Notes by: Anubhav Singh

Corrected KernelUCB algorithm

October 8, 2018

We chose RBF kernel for our tests.

Using the following definitions (clears up inconsistencies):

K(t) is Kernel Matrix at time t.

Gaussian probability distribution:-

1. mu(a, t) is the mean
2. sigma(a, t) is the standard deviation

- $k_{x^\star,t} = [k(x^\star, x_1), \dots k(x^\star, x_t)]^\intercal$
- $y_t = [r_1, \dots, r_t]^\intercal$
- $\Phi_t = [\phi(x_1)^\intercal, \dots, \phi(x_t)^\intercal]^\intercal$
- $K_t = \Phi_t^\intercal \Phi_t$
- $\hat{\sigma}_{a,t+1} = [\phi(x_{a,t+1})^\intercal(\Phi_t^\intercal\Phi_t + \gamma I)^{-1}\phi(x_{a,t+1})]^{\frac{1}{2}}$
- $\hat{\mu}_{a,t+1} = k^\intercal_{x_{a,t+1},t}(K_t + \gamma I)^{-1} y_t$

-Here phi(x) is the vector transformation of the context vector x, with kernel funciton k(x,y) = phi(x).phi(y).

Wrong! Swap the matrices

In mu (mean) formula, we are assuming that phi_upper is square matrix, So, left inverse of phi_upper = right inverse of phi_upper. Hence, inverse([phi_upper_tr][phi_upper])([phi_upper_tr]) = [phi_upper_tr][K_inv].
This enables use of Kernel matrix phi.[phi_upper] = transpose[k_x(a,t+1)]

Algorithm 1 KernelUCB with online updates

Input: N the number of actions, T the number of pulls, γ, η regularization and exploration parameters, $k(\cdot,\cdot)$ kernel function

for $t \in 1, \dots, T$ **do**
    Receive contexts $\{x_{1,t}, \dots, x_{N,t}\}$
    if $t = 1$ **then**
        $u_t \leftarrow [1, 0, \dots, 0]^\intercal$
    else
        for $n \in \{1, \dots N\}$ **do**
            $\sigma_{n,t} \leftarrow \left[k(x_{n,t}, x_{n,t}) - k^\intercal_{x_{n,t},t-1} K^{-1}_{t-1} k_{x_{n,t},t-1}\right]^{\frac{1}{2}}$
            $u_{n,t} \leftarrow k^\intercal_{x_{n,t},t-1} K^{-1}_{t-1} y_{t-1} + \frac{\eta}{\sqrt{\gamma}} \sigma_{n,t}$
        end for
    end if
    Choose action $a_t \leftarrow \arg\max u_t$ and receive reward r_t
    Store context for action a_t: $x_t \leftarrow x_{a,t}$
    Update reward history: $y_t \leftarrow [r_1, \dots, r_t]^\intercal$
    if $t = 1$ **then** ▷ initialise kernel matrix inverse
        $K_t^{-1} \leftarrow (k(x_t, x_t) + \gamma)^{-1}$
    else ▷ online update of kernel matrix inverse
        $b \leftarrow k_{x_t,t-1}$
        $K_{22} \leftarrow \left(k(x_t, x_t) + \gamma - b^\intercal K^{-1}_{t-1} b\right)^{-1}$
        $K_{11} \leftarrow K^{-1}_{t-1} + K_{22} K^{-1}_{t-1} b b^\intercal K^{-1}_{t-1}$
        $K_{12} \leftarrow -K_{22} K^{-1}_{t-1} b$
        $K_{21} \leftarrow -K_{22} b^\intercal K^{-1}_{t-1}$
        $K_t^{-1} \leftarrow \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$
    end if
end for

These can be proved to be equal using Woolbury matrix identity.
** proof included on the next page

Proof of Standard Deviation formula, in Kernel-UCB algorithm, using Woodbury Matrix Identity

Anubhav Singh

January 30, 2019

Proof

According to the Woodbury Matrix Identity:

$$(\mathbf{A}+\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{D}+\mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \tag{1}$$

Now, assign the following to matrices A, B, C and D

$$\mathbf{A}=\gamma\mathbf{I_d},\ \mathbf{B}=\mathbf{\Phi^T_{dxt}},\ \mathbf{C}=\mathbf{\Phi_{txd}},\ \mathbf{D}=\mathbf{I_t} \tag{2}$$

Using equation (1),

$$\begin{aligned}(\gamma\mathbf{I_d}+\mathbf{\Phi^T\Phi})^{-1}&=(\gamma\mathbf{I_d}+\mathbf{\Phi^T I_t^{-1}\Phi})^{-1}\\&=\gamma^{-1}\mathbf{I_d^{-1}}-\gamma^{-1}\mathbf{I_d^{-1}\Phi^T}(\mathbf{I_t}+\mathbf{\Phi}(\gamma^{-1}\mathbf{I_d^{-1}})\mathbf{\Phi^T})^{-1}\mathbf{\Phi}\gamma^{-1}\mathbf{I_d^{-1}}\\&=\gamma^{-1}\mathbf{I_d}-\gamma^{-1}\mathbf{I_d\Phi^T}(\mathbf{I_t}+\mathbf{\Phi}(\gamma^{-1}\mathbf{I_d})\mathbf{\Phi^T})^{-1}\mathbf{\Phi}\gamma^{-1}\mathbf{I_d}\\&=\gamma^{-1}\mathbf{I_d}-\gamma^{-2}\mathbf{\Phi^T}\gamma(\gamma\mathbf{I_t}+\mathbf{\Phi\Phi^T})^{-1}\mathbf{\Phi}\\&=\gamma^{-1}(\mathbf{I_d}-\mathbf{\Phi^T}(\gamma\mathbf{I_t}+\mathbf{\Phi\Phi^T})^{-1}\mathbf{\Phi})\\&=\gamma^{-1}(\mathbf{I_d}-\mathbf{\Phi^T}(\gamma\mathbf{I_t}+\mathbf{K})^{-1}\mathbf{\Phi})\end{aligned} \tag{3}$$

Applying the results obtained in (3) to the Standard Deviation Formula, we get the formula used in Algorithm-1

$$\begin{aligned}\sigma_{\mathbf{n,t}}&=[\phi^{\mathbf{T}}_{\mathbf{x_{n,t}}}(\gamma\mathbf{I}+\mathbf{\Phi^T_{t-1}\Phi_{t-1}})^{-1}_{\mathbf{t-1}}\phi_{\mathbf{x_{n,t}}}]^{1/2}\\&=[\gamma^{-1}\phi^{\mathbf{T}}_{\mathbf{x_{n,t}}}(\mathbf{I}-\mathbf{\Phi^T_{t-1}}(\gamma\mathbf{I}+\mathbf{K_{t-1}})^{-1}\mathbf{\Phi_{t-1}})\phi_{\mathbf{x_{n,t}}}]^{1/2}\\&=[\gamma^{-1}\{(\phi^{\mathbf{T}}_{\mathbf{x_{n,t}}}\phi_{\mathbf{x_{n,t}}})-(\phi^{\mathbf{T}}_{\mathbf{x_{n,t}}}\mathbf{\Phi^T_{t-1}})(\gamma\mathbf{I}+\mathbf{K_{t-1}})^{-1}(\mathbf{\Phi_{t-1}}\phi_{\mathbf{x_{n,t}}})\}]^{1/2}\\&=\gamma^{-1/2}[\mathbf{k}(\mathbf{x_{n,t}},\mathbf{x_{n,t}})-\mathbf{k}^{\mathbf{T}}_{\mathbf{x_{n,t},t-1}}(\gamma\mathbf{I}+\mathbf{K_{t-1}})^{-1}\mathbf{k_{x_{n,t},t-1}}]^{1/2}\end{aligned} \tag{4}$$