

Diplomprojekt
2012/2013

NOCTUA

Lastenheft

Nagy, Pawlowsky & Sochovsky

VERSIONIERUNG

Version	Autor	Datum	Status	Kommentar
0.1	Nagy	17.08.12	draft	Präambel, Zielbestimmungen, Produkteinsatz
0.2	Nagy	20.08.12	draft	Funktionen, Überarbeitung
0.3	Nagy	21.08.12	draft	Funktionen, Daten, Qualitätsanf., Glossar
0.4	Nagy	02.09.12	draft	Randbedingungen, Vertragsggst.
1.0	Nagy	07.09.12	draft	Produktdaten ergänzt, Nutzungs-/Vertribrsr.
1.1	Pawlowsky	10.09.12	final	Qualitätssicherung

INHALT

1. Präambel.....	1
2. Zielbestimmungen	2
3. Produkteinsatz	3
4. Produktfunktionen	4
4.1. Funktionen der Markzustandsbestimmung	4
4.2. Funktionen des Trading-Algorithmus.....	4
4.3. Funktionen der Backtesting-Software	6
5. Produktdaten	6
6. Zwingende Randbedingungen	7
6.1. Produktumgebung und Systemintegration	7
6.2. Schnittstellen	7
7. Vertragsgegenstand	8
7.1. Lieferumfang.....	8
7.2. Produktleistungen.....	8
7.3. Produktbezogene Leistungen	8
8. Qualitätsanforderungen.....	9
9. Glossar	10

1. PRÄAMBEL

Der Handel mit Wertpapieren ist in den letzten Jahren und Jahrzehnten zunehmend systematisiert und automatisiert worden. Kaum jemand trifft Handelsentscheidungen leichtfertig aus dem Bauch heraus ohne fundierte Analyse. Diese Analyse unterwirft sich aber damit einem programmatischen Schema, das ebenso gut auch automatisch, algorithmisch angewandt werden kann: trifft ein Mensch Entscheidungen nach einem genauen Schema, kann ein Computer dies ebenso und dabei sogar schneller und genauer.

Besonders gut geeignet dafür scheinen die technische Analyse, besonders die Trendbestimmung und das Trendfolgen. Auch wenn es reichlich Kritik an solchen Systemen gibt (besonders, dass sich Aktienkurse nach keiner bekannten statistischen Verteilung bewegen), wenden sehr viele Marktteilnehmer solche Systeme an. Das führt zumindest teilweise aber zu einer selbsterfüllenden Prophezeiung, da sich die Kurse am Verhalten der Majorität der Marktteilnehmer orientieren.

Ein weiterer Vorteil des Algorithmischen Trading ist die Geschwindigkeit, sowie Genauigkeit mit der Computer arbeiten können, an die Menschen nicht heranreichen. Durch systematische und statistische Entscheidungen können menschliche Emotionen aus dem Spiel gelassen und dadurch auch das Risiko besser abgeschätzt werden.

2. ZIELBESTIMMUNGEN

Es soll ein Algorithmus zur vollautomatischen Bestimmung von möglichst profitablen Handelsaktionen (Trades) auf transparenten Handelsmärkten, primär dem Aktienmarkt entwickelt werden. Auf Derivate muss dabei nicht gesondert eingegangen werden. Um dieses Ziel zu erreichen soll Forschungsarbeit auf dem Gebiet der technischen Analyse, sowie Marktzustandsbestimmung und deren Implikationen auf die Kursbewegungen von börsennotierten Handelspapieren unternommen werden. Die Technologie soll primär für kurze Perioden (Intra-Day bzw. Short-Term) entwickelt werden, sollte sich jedoch ebenfalls während des Projektes eine Eignung für längerfristige Strategien ergeben, wäre dies vorteilhaft.

Dabei sollen mögliche Anwendungen von verschiedenen Moving-Averages (MAs, Gleitender Durchschnitt), Oszillatoren zur Support- und Resistance-Level-Bestimmung und andere mehr oder weniger häufig genutzte Daten zur algorithmischen Entscheidungsfindung herangezogen werden. Ebenso soll geforscht werden, ob es möglich ist realwirtschaftliche Ereignisse in ein solches System zu integrieren und somit den Gewinn zu optimieren.

Diese üblichen technischen Indikatoren und damit verbundene Tradingstrategien sollen auf Performance und Risiko überprüft und mögliche Optimierungen erkannt und umgesetzt werden.

Um diese Größen vergleichbar zu machen, soll Software entwickelt werden, die als Backtesting-Modul fungiert und anhand von historischen Kursverläufen relevante Kennzahlen und Messgrößen errechnet.

Märkte verhalten sich in unterschiedlichen Zeitperioden und unter anderen Randbedingungen unterschiedlich. Zeitweise so stark, dass Tradingstrategien, die zeitweise gut oder sogar sehr gut funktionieren unter anderen Randbedingungen wesentlich niedrigere Erträge einbringen oder sogar Verluste verursachen. Um genau das zu Vermeiden und die Volatilität und damit das Risiko zu verringern soll versucht werden diese Randbedingungen zu bestimmen und somit spezielle Markzustände zu identifizieren. Schließlich soll der Algorithmus daraufhin gehend optimiert werden, um die besten Entscheidungen nicht nur mikroökonomisch durch den Kurs der gehandelten Aktien, sondern auch makroökonomisch durch übergeordnete Trends wie beispielsweise Kursverläufe großer Indizes oder Zinssätze zu erreichen.

Um die Ziele zusammenzufassen kann von folgendem Entwurf ausgegangen werden:

- Erkennen von unterschiedlichen **Marktzuständen**
- Entwickeln eines *parametrisierbaren* **Algorithmus** (Signalgenerator)
Beispielsweise soll der Algorithmus für unterschiedliche Marktzustände andere Parametersätze anwenden.
- Entwickeln von Software zur Prüfung von Performance und Risiko (**Backtesting**)
Um die Forschung dahingehend zu unterstützen, dass fundierte Entscheidungen und Optimierungen getroffen werden können, sowie die Ergebnisse der Forschung objektiv bewerten zu können ist solch eine Software unumgänglich.

3. PRODUKTEINSATZ

Der Algorithmus soll prinzipiell auf volatile und transparente Märkte ausgelegt werden, da auf solchen kurzzeitige Strategien erfolgsversprechender sind.

Die Signale sollen vollautomatisch handelbar sein, dennoch ist besonders bei großen Summen eine Beaufsichtigung eines Experten empfehlenswert.

Die Software soll serverseitig eingesetzt werden, weshalb diese mit der Leistung von Servern der Stand der Technik zum Zeitpunkt der Fertigstellung praktikabel sein muss, um in der zur Verfügung stehenden Zeit zu Entscheidungen zu gelangen.

4. PRODUKTFUNKTIONEN

4.1. FUNKTIONEN DER MARKZUSTANDSBESTIMMUNG

/LF10/

Vergangene Marktzustände bestimmen

Es sollen historische Marktzustände (innerhalb der letzten Jahre), auf transparenten Aktienmärkten, für die ein ausreichender Datenbestand vorhanden ist, automatisch bestimmt werden. Sollten sich verschiedene große Märkte entgegen Erwartung entsprechend unterschiedlich verhalten, dass diese keiner einheitlichen Analyse unterzogen werden können, soll primär der US-amerikanische Aktienmarkt untersucht werden. Hierbei handelt es sich um eine Gruppierung von Zeitabschnitten nach gemeinsamen Kriterien.

/LF20/

Aktuellen Marktzustand bestimmen

Dabei soll darauf geachtet werden, dass für eine frühe Erkennung möglicherweise nur ein Teil der Daten vorhanden ist, die für die historische Analyse herangezogen werden.

4.2. FUNKTIONEN DES TRADING-ALGORITHMUS

/LF110/

Trends erkennen

Durch Moving Averages soll es möglich sein Trends in Aktienkursen zu identifizieren. Dazu kommen verschiedene Crossover-Verfahren (double- / triple-crossover) oder Indikatoren, wie der MACD (Moving Average Convergence Divergence) in Frage. Es soll eine statistisch möglichst profitable Variante hierfür gefunden werden, die aufscheinende nachhaltige Trends möglichst günstig erkennt.

/LF120/

Moving-Average-Dauer bestimmen

Je nachdem, wie lange ein Trend andauert, bedingt eine Trenderkennung andere MA(-Paare) mit unterschiedlichen Laufzeiten. Durch Backtesting sollen viele verschiedene Varianten automatisch getestet werden können, um den statistisch besten Parametersatz zu ermitteln.

/LF130/

An Marktzustand anpassen

Der Algorithmus soll sich durch Parameterveränderung an den erkannten Marktzustand zur Optimierung der Performance anpassen. Dies kann beispielsweise durch verändern der MA-Paare oder durch Anpassung der Market Exposure und damit des Risikos erfolgen.

Dazu *können* die Implikationen durch Nachforschung bekannt sein, woraufhin ein Modell angewandt wird, müssen aber nicht, da auch induktiv aus den Implikationen gelernt werden kann, wonach automatisch ein Modell entsteht. (*Maschinelles Lernen*) Dabei werden für die unterschiedlichen Marktzustände verschiedene Parametersätze durchprobiert.

/LF140/

Signale generieren

Signalgeben bei potentiellen Einstiegspunkten (long signal) und Ausstiegspunkten (short signal).

/LF150/

Trend-Nachhaltigkeit bestimmen

Durch geeignete Support- und Resistance-Indikatoren soll die Nachhaltigkeit eines Trends bestimmt werden (beispielsweise *Pivot Points*, *RSI*, *CCI* oder *MAs*), um den Ausstiegspunkt zu optimieren.

/LF160/

Signale filtern

Zur Verminderung von unprofitablen, zu kurzen Trades sollen insbesondere Kaufsignale gefiltert werden. Die Trenderkennung könnte das Öfteren zu kurz anhaltende Trends erkennen, indem beispielsweise ein MA-Crossover nur für kurze Zeit besteht. Durch das Einführen eines Schwellenwertes (threshold), der überschritten werden muss, oder eine bestimmte Zeitspanne, die ein Signal überdauern muss können zu kurze Trades vermindert werden, wenn sich im Backtesting dadurch ein Vorteil herausgestellt hat.

4.3. FUNKTIONEN DER BACKTESTING-SOFTWARE

/LF210/

Performance berechnen

Die relative Performance eines Algorithmus soll in Prozent der Kapitalveränderung berechnet werden.

/LF220/

Gewinn/Risiko-Verhältnis berechnen

Bestimmung des Risikos des Algorithmus (beispielsweise anhand der Volatilität) in Verbindung mit der Performance (e.g. sharpe ratio).

5. PRODUKTDATEN

/LD10/

Historische Kursdaten

Historische Kursdaten mehrerer Aktien, sowohl für die Algorithmus-Entwicklung, als auch für die Backtesting-Software sollen gespeichert werden.

/LD20/

Algorithmen

Um auch rückwirkend auf ältere oder andere Versionen der Algorithmen zugreifen zu können, um beispielsweise Teile davon in neuen Versionen wieder zu implementieren oder auf ältere Versionen zurückzustellen, müssen diese inklusive der nötigen Zusatzinformationen gespeichert werden.

/LD30/

Algorithmen-Performance-Mapping

Die Ergebnisse der Backtesting-Software sollen für jede durchgerechnete Algorithmus-Version gespeichert werden und sollen diesen auch eindeutig wieder zugeordnet werden können. Diese Daten könnten beispielsweise in Files oder einer Datenbank gespeichert werden.

6. ZWINGENDE RANDBEDINGUNGEN

6.1. PRODUKTUMGEBUNG UND SYSTEMINTEGRATION

Der fertiggestellte Algorithmus soll in der Programmiersprache F# implementiert werden, um die Integration in eine .NET-Systemumgebung zu ermöglichen. Die Kompatibilität mit anderen Betriebssystemen als Windows kann dabei vernachlässigt werden. Für den Produktiveinsatz soll die Software auf einem Server laufen, wodurch höhere Leistungsansprüche gestellt werden können, als wenn ein Client-System zum Einsatz käme.

6.2. SCHNITTSTELLEN

Für die Entwicklung und somit den gesamten Verlauf dieses Projektes kann auf heruntergeladene Kursdaten zugegriffen werden, eine Anbindung an einen Online-Datenprovider ist also nicht vonnöten. Wichtig ist aber, dass die Schnittstelle zu den Daten möglichst generisch gehalten wird, um eine einfache zukünftige Anbindung eines solchen problemlos in kurzer Zeit abschließen zu können.

Verschiedene Algorithmen sollen möglichst einfach und schnell an die Backtesting-Software angebunden werden können, um deren Performance zu prüfen. Dafür kommen verschiedene Möglichkeiten in Frage, u.a. ein Webservice, eine API für direkten Zugriff oder das Laden des Algorithmus als DLL. Im Zuge des Projektes soll hierfür eine zufriedenstellende Lösung gefunden werden.

7. VERTRAGSGEGENSTAND

7.1. LIEFERUMFANG

Der Lieferumfang umfasst den Code des finalen Algorithmus, sowie der Backtesting-Software und außerdem eine ausreichende Dokumentation der Funktionsweise des Algorithmus. Es ist vorgesehen den Algorithmus in andere eigene Produkte zu integrieren, somit bleiben die vollen Nutzungs- und Vertriebsrechte bei den Erzeugern.

7.2. PRODUKTLEISTUNGEN

/LL10/

Minimale Performance

Die Performance des Algorithmus soll statistisch über die letzten 3 Jahre einen höheren Ertrag erzielen, als der risikofreie Zinssatz.

/LL20/

Backtesting Kursdaten

In der Backtesting-Software sollen mindestens 3 verschiedene Aktienkurse zur Verfügung stehen, wobei mindestens einer ein Index sein soll.

7.3. PRODUKTBEZOGENE LEISTUNGEN

Für dieses Projekt ist keinerlei Installations- und Wartungsarbeit vorgesehen. Eine Integration des Algorithmus in andere Produkte, sowie ein Vertrieb oder das Anbieten von Dienstleistungen durch die Verwendung der Analyse-Backtesting-Software sind nicht inbegriffen.

8. QUALITÄTSANFORDERUNGEN

Produktqualität	Sehr Gut	Gut	Normal	Irrelevant
Funktionalität		x		
Zuverlässigkeit		x		
Benutzbarkeit				x
Effizienz		x		
Änderbarkeit		x		
Übertragbarkeit				x

9. GLOSSAR

Algorithmus	Ein Algorithmus ist eine aus endlich vielen Schritten bestehende eindeutige Handlungsvorschrift zur Lösung eines Problems oder einer Klasse von Problemen.
Backtesting	Ein Algorithmus wird auf eine (historische) Datenreihe angewandt.
Derivat	Ein Finanzinstrument, dessen Wert von einem anderen Handelsgut (underlying) abhängt. (z.B. Futures oder Optionen)
Intra-Day-Trading	Kurzfristiger, spekulativer Handel mit Wertpapieren, wobei Positionen innerhalb eines Tages eröffnet und wieder geschlossen werden
MA	s. Moving Average
Market Exposure	Proportion, zu dem ein Portfolio den Schwankungen eines Marktes ausgesetzt ist
Marktzustand	Stimmung und Verhalten der Marktteilnehmer bei bestimmten Randbedingungen während einer Zeitperiode
Moving Average	MA; Gleitender Mittelwert/Durchschnitt: Tiefpassfilter, der dazu verwendet wird um Kurse zu glätten und temporäre Schwankungen zu entfernen. Dazu wird für jeden zusätzliche Wert mit den x vorhergegangenen Werten ein Durchschnitt berechnet, wodurch der MA dem Kurs hinterherhinkt.
Oszillator	Indikator, der periodische Wiederholungen aufweist bzw. in einem Bereich um einen Wert schwankt
Performance	Güte des Algorithmus anhand von Gewinn, Volatilität, Risiko und anderen Faktoren
Risiko	Wahrscheinlichkeit, mit der Gewinn oder Verlust eintritt
Support- und Resitance	Level, bei denen sich Angebot und Nachfrage voraussichtlich treffen und ein Trend nicht mehr als nachhaltig angesehen wird.

Technische Analyse	Vorhersage von Börsenkursen anhand von bereits bekannten Kennzahlen
Trade	Entweder ein Kauf (long) oder Verkauf (short), der auf eine Order folgt.