

RRL a Magas Frekvenciájú Algoritmikus Kereskedésben

Milacski Zoltán Ádám, 5. évfolyam

Témavezető:

Dr. Lőrincz András

tudományos főmunkatárs



Budapest, Eötvös Loránd Tudományegyetem

**Informatikai Kar, Programozáselmélet és Szoftvertchnológiai
Tanszék**

2012

Tartalomjegyzék

1. Bevezetés	3
2. Kereskedési alapfogalmak	5
3. Rekurrens Megerősítéses Tanulás	7
3.1. A differenciált Sharpe – hányados	8
3.2. A költségfüggvény kritikája	9
4. Optimalizáció	10
5. Kiterjesztési lehetőségek	12
6. Numerikus kísérletek	15
6.1. Adatbázis	15
6.2. Paraméterek optimalizálása	16
6.3. Új inputok hatása	24
6.4. Simított átmenet	25
6.5. Kereskedési költség	27
7. Konklúzió	31
Irodalomjegyzék	32

1. fejezet

Bevezetés

A pénzügyi idősorok előrejelzése fontos, izgalmas terület, rengeteg nyitott kérdéssel. Amellett, hogy számos kutatás tárgyát képezi, az ingyenesen letölthető tőzsdei fejlesztő környezeteknek hála ma már bárki mélyrehatóan foglalkozhat a témával. Az automatizálás lehetősége új távlatokat nyitott az elemzésben. Elemzői oldalon új eszközök, módszerek jelentek meg, de a piac is változásokon ment keresztül, hiszen a korábban tőzsdék iránt kevésbé érdeklődő informatikusok figyelme is a pénzpiacok felé fordult, akik programjaikkal még fel is gyorsították ezt a világot.

Az emberi beavatkozás nélkül kereskedő robotok elsődleges célja, hogy a nap 24 órájában mentesítsék a befektetőket a folyamatos terhelés alól. A humán kereskedőket ugyanis több tényező is korlátozza: nem képesek éjjel-nappal felügyelni a piacot, emellett befolyásolják őket saját megérzéseik, és nem tudnak pillanatszerűen cselekedni. Egy programnak ezzel szemben nincs szüksége leállásokra, nem befolyásolják zavaró tényezők, és még a reakcióideje is minimális. Egy stratégia precíz elemzése is csak algoritmizálással érhető el, ahol ezek a kvalitások is egytől egyig kiaknázásra kerülnek.

A tőzsdei előrejelzés problémájának megközelítése többféle lehet. Az alapvető feltevés szerint érdemes a múltbeli folyamatokat megvizsgálni, és azok alapján stratégiákat építeni. Ez az úgynevezett technikai elemzés. Az egyik legszéleskörűbben alkalmazott ún. mozgóátlagos eljárás család mellett számos sikeres Jelfeldolgozás, Mesterséges Intelligencia alapú módszer látott napvilágot. Ide tartozik például a periodicitások feltárása, szezonálisan ismétlődő piaci tényezők megismerése Fourier-transzformációs eszközökkel. A mintázatok egyszerre akár több skálán is előfordulhatnak, ezek wavelet analízis, gépi tanulási módszerekkel sikeresen kiaknázhatóak. Itt elsősorban adatbányászatra és megerősítéssel tanulásra érdemes gondolni: neurális hálók, döntési fák, komplett ágensek képesek döntési kérdésekre jó válaszokat adni. Számos statisztikai modell is népszerű, ide tartoznak a kopulák [13] és az autoregressziók [14].

A kereskedők vizsgálódhatnak tetszőleges időskálán, és ma már lehetőség van egé-

szen mélyreható elemzésre is. Ez az úgynevezett magas frekvenciájú kereskedés világa, amikor is a lehető legsűrűbben mintavételezzük az idősort. Ekkor piaci változások a másodperc töredéke alatt, nagyjából ezredmásodperces szinten mennek végbe, így a korábban említett okokból kifolyólag ez a robotok versenyszférája. Sajnos a technikák nagy része az alacsony frekvenciára fókuszál, amikor is a változások percben mérhetőek. A tranzakciós költségek miatt nehezebb magas frekvenciájú eset kevésbé érintett az irodalomban.

Dolgozatomban ezt a területet fogom vizsgálni: egy alacsony frekvenciájú Mesterséges Intelligenciabeli eljárást fogok magas frekvenciájú környezetbe átültetni. Ez az úgynevezett Rekurrens Megerősítéses Tanulás (Recurrent Reinforcement Learning, RRL) lesz, mely a szakirodalomban az egyik legtöbbször hivatkozott algoritmus. Moody és Saffell 2001-ben megjelent [12] cikkében szerepelt először, és sikerességének nyomán később más cikkek alapjául is szolgált. Ez egy rekurrens neuronhálón alapszik, és a direkt RL módszerek családjába tartozik. A neuronháló inputjai adják ki a szokásos RL-beli állapotot, outputja pedig az akciót, mely a tőzsdén háromféle lehet: long pozíció, short pozíció, üres pozíció. A stratégiának a neuronháló súlyai felelnek meg, melyeket a megfelelő módon kell hangolni valamilyen kockázatkezelést tartalmazó célfüggvény (pl. Sharpe – hányados) optimalizálására.

A dolgozatban bemutatásra kerül az algoritmus eredeti változata, valamint annak továbbfejlesztései. Először megvizsgálom az algoritmus hatékonyságát magas frekvenciájú környezetben. Ezt követően új technikaként próbálok ki több, magas frekvenciában szokásos inputot is az eredetileg javasolt árfolyamváltozások mellett – ilyen a Limitáras kötési könyv különböző szintjeinek használata, mely bizonyos keretek között, mint látni fogjuk, növeli a hatékonyságot. Központi kérdés tárgyát képezi továbbá az üres pozíciók kezelése, hiszen ilyen akciót az RRL eredeti formájában 0 valószínűséggel lép meg – ez alacsony frekvenciájú kereskedésnél nem okoz gondot, magas frekvenciánál viszont már túl sok a bizonytalan szituáció, és ezeket ki kell tudni zárni. Itt két irányból közelítem meg a javítás lehetőségét. Az irodalomban Dempster és Leemans küszöbölést [3] javasoltak. Ezt hasonlítom össze egy természetes módosítással, Maringer és Ramtohul [8,9] munkája nyomán. Foglalkozom a két módszer egyesítésével is a dolgozatban. A futtatások és az elemzések valódi tőzsdei adatokkal történtek.

2. fejezet

Kereskedési alapfogalmak

Munkánk során Moody és Saffel [12] cikke alapján fogunk dolgozni, ezért kezdésként az ő jelölésrendszerük kerül összefoglalásra.

Tekintsünk egy idősort, tőzsde esetében ez tipikusan az árfolyam az idő függvényében, diszkrét időben jelöljük ezt p_t -vel. Teszteléskor mindig véges mintát használunk, így $t = 0, \dots, T$.

Ekkor az árfolyam változása additívan ($r_t = p_t - p_{t-1}$) vagy multiplikatívan ($r_t = \frac{p_t - p_{t-1}}{p_{t-1}}$) számolható. Utóbbi újrabefektetésnél jöhet szóba, mikor egyenlegünk fix hányadát forgatjuk a piacon, ami rizikós, hiszen egyre több pénzt kockáztatunk. Szerencsésebb adott (maximális) részvénytávval kereskedni, ekkor a kockázat mindig ugyanakkora (vagy legalábbis felülről becsülhető). Ez éppen az additív profitnak felel meg, és mivel egy jó kereskedő mindig kerüli a kockázatokat, mi ezzel fogunk számolni.

Legyen tehát

$$r_t = p_t - p_{t-1},$$

és a fix kötési méret μ . Ekkor jelentse F_t a tőzsdén felveendő pozíciót a t pillanatban. Alapvetően

$$F_t \in \{-1, 0, 1\},$$

hiszen lehet short (-1 , csökkenésre spekuláló) és long (1 , emelkedésre spekuláló) pozíciókat nyitni, valamint üres (0) pozícióval is jelen lehetünk a piacon, mint potenciális befektető. Jegyezzük meg, hogy F_t definiálható folytonos mennyiségként is. Ekkor $F_t \in [-1, 1]$, és azt szeretnénk tudni, hogy az aktuális szituáció mennyire számít csökkenőnek vagy emelkedőnek. Ebben a változatban -1 felel meg a 100% short és 1 a 100% long döntésnek, a többi pedig a köztes állapotoknak. Természetesen adódik, hogy $\text{sign } F_t$ irányba lépünk $\mu \cdot |F_t|$ részvénytávval, így μ a maximálisan megvásárolandó részvények számává alakul. Mindkét változatban rendre $F_0 = F_T = 0$, hiszen kezdetben nem rendelkezünk részvényekkel, és a kereskedést nélkülük is kell befejeznünk.

Minden tőzsdei tranzakció pénzbe kerül. Legyen az egy részvényre eső költség δ . Egyenlegünk változása ekkor az

$$R_t = \mu(F_{t-1}r_t - \delta|F_t - F_{t-1}|)$$

képlettel számolható: az előző lépésből megmaradt javak jelenértékéből ki kell vonni az esetleges pozíciócserék költségét. A kumulált profit képlete $P_t = \sum_{t'=1}^t R_{t'}$, és ha W_0 tőkével kezdjük a kereskedést, egyenlegünk

$$W_t = W_0 + P_t$$

szerint módosul.

Az árfolyam mellett néhány brókercégnél rendelkezésre áll az úgynevezett Limitáras kötési könyv is, mely formálisan $2s$ darab árfolyammal párhuzamos idősort jelent, ahol $V_t^{a,\ell}$ az eladási és $V_t^{b,\ell}$ a vételi volumen az ℓ -edik szinten (rendre $p_t^{a,\ell}$, $p_t^{b,\ell}$ árfolyamonkon), $\ell = 1, \dots, s$. Ha valaki kiad egy tőzsdei megbízást, az megjelenik ebben a kötési könyvben valamilyen formában (tárolódik, mint limitáras, vagy végrehajtodik, mint piacaras kötés). Egy ilyen tehát árfolyamon felüli extra információkat jelent más befektetők kötéseiről, és a várakozásokat modellezi. Ez elviekben segíthet is a predikció során.

Végül megjegyezzük, hogy a $\Delta t \geq 5$ perc fix esetén a mintavételezési frekvenciát alacsonynak szokás nevezni. Ekkor a kereskedési költségek, a relatíve nagy mozgások miatt, gyakorlatilag elhanyagolhatók. Magas frekvenciájú kereskedés esetén 1 másodperc $\geq \Delta t \geq 1$ ezredmásodperc és nem feltétlenül fix, lehet a tőzsde sajátideje (osztás adat) vagy a kötési könyv sajátideje (alosztás adat). Utóbbi esetben minden kötési könyvbeli változás új t időpontot eredményez, előbbiben pedig csak akkor veszünk figyelembe egy ilyen t időpontot, ha a kötési könyv mellett az árfolyam is megváltozik. A Mesterséges Intelligencia fogalmaival élve az alosztásos adatok felelnek meg a teljesen megfigyelhető, az osztásos és alacsony frekvenciájú adatok pedig a részlegesen megfigyelhető környezeteknek.

3. fejezet

Rekurrens Megerősítéssel Tanulás

Most rátérünk Moody és Safell alap algoritmusának [12], és annak kiegészítéseinek a bemutatására. Munkánk során végig egy rekurrens neurális hálóval fogunk dolgozni, mely egy darab tőzsdei idősoron fog prédikálni és kötések végrehajtani. Az eljárást Rekurrens Megerősítéssel Tanulásnak (Recurrent Reinforcement Learning, RRL) fogjuk nevezni.

Az egyszerűség kedvéért tekintsünk egy egyrétegű neurális hálót. Ez egy lineáris szeparátor. Legyen

$$F_t = \tanh(\underline{w}_t^T \underline{x}_t),$$

ahol $\underline{x}_t$ a neuronháló input vektorát, $\underline{w}_t$ pedig a szeparáló hipersík szokásos normálvektorát jelenti. F_t itt a korábban már definiált tőzsdei lépést jelenti, ez az output érték.

Az input vektort a következő módon választjuk meg. A hálónak átadunk egy M hosszúsági historikus részt az r_t árfolyamváltozásokból, valamint az előző F_{t-1} döntést. A tengelymetszet miatt bekerülő 1-essel együtt

$$\underline{x}_t = \begin{bmatrix} 1 & r_t & r_{t-1} & \cdots & r_{t-M+1} & F_{t-1} \end{bmatrix}.$$

Ez egy $\mathbb{R}^{M+2}$ -beli vektor. Elemeit szokás feature-öknek is nevezni.

Ezt követően $\underline{x}_t$ minden koordinátájához hozzárendelünk egy súlyt:

$$\underline{w}_t = \begin{bmatrix} v & w_0 & w_1 & \cdots & w_{M-1} & u \end{bmatrix}.$$

Ez is egy $\mathbb{R}^{M+2}$ -beli vektor. A súlyok lesznek azok a paraméterek, amiket az idősoron való lépegetés mellett a célunknak megfelelő módon változtatgatunk majd.

A hipersík így ketté fogja vágni a lehetséges input-ok vektorterét short és long részekre, ennek a tudásnak a birtokában fogjuk kiértékelni az aktuális helyzetet, és dönteni, milyen irányban nyitunk pozíciót.

Megerősítéssel tanulásbeli szemlélettel azt mondhatjuk, hogy $\underline{w}_t$ a stratégiánk, ami szerint döntéseket hozunk. $\underline{x}_t$ az állapot, amiben tartózkodunk. Az $\underline{x}_t$ állapotot ezután az F_t akcióra képezzük a fentebb látott képlet segítségével.

Gyakorlatilag egy autoregresszív jellegű modellt építünk a piacról. Ráadásul F_t függ F_{t-1} -től, azaz előző lépés output-ja a következő lépésben input-tá válik. Így a neuronhálónk rekurrens típusú, és a múltbeli információk az F_{t-1} számban tömörülnek.

3.1. A differenciált Sharpe – hányados

Természetes kérdés, hogy mi szerint módosítsuk a $\underline{w}_t$ vektort. Mi számít jó, ügyes kereskedésnek?

A hatékonyság két dologból tevődik össze. Nyilvánvalóan minél nagyobb profitot szeretnénk elérni. Emellett szeretnénk, hogy az egyes tranzakciók teljesítménye minél kiegyensúlyozottabb, stabilabb legyen. Ekkor várhatóan nem fognak becsúszni kiemelkedően nagy bukások, amikből akár egy is végzetes csapást mérhet az egyenlegünkre. A két elvárást az alábbi módon összegezzük:

$$S_t = \frac{\mathbb{E}R_t}{\mathbb{D}R_t},$$

ez az úgynevezett Sharpe – hányados. Ezt szeretnénk maximalizálni: minél nagyobb várható értéket szeretnénk kapni, minél kisebb szórás mellett.

Sajnos belátható, hogy a Sharpe – hányados nem optimalizálható online módon [10]. Tekintsük ezért inkább egy becslését:

$$\hat{S}_t = \frac{A_t}{\sqrt{B_t - A_t^2}},$$

ahol $A_t = A_{t-1} + \eta(R_t - A_{t-1})$ az R_t első, $B_t = B_{t-1} + \eta(R_t^2 - B_{t-1})$ pedig az R_t második momentumának exponenciális mozgóátlaga. Ekkor η az ablakhosszért felelő bátorságparaméter. η szerint deriválva juthatunk az úgynevezett differenciált Sharpe – hányadoshoz:

$$D_t = \left. \frac{d\hat{S}_t}{d\eta} \right|_{\eta=0} = \frac{B_{t-1}(R_t - A_{t-1}) - \frac{1}{2}A_{t-1}(R_t^2 - B_{t-1})}{(B_{t-1} - A_{t-1}^2)^{\frac{3}{2}}}.$$

A Taylor – sorfejtés tulajdonságai miatt elegendő ezt maximalizálni, ugyanis

$$\hat{S}_t \approx \hat{S}_{t-1} + \left. \frac{d\hat{S}_t}{d\eta} \right|_{\eta=0} \eta + O(\eta^2).$$

Szerencsére a D_t marginális hasznosságfüggvény már online optimalizálható sztochasztikus gradiens módszerrel.

3.2. A költségfüggvény kritikája

A rendszer legfontosabb előnye, hogy van értelmezhető kockázatkezelés. D_t akkor lesz nagy, ha számlálója nagy. Mivel a számlálóban egy különbség van, a kisebbítendőnek pozitívnak, a kivonandónak negatívnak kell lennie. Előbbi teljesül, ha $R_t > A_{t-1}$, ami számunkra azt jelenti, hogy az átlagos teljesítményünkhöz képest most jobban cselekedtünk. Utóbbi pedig akkor, ha A_{t-1} és $R_t^2 - B_{t-1}$ különböző előjelűek: jól kereskedünk kis szórás mellett, vagy rosszul kereskedünk nagy szórás mellett. Itt a második eset elsőre problémásnak tűnhet, de éppen az első feltétel garantálja nekünk, hogy a rossz kereskedés ellenére most jót léptünk, ezért megéri picit bátrabbnak lenni. Problémás viszont az $R_t > A_{t-1}$, $R_t^2 > B_{t-1}$ eset, a váratlanul nagy profit esete, hiszen ekkor a kivonandó nagyobb lehet a kisebbítendőnél, így a hasznosság negatívba fordul át. Ez a Sharpe – hányados definíciójából is látszik: a nagy szórású profitábilis periódusok is büntetést kapnak, ami nagy hiba, hiszen befektetőként az ilyen profitokat is szeretjük. Ezért célszerű csak a negatív profitok szórását büntetni, így juthatunk az úgynevezett Sortino – hányadoshoz [15]:

$$DDR_t = \frac{\mathbb{E} R_t}{\mathbb{D} [\min (R_t, 0)]}.$$

Ez minden negatív profitot penalizálni fog, a nagyokat különösen. A korábbiakhoz hasonlóan:

$$DD_t = \sqrt{DD_{t-1}^2 + \eta (\min (R_t, 0))^2 - DD_{t-1}^2}$$

veszi át B_t szerepét és

$$D_t = \left. \frac{dD\hat{D}R_t}{d\eta} \right|_{\eta=0} = \begin{cases} \frac{R_t - \frac{1}{2}A_{t-1}}{DD_{t-1}} & \text{ha } R_t > 0, \\ \frac{DD_{t-1}^2(R_t - \frac{1}{2}A_{t-1}) - \frac{1}{2}A_{t-1}R_t^2}{DD_{t-1}^3} & \text{ha } R_t \leq 0. \end{cases}$$

Az $R_t > 0$ esetben láthatóan nincs büntetés a kiugróan magas profitok számára.

Hasznosságfüggvénynek szokás még a Nyereség Indexet is választani [2]:

$$I_t = \frac{\mathbb{E} [\max (R_t, 0)]}{-\mathbb{E} [\min (R_t, 0)]}.$$

Ez akkor jelent profitábilis stratégiát, ha értéke 1-nél nagyobb. Ez sem bünteti a kiugróan magas profitot, viszont a kiugróan negatívát sem, hiszen szórását egyáltalán nem vesz figyelembe. Nem fog tehát stabil hozamra törekedni.

4. fejezet

Optimalizáció

A $\underline{w}_t$ vektor módosítása sztochasztikus gradiens módszerrel történhet. Ekkor

$$\underline{w}_t = (1 - \nu) \cdot \underline{w}_{t-1} + \rho \cdot \frac{dD_t}{d\underline{w}}.$$

ν numerikus stabilitást adhat: a segítségével a kis súlyok leszoríthatók, így kizárható minden jelentéktelen változó. Egy másik konstans, ρ felel a súlyok memóriájáért. Ezeket előre meghatározzuk, így csak a derivált értéke kérdéses. Mivel D_t az R_t -beli F_t -ken keresztül függ a w -ktől, láncszabályt kell alkalmaznunk a deriválás során.

$$\frac{dD_t}{d\underline{w}} = \sum_{t'=1}^t \frac{dD_{t'}}{dR_{t'}} \left(\frac{dR_{t'}}{dF_{t'}} \cdot \frac{dF_{t'}}{d\underline{w}} + \frac{dR_{t'}}{dF_{t'-1}} \cdot \frac{dF_{t'-1}}{d\underline{w}} \right).$$

Egy batch szummát kaptunk, amit bonyolult számolni. Célszerűbb online approximálni a legfrissebb taggal:

$$\frac{dD_t}{d\underline{w}_t} \approx \frac{dD_t}{dR_t} \left(\frac{dR_t}{dF_t} \cdot \frac{dF_t}{d\underline{w}_t} + \frac{dR_t}{dF_{t-1}} \cdot \frac{dF_{t-1}}{d\underline{w}_{t-1}} \right).$$

Ebben már egyszerű deriválásokkal megkapható majdnem minden rész, egyetlen kivételtől eltekintve. A Sortino – hányados esetében:

$$\begin{aligned} \frac{dD_t}{dR_t} &= \begin{cases} \frac{1}{DD_{t-1}} & \text{ha } R_t > 0, \\ \frac{DD_{t-1}^2 - A_{t-1}R_t}{DD_{t-1}^3} & \text{ha } R_t \leq 0. \end{cases} \\ \frac{dR_t}{dF_t} &= -\mu\delta \operatorname{sign}(F_t - F_{t-1}) \\ \frac{dR_t}{dF_{t-1}} &= \mu r_t + \mu\delta \operatorname{sign}(F_t - F_{t-1}). \end{aligned}$$

A problémás tényezőnél a Backpropagation through time algoritmust kell alkalmazni [17]:

$$\frac{dF_t}{d\underline{w}_t} \approx \frac{\partial F_t}{\partial \underline{w}_t} + \frac{\partial F_t}{\partial F_{t-1}} \cdot \frac{dF_{t-1}}{d\underline{w}_{t-1}} = (1 - \tanh^2(\underline{w}_t^T \underline{x}_t)) \cdot \left(\underline{x}_t + u \cdot \frac{dF_{t-1}}{d\underline{w}_{t-1}} \right).$$

Rekurzióval tehát könnyedén frissíthetjük a súlyokat.

Tanítást többféleképpen csinálhatunk. Kijelölhetjük például az idősor elejét kezdeti tanításra. Ekkor ezen az intervallumon túltanítással megkaphatunk egy lokálisan optimális stratégiát, melyet később lépésenként változtathatunk globálisan optimális stratégiává. Másik eshetőség, hogy mindig egy lokálisan optimális stratégiát tanulunk be egy t előtti időszakon, és ezt alkalmazzuk t -ben. Mi az előbbit fogjuk alkalmazni. Online túltanításhoz elegendő nagyjából 10-szer végigmenni az idősor kijelölt periódusán.

5. fejezet

Kiterjesztési lehetőségek

Most megvizsgáljuk, hogy melyek a rendszer előnyei, hátrányai, melyek azok a pontok, ahol érdemes lenne az algoritmuson módosítani. Összefoglaljuk a szakirodalomban lelhető legfontosabb javításokat.

Az implementáció szemszögéből fontos szempont, hogy a rendszer online optimalizálható, amit mindig szeretünk, könnyebb programozhatósága és gyorsabb konvergenciája miatt. Szerencsés, hogy a számolás során nem kell a korábbi R_t értékeket felhasználni, valamint az is, hogy D_t és a deriváltak analitikusan számolható mennyiségek. Elegendő mindössze néhány változó $t - 1$ időpontbeli értékét eltárolni a rekurzió miatt.

Gyakorlati szempontból kiemelendő, hogy az indirekt RL módszerekkel szemben a Rekurrens Megerősítéses Tanulás kifejezetten jól teljesít. Ezt több cikk is visszaigazolta [4, 5]. Ennek oka a bin-ek problémájában keresendő: egy árfolyam akár 5 tizedesjegy pontosságú is lehet. Hogy az állapotok közötti apró eltéréseket is meg lehessen különböztetni, meglehetősen sok bin-be kellene őket besorolni. A táblázatos Q-learning eljárás tipikusan 10^5 állapotot enged meg [16] – a tőzsdén azonban ennél többre is szükségünk lehet.

Hátrányként megemlítenéd az autoregresszív feltevés, a háló egyrétegűsége. Semmi nem implicálja, hogy a tőzsdén lineárisan szeparálható döntési feladat előtt állunk, mégis, a gyakorlatban megmutatták [6], hogy két (és így több) réteg alkalmazása nem jelent igazi előnyt. Ezt nekünk is sikerült visszaigazolnunk, a háló egy rejtett réteggel kiegészítve nagyjából hasonló teljesítményt nyújtott. Ez azonban még nem jelenti azt, hogy más módon sem lehet nemlinearitást bevinni a rendszerbe. Humán intelligenciával ugyanis lehetséges a rendszer kiterjesztése. Megtehető például több háló párhuzamos kiképzése különféle speciális helyzetekre, ahol valamilyen q_t mennyiség birtokában indikátor változókkal választjuk ki az aktuális helyzetnek megfelelő szeparátort, és aszerint lépünk akciót. Ez a Regime-Switching RRL eljárás [8], és az itt felhasznált ötlet ebben a dolgozatban is fontos szerepet tölt be. Ekkor $k + 1$ darab egymástól független neuronháló

mellett

$$F'_t = (F_{t,0} \cdot I[q_t < c_1]) + \sum_{j=1}^{k-1} (F_{t,j} \cdot I[c_{j+1} > q_t \geq c_j]) + (F_{t,k} \cdot I[q_t \geq c_k])$$

$$F_{t,j} = \tanh(\underline{w}_{t,j}^T x_t), \quad j = 0, \dots, k$$

lehet egy választás, ahol az indikátorok teljes eseményrendszert alkotnak, és minden háló saját súlyvektorral rendelkezik. q_t lehet például egy GARCH(p, q) predikció a volatilitásra, c_j -k pedig választhatóak a korábbi GARCH értékek $(k+1)$ -kvantiliseinek. Tanítás-kor a hálók $\underline{w}_{t,j}$ vektorai különböző súlyokkal módosulnak: az éppen aktív háló nagyobb, a többi kisebb súllyal. Indikátorok helyett finomíthatjuk is az átmeneteket logisztikus függvényvel [9]:

$$I[q_t \geq c_j] \approx \frac{1}{1 + e^{-\gamma(q_t - c_j)}},$$

$$I[c_{j+1} > q_t \geq c_j] = I[c_{j+1} > q_t] \cdot I[q_t \geq c_j] \approx \frac{1}{(1 + e^{-\gamma(c_{j+1} - q_t)}) (1 + e^{-\gamma(q_t - c_j)})},$$

ahol $\gamma \rightarrow \infty$ határátmenettel közelíthető az indikátor függvény.

Várható, hogy jobb inputok beiktatása a neuronhálónál azért jobb teljesítményt jelentene, de ilyet a szakirodalomban nem találni. Az algoritmus képes csupán az árfolyamváltozókból a teljes technikai elemzés elvégzésére, hiszen a nevezetes tőzsdei indikátorok, mint feature-ök, sem javítanak az eredményeken. A Dempster–Leemans szerzőpáros sejtése szerint viszont zárt, nem publikusan hozzáférhető információk, mint például a Limitáras kötési könyv és annak statisztikái [1], hozhatnak plusz tudást a rendszerbe. Ezt nem vizsgálták, csak felvetették, mint lehetőséget.

Előnyös lehet továbbá δ megnövelése ($\delta' = \kappa \cdot \delta$), az inputok 0 várható értékű és 1 szórású változókká standardizálása, $\underline{w}_t$ egységvektorra normálása minden lépésben [3]. Esetleg felhasználható az újonnan megkapott súlyvektor egy újabb, második döntés megtételére, és kereskedhetünk eszerint. Valamint az F_t kiértékelés diszkrétizálható küszöböléssel is:

$$F'_t = \begin{cases} 1 & \text{ha } F_t > \varepsilon^+, \\ -1 & \text{ha } F_t < -\varepsilon^-, \\ 0 & \text{különben.} \end{cases}$$

Az eredeti változatnak természetesen $\varepsilon^+ = \varepsilon^- = 0$ felel meg, és feltehető, hogy $\varepsilon^+ = \varepsilon^- = \varepsilon$. A rendszer paramétereit ($\delta, \eta, \rho, \nu, \varepsilon^+, \varepsilon^-$) továbbá érdemes lehet az aktuális időszoron kioptimalizálni.

Jövőbeli kutatások szempontjából fontos lehet, hogy a gradiens módszer helyettesíthető ARP tanulási szabállyal [7], a kettő módszer összehasonlítása éppen csak, hogy elkezdődött.

Végül megjegyezzük, hogy az eddigiek természetes módon általánosíthatók több pénzügyi eszköznél való együttes befektetésre. Ez a portfólió-optimalizálás [11]. Tegyük fel, hogy n darab párhuzamos idősor áll rendelkezésünkre, azaz $\underline{p}_t \in \mathbb{R}^n$ vektor. Ekkor kaphatunk egy megfelelő $\underline{F}'_t \in \mathbb{R}^n$ vektort a következő feltételek mellett:

$$\begin{aligned} F'_{t,a} &\geq 0, \\ \sum_{a=1}^n F'_{t,a} &= 1 \\ F'_{t,a} &= \frac{\exp(F_{t,a})}{\sum_{b=1}^n \exp(F_{t,b})}, \end{aligned}$$

ami egy nemlineáris programozási feladat. Nem engedjük meg a short pozíciókat, és a μ tőkét megosztjuk a potenciális befektetések között.

6. fejezet

Numerikus kísérletek

A bemutatott RRL algoritmus eredetileg alacsony frekvenciájú kereskedésre lett kitalálva. Most rátérünk a magas frekvenciájú kereskedés tárgyalására, ahol a költségek már komoly büntetést jelentenek, és ahol trendváltások is sokkal gyakrabban következnek be. Természetes kérdés, hogy a Rekurrens Megerősítéses Tanulás miként teljesít ilyen feltételek mellett, és ha a válasz az, hogy nem elég jól, lehet-e valamit tenni a javítás érdekében, akár a már tárgyalt kiegészítések segítségével, vagy valamilyen más módszerrel.

6.1. Adatbázis

A futtatások minden esetben a Limit Order Book System – The Efficient Reconstructor¹ honlapjáról letölthető, Apple részvényre vonatkozó kötési könyv felhasználásával történtek. Az adathalmaz 2012. június 21.-éről tartalmaz információkat, 9 és 16 óra közötti időszakról. Összesen $T_1 = 400736$ alosztás, ezen belül $T_2 = 64483$ osztás rekord áll rendelkezésre $s = 10$ árfolyamszintről.

Első körben kizárólag osztásos adatokkal fogunk foglalkozni. A viszonylag magas árfolyam miatt a részvények száma $\mu = 100$, a kezdőtőke $W_0 = 10000\$$ értéken került fixálásra. Fedezeti problémáktól eltekintettünk (azaz tőkeáttételt feltételeztünk). Az árfolyam

$$p_t = \frac{p_t^{a,1} + p_t^{b,1}}{2},$$

a tranzakciós költség pedig egyelőre legyen 0. Tanításkor viszont vegyük figyelembe a szokásosnak mondható $\delta = \frac{p_t^{a,1} - p_t^{b,1}}{2}$ értéket, az úgynevezett spread felét, mely pozíció nyitáskor és zárásakor is jelentkezik.

Célfüggvénynek a differenciált Sortino – hányados került kiválasztásra, aminek nyilvánvaló előnyeiről már esett szó az előző fejezetben.

¹<http://lobster.wiwi.hu-berlin.de/>

6.2. Paraméterek optimalizálása

A paraméterek beállításához a már említett optimalizációs eljárást alkalmazhatjuk. A Sortino – hányados maximalizálására:

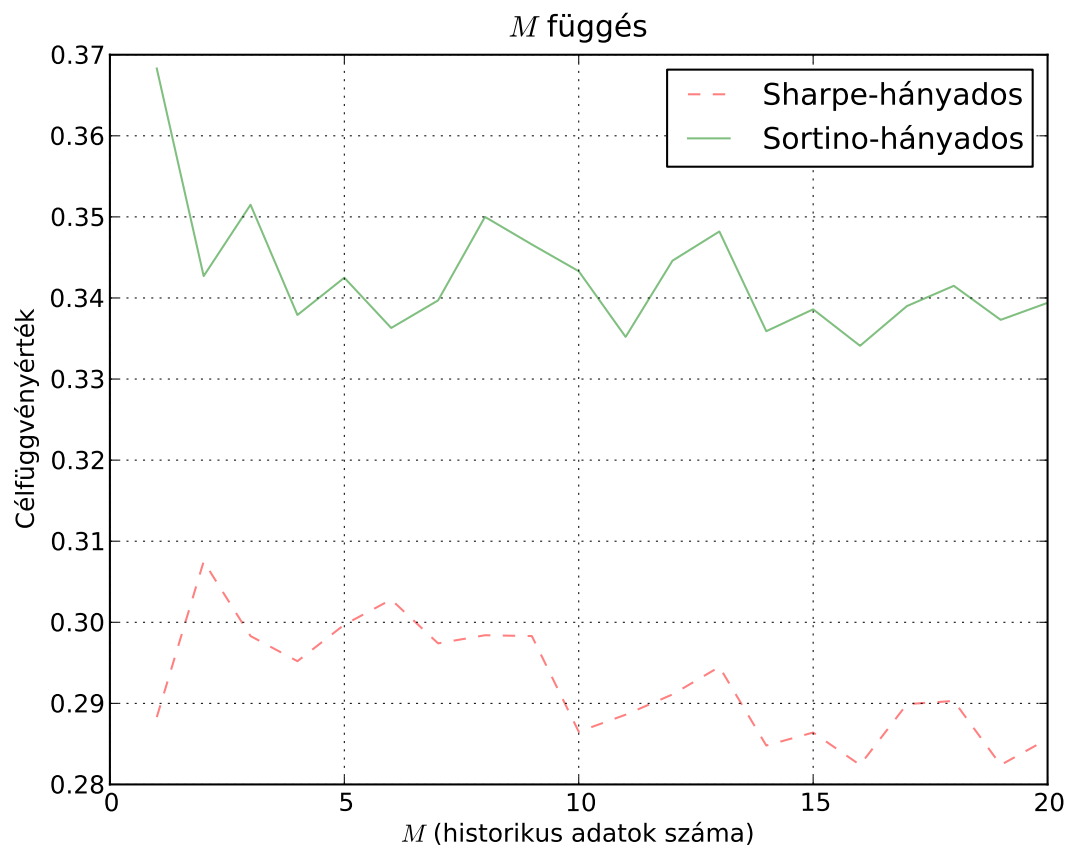
$$\max_{\varepsilon} \max_{\kappa} \max_{\nu} \max_{\eta} \max_{\rho} \max_M DDR_t.$$

Ez ebben a formában csak lokális maximumot fog garantálni, de miután a globális maximum megtalálása túl bonyolult feladat lenne, számunkra ez is elegendő lesz.

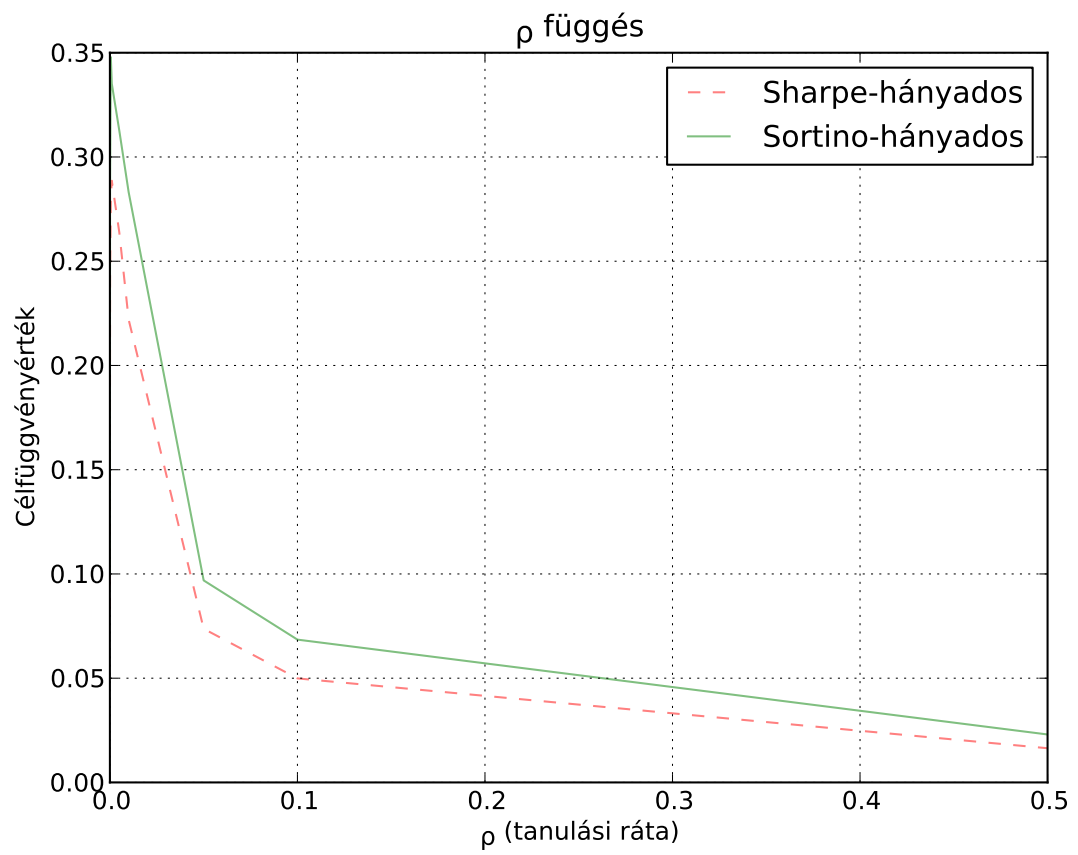
Az így kapott paraméterektől való függések a következő oldalak 6.2-től 6.7-ig számozott ábráiból olvashatók le. Némelyik paraméter optimuma szignifikánsan különbözik a hivatkozott cikkek által javasolt optimumtól, bizonyára az idősor osztásos volta miatt.

	ε	
	0	0.99
Sharpe – hányados	0.3715	0.7814
Sortino – hányados	0.3812	0.7827

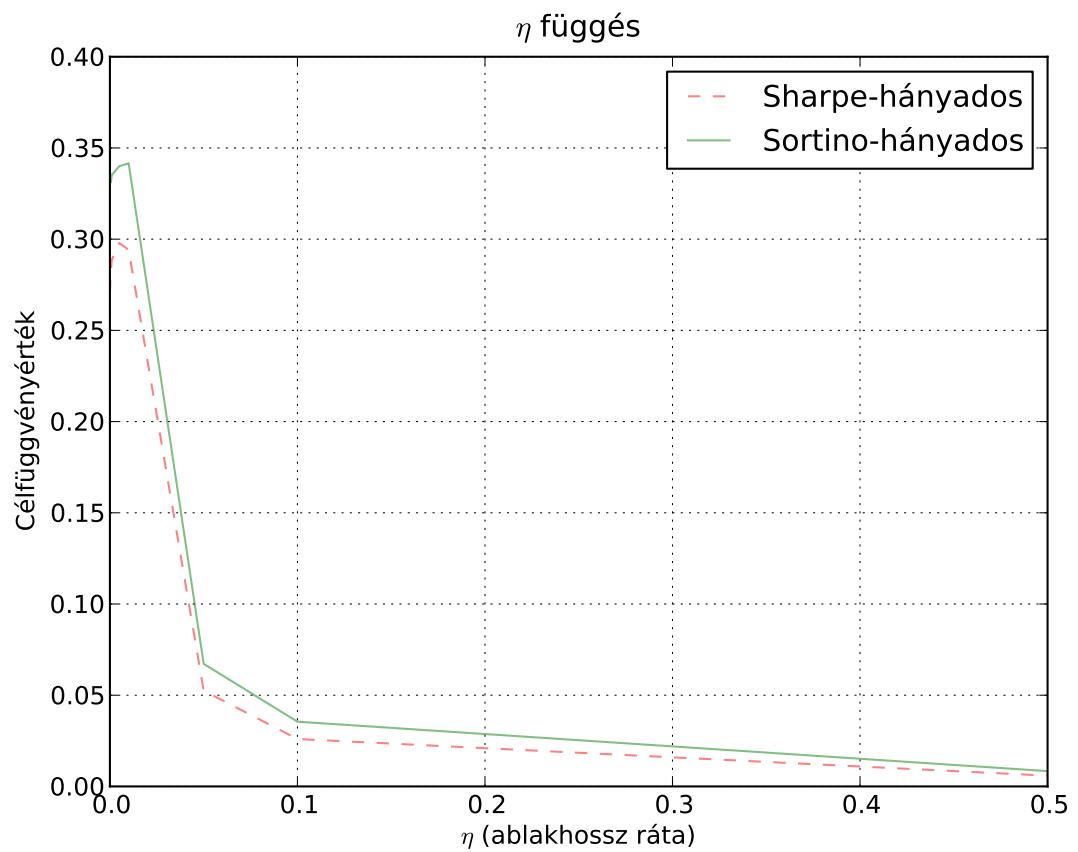
6.1. ábra. Optimális célfüggvényértékek az alap RRL-lel ($\varepsilon = 0$) és küszöböléssel ($\varepsilon = 0.99$).



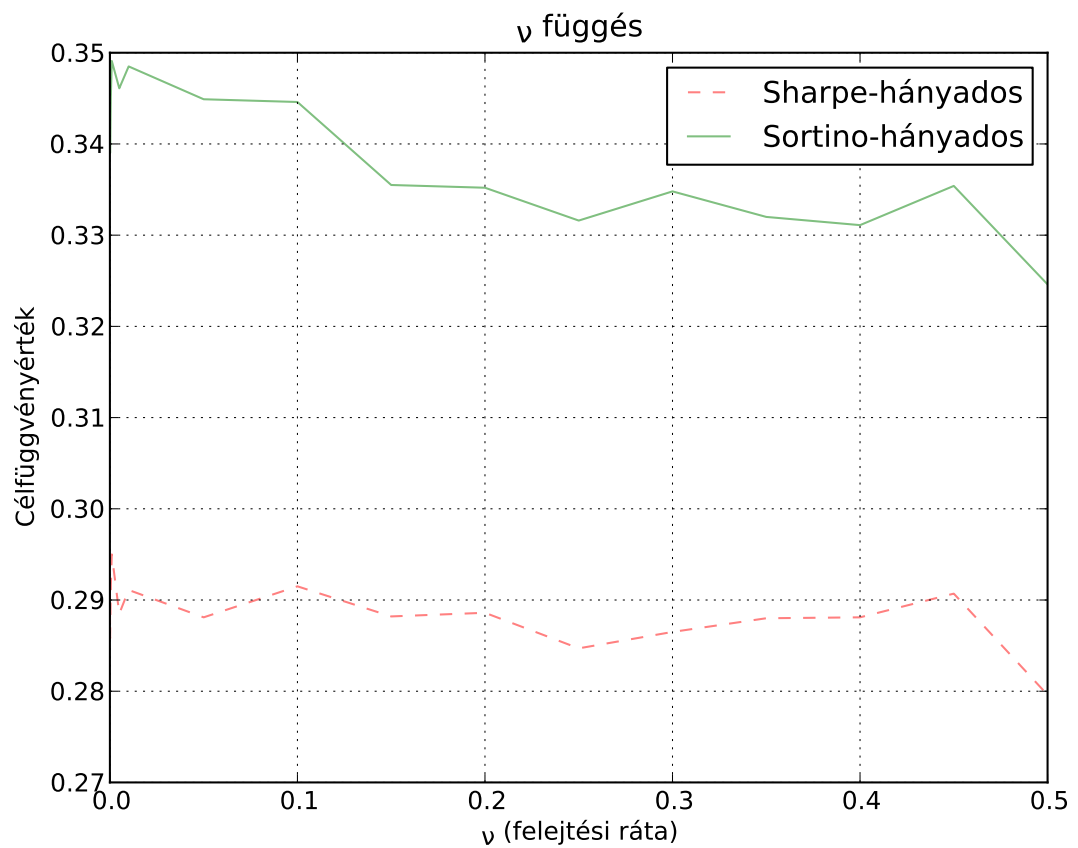
6.2. ábra. Az M paramétertől való függés. Látható, hogy csak 1 hosszú historikus adatot érdemes átadni a perceptronunknak.



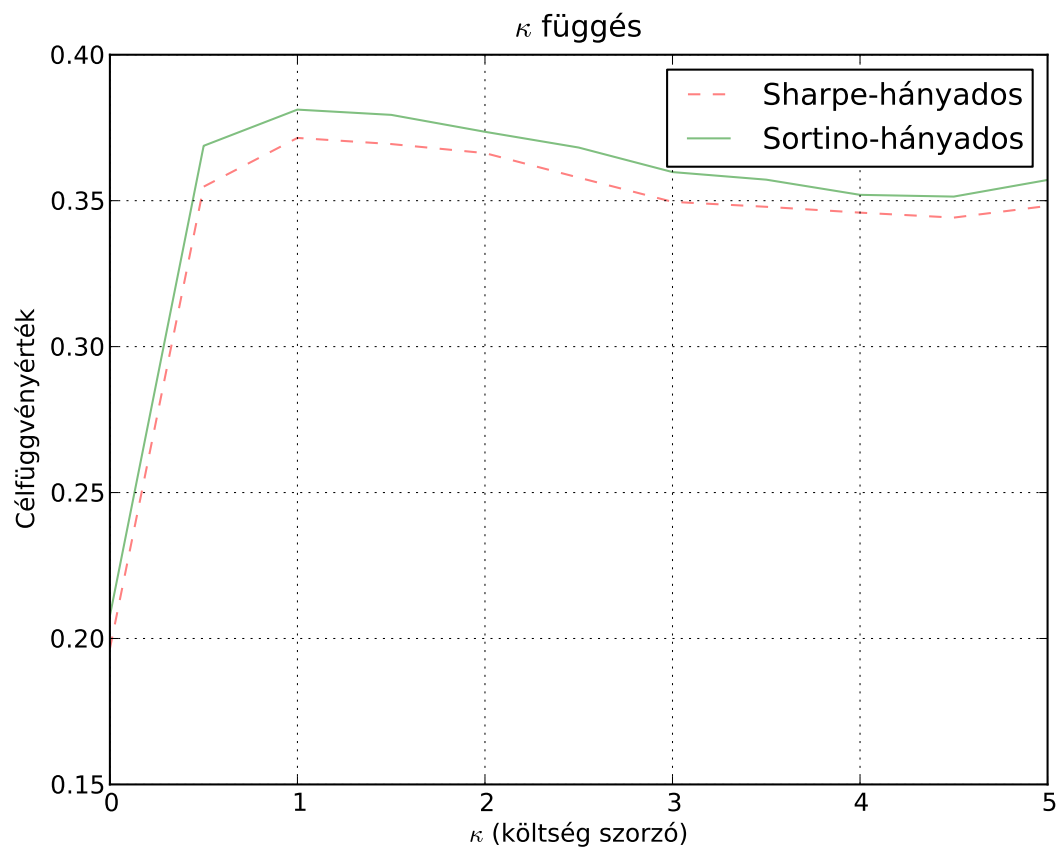
6.3. ábra. A ρ paramétertől való függés. Látható, hogy a tanulási rátának 0.0005 és 0.001 környékén van az optimális értéke.



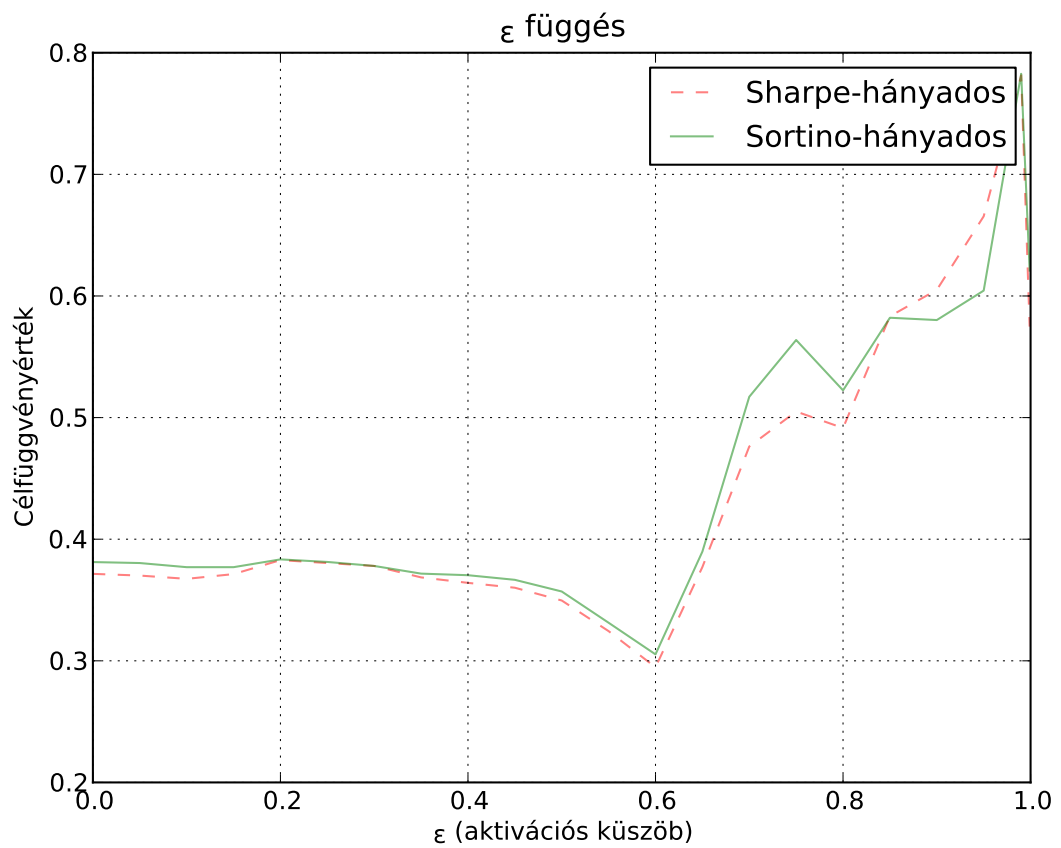
6.4. ábra. Az η paramétertől való függés. Látható, hogy az ablakhossz rátának 0.005 környékén van az optimális értéke.



6.5. ábra. A ν paramétertől való függés. Látható, hogy a felejtési rátának 0.001 környékén van az optimális értéke.

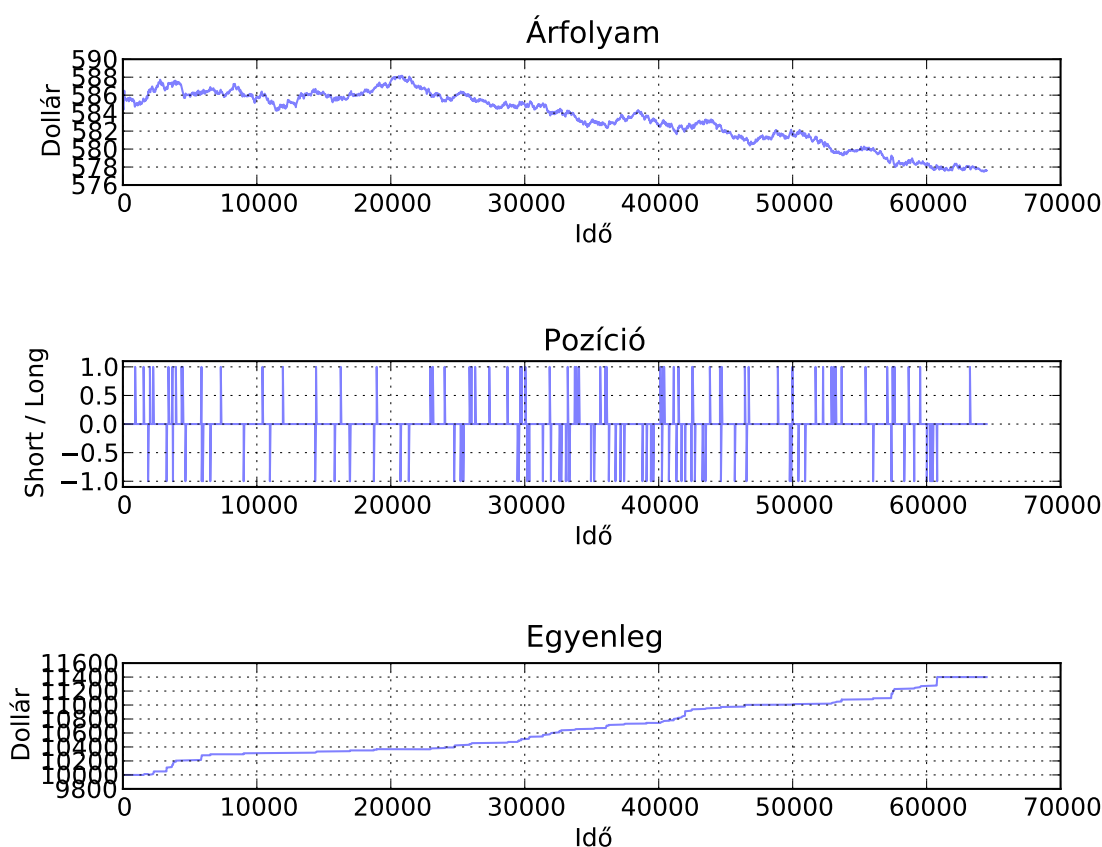


6.6. ábra. A κ paramétertől való függés. Látható, hogy a költség szorzójának 1 környékén van az optimális értéke.



6.7. ábra. Az ε paramétertől való függés. Látható, hogy a szimmetrikus aktivációs küszöbnek 0.99 környékén van az optimális értéke.

A küszöböléses változat tehát sokkal jobb az eredetinél, hiszen több, mint kétszer nagyobb hányadosokat produkált. Teljesítményét a 6.8 ábra és 6.9 táblázat szemlélteti. Itt figyelhető meg az optimalizált célfüggvény hatása a gyakorlatban: hosszútávon nagy és stabil hozamra törekszünk, azaz kicsi a lefele szórása a profitjainknak, így egyenlegünk szinte folyamatosan növekszik. A vizsgált időszakban az Apple árfolyama csökkenő trendet mutatott, az algoritmus még sem állt rá a trendkövetésre, hanem pillanatszerűen próbált pozíciókat nyitni. Ez részben a küszöbölésnek tudható be, hiszen egy magas küszöböt nem lehet hosszú időn keresztül elérni. A long és short akciók száma nagyságrendileg ugyanakkora, a trendelés ellenére mindkettőt azonos hatékonysággal sikerült meglépni. A 65.35%-os találati arány önmagában is jónak mondható, azonban a nyereségek veszteségekhez viszonyított nagysága kimagaslóan jóvá teszi a hatékonyságot. A futtatások $M = 1$, $\rho = 0.001$, $\eta = 0.01$, $\nu = 0.001$, $\kappa = 1$, $\varepsilon = 0.99$ értékek mellett történtek, globálisan optimális stratégiára törekedve.



6.8. ábra. Az árfolyam, az $\varepsilon = 0.99$ küszöbvel ellátott RRL pozíciói, egyenlege.

Short Pozíciók (nyerő %)	147	66.67%
Long Pozíciók (nyerő %)	156	64.10%
Nyerő Pozíciók (összes %-a)	198	65.35%
Vesztő Pozíciók (összes %-a)	105	34.65%
Nyerő/Vesztő %	188.57%	
Legnagyobb Visszaesés	11.0\$	
Legnagyobb Nyereség/Veszteség	27.0\$	-11.0\$
Átlagos Nyereség/Veszteség	7.75\$	-1.29\$

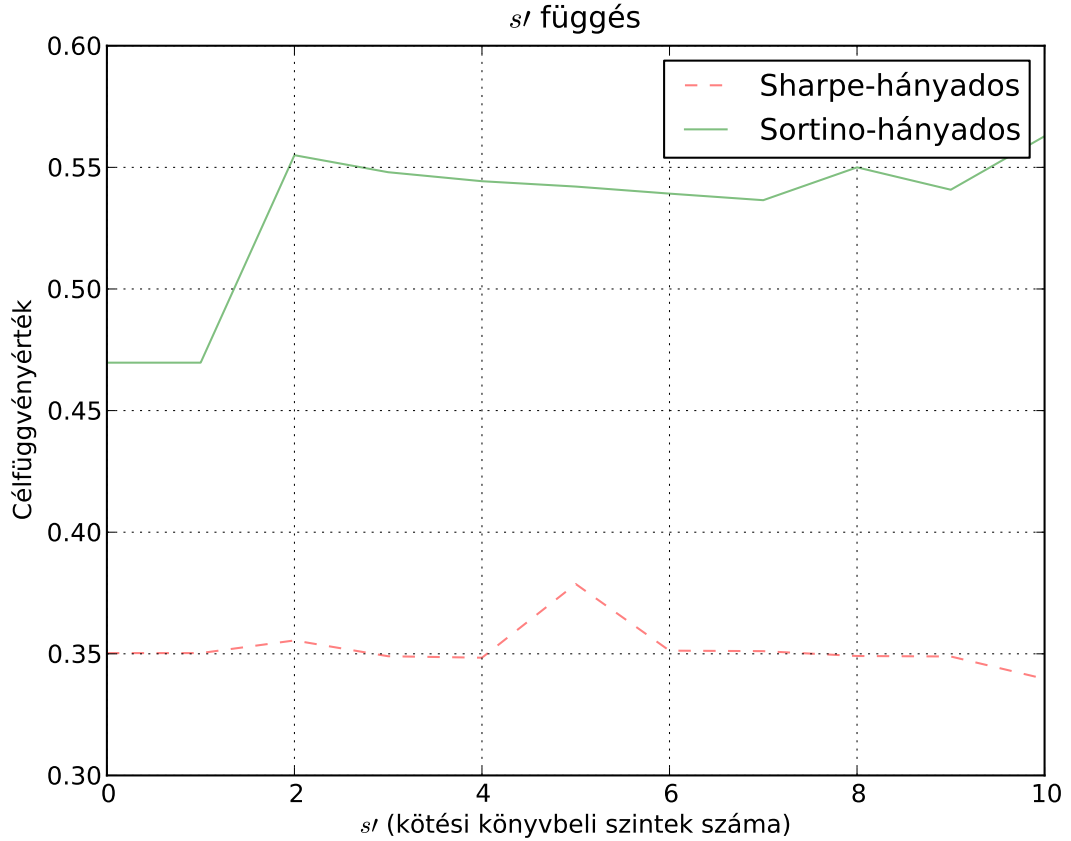
6.9. ábra. Statisztikák $\varepsilon = 0.99$ küszöböléssel.

6.3. Új inputok hatása

Ezidáig nem használtunk fel extra információkat, például a kötési könyvből. Vizsgáljuk meg, mi történik, ha az $\underline{x}_t$ vektorba $r_t, \dots, r_{t-M+1}$ mellé bevesszük a $V_t^{a,\ell} - V_{t-1}^{a,\ell}, V_t^{b,\ell} - V_{t-1}^{b,\ell}$ volumenváltozás párokat is, $\ell = 1, \dots, s', s' < s = 10$. Küszöbölt RRL esetén az információk hatása 6.10 szerint alakul.

A plusz feature-ök jól láthatóan javítják a predikciót alosztásos adatok mellett. Ha elég sűrű a mintavételezés, a kötési könyv tehát hoz némi pluszt (a Sortino-hányados a korábbi 0.4697-ről 0.5628-re nő), osztásos adatokkal nem sikerült hasonló eredményt produkálni (a Sortino-hányados ekkor 0.7827-ről 0.6499-re csökken). Ez elsősorban a teljesen és a részlegesen megfigyelhető környezet közötti különbségnek tekinthető. Az $\varepsilon = 0.99$ küszöb hozzávetőlegesen 5-ször akkora hányadosokra vezetett, mint a küszöb nélküli eset.

Az r_t árfolyamváltozás súlya $w_{r_t} = 0.5858$ lett, a volumendifferenciák első szintjén $w_{V_t^{a,1} - V_{t-1}^{a,1}} = -0.7921$, $w_{V_t^{b,1} - V_{t-1}^{b,1}} = -0.0652$, a második szinten $w_{V_t^{a,2} - V_{t-1}^{a,2}} = -0.027$ és $w_{V_t^{b,2} - V_{t-1}^{b,2}} = 0.0562$ súlyok alakultak ki. Itt érhető tetten az árfolyam trendeléséből fakadó asszimmetria, hiszen a részvény túladott volt. A többi súly a szintszám növekedésében csengett le. Az $s' = 2$ szintszám tekinthető kritikus értéknek, ezzel kaphatóak meg az igazán fontos magyarázó változók, de több szinttel további javulás tapasztalható.



6.10. ábra. Az s' paramétertől való függés alosztásos adatoknál, $\varepsilon = 0.99$ mellett. Látható, hogy a kötési könyvbéli volumenszintek számának 10 az optimális értéke.

6.4. Simított átmenet

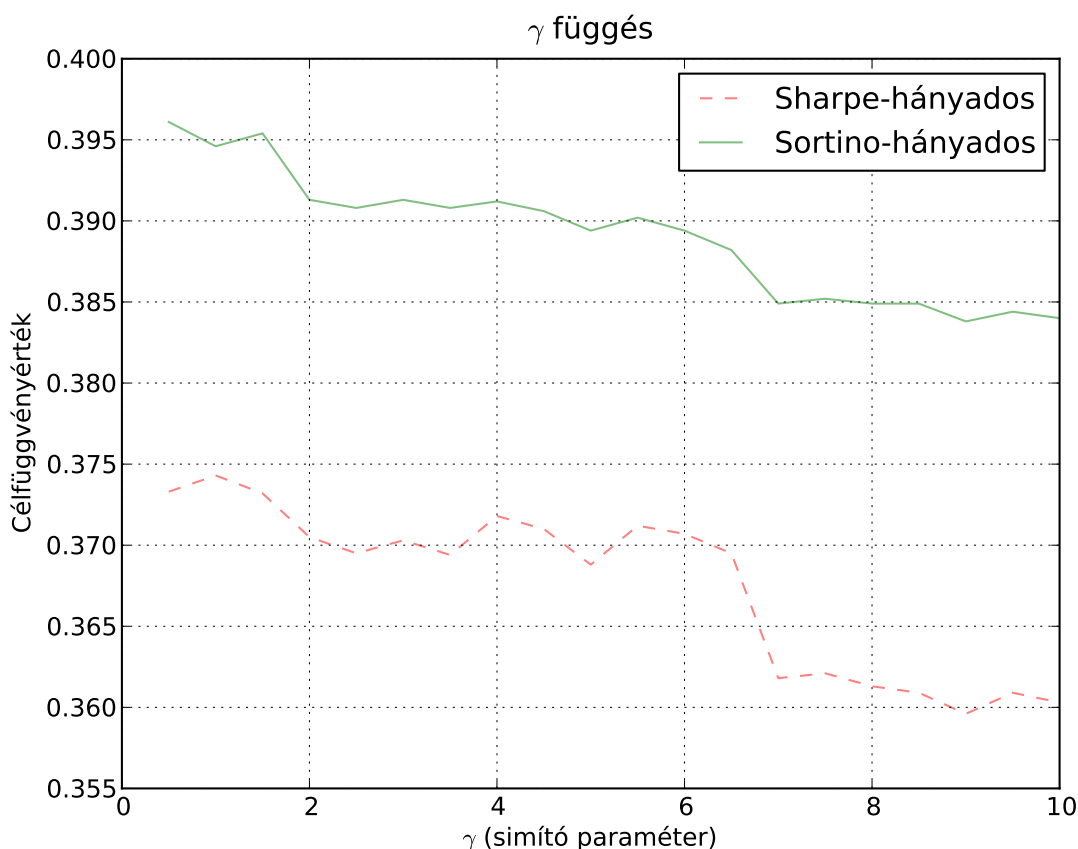
A bizonytalan szituációk kiszűrését sikeresen oldottuk meg egy ε küszöb bevezetésével. Most egy másik, saját módszert tekintünk, ahol

$$F_t = \tanh(w_t^T x_t) \cdot \frac{1}{1 + e^{-\gamma \cdot DDR_t}}$$

választással élünk. Itt most DDR_t -t úgy kezeljük, mintha egy külső konstans lenne, amit valahonnan kaptunk. Úgy teszünk, mintha csak bizonyítottan profitábilis szituációban lennénk hajlandók kereskedni, de ennél kicsit megengedőbbek vagyunk, simítjuk az igen és a nem állapotok közötti átmenetet.

Először is szükséges γ optimális beállítása, mely 6.11-en szemrevételezhető.

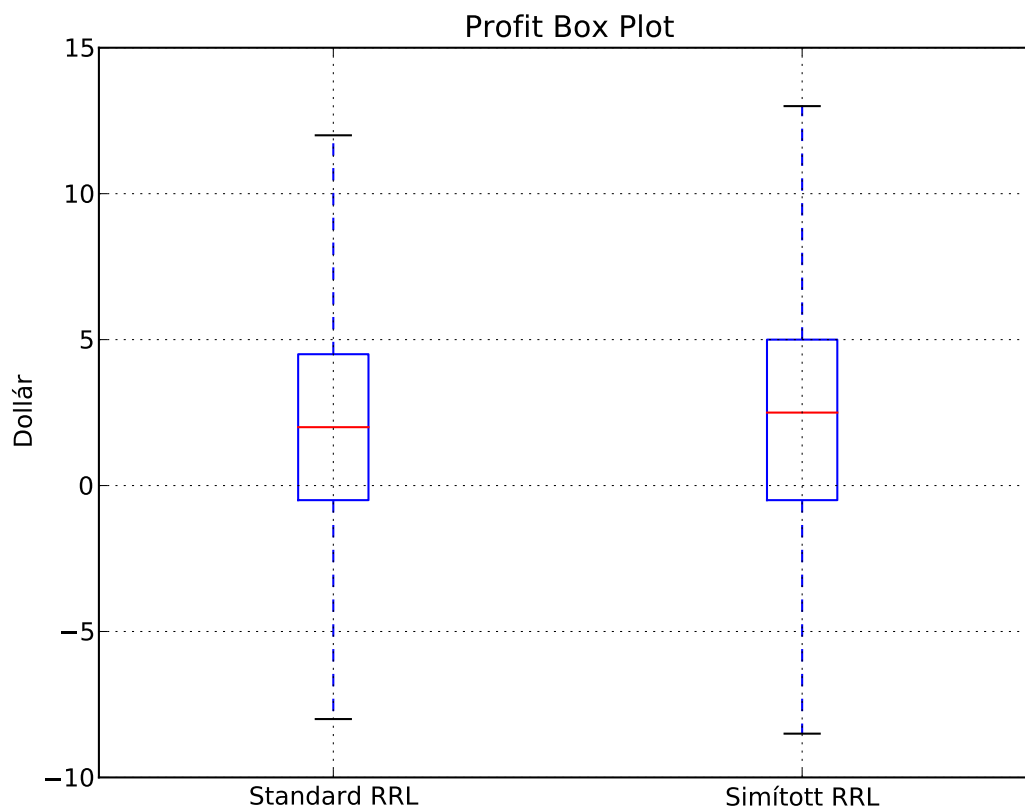
Az alap RRL-hez képest tehát sikerült picivel nagyobb hányadosokat elérni (Sharpe: 0.3733, Sortino: 0.3961), de a küszöböléssel kapott eredményt egyáltalán nem tudtuk megközelíteni. A két eljárás profitjait a 6.12 box plot segítségével tudjuk összehasonlítani.



6.11. ábra. A γ paramétertől való függés. Látható, hogy a simító paraméternek 0.5 és 1.0 környékén van az optimális értéke.

A simítás jól láthatóan kitolta a profitok mediánját, felső kvartilisét, maximumát a standard RRL eljáráshoz képest. Ez implicálja, hogy megvizsgáljuk a simított RRL küszöbölt változatát is, a két módszer ötvözetét. Így $\varepsilon = 0.499$ optimális választással 6.13 eredményeit kapjuk.

Az elért hányadosok nagyobbak a küszöbölt változaténál (Sharpe: 0.8612, Sortino: 0.9604). A felső kvartilis itt is megnőtt, viszont a medián lecsökkent, így lényegében a két módszer hasonló erejűnek mondható.



6.12. ábra. Alap RRL és Simított RRL profijainak box plot-ja.

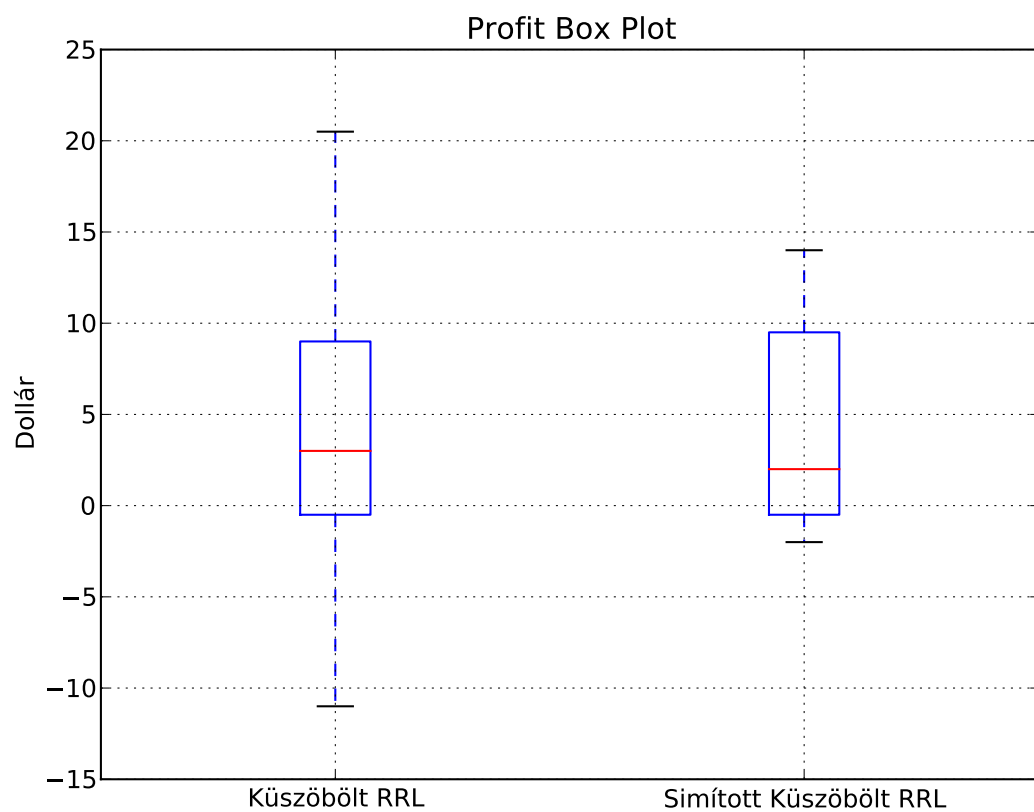
6.5. Kereskedési költség

Idáig eltekintettünk a tranzakciós költségektől. Az osztásos mintánkban az átlagos δ érték

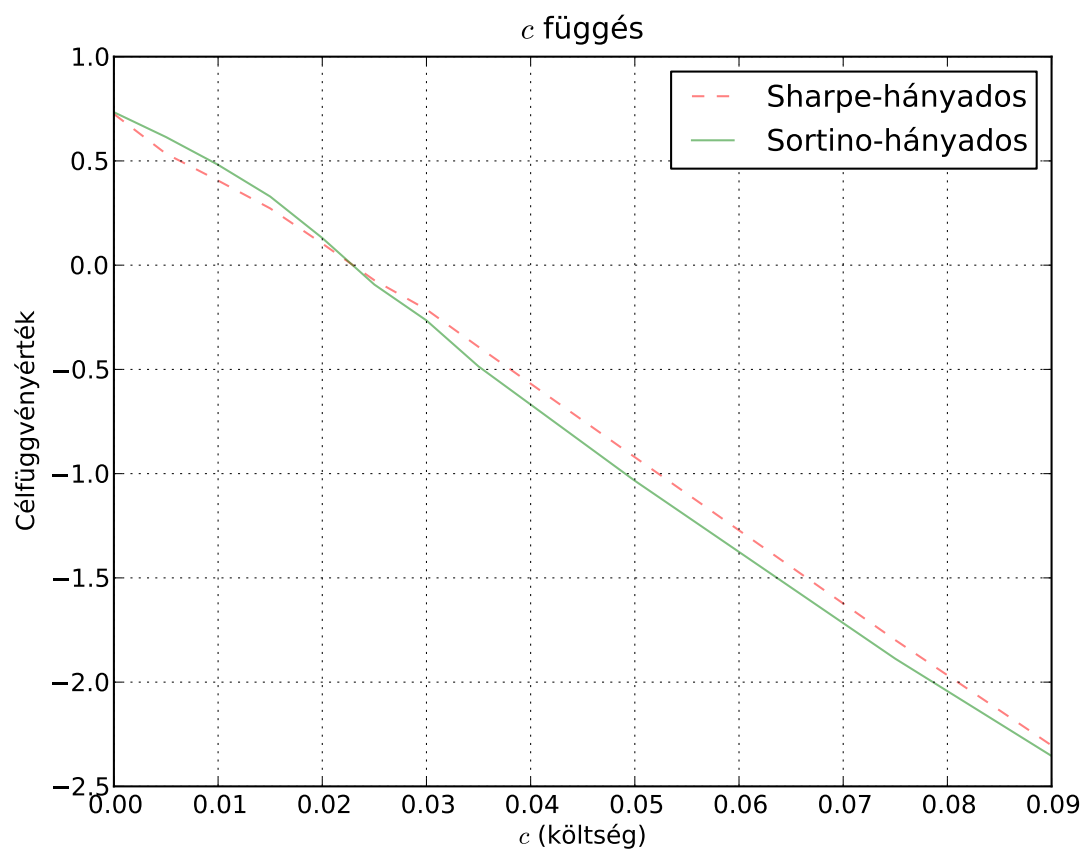
$$\sum_{t=1}^{64483} \frac{p_t^{a,1} - p_t^{b,1}}{2 \cdot 64483} = 0.0829,$$

0.0456 szórással. A spread fele így könnyedén szimulálható egy fix c értékkel. Az ettől való függés küszöbölt esetben 6.14-en látható.

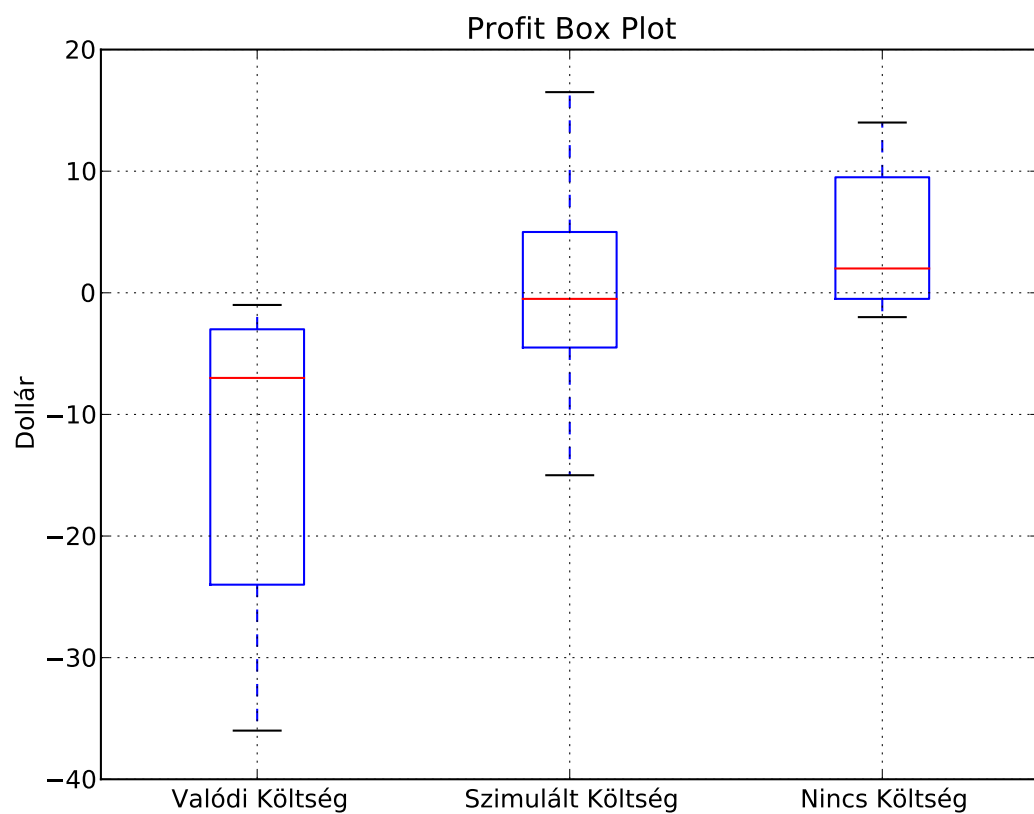
A valódi tranzakciós költségnek tehát csak a negyedét bírja el az algoritmus, így a gyakorlatban nem lehet magas frekvenciájú környezetben pénzt keresni vele, legalábbis nem ebben a formában. A valódi költséggel és a költség nélküli esettel való összevetést a 6.15 ábra mutatja.



6.13. ábra. Küszöbölt RRL és Simított Küszöbölt RRL profijainak box plot-ja.



6.14. ábra. Az c paramétertől való függés. Látható, hogy a spread kritikus értéke 0.020 és 0.025 környékén van.



6.15. ábra. Valódi, $c = 0.020$ és $c = 0$ tranzakciós költségek összevetése.

7. fejezet

Konklúzió

A Rekurrens Megerősítéses Tanulás algoritmust vizsgáltuk magas frekvenciájú környezetben, és azt találtuk, hogy az alapvetően alacsony frekvenciára használt eljárás igenis képes pontos predikcióra ilyen körülmények között is. Láttuk, hogy az eredeti algoritmushoz képest a küszöbölés komoly pluszt jelentett, hiszen így csak a legbiztosabb pozíciók váltak tranzakciókká, ezt követően pedig simított küszöböléssel hasonló eredményt értünk el. Ezenfelül kötési könyvbeli információk is segítenek a predikcióban, ha elegendően sűrűn mintavételezünk.

Az eredményeket azonban a gyakorlatban direkt módon nem lehet felhasználni, hiszen a tranzakciós költségek túl magas büntetést jelentenek. További vizsgálatot lehetne folytatni úgy, hogy csak minden j -edik osztást veszünk figyelembe, hiszen ekkor a tranzakciós költség fokozatosan egyre inkább elenyészik az árfolyamváltozások növekedése mellett. Ez ettől még magas frekvenciájú kereskedésnek számítana, hiszen továbbra is a másodperces tartományban lennénk. Esetleg tovább lehetne menni a portfólió-optimalizálás irányába, és meg lehetne nézni, hogy párhuzamosan több idősorból lehetne-e jobban előrejelezni.

Irodalomjegyzék

- [1] Mark P. Austin, Graham Bates, Michael A. H. Dempster, Vasco Leemans, and Stacy N. Williams, *Adaptive Systems for Foreign Exchange Trading*, Quantitative Finance **4** (2004).
- [2] Francesco Bertoluzzo and Marco Corazza, *Making Financial Trading by Recurrent Reinforcement Learning*, Knowledge-based Intelligent Information and Engineering Systems **4693** (2007), 619–626.
- [3] Michael A. H. Dempster and Vasco Leemans, *An Automated FX Trading System Using Adaptive Reinforcement Learning*, Expert Systems with Applications **30** (2006), 543–552.
- [4] Xin Du, Jinjian Zhai, and Koupin Lv, *Algorithm Trading using Q-Learning and Recurrent Reinforcement Learning*, 2009.
- [5] Stuart Duerson, Farhan Saleem Khan, Victor Kovalev, and Ali Hisham Malik, *Reinforcement Learning in Online Stock Trading Systems*, 2005.
- [6] Carl Gold, *FX Trading via Recurrent Reinforcement Learning*, Computational Intelligences for Financial Engineering, 2003, pp. 363–370.
- [7] Denise Gorse, *Application of Stochastic Recurrent Reinforcement Learning to Index Trading*, European Symposium on Artificial Neural Networks, 2011.
- [8] Dietmar Maringer and Tikesh Ramtohum, *Threshold Recurrent Reinforcement Learning Model for Automated Trading*, Applications of Evolutionary Computation **6025** (2010), 212–221.
- [9] ———, *Regime-Switching Recurrent Reinforcement Learning for Investment Decision Making*, Computational Management Science **9** (2012), 89–107.
- [10] Gabriel Molina, *Stock Trading with Recurrent Reinforcement Learning (RRL)*, 2006.
- [11] John Moody, Lizhong Wu, Yuansong Liao, and Matthew Saffell, *Performance Functions and Reinforcement Learning for Trading Systems and Portfolios*, Journal of Forecasting **17** (1998), 441–470.
- [12] John Moody and Matthew Saffell, *Learning to Trade via Direct Reinforcement*, IEEE Transactions on Neural Networks **12** (2001).
- [13] Andrew J. Patton, *A review of copula models for economic time series*, Journal of Multivariate Analysis **110** (2012), 4–18.
- [14] Martin Sewell, *Characterization of Financial Time Series*, 2011.
- [15] Frank A. Sortino and Lee N. Price, *Performance Measurement in a Downside Risk Framework*, The Journal of Investing **3** (1994), no. 3, 59–64.
- [16] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning*, MIT Press, Cambridge, MA. 1998.
- [17] Paul J. Werbos, *Backpropagation Through Time: What It Does and How to Do It*, Proceedings of the IEEE, 1990.