The Maximum-Margin Approach to Learning Text Classifiers: Methods, Theory, and Algorithms*

Thorsten Joachims

Universität Dortmund, Prof. K. Morik
tj@cs.cornell.edu

Diese Dissertation entwickelt und erforscht einen neuen Ansatz zum Lernen von Textklassifikationsregeln aus Beispielen. Der Ansatz stützt sich auf die Einsicht, dass bei der Textklassifikation nicht die Anzahl der Attribute die Schwierigkeit einer Lernaufgabe bestimmt, sondern dass dimensionsunahängige Komplexitätsmaße notwendig sind. Die Dissertation zeigt den Zusammenhang dieser Maße mit den statistischen Eigenschaften von Text, deren Umsetzung in effektiven und praktikablen Methoden zur Textklassifikation und ihre Implementierung in effizienten Algorithmen.

1 Einleitung und Motivation

Mit der rasanten Zunahme von Email, World Wide Web (WWW) und Intranets, hat sich die Klassifikation von natürlichsprachlichen Dokumenten in vordefinierte Klassen zu einer zentralen Methode bei der Verwaltung von online Informationen entwickelt. Diese Aufgabe wird als Textklassifikation bezeichnet und erscheint in vielen unterschiedlichen Einsatzgebieten. Zum Beispiel klassifizieren Verzeichnisse wie Yahoo! WWW-Seiten nach Thema, online Zeitungen passen sich den Lesegewohnheiten einzelner Leser an und intelligente Agenten für Service-Hotlines leiten Emails automatisch an den passenden Experten weiter. All diese Aufgaben können im Kern als Textklassifikationsprobleme modelliert werden. Während in der Vergangenheit solche und ähnliche Aufgaben manuell erledigt wurden, verlangen das stark gestiegene Volumen und möglichst schnelle Bearbeitungszeiten nach automatischen Methoden.

Ein erster Schritt zur Automatisierung von Textklassifikation ist die manuelle Generierung von automatischen Klassifikationsregeln. Hierbei programmieren Experten Regeln, die dann neue Dokumente klassifizieren können. Dieser Ansatz hat sich jedoch als schwierig und zeitintensiv herausgestellt [HW90]. In vielen Situationen ist eine manuelle Modellierung zu ineffizient und unpraktikabel, insbesondere, wenn die Anzahl der Klassen und somit die Anzahl der Regeln groß ist. Noch problematischer ist der manuelle Ansatz, wenn Textklassifikationsregeln als benutzeradaptiver Teil z. B. einer Desktopanwendung benötigt werden. In solchen Fällen ist ein Modellierungsexperte schlicht nicht verfügbar.

*Erschienen als: *Learning to Classify Text Using Support Vector Machines*, Kluwer, 2002.

Und schießlich erfordert eine manuelle Regelgenerierung auch bei der Wartung der Regeln einen hohen Aufwand, speziell wenn sich die Klassendefinitionen (wie z. B. bei dem Herausfiltern von Spam-Email) mit der Zeit verändern.

Mit Hilfe von Methoden des maschinellen Lernens können diese Probleme bei der Generierung und Wartung von Textklassifikationsregeln vermieden werden. Gegeben eine relativ kleine Menge von manuell klassifizierten Trainingsdokumenten, so kann die Aufgabe des Lernens von Textklassifikationsregeln als überwachte Lernaufgabe formuliert werden. Der Lernalgorithmus analysiert die klassifizierten Trainingsdokumente und gibt als Ergebnis eine Klassifikationsregel aus, die neue Dokumente mit größtmöglicher Genauigkeit klassifziert. Obwohl dieser Ansatz bereits von anderen verfolgt worden war, waren zu Beginn der Dissertation die entscheidenden Fragen dieses Lernproblems noch unbeantwortet. Das Ergebnis dieser Dissertation ist ein neuer Ansatz zum Lernen von Textklassifikationsregeln, welcher nicht nur die Vorhersagegenauigkeit von konventionellen Verfahren substantiell übertrifft, sondern für den hier erstmalig auch lerntheoretische Ergebnisse gezeigt werden, aus denen sowohl praktikable und effektive Methoden, als auch effiziente Algorithmen hergeleitet werden.

2 Textklassifikation

Die Aufgabe des Lernens bei der Textklassifikation besteht darin, aus einer Stichprobe von klassifizierten Trainingsbeispielen automatisch eine allgemeine Regel zu generieren, die neue Dokumente mit hoher Genauigkeit klassifiziert. Formeller betrachtet erhält ein Lernalgorithmus $\mathcal{L}$ eine Stichprobe S von n klassifizierten Dokumenten

$$(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n).$$

Die Beispiele sind unabhängig und identisch verteilt nach einer unbekannten aber festen Verteilung $\Pr(\vec{x}, y)$. Jedes Beispiel besteht aus einem Dokumentvektor $\vec{x}$ und der Klassifikation y.

Der Dokumentvektor $\vec{x}$ beschreibt den Inhalt des Dokuments, meist durch einen Vektor von Attributwerten. Die bei weitem am häufigsten eingesetzte Art der Repräsentation ist die "bag-of-words" Darstellung. Jedes Wort der Sprache entspricht einem Attribut. Der Wert eines Attributs für ein bestimmtes Dokument basiert auf der Häufigkeit, mit der das Wort im Dokument vorkommt. Abbildung 1 illustriert diese Repräsentation, wobei typischerweise noch Transformationen aus dem Information Retrieval auf diese Basisrepräsentation angewendet werden [SB88]. Zudem gibt es eine Reihe von Modellierungsoptionen (wie z. B. Stammformenreduktion, Stopwort-Elimination, Nominalphrasenerkennung, statistische Attributselektion), die jeweils speziell auf eine Textklassifikationsaufgabe angepaßt werden müssen[1].

Den Typ der Klassifikationsvariablen y bestimmt die Art der Klassifikationsaufgabe. Im

[1] Auch andere Repräsentationsformen, insbesondere solche die linguistisches und taxonomisches Wissen einbeziehen, wurden bereits ausgiebig untersucht. Allerdings haben sie sich als weniger robust, weniger effizient und oftmals nur marginal effektiver erwiesen, so dass sie in der Praxis kaum zum Einsatz kommen.

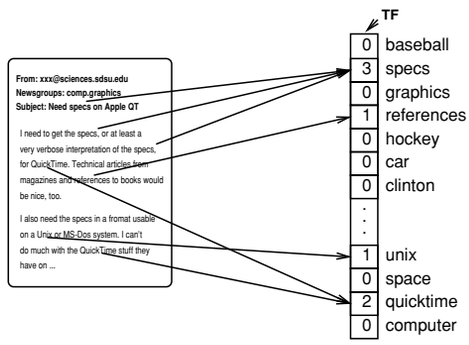

Abbildung 1: Die Repräsentation eines Dokuments als Histogramm von Worthäufigkeiten.

einfachsten Fall gibt es zwei Klassen — die im folgenden durch die Werte $y \in \{-1, 1\}$ repräsentiert werden — und der Lerner versucht eine Klassifikationsregel zu finden, die den Vorhersagefehler minimiert. Der Vorhersagefehler ist die erwartete (durchschnittliche) Häufigkeit von Fehlklassifikationen auf neuen, ebenfalls nach $\Pr(\vec{x}, y)$ gezogenen, Beispielen. Da der Vorhersagefehler somit von der unbekannten Verteilung $\Pr(\vec{x}, y)$ abhängt, kann er nicht direkt berechnet werden. Die einzige explizite Information über $\Pr(\vec{x}, y)$ ist die Trainingsstichprobe S. Basierend auf S ist es das Ziel des Lernalgorithmus $\mathcal{L}$ eine Klassifikationsregel $h_{\mathcal{L}} = \mathcal{L}(S)$ zu finden, welche den Vorhersagefehler minimiert. Hierbei ergeben sich sowohl statistische (das Schliessen von der Stichprobe auf die Verteilung $\Pr(\vec{x}, y)$), als auch algorithmische Fragestellungen (die Suche nach der Klassifikationsregel in Abhängikeit von S). Sie führen zu den folgenden Herausforderungen.

3 Herausforderungen und Ziele

Die Aufgabe des Lernens von Textklassifikationsregeln stellt eine neue Kombination von Herausforderungen für das maschinelle Lernen. Sie wird durch die folgenden Eigenschaften charakterisiert:

GROSSER EINGABERAUM: Bei der Textklassifikation ist die Eingabe zum Lernalgorithmus natürliche Sprache. Natürliche Sprache ist ausdrucksstark genug, um damit viele Phänomene in der Welt zu beschreiben — oft sogar durch mehrere, äquivalente Formulierungen. “The sentences of a language may be unlimited in number” ([Lyo68], Abschnitt 4.2.2), aber zumindest kann man aus praktischer Sicht in den meisten Fällen nicht davon ausgehen, dass man den gleichen Satz (oder sogar das gleiche Dokument) mehrmals in einem Korpus findet. Zum Beispiel ist es unwahrscheinlich, das dieser Satz jemals zuvor formuliert wurde. Deshalb trifft man bei der Textklassifikation notwendigerweise auf eine große Menge von potentiellen Beispielen. Selbst stark vereinfachende Repräsentationen, wie die oben beschriebenen bag-of-words Histogramme, führen immer noch zu Attributräumen der Dimension 30.000 und mehr.

KLEINE TRAININGSSTICHPROBEN: Für die meisten Lernalgorithmen skaliert die Anzahl der für eine hinreichend genaue Klassifikationsregel benötigten Beispiele mit der Dimension des Attributraumes. Um eine hinreichende Vorhersagegenauigkeit für diese Algorithmen zusichern zu können, müsste man Trainingsstichproben ziehen, deren Größe nicht mehr praktikabel ist. Für die Polynomklassifikation formuliert Fuhr die Faustregel, dass man mindestens 50-100 Trainingsbeispiele für jedes Attribut (oder, genauer, jede Attributkombination) haben sollte [FPB+94, Seite 188]. Im krassen Gegensatz hierzu ist man bei der Textklassifikation meist in der paradoxen Situation, dass man weniger Trainingsbeispiele als Attribute hat.

RAUSCHEN: Die meisten Dokumente enthalten Fehler. Entgegen meiner besten Anstrengungen, werden Sie wahrscheinlich Rechtschreibfehler, Tippfehler und ungrammatikalische Sätze in diesem Artikel finden. In der Terminologie des maschinellen Lernens kann man dies als (Attribut-)Rauschen bezeichnen. Zudem produziert der Prozess der manuellen Klassifikation oft fehlklassifizierte Beispiele oder die Lernaufgabe ist prinzipiell probabilistisch, was zu (Klassifikations-)Rauschen führt.

KOMPLEXE LERNAUFGABEN: Die Klassen bei der Textklassifikation basieren meist auf dem semantischen Verständnis von natürlicher Sprache durch Menschen. Zum Beispiel können die Klassen das Thema eines Dokuments beschreiben, die Leseinteressen eines Zeitungslesers oder die Dringlichkeit einer Email. Für keine dieser Aufgaben existiert eine formale und operationale Definition. Der Lernalgorithmus muß solch komplexe Konzepte approximieren können.

EFFIZIENTE BERECHENBARKEIT: Das Lernen in hochdimensionalen Eingaberäumen mit mehreren tausend Attributen und mehreren tausend Beispielen ist eine algorithmisch schwierige Aufgabe. Um brauchbare Lernmethoden für praktische Anwendungen zu erhalten, ist es notwendig, Algorithmen zu entwickeln, die speziell in hochdimensionalen Räumen effizient arbeiten.

Die Bewältigung dieser Herausforderungen motiviert die Methoden, die Theorie, und die Algorithmen, die in dieser Dissertation entwickelt werden. Das Ziel dieser Dissertation ist ein neuer Ansatz zum Lernen von Textklassifikationsregelen aus Beispielen. Es stehen weder die Methoden, noch die Theorie, noch die Algorithmen, im Vordergrund des Interesses. Vielmehr sollen für eine befriedigende Lösung des Textklassifikationsproblems alle drei Aspekte behandelt werden. Zur Zeit gibt es keinen anderen Ansatz der sowohl algorithmisch effizient, als auch im Hinblick auf die Textklassifikation theoretisch fundiert und methodisch robust in der Anwendung ist. Der im folgenden entwickelte Ansatz leidet unter keiner dieser Beschränkungen von konventionellen Lernverfahren.

Neben ihrem Beitrag zum Anwendungsproblem des Lernens von Textklassifikatoren, sollen die in dieser Dissertation entwickelten Techniken des maschinellen Lernens aber möglichst allgemein gehalten werden, so dass sie sich auch auf andere Anwendungen übertragen lassen. Im folgenden werde ich mich jedoch auf ihre Diskussion im Rahmen der Textklassifikation beschränken.

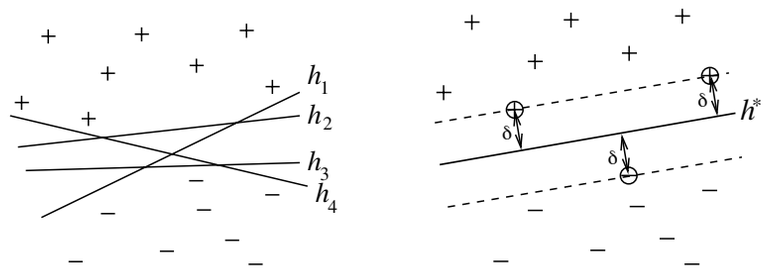

Abbildung 2: Eine binäre Klassifikationsaufgabe mit zwei Attributen. Positive Beispiele sind durch + markiert, negative Beispiele durch −. Links: Mehrere Trennende Hyperebenen mit kleiner Separationsweite. Rechts: Die Hyperebene mit maximaler Separationsweite.

4 Ergebnisse

Der Ansatz, der in dieser Dissertation verfolgt wird, fundiert auf der Einsicht, dass Separationsweite (oder "margin", wie z. B. in Vapnik's Support Vector Machines (SVMs)) ein besseres Komplexitätsmaß für die Textklassifikation ist, als die Dimension des Attributraumes. Die Separationsweite ist wie folgt definiert [Vap98]. Der einfachste Fall liegt vor, wenn es sich um eine binäre Klassifikationsaufgabe mit $y \in \{-1, 1\}$ handelt, wenn die Klassifikationsregel eine lineare Funktion (d.h. eine Hyperebene im Attributraum)

$$h(\vec{x}) = sign\{\vec{w} \cdot \vec{x} + b\} = \begin{cases} +1, & \text{if } \vec{w} \cdot \vec{x} + b > 0 \\ -1, & \text{else} \end{cases} \quad (1)$$

ist und wenn die Trainingsstichprobe $S = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$ linear trennbar ist. In diesem Fall ist die Separationsweite einer trennenden Hyperebene der minimale Abstand zu den nächstgelegenen Trainingsbeispielen. Der linke Teil von Abbildung 2 zeigt einige trennende Hyperebenen mit kleiner Separationsweite, wohingegen die Hyperebene im rechten Teil der Abbildung die maximale Separationsweite erzielt. Dies ist die Hyperebene, die von der SVM gewählt wird. Wie im weiteren erläutert, erlaubt die Größe der Separationsweite δ (und neue, verbesserte Maße) theoretisch fundierte Abschätzungen und Schranken für die Vorhersagegenauigkeit auf neuen Dokumenten. Insbesondere sei darauf hingewiesen, dass die Separationsweite δ nicht notwendigerweise von der Dimension des Attributraumes abhängt.

Basierend auf diesen Grundlagen, fassen die folgenden drei Abschnitte die Ergbnisse der Dissertation zusammen. Wie die eigentliche Ausarbeitung, gliedern sie sich in die Bereiche THEORIE, METHODEN und ALGORITHMEN.

4.1 Theorie

Der Kern des theoretischen Teils ist ein statistisches Lernmodell der Textklassifikation. Für keine der konventionellen Textklassifikationsmethoden existiert eine brauchbare Theo-

rie die erklärt, wann und warum sie für bestimmte Textklassifikationsaufgaben gut oder schlecht funktionieren. Theoretische Modelle existieren für einige Methoden, wie z. B. den naive Bayes Klassifikator. Sie sind aber zu restriktiv und nachweisbar falsch für Text [Mar61, Seite 410]. Andere Methoden, wie z. B. Entscheidungsbaumlerner, rechtfertigen sich ausschließlich durch empirische Ergebnisse. Ihre Eignung für die Textklassifikation ist nicht hinreichend verstanden.

Diese Dissertation führt erstmalig ein statistisches Lernmodell der Textklassifikation ein, das die statistischen Eigenschaften von Text mit der Vorhersagegenauigkeit des Lerners (hier der SVM) in einem theoretischen Modell verbindet. Die folgenden 5 Eigenschaften von Text in der bag-of-words Darstellung gehen in das Modell ein: (1) Die Dokumentvektoren liegen in einem hochdimensionalen Raum. (2) Die Dokumentvektoren sind spärlich besetzt, d. h. nur relativ wenige Worte der Sprache kommen in einem einzelnen Dokument vor. (3) Sprache ist redundant, so dass z. B. viele Worte in einem Dokument Hinweis auf das Thema des Dokuments sind. (4) Sprache ist heterogen, so dass man mit unterschiedlichen Worten das gleiche aussagen kann. (5) Die Worthäufigkeiten folgen dem Zipf'schen Gesetz [Zip49]. Für diese Kombination von Eigenschaften wird eine Konzeptklasse definiert, deren Lernbarkeit untersucht werden kann. Es kann gezeigt werden, dass diese Konzepte relativ schnell von einer SVM gelernt werden können, und dass man bei dieser Aufgabe durch Betrachtung der Separationsweite den "curse of dimensionality" auch ohne heuristische Attributauswahlmethoden umgehen kann. Dies ist das erste Modell der Textklassifikation, bei dem man je nach Ausprägung der 5 Eigenschaften entscheiden kann, ob eine Lernmethode gut funktionieren wird. Es identifiziert hinreichende Bedingungen dafür, wann eine SVM eine beweisbar hohe Vorhersagegenauigkeit bei einer Textklassifikationsaufgabe erreicht.

Das theoretische Modell basiert auf einer intensionalen Beschreibung der Textklassifikationsaufgabe. Wenn Trainingsdaten verfügbar werden, können sie das Modell ergänzen oder ersetzen. Mit den Trainingsdaten wird die Frage nach der Eignung eines Lernalgorithmus für eine Aufgabe zu einem statistischen Schätzproblem. Dieses Problem ist der Gegenstand des zweiten theoretischen Beitrags. Anknüpfend an [Vap98][JH99] sind das Ergebnis effiziente Schätzer der Vorhersagegenauigkeit einer SVM im Sinne der Fehlerrate, aber auch der Precision, des Recall und des $F1$-Maßes. Es wird gezeigt, dass diese sog. $\xi\alpha$-Schätzer eine konservative Approximation des Kreuzvalidierungsfehlers sind, wobei sie um mehrere Größenordungen effizienter zu berechnen sind. Desweiteren kann mit den gleichen Methoden auch der exakte Kreuzvalidierungsfehler viel effizienter bestimmt werden. Dieses theoretische Ergebnis hat direkte Auswirkungen auf die Methodik und Praxis der Textklassifikation.

4.2 Methoden

Als ersten Beitrag aus methodischer Sicht wird ein zweistufiger Prozess zur Modellselektion vorgeschlagen und untersucht. Mit Modellselektion bezeichnet man die Wahl aller Parameter, die das Verhalten des Lerners bestimmen. Es wird gezeigt, wie man mit den $\xi\alpha$-Schätzern aus dem Theorieteil für eine gegebene Aufgabe automatisch eine geeignete

	Bayes	Rocchio	C4.5	k-NN	lineare SVM	
					$\xi\alpha$	$\xi\alpha$-KV
earn	96.0	96.1	96.1	97.8	98.0	98.2
acq	90.7	92.1	85.3	91.8	95.5	95.6
money-fx	59.6	67.6	69.4	75.4	78.8	78.5
grain	69.8	79.5	89.1	82.6	91.9	93.1
crude	81.2	81.5	75.5	85.8	89.4	89.4
trade	52.2	77.4	59.2	77.9	79.2	79.2
interest	57.6	72.5	49.1	76.7	75.6	74.8
ship	80.9	83.1	80.9	79.8	87.4	86.5
wheat	63.4	79.4	85.5	72.9	86.6	86.8
corn	45.2	62.2	87.7	71.4	87.5	87.8
microavg. (all 90)	72.3	79.9	79.4	82.6	86.7	87.5

Tabelle 1: Precision/Recall Breakeven Point der konventionellen Methoden im Vergleich zur SVM mit automatische Modellselektion für den Reuters-21578 Korpus.

Kombination von Vorverarbeitungsschritten (wie Stammformreduktion, Stoppworteliminination, Wortgewichtung) und Lernparametern auswählen kann. Das Lernergebnis nach dieser automatischen Auswahl wird mit dem Ergebnis konventioneller Lernverfahren verglichen. Ein Teil dieser Ergebnisse ist in Tabelle 1 zusammengefasst. Sie zeigt den Precision/Recall Breakeven Point (PRBEP) auf den Testdaten für den Reuters-21578 Benchmark Korpus. Je höher der Wert, desto besser die Vorhersage. Jede Zeile steht für eine binäre Klassifikationsaufgabe (die 10 größten Klassen), wobei die letzte Zeile ein Mittelwert über alle 90 Aufgaben ist. Die Spalte "$\xi\alpha$" zeigt die Performanz der SVM mit reiner $\xi\alpha$-Modellselektion. Benutzt man die $\xi\alpha$-Methode um den exakten Kreuzvalidierungsfehler zu bestimmen (letzte Spalte), erhält man leicht verbesserte Ergebnisse. Um sicherzustellen, dass die konventionellen Lernverfahren nicht durch eine schlechte Parameterwahl im Vergleich benachteiligt werden, habe ich in der Tabelle zu ihren Gunsten "gemogelt". Für die konventionellen Lernverfahren wurden die Repräsentation (inkl. Attributselektion) und die Parameter gewählt, welche auf den Testdaten die besten Ergebnisse liefern. Trotzdem zeigen beide SVM Methoden substantiell bessere PRBEP als alle konventionellen Verfahren. Dies zeigt sich auch für eine weite Spanne von anderen Textklassifkationsaufgaben[2].

Der zweite methodische Beitrag ist die Einführung der Idee der Transduktion [Vap98] für die Textklassifikation. Mit der transduktiven Aufgabenstellung hat man die Möglichkeit, viele Aufgaben der Textklassifikation und des Information Retrieval genauer zu modellieren. Im Gegensatz zum (normalen) induktiven Lernen, sind bei der Transduktion die Dokumentvektoren der Testdaten Teil der Eingabe zum Lerner. Ein typisches Beispiel für eine inhärent transduktive Lernaufgabe ist Relevance Feedback. Beim Relevance Feedback kann ein Benutzer dem Information Retrieval System positive und negative Beispiele für relevante Dokumente geben. Da das IR System auf einer festen Dokumentkollektion arbeitet, kann es die zu klassifizierende Dokumente (d. h. alle Dokumente, die der Benutzer noch nicht gesehen hat) beim Lernschritt auswerten. Die Dissertation zeigt, wie diese zusätzliche Information ausgenutzt werden kann und analysiert, warum das Wissen

[2]Bei dem TREC-2001 Textklassifikations-Wettbewerb hat David Lewis mit *SVM*^{light} und exakter $\xi\alpha$-Kreuzvalidierung teilgenommen und mit großem Abstand gewonnen [Lew01].

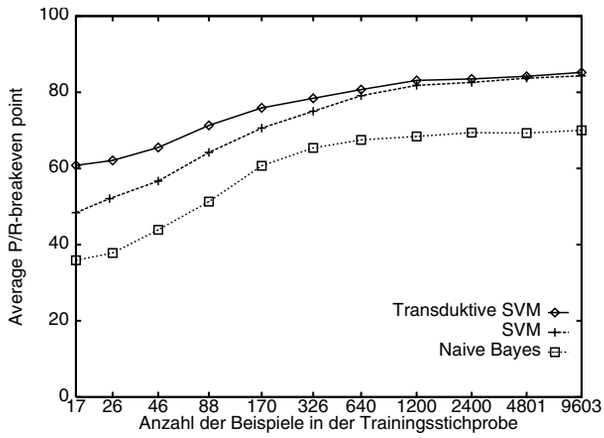

Abbildung 3: Macroaveraged PRBEP auf dem Reuters-21578 Korpus für verschiedene Anzahlen von Trainingsbeispielen und eine Testmenge der Größe 3299.

über die Lage der Testpunkte bei der Textklassifikation zu höherer Vorhersagegenauigkeit führen kann. Wiederum liegt der Schlüssel für die Beantwortung dieser Frage in dem Zusammenspiel zwischen den statistischen Eigenschaften von Text und der Separationsweite, insbesondere in der Verteilung von Wortpaarkorrelationen. Einen Teil der empirischen Ergebnisse zur Transduktion zeigt Abbildung 3. Insbesondere für kleine Trainingsmengen kann ein transduktiver SVM Ansatz die Vorhersagegenauigkeit verbessern. Es wurde somit theoretisch und empirisch gezeigt, dass die Betrachtung von Textklassifikation als transduktives Inferenzproblem die gegebenen Informationen besser ausnutzen kann. Dies ist aber nur der erste Schritt und weitere, verbesserte Ausnutzungen der Transduktionsidee erscheinen vielversprechend.

4.3 Algorithmen

Der dritte Teil der Dissertation entwickelt neue Algorithmen zum Training von induktiven und transduktiven SVMs. Im Gegensatz zu den meisten konventionellen Algorithmen beruht ihr Laufzeitverhalten nicht notwendigerweise auf der Dimension des Merkmalraumes. Die SVM Algorithmen nutzen aus, dass die Attributvektoren bei der Textklassifikation nur spärlich besetzt sind. In diesem Fall können Attributräume mit mehreren 100.000 Attributen effizient bearbeitet werden.

Aber nicht nur die Anzahl der Attribute ist typischerweise hoch, sondern oft auch die Anzahl der Trainingsbeispiele. Obwohl das Training einer induktiven SVM auf ein quadratisches Programm reduziert werden kann, sind Standardmethoden zur quadratischen Optimierung zu ineffizient oder lassen sich bei nur bei kleinen Datenmengen ($<$3000 Beispiele) anwenden. Das erste Ergebnis der Dissertation im Bereich der Algorithmen ist ein effizienter Algorithmus zum Training von induktiven SVMs. Er wurde bereits für Trainingsstichproben mit mehreren Millionen von Beispielen eingesetzt. Der Algorithmus ist

Examples	SVM^{light}	SMO	Chunking
2477	3.2	2.2	13.1
4912	8.5	8.1	40.6
9888	24.7	24.7	239.3
24692	60.0	104.9	3369.7
49749	126.7	268.3	17164.7
Skalierung	1.2	1.6	2.5

Tabelle 2: Trainingzeiten (in CPU-Sekunden) auf dem "Web" Datensatz von John Platt für eine lineare SVM.

in der Software SVM^{light}implementiert und auf dem WWW frei verfügbar[3]. Tabelle 2 vergleicht seine Laufzeit mit dem konventionellen Chunking Algorithmus und dem SMO Algorithmus [Pla99], der parallel von John Platt bei Microsoft entwickelt wurde. Abgesehen von einigen heuristischen Komponenten im SMO Algorithmus und der sog. Shrinking Methode in SVM^{light}, ergibt sich der SMO Algorithmus als Spezialfall von SVM^{light}. Auf allen untersuchten Problemen ist SVM^{light}substantiell schneller ist als der konventionelle Chunking Algorithmus und bei vielen schlägt SVM^{light}auch SMO in der empirischen Skalierbarkeit — hier auf einem Benchmark Problem von John Platt. Seit seiner Veröffentlichung in 1997 wurde SVM^{light}bereits bei üeber hundert Studien in einer Vielzahl von wissenschaftlichen Fachgebieten eingesetzt.

Das zweite Ergebnis ist ein Algorithmus für das Training von transduktiven SVMs, mit dem erstmalig auch große Testdatenmengen bearbeitet werden können. Bei der transduktiven SVM ergibt sich ein gemischtes Integer-Optimierungsproblem, für welches keine effiziente Lösung bekannt ist. Der neue Algorithmus findet eine approximative Lösung mittels einer speziellen Form von lokaler Suche. Allerdings ist, ähnlich zu den statistischen Fragen bei der Transduktion, auch dieser Algorithmus nur ein erster Schritt. Weitere Forschung zur Robustheit und Effizienz von Transduktionsalgorithmen sind notwendig und vielversprechend.

5 Zusammenfassung

Diese Dissertation verfolgt einen neuen Ansatz zur Textklassifikation, der nicht die Anzahl der Attribute, sondern die Separationsweite als zentrales Komplexitätsmaß verwendet, wodurch sich der "curse of dimensionality" bei der Textklassifikation beweisbar umgehen läßt. Dieser Ansatz verbindet erstmalig eine theoretische Fundierung mit darauf aufbauenden robusten Methoden und effizienten Algorithmen. Über das spezielle Anwendungsgebiet der Textklassifikation hinausgehend, enthält die Dissertation allgemeine Ergebnisse im Bereich des maschinellen Lernen, insbesondere Fehlerschranken für SVMs, $\xi\alpha$-Schätzer, die Analyse der Transduktion und die Trainingsalgorithmen für SVMs. Alle Techniken und Methoden sind in SVM^{light}implementiert und öffentlich verfügbar.

[3] `http://svmlight.joachims.org/`

Literaturverzeichnis

[FPB+94] N. Fuhr, U. Pfeifer, C. Bremkamp, M. Pollmann, and C. Buckley. Probabilistic Learning Approaches for Indexing and Retrieval with the TREC-2 Collection. In *The Second Text Retrieval Conference (TREC-2)*. National Institute of Standards and Technology, 1994.

[HW90] P. Hayes and S. Weinstein. CONSTRUE/TIS: a System for Content-Based Indexing of a Database of News Stories. In *Annual Conference on Innovative Applications of AI*, 1990.

[JH99] T. Jaakkola and D. Haussler. Probabilistic Kernel Regression Models. In *Conference on AI and Statistics*, 1999.

[Lew01] D. Lewis. Applying Support Vector Machines to the TREC-2001 Batch Filtering and Routing Tasks. In *Text Retrieval Conference (TREC)*, 2001.

[Lyo68] J. Lyons. *Introdutions to Theoretical Linguistics*. Cambridge University Press, London, 1968.

[Mar61] M. E. Maron. Automatic Indexing: An Experimental Inquiry. *Journal of the Association for Computing Machinery*, 8:404–417, 1961.

[Pla99] J. Platt. Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 12. MIT-Press, 1999.

[SB88] G. Salton and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley, Chichester, GB, 1998.

[Zip49] George Kingsley Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, Cambridge, MA, USA, 1949.

Thorsten Joachims ist ein Assistant Professor im Department of Computer Science an der Cornell University. Er begann seine Arbeit dort im Jahre 2001, nach einem kurzen Postdoc am Institut für autonome, intelligente System der GMD in Bonn. Ebenfalls im Jahre 2001 schloss Thorsten Joachims seine Dissertation bei Prof. Katharina Morik am Lehrstuhl für KI der Universität Dortmund ab, wo er seit 1997 als wissenschaftlicher Mitarbeiter tätig war. Sein Diplom der Informatik erhielt er im Jahre 1997, ebenfalls von der Universität Dortmund, mit einer Diplomarbeit zu "WebWatcher", einem Browsing-Assistenten für das WWW. Seine Forschungsinteressen liegen im Bereich des maschinellen Lernens und dem Information Retrieval. Speziell arbeitet er im Bereich der statistischen Lerntheorie, der Support Vector Maschinen und dem maschinellen Lernen für die Informationssuche. Bis 1996 verbrachte er einen eineinhalbjährigen Forschungsaufenthalt bei Prof. Tom Mitchell an der Carnegie Mellon University. 1991 war er ein Bundessieger beim Bundeswettbewerb Informatik der GI.