Proceedings of KFIS Fall Conference, 2003. 5. 3

KFD 회귀를 이용한 뉴럴-큐 기법

Neural-Q method based on KFD regression

조원희, 김영일, 박주영
고려대학교 제어계측공학과

Won-Hee Cho, Yong-Il Kim, Jooyoung Park
Dept. of Control and Instrumentation Engineering, Korea University
E-mail : loadneo@hanmail.net / phone : 011-9098-8970

요 약

강화학습의 한가지 방법인 Q-learning은 최근에 Linear Quadratic Regulation(이하 LQR) 문제에 성공적으로 적용된 바 있다. 특히, 시스템 모델의 파라미터에 대한 구체적인 정보없이 적절한 입·출력만으로 학습을 통해 문제의 해결이 가능하므로 상황에 따라 매우 실용적인 방법이 될 수 있다. 뉴럴-큐 기법은 이러한 Q-learning의 Q-value를 MLP(multilayer perceptron) 신경망의 출력으로 대치시켜, 비선형 시스템의 최적제어 문제를 다룰 수 있게 한 방법이다. 그러나, 뉴럴-큐 기법은 신경망의 구조를 먼저 결정한 후 역전파 알고리즘을 이용해 학습하는 절차를 행하므로, 시행착오를 통해 신경망 구조를 결정해야 한다는 점, 역전파 알고리즘의 적용에 따라 신경망의 연결강도 값들이 지역적 최적해로 수렴한다는 점등의 문제점이 있다. 본 논문에서는 뉴럴-큐 학습의 도구로 KFD 회귀를 이용하여 Q 함수의 근사 기법을 제안하고 관련 수식을 유도하였다. 그리고, 모의 실험을 통하여, 제안된 뉴럴-큐 방법의 적용 가능성을 알아보았다.

키워드 : Q-learning, LQR, 뉴럴-큐, Kernel Fischer Discriminant, KFD 회귀

1. 서론

Q-learning은, 상태와 입력공간으로 이루어지는 공간에서 정의되는 이차형식(quadratic form)의 Q 함수를 이용하여 이산시간 시스템 $x_{k+1}=Ax_k+Bu_k$의 Linear Quadratic Regulation(이하 LQR) 제어 문제에 적용된 바 있다.[1] 이러한 시도는 행렬 A와 B를 알지 못하는 상태 즉, 시스템 파라미터를 구체적으로 알지 못하는 상태에서도 학습을 통해서 LQR 문제를 해결할 수 있는 있기 때문에 상당히 유용한 방법이 될 수 있다. 그리고, 이차형식의 Q 함수를 MLP 신경망으로 대치시켜 비선형 시스템에 대한 LQR 문제로 해결하는 뉴럴-큐 학습방법도 제안되었다.[2]
그러나 뉴럴-큐 방법은 시행착오를 통해 신경망의 구조를 결정해야하는 단점과 역전파 알고리즘을 통한 지역적 최적해로의 수렴과 같은 단점을 내포하고 있는 한계가 있다.

본 논문에서는 이러한 점을 개선하기 위해 Kernel Fisher Discriminant regression(KFD 회귀)을 사용하여 함수근사 단계를 수행하는 방안을 제안하였다.[5]
본 논문의 구성은 다음과 같다. 2장에서는 Neural-Q와 KFD 회귀를 소개하고 3장에서는 KFD 회귀를 이용한 뉴럴-큐 기법에 대해 기술한다. 4장에서는 제안된 방법의 유용성을 보이기 위해 예제를 보이고 이에 대한 시뮬레이션 및 결과를 서술한다.
마지막으로 5장에서는 결론 및 향후 연구결과에 대해서 논의한다.

2. Neural-Q 방법 및 KFD 회귀

2. 1 Q 함수

강화학습의 주요 주제 중 하나는 향후 총 비용의 기대값을 근사화 하고 그것을 최소화시키는 피드백을 찾는 것이다. 특히, Q-learning에서 피드백은 상태와

action의 함수로 미래의 비용(future cost)을 나타내는 Q 함수에서 구할 수 있다. 그리고 식 (1)를 통해 LQR에 대한 Q 함수는 상태와 action의 양한정 이차형식(positive definite quadratic function)임을 알 수 있다.

$$V^*(x) = \min_u Q^*(x,u) = Q^*(x,u^*)$$

$$Q^*(x_k^*, u_k^*) = \sum_{i=k}^{\infty} r_i$$

$$= \begin{bmatrix} x_k^T & u_k^{*T} \end{bmatrix} \begin{bmatrix} H_{xx}^* & H_{xu}^* \\ H_{ux}^* & H_{uu}^* \end{bmatrix} \begin{bmatrix} x_k \\ u_k^* \end{bmatrix}$$

$$= \psi_K^{*T} H^* \psi_K^* \qquad (1)$$

식(1)의 Q 함수는 양한정 이차형식이므로 이것은 제어입력에 대한 미분계수를 0으로 설정하는 것에 의해 각 상태에 대한 greedy action을 다음과 같이 구할 수 있다.

$$\nabla_{u_k^*} Q^*(x_k, u_k^*) = 2H_{ux}^* x_k + 2H_{uu}^* u_k^* = 0 \qquad (2)$$

$$u_k^* = -(H_{uu}^*)^{-1} H_{ux}^* x_k = L^* x_k$$

따라서 Q 함수의 파라미터 H에 의해 피드백의 이득값이 결정됨을 알 수 있다.[2]

2. 2 Q-learning(LQRQL)

Q^L의 변수는 feedback L에 의해 생성된 데이터를 기반으로 계산된 값이다.

함수 Q^L은 value function의 정의를 반복해 쓰는 것에 의해 구할 수 있다.

$$Q^L(x_k, u_k) = \sum_{i=k}^{\infty} r_i(x_{k+1}, Lx_{k+1})$$

$$= r_k + \sum_{i=k+1}^{\infty} r_i = r_k + Q^L \qquad (3)$$

식 (3)은 식 (1)에 의해 다음과 같은 형태로 변형이 가능하다.

$$r_k + Q^L(x_{k+1}, Lx_{k+1}) - Q^L(x_k, u_k) = 0$$

$$r_k = Q^L(x_k, u_k) - Q^L(x_{k+1}, Lx_{k+1})$$

$$= \Phi_k^T H^L \Phi_k - \Phi_{k+1}^T H^L \Phi_{k+1} \qquad (4)$$

양한정 이차형식의 파라미터에 대한 평가는 linear least squares estimation으로 표현될 수 있다. 예를 들어, scalar 상태 x와 scalar 제어입력 u를 취하면, 아래의 식과 같다.

$$Q(x,u) = \begin{bmatrix} x & u \end{bmatrix} \begin{bmatrix} h_{xx} & h_{xu} \\ h_{ux} & h_{uu} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

$$= \begin{bmatrix} h_{xx} & h_{xu} + h_{ux} & h_{uu} \end{bmatrix} \begin{bmatrix} x^2 \\ xu \\ u^2 \end{bmatrix}$$

$$= \theta^T \zeta = Q(\zeta) \qquad (5)$$

식 (5)에서와 같이 Q 함수는 θ(H)와 ζ의 곱으로 표현된다. 이후의 장에서는 Q 함수를 사용하는 LQR 문제를 LQRQL로 표기한다.

2. 3 LQRQL을 이용한 뉴럴-큐 방법

2. 3. 1 neural nonlinear Q-functions

Q-function으로 표현되는 함수는 선형활성함수와 입력 ξ을 가진 단층 전방향 신경망 구조로 나타낼 수 있다.

$$Q(\xi, w) = \Gamma_o(w_o^T \Gamma_h(W\xi) + b_h) + b_o \qquad (6)$$

여기에서 Γ_o, Γ_h은 활성함수를 나타내고, w_o, b_o은 출력노드의 연결강도행렬과 바이어스를 나타낸다. 행렬 W은 모든 은닉노드의 연결강도행렬 w_h와 모든 바이어스 b_h 를 포함한다. 이 신경망에는 1개의 output $Q(\xi, w)$이 있으며 ξ은 Q-함수의 입력벡터를 나타낸다. 기존의 이차형식의 Q-함수은 이러한 비선형 수식을 나타낼 수 없기 때문에 뉴럴-큐 방법에서는 은닉노드를 위하여 비선형 활성함수를 도입한다. 다음 식(7)은 은닉노드 i 의 출력을 나타내는 식이다. [2]

$$\Gamma_{h,i}(w_{h,i}^T \Omega_k^T + b_{h,i}) = \tanh(w_{h,i}^T \Omega_k^T + b_{h,i}) \qquad (7)$$

2. 3. 2. 비선형 피드백 함수의 유도

학습 과정에서 얻어지는 연결강도 및 바이어스 값들을 이용하면 $\hat{\theta}$의 값도 구할 수 있다. 즉,

$$\hat{\theta}_i(\Omega_K) = \frac{\delta Q(\Omega_K)}{\delta \Omega_i}$$

$$= \sum_{j=1}^{n_h} w_{o,j} w_{h,i,j} (1 - \tanh^2(w_{h,i}^T \Omega_k + b_{h,i})) \qquad (8)$$

에서, 선형 부분과 비선형 부분을 구분하면 다음과 같아진다:

$$\hat{\theta}_i = \sum_{j=1}^{n_h} w_{o,j} w_{h,i,j}$$

$$- \sum_{j=1}^{n_h} w_{o,j} w_{h,i,j} \tanh^2(w_{h,i}^T \Omega_k + b_{h,i}) \qquad (9)$$

식 (7)의 선형부분은 은닉노드가 선형일 때 계산이 되는 $\hat{\theta}_i$와 일치하며 이 신경망에서 나온 피드백 함수 또한 선형이다. 비선형 부분은 hyperbolic tangent transfer functions에 의한 비선형 효과이다. 선형 피드백은 비선형 부분을 무시한 상황으로부터 구해지며 다음과 같다.

$$\hat{\theta}_i = \sum_{j=1}^{n_h} w_{o,j} w_{h,i,j} \qquad (10)$$

식 (10)에서 vector $\hat{\theta}$은 $\hat{H}$로 재구성되고 식 (2)를 사용해 $\hat{L}$을 구할 수 있다. 여기에서 $\hat{H}$는 선형일 경우의 H값이고 $\hat{L}$는 선형일 경우의 feedback이다. $\tilde{u_k} = \hat{L} x_k$는 input $\hat{\Omega}$를 구하는 데 사용된다. $\hat{\Omega}$는 다음과 같이 x_k 과 u_k의 조합이다.

$$\hat{\Omega} = (x_k, u_k) = (x_k, \hat{L} x_k) \qquad (11)$$

식 (11)의 결과를 식 (8) 또는 식 (9)에 대입해 $\theta(\hat{\Omega}_k)_i$를 구할 수 있다. 그 결과를 이용해 $H(\hat{\Omega}_k)$를 구한다.

nonlinear feedback은 식 (2)에 $H(\widetilde{\Omega}_k)$를 대입하면 다음과 같이 구해진다.

$$u_k=(H_{uu}(\widetilde{\Omega}_k))^{-1}(H_{ux}(\widetilde{\Omega}_k))x_k=L(\widetilde{\Omega}_k)x_k \quad (12)$$

2. 4 KFD 회귀

2. 4. 1. Kernel Fisher Discriminant

KFD의 개념은 Fisher linear discriminant 문제를 kernel feature 공간 F에서 풀이하는 것이다. 따라서, 입력 공간 내에서 비선형 discriminant를 생성한다.

$\{x_i|i=1, ..., \ell\}$를 학습 데이터로, $y\in\{-1.\ 1\}^{\ell}$을 대응하는 lable의 vector라고 하자. 그리고 $1\in R^{\ell}$를 모든 1의 vector로, $1_1, 1_2\in R^{\ell}$을 class label에 해당하는 binary (0, 1) vector로 정의한다. τ과 τ_1, τ_2는 두 개의 class에 대한 index set이다.

선형의 경우, Fisher discriminant는 식 (13)을 최소화하는 것에 의해 계산이 된다.

$$J(w)=(w^TS_Bw)/(w^TS_Ww) \quad (13)$$

여기서 $S_B=(m_2-m_1)(m_2-m_1)^T$,

$$Sw=\sum_{k=1,2}\sum_{i=\tau_k}(x_i-m_k)(x_i-m_k)^T,$$

m_k은 class k에 대한 학습데이터의 평균이다.

kernel feature 공간 F에서 문제를 해결하려면, 내적을 조건으로 학습 데이터을 이용하는 공식이 필요하다.

해당하는 학습 유형에 대한 조건으로 전개하면 다음과 같이 쓸 수 있다.

$$w=\sum_{\tau}\alpha_i\Phi(x_i) \quad (14)$$

KFD에 대한 최적화 알고리즘은 다음과 같다.

$$J(\alpha)=\frac{(\alpha^T\mu)^2}{\alpha^TN\alpha}=\frac{\alpha^TM\alpha}{\alpha^TN\alpha} \quad (15)$$

$$\mu_i=\frac{1}{\ell_i}K1_i,\quad N=KK^T-\sum_{i=1,2}\ell_i\mu_i\mu_i^T,$$

$$\mu=\mu_2-\mu_1,$$

$$M=\mu\mu^T K_{ij}=(\phi(x_i)\cdot\phi(x_j))=k(x_i,x_j)$$

그리고 다음의 식 (16)에 의해 discriminant상의 test point의 projection을 계산한다. [4]

$$(\omega\cdot\phi(x))=\sum_{\tau}\alpha_iK(x_i,x) \quad (16)$$

2. 4. 2. KFD를 Quadratic problem으로 변환

다음은 식 (14)를 양한정 이차형식 문제로 재구성하는 과정이다.[5]

1단계. 행렬 M은 랭크가 1이다.

즉, $\alpha^TM\alpha=(\alpha^T(\mu_2-\mu_1))^2$

$(\alpha^T(\mu_2-\mu_1))^2$을 0이 아닌 값으로 고정

2단계. $\alpha^TN\alpha$을 최소화 시킨다.

이와 같은 과정의 결과는 다음의 양한정 이차형식 문제가 된다.

$$\min_{\alpha}\alpha^TN\alpha+CP(\alpha) \quad (17)$$
$$s.t\ \ \alpha^T(\mu_2-\mu_1)=2$$

여기서 C는 regularization term이다.

위의 식은 다음과 같은 양한정 이차형식 문제와 일치한다.

$$\min_{\alpha,b,\zeta}\|\zeta\|^2+CP(\alpha) \quad (18)$$
$$s.t\ \ K\alpha+1b=y+\zeta$$
$$1_i^T\zeta=0\ \ \text{for}\ i=1,2$$

식 (16)에서 regularization 부분을 1-norm으로 한정을 하면 다음과 같은 식을 가정할 수 있다.

$$\|\alpha_i\|\le a_i \quad (19)$$

a. $\|\alpha_i\|=|\alpha_1|+|\alpha_2|+\ldots+|\alpha_n|\le a_1+a_2\ldots+a_n$

b. $-a_i\le\alpha_i\le a_i i=1,2,\ldots,n$

식 (17)에 의해 식 (16)은 다음과 같이 $f(x)=K\alpha+1b$를 구하는 식으로 정리된다.

$$\min_{\alpha,b,\zeta}\|\zeta\|^2+C1^Ta \quad (20)$$
$$s.t\ \ K\alpha+1b=y+\zeta$$
$$1_i^T\zeta=0\ \ \text{for}\ i=1,2$$
$$-a_i+\alpha_i\le0,\ \ -a_i-\alpha_i\le0,\ \ -a_i\le0$$

3. KFD 회귀를 이용한 뉴럴-큐 기법

강화학습 이론에 따르면, Q 함수가 올바로 구해졌을 경우에는 $Q(z_i, w)$ 값은 $Q(\widehat{z_i}, w)+r_i$와 가까워야 한다(단, 여기에서 hat 기호는 다음 시간스텝을 의미한다). 따라서, z_i을 입력 ξ_i, $\widehat{z_i}$을 입력 ξ_{i+1}으로 정하고 큐 함수의 근사식을 위해 비선형 맵핑인 식 (19)을 도입하면

$$Q(z_k,\widehat{\theta},\omega)\cong\widehat{\theta}^Tz_k+\langle w,\psi(z_k)\rangle+b \quad (21)$$

관찰된 입력 및 상태벡터로부터 Q-함수를 추정하는 문제는 다음의 근사식 f를 사용해 시간 스텝이 k 인 경우의 순간 비용 r_k를 근사하는 문제가 된다.

$$f(z_k,\widehat{z_k},\widetilde{\theta},\omega) \quad (22)$$
$$=Q(z_k,\widetilde{\theta},w)-Q(\widehat{z_k},\widetilde{\theta},\omega)$$
$$=[\widetilde{\theta}^Tz_k+\langle w,\psi(z_k)\rangle+b]-[\widetilde{\theta}^T\widehat{z_k}+\langle w,\psi(\widehat{z_k})\rangle+b]$$
$$=\widetilde{\theta}^T(z_k-\widehat{z_k})+\langle w,\psi(z_k)-\psi(\widehat{z_k})\rangle$$

여기에서 근사식 f는 상수항을 갖지 않음에 주의한다. 주어진 학습 데이터 $\{(z_i,\widehat{z_i}),r_i\}_{i=1}^m$에 대해 위의 근사를 성공적으로 수행할 수 있는 충분히 매끄러운 함수를 찾는 문제는, 다음과 같은 최적화 문제로 표현될 수 있다.

식 (20)에서

$$f(z_k, \hat{z}_k, \tilde{\theta}, \omega) = \tilde{\theta}^T(z_k - \hat{z}_k) + \langle w, \psi(z_k) - \psi(\hat{z}_k) \rangle$$
$$= \tilde{\theta}^T(z_k - \hat{z}_k) + w \cdot (\psi(z_k) - \psi(\hat{z}_k))$$
$$= \tilde{\theta}^T(z_k - \hat{z}_k) + K\alpha - \hat{K}\alpha$$

$$f(z_k, \hat{z}_k, \tilde{\theta}, \omega) - \tilde{\theta}^T(z_k - \hat{z}_k) = K\alpha - \hat{K}\alpha \quad (23)$$

식 (23)에서 좌변은 식 (18)의 y, 우변은 $K\alpha$에 상응한다고 볼 수 있다. 바이어스 b는 소거가 된다.

식 (20)은 다음과 같이 변형된다.

$$\min_{\alpha, b, \zeta} \|\zeta\|^2 + C1^T a \quad (24)$$
$$s.t \ (K - \hat{K})\alpha = y, \quad 1_i^T \zeta = 0 \quad \text{for} \ i = 1,2$$
$$-a_i + \alpha_i \le 0, \quad -a_i - \alpha_i \le 0, \quad -a_i \le 0$$

위의 문제는 전형적인 QP(quadratic programming) 문제이므로, 일반적인 QP solver를 이용하면 쉽게 풀 수 있다.

4. 모의 실험 결과

본 논문에서 제안한 방법을 평가하기 위해 참고문헌 [2]에서 제기된 바 있는 간단한 이동로봇 제어문제를 고려해 보았다. 이산시간에 대한 로봇의 상태 방정식은 식 (25)와 같다.

$$x_{k+1} = x_k + \frac{v_t}{\omega}(\sin(\psi_k + T\omega) - \sin(\psi_k))$$
$$\psi_{k+1} = \psi_k + T\omega \quad (25)$$

(x는 로봇의 위치, ψ는 로봇의 방향을 나타내는 변수, T는 고정된 샘플 시간간격이다.)

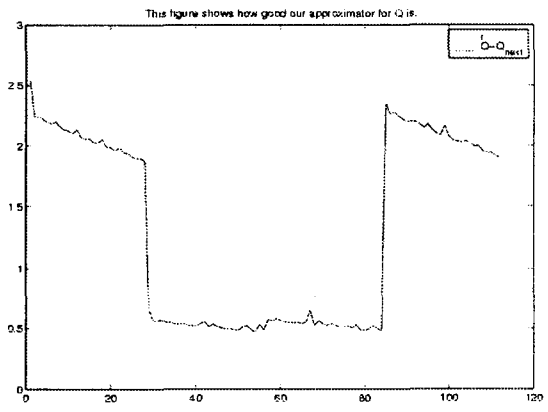

그림 1 KFD 회귀 결과

Q 함수에 대해 선형 근사를 수행한 후 오차부분에 대하여 KFD 회귀를 통해 비선형 근사를 수행한 결과, 그림 1과 같은 결과가 얻어졌다. 그리고, 위의 과정을 통하여 얻어진 비선형 Q 함수를 기반으로 식 (10)의 비선형 제어기를 구한 후 서로 다른 초기조건에 대하여 각각 적용한 결과 그림 2와 같은 결과가 얻어졌다.

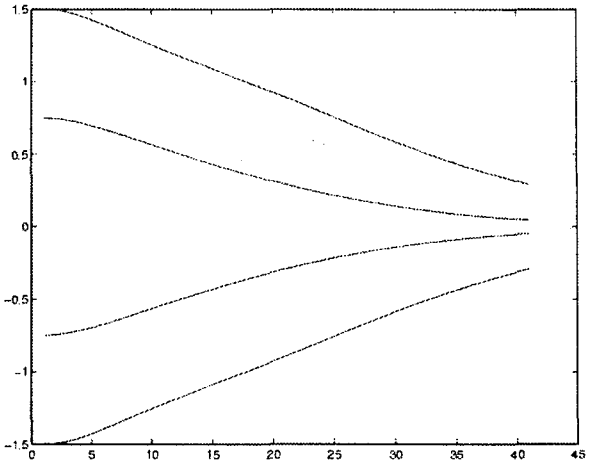

그림 2 제안한 방법으로 설계된 비선형 제어기를 로봇에 적용한 결과

5. 결 론

본 논문에서는 강화학습이 최적제어 문제에 성공적으로 적용될 수 있음과 KFD 회귀가 함수근사에 효과적으로 적용될 수 있음을 바탕으로, KFD 회귀방법을 이용한 neural-Q 기법을 제안하고 관련 수식을 유도하였다. 그리고, 유도된 비선형 최적제어 방법론을 이동로봇에 대해 적용한 결과 Q 함수의 근사가 성공적으로 이루어질 수 있음과, 이를 이용한 비선형 LQR 제어기의 유도가 가능함을 확인하였다. 관련하여 생각하여 볼 수 있는 향후과제로는, 여러 가지 예제에 대한 광범위한 시뮬레이션 수행을 통하여 확립된 방법의 장단점을 파악하고, 여타 기법과의 성능을 비교해 보는 문제 등을 들 수 있다.

6. 참고문헌

[1] S.J. Bradtke, B.E. Ydstie, and A.G. Barto, "Adaptive linear quadratic control using policy iteration," In American Control Conference, pp. 3475-3479, 1994.

[2] S.H.G. Hagen, Continuous state space Q-learning for control of nonlinear systems, PhD Thesis, Computer Science Institute, University of Amsterdam, The Netherlands, February 2001.

[3] B. Scholkopf and A.J. Smola, Learning with kernels: Support vector machines, regularization, optimization, and beyond, MIT Press, 2002.

[4] S. Mika, Fisher discriminant analysis with kernels, Neural Networks for Signal Processing IX, pages 41-48. IEEE, 1999.

[5] S. Mika, A mathematical programming approach to the kernel Fisher algorithm. Advances in Neural Information Processing Systems 13, pages 591-597. MIT Press, 2001