RLS 기반 Natural Actor-Critic 알고리즘을 이용한 트레이딩 전략

Trading Strategy Using RLS-Based Natural Actor-Critic algorithm

강대성*, 김종호*, 박주영*, 박경욱**
고려대학교 제어계측공학과*, 고려대학교 경영학과**

Daesung Kang*, Jongho Kim*, Jooyoung Park*, Kyung-Wook Park**
Dept. of Control & instrumentation Engineering, Korea University*
Dept. of Business Administration, Korea University**
E-mail : {mpkds, oyeasw, parkj, pkw}@korea.ac.kr

요 약

최근 컴퓨터를 이용하여 효과적인 트레이드를 하려는 투자자들이 늘고 있다. 본 논문에서는 많은 인공지능 방법론 중에서 강화학습(reinforcement learning)을 이용하여 효과적으로 트레이딩하는 방법에 대해서 다루려한다. 특히 강화학습 중에서 natural policy gradient를 이용하여 actor의 파라미터를 업데이트하고, value function을 효과적으로 추정하기 위해 RLS(recursive least-squares) 기법으로 critic 부분을 업데이트하는 RLS 기반 natural actor-critic 알고리즘을 이용하여 트레이딩을 수행하는 전략에 대한 가능성을 살펴 보기로 한다.

1. 서론

많은 자본이 시장으로 들어오고, 시장 변화에 대한 정보 역시 끊임없이 쏟아진다. 이 정보를 이용하여 트레이딩 신호를 정확하게 가리키는 알고리즘이 있다면 투자자들에게 매우 유용할 것이다. 본 논문의 취지는 강화학습 방법론 중 actor-critic 알고리즘을 이용하여 적절한 시점에 투자자들에게 정확한 신호를 주어 이익을 최대화시키는 것이다. 하지만 이익만을 최대화시키다보면 그에 따른 위험 역시 커지기 때문에 위험과 수익 사이의 적절한 조합 내에서 투자를 해야 한다. 위험과 수익사이의 tradeoff를 표현하는 방법 중 하나는 위험 적응 수익(risk adjusted return)을 이용하는 것이다. 그 중에서 가장 보편적으로 사용하는 Sharpe ratio를 이용해서 수익을 최대화시키지 않고 위험까지 고려한 위험 적응 수익을 최대화시키는 것이 본 논문의 목적이다. 실제로 트레이딩과 자산관리 운용의 최적화문제는 이러한 위험과 수익사이간의 적절한 조합을 찾는 것이라고 해도 과언이 아니다.

2. Sharpe ratio에 근거한 트레이딩

본 논문에서는 매 시간마다 일정한 투자 포지션 크기를 유지하는 시장에 대해서 다룬다. 투자자가 취할 수 있는 포지션은 일정한 크기를 가지는 long position과 short position, 즉 $F_t \in \{1, -1\}$이라고 가정한다. 여기서 말하는 long position이란 시장에서 물건(주식, 선물, 외환 등)을 사는 것을 말하고, short position은 반대로 물건을 파는 것을 의미한다. policy F_t는 다음과 같은 식을 통해서 구할 수 있다.

$$\pi_{\theta_t}(F_t = 1 | s_t) = 1/(1+\exp(-\phi^T\theta)) \quad (1)$$

논문에서 여러 가지 표기가 사용되는데 먼저 z_t는 시간에 따른 가격 변동을 의미하고, b_t는 $b_t = z_t - z_{t-1}$ 로 $t-1$시간일 때와 t시간일 때의

가격 변동 차이를 의미한다. 그리고 t시간일 때 트레이딩으로 얻게 되는 보상값(rewards)은 r_t로 표기하고 다음과 같이 정의한다.

$$r_t = \mu[F_{t-1} b_t - \delta |F_t - F_{t-1}|] \quad (2)$$

여기서 μ는 트레이딩하는 포지션 크기, δ는 transaction cost 비율을 의미한다. 그리고 수익은 P_T는 보상값의 합으로 나타낸다. 위험 적응 수익을 이용하기 위해 사용하는 Sharpe ratio는 수익의 평균값을 수익의 표준 편차를 나눈 것으로 구할 수 있다.

$$S_T = \frac{average\,(r_t)}{standard\ deviation\,(r_t)} \quad (3)$$

실시간 학습을 위해, 트레이딩 수익에 대한 Sharpe ratio의 영향을 계산하기 위해서 differential Sharpe ratio라고 하는 새로운 목적함수가 필요한데,

$$S_t|_{\eta>0} \approx S_t|_{\eta=0} + \eta \frac{dS_t}{d\eta}|_{\eta=0} \quad (4)$$

$\frac{dS_t}{d\eta}$가 새로운 목적함수인 differential Sharpe ratio가 된다. $\frac{dS_t}{d\eta}$는 r_t에만 의존하기 때문에 다음과 같이 정의할 수 있다.

$$D_t \equiv \frac{dS_t}{d\eta} = \frac{Y_{t-1}\Delta X_t - 1/2 X_{t-1}\Delta Y_t}{(Y_{t-1} - X_{t-1}^2)^{3/2}} \quad (5)$$

X_t와 Y_t는 r_t에 대한 각각 1차, 2차 모멘트의 exponential moving estimates를 나타낸다.[7]

$$X_t = X_{t-1} + \eta \Delta X_t = X_{t-1} + \eta(r_t - X_{t-1}) \quad (6)$$

$$Y_t = Y_{t-1} + \eta \Delta Y_t = Y_{t-1} + \eta(r_t^2 - Y_{t-1}) \quad (7)$$

3. RLS-based Natural Actor-Critic

본 논문에서는 actions $a_t \in A$, rewards $r_t \in R$, states $s \in S$인 discounted reward 강화학습 문제[1]에 대해서 다룬다. 에이전트의 목적은 수익의 discounted sum을 최대화시키는 policy를 찾는 것이다. 즉 $J(\pi)$를 찾는 것이다.

$$J(\pi) \equiv E\left\{\sum_{k=0}^{\infty} \gamma^k r_k | s_0, \pi\right\} \quad (8)$$

여기서 $\gamma \in (0,1)$인 감쇠율을 뜻하고, r_k는 상태가 s_k에서 s_{k+1}로 변할 때 얻게 되는 이익(또는 손해)이고, s_0는 시작 상태이고, π는 policy를 뜻한다. 일반적으로 policy는 다음과 같은 조건부 확률로 표현 할 수 있다.

$$\pi(a|s) \equiv \Pr\{a_t = a | s_t = s\} \quad (9)$$

목적함수를 value function과 action value function을 이용하면 다음과 같이 바꿀 수 있다.

$$J(\pi) = V^{\pi}(s_0) = \sum_a \pi(a|s_0) Q^{\pi}(s_{0,a}) \quad (10)$$
$$= \sum_s d^{\pi}(s) \sum_a (a|s) r(s,a)$$

3.1 Actor

actor의 목적은 기대 보상값 $J(\pi)$를 최대화시키는 policy 파라미터를 얻는 것이다. policy를 개선시키는 가장 직접적인 방법은 steepest gradient ascent를 사용하는 것이다.

$$\theta_{i+1} = \theta_i + \alpha \nabla_{\theta} J(\theta_i) \quad (11)$$

policy gradient $\nabla_{\theta} J(\theta)$를 추정하기 위해서 (10)을 다음과 같이 바꿀 수 있다.[2]

$$\nabla_{\theta} J(\theta) = F(\theta) w \quad (12)$$

gradient를 기반으로 하는 알고리즘 중 $\pi_{\theta}(a|s)$의 파라미터 벡터 θ를 업데이트하는 좋은 방법 중 하나는 natural gradient 방법이다.[6] 그 방법을 이용하면 J의 natural gradient는 다음과 같이 간단하게 표현된다.

$$\tilde{\nabla}_{\theta} J(\theta) = G^{-1}(\theta) \nabla J_{\theta}(\theta) = G^{-1}(\theta) F(\theta) w = w \quad (13)$$

Fisher information matrix $G(\theta)$는 (12)에서 언급된 행렬 $F(\theta)$와 정확하게 같다는 것이 [2]에 증명되었다. 결국 natural gradient는 다음과 같이 근사된다.

$$\tilde{\nabla}_{\theta} J(\theta) = G^{-1}(\theta) F(\theta) w = w \quad (14)$$

actor 파라미터 θ를 다음과 같이 간단하게 업데이트 된다.

$$\theta \leftarrow \theta + \alpha \tilde{\nabla}_{\theta} J(\theta) \approx \theta + \alpha w \quad (15)$$

α는 actor의 학습율을 의미한다.

3.2 Critic

본 논문에서는 policy π_{Θ}를 구하기 위해서 다음과 같은 선형 함수를 사용한다.

$$\tilde{A}_w(s,a) \equiv \nabla_{\theta} \log \pi_{\theta}(a|s)^T w \quad (16)$$

$$\tilde{V}_v(s) \equiv \phi^T(s) v \quad (17)$$

(16)와 (17)은 각각 advantage value function $A^{\pi_{\theta}}(s,a)$와 value function $V^{\pi_{\theta}}(s)$의 근사 함수이고, $\phi(s)$는 basis function, v는 critic 파라미터를 의미한다. 누적기여도(eligibility trace)를 이용해서 $Q^{\pi_{\theta}}(s,a) = A^{\pi_{\theta}}(s,a) + V^{\pi_{\theta}}(s)$와 $Q^{\pi_{\theta}}$를 샘플링한 값과의 차이를 최소로 만든다.

$$\psi_t(v,w)$$
$$\equiv \|\sum_{k=0}^{t} z_k[(\widetilde{V}_v(s_k)+\widetilde{A_w}(s_k,a_k))-(r_k+\gamma\widetilde{V}_v(s_{k+1})]\|^2 \quad (18)$$
$$=\|\sum_{k=0}^{t} z_k\left[\phi^T(s_k)-\gamma\phi^T(s_{k+1}),\nabla_\theta\log\pi(a_k|s_k)^T\right]\begin{bmatrix}v\\w\end{bmatrix}-\sum_{k=0}^{t} z_k r_k\|^2$$

여기서, 누적기여도 z_k는 다음과 같이 정의되고,

$$z_k=\gamma\lambda z_{k-1}+\left[\phi^T(s_k),\nabla_\theta\log\pi(a_k|s_k)^T\right]^T,$$
$$z_0=\left[\phi^T(s_0),\nabla_\theta\log\pi(a_0|s_0)^T\right]^T$$

trace-decay 파라미터 λ는 [0, 1]사이의 범위를 가진다. (18)을 최소화시키는 것은 t 시간까지 에이전트와 환경간의 전체 이력(history)을 이용하는 least squares 문제일 뿐이다. 다만 최근의 결과값을 더 강조하기 위해서는 forgetting factor $\beta\in(0,1)$을 이용하여 (18)을 근사하면 다음과 구해진다.

$$\begin{bmatrix}v_t\\w_t\end{bmatrix}=\begin{bmatrix}v_{t-1}\\w_{t-1}\end{bmatrix} \quad (19)$$
$$+K_t(r_t-[\phi^T(s_t)-\gamma\phi^T(s_{t+1}),\nabla_\theta\log\pi(a_t|s_t)^T]\begin{bmatrix}v_{t-1}\\w_{t-1}\end{bmatrix})$$

위 식에서 구해진 w_t는 actor 부분을 업데이트하는데 이용된다.

3.3 Algorithm

위 방법론의 목표는 최적의 policy 분포 $\pi_\theta(a|s)$를 구하는 파라미터 벡터 θ_t와 $\widetilde{V}_v$와 $\widetilde{A_w}$를 근사하는데 필요한 파라미터 벡터 v_t와 w_t를 구하는 것이다. 초기 파라미터 vector $\theta=\theta_0$를 가진 policy $\pi_\theta(a|s)$, $\nabla_\theta\log\pi_\theta(a|s)$ 그리고 basis function $\phi(x)$가 주어지면 알고리즘은 다음과 같은 순서로 행해진다.

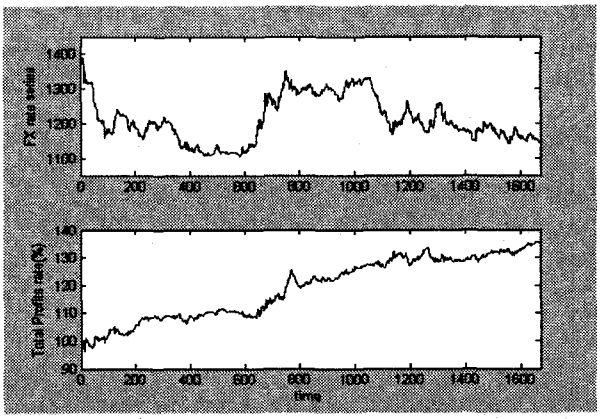
for t = 0, 1, 2, · · ·
1. $\pi_\theta(a|s)$에서 제어신호 a_t를 선택.
2. a_t를 행하고, r_t와 s_{t+1}를 관측.
3. (19)를 이용해서 v_t와 w_t를 구함.
4. policy 파라미터 업데이트.

$$\theta_{t+1}=\theta_t+\alpha w$$

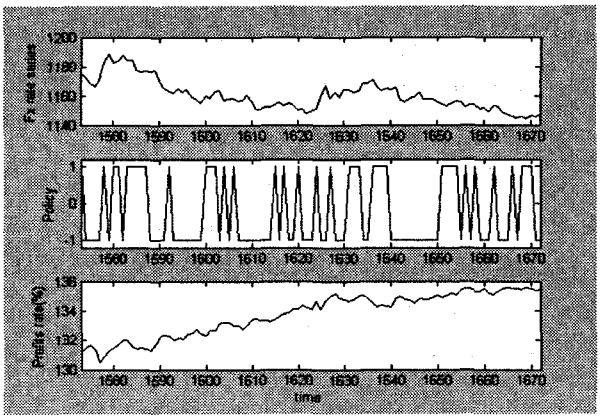
end

4. Experimental Results

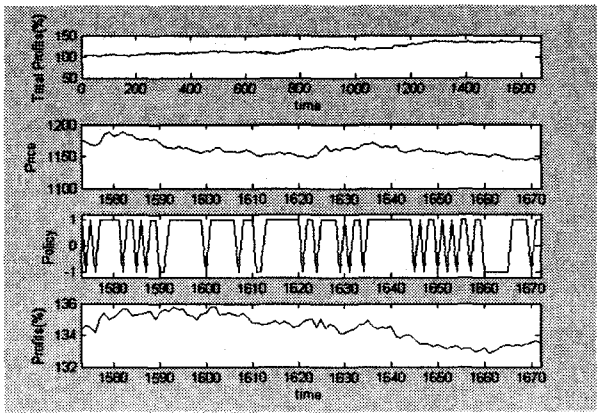
본 논문에서는 1998년 9월 23일부터 2004년 9월 22일까지의 원-달러 일일 환율 데이터에 대해 transaction cost 비율이 각각 0.2%, 0.5%, 1%인 경우를 가상하여 실험을 수행하였다. 시뮬레이션을 하기 위해 몇 가지 가정을 설정하였다. 첫째, 트레이딩을 통해서 얻는 수익은 원화로 계산된다. 둘째, 투자비가 전체 유동자금에 비해서 매우 작아서 개인의 트레이딩이 전체 환율 시장에 영향을 주지 않는다. 셋째, 투자자는 항상 자산 전체를 투자한다. 그리고 (1)의 $\phi^T\theta$ 부분은 [1]과 같이 $[F(t-1),r(t),\cdots,r(t-7),1]^T\theta$ 를 사용하였다.



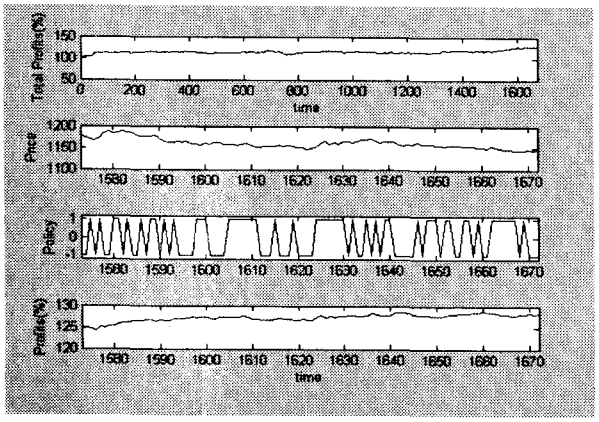
< 그림 1. Transaction cost 비율 = 0.5% >



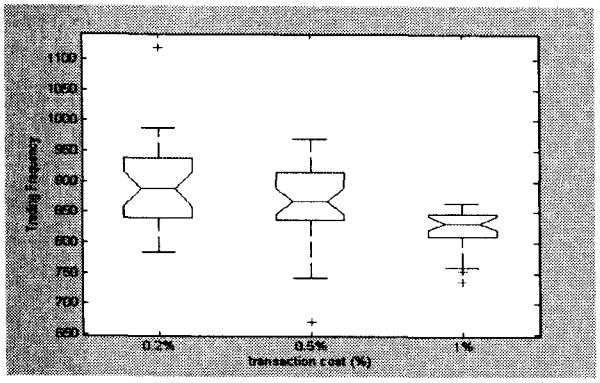
< 그림 2. Transaction cost 비율 = 0.5% >



< 그림 3. Transaction cost 비율 = 0.2% >



< 그림 4. Transaction cost 비율 = 1% >



< 그림 5. Trading Frequency >

그림 1의 상단 그래프는 6년간의 환율 변동 그래프이다. 그리고 하단 그래프는 transaction cost 비율이 0.5%일 때 논문의 알고리즘을 사용했을 때 얻게 되는 수익률 그래프이다. 그리고 그림 2, 3, 4는 각각 최종 100개의 time period를 policy와 profits에 대해서 확대한 결과 그래프이다. 마지막으로 그림 5는 transaction cost 비율이 변함에 따라서 trading 빈도수가 어떻게 변하는지를 나타내는 그림이다. 예상한대로 transaction cost 비율이 높을수록 트레이딩 빈도 수, 즉, position 변동 횟수가 줄어드는 것을 알 수 있다.

5. 결론 및 향후과제

본 논문에서는 RLS-based Natural Actor Critic 알고리즘을 이용해서 트레이딩 시스템을 학습하고, differential Sharpe ratio와 같은 평가 기준을 최적화시켰다. 결과 그래프에서도 나타났다시피 본 논문의 알고리즘은 예제로 다룬 문제에 대해 약 30% 정도의 높은 수익률을 보여주었다. 그리고 transaction cost 비율에 따르는 트레이딩 빈도의 변화에 대해서도 알아보았다. 본 논문에서는 적용 예로써, 원-달러 외환시장의 daily data에 대해 가상적인 transaction cost를 갖는 경우를 다루었지만, 향후에는 원-달러 외환시장의 half-hourly data에 대해 transaction cost가 bid/ask price에 따라 결정되는 현실적인 경우를 고려하는 것이 필요하다. 또, 이번 실험에서는 트레이딩 사이즈가 일정한 원-달러 환율에 대해서만 다루었지만 트레이딩 사이즈가 환율변동 크기에 따라서 달라지는 경우와 원-엔 환율, 원-유로 환율 등과 같은 여러 가지 환율을 하나의 포트폴리오로 형성해서 최대의 수익 올리는 것 등에 대해 다루는 것도 좋은 주제일 것이다.

6. 참고문헌

[1] R.S. Sutton and A.G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.

[2] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement learning for humanoid robotics," In *Proceedings of the Third* IEEE-RAS *International Conference on Humanoid Robots*, 2003.

[3] J. Park, J. Kim, D. Kang, "An RLS-based Natural Actor-Critic Algorithm for Locomotion of a Two-Linked Robot Arm," To appear in *Lecture Notes in Artificial Intelligence, vol. 3801, pp. 65-72, 2005.*

[4] R.S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in Neural Information Processing Systems*, vol. 12, pp. 1057-1063, 2000

[5] V. Konda and J.N. Tsitsiklis, "Actor-Critic Algorithms," *SIAM Journal on Control and Optimization*, vol. 42, no. 4, pp.1143-1166, 2003

[6] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation,* vol. 10, issue 2, pp.251-276, 1998

[7] J. Moody, M. Saffell, "Learning to Trade via Direct Reinforcement," *IEEE Transaction on Neural Networks*, vol. 12, no. 4, July 2001