcal{Y})\; \mathcal{R}^*_{M\setminus T,L^*,Q_{M\setminus T}}\right]\\ &= Q(T\times\mathcal{Y})\left[\mathcal{R}_{T,L^*,Q_T}(f_n) - \mathcal{R}^*_{T,L^*,Q_T}\right]\\ &\quad + Q((M\setminus T)\times\mathcal{Y})\left[\mathcal{R}_{M\setminus T,L^*,Q_{M\setminus T}}(f_n) - \mathcal{R}^*_{M\setminus T,L^*,Q_{M\setminus T}}\right]. \qquad (4.1)\end{aligned}$$

Da Q ein Wahrscheinlichkeitsmaß ist, gilt $Q(T \times \mathcal{Y}) \in [0,1]$ und $Q((M \setminus T) \times \mathcal{Y}) \in [0,1]$; und aufgrund der Infima, für alle $n \in \mathbb{N}$

$$\mathcal{R}_{T,L^*,Q_T}(f_n) - \mathcal{R}^*_{T,L^*,Q_T} \geq 0$$

und

$$\mathcal{R}_{M\setminus T,L^*,Q_{M\setminus T}}(f_n) - \mathcal{R}^*_{M\setminus T,L^*,Q_{M\setminus T}} \geq 0.$$

Es war vorausgesetzt, dass $\mathcal{R}_{M,L^*,Q}(f_n) - \mathcal{R}^*_{M,L^*,Q} \xrightarrow[n\to\infty]{} 0$. Da beide Terme in den eckigen Klammern in (4.1) nichtnegativ sind, folgt, dass jeder Summand gegen 0 konvergiert (für $n \to \infty$). Somit ist gezeigt, dass $\mathcal{R}_{T,L^*,Q_T}(f_n) - \mathcal{R}^*_{T,L^*,Q_T} \xrightarrow[n\to\infty]{} 0$.

□

Lemma 4.7

Um das Bayes-Risiko bezüglich einer geshifteten Verlustfunktion L^* und einer Verteilung P zu erreichen, reicht es aus, über alle Funktionen $f \in \mathcal{F}$ zu optimieren, sofern $\mathcal{F}$ dicht in $L^1(P^{\mathcal{X}})$ liegt, d. h.

$$\mathcal{R}^*_{\mathcal{X},L^*,P} = \mathcal{R}^*_{\mathcal{X},L^*,P,\mathcal{F}}.$$

Beweis zu Lemma 4.7. Es gilt, dass

$$\begin{aligned}
\mathcal{R}^*_{\mathcal{X},L^*,P} &= \inf\left\{ \int_{\mathcal{X}\times\mathcal{Y}} L^*(y, f(x))\, dP(x,y) \,\middle|\, f : \mathcal{X} \to \mathbb{R}, f \text{ messbar} \right\} \\
&= \inf\left\{ \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) - L(y,0)\, dP(x,y) \,\middle|\, f : \mathcal{X} \to \mathbb{R}, f \text{ messbar} \right\} \\
&= \inf\left\{ \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) - L(y,0)\, dP(x,y) \,\middle|\, f : \mathcal{X} \to \mathbb{R}, f \in \mathcal{F} \right\} \\
&= \mathcal{R}^*_{\mathcal{X},L^*,P,\mathcal{F}}.
\end{aligned}$$

Das ist gültig aufgrund von Steinwart & Christmann (2008, Theorem 5.31) und der Tatsache, dass f keinen Einfluss auf $L(\cdot, 0)$ ausübt. □

Damit stehen die Hilfsmittel zur Verfügung, um die Konsistenzaussage (Theorem 4.1) selbst zu beweisen.

Beweis zu Theorem 4.1. Die folgenden Zeilen enthalten den Beweis des Konsistenztheorems. Hierzu wird die Differenz der betrachteten Risiken mittels Dreiecksungleichung in zwei Teile zerlegt, die anschließend getrennt voneinander untersucht werden. Man erhält:

$$\begin{aligned}
&\left|\mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,D_n,\lambda_n}) - \mathcal{R}^*_{\mathcal{X},L^*,P}\right| \\
\leq\ &\underbrace{\left|\mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,D_n,\lambda_n}) - \mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n})\right|}_{\text{Term 1}} + \underbrace{\left|\mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n}) - \mathcal{R}^*_{\mathcal{X},L^*,P}\right|}_{\text{Term 2}}.
\end{aligned}$$

Für den weiteren Beweis ist P ein beliebiges, aber festes Wahrscheinlichkeitsmaß auf $(\mathcal{X}\times\mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$. Term 1 wird mittels stochastischer Methoden diskutiert, insbesondere mittels der Hoeffding-Ungleichung in Hilberträumen. Der zweite Term verschwindet asymptotisch; Letzteres kann durch Betrachtung der sogenannten Approximationsfehler-Funktion gezeigt werden. Beide Argumentationen zusammen liefern stochastische Konvergenz gegen 0.

Für Term 1 erhält man die folgende Abschätzung:

$$\begin{aligned}
&\left|\mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,D_n,\lambda_n}) - \mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n})\right| \qquad (4.2)\\
=\ &\left| \int_{\mathcal{X}\times\mathcal{Y}} L^*\left(y, f^{comp}_{L^*,D_n,\lambda_n}(x)\right) - L^*\left(y, f^{comp}_{L^*,P,\lambda_n}(x)\right)\ dP(x,y) \right| \\
=\ &\left| \int_{\mathcal{X}\times\mathcal{Y}} L\left(y, f^{comp}_{L^*,D_n,\lambda_n}(x)\right) - L(y,0) - L\left(y, f^{comp}_{L^*,P,\lambda_n}(x)\right) + L(y,0)\ dP(x,y) \right| \\
\leq\ &|L|_1 \int_{\mathcal{X}\times\mathcal{Y}} \left|f^{comp}_{L^*,D_n,\lambda_n}(x) - f^{comp}_{L^*,P,\lambda_n}(x)\right|\ dP(x,y) \\
\leq\ &|L|_1 \int_{\mathcal{X}\times\mathcal{Y}} \sum_{b=1}^{B} w_b(x)\ \left|f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x) - f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
=\ &|L|_1 \sum_{b=1}^{B} \int_{\mathcal{X}\times\mathcal{Y}} w_b(x)\ \left|f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x) - f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
=\ &|L|_1 \sum_{b=1}^{B} \int_{\mathcal{X}_b\times\mathcal{Y}} w_b(x)\ \left|f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}}(x) - f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right|\ dP(x,y) \\
\leq\ &|L|_1 \sum_{b=1}^{B} P(\mathcal{X}_b\times\mathcal{Y})\ \|f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}} - f_{b,L^*,P_b,\lambda_{(n_b,b)}}\|_{\mathcal{X}_b\text{-}\infty} \\
\overset{(1.1)}{\leq}\ &|L|_1 \sum_{b=1}^{B} P(\mathcal{X}_b\times\mathcal{Y})\ \|k_b\|_{\mathcal{X}_b\text{-}\infty}\ \|f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}} - f_{b,L^*,P_b,\lambda_{(n_b,b)}}\|_{H_b}.
\end{aligned}$$

Für alle $b \in \{1,\ldots,B\}$ konvergiert der letzte Faktor $\left\|f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}} - f_{b,L^*,P_b,\lambda_{(n_b,b)}}\right\|_{H_b}$ in Wahrscheinlichkeit bzgl. P_b gegen 0 für $n_b \to \infty$ (was durch Voraussetzung **(R3)** sichergestellt ist, sofern $n \to \infty$). Somit konvergiert der gesamte Ausdruck und damit die Differenz in (4.2) in Wahrscheinlichkeit bzgl. P gegen 0. Detailliert kann

Letzteres wie folgt gezeigt werden: Für jedes $b \in \{1, \ldots, B\}$ garantiert Christmann, Van Messem & Steinwart (2009, Theorem 7) für alle $n \in \mathbb{N}$ die Existenz einer beschränkten, messbaren Funktion $h_{(n,b)} : \mathcal{X}_b \times \mathcal{Y} \to \mathbb{R}$ mit $\|h_{(n,b)}\|_{\mathcal{X}_b\text{-}\infty} \leq |L|_1$ und außerdem für jedes Wahrscheinlichkeitsmaß $\widetilde{P}_b$ auf $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b \times \mathcal{Y}})$:

$$\left\| f_{b,L^*,P_b,\lambda_{(n_b,b)}} - f_{b,L^*,\widetilde{P}_b,\lambda_{(n_b,b)}} \right\|_{H_b} \leq \frac{1}{\lambda_{(n_b,b)}} \left\| \mathbb{E}_{P_b}\left[h_{(n,b)}\Phi_b\right] - \mathbb{E}_{\widetilde{P}_b}\left[h_{(n,b)}\Phi_b\right] \right\|_{H_b},$$

wobei $\Phi_b : \mathcal{X}_b \to H_b,\ \Phi_b(x) = k_b(\cdot, x)$ für alle $x \in \mathcal{X}_b$. Sei $\varepsilon_b \in]0,1[$. Für diesen Beweis besteht besonderes Interesse an $\widetilde{P}_b := D_{n,b}$. Sei

$$S_{(n,b)} := \left\{ \mathcal{D}_{n,b} \in (\mathcal{X}_b \times \mathcal{Y})^{n_b} \,\middle|\, \left\| \mathbb{E}_{P_b}\left[h_{(n,b)}\Phi_b\right] - \mathbb{E}_{D_{n,b}}\left[h_{(n,b)}\Phi_b\right] \right\|_{H_b} \leq \frac{\lambda_{(n_b,b)}\varepsilon_b}{|L|_1} \right\},$$

wobei $n_b := |\mathcal{D}_{n,b}|$. Für alle $\mathcal{D}_{n,b} \in S_{(n,b)}$ folgt dann

$$\left\| f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}} - f_{b,L^*,P_b,\lambda_{(n_b,b)}} \right\|_{H_b} \leq \frac{\varepsilon_b}{|L|_1}.$$

Betrachtet wird nun die Wahrscheinlichkeit von $S_{(n,b)}$. Wie in Christmann, Van Messem & Steinwart (2009) unter Anwendung der Hoeffding-Ungleichung in Hilberträumen ausgeführt, gilt: $P_b^{n_b}(S_{(n,b)}) \to 1$ für $n_b \to \infty$. Somit:

$$\left\| f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}} - f_{b,L^*,P_b,\lambda_{(n_b,b)}} \right\|_{H_b} \to 0 \ \text{ in Wahrscheinlichkeit bzgl. } P_b, \quad n_b \to \infty.$$

Für alle $b \in \{1, \ldots, B\}$ ist nun bekannt, dass $\left\| f_{b,L^*,D_{n,b},\lambda_{(n_b,b)}} - f_{b,L^*,P_b,\lambda_{(n_b,b)}} \right\|_{H_b} \to 0$ in Wahrscheinlichkeit bezüglich P_b, wenn $n_b \to \infty$. Somit verschwindet Term 1 stochastisch, d. h.

$$|\mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,D_n,\lambda_n}) - \mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n})| \xrightarrow[n\to\infty]{} 0 \tag{4.3}$$

in Wahrscheinlichkeit bezüglich P.

Noch zu zeigen ist das asymptotische Verhalten des zweiten Terms. Es gilt

$$\mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n}) - \mathcal{R}^*_{\mathcal{X},L^*,P} \geq 0,$$

denn $f^{comp}_{L^*,P,\lambda_n}$ ist eine messbare Funktion (da als Voraussetzung nur messbare Kerne verwendet wurden, vgl. Lemma 4.24 in Steinwart & Christmann (2008)) und $\mathcal{R}^*_{\mathcal{X},L^*,P}$ ist das Infimum von $\mathcal{R}_{\mathcal{X},L^*,P}(\cdot)$ über alle messbaren Funktionen von $\mathcal{X}$ nach $\mathbb{R}$. Die Analyse kann also ohne Betragsstriche fortgesetzt werden.

Zusätzlich werden Eigenschaften der sogenannten inneren Risiken in Erinnerung gerufen, siehe beispielsweise Steinwart & Christmann (2008, S. 51f). Das innere Risiko bezüglich eines zugehörigen Risikos $\mathcal{R}_{\Xi,L^*,P}$ wird mittels

$$\mathcal{C}_{L^*,P(\cdot|x),x}(t) := \int_{\mathcal{Y}} L^*(y,t)\, dP(y|x), \quad x \in \Xi,\ t \in \mathbb{R}, \tag{4.4}$$

definiert, wobei $P(Y|x)$ die bedingte Verteilung von Y gegeben $X = x$ bezeichne und das kleinste innere Risiko durch

$$\mathcal{C}^*_{L^*,P(\cdot|x),x} := \inf\left\{\mathcal{C}_{L^*,P(\cdot|x),x}(t) \;\middle|\; t \in \mathbb{R}\right\}, \quad x \in \Xi,$$

definiert werde. Steinwart & Christmann (2008) beweisen den Zusammenhang zwischen Risiken und inneren Risiken: Für jeden vollständig messbaren Raum Ξ gilt:

$$\mathcal{R}^*_{\Xi,L^*,P} = \int_{\Xi} \mathcal{C}^*_{L^*,P(\cdot|x),x} \, dP^{\Xi}(x) \tag{4.5}$$

und

$$\mathcal{R}_{\Xi,L^*,P}(f) = \int_{\Xi} \mathcal{C}_{L^*,P(\cdot|x),x}(f(x)) \, dP^{\Xi}(x) \tag{4.6}$$

für alle messbaren $f : \Xi \to \mathbb{R}$.

Durch die Anwendung von Lemma 4.4 und der Voraussetzungen an die Gewichte **(W1)** und **(W2)** erhält man:

$$\begin{aligned}
\mathcal{R}^*_{\mathcal{X},L^*,P} &= \sum_{I\subset\{1,\dots,B\}} \mathcal{R}^*_{\mathcal{X}_I,L^*,P} \\
&= \sum_{I\subset\{1,\dots,B\}} \int_{\mathcal{X}_I} \mathcal{C}^*_{L^*,P(\cdot|x),x} \, dP^{\mathcal{X}}(x) \\
&= \sum_{I\subset\{1,\dots,B\}} \int_{\mathcal{X}_I} \underbrace{\left[\sum_{b=1}^{B} w_b(x)\right]}_{=1} \mathcal{C}^*_{L^*,P(\cdot|x),x} \, dP^{\mathcal{X}}(x) \\
&= \sum_{I\subset\{1,\dots,B\}} \sum_{b=1}^{B} \int_{\mathcal{X}_I} w_b(x) \, \mathcal{C}^*_{L^*,P(\cdot|x),x} \, dP^{\mathcal{X}}(x) \\
&= \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} \int_{\mathcal{X}_I} w_b(x) \, \mathcal{C}^*_{L^*,P(\cdot|x),x} \, dP^{\mathcal{X}}(x).
\end{aligned} \tag{4.7}$$

Es gilt zu beachten, dass **(W2)**, d. h. $w_b(x) = 0$ für alle $x \notin \mathcal{X}_b$, $b \in \{1, \dots, B\}$, tatsächlich für obenstehende Gleichheitskette genutzt wurde.

Die Gleichung (4.7) wird in einem der nächsten Schritte genutzt. Zunächst wird aber eine obere Schranke für Term 2 hergeleitet. Man erhält:

$$
\begin{aligned}
0 &\leq \mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n}) - \mathcal{R}^*_{\mathcal{X},L^*,P} \\
&= \int\limits_{\mathcal{X}\times\mathcal{Y}} L^*\left(y, f^{comp}_{L^*,P,\lambda_n}(x)\right)\ dP(x,y) - \mathcal{R}^*_{\mathcal{X},L^*,P} \\
&= \int\limits_{\mathcal{X}\times\mathcal{Y}} L^*\left(y, \sum_{b=1}^{B} w_b(x) f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right)\ dP(x,y) - \mathcal{R}^*_{\mathcal{X},L^*,P}.
\end{aligned}
$$

Da L^* als konvex bzgl. des letzten Arguments vorausgesetzt wurde, kann wie folgt fortgefahren werden:

$$
\begin{aligned}
0 &\leq \mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n}) - \mathcal{R}^*_{\mathcal{X},L^*,P} \\
&\leq \int\limits_{\mathcal{X}\times\mathcal{Y}} \sum_{b=1}^{B} w_b(x)\ L^*\left(y, f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right)\ dP(x,y) - \mathcal{R}^*_{\mathcal{X},L^*,P} \\
&= \sum_{I\subset\{1,\ldots,B\}} \int\limits_{\mathcal{X}_I\times\mathcal{Y}} \sum_{b=1}^{B} w_b(x) \left(L^*(y, f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right)\ dP(x,y) - \mathcal{R}^*_{\mathcal{X},L^*,P},
\end{aligned}
$$

was unter Verwendung von Voraussetzung **(W2)**, (4.4) und (4.7) zu

$$
\begin{aligned}
0 &\leq \mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n}) - \mathcal{R}^*_{\mathcal{X},L^*,P} \\
&\leq \sum_{I\subset\{1,\ldots,B\}} \sum_{b\in I} \int\limits_{\mathcal{X}_I} w_b(x)\ \mathcal{C}_{L^*,P(\cdot|x),x}\left(f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right)\ dP^{\mathcal{X}}(x) - \mathcal{R}^*_{\mathcal{X},L^*,P} \\
&= \sum_{I\subset\{1,\ldots,B\}} \sum_{b\in I} \int\limits_{\mathcal{X}_I} w_b(x)\ \mathcal{C}_{L^*,P(\cdot|x),x}\left(f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right)\ dP^{\mathcal{X}}(x) \\
&\quad - \sum_{I\subset\{1,\ldots,B\}} \sum_{b\in I} \int\limits_{\mathcal{X}_I} w_b(x)\ \mathcal{C}^*_{L^*,P(\cdot|x),x}\ dP^{\mathcal{X}}(x) \\
&= \sum_{I\subset\{1,\ldots,B\}} \sum_{b\in I} \int\limits_{\mathcal{X}_I} w_b(x)\ \left[\mathcal{C}_{L^*,P(\cdot|x),x}\left(f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right) - \mathcal{C}^*_{L^*,P(\cdot|x),x}\right]\ dP^{\mathcal{X}}(x)
\end{aligned}
$$

führt. Die Gewichte $w_b(\cdot)$ nehmen Werte in $[0,1]$ an, $b \in \{1,\ldots,B\}$, und der Term in den eckigen Klammern, d. h. das innere Exzess-Risiko, ist nichtnegativ. Somit

erhält man mit (4.5) und (4.6)

$$
\begin{aligned}
0 &\leq \mathcal{R}_{\mathcal{X},L^*,P}(f^{comp}_{L^*,P,\lambda_n}) - \mathcal{R}^*_{\mathcal{X},L^*,P} \\
&\leq \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} \int_{\mathcal{X}_I} \left[\mathcal{C}_{L^*,P(\cdot|x),x}\left(f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right) - \mathcal{C}^*_{L^*,P(\cdot|x),x}\right] \, dP^{\mathcal{X}}(x) \\
&= \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} \left[\int_{\mathcal{X}_I} \mathcal{C}_{L^*,P(\cdot|x),x}\left(f_{b,L^*,P_b,\lambda_{(n_b,b)}}(x)\right) \, dP^{\mathcal{X}}(x) - \int_{\mathcal{X}_I} \mathcal{C}^*_{L^*,P(\cdot|x),x} \, dP^{\mathcal{X}}(x)\right] \\
&= \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} \left[\mathcal{R}_{\mathcal{X}_I,L^*,P}(f_{b,L^*,P_b,\lambda_{(n_b,b)}}) - \mathcal{R}^*_{\mathcal{X}_I,L^*,P}\right] \\
&= \sum_{I\subset\{1,\dots,B\}} \sum_{b\in I} P(\mathcal{X}_I\times\mathcal{Y}) \, \left[\mathcal{R}_{\mathcal{X}_I,L^*,P_I}(f_{b,L^*,P_b,\lambda_{(n_b,b)}}) - \mathcal{R}^*_{\mathcal{X}_I,L^*,P_I}\right], \qquad (4.8)
\end{aligned}
$$

wobei

$$
P_I := \begin{cases} \frac{1}{P(\mathcal{X}_I\times\mathcal{Y})} \, P_{|\mathcal{X}_I\times\mathcal{Y}} & , \text{ falls } \; P(\mathcal{X}_I\times\mathcal{Y}) > 0 \\ 0 & , \text{ sonst} \end{cases}.
$$

Bezugnehmend auf die Differenzen der lokalen Risiken in der vorangehenden Formel sei nun für alle $b \in \{1,\dots,B\}$ und für alle $f \in H_b$ die sogenannte Approximationsfehler-Funktion $A_{b,f} : [0,\infty[\to \mathbb{R}$ definiert durch

$$
\lambda \mapsto \mathcal{R}_{\mathcal{X}_b,L^*,P_b,\lambda_b}(f) - \mathcal{R}^*_{\mathcal{X}_b,L^*,P_b,H_b} = \mathcal{R}_{\mathcal{X}_b,L^*,P_b}(f) + \lambda_b\|f\|^2_{H_b} - \mathcal{R}^*_{\mathcal{X}_b,L^*,P_b,H_b}. \qquad (4.9)
$$

Aus Steinwart & Christmann (2008, Lemma A.6.4) ist bekannt, dass – falls Nullfolgen $(\lambda_{(n_b,b)})_{n_b\in\mathbb{N}}$, $\lambda_{(n_b,b)} > 0$ für alle $n_b \in \mathbb{N}$ und für alle $b \in \{1,\dots,B\}$, verwendet werden –

$$
\inf_{f\in H_b} A_{b,f}(\lambda_{(n_b,b)}) \to \inf_{f\in H_b} A_{b,f}(0) \quad \text{für } n_b \to \infty.
$$

Da eine SVM als Minimierer des regularisierten Risikos für alle $n \in \mathbb{N}, b \in \{1,\dots,B\}$, definiert ist, gilt:

$$
\inf_{f\in H_b} A_{b,f}(\lambda_{(n_b,b)}) = \mathcal{R}_{\mathcal{X}_b,L^*,P_b}(f_{b,L^*,P_b,\lambda_{(n_b,b)}}) + \lambda_{(n_b,b)}\|f_{b,L^*,P_b,\lambda_{(n_b,b)}}\|^2_{H_b} - \mathcal{R}^*_{\mathcal{X}_b,L^*,P_b,H_b}.
$$

Aufgrund von Steinwart & Christmann (2008, Theorem 5.31) und (4.9) folgt $\inf_{f\in H_b} A_{b,f}(0) = 0$. Somit erhält man für alle $b \in \{1,\dots,B\}$:

$$
\begin{aligned}
0 &\leq \limsup_{n_b\to\infty} \left(\mathcal{R}_{\mathcal{X}_b,L^*,P_b,\lambda_b}(f_{b,L^*,P_b,\lambda_{(n_b,b)}}) - \mathcal{R}^*_{\mathcal{X}_b,L^*,P_b,H_b}\right) \\
&= \limsup_{n_b\to\infty} \left(\mathcal{R}_{\mathcal{X}_b,L^*,P_b}(f_{b,L^*,P_b,\lambda_{(n_b,b)}}) + \lambda_{(n_b,b)}\|f_{b,L^*,P_b,\lambda_{(n_b,b)}}\|^2_{H_b} - \mathcal{R}^*_{\mathcal{X}_b,L^*,P_b,H_b}\right) \\
&\leq \limsup_{n_b\to\infty} \left(\inf_{f\in H_b} A_{b,f}(\lambda_{(n_b,b)}) - \inf_{f\in H_b} A_{b,f}(0)\right) = 0.
\end{aligned}
$$

Der Abstand zum kleinsten L^*-Risiko über H_b verschwindet also. Wegen Lemma 4.7 gilt für alle $b \in \{1, \dots, B\}$, dass $\mathcal{R}^*_{\mathcal{X}_b, L^*, P_b} = \mathcal{R}^*_{\mathcal{X}_b, L^*, P_b, H_b}$. Somit verschwinden alle lokalen Differenzen zum jeweiligen Bayes-Risiko

$$\mathcal{R}_{\mathcal{X}_b, L^*, P_b}(f_{b, L^*, P_b, \lambda_{(n_b, b)}}) - \mathcal{R}^*_{\mathcal{X}_b, L^*, P_b}, \quad b = 1, \dots, B.$$

Da für alle $I \subset \{1, \dots, B\}$ gilt, dass $\mathcal{X}_I \subset \mathcal{X}_b$ für wenigstens ein $b \in \{1, \dots, B\}$, folgt mit Corollar 4.6:

$$\mathcal{R}_{\mathcal{X}_I, L^*, P_I}(f_{b, L^*, P_b, \lambda_{(n_b, b)}}) - \mathcal{R}^*_{\mathcal{X}_I, L^*, P_I} \xrightarrow[n \to \infty]{} 0 \quad \text{für alle } I \subset \{1, \dots, B\} \text{ und alle } b \in I.$$

Die Nutzung dessen in (4.8) liefert

$$|\mathcal{R}_{\mathcal{X}, L^*, P}(f^{comp}_{L^*, P, \lambda_n}) - \mathcal{R}^*_{\mathcal{X}, L^*, P}| \to 0 \text{ für } n \to \infty. \tag{4.10}$$

Durch Kombination der Ergebnisse aus (4.3) und (4.10) erhält man

$$|\mathcal{R}_{\mathcal{X}, L^*, P}(f^{comp}_{L^*, D_n, \lambda_n}) - \mathcal{R}^*_{\mathcal{X}, L^*, P}| \to 0 \text{ in Wahrscheinlichkeit bezüglich } P.$$

□

4.3 Robustheit im Sinne des *maxbias*

Neben Berechenbarkeit und Konsistenz ist Robustheit (oder im weiteren Sinn: Stabilität) von großem Interesse für statistische Methoden.[45] Gezeigt werden daher auch Robustheitseigenschaften der vorgestellten Prädiktoren.[46] Stabilitätsresultate (*total stability*) für global gelernte Support Vector Machines wurden in Christmann, Xiang & Zhou (2018) gezeigt. Es bezeichne im Folgenden $\mathcal{M}_1(M)$ die Menge der Wahrscheinlichkeitsmaße auf der Borel-σ-Algebra einer Menge M. Für alle $b \in \{1, \dots, B\}$ und $\varepsilon_b \in [0, \frac{1}{2}[$ sei wieder $P_b := P(\mathcal{X}_b \times \mathcal{Y})^{-1} P_{|\mathcal{X}_b \times \mathcal{Y}}$, falls $P(\mathcal{X}_b \times \mathcal{Y}) \neq 0$, und das Nullmaß sonst. Für eine Verteilung P auf $\mathcal{X} \times \mathcal{Y}$ sei die ε_b-Kontaminationsumgebung von P (oder P_b) auf $\mathcal{X}_b \times \mathcal{Y}$ definiert durch

$$N_{b, \varepsilon_b}(P) := N_{b, \varepsilon_b}(P_b) := \left\{ (1 - \varepsilon_b) P_b + \varepsilon_b \tilde{P}_b \;\middle|\; \tilde{P}_b \in \mathcal{M}_1(\mathcal{X}_b \times \mathcal{Y}) \right\}.$$

Sei außerdem $\varepsilon := (\varepsilon_1, \dots, \varepsilon_B) \in [0, \frac{1}{2}[^B$. Somit kann

$$N_\varepsilon(P) := \left\{ \tilde{P} \in \mathcal{M}_1(\mathcal{X} \times \mathcal{Y}) \;\middle|\; \tilde{P}_b \in N_{b, \varepsilon_b} \text{ für alle } b \in \{1, \dots, B\} \right\}$$

[45] Eine einleitende und mit historischen Bezügen versehene Diskussion dieser Feststellung findet sich in den ersten Abschnitten von Hampel, Ronchetti, Rousseeuw & Stahel (1986).

[46] Der Abschnitt zur Robustheit im Sinne des *maxbias* entspricht dem in der Veröffentlichung Dumpert & Christmann (2018).

als zusammengesetzte ε-Kontaminationsumgebung von P auf $\mathcal{X}\times\mathcal{Y}$ definiert werden.

Auf Basis dieser ist es möglich, eine obere Schranke für den sogenannten *maxbias* herzuleiten, d. h. eine obere Schranke für die mögliche Differenz zweier zusammengesetzter Prädiktoren, die jeweils aus lokal gelernten SVMs gebildet werden. Die Supremumsnorm auf $\mathcal{X}_b$ wird wie bisher mit $\|\cdot\|_{\mathcal{X}_b\text{-}\infty}$ bezeichnet, d. h. für eine Funktion $f : \mathcal{X} \to \mathbb{R}$ sei

$$\|f\|_{\mathcal{X}_b\text{-}\infty} := \sup\{|f(x)| \mid x \in \mathcal{X}_b\}$$

und für einen Kern $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ sei

$$\|k\|_{\mathcal{X}_b\text{-}\infty} := \sup_{x\in\mathcal{X}_b} \sqrt{k(x,x)}.$$

Theorem 4.8
*Sei L eine konvexe, Lipschitz-stetige Verlustfunktion (mit Lipschitz-Konstante $|L|_1 \neq 0$) und L^* die zugehörige geshiftete Version. Für alle $b \in \{1,\ldots,B\}$ sei k_b ein messbarer und beschränkter Kern auf $\mathcal{X}$; die zugehörigen RKHS H_b seien separabel. Die Regionalisierungsmethode erfülle **(R1)**, **(R2)** und **(R3)**. Dann gilt für alle Verteilungen P auf $(\mathcal{X}\times\mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$ und alle $\lambda := (\lambda_1,\ldots,\lambda_B) \in [0,\infty[^B$, dass*

$$\sup_{\tilde{P}\in N_\varepsilon(P)} \left\| f^{comp}_{L^*,\tilde{P},\lambda} - f^{comp}_{L^*,P,\lambda} \right\|_\infty \leq 2\ |L|_1 \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty}\ \frac{\varepsilon_b}{\lambda_b}\ \|k_b\|^2_{\mathcal{X}_b\text{-}\infty}. \tag{4.11}$$

Diese obere Schranke ist gleichmäßig in dem Sinne, dass sie für alle Verteilungen P und alle Gewichte, die **(W1)** und **(W2)** erfüllen, d. h. $\sum_{b=1}^{B} w_b(x) = 1$ für alle $x \in \mathcal{X}$ und $w_b(x) = 0$ für alle $x \notin \mathcal{X}_b$ und für alle $b \in \{1,\ldots,B\}$, gilt.

Beispiel 4.9
Sei $d \in \mathbb{N}$, $\mathcal{X} \subset \mathbb{R}^d$, k_b ein Gaußkern, d. h. $k_b(x,x') := \exp\left(-\gamma_b^{-2}\|x-x'\|_2^2\right), \gamma_b > 0$, für alle $b \in \{1,\ldots,B\}$, und sei L die hinge-Verlustfunktion (für Klassifikation) oder die τ-pinball-Verlustfunktion (für Quantilsregression, $\tau \in]0,1[$). Dann liefert Theorem 4.8 die obere Schranke

$$\sup_{\tilde{P}\in N_\varepsilon(P)} \left\| f^{comp}_{L^*,\tilde{P},\lambda} - f^{comp}_{L^*,P,\lambda} \right\|_\infty \leq \sum_{b=1}^{B} \frac{1}{\lambda_b}.$$

Beweis zu Theorem 4.8. Der Beweis kann mit einfachen Mitteln geführt werden: Dreiecksungleichung, obere Grenzen für Suprema, die vorausgesetzten Eigenschaften an die Gewichte sowie Ungleichung aus Proposition 1.1. Die Totalvariationsnorm, vgl. beispielsweise Denkowski, Migórski & Papageorgiou (2003, S. 158), auf $\mathcal{X}_b \times \mathcal{Y}$

sei im Folgenden mit $\|\cdot\|_{(\mathcal{X}_b\times\mathcal{Y})\text{-TV}}$ bezeichnet. Dann lässt sich zeigen:

$$
\begin{aligned}
&\sup_{\tilde{P}\in N_\varepsilon(P)} \left\| f^{comp}_{L^*,\tilde{P},\lambda} - f^{comp}_{L^*,P,\lambda} \right\|_\infty \\
&= \sup_{\tilde{P}\in N_\varepsilon(P)} \sup_{x\in\mathcal{X}} \left| \sum_{b=1}^{B} w_b(x) \left(f_{b,L^*,\tilde{P}_b,\lambda_b}(x) - f_{b,L^*,P_b,\lambda_b}(x) \right) \right| \\
&\leq \sup_{\tilde{P}\in N_\varepsilon(P)} \sup_{x\in\mathcal{X}} \sum_{b=1}^{B} \left| w_b(x) \left(f_{b,L^*,\tilde{P}_b,\lambda_b}(x) - f_{b,L^*,P_b,\lambda_b}(x) \right) \right| \\
&\leq \sup_{\tilde{P}\in N_\varepsilon(P)} \sum_{b=1}^{B} \sup_{x\in\mathcal{X}} \left| w_b(x) \left(f_{b,L^*,\tilde{P}_b,\lambda_b}(x) - f_{b,L^*,P_b,\lambda_b}(x) \right) \right| \\
&\leq \sup_{\tilde{P}\in N_\varepsilon(P)} \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \sup_{x\in\mathcal{X}_b} \left| f_{b,L^*,\tilde{P}_b,\lambda_b}(x) - f_{b,L^*,P_b,\lambda_b}(x) \right| \\
&= \sup_{\tilde{P}\in N_\varepsilon(P)} \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \left\| f_{b,L^*,\tilde{P}_b,\lambda_b} - f_{b,L^*,P_b,\lambda_b} \right\|_{\mathcal{X}_b\text{-}\infty} \\
&\leq \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \sup_{\tilde{P}\in N_\varepsilon(P)} \left\| f_{b,L^*,\tilde{P}_b,\lambda_b} - f_{b,L^*,P_b,\lambda_b} \right\|_{\mathcal{X}_b\text{-}\infty} \\
&\leq \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \sup_{\tilde{P}_b\in N_{b,\varepsilon_b}(P_b)} \left\| f_{b,L^*,\tilde{P}_b,\lambda_b} - f_{b,L^*,P_b,\lambda_b} \right\|_{\mathcal{X}_b\text{-}\infty} \\
&\overset{(1.1)}{\leq} \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \sup_{\tilde{P}_b\in N_{b,\varepsilon_b}(P_b)} \|k_b\|_{\mathcal{X}_b\text{-}\infty} \left\| f_{b,L^*,\tilde{P}_b,\lambda_b} - f_{b,L^*,P_b,\lambda_b} \right\|_{H_b} \\
&\leq \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \sup_{\tilde{P}_b\in N_{b,\varepsilon_b}(P_b)} \|k_b\|_{\mathcal{X}_b\text{-}\infty} \varepsilon_b \frac{1}{\lambda_b} |L|_1 \|k_b\|_{\mathcal{X}_b\text{-}\infty} \|\tilde{P}_b - P_b\|_{(\mathcal{X}_b\times\mathcal{Y})\text{-TV}} \\
&\leq 2\ |L|_1 \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \frac{\varepsilon_b}{\lambda_b} \|k_b\|^2_{\mathcal{X}_b\text{-}\infty} .
\end{aligned}
$$

Die vorletzte Ungleichung ist gültig aufgrund von Christmann, Van Messem & Steinwart (2009, Theorem 12), die letzte Ungleichung aufgrund der Tatsache, dass die Totalvariation der Differenz zweier Wahrscheinlichkeitsmaße stets nach oben durch 2 abgeschätzt werden kann. □

4.4 Robustheit im Sinne der Influenzfunktion

In diesem Abschnitt[47] wird ein anderer Robustheitsbegriff betrachtet, die sogenannte Influenzfunktion im Sinne von Hampel (1968), siehe auch Hampel, Ronchetti, Rousseeuw & Stahel (1986). Betrachtet wird dabei ein statistischer Operator S, der

[47] Der Abschnitt zur Robustheit im Sinne der Influenzfunktion entspricht der Veröffentlichung Dumpert (2019b).

jeder Verteilung P auf der Borel-σ-Algebra $\mathfrak{B}_M$ einer geeigneten Menge M ein Element eines Banachraumes, d. h. in der in dieser Arbeit betrachteten Situation einen (zusammengesetzten) Prädiktor $f_{L^*,P,\lambda}$ zuordnet. Bei SVMs ohne Regionalisierung ist dieser Prädiktor sogar Element eines Hilbertraums.

Definition 4.10
Die Influenzfunktion von S an einer Stelle z für eine Verteilung P ist (sofern der Grenzwert existiert) definiert als

$$\mathrm{IF}(\delta_z; S, P) := \lim_{\varepsilon \searrow 0} \frac{S\left((1-\varepsilon)P + \varepsilon\delta_z\right) - S(P)}{\varepsilon},$$

wobei δ_z die Dirac-Verteilung (Einpunktverteilung) im Punkt z ist.

Die Influenzfunktion kann in dem Sinne interpretiert werden, dass sie den Einfluss einer infinitesimal kleinen Störung der Originalverteilung P in Richtung einer Dirac-Verteilung im Punkt z auf die interessierende Größe $S(P)$ (also den Prädiktor) misst. Sofern die Influenzfunktion existiert und sofern sie stetig und linear ist, handelt es sich um eine Gâteaux-Ableitung des statistischen Operators $S : \mathcal{M}_1(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}}) \to H, P \mapsto f_{L^*,P,\lambda}$ in Richtung der Mischungsverteilung $(1-\varepsilon)P + \varepsilon\delta_z$.[48] Mit dieser Interpretation im Hinterkopf leuchtet es ein, dass man an Bedingungen dafür interessiert ist, wann die vorgeschlagene statistische Methode eine beschränkte Influenzfunktion besitzt; je kleiner die obere Schranke, desto robuster ist die Methode. Die Influenzfunktion selbst ist wiederum eine Funktion, die eine Dirac-Verteilung δ auf $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$ auf einen Prädiktor in einem RKHS abbildet, d. h. $\mathrm{IF}(\delta; S, P)(\cdot) \in H$. Es besteht also die Möglichkeit, $\mathrm{IF}(\delta; S, P)(\cdot)$ an einem Punkt $x \in \mathcal{X}$ auszuwerten und wegen Proposition 1.2 eine reelle Zahl ($\mathrm{IF}(\delta; S, P)(x) \in \mathbb{R}$ für alle $x \in \mathcal{X}$) als Ergebnis zu erhalten, sofern ein stetiger und beschränkter Kern verwendet wird.

Proposition 4.11
Wie in Christmann, Van Messem & Steinwart (2009) bewiesen, existiert die Influenzfunktion in der nichtregionalisierten Situation und ist beschränkt, falls $\mathcal{X}$ ein vollständiger, separabler metrischer Raum, H der RKHS eines beschränkten und stetigen Kerns k sowie L eine konvexe und Lipschitz-stetige Verlustfunktion mit stetigen partiellen (Fréchet-)Ableitungen (bezüglich des letzten Argumentes) $L'(y,\cdot)$ und $L''(y,\cdot)$ mit $\sup_{y\in\mathcal{Y}} \|L'(y,\cdot)\|_\infty \in]0,\infty[$ und $\sup_{y\in\mathcal{Y}} \|L''(y,\cdot)\|_\infty < \infty$ ist. Die obere Schranke der Influenzfunktion in H-Norm ist $2\,\lambda^{-1}\|k\|_\infty |L|_1$. In Supremumsnorm ergibt sich die obere Schranke dann zu $2\,\lambda^{-1}\|k\|_\infty^2 |L|_1$ aufgrund der Ungleichung aus Proposition 1.1.

[48] Tatsächlich stellt die Existenz der Influenzfunktion einen schwächeren Begriff als Gâteaux-Differenzierbarkeit dar, siehe Huber (1977).

Wie bei den Voraussetzungen der universellen Konsistenz und der *maxbias*-Robustheit können auch hier alle Voraussetzungen geprüft werden, ohne irgendetwas über die den Daten zugrundeliegende Verteilung P zu wissen. Ein Standardbeispiel für die Anwendung wäre hier $\mathcal{X} = \mathbb{R}^d$ für ein $d \in \mathbb{N}$, Gaußkern $k(x, \tilde{x}) = \exp\left(-\gamma^{-2}\|x - \tilde{x}\|_2^2\right)$, $x, \tilde{x} \in \mathcal{X}$, für ein $\gamma > 0$ und die logistische Verlustfunktion für Regression $L(y,t) := -\ln\left(4\exp(y-t)(1+\exp(y-t))^{-2}\right)$ beziehungsweise für Klassifikation $L(y,t) := \ln(1 + \exp(-yt))$. Diese Verlustfunktionen erfüllen die benötigten Voraussetzungen, führen allerdings nur zu konvexen Optimierungsproblemen (statt zu quadratischen Programmen mit Box-Constraints, die bei der Verwendung nichtglatter Verlustfunktionen wie beispielsweise der hinge-Verlustfunktion für Klassifikation oder der ε-insensitive-Verlustfunktion für Regression resultieren, siehe Tabelle 1.1). Es gibt jedoch Erweiterungen der Beweise zu Robustheitseigenschaften von Support Vector Machines auch für nichtglatte Verlustfunktionen, siehe Christmann & van Messem (2008), Christmann, Van Messem & Steinwart (2009) und Van Messem & Christmann (2010). Diese Erweiterungen auf den lokalisierten Fall zu übertragen, ist jedoch nicht Teil der vorliegenden Arbeit.

In der globalen, d. h. in der nichtregionalisierten Situation, kann die Influenzfunktion wie folgt umgeschrieben werden:

$$\mathrm{IF}(\delta_z; S, P) = \lim_{\varepsilon \searrow 0} \frac{f_{L^*,(1-\varepsilon)P+\varepsilon\delta_z,\lambda} - f_{L^*,P,\lambda}}{\varepsilon}.$$

Diese Neuformulierung wird genutzt, um eine Influenzfunktion für den zusammengesetzten Prädiktor, der in (3.1) definiert wurde, zu entwerfen. Zu beachten ist hierbei, dass der zusammengesetzte Prädiktor im Allgemeinen nicht Element eines Hilbertraums ist. Er ist jedoch Element von $L^\infty(P^{\mathcal{X}})$ auf $\mathcal{X}$ und somit ein Element eines Banachraums, sofern nur beschränkte Kerne verwendet werden, um die lokalen SVMs zu lernen. Somit kann Hampels Definition auch in der regionalisierten Situation eingesetzt werden. $\mathrm{IF}^{comp}(\delta_z; S, P)$, d. h. die Influenzfunktion des zusammengesetzten Prädiktors, sei wie folgt definiert:

Definition 4.12
Die Influenzfunktion des zusammengesetzten Prädiktors, wie in (3.1) konstruiert, ist (sofern der Grenzwert existiert) definiert als

$$\mathrm{IF}^{comp}(\delta_z; S, P) := \lim_{\varepsilon \searrow 0} \frac{f^{comp}_{L^*,(1-\varepsilon)P+\varepsilon\delta_z,\lambda} - f^{comp}_{L^*,P,\lambda}}{\varepsilon},$$

wobei $S : \mathcal{M}_1(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}}) \to L^\infty(P^{\mathcal{X}})$, $S(P) = f^{comp}_{L^*,P,\lambda}$.

In der regionalisierten Situation ist IF^{comp} selbst wieder eine Funktion, die eine Dirac-Verteilung δ auf $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$ auf einen Prädiktor in einem Banachraum abbildet,

d. h. $\mathrm{IF}^{comp}(\delta; S, P)(\cdot) \in L^\infty(P^{\mathcal{X}})$, sofern beschränkte Kerne eingesetzt werden, vgl. die Ungleichung aus Proposition 1.1. Es ist nun möglich zu zeigen, dass auch zusammengesetzte Prädiktoren, wie sie in dieser Arbeit in (3.1) vorgeschlagen werden, eine beschränkte Influenzfunktion besitzen. Für den Beweis wird die folgende Notation eingeführt:

$$\tilde{P}_{b,\varepsilon,z} := \begin{cases} (1-\varepsilon)P_b + \varepsilon\delta_z & , \text{ falls } \ z \in \mathcal{X}_b \times \mathcal{Y} \\ P_b & , \text{ sonst} \end{cases}.$$

Somit steht $\tilde{P}_{b,\varepsilon,z}$ für die Mischungsverteilung auf $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b \times \mathcal{Y}})$, falls die Support Vector Machine auf $\mathcal{X}_b$ von δ_z betroffen ist. Andernfalls ist $\tilde{P}_{b,\varepsilon,z} = P_b$. Diese Notation ist notwendig, um sicherzustellen, dass eine lokale SVM stets bezüglich eines Wahrscheinlichkeitsmaßes gelernt wird. Die lokale Influenzfunktion IF_b ist 0 in allen Fällen, in welchen $\tilde{P}_{b,\varepsilon,z} = P_b$, $b \in \{1, \ldots, B\}$. Für punktweise definierte Funktionen $g : U \to \mathbb{R}$ mit $U \supset \mathcal{X}_b$ sei $\|g\|_{\mathcal{X}_b\text{-}\infty} := \sup_{x \in \mathcal{X}_b} g(x)$, $b \in \{1, \ldots, B\}$; falls g nicht punktweise definiert ist, sei $\|g\|_{\mathcal{X}_b\text{-}\infty} := \inf \left\{ K \geq 0 \,\middle|\, |g| \leq K \ P_b^{\mathcal{X}_b}\text{-f. s.} \right\}$, $b \in \{1, \ldots, B\}$.

Theorem 4.13 (Existenz)
Sei für alle $b \in \{1, \ldots, B\}$ $\mathcal{X}_b$ ein vollständiger, separabler metrischer Raum, H_b der RKHS eines beschränkten und stetigen Kerns k_b sowie L eine konvexe und Lipschitz-stetige Verlustfunktion mit stetigen partiellen (Fréchet-)Ableitungen (bezüglich des letzten Arguments) $L'(y, \cdot)$ und $L''(y, \cdot)$ mit $\sup_{y \in \mathcal{Y}} \|L'(y, \cdot)\|_{\mathcal{X}_b\text{-}\infty} \in]0, \infty[$ und $\sup_{y \in \mathcal{Y}} \|L''(y, \cdot)\|_{\mathcal{X}_b\text{-}\infty} < \infty$, $b \in \{1, \ldots, B\}$. Dann existiert $\mathrm{IF}^{comp}(\delta_z; S, P)$ und ist beschränkt.

Sind die Voraussetzungen **(R1)** bis **(R4)** erfüllt, so ist sichergestellt, dass die Regionalisierungsmethode Regionen $\mathcal{X}_b$ aus dem Eingaberaum $\mathcal{X}$ bildet, die für dieses Theorem benötigt werden. Stetige Kerne sind offensichtlich auch messbar, die zugehörigen reproduzierenden Kern-Hilberträume sind separabel, siehe Steinwart & Christmann (2008, Lemma 4.33). Außerdem impliziert $\sup_{y \in \mathcal{Y}} \|L'(y, \cdot)\|_{\mathcal{X}_b\text{-}\infty} \in]0, \infty[$ bereits die Lipschitz-Stetigkeit von L mit $|L|_1 \neq 0$. Das ist nützlich für einen fairen Vergleich der Voraussetzungen zwischen den verschiedenen Theoremen zu Konsistenz (Theorem 4.1) und Robustheit (Theoreme 4.13 und 4.8).

Beweis zu Theorem 4.13. Um dieses Resultat zu beweisen, wird erneut der Prädik-

tor zerlegt:

$$\begin{aligned}
\mathrm{IF}^{comp}(\delta_z; S, P) &= \lim_{\varepsilon \searrow 0} \frac{f^{comp}_{L^*,(1-\varepsilon)P+\varepsilon\delta_z,\lambda} - f^{comp}_{L^*,P,\lambda}}{\varepsilon} \\
&= \lim_{\varepsilon \searrow 0} \frac{\sum_{b=1}^{B} w_b f_{b,L^*,\tilde{P}_{b,\varepsilon,z},\lambda_b} - \sum_{b=1}^{B} w_b f_{b,L^*,P_b,\lambda_b}}{\varepsilon} \\
&= \lim_{\varepsilon \searrow 0} \sum_{b=1}^{B} w_b \, \frac{f_{b,L^*,\tilde{P}_{b,\varepsilon,z},\lambda_b} - f_{b,L^*,P_b,\lambda_b}}{\varepsilon} \\
&= \sum_{b=1}^{B} w_b \, \lim_{\varepsilon \searrow 0} \frac{f_{b,L^*,\tilde{P}_{b,\varepsilon,z},\lambda_b} - f_{b,L^*,P_b,\lambda_b}}{\varepsilon} \\
&= \sum_{b=1}^{B} w_b \, \mathrm{IF}_b(\delta_z; S_b, P_b),
\end{aligned} \tag{4.12}$$

wobei S_b der statistische Operator auf $\mathcal{M}_1(\mathcal{X}_b \times \mathcal{Y})$ ist, d. h. $S_b : \mathcal{M}_1(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b \times \mathcal{Y}}) \to H_b$, $S_b(P_b) = f_{b,L^*,P_b,\lambda_b}$. Wegen Proposition 4.11 existiert die Influenzfunktion jeder lokalen SVM und ist beschränkt. Obige (endliche) Summe existiert somit und ist ebenfalls beschränkt. □

Die oberen Schranken der lokalen Influenzfunktionen, vgl. Christmann, Van Messem & Steinwart (2009), können herangezogen werden, um eine obere Schranke für die Influenzfunktion des zusammengesetzten Prädiktors herzuleiten. Jede lokale Influenzfunktion IF_b, $b \in \{1, \ldots, B\}$, ist durch ${\lambda_b}^{-1} \|k_b\|_{\mathcal{X}_b\text{-}\infty} |L|_1 \|P - \delta_z\|_{(\mathcal{X}_b \times \mathcal{Y})\text{-TV}}$ in H_b-Norm beschränkt, wobei $\|k_b\|_{\mathcal{X}_b\text{-}\infty} := \sup_{x \in \mathcal{X}_b} \sqrt{k_b(x,x)}, b \in \{1, \ldots, B\}$, H_b der RKHS von k_b und $\|\cdot\|_{(\mathcal{X}_b \times \mathcal{Y})\text{-TV}}$ die Totalvariationsnorm auf dem Raum der Verteilungen auf $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b \times \mathcal{Y}})$ ist[49]. Wegen der Ungleichung aus Proposition 1.1 erhält man $\|\mathrm{IF}_b(\delta_z; S_b, P_b)\|_{\mathcal{X}_b\text{-}\infty} \leq \|\mathrm{IF}_b(\delta_z; S_b, P_b)\|_{H_b} \|k_b\|_{\mathcal{X}_b\text{-}\infty}$.

Theorem 4.14 (Obere Schranke)
Unter den Voraussetzungen von Theorem 4.13 gilt:

$$\|\mathrm{IF}^{comp}(\delta_z; S, P)\|_\infty \leq \; 2 \; |L|_1 \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \; {\lambda_b}^{-1} \|k_b\|^2_{\mathcal{X}_b\text{-}\infty}, \tag{4.13}$$

wobei $\|\mathrm{IF}^{comp}(\delta_z; S, P)\|_\infty := \inf\left\{K \geq 0 \,\middle|\, |\mathrm{IF}^{comp}(\delta_z; S, P)(\cdot)| \leq K \; P^{\mathcal{X}}\text{-}f.\,s.\right\}$, $b \in \{1, \ldots, B\}$.

Beweis. Mittels Theorem 4.13 ist es möglich, direkt eine obere Schranke für die Influenzfunktion des zusammengesetzten Prädiktors herzuleiten. Eingesetzt werden

[49] Für Details hierzu siehe wiederum Denkowski, Migórski & Papageorgiou (2003, S. 158).

die Dreiecksungleichung und die Tatsache, dass die Gewichte w_b außerhalb von $\mathcal{X}_b$ verschwinden, $b \in \{1, \ldots, B\}$.

$$\begin{aligned}
\|\mathrm{IF}^{comp}(\delta_z; S, P)\|_\infty &= \left\| \sum_{b=1}^{B} w_b \ \mathrm{IF}_b(\delta_z; S_b, P_b) \ \right\|_\infty \\
&\leq \sum_{b=1}^{B} \|w_b \ \mathrm{IF}_b(\delta_z; S_b, P_b)\|_\infty \\
&\leq \sum_{b=1}^{B} \|w_b \ \mathrm{IF}_b(\delta_z; S_b, P_b)\|_{\mathcal{X}_b\text{-}\infty} \\
&= \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \ \|\mathrm{IF}_b(\delta_z; S_b, P_b)\|_{\mathcal{X}_b\text{-}\infty} \\
&\leq \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \ \|\mathrm{IF}_b(\delta_z; S_b, P_b)\|_{H_b} \|k_b\|_{\mathcal{X}_b\text{-}\infty} \\
&\leq \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \ {\lambda_b}^{-1} \|k_b\|^2_{\mathcal{X}_b\text{-}\infty} |L|_1 \|P - \delta_z\|_{(\mathcal{X}_b \times \mathcal{Y})\text{-TV}} \\
&\leq 2 \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty} \ {\lambda_b}^{-1} \|k_b\|^2_{\mathcal{X}_b\text{-}\infty} |L|_1.
\end{aligned}$$

Die letzte Ungleichung ist wiederum aufgrund der allgemeinen (und sehr groben) Abschätzung der Totalvariationsnorm für die Differenz zweier Verteilungen gültig. Die Ungleichung zuvor folgt aus Christmann, Van Messem & Steinwart (2009, Theorem 12) unter Anwendung des Darstellungssatzes für Support Vector Machines mit konvexen und Lipschitz-stetigen Verlustfunktionen, vgl. Christmann, Van Messem & Steinwart (2009, Theorem 7). □

Die Betrachtung der Abschätzung zeigt einen Zielkonflikt zwischen zwei wichtigen Eigenschaften von statistischen Methoden im Allgemeinen und Support Vector Machines im Speziellen: Eine Voraussetzung für die Konsistenz des zusammengesetzen Prädiktors in Theorem 4.1 ist, dass $\lambda_{(n_b,b)} \to 0$ für alle $b \in \{1, \ldots, B\}$. Betrachtet man jedoch Ungleichung (4.13), so wird deutlich, dass je kleiner λ_b, desto größer wird die obere Schranke für die Influenzfunktion. Dies zeigt den Zielkonflikt zwischen Konsistenz und Robustheit von Prädiktoren, die auf lokal gelernten SVMs basieren. Dieser Konflikt existiert für Support Vector Machines im Allgemeinen, nicht nur in der regionalisierten Situation. Das gleiche Problem tritt auch bei der Robustheit im Sinne des *maxbias* auf und ist hinlänglich bekannt bei sogenannten *ill-posed problems* im Allgemeinen und auch bei anderen Robustheitsbegriffen, siehe beispielsweise Hable & Christmann (2013).

Wie in Christmann & Steinwart (2004) dargestellt, ist es möglich, die Eigenschaften der Influenzfunktion nicht ausschließlich für eine Dirac-Verteilung δ_z zu zeigen,

sondern auch für beliebige Verteilungen Q auf $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$. Im Falle des hier behandelten zusammengesetzten Prädiktors ist dies ebenfalls möglich. Analog zu P und P_b sei

$$Q_b := \begin{cases} Q(\mathcal{X}_b \times \mathcal{Y})^{-1} \, Q_{|_{\mathcal{X}_b\times\mathcal{Y}}} & , \text{ falls } \; Q(\mathcal{X}_b \times \mathcal{Y}) > 0 \\ 0 & , \text{ sonst} \end{cases},$$

d. h. Q_b ist ein Wahrscheinlichkeitsmaß auf den Regionen $(\mathcal{X}_b \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}_b\times\mathcal{Y}})$, falls der Träger von Q Anteil an $\mathcal{X}_b \times \mathcal{Y}$ hat, und das Nullmaß sonst. Dies wird genutzt, um

$$\tilde{P}_{b,\varepsilon,Q} := \begin{cases} (1-\varepsilon)P_b + \varepsilon Q_b & , \text{ falls } \; Q_b \neq 0 \\ P_b & , \text{ sonst} \end{cases}$$

zu definieren. In allen Regionen $\mathcal{X}_b \times \mathcal{Y}$, in welchen $\tilde{P}_{b,\varepsilon,Q} = P_b$ gilt, ist die lokale Influenzfunktion IF_b somit 0.

Corollar 4.15
Unter den Voraussetzungen von Theorem 4.13 existiert $\mathrm{IF}^{comp}(Q; S, P)$ *und ist beschränkt mit oberer Schranke* $2\,|L|_1 \sum_{b=1}^{B} \|w_b\|_{\mathcal{X}_b\text{-}\infty}\, {\lambda_b}^{-1} \|k_b\|^2_{\mathcal{X}_b\text{-}\infty}$ *(gleichmäßig für alle Verteilungen* P *und* Q *auf* $(\mathcal{X} \times \mathcal{Y}, \mathfrak{B}_{\mathcal{X}\times\mathcal{Y}})$*).*

Beweis. Der Beweis verläuft analog zu den Beweisen der Theoreme 4.13 und 4.14 unter Berücksichtigung, dass die lokalen Influenzfunktionen existieren und beschränkt sind. □

Beispiel 4.16
Seien $d \in \mathbb{N}$, $\mathcal{X} = \mathbb{R}^d$, k_b *ein Gaußkern, d. h.* $k_b(x, x') = \exp\left(-\gamma_b^{-2}\|x - x'\|_2^2\right), \gamma_b > 0$, *für alle* $b \in \{1, \ldots, B\}$, *und sei* L *die logistische Verlustfunktion für Regression oder Klassifikation. Dann liefern Theorem 4.14 und Corollar 4.15 die gleichmäßige obere Schranke* $2 \sum_{b=1}^{B} \lambda_b^{-1}$ *für die Influenzfunktion des zusammengesetzten Prädiktors.*

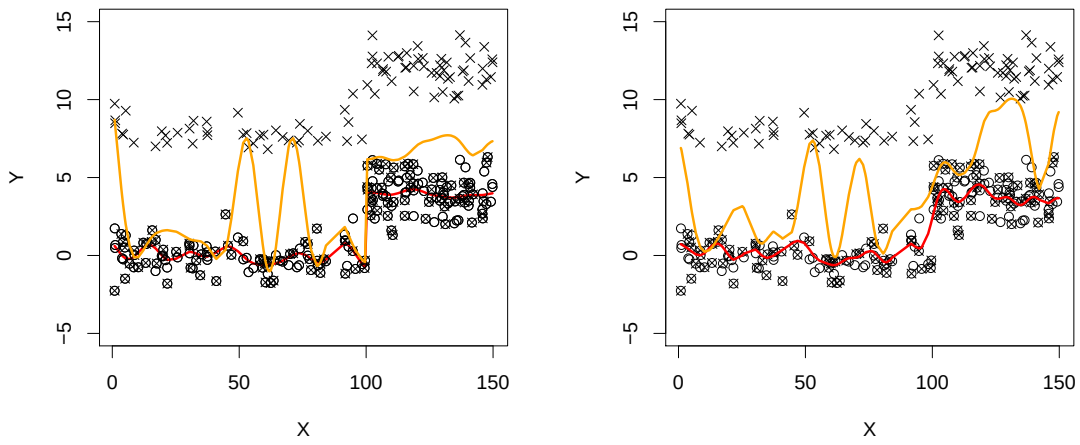
4.5 Vergleich der Robustheitsbegriffe

Robustheit im Sinne des *maxbias* und im Sinne der Influenzfunktion liefert jeweils gleichmäßige obere Schranken in dem Sinne, dass sie gültig sind für alle Verteilungen P und alle Gewichte, die **(W1)** und **(W2)**, d. h. $\sum_{b=1}^{B} w_b(x) = 1$ für alle $x \in \mathcal{X}$ und $w_b(x) = 0$ für alle $x \notin \mathcal{X}_b$ und für alle $b \in \{1, \ldots, B\}$, erfüllen. Für Robustheit im Sinne des *maxbias* muss – im Unterschied zur Influenzfunktion – die Voraussetzung, dass die Regionen $\mathcal{X}_b$, $b \in \{1, \ldots, B\}$, vollständig und die Verlustfunktionen differenzierbar sind, nicht erfüllt sein, um eine obere Schranke herleiten zu können. Andererseits nutzt der Beweis der Existenz der lokalen Influenzfunktionen, siehe

Christmann, Van Messem & Steinwart (2009, Theorem 10), einen Satz über implizite Funktionen in Banachräumen und benötigt die Vollständigkeitsvoraussetzung für $\mathcal{X}_b$ für alle $b \in \{1, \ldots, B\}$ und die Stetigkeit der Kerne k_b, um zeigen zu können, dass eine entsprechende Inverse existiert. Die Eigenschaften der Influenzfunktion sind insofern nur unter stärkeren Voraussetzungen zu beweisen; andererseits liefert die Influenzfunktion aber auch die differenzierbare Abhängigkeit des zusammengesetzten Prädiktors von den Daten und nicht nur die stetige Abhängigkeit wie bei Betrachtung des *maxbias*, mithin also eine stärkere Aussage. Für Anwendungszwecke mag jedoch Robustheit im Sinne des *maxbias* ein eingängigeres Konzept darstellen.

Eine weitere Vergleichsmöglichkeit besteht zwischen regionalisierter und nicht regionalisierter Situation. Im letzteren Fall gibt es nur eine Region (d. h. $B = B_n = 1$). Nutzt man diese Einsicht in (4.13) bzw. (4.11) und vergleicht die Ergebnisse mit Proposition 4.11 bzw. Christmann, Van Messem & Steinwart (2009, Theorem 12), ist zu erkennen, dass sich die Robustheit durch die Verwendung eines zusammengesetzten Prädiktors nicht verschlechtert im Vergleich zur Situation der Verwendung einer globalen Support Vector Machine. (Man beachte hierzu, dass $\|w_b\|_{\mathcal{X}_b\text{-}\infty}$ für gewöhnlich 1 ist, $b \in \{1, \ldots, B\}$. Andernfalls würde eine Region existieren, die keinen einzigen Punkt $(x, y) \in \mathcal{X} \times \mathcal{Y}$ für sich selbst besitzt, d. h. eine Region, die alle ihre Punkte mit wenigstens einer weiteren Region teilt. Dies erscheint unrealistisch als Ergebnis einer Regionalisierung.)

Dass die hergeleiteten oberen Schranken (als Preis für die Allgemeingültigkeit) in der Tat sehr grob sind, wird durch folgendes Beispiel zuzüglich Abbildung 4.1 illustriert. Gelernt werden insgesamt vier Funktionen, zwei globale SVMs (links) und



(a) global (b) lokal

Abbildung 4.1: Illustration zur Robustheit

zwei Prädiktoren auf Basis lokal gelernter SVMs (rechts): Jeweils eine anhand eines ungestörten Datensatzes (rot), jeweils eine zweite anhand eines Datensatzes aus einer Kontaminationsumgebung (orange). Jede der vier Funktionen wurde anhand von 200 Datenpunkten gelernt. Datenpunkte, die im ungestörten Datensatz enthalten waren, werden durch einen Kreis dargestellt; Datenpunkte, die im Datensatz aus der Kontaminationsumgebung enthalten waren, durch ein Kreuz. Datenpunkte, die in beiden Datensätzen vorhanden waren, sind dementsprechend an der Kombination von Kreis und Kreuz zu erkennen. Für das Beispiel wurden additive Fehler η unterstellt. $\mathcal{N}$ bezeichne eine Normalverteilung. Der ungestörte Datensatz besteht aus 100 Datenpunkten mit gleichverteilten $x \in [0, 100]$ und $y|x \sim 0$, $\eta|x \sim \mathcal{N}(0, 1)$ sowie 100 Datenpunkten mit gleichverteilten $x \in [100, 150]$ und $y|x \sim \sin(x) + 4$, $\eta|x \sim \mathcal{N}(4, 1)$. Für die Verteilung aus der Kontaminationsumgebung folgen im linken Teil 60 Datenpunkte der ursprünglichen Verteilung, 40 Datenpunkte sind um 8 nach oben verschoben; im rechten Teil folgen 51 Datenpunkte der ursprünglichen Verteilung, 49 Datenpunkte sind um 8 nach oben verschoben. Die Funktionen, die anhand der Datensätze aus der Kontaminationsumgebung gelernt wurden, verwenden den bzw. die gleichen Regularisierungsparameter λ bzw. (λ_1, λ_2) und jeweils den gleichen Gauß-Kern bzw. die gleichen Gauß-Kerne. Als Verlustfunktion wird stets die ε-insensitive-Verlustfunktion (mit $\varepsilon = 0.1$) genutzt. Somit liegen jeweils zwei mittels des *maxbias* vergleichbare Prädiktoren vor.[50] Als theoretische obere Schranken für den *maxbias* erhält man ca. 1286 im Fall der Prädiktoren auf Basis lokal gelernter SVMs (tatsächlicher maximaler Unterschied ca. 8) und 89 im Fall der global gelernten SVM (tatsächlicher maximaler Unterschied ca. 7).[51]

Abschließend sei bemerkt, dass weitere Robustheits- und Stabilitätsbegriffe existieren, die in dieser Arbeit jedoch nicht untersucht wurden; siehe beispielsweise Xu, Caramanis & Mannor (2009) oder Hable & Christmann (2011).

[50] Die Berücksichtigung unterschiedlicher Regularisierungsparameter und unterschiedlicher Kerne wird – in der globalen Situation – in Christmann, Xiang & Zhou (2018) behandelt.

[51] Hier wird der Zielkonflikt zwischen Genauigkeit und Robustheit besonders deutlich: Die lokale Methode ist genauer (gemessen am *root mean squared error*), weist jedoch eine größere obere Schranke für den *maxbias* auf.

Kapitel 5

Testrechnungen

In diesem Kapitel werden anhand exemplarischer Rechnungen die theoretischen Resultate empirisch veranschaulicht. Die verschiedenen Beispiele gehen dabei auf unterschiedliche Aspekte der theoretischen Arbeit ein und heben einzelne Aspekte besonders hervor. In den Ausführungen beschreibe $\mathcal{N}$ eine Normalverteilung, $\mathcal{U}$ eine stetige Gleichverteilung. Die weiteren Verteilungen ergeben sich aus ihren Bezeichnungen (Cauchy-Verteilung, Exponentialverteilung usw.). Nicht eingesetzt wurde jeweils die auch in Chang, Guo, Lin & Lu (2010) benannte Option, (hinreichend) reine Regionen, die durch den Baum gefunden wurden, nicht mehr zusätzlich dem Training einer SVM zuzuführen. Die Testrechnungen nutzen diesen zusätzlich möglichen zeitlichen Vorteil also nicht aus. Alle Zeitangaben in diesem Kapitel sind in Sekunden.

5.1 Simulationsbeispiel zur Klassifikation

Untersuchungen zur Berechenbarkeit[52] (computability) können mithilfe des R-Pakets `liquidSVM` (Steinwart & Thomann, 2017; R Core Team, 2018) durchgeführt werden, welches die Regionen durch Voronoi-Diagramme bildet, d. h. die Regionen überlappen sich nicht. Im Folgenden werden fünf Szenarien für einen Beispieldatensatz zur binären Klassifikation dargestellt. Die Klassifikationsgrenze verläuft nichtlinear in den zwei verwendeten Dimensionen.

1. Eine globale SVM basierend auf `liquidSVM`, ohne dass Regionen gebildet werden;

2. drei lokale SVMs basierend auf `liquidSVM` auf händisch vorgegebenen Regionen;

[52] Dieser Abschnitt basiert auf dem eingereichten, aber noch nicht erschienenen Aufsatz Dumpert (2019a).

3. soviele lokale SVMs, wie `liquidSVM` bei der internen Regionalisierung findet (`partition_choice=5`);

4. soviele lokale SVMs, wie `liquidSVM` bei der internen Regionalisierung findet (`partition_choice=6`);

5. ein zusammengesetzter Prädiktor basierend auf lokalen SVMs berechnet mit `liquidSVM` auf Regionen, die zuvor mittels eines Baumes mit `rpart` ermittelt wurden; für Details zu `rpart` siehe Therneau & Atkinson (2018).

Die Stichprobe enthält die verfügbare Information über die Verteilung der blauen und roten Klasse. Die wahre Verteilung der beiden Klassen ist in Abbildung 5.1 dargestellt. Um vergleichbare Werte für die Laufzeiten zu erhalten, wurde die Parallelisierung für dieses Beispiel abgeschaltet. (Damit wurde ein Vorteil der zusammengesetzten Prädiktoren, nämlich die einfach umzusetzende Parallelisierbarkeit, nicht genutzt. Vorteile entstehen also allenfalls aufgrund der geringeren Anzahl an Datenpunkten, die zum Lernen je Region verwendet werden.) Die fünf Szenarien wurden hinsichtlich der benötigten Laufzeit (time; gemessen mit `proc.time()` (user)) und der erreichten Genauigkeit (accuracy; Anzahl der korrekt klassifizierten Datenpunkte geteilt durch die Anzahl der Datenpunkte im Testdatensatz) auf einem Testdatensatz mit 25000 Datenpunkten verglichen. Es gab 10 Durchläufe (verschiedene Seeds) pro Szenario und untersuchter Trainingsdatensatzgröße (750, 10000, 50000). Die Berechnungen wurden auf einer (Intel Xeon CPU (E5-2640 v4), 2.4 GHz, 16 GB RAM, 64-bit Windows 7)-Maschine mit R 3.5.1 durchgeführt.

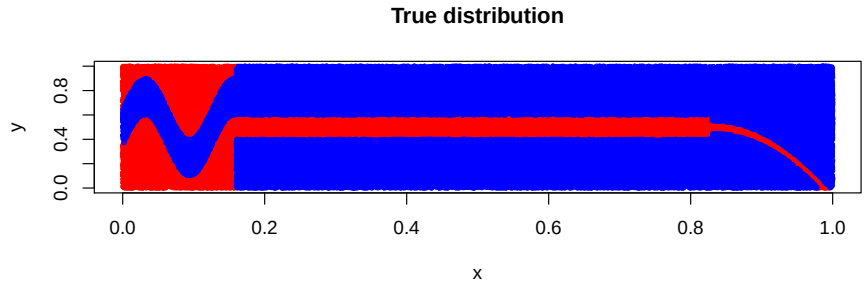
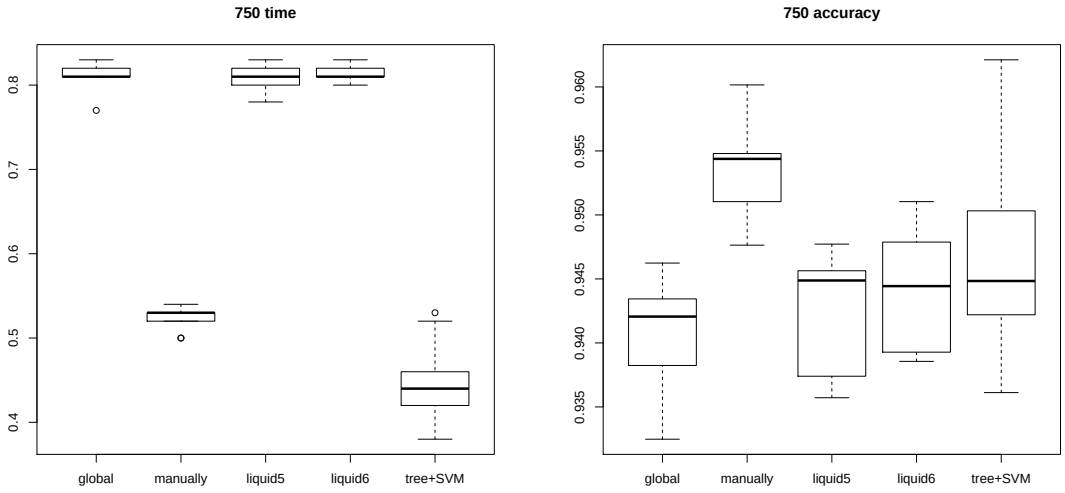


Abbildung 5.1: Wahre Verteilung der beiden Klassen (rot und blau)

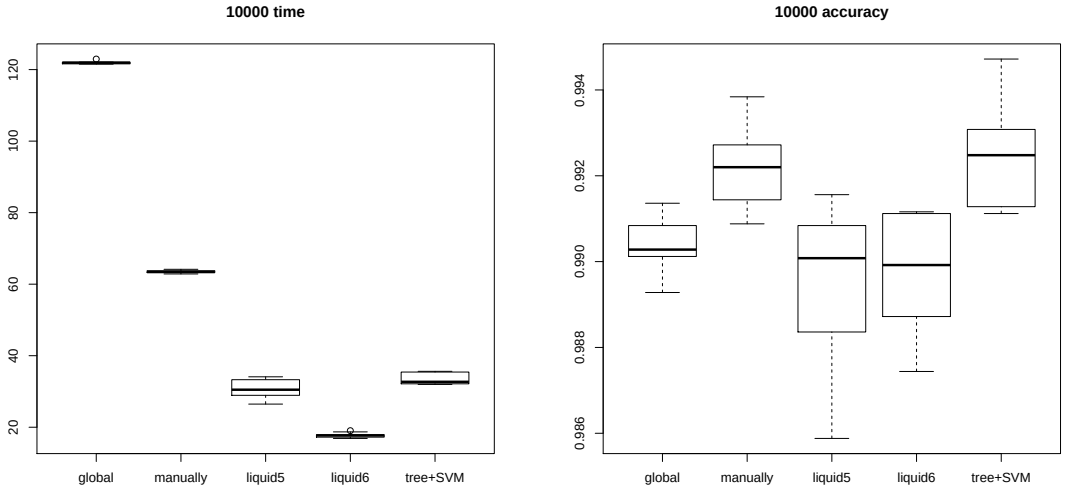
Die Resultate sind in sechs Boxplots zusammengefasst (Abbildungen 5.2, 5.3, und 5.4).

Die Simulationen zeigen für dieses Beispiel, dass der Ansatz der Regionalisierung und die Verwendung eines zusammengesetzten Prädiktors mindestens zu gleich guten Genauigkeiten, wie eine globale SVM sie erreicht, führt. Ob es auch einen Vorteil im Hinblick auf die Laufzeit gibt, hängt davon ab, wie schnell der Baum die Regionalisierung durchführt und wie hoch die Overheadkosten durch die Aufteilung der



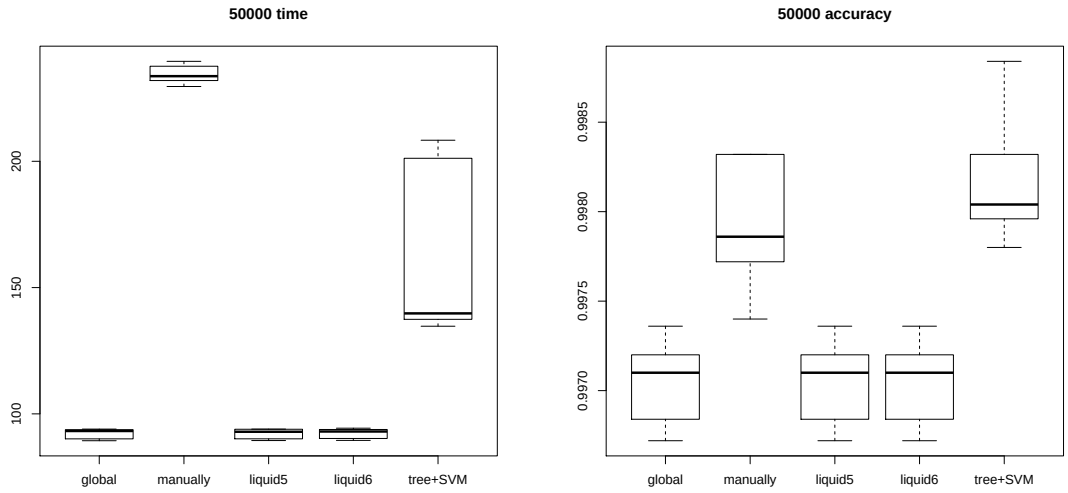
(a) time (b) accuracy

Abbildung 5.2: Zusammenfassung der Resultate für 750 Trainingspunkte



(a) time (b) accuracy

Abbildung 5.3: Zusammenfassung der Resultate für 10000 Trainingspunkte



(a) time (b) accuracy

Abbildung 5.4: Zusammenfassung der Resultate für 50000 Trainingspunkte

Datensätze etc. sind. Ob eine Aufteilung mithilfe eines Baumes oder auf Basis von Voronoi-Diagrammen besser geeignet ist, lässt sich auf Grundlage dieser Simulation nicht entscheiden. Außerdem ist nicht klar, warum die globale SVM aus `liquidSVM` im Szenario mit 50000 Trainingsdatenpunkten (ohne Regionalisierung und ohne Parallelisierung) so schnell gelernt werden kann.

5.2 Simulationsbeispiel zur Regression

Ähnlich wie beim Simulationsbeispiel zur Klassifikation, dieses Mal jedoch in Form einer Medianregression, wird exemplarisch gezeigt, welche Vorzüge die Nutzung von Prädiktoren bestehend aus lokal gelernten Support Vector Machines bietet.[53] Betrachtet werde als wahrer funktionaler Zusammenhang zwischen dem eindimensionalen X und dem ebenfalls eindimensionalen Y die Abbildung

$$y = f(x) = \begin{cases} 0.7x & , \ 0 \leq x \leq 3 \\ 10 + x + \frac{1}{100} \sin(10x)x^4 & , \ 3 < x \leq 6 \\ 5 & , \ 6 < x \leq 30 \\ -20 - 0.4\,(x - 27)^2 & , \ 30 < x \leq 33 \end{cases} .$$

Die Datenpunkte, die beobachtet werden und somit im Trainingsdatensatz zur Ver-

[53] Dieses Beispiel wurde bereits auf verschiedenen Fachtagungen als Motivation für den in dieser Arbeit vorgestellten Ansatz präsentiert.

fügung stehen, sind jedoch (additiv) fehlerbehaftet mit

$$\varepsilon|x \sim \begin{cases} \mathcal{U}(-1,1) & , \ 0 \leq x \leq 3 \\ \text{Exp}(0.5) - \frac{\ln 2}{0.5} & , \ 3 < x \leq 6 \\ \text{Cauchy}(0,1) & , \ 6 < x \leq 30 \\ \mathcal{N}(0,4) & , \ 30 < x \leq 33 \end{cases} .$$

Insbesondere treten Fehlerverteilungen mit nichtexistierenden Momenten und schiefe Fehlerverteilungen auf. Zunächst sollen 600 Datenpunkte im Trainingsdatensatz enthalten sein. Trainingsdaten (schwarz) und wahrer Zusammenhang (rot) sind in Abbildung 5.5 dargestellt. Es ist zu beachten, dass hier nicht alle Trainingsdatenpunkte abgebildet werden, da insbesondere aufgrund des zum Teil enthaltenen additiven Cauchy-Fehlers eine sehr große Streuung auf der Ordinate vorhanden ist. Alle Datenpunkte abzubilden, würde die Aussagekraft der Abbildung deutlich schmälern. Abbildung 5.6 zeigt die Anwendung einer auf Basis dieser 600 Trainingsdatenpunk-

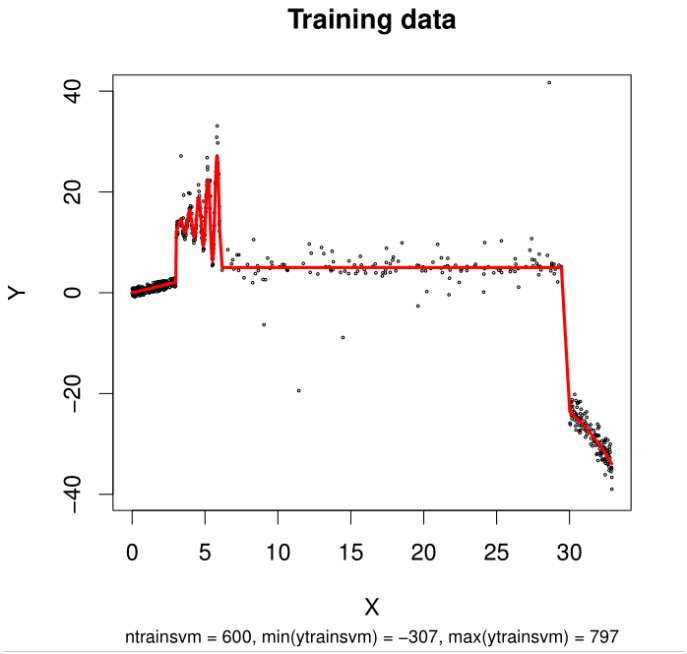


Abbildung 5.5: Trainingsdaten und wahrer Zusammenhang

te gelernten Support Vector Machine auf größtenteils ebenfalls abgebildete 36000 Testdatenpunkte als zusätzliche blaue Kurve. Es ist deutlich zu erkennen, dass die vorhandenen 600 Trainingsdatenpunkte nicht ausreichen, um den zugrundeliegenden funktionalen Zusammenhang in dieser Fehlersituation mit einer SVM zu erlernen. Das notwendige Ausmitteln zwischen den unterschiedlichen Charakteristika in den vier verschiedenen Bereichen des Datensatzes wird offenbar.

Wird jedoch eine Regionalisierungsmethode eingesetzt, um geeignete Regionen zu finden (hier: wiederum ein Baum), auf denen im Anschluss lokale SVMs gelernt werden, so stellt sich das in Abbildung 5.7 dargestellte – deutlich bessere – Ergebnis

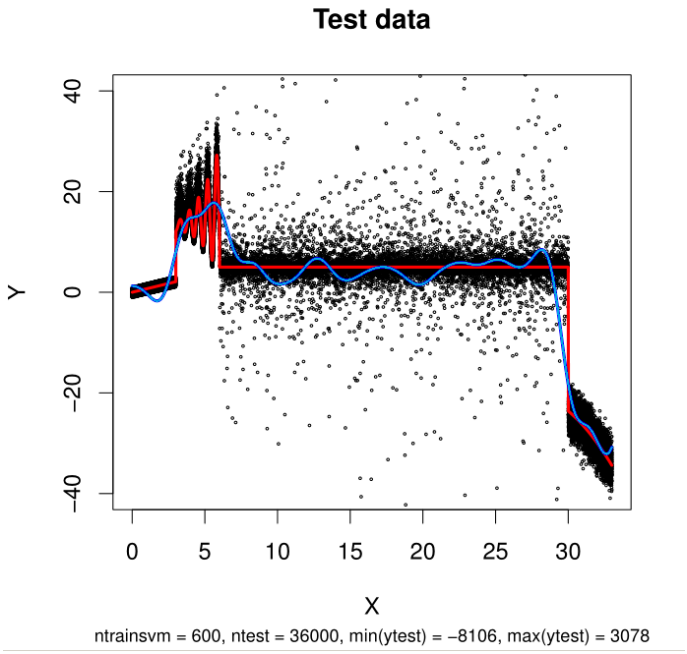


Abbildung 5.6: Testdaten, wahrer Zusammenhang und globale SVM

ein. Außer im dritten Bereich (konstant, additiver Cauchy-Fehler) wird die zugrundeliegende Verteilung (rot) durch den zusammengesetzten Prädiktor (blau) sehr gut erfasst; und das, obwohl pro Region (offensichtlich wurden vier Regionen durch den Baum gebildet) im Mittel nur 150 Datenpunkte zum Trainieren der lokalen SVM zur Verfügung standen. Für einen aufgrund äußerer Gegebenheiten (Erhebung bereits abgeschlossen, Experiment nicht mehr reproduzierbar, Kosten) oder technischer Einschränkungen (Speicher, Laufzeit) festen Umfang des Trainingsdatensatzes wird anhand dieses Beispiels offenbar, dass ein aus lokal gelernten Support Vector Machines zusammengesetzter Prädiktor hinsichtlich der Genauigkeit bessere Ergebnisse liefert als eine global gelernte SVM. Zum Vergleich weisen die Abbildungen 5.8 und 5.9 die Ergebnisse bei größeren Umfängen des Trainingsdatensatzes aus. Während für 6000 Trainingsdatenpunkte der lokale Ansatz den zugrundeliegenden funktionalen Zusammenhang (trotz der enthaltenen Fehler) annähernd perfekt erlernt hat, zeigt die globale SVM weiterhin unerwünschtes Verhalten (insbesondere im zweiten und dritten Abschnitt der Daten): Sie kann weder das Oszillieren mit ansteigender Amplitude noch den konstanten Teil brauchbar approximieren. Um die Support Vector Machines zu lernen, wurde jeweils das R-Paket `liquidSVM` (Steinwart & Thomann, 2017; R Core Team, 2018), jedoch ohne Einsatz der dort intern implementierten Regionalisierungsmethode (Voronoi-Diagramme), verwendet.

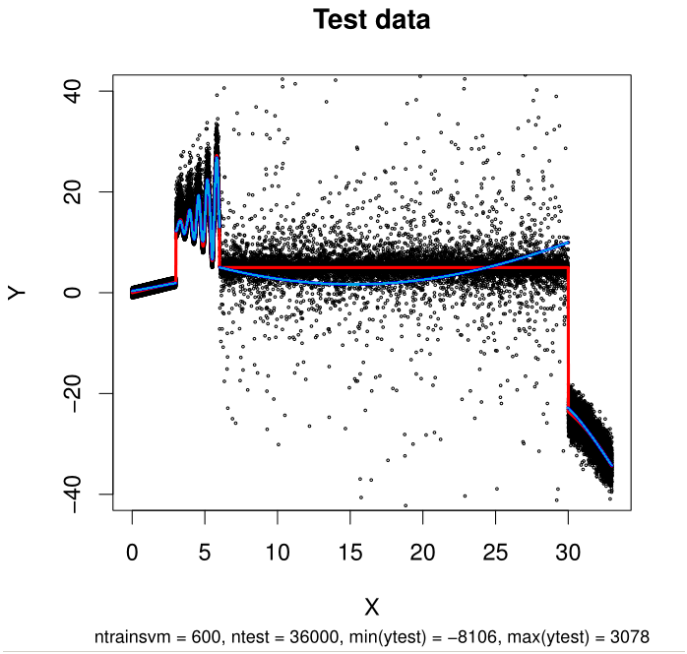
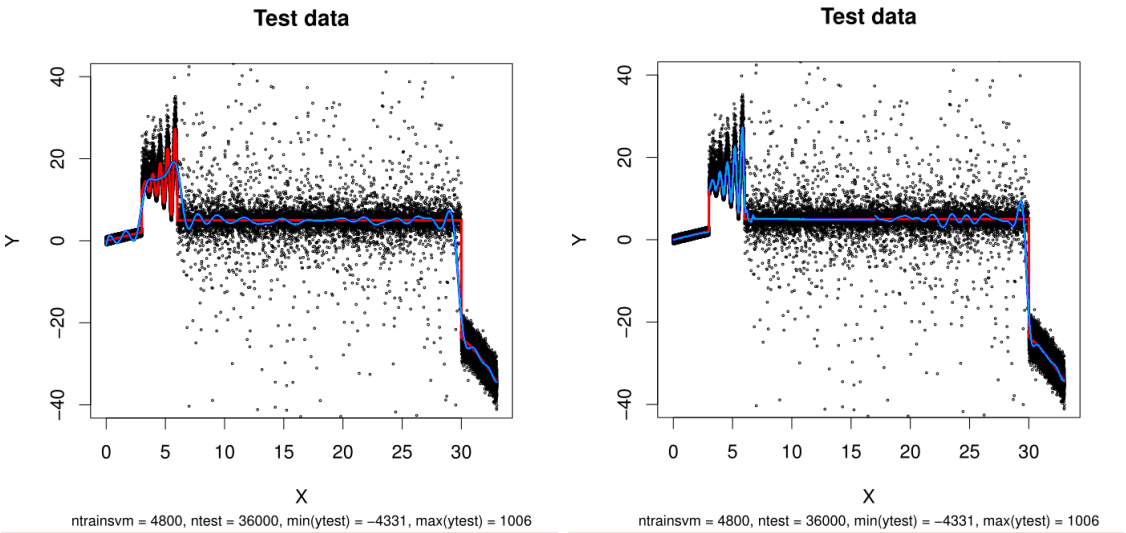
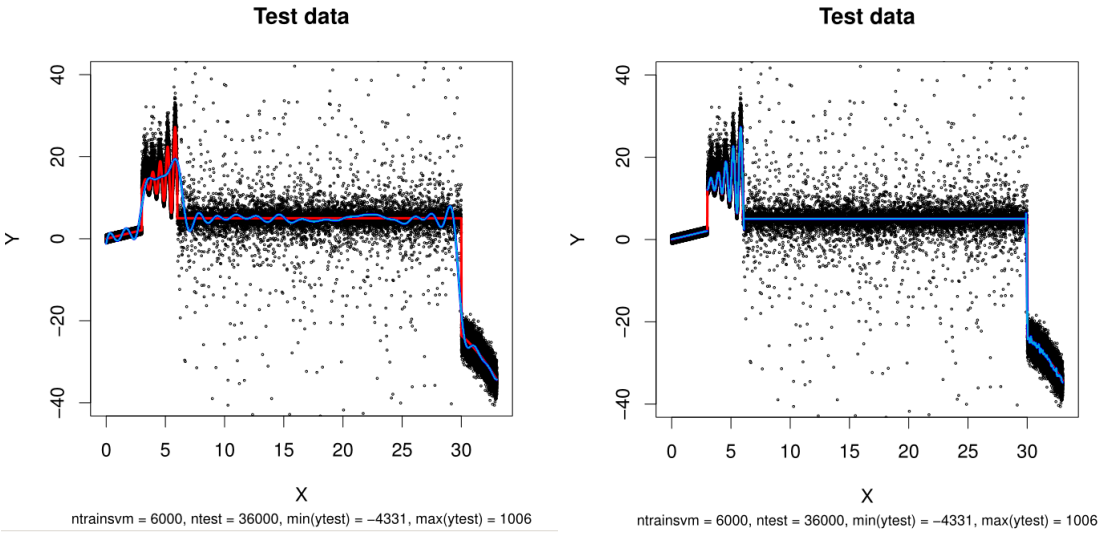


Abbildung 5.7: Testdaten, wahrer Zusammenhang und zusammengesetzter Prädiktor auf Basis lokaler SVMs ($n_{train} = 600$)



(a) global (b) lokal

Abbildung 5.8: Testdaten, wahrer Zusammenhang und Prädiktoren ($n_{train} = 4800$)



(a) global (b) lokal

Abbildung 5.9: Testdaten, wahrer Zusammenhang und Prädiktoren ($n_{train} = 6000$)

5.3 Simulationsbeispiel zur Regression in höheren Dimensionen

Nach dem empirischen Aufzeigen der prinzipiellen Möglichkeit und des prinzipiellen Nutzens des Einsatzes zusammengesetzter Prädiktoren auf Basis lokal gelernter Support Vector Machines in den vorangegangenen Abschnitten werden nun Daten in höherer Dimension und in jeweils verschiedener Anzahl betrachtet. Wiederum handelt es sich um simulierte Daten. Dabei gilt für die auftretenden Variablen: $X_1 \sim \mathcal{U}(-10, 10)$; $X_2 \sim \delta_0$ für die erste Hälfte der jeweils zur Verfügung stehenden Beobachtungen und $X_2 \sim \delta_4$ für die zweite Hälfte; $X_3 \sim \frac{1}{6}(\delta_1 + \delta_2 + \delta_3 + \delta_4 + \delta_5 + \delta_{99})$; $X_4 \sim \frac{1}{11}\sum_{i=0}^{10} \delta_i$; $X_7 \sim \text{Cauchy}(0, 1)$; $X_{12} \sim \mathcal{N}(0, 1)$; sowie $X_v \sim \mathcal{U}(-10, 10)$ für $v \in \{5, 6, 8, 9, 10, 11, 13, 14, 15\}$. Insgesamt liegen also 15 erklärende Variablen vor. Das Muster selbst, d. h. die Variable Y, für die die Regression durchzuführen ist, besitzt zu diesen aber einen funktionalen Zusammenhang, der nur von X_1 bis X_4 abhängt. Alle übrigen Variablen bilden Rauschen ab. Für die wahren Werte, mit welchen am Ende der Simulation die Vorhersagen verglichen werden, gilt:

$$Y = \sin(X_1) + X_2^3 + \frac{1}{2}X_2X_3 + X_1X_4 + X_2X_4.$$

In den Trainingsdaten sind jedoch nicht die wahren Y-Werte enthalten, sondern die wahren Y-Werte mit additivem Cauchy(0,1)-Fehler. Da das Muster im Vorfeld zwar in der Simulation, jedoch nicht in der Praxis bekannt ist und auch die Interaktionen dort nicht a priori zwingend aus einer Theorie ableitbar sind, wurden die Simulationen mit der Formel

```
yverr ~ x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15
```

durchgeführt, wobei `yverr` für die (verrauschten) Y-Werte mit Cauchy-Fehler steht.

Für diesen Abschnitt wurde untersucht, wie sich der *Root Mean Squared Error* (RMSE) und die Laufzeit für unterschiedliche Anzahlen an Trainingsdaten (z. T.[54] auch im Unterschied zum Einsatz von nur einer globalen SVM) verhalten. Für die Anzahl an Trainingsdatenpunkten wurde

$$n \in \{5000, 10000, 15000, 25000, 125000, 500000, 1000000\}$$

betrachtet; der Testdatensatz umfasste jeweils 20000 Datenpunkte. Für die Regionalisierung wurde im ersten Schritt ein Baum gelernt. Verwendet wurde hierfür das Paket `rpart` (Therneau & Atkinson, 2018). Als Parameter wurden hier zusätzlich eingestellt: `cp=0`, `minbucket=1000` und `xval=0`. Für die anschließende Bestimmung der lokalen Support Vector Machines fand das Paket `liquidSVM` (Steinwart & Thomann, 2017) mit der Funktion `svmQuantileRegression` und den Einstellungen `weights = c(0.5)`, `clipping = 0`, `do.select = TRUE`, `partition_choice = 0`, `threads = 0`, `random_seed = SEED`, `useCells = FALSE`, `grid_choice = 0` Verwendung. Die Berechnungen (inkl. die Aufteilungen des Datensatzes in Trainings- und Testdaten) wurden für $SEED \in \{2019, 2020, 2021\}$ und für jeden Seed insgesamt dreimal durchgeführt. Die Zeitmessung erfolgte mittels der Funktion `proc.time()`. Die eingesetzte R-Version war R 3.5.1 (R Core Team, 2018) auf einem Arbeitsplatz-PC mit Windows 7 (64-Bit) als Betriebssystem, einem Prozessor des Typs Intel Xeon CPU E5-2640 v4 mit 2.4 GHz Taktung und 16 GB RAM. Die Grafiken dieses Abschnitts zeigen die Ergebnisse der Simulationsstudie. Zunächst sei in Abbildung 5.10 der RMSE für den auf lokalen SVMs basierenden Prädiktor (Dreieck mit Spitze nach unten) und die globale SVM (Kreis mit Kreuz) vergleichend aufgezeigt. Der RMSE des auf lokalen SVMs basierenden Prädiktors ist stets geringer als der der globalen SVM; der Vorzug des lokalen Lernens, auf lokale Charakteristika des Datensatzes eingehen zu können, tritt also in Erscheinung. Betrachtet man die Laufzeiten, so ist in den Abbildungen 5.11, 5.12 (exemplarisch für 5000 und 25000 Trainingsdatenpunkte) und 5.13 zum einen zu erkennen, dass die Regionalisierungsmethode kaum Zeit in Anspruch nimmt. Zum anderen wird der Vorteil des lokalisierten Ansatzes auch im Hinblick auf die Laufzeit offensichtlich.

Dass diese Laufzeitvorteile aus der reduzierten Anzahl an Datenpunkten in den einzelnen Regionen resultieren, folgt aus der Komplexität von Support Vector Machines. Betrachtet werden daher in diesem Abschnitt auch noch die Entwicklung

[54] Diese Einschränkung ist erforderlich, da die Berechnung der Vergleichswerte für globale SVMs ab einer bestimmten Anzahl an Datenpunkten nicht mehr möglich war (Arbeitsspeicher) oder nicht mehr sinnvoll erschien (Laufzeit).

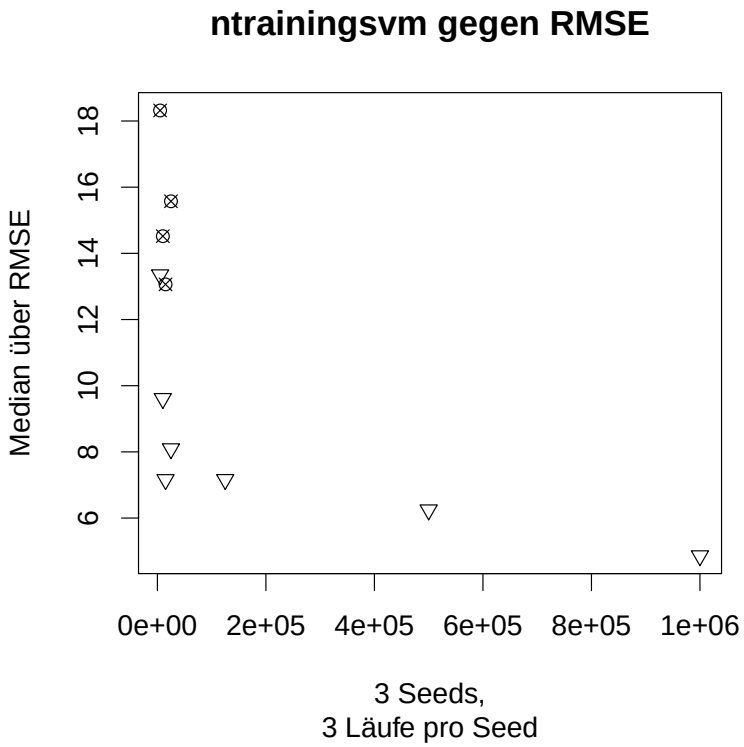
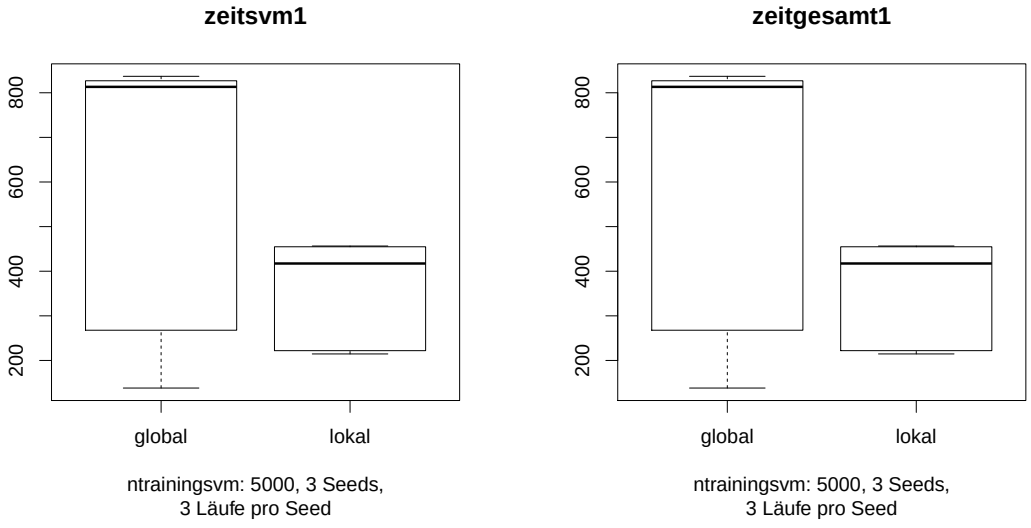
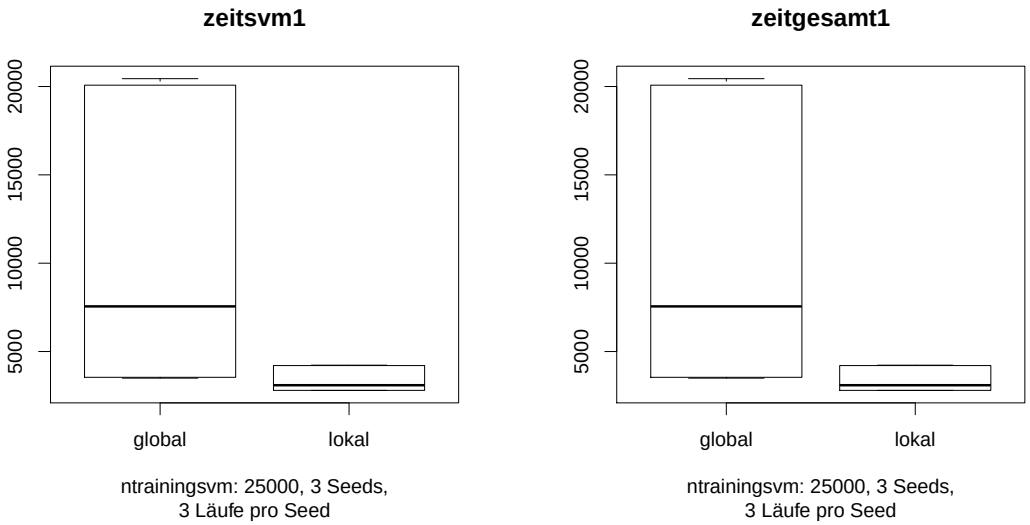


Abbildung 5.10: RMSE im Vergleich



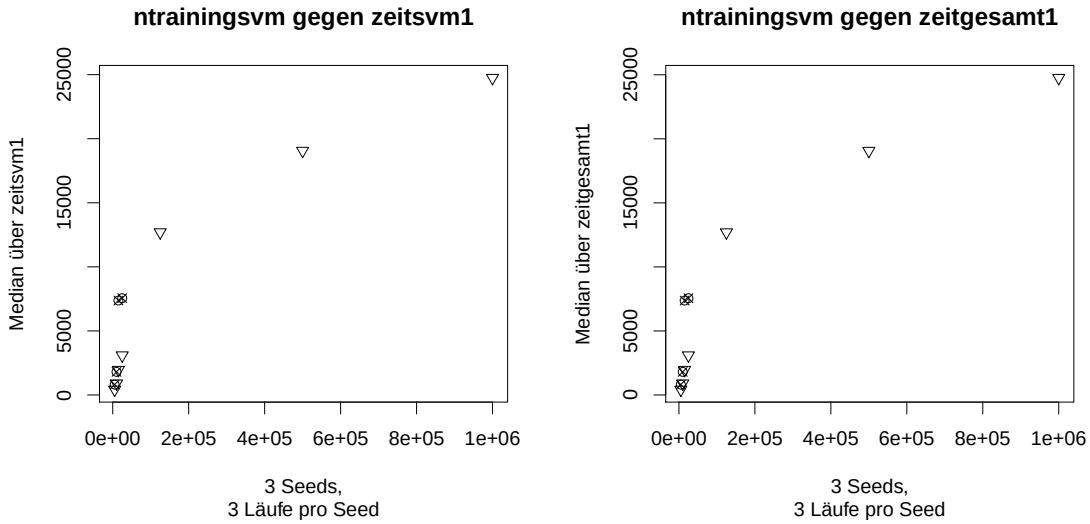
(a) SVM (b) gesamt

Abbildung 5.11: Laufzeiten im Vergleich für 5000 Datenpunkte



(a) SVM (b) gesamt

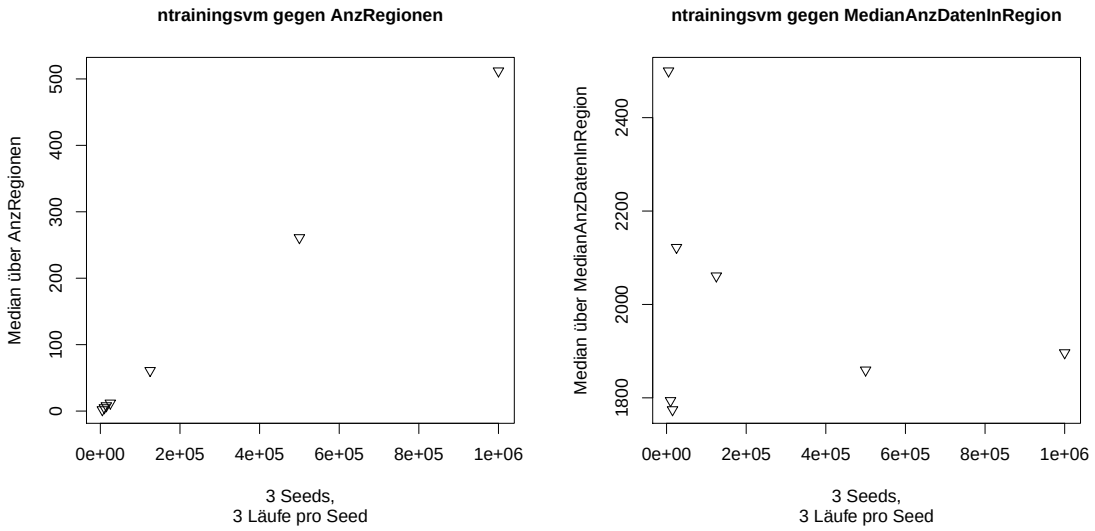
Abbildung 5.12: Laufzeiten im Vergleich für 25000 Datenpunkte



(a) SVM (b) gesamt

Abbildung 5.13: Laufzeiten im Vergleich zur Größe des Trainingsdatensatzes

der Anzahl der Regionen sowie minimale, mittlere und maximale Anzahl an Trainingsdatenpunkten in den Regionen (jeweils in Abhängigkeit von der Größe des Trainingsdatensatzes). Die Ergebnisse werden in den Abbildungen 5.14 und 5.15 dargestellt.



(a) Anzahl der Regionen

(b) Median der Anzahl der Datenpunkte in den Regionen

Abbildung 5.14: Betrachtungen der Regionen 1

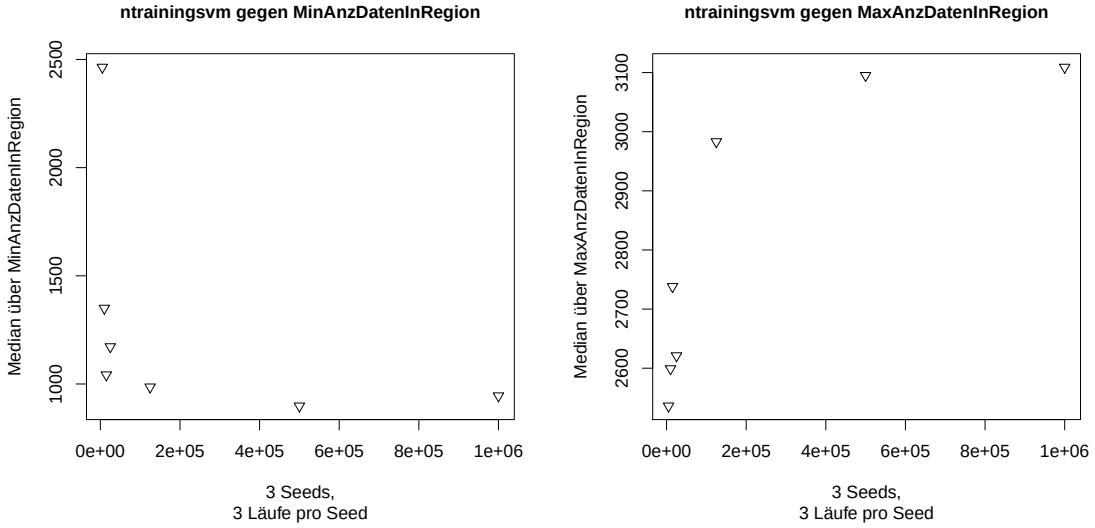
5.4 Bayern

Nachrichtlich wird hier noch mitgeteilt, dass das Lernen eines zusammengesetzten Prädiktors auf dem Datensatz wie in Abschnitt 5.3 konstruiert und mit den dort genannten Einstellungen, jedoch mit 13 Millionen Beobachtungszeilen[55], bei Einsatz des R-Pakets `liquidSVM`, parallelisiert mit 32 Kernen und auf einer Maschine mit Ubuntu 14.04.5 LTS (64-Bit) als Betriebssystem, Prozessoren des Typs Intel Xeon CPU E5-2690 mit 2.9 GHz Taktung und 377 GB RAM mit R 3.4.4 ca. sieben Stunden dauert. Insgesamt wurden dabei durch den Baum 6353 Regionen gebildet. Die minimale Anzahl an Datenpunkten, die zum Lernen einer lokalen SVM auf einer Region zur Verfügung standen, betrug 1397, der Median 1994 und die maximale Anzahl 3091.

5.5 Klassifikation anhand des SUSY-Datensatzes

Der SUSY-Datensatz (Baldi, Sadowski & Whiteson, 2014) aus dem UCI-Repository (Dua & Graff, 2017), einer der größten dort vorhandenen Datensätze für Klassifikati-

[55] Einwohnerzahl Bayerns, siehe Bayerisches Landesamt für Statistik (2019)



(a) Minimum der Anzahl der Datenpunkte in den Regionen

(b) Maximum der Anzahl der Datenpunkte in den Regionen

Abbildung 5.15: Betrachtungen der Regionen 2

on, wird im Folgenden herangezogen, um zusätzlich zu den in den vorangegangenen Abschnitten beschriebenen Simulationen auch Vergleiche auf einem frei verfügbaren Referenz-Datensatz liefern zu können. Dabei wurde einerseits die in dieser Arbeit beschriebene Methode, andererseits ein Random Forest (Breiman, 2001; Athey, Tibshirani & Wager, 2019) zu Vergleichszwecken gerechnet. Der SUSY-Datensatz enthält fünf Millionen Beobachtungen zu jeweils 18 erklärenden und einer zu erklärenden Variablen. Fehlende Werte sind nicht vorhanden.[56] Die Berechnungen wurden auf einer Maschine mit Ubuntu 14.04.5 LTS (64-Bit) als Betriebssystem, Prozessoren des Typs Intel Xeon CPU E5-2690 mit 2.9 GHz Taktung und 377 GB RAM mit R 3.4.4 auf 10 Kernen (threads=10 für den Random Forest und die SVM) durchgeführt. Für das Setting des zusammengesetzten Prädiktors kamen erneut `rpart` und `liquidSVM` (dieses Mal zur Klassifikation mit `mcSVM()`) zum Einsatz. Die Einstellungen bei `liquidSVM` blieben unverändert, erlaubten insbesondere wieder paralleles Rechnen auf 10 Kernen sowie die Nutzung des voreingestellten Gitters. Bei `rpart` wurden folgende Spezifikationen vorgenommen: `cp = -1`, `minbucket = 1500`, `minsplit = 1` und `xval = 0`. Um brauchbare Regionen zu finden, wurde der Baum darüber hinaus – und entgegen der ursprünglichen Intention – mit der gleichen Anzahl an Trainingsdaten gelernt wie anschließend die SVM. Die vier Millionen Trainingsdatenpunkte wurden durch zufälliges Ziehen mit Zurücklegen aus dem gesamten Datenmaterial gezogen, die Trainingsdatenpunkte für die SVM als einfache Zufallsstichprobe aus dem gesamten Datenmaterial; der Testdatensatz bestand aus den eine Million nicht zum Trainieren der SVM genutzten Datenpunkten. Beim Random Forest,

[56] Weitere Details sind auf der Website des UCI Machine Learning Repository einsehbar.

gerechnet mit den Standardeinstellungen im R-Paket `ranger` (Wright & Ziegler, 2017), wurde ebenso vorgegangen, der vorgelagerte Schritt für den Baum entfiel jedoch. Die Trainingsdatensätze bestanden jeweils aus vier Millionen Datenpunkten, der Testdatensatz jeweils aus einer Million Datenpunkten. Die Berechnungen (inkl. die Aufteilungen des Datensatzes in Trainings- und Testdaten) wurden für $SEED \in \{2019, 2020, 2021\}$ und für jeden Seed insgesamt dreimal durchgeführt. Die Zeitmessung erfolgte mittels der Funktion `proc.time()`. Die Ergebnisse der Durchläufe zeigen einen klaren Zeitvorteil für die in dieser Arbeit vorgeschlagene Methode gegenüber dem Random Forest bei vergleichbarer Genauigkeit (beide bei ca. 80 %).[57] Die Anzahl der Regionen für den vorgeschlagenen zusammengesetzten Prädiktor lag bei ungefähr 2000, der Medianwert für die Anzahl der in den Regionen vorhandenen Trainingsdatenpunkte bei ca. 1800 (Minimum im Schnitt bei ca. 1400 Datenpunkten, Maximum bei ca. 43000, arithmetisches Mittel bei ca. 2000, Standardabweichung bei ca. 1000 Datenpunkten). Die Abbildung 5.16 veranschaulicht die Ergebnisse hinsichtlich der Laufzeiten, Abbildung 5.17 hinsichtlich der Genauigkeit.

(a) RF und SVM alleine

(b) RF und SVM zzgl. Baum

Abbildung 5.16: Laufzeiten im Vergleich

[57] Die Verschiebung der Boxplots in Abbildung 5.17 sieht dramatischer aus als sie ist. Auch die anderen Gütemaße wie Sensitivität (bei ca. 87 %), Spezifität (bei ca. 71 %), positiver bzw. negativer Vorhersagewert (bei ca. 78 % bzw. bei ca. 82 %) usw. sind vergleichbar.

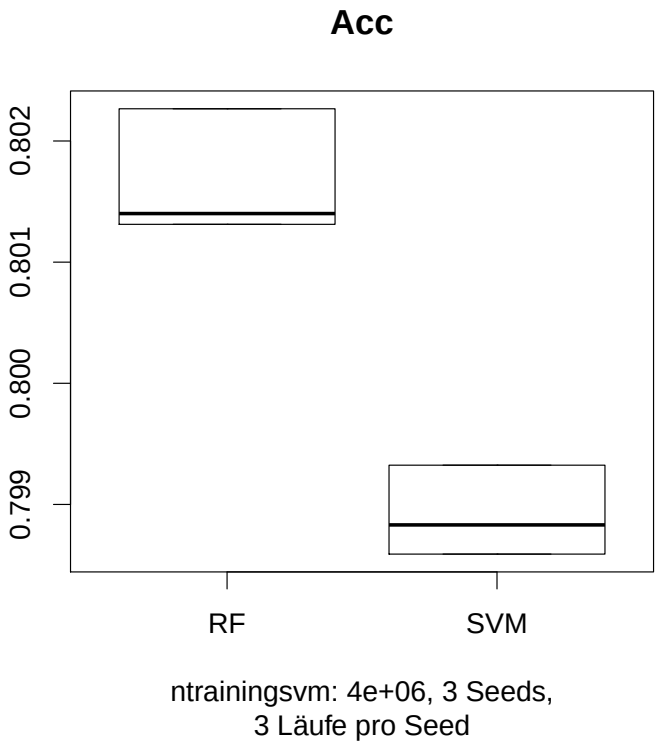


Abbildung 5.17: Genauigkeit (Accuracy – Acc) im Vergleich

Kapitel 6

Zusammenfassung und Ausblick

Statistische Methoden und somit auch Methoden des statistischen maschinellen Lernens sollten danach bewertet werden, ob sie die folgenden Kriterien erfüllen:

- Sie sollten in der Lage sein, ihre Aufgabe (im Falle des maschinellen Lernens: die Prädiktion) sehr gut zu erfüllen;
- sie sollten in kontrollierter Weise auf verrauschte oder anderweitig fehlerbehaftete oder sich verändernde Daten reagieren;
- sie sollten in für die jeweilige Anwendung vertretbarer Zeit und mit vertretbarem Aufwand berechenbar sein;
- und sie sollten ein gewisses Maß an Interpretierbarkeit aufweisen.

Support Vector Machines, spätestens seit 1995 auch unter diesem Namen in der wissenschaftlichen Community bekannt, liefern empirisch und theoretisch fundiert gute Prädiktionen und sind darüber hinaus auch robust gegenüber kleineren Veränderungen in den Trainingsdaten. Neben der mangelnden Interpretierbarkeit, auf die in der vorliegenden Arbeit nicht eingegangen wurde, leiden Support Vector Machines jedoch auch unter dem im Vergleich zu anderen Methoden – wie beispielsweise (generalisierten) linearen Modellen – sehr hohen Aufwand bei der Berechnung, wobei sich dieser Aufwand in Rechenzeit und Speicherplatz ausdrücken lässt. Insbesondere die vergleichsweise hohe Rechenzeit lässt Support Vector Machines gegebenenfalls unattraktiv für die Anwendung erscheinen. Verschiedene Ansätze, dieses Problem zu lösen, sind in der Literatur bekannt; darunter auch der Ansatz des lokalen Lernens durch Bildung von Regionen. Während Letzteres in Implementierungen bereits verfügbar ist, fehlte bislang eine (höchst allgemeine) theoretische Untersuchung dieser Herangehensweise: Insbesondere stellte sich die Frage, ob die wünschenswerten und theoretisch gesicherten Eigenschaften von Support Vector Machines (universelle

Konsistenz und Robustheit unter vollständig überprüfbaren Voraussetzungen) erhalten bleiben, wenn man mittels Regionalisierung und lokalem Lernen die Berechenbarkeit verbessert. Auf Basis dieser Arbeit kann die Frage belegt durch theoretische Resultate und veranschaulicht in Simulationen bejaht werden.

Offen und somit weiteren Forschungen vorbehalten bleiben Verallgemeinerungen auf weitere Verlustfunktionen (beispielsweise solche, die nur lokal Lipschitz-Bedingungen erfüllen) oder schwächere Differenzierbarkeitsbegriffe (bei den Voraussetzungen für die Existenz der Influenzfunktion). Auch wurde die asymptotische Verteilung des Prädiktors, wie in Hable (2012) für die globale Support Vector Machine gezeigt, noch nicht für den zusammengesetzten Prädiktor untersucht. Außerdem bleibt offen, ob eine gemeinsame Betrachtung von Regionalisierungsmethode und dem zusammengesetzten Prädiktor in dem Sinne möglich ist, dass eine denkbare Varianz in der Bildung der Regionen in die theoretischen Untersuchungen des Prädiktors – wie für den dortigen Spezialfall bei Chang, Guo, Lin & Lu (2010) gezeigt – mit einbezogen werden kann.

Anhang A

Zum Einsatz geshifteter Verlustfunktionen

Um den Nutzen der Verwendung geshifteter Verlustfunktionen anstelle der Verlustfunktion selbst darzustellen, wird die Idee der Support Vector Machine als Minimierer eines regularisierten Risikos betrachtet. Wenn das betrachtete Risiko unendlich ist, gibt es keinen Minimierer. Wie in Christmann, Van Messem & Steinwart (2009) gezeigt, gibt es zwei Bedingungen dafür, dass das Risiko endlich ist. Für Lipschitzstetige Verlustfunktionen L mit Lipschitz-Konstante $|L|_1$ und unter der Voraussetzung, dass sich die zugrundeliegende Verteilung P in die Randverteilung $P^{\mathcal{X}}$ und die reguläre bedingte Verteilung $P(y|x)$ zerlegen lässt, kann man zeigen:

$$|\mathcal{R}_{L,P}(f)| = \left| \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x))\, dP(x,y) \right| = \left| \int_{\mathcal{X}\times\mathcal{Y}} L(y, f(x)) - \underbrace{L(y,y)}_{=0 \text{ vgl. Abschnitt 1.4}} dP(x,y) \right|$$

$$\leq |L|_1 \int_{\mathcal{X}} \int_{\mathcal{Y}} |f(x) - y|\, dP(y|x)\, dP^{\mathcal{X}}(x)$$

$$\leq |L|_1 \int_{\mathcal{X}} |f(x)|\, dP^{\mathcal{X}}(x) + |L|_1 \int_{\mathcal{X}} \int_{\mathcal{Y}} |y|\, dP(y|x)\, dP^{\mathcal{X}}(x).$$

Der Ausdruck ist endlich, sofern $f \in L^1(P^{\mathcal{X}})$ und $\int_{\mathcal{X}} \int_{\mathcal{Y}} |y|\, dP(y|x)\, dP^{\mathcal{X}}(x) = \mathbb{E}|Y|$ endlich ist. Letzteres kann problematisch sein, falls $\mathcal{Y}$ unbeschränkt ist (wie beispielsweise in allgemeinen Regressionsproblemen). Verteilungen ohne endliches erstes Moment, wie z. B. die Cauchy-Verteilung, sind daher ausgeschlossen, wenn die Verlustfunktion L selbst betrachtet wird. Wird die geshiftete Version L^* eingesetzt, so erhält man

$$|\mathcal{R}_{L^*,P}(f)| \leq \int_{\mathcal{X}\times\mathcal{Y}} |L(y, f(x)) - L(y,0)|\, dP(x,y) \leq |L|_1 \int_{\mathcal{X}} |f(x)|\, dP^{\mathcal{X}}(x),$$

was endlich ist, wenn lediglich $f \in L^1(P^{\mathcal{X}})$. Es gibt nun also keine Momentenbedingung mehr an Y.[58]

Sogar dann, wenn die betrachtete Verlustfunktion nicht Lipschitz-stetig ist, kann shiften sinnvoll sein. Um das zu belegen, betrachtet man die (nur lokal Lipschitz-stetige) Kleinste-Quadrate-Verlustfunktion $L_{LS}(y,t) := (y-t)^2$. Das zugehörige Risiko ist dann (sofern die Zerlegung von P möglich ist) gegeben durch

$$\begin{aligned}
\mathcal{R}_{\mathcal{X},L_{LS},P}(f) &= \int_{\mathcal{X}} \int_{\mathcal{Y}} \big(y - f(x)\big)^2 \, dP(y|x) \, dP^{\mathcal{X}}(x) = \\
&= \int_{\mathcal{X}} \int_{\mathcal{Y}} y^2 - 2yf(x) + f(x)^2 \, dP(y|x) \, dP^{\mathcal{X}}(x) = \\
&= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} y^2 \, dP(y|x) - 2f(x) \int_{\mathcal{Y}} y \, dP(y|x) \right) + f(x)^2 \, dP^{\mathcal{X}}(x) = \\
&= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} y^2 \, dP(y|x) - 2f(x) \int_{\mathcal{Y}} y \, dP(y|x) \right) dP^{\mathcal{X}}(x) \\
&\quad + \int_{\mathcal{X}} f(x)^2 \, dP^{\mathcal{X}}(x).
\end{aligned}$$

Ein wohldefiniertes und endliches Risiko wird also erzielt, wenn $f \in L^2(P^{\mathcal{X}})$ und zusätzlich

$$\int_{\mathcal{X}} \int_{\mathcal{Y}} y^2 \, dP(y|x) \, dP^{\mathcal{X}}(x) = \mathbb{E}(Y^2)$$

endlich ist. Shiften von L_{LS} führt zu $L^*{}_{LS}(y,t) = (y-t)^2 - y^2$ mit zugehörigem Risiko

$$\begin{aligned}
\mathcal{R}_{L^*{}_{LS},P}(f) &= \int_{\mathcal{X}} \int_{\mathcal{Y}} y^2 - 2yf(x) + f(x)^2 - y^2 \, dP(y|x) \, dP^{\mathcal{X}}(x) = \\
&= \int_{\mathcal{X}} \left(-2f(x) \int_{\mathcal{Y}} y \, dP(y|x) + f(x)^2 \right) dP^{\mathcal{X}}(x),
\end{aligned}$$

welches endlich und wohldefiniert ist, falls $f \in L^2(P^{\mathcal{X}})$ und $\mathbb{E}(Y)$ endlich ist. Somit reduziert die Verwendung der geshifteten Verlustfunktion die Momentenbedingung an Y von einer zweiter Ordnung zu einer erster Ordnung.[59]

[58] Dieser Aspekt wurde im Bezug auf SVMs bereits in Christmann, Van Messem & Steinwart (2009, Proposition 3 (ii)) ausgeführt.

[59] Auch dieser Aspekt wurde im Bezug auf SVMs bereits ausgeführt, unter anderem in Eberts (2015, Kapitel 2).

Quellenverzeichnis

Agarwal, S. & Niyogi, P. (2009). Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, *10*, 441–474.

Alquier, P., Cottet, V., & Lecué, G. (2019). Estimation bounds and sharp oracle inequalities of regularized procedures with lipschitz loss functions. *The Annals of Statistics*, *47*(4), 2117–2144.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *68*, 337–404.

Ash, R. B. & Doleans-Dade, C. (2000). *Probability and measure theory*. San Diego: Academic Press.

Athey, S., Tibshirani, J., & Wager, S. (2019). Generalized random forests. *Ann. Stat.*, *47*(2), 1148–1178.

Aurenhammer, F. (1991). Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Computing Surveys*, *23*(3), 345–405.

Bach, F. (2013). Sharp analysis of low-rank kernel matrix approximations. In Shalev-Shwartz, S. & Steinwart, I. (Eds.), *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, (pp. 185–209).

Baldi, P., Sadowski, P., & Whiteson, D. (2014). Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, *5*, art. no. 4308.

Bayerisches Landesamt für Statistik (2019). Bevölkerungsstand Bayerns am 31. Dezember 2016. *Bayerisches Landesamt für Statistik – Statistische Berichte*, *A1100C 201644*, 1–15.

Beck, M., Dumpert, F., & Feuerhake, J. (2018). *Proof of Concept Machine Learning*. Wiesbaden: Statistisches Bundesamt.

Bennett, K. P. & Blue, J. A. (1998). A support vector machine approach to decision trees. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence*, volume 3, (pp. 2396–2401).

Berlinet, A. & Thomas-Agnan, C. (2001). *Reproducing kernel Hilbert spaces in probability and statistics*. New York: Springer.

Blanzieri, E. & Bryl, A. (2007). Instance-based spam filtering using SVM nearest neighbor classifier. In *FLAIRS Conference*, (pp. 441–442).

Blanzieri, E. & Melgani, F. (2008). Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Transactions on Geoscience and Remote Sensing*, *46*, 1804–1811.

Blaschzyk, I. & Steinwart, I. (2018). Improved classification rates under refined margin conditions. *Electronic Journal of Statistics*, *12*(1), 793–823.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, (pp. 144–152).

Bottou, L. & Vapnik, V. (1992). Local learning algorithms. *Neural computation*, *4*, 888–900.

Bousquet, O. & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research*, *2*(3), 499–526.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical Science*, *16*(3), 199–231.

Cao, Q., Guo, Z.-C., & Ying, Y. (2016). Generalization bounds for metric and similarity learning. *Machine Learning*, *102*, 115–132.

Caponnetto, A. & De Vito, E. (2007). Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, *7*, 331–368.

Caruana, R., Karampatziakis, N., & Yessenalina, A. (2008). An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, (pp. 96–103).

Caruana, R. & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, (pp. 161–168).

Chang, F., Guo, C.-Y., Lin, X.-R., & Lu, C.-J. (2010). Tree decomposition for large-scale SVM problems. *Journal of Machine Learning Research*, *11*, 2935–2972.

Cheng, H., Tan, P.-N., & Jin, R. (2007). Localized support vector machine and its efficient algorithm. In *SDM*, (pp. 461–466).

Cheng, H., Tan, P.-N., & Jin, R. (2010). Efficient algorithm for localized support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, *22*, 537–549.

Christmann, A. (2002). Classification based on the support vector machine and on regression depth. In *Statistical Data Analysis Based on the L1-Norm and Related Methods* (pp. 341–352). New York: Springer.

Christmann, A., Dumpert, F., & Xiang, D.-H. (2016). On extension theorems and their connection to universal consistency in machine learning. *Analysis and Applications*, *14*(6), 795–808.

Christmann, A. & Hable, R. (2012). Consistency of support vector machines using additive kernels for additive models. *Computational Statistics & Data Analysis*, *56*(4), 854–873.

Christmann, A. & Steinwart, I. (2004). On robust properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, *5*, 1007–1034.

Christmann, A., Steinwart, I., & Hubert, M. (2007). Robust learning from bites for data mining. *Computational Statistics & Data Analysis*, *52*(1), 347–361.

Christmann, A. & van Messem, A. (2008). Bouligand derivatives and robustness of support vector machines for regression. *Journal of Machine Learning Research*, *9*(6), 915–936.

Christmann, A., Van Messem, A., & Steinwart, I. (2009). On consistency and robustness properties of support vector machines for heavy-tailed distributions. *Statistics and Its Interface*, *2*, 311–327.

Christmann, A., Xiang, D., & Zhou, D.-X. (2018). Total stability of kernel methods. *Neurocomputing*, *289*, 101–118.

Claeskens, G., Croux, C., & Kerckhoven, J. V. (2008). An information criterion for variable selection in support vector machines. *Journal of Machine Learning Research*, *9*(Mar), 541–558.

Clémençon, S., Lugosi, G., & Vayatis, N. (2008). Ranking and empirical minimization of U-statistics. *The Annals of Statistics*, *36*, 844–874.

Cleveland, W. S. & Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical As-*

sociation, *83*(403), 596–610.

Collobert, R., Bengio, S., & Bengio, Y. (2002). A parallel mixture of SVMs for very large scale problems. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14* (pp. 633–640). MIT Press.

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cristianini, N. & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Cucker, F. & Zhou, D. X. (2007). *Learning theory: an approximation theory viewpoint*. Cambridge: Cambridge University Press.

De Mol, C., De Vito, E., & Rosasco, L. (2009). Elastic-net regularization in learning theory. *Journal of Complexity*, *25*(2), 201–230.

Denkowski, Z., Migórski, S., & Papageorgiou, N. S. (2003). *An Introduction to Nonlinear Analysis: Theory*. New York: Kluwer Academic/Plenum Publishers.

Devroye, L. (1982). Any discrimination rule can have an arbitrarily bad probability of error for finite sample size. *IEEE Trans. Pattern Anal. Mach. Intell.*, *4*, 154–157.

Dua, D. & Graff, C. (2017). UCI machine learning repository.

Duchi, J. C., Jordan, M. I., Wainwright, M. J., & Zhang, Y. (2014). Optimality guarantees for distributed statistical estimation, arxiv:1405.0782v2.

Dudley, R. M. (2004). *Real analysis and probability*. Cambridge: Cambridge University Press.

Dumpert, F. (2019a). Predictability, stability, and computability of locally learnt SVMs. Accepted with minor revisions.

Dumpert, F. (2019b). Quantitative robustness of localized support vector machines. To appear in Communications on Pure and Applied Analysis, preprint: https://arxiv.org/abs/1903.01334.

Dumpert, F. & Christmann, A. (2018). Universal consistency and robustness of localized support vector machines. *Neurocomputing*, *315*, 96–106.

Dunford, N. & Schwartz, J. T. (1958). *Linear operators, part I*. New York: Interscience Publishers.

Eberts, M. (2015). *Adaptive rates for support vector machines*. Aachen: Shaker Verlag; Stuttgart: Univ. Stuttgart (Diss. 2014).

Eberts, M. & Steinwart, I. (2011). Optimal learning rates for least squares SVMs using Gaussian kernels. In *Advances in neural information processing systems*, (pp. 1539–1547).

Eberts, M. & Steinwart, I. (2013). Optimal regression rates for SVMs using gaussian kernels. *Electronic Journal of Statistics*, *7*, 1–42.

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. (2009). *Regression*. Berlin: Springer.

Fan, J., Hu, T., Wu, Q., & Zhou, D.-X. (2016). Consistency analysis of an empirical minimum error entropy algorithm. *Applied and Computational Harmonic Analysis*, *41*, 164–189.

Farooq, M. & Steinwart, I. (2017). An SVM-like approach for expectile regression. *Journal Computational Statistics and Data Analysis*, *109*, 159–181.

Farooq, M. & Steinwart, I. (2019). Learning rates for kernel-based expectile regression. *Machine Learning*, *108*(2), 203–227.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, *15*, 3133–3181.

García-Pedrajas, N. & de Haro-García, A. (2012). Scaling up data mining algorithms: review and taxonomy. *Progress in Artificial Intelligence*, *1*(1), 71–87.

Ghatak, A. (2017). *Machine learning with R*. Singapur: Springer.

Gu, Q. & Han, J. (2013). Clustered support vector machines. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, (pp. 307–315).

Guo, Z.-C., Hu, T., & Shi, L. (2018). Gradient descent for robust kernel-based regression. *Inverse Problems*, *34*(6), 065009.

Guo, Z.-C., Lin, S.-B., & Zhou, D.-X. (2017). Learning theory of distributed spectral algorithms. *Inverse Problems*, *33*(7), 29 pages.

Guo, Z.-C., Ying, Y., & Zhou, D.-X. (2017). Online regularized learning with pairwise loss functions. *Advances in Computational Mathematics*, *43*(1), 127–150.

Guyon, I. & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157–1182.

Hable, R. (2012). Asymptotic normality of support vector machine variants and other regularized kernel methods. *Journal of Multivariate Analysis*, *106*, 92–117.

Hable, R. (2013). Universal consistency of localized versions of regularized kernel methods. *Journal of Machine Learning Research*, *14*, 153–186.

Hable, R. & Christmann, A. (2011). On qualitative robustness of support vector machines. *Journal of Multivariate Analysis*, *102*(6), 993–1007.

Hable, R. & Christmann, A. (2013). Robustness versus consistency in ill-posed classification and regression problems. In *Classification and Data Mining* (pp. 27–35). Berlin: Springer.

Hampel, F. R. (1968). *Contributions to the theory of robust estimation.* Doctoralthesis, University of California, Berkeley.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., & Stahel, W. A. (1986). *Robust statistics. The approach based on influence functions.* John Wiley & Sons, Hoboken, NJ.

Hang, H. (2015). *Statistical learning of kernel-based methods for non-i.i.d. observations.* Doctoralthesis, Universität Stuttgart.

Hang, H. & Steinwart, I. (2014). Fast learning from α-mixing observations. *Journal of Multivariate Analysis*, *127*, 184–199.

Herbrich, R., Graepel, T., & Obermayer, K. (1999). Support vector learning for ordinal regression. In *Artificial Neural Networks. ICANN 99*, volume 1, (pp. 97–102).

Hermes, L. & Buhmann, J. M. (2000). Feature selection for support vector machines. In *Proceedings of the 15th International Conference on Pattern Recognition 2000*, volume 2, (pp. 712–715).

Höffgen, K.-U., Simon, H.-U., & Van Horn, K. S. (1995). Robust trainability of single neurons. *J. Comput. Syst. Sci.*, *50*(1), 114–125.

Hoffmann-Jørgensen, J. (2003). *Probability with a view toward statistics, volume I.* Boca Raton: Chapman & Hall/CRC.

Horn, D., Demircioğlu, A., Bischl, B., Glasmachers, T., & Weihs, C. (2018). A comparative study on large scale kernelized support vector machines. *Advances*

in Data Analysis and Classification, *12*(4), 867–883.

Hu, T., Fan, J., Wu, Q., & Zhou, D.-X. (2013). Learning theory approach to minimum error entropy criterion. *Journal of Machine Learning Research*, *14*, 377–397.

Huber, P. J. (1967). The behavior of maximum likelihood estimates under nonstandard conditions. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, number 1, (pp. 221–233).

Huber, P. J. (1977). *Robust statistical procedures.*, volume 27. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Huber, P. J. & Ronchetti, E. M. (2009). *Robust statistics* (2 ed.). Hoboken, NJ: John Wiley & Sons.

Kadri, H., Duflos, E., Preux, P., Canu, S., & Davy, M. (2010). Nonlinear functional regression: a functional RKHS approach. In Teh, Y. W. & Titterington, M. (Eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, (pp. 374–380)., Chia Laguna Resort, Sardinia, Italy.

Kadri, H., Duflos, E., Preux, P., Canu, S., Rakotomamonjy, A., & Audiffren, J. (2016). Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, *17*, 1–54.

König, C., Schröder, J., & Wiegand, E. (2017). *Big Data: Chancen, Risiken, Entwicklungstendenzen*. Springer.

Kotsiantis, S. B. (2007). Supervised machine learning: a review of classification techniques. *Informatica, Ljubl.*, *31*(3), 249–268.

Lieb, E. H. & Loss, M. (2001). *Analysis* (2 ed.)., volume 14. Providence, RI: American Mathematical Society (AMS).

Lin, J., Rosasco, L., & Zhou, D.-X. (2016). Iterative regularization for learning with convex loss functions. *Journal of Machine Learning Research*, *17*(1), 2718–2755.

Lin, S.-B., Guo, X., & Zhou, D.-X. (2017). Distributed learning with regularized least squares. *Journal of Machine Learning Research*, *18*(92), 1–31.

Meister, M. & Steinwart, I. (2016). Optimal learning rates for localized SVMs. *Journal of Machine Learning Research*, *17*, 1–44.

Micchelli, C. A. & Pontil, M. (2005). On learning vector-valued functions. *Neural computation*, *17*, 177–204.

Mittelstraß, J. (2004). *Enzyklopädie Philosophie und Wissenschaftstheorie. 4 Bände.* Stuttgart und Weimar: Sonderausgabe. Metzler.

Mücke, N. (2017a). *Direct and inverse problems in machine learning.* Doctoralthesis, Universität Potsdam.

Mücke, N. (2017b). Reducing training time by efficient localized kernel regression, arxiv e-prints 1707.03220v3.

Mukherjee, S. & Zhou, D.-X. (2006). Learning coordinate covariances via gradients. *Journal of Machine Learning Research*, *7*, 519–549.

Owhadi, H. & Scovel, C. (2017). Separability of reproducing kernel spaces. *Proceedings of the American Mathematical Society*, *145*(5), 2131–2138.

Paulsen, V. I. & Raghupathi, M. (2016). *An introduction to the theory of reproducing kernel Hilbert spaces.* Cambridge: Cambridge University Press.

Pepe, M. S. (2004). *The statistical evaluation of medical tests for classification and prediction.*, volume 31. Oxford: Oxford University Press.

R Core Team (2018). *R: A Language and Environment for Statistical Computing.*

Wien: R Foundation for Statistical Computing.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (pp. 1135–1144)., New York, NY.

Rida, A., Labbi, A., & Pellegrini, C. (1999). Local experts combination through density decomposition. In *International Workshop on AI and Statistics, Uncertainty '99*. Morgan Kaufmann.

Rosasco, L., De Vito, E., Caponnetto, A., Piana, M., & Verri, A. (2004). Are loss functions all the same? *Neural Computation*, *16*, 1063–1076.

Ruppert, D. & Wand, M. P. (1994). Multivariate locally weighted least squares regression. *The Annals of Statistics*, *22*(3), 1346–1370.

Russell, S. J. & Norvig, P. (2016). *Artificial intelligence: a modern approach* (3 ed.). Upper Saddle River: Pearson.

Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *3*, 210–229.

Schölkopf, B. & Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Cambridge: MIT press.

Segata, N. & Blanzieri, E. (2010). Fast and scalable local kernel machines. *Journal of Machine Learning Research*, *11*, 1883–1926.

Shalev-Shwartz, S. & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge: Cambridge University Press.

Shalev-Shwartz, S., Shamir, O., Srebro, N., & Sridharan, K. (2010). Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, *11*, 2635–2670.

Shawe-Taylor, J. & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge: Cambridge University Press.

Shmueli, G. (2010). To explain or to predict? *Statistical Science*, *25*(3), 289–310.

Si, S., Hsieh, C.-J., & Dhillon, I. S. (2017). Memory efficient kernel approximation. *Journal of Machine Learning Research*, *18*(20), 1–32.

Simon, H. A. (1983). Why should machines learn? In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning – An Artificial Intelligence Approach* (pp. 25–37). Palo Alto, CA: Tioga Pub. Co.

Smale, S. & Yao, Y. (2006). Online learning algorithms. *Foundations of Computational Mathematics*, *6*(2), 145–170.

Steinwart, I. (2007). How to compare different loss functions and their risks. *Constructive Approximation*, *26*, 225–287.

Steinwart, I. & Christmann, A. (2008). *Support vector machines*. New York: Springer.

Steinwart, I. & Christmann, A. (2009). Sparsity of SVMs that use the epsilon-insensitive loss. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *Advances in Neural Information Processing Systems 21* (pp. 1569–1576). Red Hook: Curran Associates.

Steinwart, I. & Christmann, A. (2011). Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, *17*(1), 211–225.

Steinwart, I., Hush, D., & Scovel, C. (2009). Learning from dependent observations. *Journal of Multivariate Analysis*, *100*(1), 175–194.

Steinwart, I. & Thomann, P. (2017). liquidSVM: A fast and versatile SVM package,

arxiv e-prints 1702.06899.

Strohriegl, K. (2018). *On Robustness and Consistency of Support Vector Machines for non-i.i.d. Observations*. Doctoralthesis, Universität Bayreuth.

Strohriegl, K. & Hable, R. (2016). Qualitative robustness of estimators on stochastic processes. *Metrika*, *79*(8), 895–917.

Suthaharan, S. (2014). Big data classification: Problems and challenges in network intrusion prediction with machine learning. *ACM SIGMETRICS Performance Evaluation Review*, *41*(4), 70–73.

Therneau, T. & Atkinson, B. (2018). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-13.

Thomann, P., Blaschzyk, I., Meister, M., & Steinwart, I. (2017). Spatial decompositions for large scale SVMs. In *Proceedings of Machine Learning Research: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics 2017*, volume 54, (pp. 1329–1337).

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*, 1134–1142.

Van Messem, A. & Christmann, A. (2010). A review on consistency and robustness properties of support vector machines for heavy-tailed distributions. *Advances in data analysis and classification*, *4*(2-3), 199–220.

Vapnik, V. & Bottou, L. (1993). Local algorithms for pattern recognition and dependencies estimation. *Neural Computation*, *5*, 893–909.

Vapnik, V. N. (2000). *The nature of statistical learning theory* (2 ed.). New York: Springer.

Vapnik, V. N. & Tscherwonenkis, A. J. (1979). *Theorie der Zeichenerkennung. Übersetzung aus dem Russischen*. Elektronisches Rechnen und Regeln. Berlin: Akademie-Verlag.

Wainberg, M., Alipanahi, B., & Frey, B. J. (2016). Are random forests truly the best classifiers? *Journal of Machine Learning Research*, *17*(1), 3837–3841.

Wang, L., Zhu, J., & Zou, H. (2006). The doubly regularized support vector machine. *Statistica Sinica*, *16*, 589–615.

Wendland, H. (2005). *Scattered data approximation*. Cambridge: Cambridge University Press.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVMs. In *Advances in neural information processing systems*, (pp. 668–674).

Wright, M. & Ziegler, A. (2017). ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, *77*(1), 1–17.

Wu, D., Bennett, K. P., Cristianini, N., & Shawe-Taylor, J. (1999). Large margin trees for induction and transduction. In *ICML*, (pp. 474–483).

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2003). Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, *15*, 505–512.

Xu, H., Caramanis, C., & Mannor, S. (2009). Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, *10*, 1485–1510.

Ying, Y. & Pontil, M. (2008). Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, *8*(5), 561–596.

Ying, Y. & Zhou, D.-X. (2006). Online regularized classification algorithms. *IEEE Transactions on Information Theory*, *52*(11), 4775–4788.

Zakai, A. & Ritov, Y. (2009). Consistency and localizability. *Journal of Machine Learning Research*, *10*, 827–856.

Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, (pp. 2126–2136).

Zhang, X., Wu, Y., Wang, L., & Li, R. (2016). Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *78*(1), 53–76.

Zhu, J., Rosset, S., Hastie, T., & Tibshirani, R. (2004). 1-norm support vector machines. *Advances in neural information processing systems*, *16*(1), 49–56.

Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, *67*(2), 301–320.

Eigene wissenschaftliche Publikationen

Wissenschaftliche Veröffentlichungen zur Theorie des Machine Learning (peer reviewed):

- Dumpert F. (2019) Predictability, stability, and computability of locally learnt SVMs. Accepted with minor revisions bei Archives of Data Science, Series A.
- Dumpert F. (2019) Quantitative robustness of localized support vector machines. To appear in Communications on Pure and Applied Analysis. Preprint: https://arxiv.org/abs/1903.01334.
- Dumpert F., Christmann A. (2018) Universal consistency and robustness of localized support vector machines. Neurocomputing, Vol 315, 2018, S. 96–106.
- Christmann A., Dumpert F., Xiang D.-H. (2016) On extension theorems and their connection to universal consistency in machine learning. Analysis and Applications, Vol 14, No. 6, 2016, S. 795–808.

Wissenschaftliche Veröffentlichungen zu angewandter Statistik (peer reviewed):

- Dumpert F., Beck M. (2017) Einsatz von Machine-Learning-Verfahren in amtlichen Unternehmensstatistiken. AStA Wirtschafts- und Sozialstatistisches Archiv, Band 11, Heft 2, 2017, S. 83–106.

Wissenschaftliche Veröffentlichungen zu angewandter Statistik (Statistisches Bundesamt):

- Beck M., Dumpert F., Feuerhake J. (2018) Machine Learning in Official Statistics. https://arxiv.org/abs/1812.10422
- Finke C., Dumpert F., Beck M. (2017) Verdienstunterschiede zwischen Männern und Frauen. WISTA Wirtschaft und Statistik, Ausgabe 2/2017, S. 43–62.
- Feuerhake J., Dumpert F. (2016) Erkennung nicht relevanter Unternehmen in den Handwerksstatistiken. WISTA Wirtschaft und Statistik, Ausgabe 2/2016, S. 79–94.
- Dumpert F., von Eschwege K., Beck M. (2016) Einsatz von Support Vector Machines bei der Sektorzuordnung von Unternehmen. WISTA Wirtschaft und Statistik, Ausgabe 1/2016, S. 87–97.

Sonstige wissenschaftliche Veröffentlichungen:

- Kreisel T., Dumpert F. (2015) Das Giecher Friedhofsproblem. https://eref.uni-bayreuth.de/id/eprint/9116