# Derivation of TRPO

1.The TRPO (trust region policy optimization) algorithm tries to optimizing:

$$\max \ \mathscr{L}(\theta, \theta_{old})$$

$$s.t. \ D_{KL}(\theta || \theta_{old}) \leq \delta$$

2.Perform first order approximation to $\mathscr{L}(\theta, \theta_{old})$:

$$\begin{aligned}\mathscr{L}(\theta, \theta_{old}) &\simeq \mathscr{L}(\theta_{old}, \theta_{old}) + \nabla_\theta \mathscr{L}(\theta, \theta_{old})^T |_{\theta=\theta_{old}} (\theta - \theta_{old}) \\ &= \nabla_\theta \mathscr{L}(\theta, \theta_{old})^T |_{\theta=\theta_{old}} (\theta - \theta_{old}) \\ &= g^T(\theta - \theta_{old}) \qquad \text{substitution}\end{aligned}$$

3.Perform second order approximation to $D_{KL}(\theta || \theta_{old})$:

$$\begin{aligned}D_{KL}(\theta || \theta_{old}) &\simeq D_{KL}(\theta_{old} || \theta_{old}) + \nabla_\theta D_{KL}(\theta || \theta_{old}) |_{\theta=\theta_{old}} (\theta - \theta_{old}) + \frac{1}{2}(\theta - \theta_{old})^T \nabla_\theta^2 D_{KL}(\theta || \theta_{old}) |_{\theta=\theta_{old}} (\theta - \theta_{old}) \\ &= \frac{1}{2}(\theta - \theta_{old})^T \nabla_\theta^2 D_{KL}(\theta || \theta_{old}) |_{\theta=\theta_{old}} (\theta - \theta_{old}) \\ &= \frac{1}{2}(\theta - \theta_{old})^T) H (\theta - \theta_{old}) \qquad \text{substitution}\end{aligned}$$

4.Reformulate original problem:

$$\max \ g^T(\theta - \theta_{old})$$

$$s.t. \ \frac{1}{2}(\theta - \theta_{old})^T H (\theta - \theta_{old}) \leq \delta$$

5.Applying Lagrange Multiplier Method:

Define Lagrange function:

$$\mathscr{L}(\theta, \lambda) = -g^T(\theta - \theta_{old}) + \lambda \left( \frac{1}{2}(\theta - \theta_{old})^T H (\theta - \theta_{old}) - \delta \right).$$

The solution should satisfy the K.K.T conditions:

$$
\begin{cases}
\nabla_\theta \mathscr{L}(\theta,\lambda) = 0; \Rightarrow -g + \lambda H(\theta - \theta_{old}) = 0 \\
\lambda \geq 0; \\
\lambda \left( \frac{1}{2}(\theta - \theta_{old})^T H(\theta - \theta_{old}) - \delta \right) = 0. \Rightarrow \lambda = 0 \vee \frac{1}{2}(\theta - \theta_{old})^T H(\theta - \theta_{old}) - \delta = 0
\end{cases}
$$

It's trivial that $\lambda \neq 0$ (otherwise $g = \mathbf{0}$, which introduces contradiction).

Combining K.K.T conditions, obtaining:

$$(\theta - \theta_{old}) = \frac{1}{\lambda} H^{-1} g$$

$$\frac{1}{2}(\frac{1}{\lambda}H^{-1}g)^T \frac{1}{\lambda} g = \delta$$

$$\Leftrightarrow \frac{1}{\lambda^2} g^T (H^{-1})^T g = 2\delta$$

$$\Leftrightarrow \frac{1}{\lambda^2} g^T H^{-1} g = 2\delta \qquad \text{(symmetry of Hessian matrix)}$$

$$\Leftrightarrow \frac{1}{\lambda} = \sqrt{\frac{2\delta}{g^T H^{-1} g}}$$

The iterative equation turns out to be:

$$\theta = \theta_{old} + \sqrt{\frac{2\delta}{g^T H^{-1} g}} H^{-1} g$$

## 6.Optimizing with Vector-Product strategy

We notice the computation of $H^{-1}$ can be very expensive and occupied when the matrix become large, however, computation of $H^{-1}g$ can be much easier making use of **CG**(conjugate gradient descent) method.For detail of CG, referring 《Numerical Optimization》should be helpful.