

**Utveckling av portföljstrategier baserade på svagt
kointegrerade finansiella instrument med AdaBoosting**

Helena Nilsson

Februari 15, 2009

Abstract

Financial analysts are constantly trying to find new trading strategies in order to increase their revenue. In recent years theories involving mean-reverting have grown to be successful and especially pairs trading. However, in time these models become well-known and commonly used, which will decrease the revenue and ultimately the models become harder to master. In that case, one might use more advanced techniques and strategies to combine the characteristics of the spread in addition to mean-reverting techniques to find trading strategies with a more stable out-of-sample result.

This master thesis examines if the AdaBoost algorithm can help create portfolios with stable revenue. The thesis explains how the ensemble learning algorithm AdaBoost is transformed into an optimizing algorithm which efficiently combines different qualities of the spread into a trading strategy. First the thesis gives a short presentation of the AdaBoost algorithm and later describes how the algorithm is implemented due to chosen trading signals. Then the trading strategy is trained and tested on different stock indexes. Finally the result of the trading strategy is presented; it shows positive revenue on tested stock indexes.

Sammanfattning

Finansanalytiker försöker ständigt hitta nya modeller och handelsstrategier för att öka sin avkastning. På senare år har teorier som bygger på mean-reverting tekniker blivit populära och speciellt pairs trading. Efterhand blir dessa modeller kända och välanvända och då minskar avkastningen och de blir allt svårare att bemästra. För att motverka detta tvingas man därför att utveckla mer avancerade strategier som kan kombinera och använda sig av andra egenskaper hos spreaden, utöver mean-reverting, för att hitta en handelsstrategi som ger stabila out-of-sample resultat.

Detta examensarbete undersöker hur man med hjälp av AdaBoost algoritmen kan skapa portföljer med stabil avkastning. Rapporten beskriver hur klassificeringsalgoritmen AdaBoost görs om till en optimeringsalgoritm som genom att kombinera flera olika egenskaper hos spreaden effektivt kan ta fram en optimal handelsstrategi. I rapporten görs en kortfattad presentation av AdaBoosten algoritmen för att sedan beskriva implementeringen av algoritmen med hänsyn till valda handelssignaler. Tillslut tränas och testat handelsstrategin på undersökta aktieindex och resultatet presenteras. För den undersökta perioden uppvisar handelsstrategin ett positivt riskjusterat resultat.

Förord

Detta examensarbete har gjorts inom inriktningen finansiell matematik på Matematik Institutionen vid Kungliga Tekniska Högskolan. Examensarbetet omfattar 30 poäng och har utförts på Första AP-fonden.

Jag vill rikta stort tack till min handledare Peter Raicevic på Första AP-fonden. Jag vill också tacka min examinator Timo Koski vid KTH.

Stockholm den 15 februari 2009

Helena Nilsson

Innehåll

1 Inledning	1
1.1 Bakgrund	1
1.2 Problemformulering	1
1.4 Avgränsningar	2
1.5 Disposition	2
2 Teori	3
2.1 Kointegration	3
2.2 Pairs Trading	5
2.3 Standard AdaBoosten	6
2.4 Simulated Annealing	8
3 Metod	9
3.1 Data	9
3.2 Beräkning	9
3.3 Handelsstrategi	10
3.3.1 Generaliserad AdaBoost	10
3.4 Handelssignaler	12
3.4.1 EMA Mean Reverse	12
3.5 Stokastisk sökalgoritm	13
4. Resultat	14
4.1 Avkastningsberäkning	14
4.2 In-sample resultat från datamängd 1	14
4.2.1 Resultat från träningsperioden	14
4.2.2 Resultat från AdaBoosten	15
4.2.3 Resultat från testperioden	17
4.3 Out-of-sample resultat	24
4.3.1 Out-of-sample resultat för datamängd 1	25
4.3.2 Out-of-sample resultat för datamängd 2	27
4.4 Ackumulerad avkastning	30
4.5 Bootstrap metoden	33
5 Analys	36
5.1 Analys av in-sample resultatet	36
5.2 Analys av out-of-sample resultatet	36
6 Slutsats	38
Referenslista	39

1 Inledning

Detta kapitel börjar med en bakgrund till behovet av nya handelsstrategier. Efter det kommer problemformulering och avgränsningar att presenteras. Till sist, ges uppsatsens disposition.

1.1 Bakgrund

På senare år har tekniker som bygger på mean-reverting och speciellt pairs trading blivit alltmer populära. Till följd av detta har typiska par och mängder av kointegrerade instrument blivit välkända, vilket leder till att felprissättningar försvinner från marknaden. Antalet upptäckta fungerande kombinationer minskar kraftigt och detta minskar chansen till positiv avkastning. Därför strävar man efter att utveckla nya handelsstrategier som utnyttjar att spreaden har andra egenskaper utöver mean-reverting så som multi-scale trender och andra komplexa dynamiker. Genom att använda en mer allmän handelsstrategi som kan handla med flera olika typer av komplexa spreadtidsserier försöker man öka sin avkastning och stabiliteten hos out-of-sample-resultatet jämfört med resultatet hos pairs trading.

1.2 Problemformulering

Det här examensarbetet har som uppgift att undersöka om det går att använda en handelsmetod baserad på AdaBoost-algoritmen, istället för den förhärskade mean-reverting metoden. AdaBoost är en inlärningsalgoritm som uppmuntrar nya modeller att bli experter på områden som redan använda modeller inte är så bra på. Den grundläggande idén är att algoritmen kombinerar relativt enkla bashypoteser/modeller till en slutgiltig förutsägelse, en stark basmodell. En mer ingående förklaring av AdaBoosten finns under kapitel 2.3.

Examensarbetet baseras på artiklarna [9] och [13] som undersöker om AdaBoost-algoritmen kan användas i handelssammanhang. Artiklarna testar handelsmodellen på två aktieindex, S&P mid-cap (MID) och S&P 500 index (SPX) vilket resulterar i en positiv riskjusterad avkastning under den i artiklarna undersökta tidsperioden. Detta examensarbete har till uppgift att skapa en liknande handelsmodell som i artikeln, samt att testa den på andra marknader.

Precis som i artiklarna kommer examensarbetet att använda sig av databrytning, dvs. vi söker efter en bästa lösning på historiska data. De parameterinställningarna som fås fram kommer senare att användas för att utvärdera handelsmodellen på ytterligare marknader.

Examensarbetet vill undersöka om AdaBoost-algoritmen är en lönsam strategi vid aktiehandel.

1.4 Avgränsningar

Detta examensarbete behandlar två olika datamängder. Den första datamängden består av två stycken aktieindex och den andra datamängden består av 10 stycken aktieindex. Dessa 10 kommer att paras ihop slumpmässigt. Av tidsskäl hinner jag bara analysera 14 stycken av dessa par. Resterande aktieindexpar lämnas till ytterligare analys av handelsmodellen.

Uppsatsen tar inte hänsyn till de kostnader som uppstår vid utförandet av strategin, till exempel transaktionskostnader.

1.5 Disposition

Uppsatsen är indelad i sex kapitel. Efter inledningskapitlet behandlas teori och redogörs för tidigare forskning. I kapitel tre beskrivs hur data inhämtats, behandlats och vilken metod som använts. Det fjärde kapitlet innehåller resultatet, i det femte analyseras resultatet och i det sista kapitlet redovisa slutsatser samt ges förslag på vidareutveckling av examensuppgiften.

2 Teori

Nedan redogörs för några befintliga modeller som finns för beräkning av portföljstrategier. Sedan redogörs för AdaBoost algoritmen, den algoritm som detta examensarbete bygger på. Kapitlet tar också upp Simulated Annealing, en optimeringsalgoritm som kan användas för optimering av portföljer.

2.1 Kointegration

Kointegration handlar om relationer mellan tidsserier. Att två tidsserier är kointegrerade betyder att det finns ett långsiktigt samband mellan dem. Tidsserierna följer varandra, men de behöver inte följa varandra synkront varje dag utan de kan under kortare perioder vandra iväg i olika riktning förutsatt att de inom kort återkommer till ett jämviktsläge [1].

Den mest populära metoden för att testa om kointegration förekommer är OLE (Ordinary Least Square) regression vilken har utvecklats av Engler och Granger. Ett primärt krav är att tidsserierna är icke-stationära av samma grad [4].

För att förstå begreppet kointegration måste man därför först förstå skillnaden mellan en stationär och en icke-stationär tidsserie. Det gör man lättast genom att titta på definitionen för en stationär (svagt stationär) tidsserie, Y_t som enligt [2] definieras av:

1. $E(Y_t) = \mu$
2. $Var(Y_t) = E(Y_t - \mu)^2 = \sigma^2 < \infty$
3. $Cov(Y_{t+h}, Y_t) = \gamma_h = E[(Y_t - \mu)(Y_{t+h} - \mu)]$

Enligt punkterna 1 och 2 ska en stationär tidsseries förväntade medelvärde och varians vara konstanta och oberoende av tiden t . Punkt 3 kräver att kovariansen mellan två Y -värden är densamma oberoende av vilket Y -värde man använder som utgångspunkt. I praktiken betyder detta att om man chockar en stationär tidsserie så kommer chocken inte ge några permanenta effekter på tidsserien utan tidsserien kommer inom kort att återgå till sitt jämviktsläge. Att chocka en tidsserie avser både negativa och positiva händelser som påverkar värdet på tidsserien. Större och kraftigare chocker på tidsserien ger upphov till större volatilitet än om chockerna varit mindre kraftiga. Många makroekonomiska tidsserier är icke stationära och på dessa tidsserier ger en eventuell chock permanenta effekter [3].

Då de tre ovanstående kraven inte uppfylls är den aktuella tidsserien icke-stationär vilket gör att regressionsresultat inte är pålitliga vid hypotesprövningar och att påvisade samband mellan tidsserierna kan bero på falsk regression pga. trenden [4]. Med falsk regression menas att de statistiska testerna, t - och F -tester, ger felaktigt utslag på grund av att trenden gör att det finns samband mellan variablerna som egentligen inte existerar. Prognoser av testen är inte heller tillförlitliga eftersom prognosmodeller antar att chocker är temporära när de egentligen är permanenta.

För att få bukt med detta har man utvecklat metoder för att omvandla icke-stationära tidsserier till stationära. Detta görs genom differentiering enligt formeln:

$$\Delta y_t = y_t - y_{t-1}$$

Icke-stationära och stationära tidsserier betecknas I(d), där d står för antalet differentieringar som krävs för att en icke-stationär tidsserie ska bli stationär. I praktiken är de flesta makroekonomiska tidsserier av graden I(1) och behöver därför differentieras en gång [1].

Genom att utföra ett Dickey-Fuller (DF) test kan man undersöka huruvida respektive serie är stationär [1]. Testet utgår ifrån en modell där serien y bestäms av:

$$y_{t+1} = \alpha + \beta y_t + s_t$$

y är värdet vid tidpunkten t+1 som genom regression relateras till värdet på y vid tidpunkten innan, dvs. t och s_t är en felterm [5]. Om $\beta=1$ har den aktuella tidsserien en enhetsrot (eng. unit root) vilket innebär att den är icke-stationär [6]. I praktiken utförs testet genom följande regression:

$$y_{t+1} - y_t = \Delta y_t = \alpha + \beta y_t + \sum_{j=1}^{p-1} \beta_j \Delta y_{t-j} + s_t \quad (\text{Augmented-Dickey-Fuller, ADF})$$

Om den uppskattade variabeln β inte är signifikant mindre än ett, förkastas inte nollhypotesen för icke-stationär och tidsserien måste differentieras innan den kan användas [7].

Orsaken till att man måste förstå skillnaden mellan en stationär och icke-stationär tidsserie är att de två tidsserier måste vara icke-stationära och av samma grad för att de ska kunna vara kointegrerade. Tanken om kointegration kan förklaras utifrån formeln nedan:

$$z_t = x_t - \beta y_t$$

Idén bakom formeln är att efter att man har beräknat differensen mellan variablerna x_t och y_t så ska z_t bli stationär. Detta gäller dock bara om x_t och y_t är integrerade av samma grad, eftersom z_t blir integrerad av graden I(d-b). Därför måste d och b vara lika för att z_t ska bli stationär, I(0). Vektorn β kallas för kointegrationsvektorn och markerar att om differentieringen ska fungera måste skalan göras om för en av variablerna [1].

Den mest använda och populära metoden för att testa efter kointegration introducerades som tidigare beskrivits av Engle-Granger (1983) och baseras på ”kointegrerad regression”.

Engle-Graners kointegrationstest utförs i två steg:

1. Börja med att skatta $x_t = \alpha + \beta_1 y_t + s_t$, där x_t och y_t är icke-stationära tidsserier av samma grad.

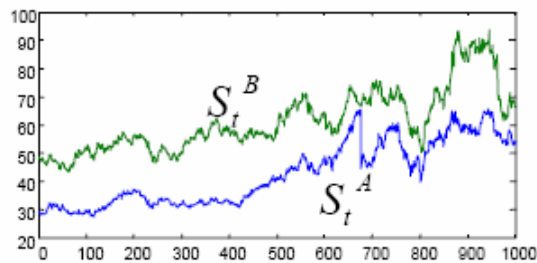
2. Utför därefter ADF-testet på residualen $\{s_t\}$. Om resultatserien saknar enhetsrot är den stationär. Serierna $\{x_t, y_t\}$ är därmed kointegrerade och dess kointegrationsvektor är $(1 - \alpha_1)'$.

Att en mängd variabler är kointegrerade betyder även att de är mean-reverting, dvs. att modellens spread strävar tillbaka till sitt långsiktiga medelvärde. Däremot gäller inte det omvända, dvs. mean-reverting betyder inte nödvändigtvis att variablerna är kointegrerade.

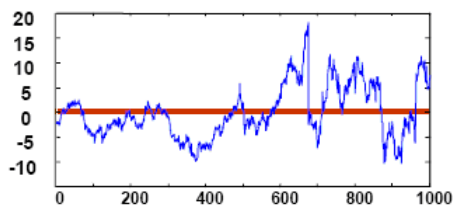
Det är standard att utföra kointegrationsanalyser på logaritmen av priset. Spreaden beräknas då enligt $s_t = \ln(x_t) - \beta_1 \ln(y_t)$. Genom att använda logaritmen för spreaden kompenserar man för effekten av generella ökningar/minskningar av aktiekurserna, dvs. börsens generella uppgång/nedgång. Man kan bortse från α utan att förlora allmängiltighet [8].

2.2 Pairs Trading

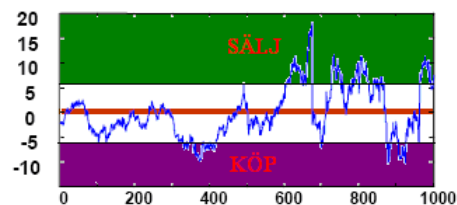
Pairs trading [12] är en marknadsneutral aktiestrategi som baseras på kointegration eller liknande relevanta tekniker. Strategin går ut på att hitta två aktier vars kurser historiskt sett är kointegrerade. Då två tidsserier är kointegrerade gäller enligt ovan att aktierna har ett långsiktigt samband och spreaden, $s_t = \ln(x_t) - \beta_1 \ln(y_t)$ är stationär och därmed även mean-reverting [9].



Figur 2.2.1 Två stycken kointegrerade aktier



Figur 2.2.2 Aktieparets mean-reverting spread.



Figur 2.2.3 Spreadens köp och sälj nivåer

Då man handlar med pairs trading utnyttja man egenskapen att spreaden är mean-reverting (se figur 2.2), dvs man utgår ifrån att även om spreaden divergerar från sitt medelvärde kommer den inom kort att komma tillbaka till medelvärdet igen. När spreaden divergerar från medelvärdet med ett sedan förut bestämt tröskelvärde ingår man korta/långa-positioner på den underliggande portföljen, $\pi_t = x_t - \beta_1 y_t$ (se figur 2.3). När sedan spreaden återvänder till medelvärdet går

man ur positionen. Man ingår alltså en långposition på portföljen, $\pi_t = x_t - \beta_1 y_t$, då spreadens värde hamnar under det lägsta tröskelvärdet, dvs. man köper 1 stycken x_t -aktier och blankar β_1 stycken y_t -aktier. För att ingå en kortposition gäller istället att spreadens värde ska ligga över det översta tröskelvärdet, dvs. man säljer/blankar 1 stycken x_t -aktier och köper β_1 stycken y_t -aktier.

Genom att på samma gång köpa den ena aktien och sälja den andra kan man tjäna pengar på båda positionerna, samtidigt som man reducerar risken för stora marknadsrörelser. Om båda aktierna skulle falla kommer man trots fallet att tjäna pengar på den ena aktien. Pairs trading strategin leder således till att investeraren kan göra vinst oberoende av om marknaden går upp eller ner, en marknadsneutral aktiestrategi.

Den största risken med pairs trading är att båda positionerna går åt fel håll, det vill säga den aktie du vill ska gå upp går istället ner och vice versa. Detta innebär att du förlorar pengar på båda positionerna. Risken kan begränsas genom att du använder stop loss order och automatiskt stänger en eller båda positionerna om det går åt fel håll.

2.3 Standard AdaBoosten

Boosting är en generell metod för att förbättra noggrannheten hos inlärningsalgoritmer. Metoden bygger på tidigare teorier som PAC, Probably Approximately Correct, som skapades av L G Valiant 1984 [14]. PAC beskriver en domänoberoende inlärningsmetod och dess statistiska egenskaper. Valiant var den förste att ställa frågan om en svag inlärningsalgoritm som i PAC ger ett resultat som är endast lite bättre än en slumpmässig gissning kan "boostas" till en stark inlärningsalgoritm med mer exakt resultat. Valiant kom fram till att man med hjälp av att boosta en svag inlärningsalgoritm kunde minska felet i hypotesen till godtyckliga nivåer.

Scapire utvecklade den första bevisbart fungerande boosting algoritmen. Det var också han och Freund som 1995 introducerade den första AdaBoosting algoritmen [15]. Den grundläggande idén bakom AdaBoosting och andra inlärningsalgoritmer är att de kombinerar relativt enkla bashypoteser/modeller, så kallade svaga klassificerare till en slutgiltig förutsägelse, stark klassificerare. Boosting bidrar till att nya modeller bli bra på områden där redan använda modeller inte är så effektiva. Basmodellerna viktas efter sitt resultat och en viktad linjärkombination av dessa bildar sedan den starka klassificeraren, själva förutsägelsen.

AdaBoost algoritmen är snabb, enkel och lätt att programmera och till skillnad från andra boosting algoritmer behöver den ingen tidigare kännedom om prestanda hos den svaga hypotesen och kan därför flexibelt kombineras med andra metoder för att hitta svaga hypoteser. Utmärkande för AdaBoost algoritmen är att precisionen hos den slutgiltiga klassificeraren ökar då precisionen hos någon av de underliggande klassificerarna ökar. I klassificeringssammanhang definieras precision som antalet rättklassificeringar/(antalet rättklassificeringar+antalet felklassificeringar) Hos andra boosting algoritmer beror precisionen för den slutgiltiga klassificeraren bara av den underliggande klassificerare vars utförande

är sämst. "Ada" står för adaptiv dvs. anpassningsbar. Med anpassningsbar menas just att AdaBoosten anpassar sig efter felgraden hos de individuella svaga klassificerarna.

AdaBoost algoritmen består av följande steg [14]:

Givet: $((x_1, y_1), \dots, (x_n, y_n))$ där varje $x_i \in X, y_i \in Y = \{-1, +1\}$

Initialisera $D_1(i) = 1/m, i = 1, \dots, m$ (sannolikhetsfördelning)

För $t = 1, \dots, T$

1. Träna de svaga klassificerarna med hjälp av fördelningen D_t
2. Hitta den svaga klassificeraren $h_t : X \rightarrow \{-1, +1\}$ som minimerar felet ε_t , där $\varepsilon_t = D_t(t)[h_t(x_i) \neq y_i]$
3. Förutsättning $\varepsilon_t \leq 0,5$, annars error
4. Välj $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$

5. Uppdatera:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Z är en normaliseringsfaktor (Vald så att D_{t+1} är en fördelning)

Output: Den slutgiltiga hypotesen, den starka klassificeraren

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

I de svaga kvalificerarna används viktade träningsset. Fördelningens vikter på träningssexempel i under iteration t betecknas $D_t(i)$. Från början (då $t = 1$) sätts vikterna lika, $D_1(i) = 1/m$, men efter varje iteration kommer vikterna att ändras så att vikter som hör till punkter som felklassificerats ges en högre vikt under nästa iteration. För de rätt klassificerade punkterna gäller motsatsen, dvs. minskad vikt vid nästa iteration. Detta leder till att de svaga klassificerarna vid nästa iteration kommer att fokusera på de punkter som felklassificerades i tidigare iterationer.

Vid varje iteration $t = 1, \dots, T$ plockar AdaBoosten ut en mängd D_t slumpmässigt från träningsmängden och applicerar de svaga klassificerarna på denna mängd. Algoritmen väljer den klassificerare $h_t : X \rightarrow \{-1, +1\}$ som minimerar felet, ε_t . Då denna hittats beräknas α_t -värdet och vikterna uppdateras inför nästa iterering.

Efter T iterationer kombinerar AdaBoosten de svaga klassificerarna till en stark klassificerare. Hur stor del av den slutgiltiga klassificeraren som varje svag klassificerare utgör bestäms av α_i -värdet vilket beräknats i steg 3.

Det finns många olika tillämpningar och utvecklingar av AdaBoosten. I mitt examensarbete har jag använt en generaliserad AdaBoost som är mer anpassad till ett finansiellt arbetsområde [9]. Den generaliserade AdaBoosten beskrivs under kapitel 3.

2.4 Simulated Annealing

Simulated Annealing [11] är en teknik som används vid optimering av funktioner över stora mängder, särskilt då en global extrempunkt är gömd bakom flera, svaga, lokala extrempunkter. Den ursprungliga idén bakom simulated annealing är termodynamik, framför allt hur flytande vätskor fryser och kristalliseras. Då en vätskas temperatur är tillräckligt hög kan molekylerna röra sig helt fritt, de kan gå från höga energinivåer till låga och tvärtom. Om temperaturen sänks förlorar de sin rörlighet och vid tillräckligt låg temperatur bildar de kristaller. Kristaller har systemets lägsta energinivå, dvs. ett globalt minimum. Det är precis den här processen vi vill efterlikna med Simulated Annealing för att hitta en global minimipunkt.

Simulated annealings största fördel gentemot andra metoder är dess förmåga att undvika att bli fast i ett lokalt minimum. För att lyckas med detta krävs att man kyler ett system tillräckligt långsamt ty annars kommer det att bildas ett metastabilt tillstånd i strukturen som gör att vi fastnar i lokala minimipunkter och aldrig hittar fram till den önskvärda globala minimipunkten .

För att simulera jämviktsläget genererar man många olika tillstånd, genom att från ett givet tillstånd slumpa fram små förändringar kring detta. Om det nya framslumpade tillståndet har lägre energi kommer det nya tillståndet att accepteras och vi fortsätter vår sökning från den nya punkten. Om däremot en punkt har högre energi ges sannolikheten att den ska accepteras med $\exp(-\frac{\nabla E}{T})$. Där E står för energinivåer, $\nabla E = E(n+1) - E(n)$ och T är en kontrollparameter som representerar villigheten att acceptera ett sämre val (Formeln har sitt ursprung i Boltzmanns sannolikhets fördelning, $\exp(\frac{-E}{kT})$, som är naturens egen minimeringsformel enligt termodynamiken). T minskas med tiden. Ett tillstånd med högre energi kan alltså accepteras med en viss sannolikhet. Följs detta konsekvent kommer vi tillslut att hamna i ett jämviktsläge .

För att Simulated annealing ska lyckas hitta detta jämviktsläge krävs att starttemperaturen T är tillräckligt hög för att partiklarna ska kunna hoppa omkring på ett slumpmässigt sätt.

3 Metod

I detta kapitel presenteras den metod som har använts i uppsatsen. Först diskuteras hur datainsamlingen har gått tillväga. Därefter beskrivs det hur handelsstrategin är utförd. Tillslut presenteras hur resultatet har beräknats.

3.1 Data

Detta examensarbete behandlar två olika datamängder. Datamängd 1 som består av två stycken aktieindex, S&P mid-cap (MID) och S&P 500 index (SPX) från 1991-12-06 till 2008-06-06 samt datamängd 2 som består av 10 stycken aktieindex, MSCI Österrike, MSCI Belgien, MSCI Finland, MSCI Frankrike, MSCI Tyskland, MSCI Grekland, MSCI Italien, MSCI Nederländerna, MSCI Portugal och MSCI Spanien från 1990-04-30 till 2008-08-20. Den data som behövts för den empiriska delen av examensarbetet, är insamlad från Bloomberg.

3.2 Beräkning

Alla beräkningar i examensarbetet är utförda i MATLAB.

Följande tidsperioder används för att beräkna utfallet av handelsmodellen.

Datamängd 1		
1. Träning	2. Test	3. Out-Of-Sample
06 Dec 1991 - 30 Nov 2001	03 Dec 2001 - 30 Jun 2006	03 Jul 2006 - 06 Jun 2008

Datamängd 2	
1.Träning	2.Out-Of-Sample
30 Apr 1990 - 24 Nov 2000	27 Nov 2000 - 20 Aug 2008

Tabell 3.2.1 I rapporten undersökta datamängder

Till att börja med kommer beräkningar med handelsmodellen att ske på datamängd 1. Här tränas och testas modellen för att hitta den bästa lösningen på historiska data. Detta görs genom att söka efter de inställningar av modellen som ger bäst resultat på datamängd 1:s test- och träningsperiod. Då detta resultat har erhållits används den framkomna modellen för fortsatt utvärdering och tester på out-of-sample data. Out-of-sample är den data som inte finns tillgänglig vid det tillfället då strategin skapas och används som referens på hur väl strategin presterar.

För att undersöka modellens out-of-sample resultat testas modellen på datamängd 1:s out-of-sample period. All out-of-sample data blev tillgängligt först i slutet av examensarbetet och har därför inte kunnat påverka tidigare utvecklingar av modellen. Inga justeringar av modellen är tillåtna under out-of-sample beräkningarna.

Modellen testats också på en annan datamängd, datamängd 2. Även detta data blev tillgängligt först i slutet på examensarbetet och har inte påverkat inställningarna av modellen. Först tränas modellen på datamängd 2:s träningsperiod, vars längd väljs så att den är ungefär lika lång som datamängd 1:s träningsperiod. Sen testas modellen på datamängd 2:s out-of-sample data.

3.3 Handelsstrategi

På lång sikt är tidsserierna som används i examensarbetet inte tillräckligt stabila för att enbart kunna använda kointegrations-baserade tekniker som t.ex. pairs trading och liknade verktyg för att ta fram en stabil portföljstrategi. Här måste man utnyttja att spreaden har andra egenskaper utöver de enkla mean-reverting egenskapen som används i kointegrations-baserade tekniker. För att ta fram stabila portföljstrategier baserade på spreads från svagt kointegrerade instrument bör man även utnyttja karakteristiska drag som t.ex. trender av olika längder. Man vill därför använda en mer allmän spreadhandelsstrategi som kan handla med flera olika typer av komplexa spreadtidsserier. Ju mer komplex spreaddynamik som man använder desto lättare är det att garantera att flera olika regimer täcks i träningsfasen. Det här kan signifikant öka stabiliteten hos strategins out-of-sample jämfört med en traditionell pairs trading strategi.

3.3.1 Generaliserad AdaBoost

Med hjälp av kointegration skapas i AdaBoosten en syntetisk tillgång $s_t = \ln(S_{0,t}) - \beta_1 \ln(S_{1,t})$. Formel kan också skrivas som $s_t^* = S_{0,t} * \prod_{j=1}^n S_{j,t}^{-\beta_j}$. Då $\beta_j = 0$ fås original målserien S_0 . Denna syntetiska tillgången tillhandahåller en optimal statistisk hedge för måltillgången, S_0 . (Enligt ovanstående definition på Pairs trading beskrivs hur man handlar med hedgen.) Då β_1 varierar kan man komma fram till massor med olika spreaddynamiker, s_t . Dessa kallas för svaga basmodeller och AdaBoosten har till uppgift att kombinera ihop dessa till en stark modell.

Då vi sysslar med handlingsstrategier och portföljoptimering är en direkt användning av standard Adaboosten inte möjligt eftersom vi inte vill lösa ett klassificeringsproblem utan ett optimeringsproblem. För att man ska kunna använda AdaBoost algoritmen på vårt optimeringsproblem krävs därför en generalisering av AdaBoost algoritmen [9] som kan hantera ett två-klassificeringsproblem, där klassificeraren returnerar antingen 1 eller -1. Den positiva ettan betecknar att ett intervall med längden τ har en avkastning, r , som är större än det förutbestämt tröskelvärde r_c (dvs. $r \geq r_c$). Den negativa ettan står för $r \leq r_c$. Genom att beräkna avkastningen på intervall av längden τ skiftade med längden $\nabla \tau$ och beteckna varje intervall med -1 eller 1 enligt ovan fås en symbolisk avkodad tidsserie av strategisk avkastning. Här vill vi alltså inte

klassificera korrekt mellan -1 och 1 utan maximera antalet 1:or. Det är vårt optimeringsproblem.

Följande steg ingår i den generaliserade AdaBoosten:

$$w_n^{(1)} = 1/N \quad (1.1)$$

$$\varepsilon_t = \sum_{n=1}^N (w_n^{(t)} I(-y_n h_t(x_n))) \quad (1.2)$$

$$\gamma_t = \sum_{n=1}^N (w_n^{(t)} y_n h_t(x_n)) \quad (1.3)$$

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1+\gamma_t}{1-\gamma_t}\right) - \frac{1}{2} \ln\left(\frac{1+\rho}{1-\rho}\right) \quad (1.4)$$

$$w_n^{(t+1)} = w_n^{(t)} \exp(-\alpha_t y_n h_t(x_n)) / Z_t \quad (1.5)$$

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x) / \sum_{t=1}^T \alpha_t \quad (1.6)$$

Parametrar	Förklaring
$h_t(x_n)$	Den bästa bashypotesen vid t:te iterationen
N	Antalet punkter i träningsdata
f(x)	Är den slutgiltiga viktade linjärkombinationen av hypoteserna
ρ	Reglerings parameter
x_n	Mängden av en klassificerades n:tes datapunkt
y_n	Klassetikett (dvs. -1 eller +1).
T	Antalet boosting iterationer
$w_n^{(t)}$	Vikten av den n:te datapunkten vid t:te iterationen
Z_t	Viktad normaliserings konstant vid t:te iterationen

AdaBoosten startar med lika och normaliserade vikter för alla träningsdata (steg 1). Steg 2-5 genomförs vid varje iteration, till stopp kriteriet $\gamma_t < \rho$ (dvs.

$\varepsilon_t \geq \frac{1}{2}(1-\rho)$) eller $\gamma_t = 1$ (dvs. $\varepsilon_t = 0$). Här anropar AdaBoosten en pool med

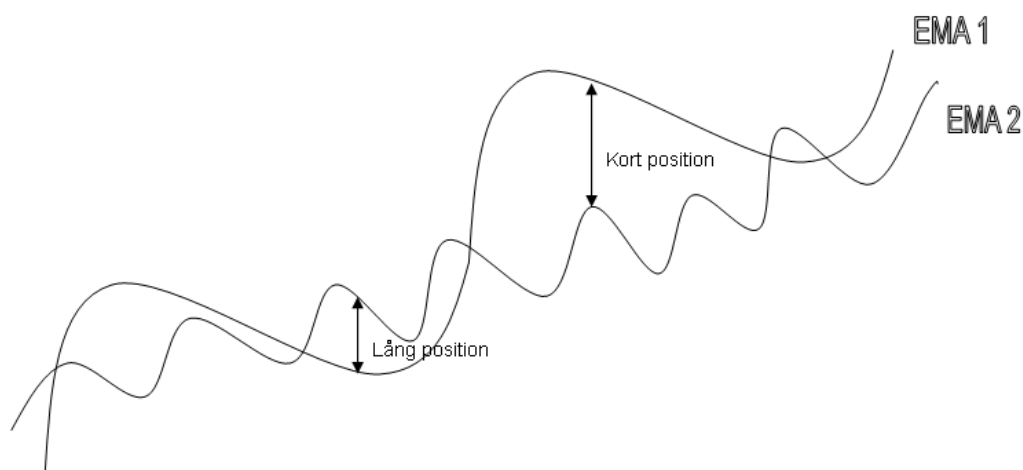
svaga basstrategier och strategin med flest 1:or väljs, dvs den strategi som har minst fel enligt punkt 1.2. I steg 3-5 uppdateras vikterna som ska användas i nästa iteration. I steg 5 straffas de datapunkter som felklassificerats av den då valde svaga modellen, dvs. deras vikter kommer att minska till nästa iteration. Vikterna till de punkter som rättklassificeras kommer däremot att öka till nästa iteration. Steg 6 sätter ihop den slutgiltiga viktade linjärkombinationen av bashypoteserna.

3.4 Handelssignaler

3.4.1 EMA Mean Reverse

För att bestämma om man ska gå in i en lång eller kort position beräknar man spreadens EMA(n,a), exponentiellt glidande medelvärde. EMA(n,a) är en metod som använder n historiska data och viktar dessa exponentiellt med en avklingning a , dvs. det senaste dagsdatat får stor vikt medan ju längre bort man kommer från den valda dagen desto mindre vikt får dessa dagar.

EMA Mean Reverse går ut på att man för varje dag beräknar två olika EMA, EMA1 och EMA2. EMA1 beräknas på ett större n -värde, dvs. den beräknas på fler historiska data än vad EMA2 gör. På så sätt får man två olika EMA-kurvor. Eftersom EMA1 beräknas på en längre datamängd kommer denna kurva att vara lugnare och svänga mindre än EMA2.



Figur 3.4.1

För att handla på dessa EMA-kurvor testas vilken EMA som för dagen är störst. Om $EMA2 > EMA1$ anses trenden vara uppåtgående, och vi kommer att ta en lång position. Om däremot $EMA2 < EMA1$ anses trenden vara neråtgående och vi kommer att ta en kort position.

Handelsstrategin bestäms av följande variabler, (n, a, m, α, β) , vilka styr handelssignalerna. I examensarbetet slumpas variabelernas startvärden fram vid varje iteration i AdaBoosten. Värdena som parametrarna kan hamna emellan är som följer:

Parametrar	Lägsta värdet	Högsta värdet	Förklaring
N	2	130	Antalet historiska data för EMA1
A	0.96	0.99	Storleken på avklingningen
M	2	130	Antalet historiska data för EMA2
A	0	2	Tröskelmått för volatiliteten
β	- 1	1	Parameter i spreaden

Handeln med korta eller långa positioner sker på nästkommande dag.

3.5 Stokastisk sökalgoritm

I examensarbetet gjordes ett försök med att använda Simulated Annealing för att ta fram den optimala handelsstrategin. Tanken var att denna metod enkelt skulle kunna hitta den handelsmodell som gav högst avkastning vid gällande vikter. Detta visade sig dock vara mycket svårare än man trodde pga. att det är svårt att ställa in rätt värde på temperaturen T och avklingningsvärde. Om dessa är felinställda finns det en risk att Simulated Annealing inte hittar rätt globala minimipunkt. Resultatet blev att metoden trots att inga parametrar ändrats ändå hittade olika globala minimipunkter vid varje körning. Pga. detta valdes en annan metod, stokastisk sökalgoritm, för att kunna ta fram den globala minimipunkten. Till skillnad från Simulated Annealing accepterar inte denna metod ett tillstånd som är sämre än det valda tillståndet. Det är också skillnad i hur man väljer det första tillståndet som ska testas. I den nya metoden slumpas man fram ett antal punkter i parameterrummet, i detta fall 5000 stycken och väljer sedan ut de 10 bästa punkterna, dvs. de punkterna i parameterrummet med högst avkastning av de 5000 stycken punkterna. Därefter görs en deterministisk sökning i 200 steg runt varje punkt. Om en ny punkt har ett tillstånd som är bättre än den gällande punktens tillstånd accepteras den nya punkten och sökningen fortsätter från denna punkt. Till slut har man fått 10 stycken nya tillstånd och av dessa väljs den punkt med högst avkastning till den globala minimipunkten.

4. Resultat

I början av detta kapitel beskrivs hur avkastningsberäkningarna utförts. Därefter presenteras resultatet av handelsstrategin. Först illustreras in-sample resultatet, senare i kapitel tre presenteras out-of sample resultatet, i fjärde kapitlet illustreras ackumulerad avkastning och sist beskrivs och visas statistiska tester.

4.1 Avkastningsberäkning

Utvärdering och jämförelser av portföljer baseras oftast på avkastningen för respektive portfölj under en viss tidsperiod. En svaghet med detta sätt att bedöma portföljer är att man inte tar hänsyn till vilken risk portföljen har, hur marknaden där portföljen placerat sitt kapital har presterat eller hur duktiga förvaltarna för portföljen är. Ett sätt att göra utvärderingarna mer rättvisa är att man förutom avkastningen också tar hänsyn till portföljens risk och sätter avkastningen i relation till risken.

I denna uppsats har det riskjusterande avkastningsmättet informationskvoten (IR) använts för att jämföra resultat. IR beskriver relationen mellan den aktiva avkastningen och den aktiva avkastningens standardavvikelse och beräknas på följande sätt:

$$IR_{\text{årlig}} = \sqrt{250} \frac{\mu}{\sigma}$$

där μ och σ är medelvärdet och standardavvikelsen av den dagliga avkastningen. Genom att multiplicera med $\sqrt{250}$ fås den årliga informationskvoten¹. Om en positiv kvot påvisas har en riskjusterad överavkastning erhållits gentemot index under samma period.

4.2 In-sample resultat från datamängd 1

Här visas resultatet för handelsmodellens in-sample period. Under in-sample perioden justeras variabler och inställningar i handelsmodellen för att uppnå ett tillfredställande resultat.

4.2.1 Resultat från träningsperioden

Under träningsperioden appliceras Adaboosting på träningsdata. Antalet iterationer (T) som utförs i AdaBoosten sattes till T=20, eftersom resultat vid färre iterationer inte blir tillförlitligt och beräkningar med fler iterationer endast ger marginella förbättringar och ökar beräkningstiden. Vid varje iteration väljer en

¹ Det årliga medelvärdet skalas linjärt, 250μ medan standardavvikelsen skalas $\sqrt{250}\sigma$

Stokastisk Sökalgoritm (kap 3.5) en handelsstrategi, (n, a, m, α, β) , som enligt då gällande vikter, w , är den strategi som enligt aktuell handelssignal (tex EMA Mean Reverse) ger den högsta avkastningen. Då alla iterationer utförts erhålls den slutgiltiga basstrategin.

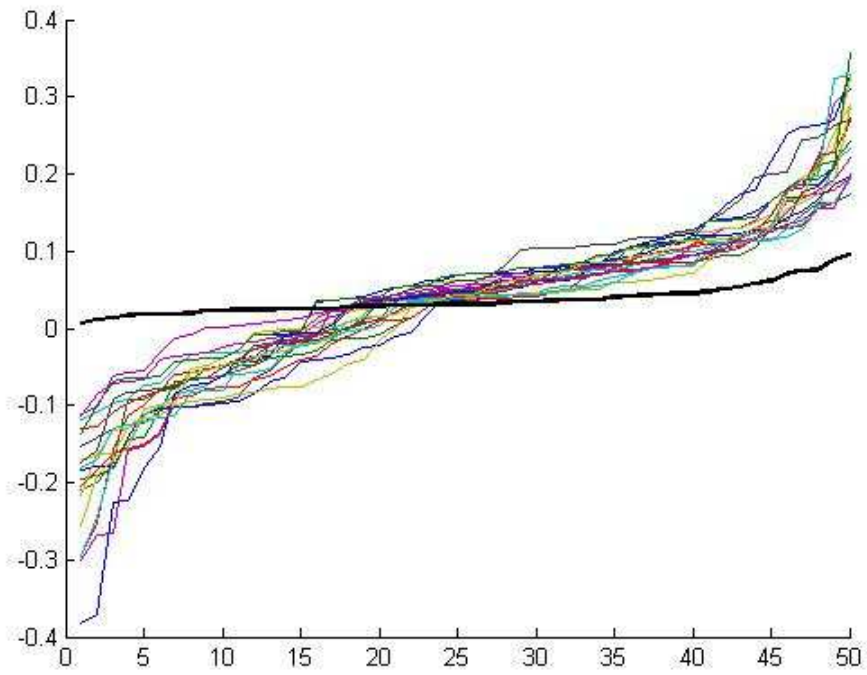
Avkastningen blir olika beroende på vilka värden på τ , $\Delta\tau$ och tröskelvärde r som väljs. Av nedanstående tabell framgår vilka värden på och kombinationer av τ , $\Delta\tau$ och r som har använts vid beräkningarna,

	T	$\Delta\tau$	r
1	5	5	0.02
2	10	10	0.02
3	40	10	0.02
4	60	20	0.02
5	120	40	0.02
6	5	5	0.05
7	10	10	0.05
8	40	10	0.05
9	60	20	0.05
10	120	40	0.05
11	5	5	0.07
12	10	10	0.07
13	40	10	0.07
14	60	20	0.07
15	120	40	0.07

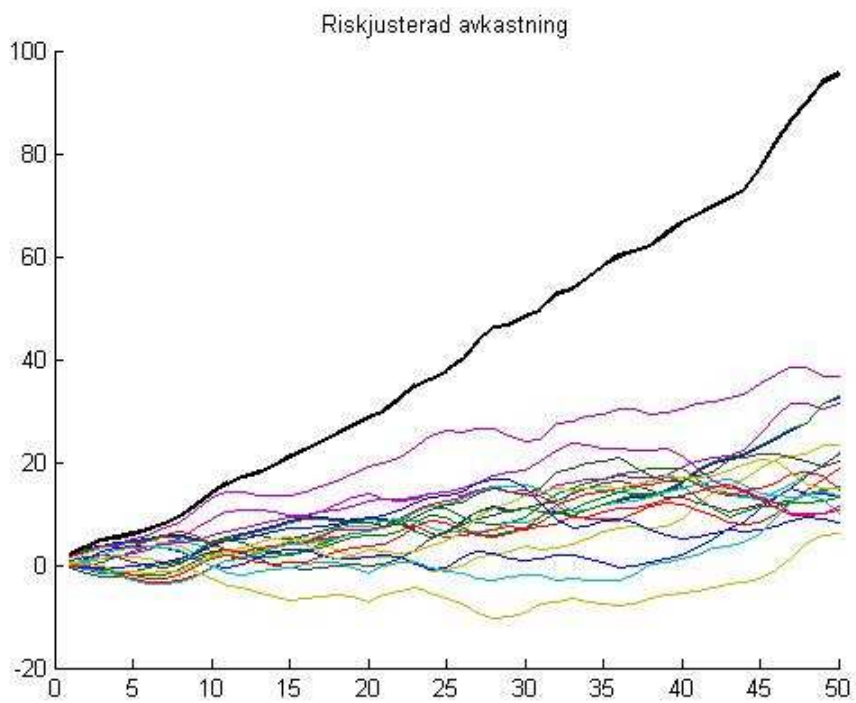
Tabell 4.2.1 Tabell över de variabelkombinationer som testats

4.2.2 Resultat från AdaBoosten

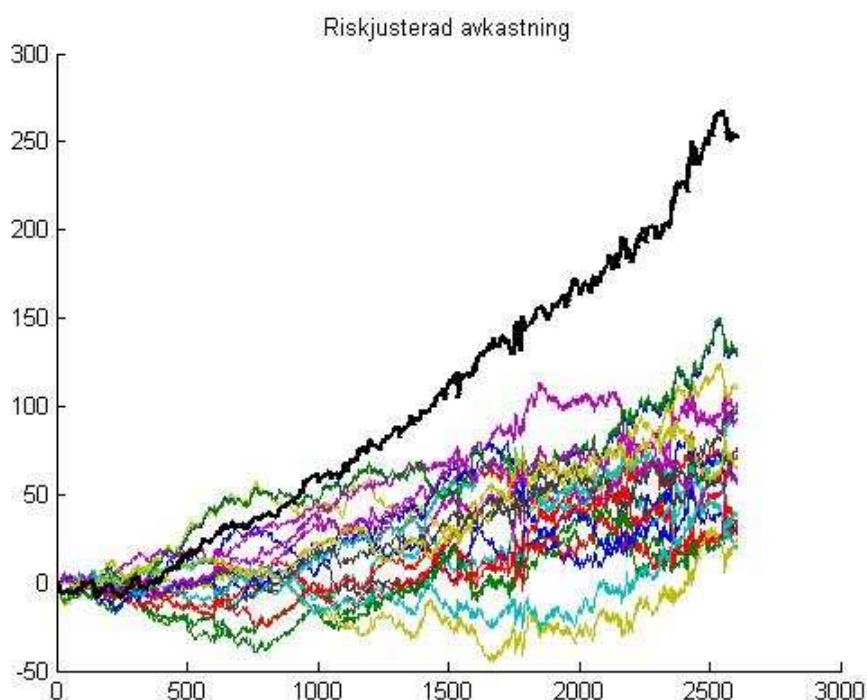
Nedan illustreras resultatet från AdaBoosten då testdata från period 1 har applicerats med $\tau = 120$, $\Delta\tau = 40$ och $r = 0.07$. Här visas fördelningen av den årliga avkastningen (%). Den tjocka linjen är den slutgiltiga boostade MID-SPX multi-spread portföljstrategin och de andra linjerna visar varje svag handelsstrategi. Jag har valt att endast visa figurer från denna variabelkombination, då figurerna från de andra kombinationerna har snarlika utseenden och därför inte tillför någon ny insikt till resultatet.



Figur 4.2.1: Fördelning av den årliga avkastningen (%) för $\tau=120$ dagars perioder för MID-SPX



Figur 4.2.2: Riskjusterad avkastning för $\tau=120$ dagars perioder för MID-SPX



Figur 4.2.3: Riskjusterad avkastning för den dagliga avkastningen för MID-SPX

4.2.3 Resultat från testperioden

Av träningen fås en strategi som sen används för att handla på testdatan. I tabell 4.2 visas IR för alla testade variabelkombinationer för in-sample testdata. Kolumnen längst till höger visar IR för en portfölj bestående av de fem strategier med samma r-värde och längst ner visas IR för den totala portföljen, bestående av alla 15 modellerna

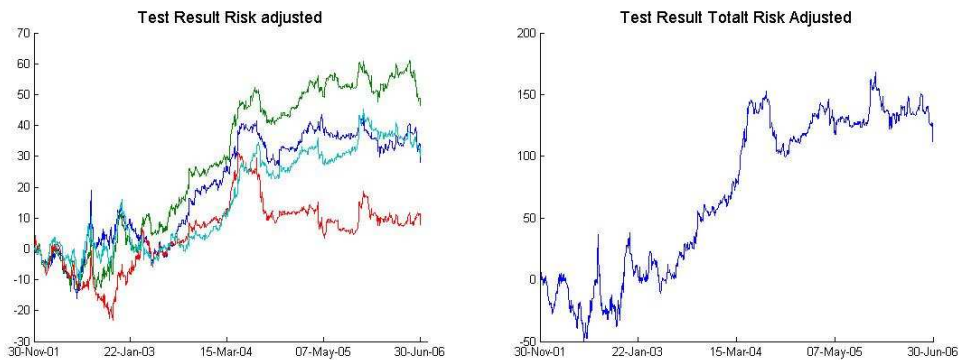
r=0.02	1	2	3	4	5	Portfölj
IR	0,4537	0,6362	0,1261	0,2875	0,7619	0,5785
r=0.05	6	7	8	9	10	Portfölj
IR	0,4768	0,4803	0,4721	0,5862	0,7145	0,5669
r=0.07	11	12	13	14	15	Portfölj
IR	0,2771	0,3005	0,6128	0,6945	0,8453	0,6119

	Totala portfölj (15 stycken strategier)
IR	0.6416

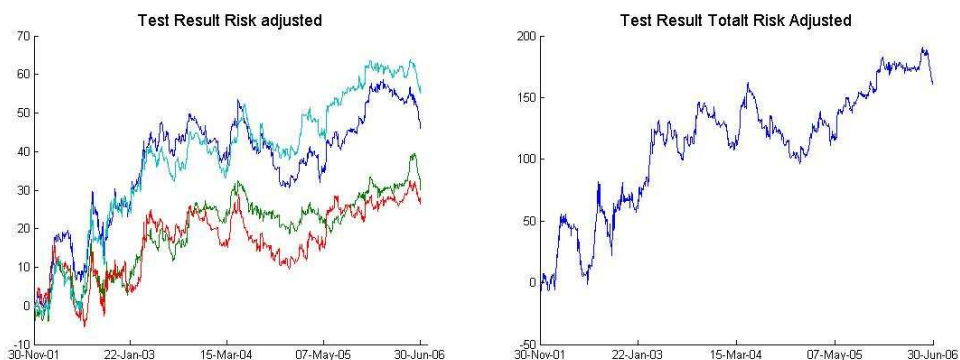
Tabell 4.2.2: IR för modellerna och deras portföljer

Nedan illustreras 30 stycken figurer som visar daglig riskjusterad avkastningen för de olika variabelkombinationerna. Till varje variabelkombination hör två figurer och en bildtext, som anger vilken variabelkombination som visas. Den

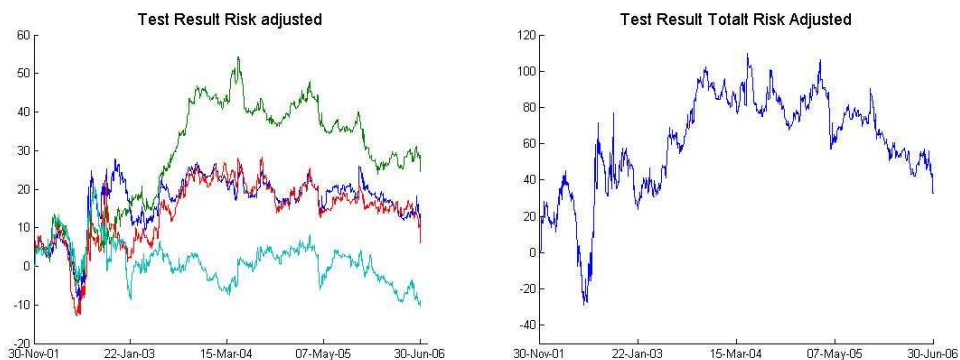
första figuren illustrerar fyra kurvor, där varje kurva representerar riskjusterad avkastningen vid ett beräkningstillfälle för gällande variabelkombinationen. Eftersom varje körning ger olika resultat pga. att det finns en slump i initialvärdena har jag valt att upprepa beräkningen fyra gånger. Totalt visas alltså fyra stycken beräkningar på samma variabelkombination. Därefter visas en figur med endast en kurva, denna kurva representerar summan av de riskjusterade avkastningarna från de fyra kurvorna till vänster.



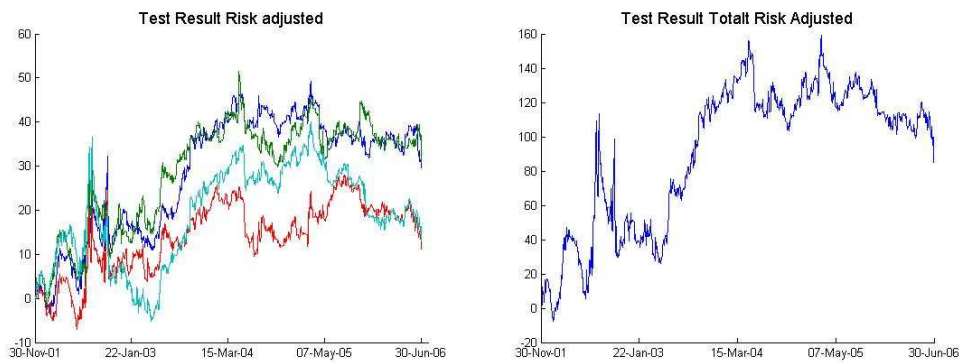
Figur 4.2.4 För $\tau = 5$, $\Delta\tau = 5$ och $r=0.02$



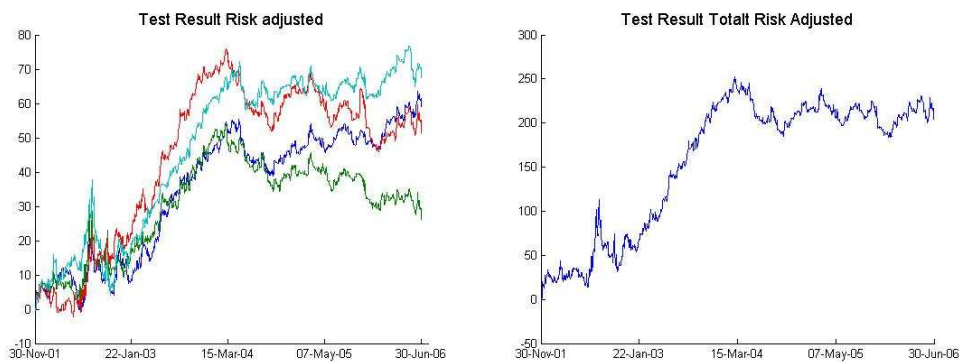
Figur 4.2.5 För $\tau = 10$, $\Delta\tau = 10$ och $r=0.02$



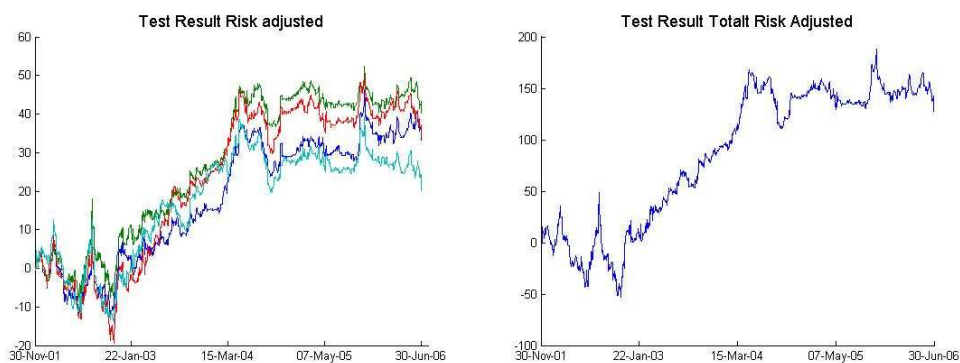
Figur 4.2.6 För $\tau = 40$, $\Delta\tau = 10$ och $r=0.02$



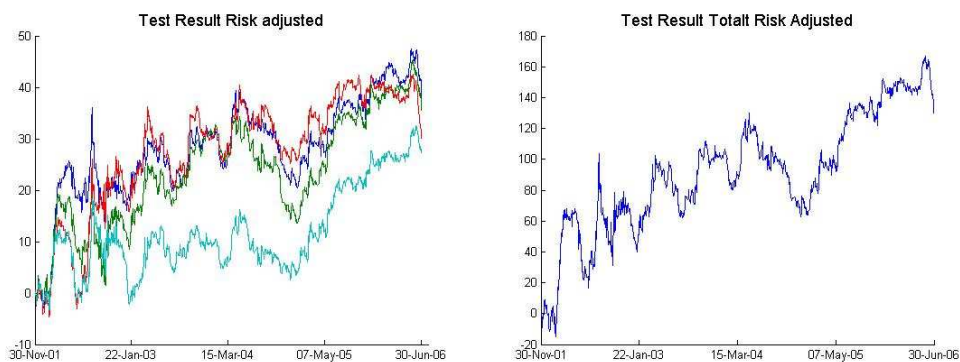
Figur 4.2.7 För $\tau = 60$, $\Delta\tau = 20$ och $r = 0.02$



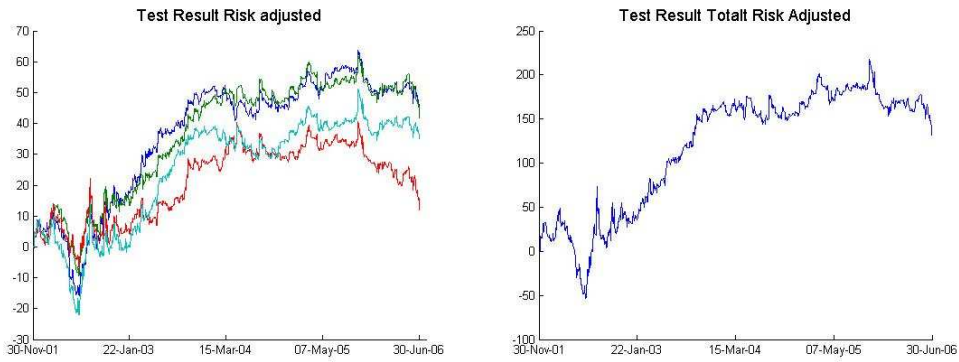
Figur 4.2.8 För $\tau = 120$, $\Delta\tau = 40$ och $r = 0.02$



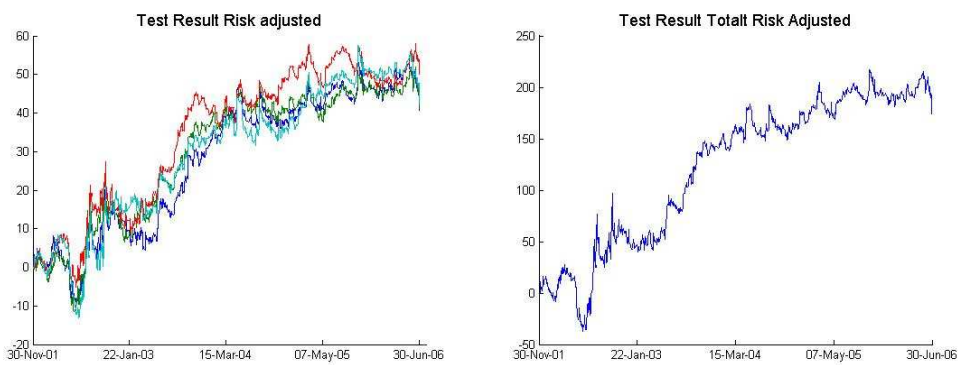
Figur 4.2.9 För $\tau = 5$, $\Delta\tau = 5$ och $r = 0.05$



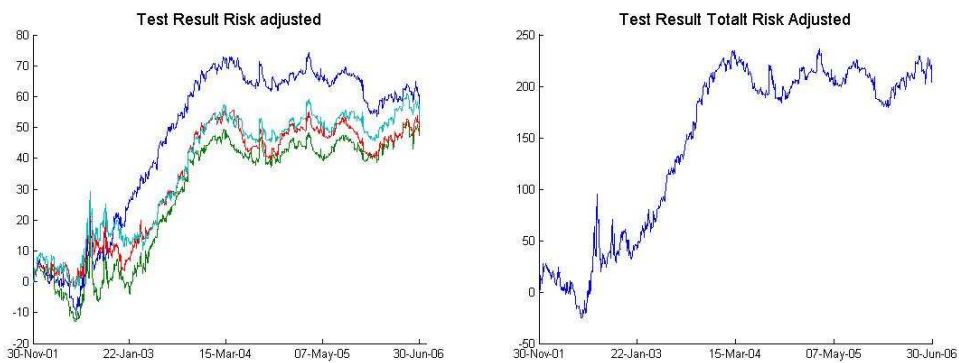
Figur 4.2.10 För $\tau = 10$, $\Delta\tau = 10$ och $r = 0.05$



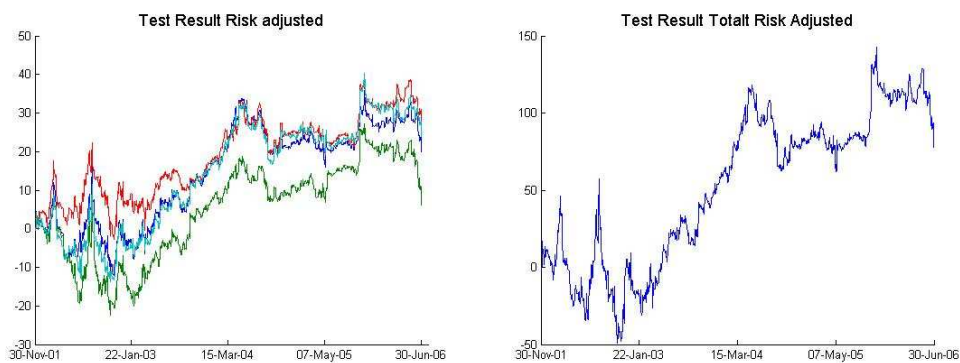
Figur 4.2.11 För $\tau=40$, $\Delta\tau=10$ och $r=0.05$



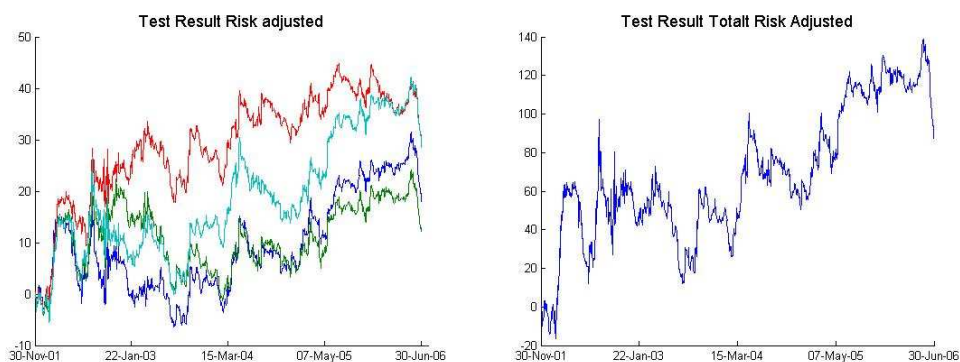
Figur 4.2.12 För $\tau=60$, $\Delta\tau=20$ och $r=0.05$



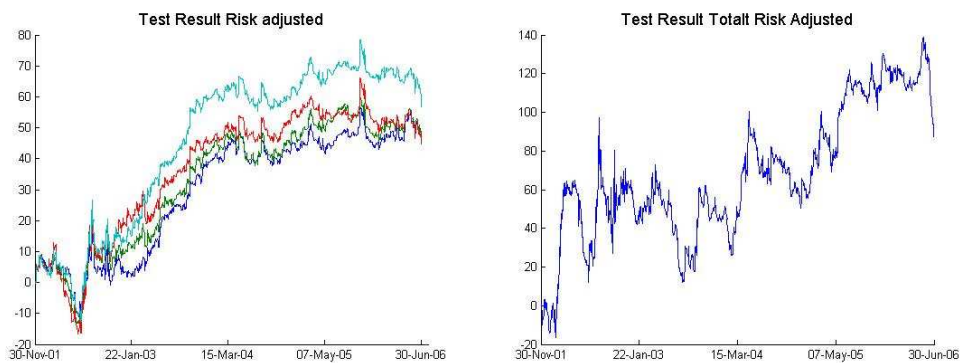
Figur 4.2.13 För $\tau=120$, $\Delta\tau=40$ och $r=0.05$



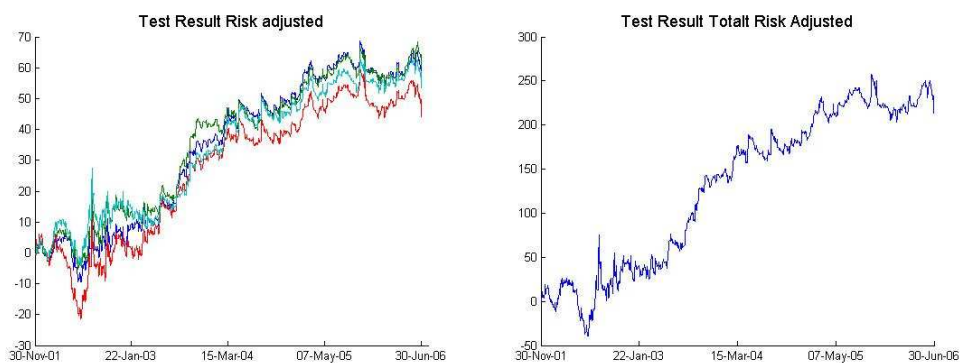
Figur 4.2.14 För $\tau = 5$, $\Delta\tau = 5$ och $r=0.07$



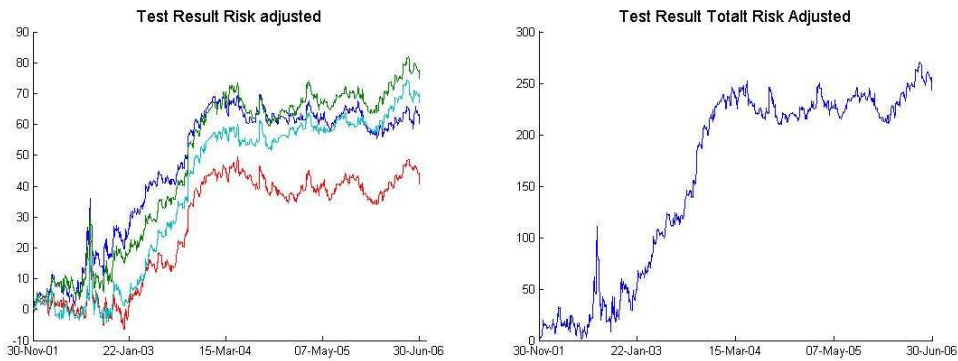
Figur 4.2.15 För $\tau = 10$, $\Delta\tau = 10$ och $r=0.07$



Figur 4.2.16 För $\tau = 40$, $\Delta\tau = 10$ och $r=0.07$

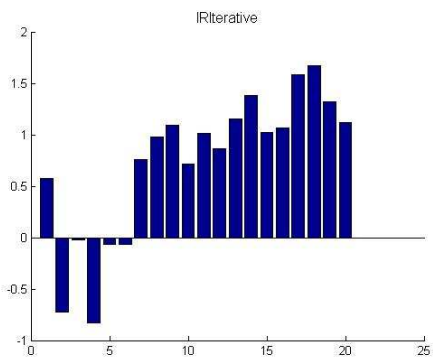


Figur 4.2.17 För $\tau = 60$, $\Delta\tau = 20$ och $r=0.07$

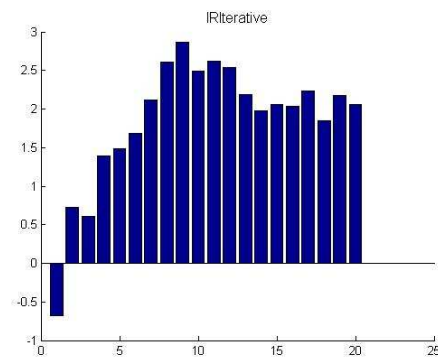


Figur 4.2.18 För $\tau = 120$, $\Delta\tau = 40$ och $r=0.07$

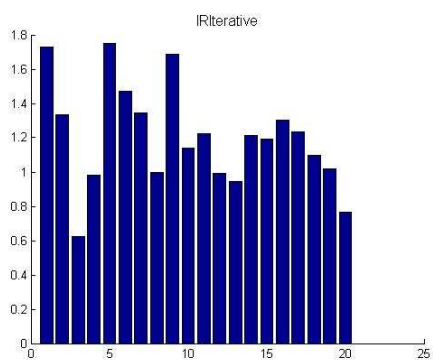
Härefter följer figurer som visar hur IR hos summan av de fyra kurvorna förändras från iteration till iteration i AdaBoosten. Varje stapel symboliserar en iteration och sammanlagt finns alltså 20 staplar.



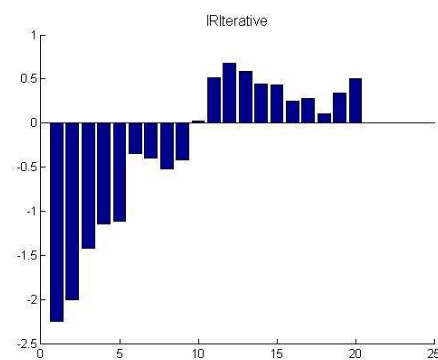
Figur 4.2.19 För $\tau = 5$, $\Delta\tau = 5$ och $r=0.02$



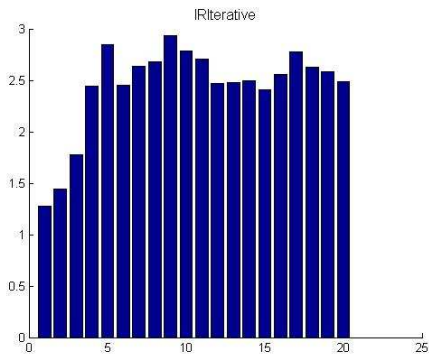
Figur 4.2.20 För $\tau = 10$, $\Delta\tau = 10$ och $r=0.02$



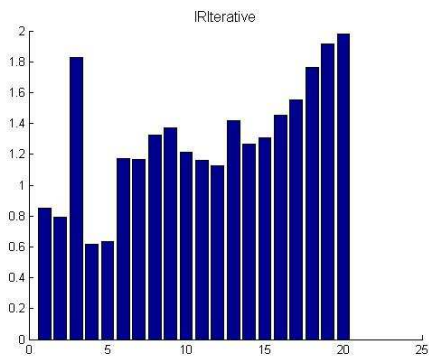
Figur 4.2.21 För $\tau = 40$, $\Delta\tau = 10$ och $r=0.02$



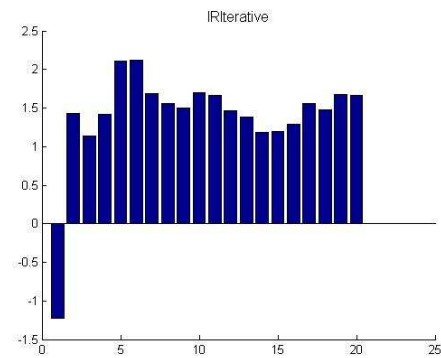
Figur 4.2.22 För $\tau = 60$, $\Delta\tau = 20$ och $r=0.02$



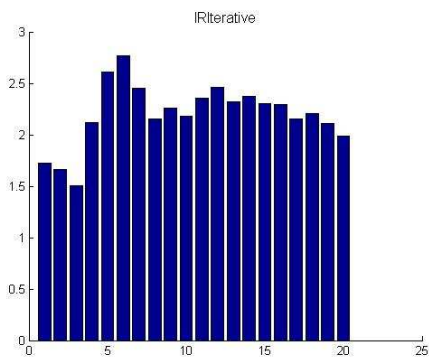
Figur 4.2.23 För $\tau = 102$, $\Delta\tau = 40$ och $r=0.02$



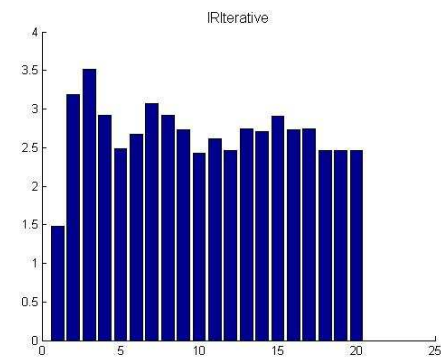
Figur 4.2.24 För $\tau = 5$, $\Delta\tau = 5$ och $r=0.05$



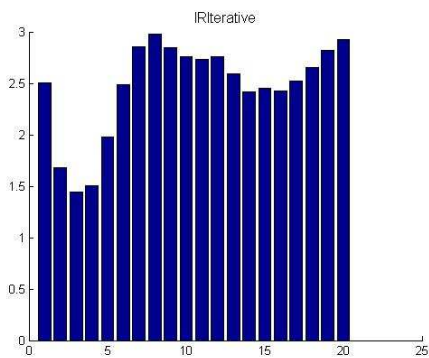
Figur 4.2.25 För $\tau = 10$, $\Delta\tau = 10$ och $r=0.05$



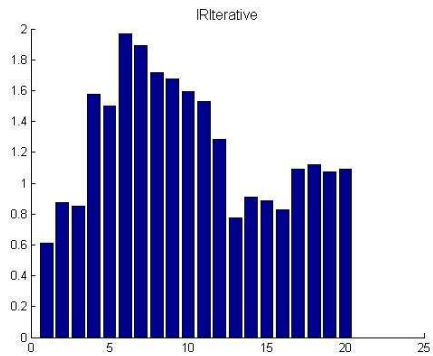
Figur 4.2.26 För $\tau = 40$, $\Delta\tau = 10$ och $r=0.05$



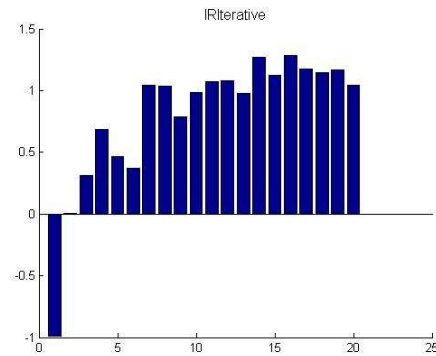
Figur 4.2.27 För $\tau = 60$, $\Delta\tau = 20$ och $r=0.05$



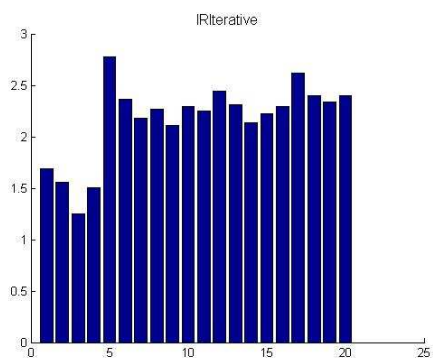
Figur 4.2.28 För $\tau = 102$, $\Delta\tau = 40$ och $r=0.05$



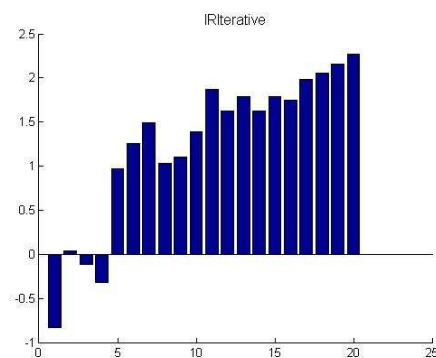
Figur 4.2.29 För $\tau=5$, $\Delta\tau=5$ och $r=0.07$



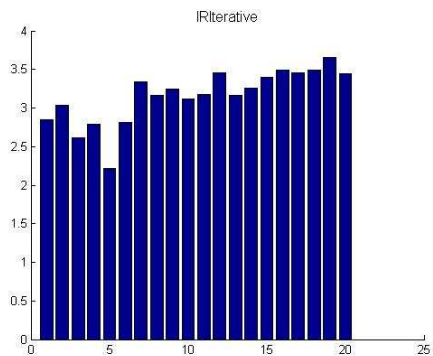
Figur 4.2.30 För $\tau=10$, $\Delta\tau=10$ och $r=0.07$



Figur 4.2.31 För $\tau=40$, $\Delta\tau=10$ och $r=0.07$



Figur 4.2.32 För $\tau=60$, $\Delta\tau=20$ och $r=0.07$



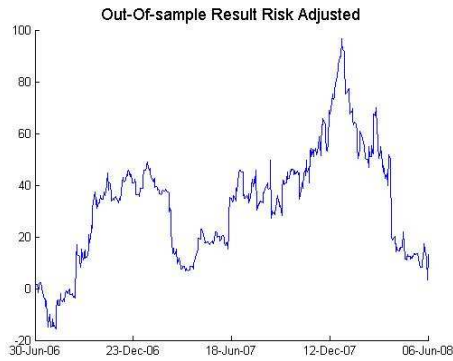
Figur 4.2.33 För $\tau=102$, $\Delta\tau=40$ och $r=0.07$

4.3 Out-of-sample resultat

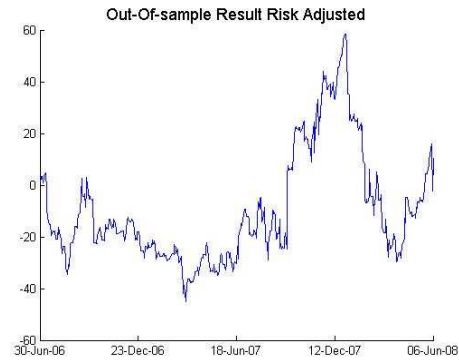
Beräkningar från in-sample perioden lägger grunden till ytterligare tester. Eftersom vår handelsmodell har gett tillfredställande resultat på in-sample perioden vill vi nu undersöka resultatet då vi applicerar våra modell på out-of-sample data. Vi undersöker resultatet på out-of-sample för bägge datamängderna. All out-of-sample data har inte varit tillgängliga förrän i slutet av examensarbetet, och inga justeringar har gjorts för att anpassa modellen till dessa data.

4.3.1 Out-of-sample resultat för datamängd 1

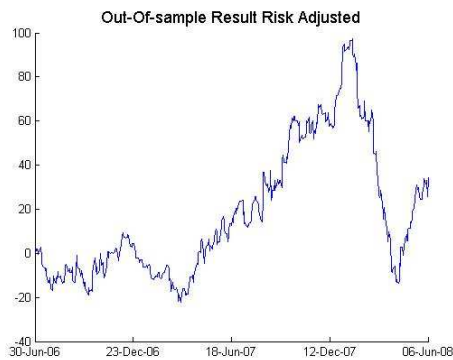
Vi börjar med att i figurerna nedan visa out-of-sample resultatet på datamängd 1. Figurerna visar endast den del av testdatat som är out-of-sample, dvs. inte den del som redan visats i figurer under in-sample kapitlet.



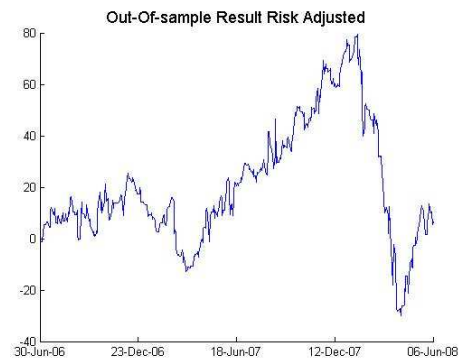
Figur 4.3.1.1 För $\tau = 5$, $\Delta\tau = 5$ och $r=0.02$



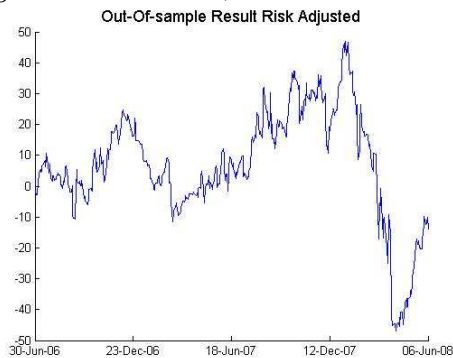
Figur 4.3.1.2 För $\tau = 10$, $\Delta\tau = 10$ och $r=0.02$



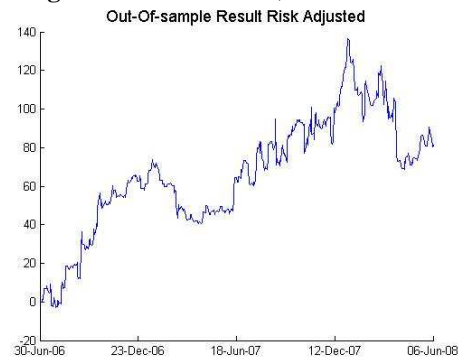
Figur 4.3.1.3 För $\tau = 40$, $\Delta\tau = 10$ och $r=0.02$



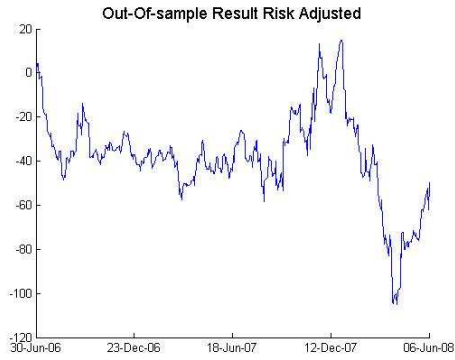
Figur 4.3.1.4 För $\tau = 60$, $\Delta\tau = 20$ och $r=0.02$



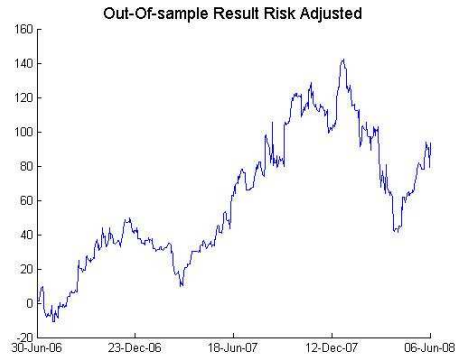
Figur 4.3.1.5 För $\tau = 120$, $\Delta\tau = 40$ och $r=0.02$



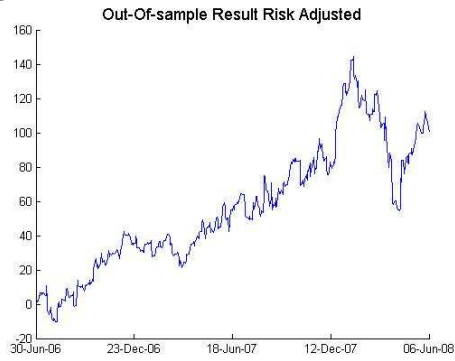
Figur 4.3.1.6 För $\tau = 5$, $\Delta\tau = 5$ och $r=0.05$



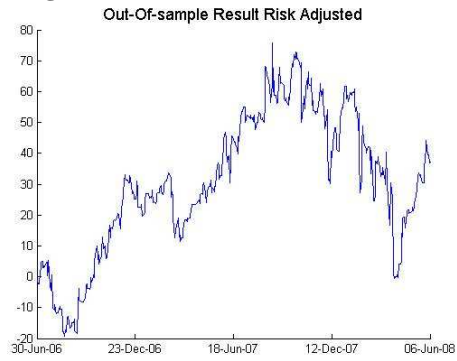
Figur 4.3.1.7 För $\tau = 10$, $\Delta\tau = 10$ och $r = 0.05$



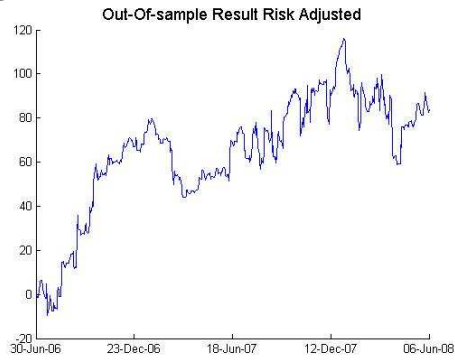
Figur 4.3.1.8 För $\tau = 40$, $\Delta\tau = 10$ och $r = 0.05$



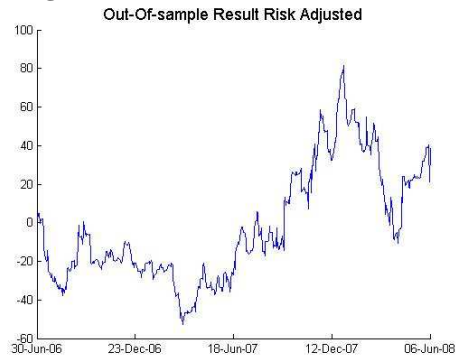
Figur 4.3.1.9 För $\tau = 60$, $\Delta\tau = 20$ och $r = 0.02$



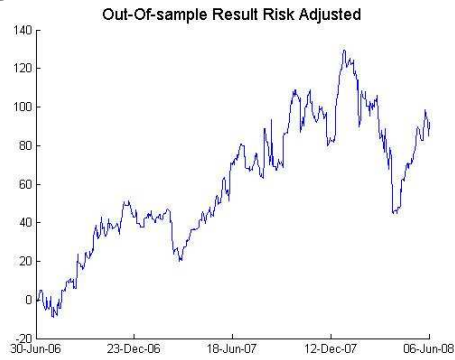
Figur 4.3.1.10 För $\tau = 120$, $\Delta\tau = 40$ och $r = 0.05$



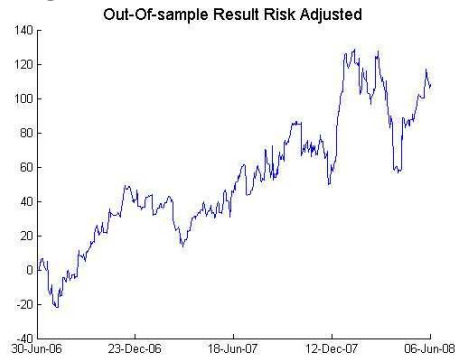
Figur 4.3.1.11 För $\tau = 5$, $\Delta\tau = 5$ och $r = 0.07$



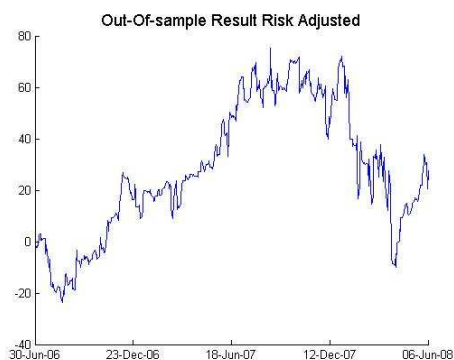
Figur 4.3.1.12 För $\tau = 10$, $\Delta\tau = 10$ och $r = 0.07$



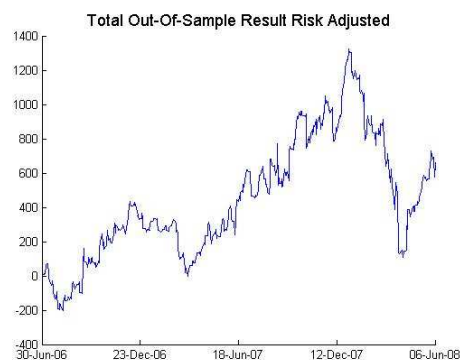
Figur 4.3.1.13 För $\tau = 40$, $\Delta\tau = 10$ och $r = 0.07$



Figur 4.3.1.14 För $\tau = 60$, $\Delta\tau = 20$ och $r = 0.07$



Figur 4.3.1.15 För $\tau = 120$, $\Delta\tau = 40$ och $r=0.07$



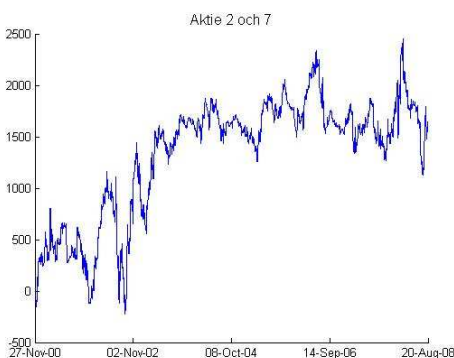
Figur 4.3.1.16 Totala portföljen

$r=0.02$	1	2	3	4	5	Portfölj
IR	0.1031	0.090	0.3599	0.0540	-0.1296	0.1180
$r=0.05$	6	7	8	9	10	Portfölj
IR	0.6702	-0.3701	0.7249	0.8194	0.3157	0.5314
$r=0.07$	11	12	13	14	15	Portfölj
IR	0.6946	0.2811	0.7080	0.8307	0.2284	0.6586
	Totala portfölj (15 stycken strategier)					
IR	0.4651					

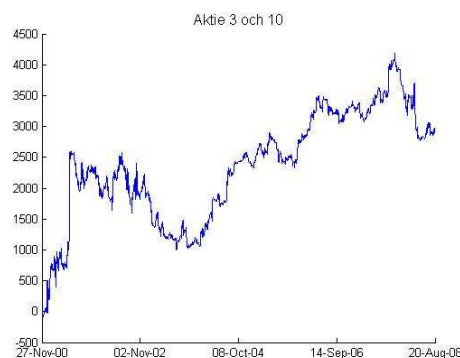
Tabell 4.3.1.1 IR-resultat från perioden 30-JUN-2006 till 6-JUN-2008, på datamängd 1

4.3.2 Out-of-sample resultat för datamängd 2

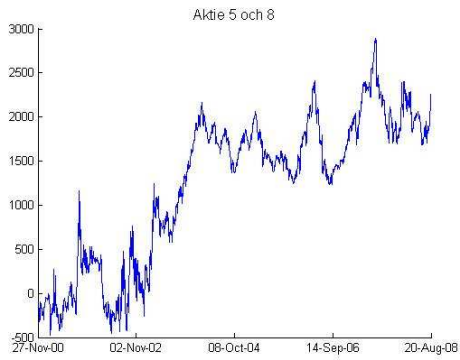
Nästa steg är att testa vår handelsmodell på en annan datamängd, datamängd 2. Nedanstående figurer visar riskjusterad avkastning för dennas testperiod. De tio underliggande tidsserierna har parats ihop slumpmässigt och varje figur illustrerar en slumpmässig hopparring . Sista figuren visar den totala portföljen för de sammanlagda par-figurerna.



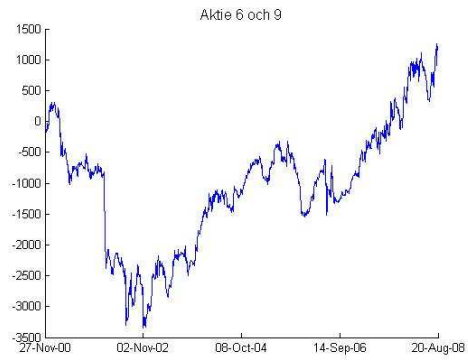
Figur 4.3.2.1 IR= 0.2514



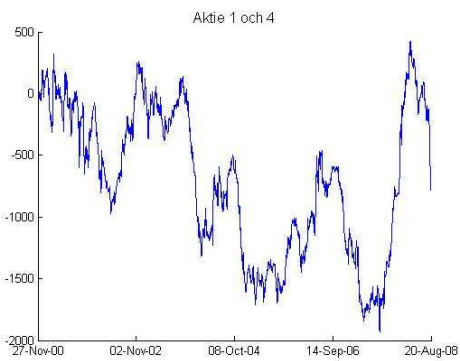
Figur 4.3.2.2 IR= 0.3974



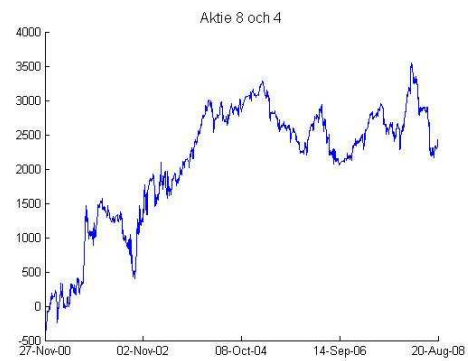
Figur 4.3.2.3 IR= 0.2912



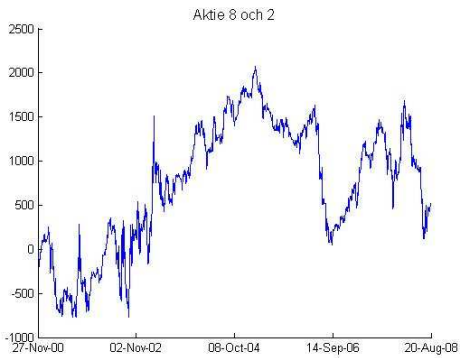
Figur 4.3.2.4 IR= 0.1708



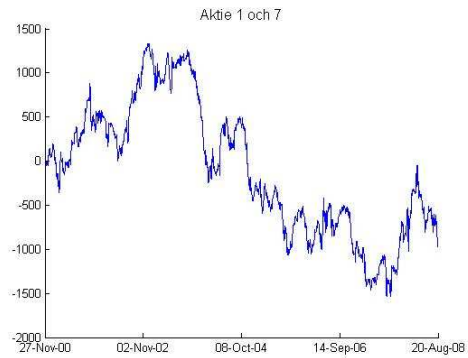
Figur 4.3.2.5 IR= -0.1200



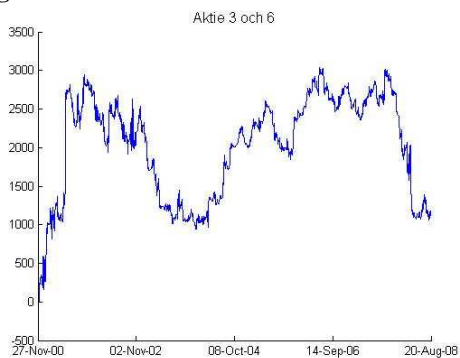
Figur 4.3.2.6 IR= 0.4191



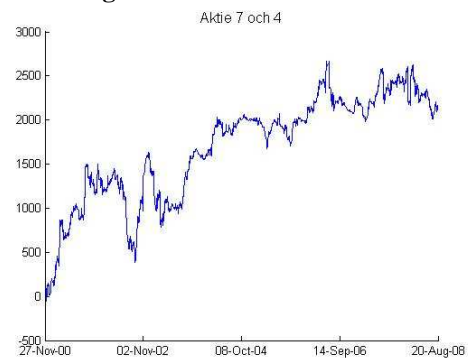
Figur 4.3.2.7 IR= 0.0704



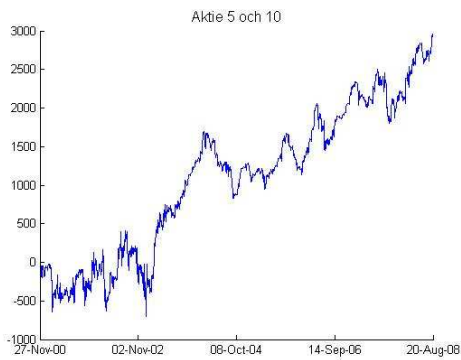
Figur 4.3.2.8 IR= -0.1583



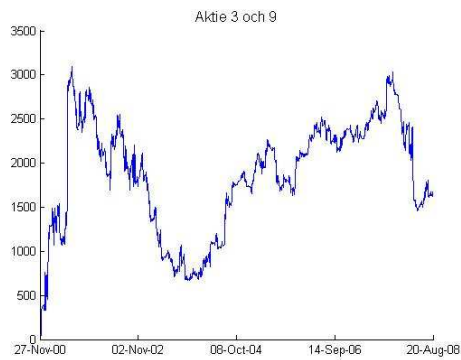
Figur 4.3.2.9 IR= 0.1539



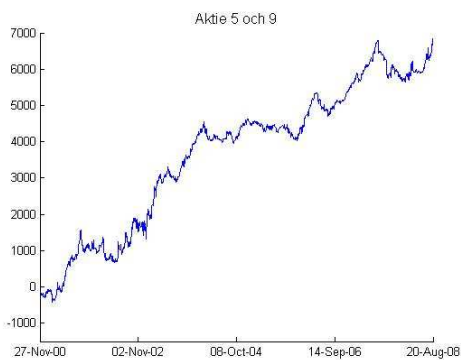
Figur 4.3.2.10 IR= 0.5057



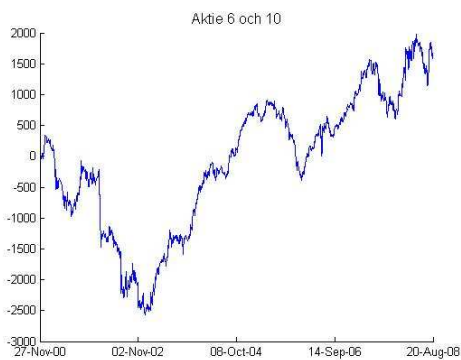
Figur 4.3.2.11 IR= 0.4892



Figur 4.3.2.12 IR= 0.2291



Figur 4.3.2.13 IR=1.1394



Figur 4.3.2.14 IR= 0.2524



Figur 4.3.2.15 IR=0.5841

	Aktie 2 & 7	Aktie 3 & 10	Aktie 5 & 8	Aktie 6 & 9	Aktie 1 & 4	Aktie 8 & 4
IR	0,2514	0,3974	0,2912	0,1708	-0,12	0,4191

	Aktie 8 & 2	Aktie 1 & 7	Aktie 3 & 6	Aktie 7 & 4	Aktie 5 & 10	Aktie 3 & 9
IR	0,0704	-0,1583	0,1539	0,5057	0,4892	0,2291

	Aktie 5 & 9	Aktie 6 & 10	Totala
IR	1,1394	0,2524	0,5841

Tabell 4.3.2.1 IR-resultat för perioden 27-NOV-2000 till 20-AUG-2008, på datamängd 2

4.4 Ackumulerad avkastning

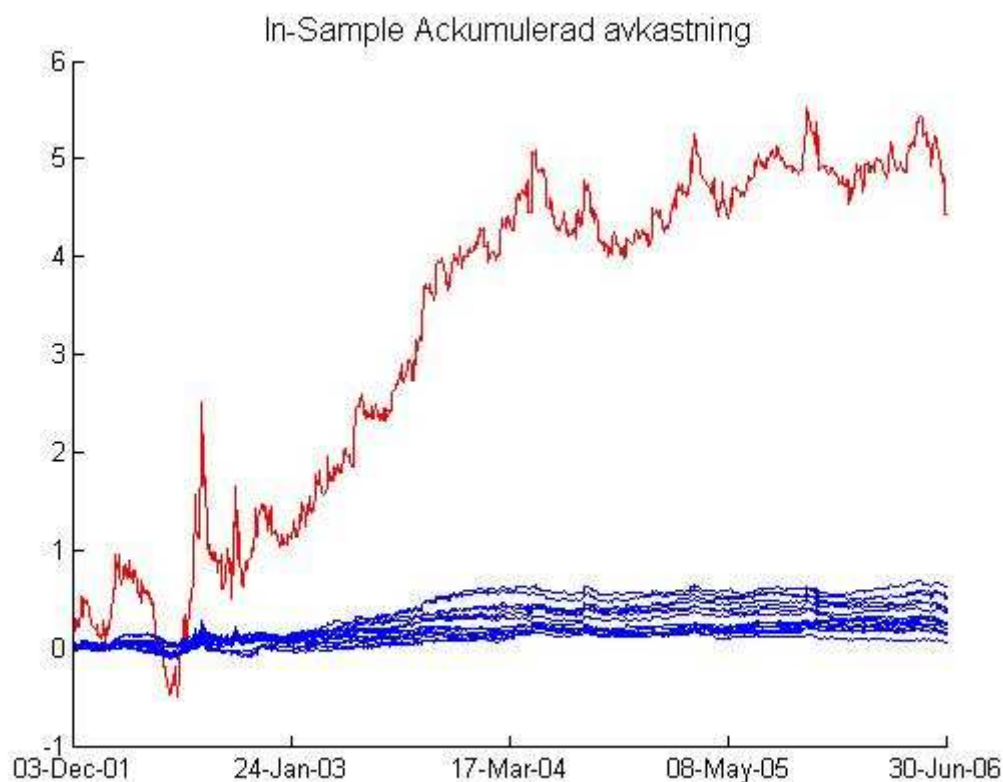
Vi beräknar den ackumulerade avkastningen z_t^i för testmodell i genom att lägga ihop individuell daglig avkastning r_t^i ,

$$z_t^i = \sum_{k=1}^t r_k^i$$

Den totala avkastningen från alla 15 modellerna Z_t beräknas som summan av de ackumulerade avkastningarna z_t^i från varje modell,

$$Z_t = \sum_{i=1}^{15} z_t^i$$

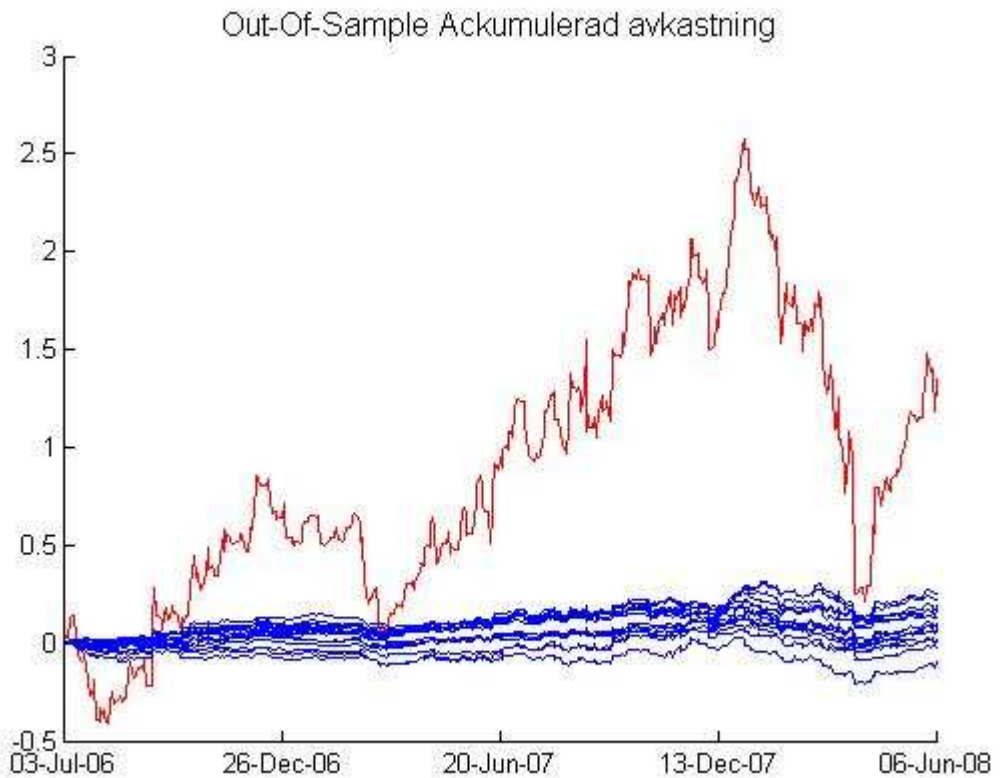
Figurena 4.2.2.1- 4.2.2.6 visar avkastningen för uppsatsens fyra olika beräknings perioder.



Figur 4.4.1 Avkastning under perioden 3-DEC-2001 till 30-JUN-2006, på datamängd 1

Test	Resultat (kr)	Akkumulerat resultat (kr)
1	0.1548	0.1548
2	0.2484	0.4032
3	0.0508	0.4540
4	0.1338	0.5878
5	0.3684	0.9562
6	0.2039	1.1601
7	0.2596	1.4197
8	0.2463	1.6660
9	0.3831	2.0491
10	0.5153	2.5644
11	0.1495	2.7139
12	0.1990	2.9129
13	0.4014	3.3143
14	0.5047	3.8190
15	0.6211	4.4401

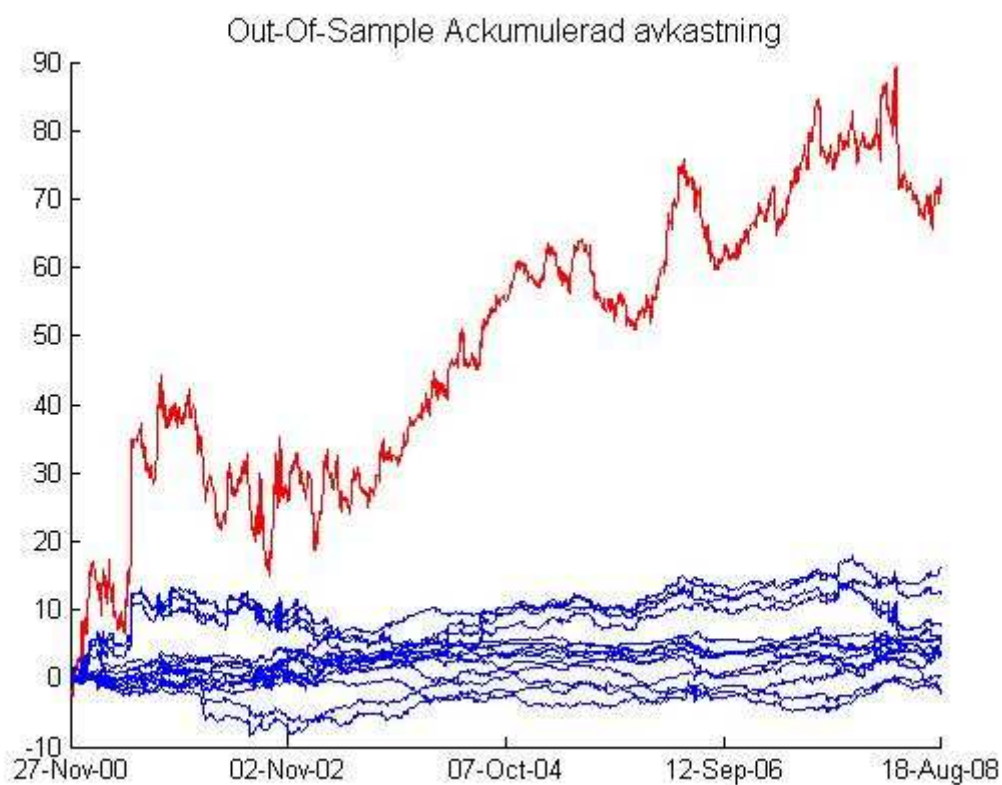
Tabell 4.4.2 Avkastning för varje variabelkombination och dess ackumulerade avkastning



Figur 1.4.3 Avkastning under perioden 3-JUL-2006 till 6-JUN-2008, på datamängd 1

Test	Resultat (kr)	Akkumulerat resultat (kr)
1	0.0157	0.1548
2	0.0170	0.1718
3	0.0596	0.2314
4	0.0097	0.2411
5	-0.0256	0.2155
6	0.1294	0.3449
7	-0.0979	0.247
8	0.1707	0.4177
9	0.2196	0.6373
10	0.0918	0.7291
11	0.1572	0.8863
12	0.0865	0.9728
13	0.1865	1.1593
14	0.2552	1.4145
15	0.0705	1.4850

Tabell 4.4.3 Avkastning för varje variabelkombination och dess ackumulerade avkastning



Figur 4.4.4 Avkastning under perioden 27-NOV-2000 till 20-AUG-2008, på datamängd 2

Aktiepar	Resultat (kr)	Ackumulerat resultat (kr)
2 och 7	3,6063	3,9484
6 och 9	3,9672	7,9156
1 och 4	-1,9998	5,9158
3 och 10	12,6559	18,5717
5 och 8	5,5842	24,1559
5 och 9	16,1401	40,296
6 och 10	3,9423	44,2383
8 och 4	4,0204	48,2587
8 och 2	0,5115	48,7702
1 och 7	-1,7303	47,0399
3 och 6	6,294	53,3339
7 och 4	5,7427	59,0766
5 och 10	6,1533	65,2299
3 och 9	7,9628	73,1927

Tabell 4.4.4 Avkastning för varje aktieindexpar och dess ackumulerade avkastning

4.5 Bootstrap metoden

I detta kapitel beskrivs Bootstrap metoden [16], en statistisk metod som används för att skatta hur bra vår handelsmodell är jämfört med en stor mängd slumpmässigt genererade handelsstrategier.

Tanken med Bootstrap metoden är att skapa ett nytt urval med observationer från det

ursprungliga urvalet, vilket i detta fall är vår dagliga avkastningsvektor. Det ursprungliga urvalet/stickprovet antas vara ett slumpmässigt urval från en större population, där varje observation antas vara oberoende och lika fördelad.

Bootstrap metoden används på vår handelsmodells dagliga avkastningsvektor $\{r_t^i\}_{t=1}^v$, där v är antalet handelsdagar. Vi börjar med att subtrahera medelvärdet

$\mu_r^i = \frac{1}{v} \sum_{t=1}^v r_t^i$ på varje r_t^i , därmed fås en mellanliggande sekvens

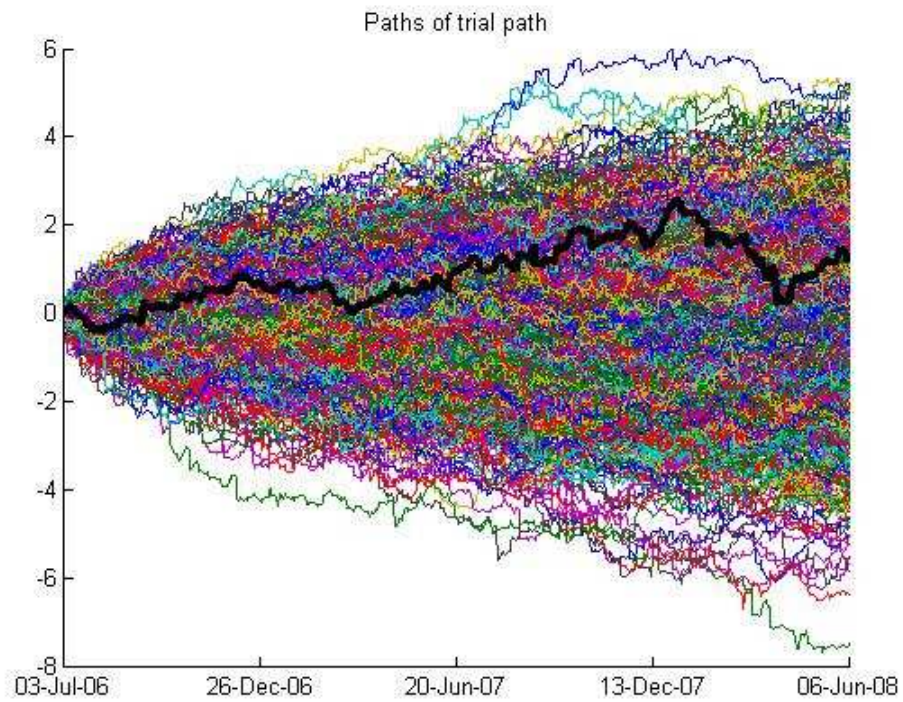
$\{g_t^i\}_{t=1}^v$, $g_t^i = r_t^i - \mu_r^i$ med ett förväntat väntevärde lika med noll. Från denna sekvens skapar vi L stycken slumpmässiga samples med längden v . Sen beräknar

vi L stycken $Z_t = \sum_{k=1}^t \sum_{i=1}^6 g_k^i$ upp till $t = v$. Ett signifikanstest utförs sedan för att se

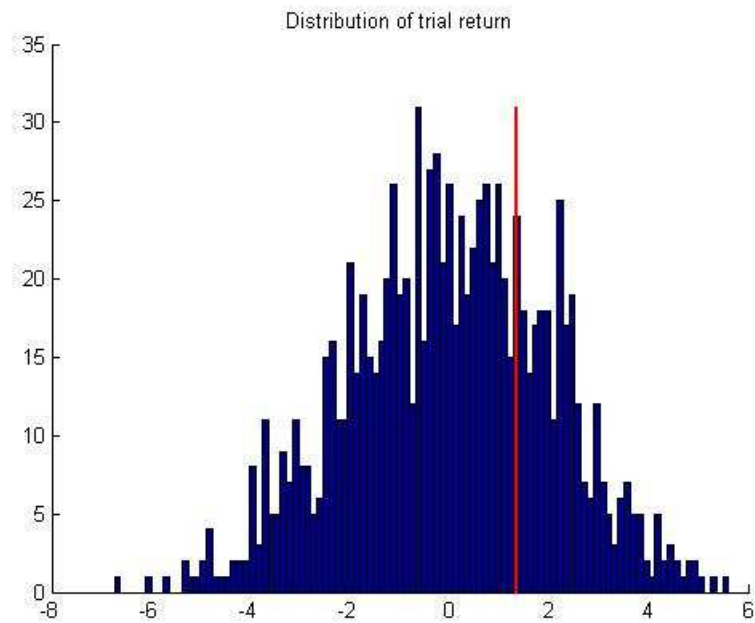
hur stor sannolikheten är att den ackumulerade avkastningen Z_t för en slumpmässig sampler hamnar ovanför den underliggande tidsseriens ackumulerande avkastning, \hat{Z}_t , dvs. $P(Z_v \geq \hat{Z}_v)$. I vänster kolumn i tabell 4.5.1 visas P-värden för vår handelsmodell beräknade med bootstrap metoden. P-värdet är ett mått på sannolikheten att en slumpmässigt samplad ackumulerad avkastning ger lika hög eller högre ackumulerad avkastning än den som fås av vår modell.

	Bootstrap metoden
Datamängd 1, 3/JUL/2006-6/JUN/2008	0.2790
Datamängd 2, 27/NOV/2000- 20/AUG/2008	0.0657

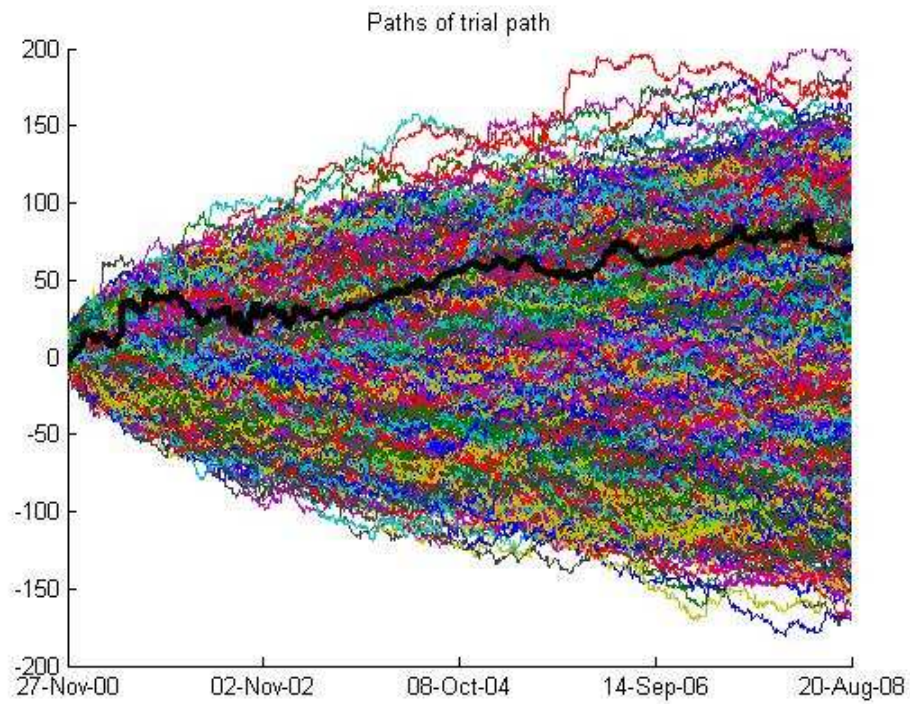
Tabell 4.5.1 P-värden för \hat{Z}_t



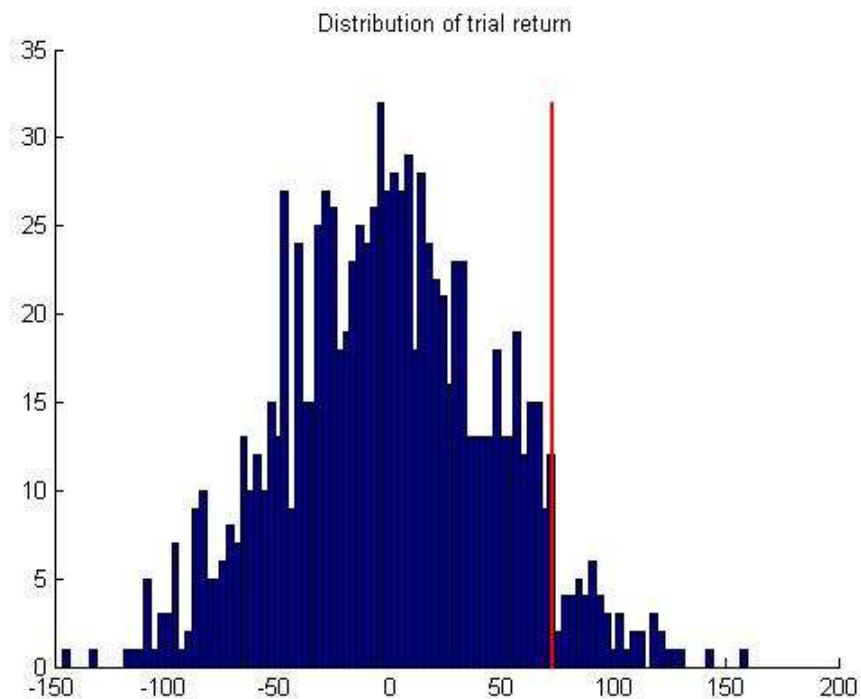
Figur 4.5.1 Bootstrap metodens Z_t för period 3-JUL-2006 till 6-JUN-2008, på datamängd 1. Den tjocka linjen är \hat{Z}_t , serien som ges av vår modell



Figur 4.5.2 Fördelningen av ackumulerad avkastning för period 3-JUL-2006 till 6-JUN-2008, på datamängd 1 för Bootstrap metoden. Den ljusare linjen indikerar vår modells ackumulerade avkastning.



Figur 4.5.3 Bootstrap metodens Z_t för period 27-NOV-2000 till 20-AUG-2008, på datamängd 2. Den tjocka linjen är \hat{Z}_t , serien som ges av vår modell



Figur 4.5.4 Fördelningen av ackumulerad avkastning för period 27-NOV-2000 till 20-AUG-2008, på datamängd 2 för Bootstrap metoden. Den ljusare linjen indikerar vår modells ackumulerade avkastning.

5 Analys

I detta kapitel analyseras resultatet av handelsmodellen.

5.1 Analys av in-sample resultatet

Vi börjar med att undersöka in-sample resultatet för tidsperiod 1:s träningsperioden. Av figur 4.2.2 och 4.2.3 framgår att den av AdaBoosten genererade slutgiltiga handelsstrategin ger en positiv riskjusterad avkastning som är större än alla de 20 stycken svaga basstrategierna. Figur 4.2.1 visar den årliga avkastningen i procent för $\tau = 120$ dagars perioder. Här ses att den starka handelsstrategin ger en avkastningskurva som under hela träningsperioden är positiv, precis vad som önskats. Jag har valt att bara visa figurer för $\tau = 120$, $\Delta\tau = 40$ och $r = 0.07$, detta eftersom alla andra variabelkombinationer som testats genererar snarlika figurer. Av figureerna kan man dra slutsatsen att AdaBoosten fungerar som önskat, d.v.s. den genererar en handelsstrategi som ger en positiv procentuell avkastning på hela perioden och avkastningen blir högre än för någon av de svaga basstrategierna.

Vidare undersöks resultatet för in-samples testdata. Figur 4.2.4-4.2.18 visar den riskjusterade avkastningen som genereras av varje variabelkombination. Här framgår att dessa blir allt igenom positiva. I figureerna till vänster, de som består av 4 stycken kurvor, ses att kurvorna följer varandra ganska tätt i de flesta fallen, med undantag för de första figureerna där de på slutet divergera lite mer. Av figureerna kan man dra slutsatsen att modellen är rätt så pålitlig för upprepade försök, med undantag då $r = 0.02$.

I tabellen 4.2.2 ses att IR för hela testdatat blir positivt vilket är väldigt bra och den totala portföljens IR blir därför också positiv. IR ökar vanligtvis då storleken på τ och på $\Delta\tau$ ökar och bäst IR hittas för $\tau = 120$ och $\Delta\tau = 40$. Detta beror på att då intervallen ökar i längd så minskar risken, i och med att man tar en större risk vid kortare tidsperioder.

För korta tidsperioder spelar tillfälligheterna en större roll och detta ökar risken. I figureerna 4.2.19-4.2.33 visas hur IR förändras med antalet iterationer. Här ses att IR ökar ganska mycket under de första iterationerna för att sedan ligga relativt stilla under de senare iterationerna. Detta tyder på att man kanske inte skulle behöva fullt så många iterationer som 20 stycken för att få fram ett tillförlitligt resultat, det kanske skulle räcka med 15 stycken iterationer? Här ses också att AdaBoosten lyckas vända negativa IR resultat till positiva i samtliga fall där de var negativa i den första iterationen.

5.2 Analys av out-of-sample resultatet

Nu går vi över till att titta på resultaten från out-of-sample simuleringen. Figureerna 4.3.1.1-4.3.1.16 åskådliggör den riskjusterade avkastningen för out-of-sample resultaten för datamängd 1. Här ses att tre variabelkombinationer (20 %)

ger negativt IR resultat men att den totala portföljen resulterar i ett positiv IR resultat på 0.4651. Detta resultat är lägre än IR för in-samples resultatet men fortfarande ett bra värde på IR. Precis som tidigare ses här att modellen är minst pålitlig vid låga värden på r , $r=0.02$, eftersom det främst är dessa IR-värden som har ändrats till det sämre från in-sample till out-of-sample resultatet.

Figur 4.3.2.1-4.3.2.15 visar den riskjusterade avkastningen för 14 stycken aktieindexpars-modeller för datamängd 2. Vi ser att av dessa får 86 % positiva IR resultat medan resterande 14 % blir negativa. Den totala portföljen resulterar även här i ett positivt IR på 0,5841. Så trots att vissa av aktieindexparen genererar en negativ avkastning kommer den slutgiltiga portföljen att vara positiv.

Figurerna med ackumulerad avkastning visar hur portföljernas totala avkastning blir starkt positiv jämfört med underliggande aktieindex. Här ses att den totala avkastningen i kronor blir positiv för samtliga testperioder.

I kapitel 4.5 undersöks kvalitén på vår handelsmodell med hjälp av den statistiska metoden Bootstrap. I tabell 4.5.1 illustreras p-värdet för de två undersökta datamängdernas out-of-sample. P-värdet beskriver sannolikheten för att en slumpmässigt vald handelsstrategi ger en lika bra eller bättre avkastning under tidsperioden. Att döma av det låga p-värdet, 0,0657, för datamängd 2 så verkar det som om vår modell är relativt säker när det gäller handel med aktieparen. Däremot är P-värdet för datamängd 1:s out-of-sample inte alls lika bra. Värdet ligger på hela 0.27 så sannolikheten att en slumpmässigt vald strategi ger lika bra eller bättre resultat än vår handelsstrategi är 27 %.

6 Slutsats

Detta examensarbete inleddes med två frågeställningar. Den första var om AdaBoost algoritmen är en lönsam strategi vid aktiehandel. Om strategin inte är det, vad finns det då för anledningar till det, var den andra? Dessa två frågor ledde fram till examensarbetets syfte:

”Kan man med hjälp av AdaBoosting hitta mer generella samband mellan replikering och aktieindex samt ta fram dynamiska handelsstrategier som visar positiva resultat på i rapporten undersökta tidsserier ”

Sammanfattningsvis har studien av vår handelsmodell gett ett positivt resultat för bägge datamängder som testats. Men ett positivt resultat har tyvärr inte genererats av alla variabelkombinationer och aktieindexparkombinationer som modellen testats på. T.ex. så har drygt 14 % av de testade aktieindexparen haft ett negativt riskjusterat resultat och samma siffra är för datamängd 1:s out-of-sample resultatet 20 %. Resterande procent har dock genererat positiva riskjusterande resultat vilket har gjort att de totala portföljerna trots några negativa resultat ändå fått en positiv riskjusterad avkastning.

I början av rapporten identifierades begränsningar med pairs trading. Längre fram i rapporten föreslogs en allmän boosting-baserad handelsmodell för att på bästa sätt kunna hitta och optimera portföljstrategier. Studien har visat att AdaBoosting effektivt kan upptäcka kompletterande handelsstrategier för att forma marknadsneutrala portföljer som ger stabil avkastning för en kort-till-medium lång investerings horisont. Vanligtvis är detta inte uppnåeligt för varken individuella dynamiska strategier eller för statistiska buy&hold strategier.

Ett sätt som kanske skulle kunna öka avkastningen i vår handelsmodell är om AdaBoosten inte bara skulle välja den modell som har flest perioder som överstiger tröskelvärdet utan att AdaBoosten också skulle kunna kombinera detta med att välja den modell som har högst avkastning. D.v.s. om två modeller har lika många perioder som överstiger tröskelvärdet så ska modellen välja den strategi som har störst avkastning. Det skulle kunna öka den totala avkastningen för slutmodellen, den starka basstrategin. Detta har dock inte testats i examensarbetet utan lämnas som förslag till vidare studier eller utveckling av handelsmodellen.

En mer detaljerad analys av handelsmodellens resultat på andra variabelkombinationer och aktieindexpar är önskvärt men faller utanför tidsramen för detta examensarbete. Man skulle såklart vilja testa alla möjliga aktieindexparkombinationer (10! stycken) och hade detta gjorts så skulle man antagligen ha fått en annan procentuell uppdelning mellan de positiva och negativa resultaten än de som uppvisats ovan i denna rapport. Detta lämnas också till vidare studier av modellen.

Sammanfattningsvis kan det konstateras att handelsstrategin som bygger på AdaBoost algoritmen har kunnat skapa ett totalt positivt resultat på de undersökta datamängderna.

Referenslista

- [1] A.N. Burgess, *Using cointegration to hedge and trade international equities*. Applied quantitative methods for trading and investment, edited by C. Dunis, J. Laws, and P.Naim, Wiley, 2003
- [2] Peter J Brockwell and Richards A. Davis. *Introduction to Time Series and Forecasting*. 2nd ed. New York: Springer Verlag; 2002 p. 15-16.
- [3] Peter Englund, Torsten Persson och Timo Teräsvirta (2003) "Statistiska metoden för ekonomiska tidsserier." Ekonomisk Debatt, årgång 31, nr 8.
URL: <http://www.ne.su.se/ed/pdf/31-8-petptt.pdf>. [Läst: 3/11-2008]
- [4] Enders, Walter (1995), *Applied Econometric Time Series*. New York, John Wiley & sons, Sid: 216, ISBN: 0-471-11163-5
- [5] Gujarati, Damodar (1995), *Basic Econometrics*. New York, McGraw-Hill, Third edition, Sid: 718, ISBN: 0-07-113964-8
- [6] Enders, Walter (1995), *Applied Econometric Time Series*. New York, John Wiley & sons, Sid: 221, ISBN: 0-471-11163-5
- [7] Burgees N. A. (1999) *A Computational Methodology for Modelling the Dynamics of Statistical Arbitrage*,
URL: <http://www.cocreativity.net/phdSkel.pdf>
- [8] Binh Do, Robert Faff och Kais Hamza (2006), *A new approach for modelling and estimation for Pairs Trading*
URL: http://www.fma.org/Stockholm/Papers/PairsTrading_BinhDo.pdf),
[Läst: 25/10-2008]
- [9] Valeriy V. Gavrishchaka, *Discovery of multi-spread portfolio strategies for weakly-cointegrated instruments using boosting-based optimization*,
URL: http://www.atlantis-press.com/php/download_paper.php?id=174, [Läst: 22/11-2008]
- [10] Carol Alexander och Anca Dimitriu (2004), *A Comparison of Cointegration and Tracking Error Models for Mutual Funds and Hedge Funds*,
URL: <http://www.icmacentre.rdg.ac.uk/pdf/discussion/DP2004-04.pdf>, [Läst: 22/10-2008]
- [11] W.H Press, S.A Teukolsky, W.T Vetterling, och B.P Flannery, *Numerical Recipes in C: The Art of scientific Computation*, Cambridge University Press, 1992, Sid: 444-455
- [12] M Whistler (2004), *Trading Pairs*, John Wiley&Sons, New Jersey

[13] Valeriy V. Gavrishchaka, *Boosting Frameworks in Financial Applications: From Volatility Forecasting to Portfolio Strategy Optimization*

URL: <http://fs.mis.kuas.edu.tw/~cobol/JCIS2005/papers/252.pdf>, [Läst: 22/11-2008]

[14] Robert E. Schapire, *A brief introduction to boosting*, Proceeding of the Sixteen International Joint Conference on Artificial Intelligence, 1999

[15] Yoav Freund and Robert E. Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of Computer and System Sciences, 55(1):119–139, August 1997.

URL: http://www.face-rec.org/algorithms/Boosting-Ensemble/decision-theoretic_generalization.pdf, [Läst: 21/11-2008]

[16] David Arson, *Evidence-Based Technical Analysis: Applying the Scientific Method and Statistical Inference to Trading Signals*. Wiley, Upper Saddle River, NJ, USA, 2006