

# Analyse und Prognose Ökonomischer Zeitreihen mit Support Vector Machines

Manuel Möller

Deutsches Forschungszentrum für Künstliche Intelligenz  
manuel.moeller@dfki.de

Nico Schlitter

Otto-von-Guericke-Universität Magdeburg  
Nico.Schlitter@iti.cs.uni-magdeburg.de

Die Faszination des Aktienhandels geht sicherlich zu einem nicht geringen Anteil davon aus, dass ein Aktionär mit einfachen Transaktionen innerhalb kurzer Zeit große Gewinne erzielen kann, wenn er die zukünftige Entwicklung eines Unternehmens richtig vorherzusagen vermag. Die Idee, dass sich ausschließlich durch eine Analyse bereits bekannter Aktienkurswerte eine Prognose für die Zukunft erstellen lässt ist nicht neu. Den Wertverlauf einer Aktie als Funktion in Abhängigkeit von der Zeit anzunähern, wurde mit unterschiedlichsten Verfahren in der Vergangenheit versucht. Ziel dieser Arbeit ist es zu zeigen, dass die aus der Statistical Learning Theory von Vapnik und Chervonenkis hervorgegangenen Support Vector Machines in ihrer Prognosegüte den häufig eingesetzten künstlichen neuronalen Netzen (KNN) ebenbürtig und dabei wesentlich rechenzeiteffizienter sind.

## 1 Einleitung

Grundsätzlich lässt sich die Prognose anhand von wirtschaftlichen Kenndaten eines Unternehmens von einer Prognose auf Basis der letzten bekannten Kurswerte unterscheiden. Die erste Art wird *Fundamentalanalyse* genannt, die zweite *technische Analyse*. In dieser Arbeit wird die technische Analyse unter Anwendung eines maschinellen Lernverfahrens aus dem Bereich der Künstlichen Intelligenz untersucht, den sog. *Support Vector Machines* (SVMs).

Die Idee, dass sich ausschließlich durch eine Analyse bereits bekannter Aktienkurswerte eine Prognose für die Zukunft erstellen lässt, ist nicht neu. Den Wertverlauf einer Aktie als Funktion in Abhängigkeit von der Zeit anzunähern, wurde in der Vergangenheit u.a. mit künstlichen neuronalen Netze (KNN) versucht.

Dabei konnte gezeigt werden, dass sich auf diese Weise grundsätzlich Renditen erzielen lassen [Schlitter 2006], [Yao&Tan 2001], [Mohan et al. 2005]. Durch die speziellen Eigenschaften der KNN als maschinelles Lernverfahren ist der Rechenaufwand dafür allerdings

verhältnismäßig hoch. Ziel dieser Arbeit ist zu zeigen, dass SVMs ebenso gut geeignet sind wie KNN und bei vergleichbarer Prognosequalität mit wesentlich geringerer Rechenzeit auskommen.

In Abschnitt 2 werden dazu zunächst einige ökonomische Grundlagen dargestellt, die für die Anwendung der SVMs in einer technischen Analyse zwingend erforderlich sind. In Abschnitt 3 wird die Datenvorverarbeitung beschrieben, die notwendig ist, um die zur Verfügung stehenden Kursdaten als Trainingsdaten in den SVMs verwenden zu können. Abschnitt 4 stellt überblickartig die verwendeten Finanzindikatoren vor, die als Referenzmaßstab für die Bewertung der Prognosegüte der SVMs und zur Datenaufbereitung verwendet wurden. Um die SVMs in ihrer Prognosegüte mit anderen Verfahren vergleichen zu können, wurde für diese Arbeit eine objektorientierte Simulationsumgebung implementiert, in der als Agenten implementierte Handelsstrategien im Zeitraum von 1998 bis 2004 auf Basis historischer Kursdaten um die beste Prognosegüte konkurrieren können. Abschnitt 5 beschreibt dieses System. In Abschnitt 6 werden die gesammelten Daten analysiert und den Ergebnissen aus [Schlitter 2006] gegenübergestellt. Anschließend werden die gewonnenen Erkenntnisse zusammengefasst und daraus resultierende, neue Forschungsfragen kurz skizziert.

## 2 Ökonomische Grundlagen

Prinzipiell gehört der für diese Arbeit gewählte Ansatz in den Bereich der *Finanzanalyse*. Nach der Definition in [Rehkugler&Poddig 1994, 3] geht es bei einer Finanzanalyse im allgemeinen darum, »einer Menge von Zuständen oder Situationen (z.B. Situationen an Finanzmärkten, definiert durch die Zustände bestimmter ökonomischer Variablen wie Zinsen, Auftragsgänge usw.) jeweils eine (sinnvolle) Handlungsvorschrift oder Folgerung (z.B. Aktie kaufen, halten oder verkaufen) zuzuordnen«.

Diese Arbeit baut auf der Technischen Analyse auf. Sie »lässt sich dahingehend charakterisieren, dass sie (nahezu) ausschließlich auf der Zeitreihe des Preises oder der Rendite des zu analysierenden Anlageobjektes aufsetzt« [Poddig 1999, 8]. Die Idee ist, dass alle relevanten Unternehmensdaten bereits in der Kursentwicklung (z.T. wird auch die Umsatzentwicklung einbezogen) enthalten sind und nicht wie bei der Fundamentalanalyse getrennt erhoben werden müssen.

Im Gegensatz dazu betont die *Random-Walk-Hypothese* die Wichtigkeit, Unternehmensdaten zu erfassen. Sie ist gleichzeitig die Formulierung der schärfsten Kritik an der technischen Analyse und geht auf Louis Bachelier und seine Schrift [Bachelier 1900] »Theorie de la speculation« aus dem Jahr 1900 zurück. Die orthodoxe Form der random-Walk-Hypothese geht davon aus, dass der zukünftige Wert einer Aktie  $j$  ausschließlich vom aktuellen Wert  $K_j(t)$  und dem Wert einer Zufallsvariablen  $e$  abhängt:

$$K_j(t + 1) = K_j(t) + e_j(t) \quad (1)$$

Dabei soll für  $e$  gelten:

1.  $e$  ist normalverteilt und die Häufigkeitsverteilung bleibt über die Zeit konstant.
2. Der Erwartungswert der Kursveränderung ist 0.
3. Aufeinander folgende Kursänderungen sind unabhängige Zufallsereignisse.

Als eine Vergleichsprognosestrategie wurde ein Handelsagent implementiert, der die Random-Walk-Hypothese als Grundlage seiner Kaufentscheidungen verwendet.

### 3 Datenuniversum

In dieser Arbeit werden SVMs zur Analyse und Prognose von Zeitreihen eingesetzt. Für eine Erläuterung der theoretischen Grundlagen sei auf [Vapnik 1998], [Vapnik&Chervonenkis 1968] und [Vapnik&Chervonenkis 1974] verwiesen. Eine ausführliche Erörterung des SVM-Algorithmus findet sich etwa in [Schölkopf 1997].

Als Datenbasis standen die täglichen, vollständigen Tagesschlusskurse von 1744 deutschen und internationalen Wertpapieren und Aktienindizes aus dem Zeitraum vom 1.1.1998 bis zum 30.12.2004 zur Verfügung. Diese Daten umfassten für jeden Handelstag und für jeden Kurs Start-, Maximal-, Minimal- und Schluss-Wert. Als Prognosezeitreihen wurden der *Deutsche Aktien Index* (DAX) und der *Dow Jones Industrial Average-Index* (DJIA) untersucht. Der Gesamtzeitraum wurde in fünf Simulationszeiträume aufgeteilt (s. Tabelle 3). Diese Zeiträume decken Abschnitte im Aktienmarkt mit unterschiedlichen Trends ab.

Tabelle 1: untersuchte Simulationszeiträume

Bezeichner	Beginn	Ende
A	01.06.2002	29.11.2002
B	01.06.2003	28.11.2003
C	01.06.2004	30.11.2004
D	01.12.2002	30.05.2003
E	01.12.2003	30.05.2004

#### Behandlung fehlender Werte

Im Gegensatz zu vielen Dataminingproblemen sind die Daten des hier verwendeten Kursuniversums an sich vollständig und korrekt. Durch unterschiedliche Feiertage in den USA und Deutschland kommt es allerdings mitunter zu Lücken an Tagen, an denen für einen der Handelsplätze kein Wert vorliegt, weil die Börse an diesen Tagen geschlossen war. Fehlende Werte durch Interpolation – bspw. mit Mittelwertbildung oder linearer Regression – zu ersetzen verfälscht prinzipiell die Eingabedaten. Für diese Arbeit wurde deshalb die Streichung gewählt. Der Informationsverlust wird als Preis in Kauf genommen für die Sicherheit, dass auf diese Weise die Prognosegüte im schlechtesten Fall lediglich unterschätzt wird.

### 4 Transformationen der Zeitreihendaten

Aus der Finanzanalyse sind zahlreiche Verfahren zur Vorverarbeitung von ökonomischen Zeitreihen bekannt. Eine wichtige Operation ist dabei die Trendbereinigung. Sie sorgt dafür, dass das maschinelle Lernverfahren als Eingabe nicht den absoluten Kurswert (gewissermaßen die Randfunktion des Kurses) sondern die Wertveränderung (die erste Ableitung) erhält. Der *Trend* bezeichnet dabei das Vorzeichen der ersten Ableitung, die *Rendite* den Betrag inkl. Vorzeichen.

Kompliziertere Indikatoren wie der Overbought/Oversold Oszillator oder der MACD-Trigger sind Beispiele aus der Finanzanalyse, wie sich auf rel. einfache Weise aus dem historischen Kursverlauf Handelssignale erzeugen lassen. Diese Indikatoren wurden in der vorliegenden Arbeit als Transformation für die Trainingsdaten der SVMs untersucht. Tabelle 2 gibt einen Überblick über alle für diese Untersuchung implementierten Transformationen und Indikatoren.

Tabelle 2: alle Zeitreihentransformationen im Überblick

Name	Funktionsbezeichnung
gleitender Durchschnitt	$GD_t(\bar{X}, n) = \frac{1}{n} \sum_{i=1}^n \bar{X}(t-i)$
stetige Rendite	$r_s(i) = \ln(\bar{Z}(i)) - \ln \bar{Z}(i-n)$
diskrete Rendite	$r_d(i) = \frac{\bar{Z}(i) - \bar{Z}(i-n)}{\bar{Z}(i-n)}$
absolutes Momentum	$AbsMom(\bar{X}, n, t) = \bar{X}(t) - \bar{X}(t-n)$
relatives Momentum	$RelMom(\bar{X}, n, t) = \frac{\bar{X}(t)}{\bar{X}(t-n)}$
Trendoszillator	$TO(\bar{X}, n, m, t) = \frac{avg(\bar{X}, t, n)}{avg(\bar{X}, t, m)}$
Trend	s.u.
relative Stärke	$RS(\bar{X}, n, t) = \frac{\bar{X}}{avg(\bar{X}, t, n)}$
relative Stärke Index	s.u.
Overbought/sold Oszil.	$\bar{Z}(t) \frac{\bar{X}(t) - \min(\bar{X}, t, n)}{\max(\bar{X}, t, n) - \min(\bar{X}, t, n)}$
MACD-Trigger	$\bar{T} = avg(avg(\bar{X}, t, n) - avg(\bar{X}, t, m), t, o)$
Volatilität	Standardabweichung $\sigma$ der logarithmierten Renditen
Entfernung vom Minimum	$\bar{Z}(t) = \bar{X}(t) - \min(\bar{X}, t, n)$
Entfernung vom Maximum	$\bar{Z}(t) = \max(\bar{X}, t, n) - \bar{X}(t)$
Addition	$\bar{Z} = ADD(\bar{X}, \bar{Y})$
Subtraktion	$\bar{Z} = SUB(\bar{X}, \bar{Y})$
zeitliche Verschiebung	$\bar{Z} = MV(\bar{X}, n)$

## Trend

$$Trend(i) = \text{sgn}(\bar{X}(i) - \bar{X}(i-1))$$

$$\text{sgn}(x) = \begin{cases} +1 & \text{falls } x > 0 \\ 0 & \text{falls } x = 0 \\ -1 & \text{falls } x < 0 \end{cases}$$

## Relative Stärke Index

$$G(t) = \sum_{i=t-n}^t \begin{cases} 1 & AbsMom(\bar{X}, i, n) \geq 0 \\ 0 & \text{sonst} \end{cases}$$

$$V(t) = \sum_{i=t-n}^t \begin{cases} 1 & AbsMom(\bar{X}, i, n) < 0 \\ 0 & \text{sonst} \end{cases}$$

$$\bar{Z}(t) = 1 - \frac{1}{1 + \frac{G(t)}{V(t)}}$$

### 4.1 Vektorisierung der Zeitreihendaten

Die Trainingsvektoren haben den in Formel 2 dargestellten Aufbau. Die erste Komponente bestimmt sich jeweils aus dem Wert der Zielzeitreihe ( $ZR_Z$ ) am Tag mit dem Index  $i$ . Danach folgen der Reihe nach die Werte aller  $l$  für die Trainingsmenge ausgewählten Zeitreihen  $ZR_1 \dots ZR_l$  am jeweils unmittelbar davor liegenden Tag  $i-1$ . Da der Trainingszeitraum vom Tag mit dem Index  $t-n-1$  bis zum Tag  $t-1$  reicht, läuft  $i$  im Intervall

$i \in [t - n, t]$ . Auf diese Weise wird eine Trainingsmenge mit  $n$  Vektoren der Dimension  $l + 1$  erzeugt.

$$\begin{pmatrix} ZR_Z(i) \\ ZR_1(i - 1) \\ ZR_2(i - 1) \\ \vdots \\ ZR_l(i - 1) \end{pmatrix} \quad (2)$$

Soll eine Prognose erzeugt werden, so wird aus den Zeitreihendaten für den aktuellen Handelstag  $t$  ein Testvektor erzeugt. In der ersten Komponente steht eine 0, weil sie die zu prognostizierende Zielklasse für ein Klassifikationsproblem darstellt.

## 4.2 Strategien für die Auswahl der Trainingsdaten

### 4.2.1 Naive Auswahlstrategie

Ein naiver Ansatz zur Auswahl der relevanten Eingabezeitreihen ist die Wahl der im jeweils prognostizierten Aktienindex abgebildeten Aktien. Zur Trendbereinigung wurde von jeder Zeitreihe das relative Momentum gebildet. Diese Auswahl kommt ohne den immensen Rechenaufwand der im folgenden Abschnitt beschriebenen Auswahl auf Basis von Korrelationstests aus.

### 4.2.2 Auswahl auf Basis von Korrelationstests

Dabei handelt es sich um ein empirisches statistisches Verfahren zur Bestimmung geeigneter Zeitreihen für die Modellbildung in induktiven maschinellen Lernverfahren im Kontext der Kursprognose [Baun 1994, 156f], [Poddig&Wallem 1994, 327], [Poddig 1999, 401ff]. Kerngedanke ist, dass eine Zeitreihe Informationen zur Prognose einer Zielzeitreihe liefern kann, wenn die beiden Zeitreihen in der Vergangenheit wenigstens zu einem gewissen Grad korreliert waren.

Für diese Arbeit standen rund 23 Mio. einzelne Datenpunkte aus 1744 verschiedenen Aktienkursen zur Verfügung. Aus der Kombination von mehreren Transformationen ergeben sich potentiell unendlich viele verschiedene Zeitreihen, die in die Trainingsmenge eingehen können. In [Schlitter 2006] werden mehrere Verfahren beschrieben, um diese Datenvielfalt nach ihrer Relevanz für die Prognose der jeweiligen Zielzeitreihe zu sortieren. Hier werden im folgenden nur einige wesentliche Eckpunkte dieser Verfahren beschrieben, für Details sei auf die genannte Arbeit verwiesen.

Auf jeden der zur Verfügung stehenden Kursverläufe wurden diverse Kombinationen von Eingabetransformationen angewendet. Die resultierenden Zeitreihen wurden zudem zeitlich zwischen null und neun Börsentagen verschoben (Timelags), um verzögerte Abhängigkeiten zwischen Eingabe- und Zielzeitreihe zu berücksichtigen. Die statistischen Tests wurden in den drei Zeiträumen A, B und C sowie mit 11 verschiedenen Zeitreihenlängen zwischen 11 und 520 Tagen auf allen zur Verfügung stehenden Zeitreihen durchgeführt. Auf diese Weise entstanden für DAX und DJIA jeweils

$$20\,143\,200 = \left( \overbrace{1744}^{\text{Wertpapiere}} \cdot \overbrace{35}^{\text{Transformationen}} + \overbrace{224}^{\text{Transformationen}} \right) \cdot \overbrace{10}^{\text{Timelags}} \cdot \overbrace{11}^{\text{Längen}} \cdot \overbrace{3}^{\text{Zeitpunkte}}$$

zu vergleichende Zeitreihen. Aus jeder dieser neuen Zeitreihen und der Zielzeitreihe wurden Paare gebildet, für die mit vier verschiedenen Tests die Korrelation berechnet wurde. Eine starke Korrelation ist ein Hinweis darauf, dass die Zeitreihe für die Trainingsmenge geeignet ist, weil ihr Verlauf ähnlich dem Verlauf der Zielzeitreihe ist. Zur Anwendung kamen vier verschiedene Testverfahren: Die Korrelationsanalyse nach *Bravis-*

*Pearson*, der *Delta-Test* nach [Pi&Peterson 1993], die *Average Mutual Information* nach [Krischer&Kruel 1992] sowie die *Signumparität*, die die Übereinstimmung des Prognosevorzeichens für jeden Handelstag überprüft.

Für die Berechnung dieser großen Anzahl von Korrelationen wurde auf eine eigene Implementation verzichtet. Statt dessen wurden die Ergebnisse aus [Schlitter 2006] übernommen, für deren Berechnung ca. 30 Einzel-CPU-Jahre Rechenzeit auf dem 512-Knoten-Cluster an der TU-Chemnitz aufgewendet werden mussten.

## 5 Simulationsumgebung

### 5.1 Handelsstrategie

Als gültige Handelssignale wurden Kaufen (**buy**), Verkaufen (**sell**) und keine Operation (**nop**) verwendet. Gehandelt wurde stets mit der Zielzeitreihe indem Anteile des Indexes ge- bzw. verkauft wurden. Das Signal **buy** führte immer dazu, dass für das gesamte zur Verfügung stehende Kapital Wertpapiere der Zielzeitreihe gekauft wurden. Dabei wurde berücksichtigt, dass immer nur diskrete Anzahlen von Wertpapieren gekauft werden können. Der übrig bleibende Restbetrag verblieb als freies Kapital. Die marktüblichen Transaktionsgebühren (im folgenden auch als Spesen bezeichnet) wurden berücksichtigt und bei jeder Transaktion automatisch in Rechnung gestellt. Das Signal **sell** führte immer dazu, dass alle zuvor gekauften Anteile abgestoßen und der Erlös nach Abzug der Spesen dem freien Kapital gutgeschrieben wurde.

Die Trainingsvektoren wurden den beiden disjunkten Klassen *steigend* (+1) und *fallend* (-1) zugeordnet. Das Klassifikationsergebnis für einen neuen Vektor ordnete dem Vektor folglich eine dieser beiden Klassen zu. Das verwendete SVMlight lieferte als Ergebnis jedoch nicht  $\pm 1$  sondern einen kontinuierlichen Wert, der als Konfidenz der Klassifikation interpretiert wurde. Alle Ergebnisse  $< 0$  wurden deshalb der Klasse *fallend* und alle Werte  $\geq 0$  der Klasse *steigend* zugeordnet.

Unterschritt das Klassifikationsergebnis im Fall  $< 0$  den Schwellenwert  $\epsilon_{sell}$  oder überschritt es im Fall  $\geq 0$  den Schwellenwert  $\epsilon_{buy}$  so wurde das entsprechende Handelssignal ausgelöst.

Statt auf die Details der Implementation einzugehen, sollen an Hand des Flussdiagramms in Abbildung 1 die notwendigen Schritte für die Erzeugung eines Handelssignals erläutert werden. Der Ablauf lässt sich grob in zwei Phasen aufteilen. In Phase 1 wird ein Modell auf Basis vergangener Handelstage erlernt. In Phase 2 wird dieses Modell auf die Daten des aktuellen Handelstags angewendet, um eine Prognose für den nächsten Handelstag zu erzeugen.

### 5.2 Systemarchitektur

Abbildung 2 zeigt ein stark abstrahiertes Schema der in *Perl*<sup>1</sup> implementierten Simulationsumgebung. Sie wurde objekt-orientiert realisiert. In ihr konnten verschiedene Handels-Agenten (im folgenden kurz *Agenten*) an einem simulierten Börsenhandel teilnehmen. Wurde ein Agent in die Simulationsumgebung eingefügt, so legte die Simulationsumgebung automatisch ein Depot mit 10000 Euro Startguthaben für ihn an. In Abb. 2 wurden vier verschiedene Agenten stellvertretend für bestimmte Handelsstrategien eingesetzt, die im folgenden kurz erläutert werden sollen.

**BuyAndHold** Eine der einfachsten Anlagestrategien ist, einfach alles Kapital in Aktien anzulegen und die Aktien erst wieder zu verkaufen, wenn man das Kapital für andere

---

<sup>1</sup>Im Internet: <http://www.cpan.org/>

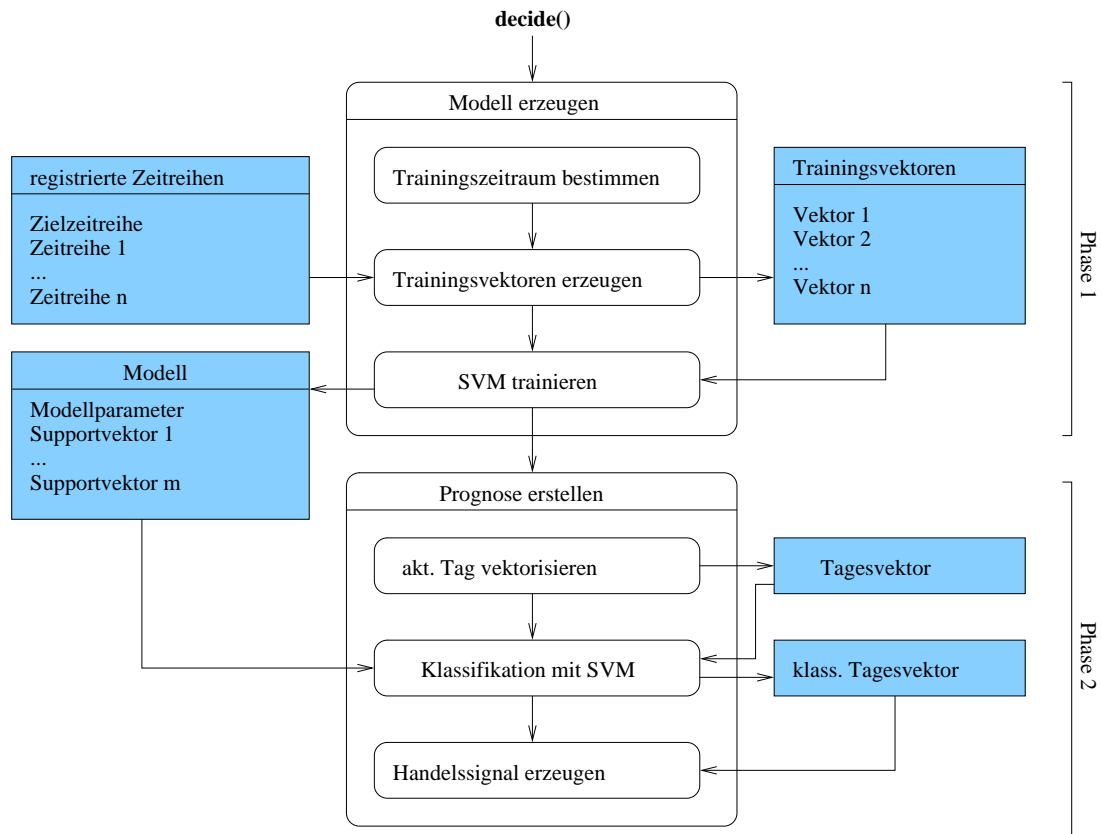


Abbildung 1: Flussdiagramm für Generierung eines Handelssignals mit SVMs

Zwecke braucht. Der Hintergrundgedanke ist, dass der Aktienmarkt tendenziell nach oben entwickelt. Diese Strategie versucht, diese Tendenz in Gewinn umzusetzen und vermeidet Spesen weitgehend.

**GD** Es wird ein gleitender Durchschnitt der Kursnotierungen der letzten  $n$  Tage gebildet. Schneidet der aktuelle Kurs diese Linie von unten nach oben wird ein Kaufsignal erzeugt; in der Gegenrichtung ein Verkaufssignal.

**SVMTP** Eine SVM wird eingesetzt, die jeden Tag neu mit den (transformierten) Kursdaten der letzten  $n$  Tage trainiert wird und an jedem Handelstag eine Handelsentscheidung treffen kann.

**SVMWP** Auch hier kommt eine SVM zum Einsatz, sie trifft aber nur einmal pro Woche eine Handelsentscheidung. Auf diese Weise soll vermieden werden, dass Kursgewinne durch die Spesen zu häufiger Kauf-Verkauf-Aktionen aufgeessen werden.

### 5.3 Statistik und Performanz-Messung

Für die Datenvorverarbeitung, sowie für die SVMs gab es diverse Parameter, die variiert werden konnten. Während jedes Simulationsdurchlaufes wurden diverse statistische Daten gesammelt und in einer zentralen Datenbank gespeichert. An jedem Handelstag wurde erfasst, welchen Wert das Aktiendepot jedes der registrierten Agenten gehabt hätte, wenn es an diesem Tag komplett liquidiert worden wäre. Um der großen Zahl zu testender Parameterkombinationen Herr zu werden, wurde das System für eine parallele Anwendung auf mehreren Rechenknoten angepasst. Insgesamt standen maximal 279 PCs (jeweils mind. 1 GB RAM und 1,8 GHz CPU) zur Verfügung.

## 6 Ergebnisse

Als Datenbasis standen nach Ende der Batch-Tests über 140 000 Simulationsergebnisse zur Verfügung. Da ein Simulationsergebnis jeweils das Handelsgeschehen eines halben Jahres abbildet, entspricht diese Zahl mehr als 70 000 Jahren simulierten Handelsgeschehens. In den Tests konnte gezeigt werden, dass mit SVMs auf handelsüblicher PC-Hardware problemlos ein täglich aktualisiertes Modell realisierbar ist. Die Trefferquoten eines solchen Modells erreichen bei geeigneten Parametern für Auswahlverfahren und SVM mit 60 bis 70 Prozent für Zeitreihen beachtliche Werte und schlagen die wesentlich rechenzeitintensiveren KNN aus [Schlitter 2006].

Gesucht wurde nach Parameterkombinationen, die die summierte Trefferquote in den drei Simulationszeiträumen A, B und C maximiert. Es handelt sich dabei um eine Suche in der Menge aller Ergebnisse. Diese Suche wurde getrennt für lineare, polynomielle und RBF-Kernel durchgeführt. Dabei wurden für jeden untersuchten Kernel der Mittelwert der Trefferquote und die Standardabweichung in den drei Simulationszeiträumen berechnet. Die Tabelle 3 fasst die Ergebnisse dieser Suche zusammen. Auf Grund des begrenzten

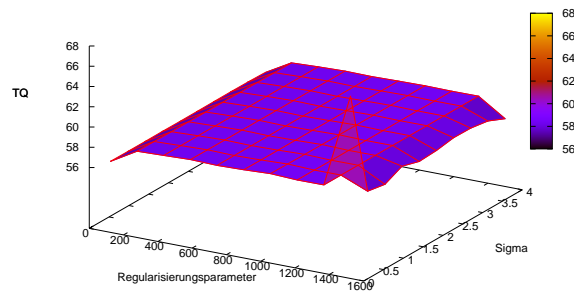


Abbildung 3: DJIA, Zeitraum A

Tabelle 3: Ergebnisse der Suche nach global optimalen Parameterkombinationen

Zielzeitreihe	DAX			DJIA		
<b>Kerneltyp</b>	linear	polynomiell	RBF	linear	polynomiell	RBF
<b>Regularisierungspara.</b>	100	100	500	100	1500	1000
<b>Polynomgrad</b>	0	2	0	0	3	0
$\sigma$	0	0	0.1	0	0	0.1
<b>Trainingszeitraum</b>	150	150	50	150	120	100
<b>sum. Trefferquote</b>	202.63	200.27	193.98	202.62	204.19	199.46
<b>avg.</b>	67.54	65.70	64.66	67.54	68.06	66.49

Platzes sei für eine erschöpfende Darstellung der Ergebnisse an dieser Stelle auf die vollständige Arbeit in [Möller 2006] verwiesen.

Beim Vergleich der Ergebnisse dieser Studie mit den Ergebnissen der KNN aus [Schlitter 2006] zeigt sich, dass die künstlichen neuronalen Netze den SVMs sowohl bei der Prognosegüte als auch bei der zum Training notwendigen Rechenzeit unterlegen sind. Der Vorteil rührt unter anderem aus dem deterministischen Verhalten der SVMs bei der Modellerzeugung, wohingegen die KNN mittels zufälliger Wahl der Kantengewichte initialisiert werden. Um den stochastischen Einfluss dieser zufälligen Initialisierung zu vermindern, werden Netze mit verschiedenen Kanteninitialisierungen trainiert, woraus folglich der höhere Rechenzeitbedarf resultiert.

Darüber hinaus vereinfacht das deterministische Verhalten der SVMs die Optimierung der zugehörigen Lernparameter. Die zu optimierende Funktion kann unmittelbar berechnet werden und muß nicht aus den Ergebnissen mehrerer Netze mit zufälliger Kanteninitialisierung aggregiert werden.

Ein weiterer Vorteil der SVMs ist deren bessere Generalisierungsfähigkeit. Anders als bei den Versuchen mit den KNN liegen die Prognosegüten für Trainings- und Testmenge dicht beieinander.

## 7 Fazit und Ausblick

Bei der Untersuchung hat sich gezeigt, dass Support Vector Machines grundsätzlich für die Analyse und Prognose ökonomischer Zeitreihen geeignet sind. Weiterhin konnte gezeigt werden, dass sich mit Handelsstrategien auf Basis dieser Prognosen grundsätzlich positive Renditen erzielen lassen. Die dabei erzielte Prognosegüte ist vergleichbar mit den Ergebnissen aus [Schlitter 2006]. Die zur Modellerzeugung benötigte Rechenzeit beträgt hingegen nur einen Bruchteil der zum Training der neuronalen Netze notwendigen.

Eine typische Schwachstelle von Prognosesystemen auf Basis maschineller Lernverfahren ist die Erzeugung von Handelssignalen aus den vom Modell erzeugten Prognosen. Typischerweise werden so viele Handelssignale erzeugt, dass die Renditen aus den korrekt prognostizierten Kursschwankungen durch die für jede Transaktion am Aktienmarkt anfallenden Handelsgebühren zunichte gemacht werden. Die Verbesserung der zugrunde liegenden Handelsstrategien steht somit im Zentrum zukünftiger Forschung, um die Praxistauglichkeit des hier beschriebene Verfahrens zu erreichen. Ein selbst adaptives System, das sich selbständige an neue Marktsituationen anpasst, ist hier wünschenswert. Das beschriebene Problem der hohen Transaktionskosten kann jedoch auch ökonomisch gelöst werden. Anstatt sich beim Handel ausschließlich auf Aktien zu konzentrieren, können beispielsweise Optionsscheine verwendet werden. Diese bieten die Möglichkeit sowohl auf fallende als auch auf steigende Kurse zu spekulieren. Der bei Optionsscheinen entstehende Hebeleffekt übersteigt die beim Aktienhandel möglichen Renditen um ein vielfaches und relativiert somit die anfallenden Transaktionskosten.

Sowohl [Möller 2006] als auch [Schlitter 2006] zeigen deutlich, dass die Qualität der Prognosen, maßgeblich von den verwendeten Prediktoren abhängt. Neue Methoden zur Auswahl geeigneter Zeitreihen als Eingabe für den Modellerzeugungsprozess sind deshalb unabdingbar. Darüber hinaus erwarten wir eine Steigerung der Prognosegüte, wenn neben der Verwendung von ökonomische Zeitreihen auch aktuelle Wirtschaftsmeldungen bei der Modellerzeugung berücksichtigt werden. Die Arbeiten von Kroha [Kroha et al. 2005, Kroha et al. 2006] auf dem Gebiet des Textminings beinhalten dahingehend vielversprechende Ergebnisse.

## Literatur

- [Bachelier 1900] Bachelier, Louis, *Theorie de la spéculation*, Dissertation, Université de Paris, Paris, 1900.
- [Baun 1994] Susanne Baun: »Neuronale Netze in der Aktienkursprognose«, in: »Neuronale Netze in der Ökonomie«, herausgegeben von Rehkugler, H. & Zimmermann, H. G., Seiten 131-207, Vahlen, München, 1994.
- [Krischer&Kruel 1992] Krischer, K. & Kruel, Th.-M.: »Ein Programm zur Berechnung der gegenseitigen Information aus Zeitserien, die durch die Takens'sche Versatzzeitmethode auseinander hervorgegangen sind«, 1992.
- [Kroha et al. 2005] Petr Kroha and R. Baeza-Yates. A case study: News classification based on dexa'2005, workshop on theory and applications of knowledge management takma'2005. In *Proceedings of 16th International Conference*, pages 428–432, Copenhagen, August 2005. IEEE Computer Society.
- [Kroha et al. 2006] Petr Kroha, R. Baeza-Yates, and Björn Krellner. Text mining of business news for forecasting. In *Proceedings of Seventh International Workshop on Theory and Applications of Knowledge Management*, September 2006.

- [Mohan et al. 2005] Mohan, N., Jha P., Laha, A.K., Dutta, D.: »Artificial Neural Network Models for Forecasting Stock Price Index in Bombay Stock Exchange«, Indian Institute of Management Ahmedabad, Research and Publication Department in its series IIMA Working Papers with number 2005-10-01
- [Möller 2006] Möller, Manuel: »Analyse und Prognose ökonomischer Zeitreihen mit Support Vector Machines«, Diplomarbeit, TU-Chemnitz
- [Perridon 2002] Perridon, L.: »Finanzwirtschaft der Unternehmung«, 11. überarbeitete und erweiterte Auflage, Vahlen Verlag, München
- [Pi&Peterson 1993] Pi, Hong & Peterson, Carsten: »Finding the Embedding Dimension and Variable Dependencies in Time Series«, in: Neural Computation, 6:509-520, 1993.
- [Poddig 1999] Poddig, Thorsten: »Handbuch Kursprognose. Quantitative Methoden im Asset Management«, Uhlenbruch Verlag, Bad Soden, 1999
- [Poddig&Wallem 1994] Poddig, Thorsten & Wallem, Axel: »Wechselkursprognosen« in: »Neuronale Netze in der Ökonomie«, herausgegeben von Rehkugler, H. & Zimmermann, H. G., Seiten 291-336, Vahlen, München, 1994.
- [Rehkugler&Poddig 1994] Rehkugler, H & Poddig, Th.: »Kurzfristige Wechselprognosen mit künstlichen neuronalen Netzwerken« in Bol, G., Nakhaeizadeh, G. & Vollmer, K.-H.: »Finanzmarktanwendungen neuronaler Netze und ökonometrischer Verfahren«, S. 1-24, Heidelberg
- [Schlitter 2006] N. Schlitter: »Analyse und Prognose ökonomischer Zeitreihen mittels evolutionär optimierter Backpropagation-Netze«, Diplomarbeit, TU-Chemnitz
- [Schölkopf 1997] Bernhard Schölkopf: „Support Vector Learning“, GMD-Forschungszentrum Informationstechnik GmbH, München, Wien, Oldenbourg
- [Vapnik 1998] V.N. Vapnik: »Statistical Learning Theory«, Jon Wiley & Sons Inc., New York, USA
- [Vapnik&Chervonenkis 1968] V.N. Vapnik & A. Chervonenkis: »Uniform convergence of frequencies of occurrence of events to their probabilities«, Dokl. Akad. Nauk, SSSR, Ausgabe 181, S. 915-918
- [Vapnik&Chervonenkis 1974] V.N. Vapnik & A. Chervonenkis »Theory of Pattern Recognition«, Nauka, Moscow (deutsche Übersetzung: W. Vapnik & A. Tscherwonkis: »Theorie der Zeichenerkennung«, Akademie-Verlag, Berlin 1979
- [Yao&Tan 2001] Yao, Jing Tao & Tan, Chew Lim: »Guidelines for Financial Forecasting with Neural Networks«, Proceedings of International Conference on Neural Information Processing, Shanghai, China, 14-18 November, 2001, pp757-761

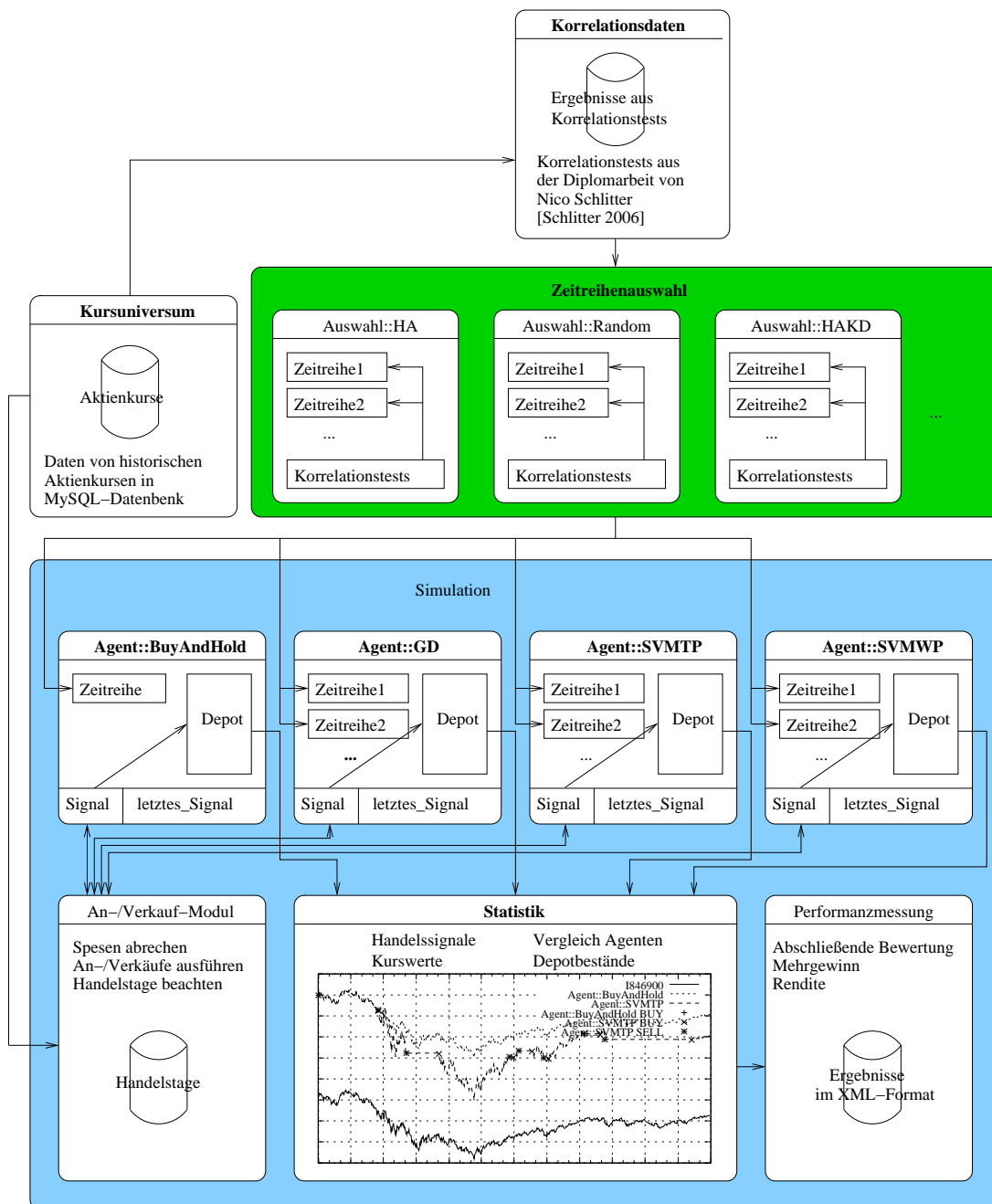


Abbildung 2: Architektur der Simulationsumgebung