Lasso における ADMM の最適化手法

武山尚生

2020 年 3 月 10 日

1　拡張ラグランジュ法

拡張ラグランジュ法は，ラグランジュ法とペナルティ法を組み合わせた問題を最適化するための手法で，ADMM は拡張ラグランジュ法に改良を加えたものである．まずは，拡張ラグランジュ法がどのように成り立っているかを説明するため，元になっているラグランジュ法から説明をする．ラグランジュ法では以下の等式制約付き最適化問題を解く．

$$\min_{\mathbf{x}}\ f(\mathbf{x})\ \text{s.t.}\ c_i(\mathbf{x}) = 0 \tag{1}$$

この場合，新しいコスト関数を以下のように定義し，x と h_i について偏微分することで，コスト関数が最小化される点を探す．

$$L(\mathbf{x};\lambda) = f(\mathbf{x}) + \sum_{i=1}^{m} h_i c_i(\mathbf{x}) \tag{2}$$

$L(\mathbf{x};\lambda)$ が最小になる点が極値であるならば，$\frac{d\mathbf{L}}{d\mathbf{x}} = \frac{d\mathbf{L}}{d\lambda} = 0$ を満たす．

次に，ペナルティ法の説明をする．ペナルティ法も同様に等式制約付き最適化問題を解く手法の一つである．ペナルティ法はラグランジュ法とは異なり，制約条件の値の 2 乗を罰金項として課し，偏微分することでコスト関数が最小化される点を探す．

$$L_{\text{pen.}}(\mathbf{x}) = f(\mathbf{x}) + \frac{\mu[t]}{2}\sum_{i=1}^{m} c_i^2(\mathbf{x}) \tag{3}$$

この方法では，ラグランジュ法による連立方程式が解けない場合，勾配法を用いて目的の値を探索することが可能だが，罰金項の係数 $\mu[t]$ が非常に大きな値にならないと最適解に収束しないことが分かっており，計算に非常に時間がかかる．

拡張ラグランジュ法は，この 2 つの手法を組み合わせた手法であり，以下の関数を最適化する．

$$L_{\text{aug.}}(\mathbf{x};\boldsymbol{h}) = f(\mathbf{x}) + \sum_{i=1}^{m} h_i[t]c_i(\mathbf{x}) + \frac{\mu}{2}\sum_{i=1}^{m} c_i^2(\mathbf{x}) \tag{4}$$

これは，μ の値を固定し，h_i を $h_i[t+1] = h_i[t] + \mu c_i(\mathbf{x})$ と更新する．罰金法でのコストを，ラグランジュの未定乗数で制約条件を満たすようにすることが目的である．この時，h_i は有限の値で

ある μ でも収束するため，罰金法よりも計算量を抑えることが出来る．また，拡張ラグランジュ法におけるコスト関数は以下のように平方完成が出来る．

$$L_{\text{aug.}}(\mathbf{x};\boldsymbol{h}) = f(\mathbf{x}) + \frac{\mu}{2}\sum_{i=1}^{m}\left(c_i(\mathbf{x}) + \frac{h_i[t]}{\mu}\right)^2 \tag{5}$$

$u_i[t] = h_i[t]/\mu$ とすると，以下の式が得られる．

$$L_{\text{aug.}}(\mathbf{x};\boldsymbol{h}) = f(\mathbf{x}) + \frac{\mu}{2}\sum_{i=1}^{m}(c_i(\mathbf{x}) + u_i[t])^2 \tag{6}$$

この時の $u_i[t]$ の更新則は $u_i[t+1] = u_i[t] + c_i(\mathbf{x})$ で表され，制約条件の値が $u_i[t]$ に加算されることで $L_{\text{aug.}}$ の補正が行われる．

以上が拡張ラグランジュ法の基本的な流れになる．具体例として，Lasso に拡張ラグランジュ法を適用した場合の例を挙げる．Lasso において解くべき問題は以下の通りである．

$$\min_{\mathbf{x}} \ |\mathbf{x}| \ \text{s.t.} \ \mathbf{y} - \mathbf{A}\mathbf{x} = 0 \tag{7}$$

拡張ラグランジュ法より，以下のコスト関数の最小化問題に帰着出来る．

$$L_{\text{arg.}}(\mathbf{x};\boldsymbol{h}[t],\mu[t]) = \|\mathbf{x}\|_1 + (\boldsymbol{h}[t])^{\mathrm{T}}(\mathbf{y} - A\mathbf{x}) + \frac{\mu}{2}\|\mathbf{y} - A\mathbf{x}\|_2^2 \tag{8}$$

ラグランジュの未定乗数の更新則は

$$\boldsymbol{h}[t+1] = \boldsymbol{h}[t] + \mu(\mathbf{y} - A\mathbf{x}) \tag{9}$$

である．両辺に罰則項の微分係数である A^T を掛けて，$\mu = 1/\lambda$ と置いた時，

$$\mathbf{p}[t+1] = \mathbf{p}[t] + \frac{1}{L\lambda}A^{\mathrm{T}}(\mathbf{y} - A\mathbf{x}) \tag{10}$$

となる．これは ISTA にも見られた通り，近接写像の極値を求めるアルゴリズムと同じである．
p を最適化することで，コスト関数 $L_{\text{arg.}}$ を最適化出来る．以上が拡張ラグランジュ法の例である．

2 ADMM

ADMM は拡張ラグランジュ法に改良を加えた手法で，2 つのコスト関数の最小化問題である

$$\min_{\mathbf{x}}\{f(\mathbf{x}) + g(\mathbf{x})\} \tag{11}$$

を，制約付き最適化問題である

$$\min_{\mathbf{x},\mathbf{z}}\{f(\mathbf{z}) + g(\mathbf{x})\} \quad \text{s.t.} \quad \mathbf{x} - \mathbf{z} = 0 \tag{12}$$

に置き換えて解く手法である．この場合も，拡張ラグランジュ法を適用して

$$L_{\text{aug.}}(\mathbf{x},\mathbf{z};\mathbf{h}[t]) = f(\mathbf{z}) + g(\mathbf{x}) + (\mathbf{h}[t])^{\mathrm{T}}(\mathbf{x} - \mathbf{z}) + \frac{\mu}{2}\|\mathbf{x} - \mathbf{z}\|_2^2 \tag{13}$$

とコスト関数を設定する．平方完成すると，

$$L_{\mathrm{aug.}}(\mathbf{x},\mathbf{z};\mathbf{h}[t]) = f(\mathbf{z}) + g(\mathbf{x}) + \frac{\mu}{2}\left\|\mathbf{x}-\mathbf{z}+\frac{\mathbf{h}[t]}{\mu}\right\|_2^2 \tag{14}$$

と表せる．ADMM では，コスト関数を f(z) と g(x) に分離してそれぞれの最小化問題を解くことで，$L_{\mathrm{aug.}}\ (\mathbf{x},\mathbf{z};\mathbf{h}[t])$ の最適化を行う．ADMM では，x と z を以下のように更新する．

$$\mathbf{x}[t+1] = \mathrm{argmin}_{\mathbf{x}}\left\{g(\mathbf{x}) + \frac{\mu}{2}\left\|\mathbf{x}-\mathbf{z}[t]+\frac{\mathbf{h}[t]}{\mu}\right\|_2^2\right\} \tag{15}$$

$$\mathbf{z}[t+1] = \mathrm{argmin}_{\mathbf{x}}\left\{f(\mathbf{z}) + \frac{\mu}{2}\left\|\mathbf{x}[t+1]-\mathbf{z}+\frac{\mathbf{h}[t]}{\mu}\right\|_2^2\right\} \tag{16}$$

但し，ラグランジュの未定乗数は以下のように更新する．

$$\mathrm{h}[t+1] = \mathrm{h}[t] + \mu(\mathrm{x}-\mathrm{z}) \tag{17}$$

x についての最小化問題は以下の更新式で求められる．

$$\mathbf{x}[t+1] = \left(\mu I + \frac{1}{\lambda}A^{\mathrm{T}}A\right)^{-1}\left(\frac{1}{\lambda}A^{\mathrm{T}}\mathbf{y} + \mu(\mathbf{z}-\mathbf{u})\right) \tag{18}$$

また，z は軟判別閾値関数を用いて更新式を得られる．

$$\mathbf{z}[t+1] = S_{1/\mu}(\mathbf{x}[t+1] + \mathbf{u}[t] \text{ と} \tag{19}$$

この更新式を交互に解くことで，コスト関数を最小化出来る．Coordinate descent 法や FISTA とは異なり，近接写像を用いずに直接最適解を求める更新式となっているため，逆行列の計算が一回必要ではあるものの，極めて収束が早いという特徴がある．以上が ADMM の説明となる．