ADMM Tutorial

赖泽强

2020 年 12 月 22 日

目录

1 Basic

ADMM 是针对优化问题的一种解法。优化问题则是说，我们希望在给定一些约束的情况下，去寻找一个解来最小化一个我们定义的目标函数。这个过程可以形式化地定义为：

$$P : \min_{x \in D} f(x)$$

其中 f 是目标函数，D 是由约束条件划定的 x 的取值范围。

ADMM 尝试解决的则是一种特殊的优化问题。在这个问题中，我们的目标函数是一个均方误差，约束条件则是一个 L1 范数。我们使用朗格朗日法将有约束问题转化为无约束问题，就变成了公式1所示的形式。

$$\min_x \frac{1}{2} \sum_{i=1}^{n} (y_i - x_i)^2 + \lambda \sum_{(i,j) \in E} |x_i - x_j| \tag{1}$$

通常我们将这个问题称为 **2d fused lasso** 或 **2d total variation denoising** 问题

我们可以使用各种各样的优化算法来解这个特殊的问题，例如 Proximal gradient descent，Coordinate descent，但 ADMM 是这些算法中收敛最快的（即需要迭代次数少）[1]。

1.1 Vanilla ADMM

那么 ADMM 是怎么做的呢？ADMM 先是引入了一个新的变量 v，并约束 $x = v$，然后解公式1所示的无约束问题，就变成了解公式2所示的有约束问题。

$$(\widehat{\boldsymbol{x}}, \widehat{\boldsymbol{v}}) = \underset{\boldsymbol{x},\boldsymbol{v}}{\operatorname{argmin}} \quad f(\boldsymbol{x}) + \lambda g(\boldsymbol{v}), \quad \text{subject to } \boldsymbol{x} = \boldsymbol{v} \tag{2}$$

[1]需要注意的是，ADMM 在这个问题上快，不代表它在其它问题上也快。

然后我们用增广朗格朗日法再将其变成无约束问题，变成公式10的形式[2]。

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{u}) = f(\boldsymbol{x}) + \lambda g(\boldsymbol{v}) + \boldsymbol{u}^T(\boldsymbol{x} - \boldsymbol{v}) + \frac{\rho}{2}\|\boldsymbol{x} - \boldsymbol{v}\|^2 \tag{3}$$

优化这个方程可以使用分步优化的方法，即先选取一个优化变量，然后固定其它变量，对刚刚选取的变量进行优化，依次选取所有需要优化的变量重复进行。这个过程可以用公式4描述。

$$\begin{aligned}
\boldsymbol{x}^{(k+1)} &= \underset{\boldsymbol{x}\in\mathbb{R}^n}{\operatorname{argmin}} \quad f(\boldsymbol{x}) + \tfrac{\rho}{2}\left\|\boldsymbol{x} - \tilde{\boldsymbol{x}}^{(k)}\right\|^2 \\
\boldsymbol{v}^{(k+1)} &= \underset{\boldsymbol{v}\in\mathbb{R}^n}{\operatorname{argmin}} \quad \lambda g(\boldsymbol{v}) + \tfrac{\rho}{2}\left\|\boldsymbol{v} - \widetilde{\boldsymbol{v}}^{(k)}\right\|^2 \\
\overline{\boldsymbol{u}}^{(k+1)} &= \overline{\boldsymbol{u}}^{(k)} + \left(\boldsymbol{x}^{(k+1)} - \boldsymbol{v}^{(k+1)}\right)
\end{aligned} \tag{4}$$

其中第三个优化 u 的式子，我们是要让 u 最大，用这种方式强迫 x 和 v 更接近。然后因为我们求导有解析解，我们可以直接使用梯度上升法。

至于 ADMM 这种做法为什么会获得更快的收敛速度，我还没有深入研究。

1.2 Plug-and-Play ADMM

对于公式4里的第二个式子，定义 $\sigma = \sqrt{\lambda\rho}$，我们可以将其改写成：

$$\boldsymbol{v}^{(k+1)} = \underset{\boldsymbol{v}\in\mathbb{R}^n}{\operatorname{argmin}} \quad g(\boldsymbol{v}) + \frac{1}{2\sigma^2}\left\|\boldsymbol{v} - \tilde{\boldsymbol{v}}^{(k)}\right\|^2 \tag{5}$$

直观的，我们可以把这个优化过程看出一个降噪的过程，其中 σ 是高斯噪声的强度 (我们假设噪声是高斯噪声)。我们可以把 v 当成降噪后的图像，v^k 看出带噪声的图像。$g(v)$ 是说我们降噪后的图像要是一个图像（满足先验 g)，后面一项则是说降噪后的图像和原图像要接近。

因此，我们可以使用一个降噪器去替代这个优化过程。每一步优化，我们都将 v^k 输入一个降噪器获得 v^{k+1}。

具体为什么说这个形式很像降噪，我们需要先回顾一下降噪的优化算法是什么样的。

对于一个降噪问题，我们形式化为如下的优化问题：

$$\widehat{\boldsymbol{x}}_{\mathrm{map}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x} \mid \boldsymbol{y}) \tag{6}$$

其中 y 是输入的噪声图像，x 是去噪图像。

使用贝叶斯公式，加负号，我们可以将其转换成如下的优化问题：

$$\widehat{\boldsymbol{x}}_{\mathrm{map}} = \arg\min_{\boldsymbol{x}}\{-\ln p(\boldsymbol{y} \mid \boldsymbol{x}) - \ln p(\boldsymbol{x})\} \tag{7}$$

如果我们假设噪声是高斯噪声，那么 $e = y - x$ 应该服从正态分布, 即 $\boldsymbol{e} \sim \mathcal{N}\left(\mathbf{0}, \sigma_e^2\boldsymbol{I}\right)$。

因为 $p(y|x)$ 是给定原始图像，噪声图像出现的概率，既然我们知道噪声的概率分布，那 $p(y|x)$ 事实上应该就是噪声出现的概率。因此我们可以将正态分布的公式代入，求解出 $-lnp(y|x)$：

$$-logP(y|x) = -log(\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(y-x)^2}{2\sigma}}) = \frac{(y-x)^2}{2\sigma} + log(\sigma\sqrt{2\pi}) \propto \frac{(y-x)^2}{2\sigma}$$

到这里，就不难看出为什么我们说 ADMM 优化 v 的步骤可以看成一个降噪过程了，对比公式5和公式7，前者的 g(v) 就相当于后者的 $-lnp(x)$，前者的后一个平方差项和后者则是完全一致的。

[2]为什么要变成这种形式？一个直观的解释是新的函数更凸，而凸函数在优化是具有很好的性质，如收敛快，更容易获得更优解等。参见：交替方向乘子法（ADMM）算法的流程和原理是怎样的？ - 大大大的 v 的回答 - 知乎 `https://www.zhihu.com/question/36566112/answer/118715721`

1.3 Tuning Free PnP ADMM

在 PnP ADMM 中，观察公式4，我们知道这个算法存在两个超参数 ρ 和 σ。Tuning Free PnP ADMM[1] 这个算法就是使用强化学习的方法去自动寻找**每一步迭代**最适合的参数。

因为每一步迭代都可以使用不同的超参数，因此有时候可以获得比人工调参，甚至穷举[3]更优的结果。

当然，这个算法最大的好处还是不用自动化了调参的过程。

2 Examples

2.1 ADMM 1D TV Denosing

1D TV Denosing 问题描述如下，其中 F 是一个 Difference matrix，主对角线全是 1，主对角线上方元素是-1。

$$\text{minimize} \quad \frac{1}{2}\|x-y\|_2^2+\lambda\|Fx\|_1 \tag{8}$$

ADMM 形式：

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x-y\|_2^2+\lambda\|z\|_1 \\ \text{subject to} & Fx-z=0 \end{array} \tag{9}$$

增广朗格朗日形式：

$$L_\rho(x,z,\nu)=\frac{1}{2}\|x-y\|_2^2+\lambda\|z\|_1+\nu^T(Fx-z)+\frac{\rho}{2}\|Fx-z\|_2^2 \tag{10}$$

令 $\mu=\nu/\rho$，可以验证：

$$\nu^T(Fx-z)+\frac{\rho}{2}\|Fx-z\|_2^2=\frac{\rho}{2}\|Fx-z+\mu\|_2^2-\frac{\rho}{2}\|\mu\|_2^2 \tag{11}$$

因此，新的增广朗格朗日形式可以写成：

$$L_\rho(x,z,\nu)=\frac{1}{2}\|x-y\|_2^2+\lambda\|z\|_1+\frac{\rho}{2}\|Fx-z+\mu\|_2^2-\frac{\rho}{2}\|\mu\|_2^2 \tag{12}$$

因此，ADMM 的分布优化步骤为：

$$\begin{aligned} x^{(k+1)}&=\arg\min_x\left(\frac{1}{2}\|x-y\|_2^2+\frac{\rho}{2}\left\|Fx-z^{(k)}+\mu^{(k)}\right\|_2^2\right) \\ z^{(k+1)}&=\arg\min_z\left(\lambda\|z\|_1+\frac{\rho}{2}\left\|Fx^{(k+1)}-z+\mu^{(k)}\right\|_2^2\right) \\ \nu^{(k+1)}&=\nu^{(k)}+Fx^{(k+1)}-z^{(k+1)} \end{aligned} \tag{13}$$

这三个优化步骤都有解析解，如下 (牢记：$\mu=\nu/\rho$)：

$$\begin{aligned} x^{k+1}&:=\left(I+\rho F^TF\right)^{-1}\left(y+\rho F^T\left(z^k-\mu^k\right)\right) \\ z^{k+1}&:=T_{\lambda/\rho}\left(Fx^{k+1}+\mu^k\right) \\ \nu^{k+1}&:=\nu^k+Fx^{k+1}-z^{k+1} \end{aligned} \tag{14}$$

具体含义和推导见下：

[3]穷举是指在最开始穷举得到一个最优参数，但每一步迭代参数相同，因为每一步迭代都穷举并不现实。

求解 x: 优化 x 等价于求解一个最小二乘问题。推导如下：

最小二乘法的目标函数为：

$$L(D,\vec{\beta}) = \|X\vec{\beta} - Y\|^2 \tag{15}$$

有解析解：

$$(X^T X)^{-1} X^T Y \tag{16}$$

改写公式13中 x 的目标函数：

$$x^{(k+1)} = \arg\min_x \left(\|x - y\|_2^2 + \left\| \sqrt{\rho} F x - \sqrt{\rho}(z^{(k)} - \mu^{(k)}) \right\|_2^2 \right) \tag{17}$$

我们把这两个二范数合起来写成矩阵形式：

$$\min_x \left\| \begin{bmatrix} I \\ \sqrt{\rho} F \end{bmatrix} x - \begin{bmatrix} y \\ \sqrt{\rho}\left(z^{(k)} - \mu^{(k)}\right) \end{bmatrix} \right\|_2^2 \tag{18}$$

把 x 当成 β, 把式子前面的矩阵当成 X，把后面的矩阵当成 Y，可知，优化 x 等价于一个最小二乘问题。

代入最小二乘法的解析解公式：

$$\begin{aligned} x^{(k+1)} &= \left(I + \rho F^T F\right)^{-1} \left[I, \sqrt{\rho} F^T\right] \begin{bmatrix} y \\ \sqrt{\rho}\left(z^{(k)} - \mu^{(k)}\right)] \end{bmatrix} \\ &= \left(I + \rho F^T F\right)^{-1} \left(y + \rho F^T \left(z^{(k)} - \mu^{(k)}\right)\right) \end{aligned} \tag{19}$$

求解 z: z 是一个一维向量，将 z 的目标函数展开，令 $v = Fx + \mu$, 我们可以得到：

$$\begin{aligned} &\underset{z}{\text{minimize}}\ \lambda \sum_{n=1}^{N} |z[n]| + \frac{\rho}{2} \sum_{n=1}^{N} (z[n] - v[n])^2 \\ =&\text{minimize}_z\ \sum_{n=1}^{N} \left(\lambda |z[n]| + \frac{\rho}{2}(z[n] - v[n])^2\right) \end{aligned} \tag{20}$$

因为 z 的每一个分量没有关系，我们可以单独优化 z 的每一个分量：

$$\underset{z \in R}{\text{minimize}}\ \lambda |z| + \frac{\rho}{2}(z - v)^2 \tag{21}$$

这个函数除了 0 处处可导，且是个凸函数，导数为

$$\frac{df}{dz} = \begin{cases} \lambda + z - \rho v, & z > 0 \\ -\lambda + z - \rho v, & z < 0 \end{cases} \tag{22}$$

凸函数的极值点就是最值点，因此 z 的最优解为导数为 0 的地方。当 $|v| > \lambda/\rho$ 时，导数可以取到 0；反之，z 等于 0 时取到导数绝对值最小的位置，即最优解。

$$z^\star = \begin{cases} \rho v - \lambda, & v > \lambda/\rho \\ 0, & |v| \le \lambda/\rho \\ \rho v + \lambda, & v < -\lambda/\rho \end{cases} \tag{23}$$

用 $T_\lambda(\cdot)$ 表示这个函数，则

$$z^{(k+1)} = T_{\lambda/\rho}(v) = T_{\lambda/\rho}(Fx + \mu) \tag{24}$$

$T_\lambda(\cdot)$ 被称为 **soft thresholding** 或 **shrinkage** operator。

求解 ν: 使用公式10和求导即可：

$$\frac{d\nu}{dL} = (Fx - z)/\rho \tag{25}$$

ν 要最大化，使用梯度上升法。

2.1.1 参数设置

ADMM 1D TV Denosing 就两个超参数 λ 和 ρ ，参数设置基本不会影响大致结果，但是要注意的是 ρ 不要小于 1，否则可能会造成 x,z 更新之后过大，出现 INF。

References

[1] Kaixuan Wei et al. “Tuning-free Plug-and-Play Proximal Algorithm for Inverse Imaging Problems”. In: *arXiv preprint arXiv:2002.09611* (2020) (cit. on p. 3).