

分裂 Bregman 方法在图像处理中的一些应用

Some Applications of Split Bregman Method in Image Processing

姓名: 谭 兵

导师: 周彬副教授

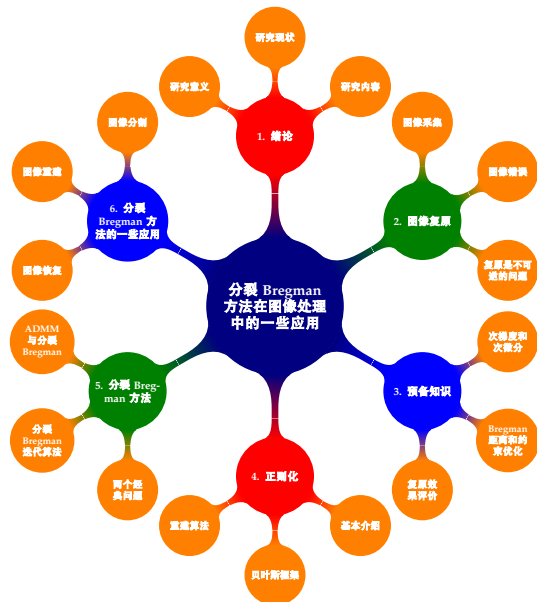
专业: 数学与应用数学



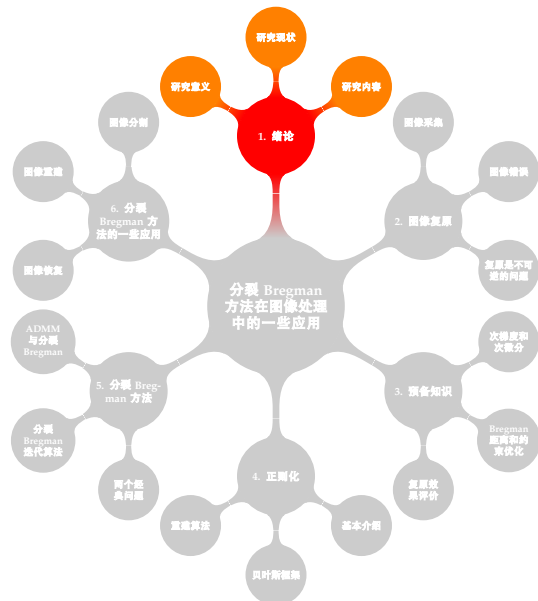
2014 级学士学位论文答辩

2018-06-10

内容提纲



绪论



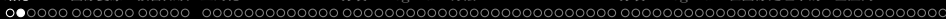


题目来源

数学，计算机科学与工程

排名	研究前沿	核心文献	引用次数	核心文献平均出版年
1	高能可充电锂-空气电池	49	2,006	2010.8
2	非线性分数阶微分方程的边界值问题	47	1,172	2010.2
3	生物柴油燃料燃烧的化学动力学反应机制	49	1,555	2010.0
4	非局部铁摩辛柯梁理论和碳纳米管	39	1,480	2009.8
5	受约束的全变分图像去噪与恢复	49	2,741	2009.7
6	石墨烯晶体管	16	2,270	2009.7
7	分析新一代 DNA 测序数据	6	2,025	2009.6
8	纳米流体传热	40	1,928	2009.6
9	二氧化碳捕获中的钙循环过程	36	1,562	2009.6
10	差分进化算法和 Memetic 计算	30	1,351	2009.6

来源: 汤森路透 Essential Science Indicators



研究目的及意义

研究意义

诸多学者提出各种方法和算法来修复或恢复各种信号、图像和视频信息, 面向的情形可能是稀疏的数据源, 或者噪音和模糊部分需要修正, 或者需要补充丢失的数据。这些技术的应用包括医学图像处理和情报收集。具体例如在充满噪音的视频中追踪运动物体, 由卫星观测来定位地面物体, 无人驾驶飞机的控制, 和在 CT 扫描中使用最小的放射剂量。

研究目的

针对不同的图像处理目的, 已产生许多模型和算法。利用变分法完成建模并设计求解算法是当前解决图像处理问题的重要途径之一, 而分裂 Bregman 方法则是近年发展起来的求解变分模型的重要手段之一。



研究目的及意义

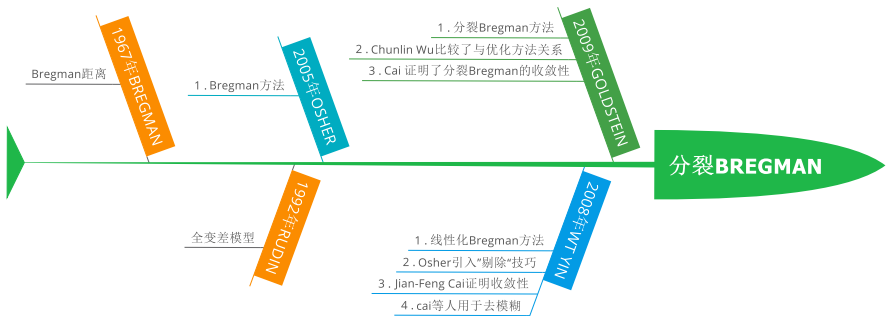
研究意义

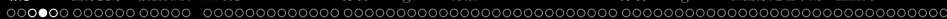
诸多学者提出各种方法和算法来修复或恢复各种信号、图像和视频信息, 面向的情形可能是稀疏的数据源, 或者噪音和模糊部分需要修正, 或者需要补充丢失的数据。这些技术的应用包括医学图像处理和情报收集。具体例如在充满噪音的视频中追踪运动物体, 由卫星观测来定位地面物体, 无人驾驶飞机的控制, 和在 CT 扫描中使用最小的放射剂量。

研究目的

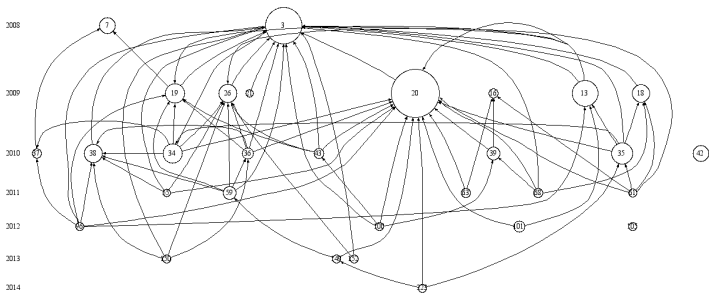
针对不同的图像处理目的, 已产生许多模型和算法。利用变分法完成建模并设计求解算法是当前解决图像处理问题的重要途径之一, 而分裂 Bregman 方法则是近年发展起来的求解变分模型的重要手段之一。

国内外研究现状





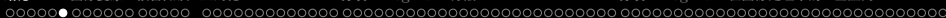
核心文献之间的引用关系



- ③ Yin WT, Osher S et al, Bregman Iterative Algorithms for ℓ_1 -Minimization with Applications to Compressed Sensing, [2008]
- ⑳ Goldstein T, Osher S, The Split Bregman Method for L1-Regularized Problems, [2009]

关键人物: 多篇核心文献的作者





研究方法和研究内容

第一部分

综述分裂 Bregman 方法国内外研究现状, 介绍图像复原的基本知识;

第二部分

介绍 Bregman 距离等预备知识, 引入正则化思想和贝叶斯框架;

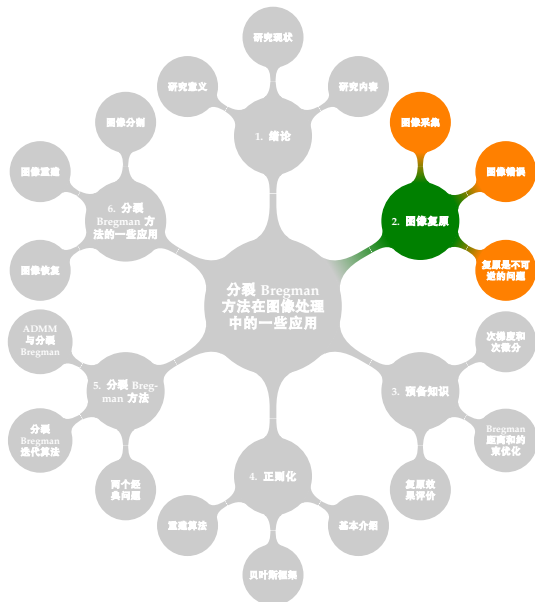
第三部分

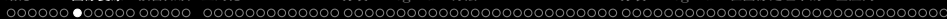
阐述分裂 Bregman 迭代算法的思想、步骤及收敛性, 介绍分裂 Bregman 迭代算法在图像处理中的应用, 并通过数值实验说明其优越性;

第四部分

总结论文内容, 并对进一步的工作进行展望。

图像复原





图像采集

图像采集

图像通过图像系统获取, 图像系统包括光学部件和检测器, 他们都必须考虑到成像过程的准确建模。



图像采集

光学系统

光学系统可以被概括为一种装置, 其收集来自物体的光在平面中形成的图像, 因此称为图像平面。

下面的积分描述物体 $x(s)$ 的强度与获得的图像 $\tilde{f}(s)$ 之间的线性关系:

$$\tilde{f}(s) = \int H(s - s') x(s') ds'$$

H 被称为脉冲响应。



图像采集

光学系统

光学系统可以被概括为一种装置, 其收集来自物体的光在平面中形成的图像, 因此称为图像平面。

下面的积分描述物体 $x(s)$ 的强度与获得的图像 $\tilde{f}(s)$ 之间的线性关系:

$$\tilde{f}(s) = \int H(s - s') x(s') ds'$$

H 被称为脉冲响应。

检测器

除了光学部件之外, 图像系统的另一个基本装置是检测器, 它位于图像平面内并测量入射辐射, 将其转换为电信号。



图像中形成的错误

模糊

一般来说, 模糊及其起源可以分为三大类, 每一类都有自己的 PSF:

- 光学模糊, 常被称为“离焦模糊”, 这是由于图像平面偏离光学透镜的焦点造成的;
- 机械模糊, 或称为“运动模糊”, 其由图像采集过程期间目标物体或图像装置的快速机械运动引起的;
- 中等诱导模糊, 这是由于光线通过其传播的介质的散射或光学湍流造成的。



图像中形成的错误

噪声

- 高斯噪声:均值为 μ 和方差为 σ^2 的单变量高斯噪声的密度函数是

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

- 泊松噪声:在给定时间间隔 T 内接收 k 个粒子的概率由下式给出:

$$p(k) = \frac{e^{-\lambda} \lambda^k}{k!} \quad k = 0, 1, 2, \dots$$

其中 λ 与 T 成正比,是事件的预期值。



图像复原是不可逆的问题

使用算子理论:在定义了 x 和 \tilde{f} 的函数空间上,考虑算子 T :

$$T\{x\} \rightarrow \tilde{f}$$

我们有

$$T\{x\} = \int H(s - s')x(s') ds'$$

然后图像恢复的问题就是找到逆变换 T^{-1} 使得

$$T^{-1}\{\tilde{f}\} \rightarrow x$$

由于这个原因,图像处理中发现的大多数问题都是反问题。



图像复原是不可逆的问题

使用算子理论:在定义了 x 和 \tilde{f} 的函数空间上,考虑算子 T :

$$T\{x\} \rightarrow \tilde{f}$$

我们有

$$T\{x\} = \int H(s - s')x(s') ds'$$

然后图像恢复的问题就是找到逆变换 T^{-1} 使得

$$T^{-1}\{\tilde{f}\} \rightarrow x$$

由于这个原因,图像处理中发现的大多数问题都是反问题。



图像复原是不可逆的问题

一个问题通过使用稳定算法计算得到了“好”的解决方案,前提该问题是适当且条件充足的。

适定问题

在 Hadamard 意义上, 一个问题 是适定的, 如果满足以下三条性质:

- 存在解决方案;
- 该解决方案是唯一的;
- 该解决方案连续依赖于定解条件, 即解是稳定的。



图像复原是不可逆的问题

图像恢复的不适定性可以用 Riemann-Lebesgue 引理来证明,它指出:

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') \sin(\alpha s') ds' = 0$$

然后有

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') [x(s') + \sin(\alpha s')] ds' = \int H(s - s') x(s') ds' = \tilde{y} \quad (2.1)$$

换句话说,无限频率的正弦曲线可以被添加到物体分布 x , 并且结果与图像分布 \tilde{y} 相同。所以图像恢复问题是一个不适定的问题。

(2.1)的直接含义是我们选择一个小的值 ϵ_1 , 存在一个值 A 使得

$$\int H(s - s') \sin(\alpha s') ds' < \epsilon_1 \quad \forall \alpha \geq A$$

因此

$$\int H(s - s') [x(s') + \sin(\alpha s')] ds' = \tilde{y} + \epsilon \quad \forall \alpha \geq A, |\epsilon| < \epsilon_1$$



图像复原是不可逆的问题

图像恢复的不适定性可以用 Riemann-Lebesgue 引理来证明,它指出:

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') \sin(\alpha s') ds' = 0$$

然后有

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') [x(s') + \sin(\alpha s')] ds' = \int H(s - s') x(s') ds' = \tilde{y} \quad (2.1)$$

换句话说,无限频率的正弦曲线可以被添加到物体分布 x , 并且结果与图像分布 \tilde{y} 相同。所以图像恢复问题是一个不适定的问题。

(2.1)的直接含义是我们选择一个小的值 ϵ_1 , 存在一个值 A 使得

$$\int H(s - s') \sin(\alpha s') ds' < \epsilon_1 \quad \forall \alpha \geq A$$

因此

$$\int H(s - s') [x(s') + \sin(\alpha s')] ds' = \tilde{y} + \epsilon \quad \forall \alpha \geq A, |\epsilon| < \epsilon_1$$



图像复原是不可逆的问题

图像恢复的不适定性可以用 Riemann-Lebesgue 引理来证明,它指出:

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') \sin(\alpha s') ds' = 0$$

然后有

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') [x(s') + \sin(\alpha s')] ds' = \int H(s - s') x(s') ds' = \tilde{y} \quad (2.1)$$

换句话说,无限频率的正弦曲线可以被添加到物体分布 x ,并且结果与图像分布 \tilde{y} 相同。所以图像恢复问题是一个不适定的问题。

(2.1)的直接含义是我们选择一个小的值 ϵ_1 ,存在一个值 A 使得

$$\int H(s - s') \sin(\alpha s') ds' < \epsilon_1 \quad \forall \alpha \geq A$$

因此

$$\int H(s - s') [x(s') + \sin(\alpha s')] ds' = \tilde{y} + \epsilon \quad \forall \alpha \geq A, |\epsilon| < \epsilon_1$$



图像复原是不可逆的问题

图像恢复的不适定性可以用 Riemann-Lebesgue 引理来证明, 它指出:

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') \sin(\alpha s') ds' = 0$$

然后有

$$\lim_{\alpha \rightarrow \infty} \int H(s - s') [x(s') + \sin(\alpha s')] ds' = \int H(s - s') x(s') ds' = \tilde{y} \quad (2.1)$$

换句话说, 无限频率的正弦曲线可以被添加到物体分布 x , 并且结果与图像分布 \tilde{y} 相同。所以图像恢复问题是一个不适定的问题。

(2.1)的直接含义是我们选择一个小的值 ϵ_1 , 存在一个值 A 使得

$$\int H(s - s') \sin(\alpha s') ds' < \epsilon_1 \quad \forall \alpha \geq A$$

因此

$$\int H(s - s') [x(s') + \sin(\alpha s')] ds' = \tilde{y} + \epsilon \quad \forall \alpha \geq A, |\epsilon| < \epsilon_1$$

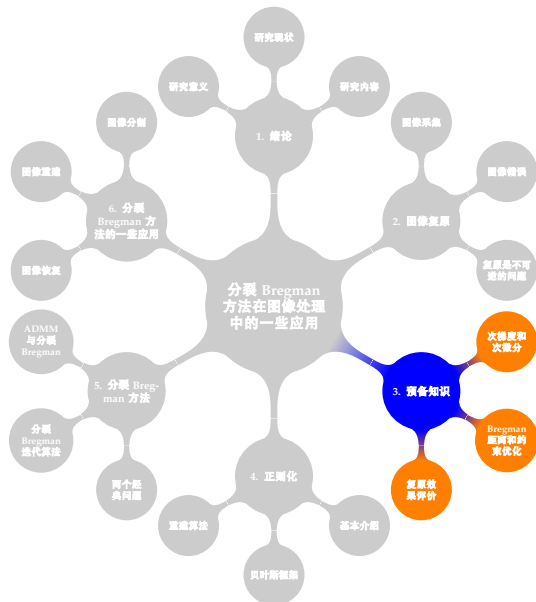


图像复原是不可逆的问题

图像恢复是一个不可逆的问题

这表明, 如果选择无限小的值 ϵ 并将其添加到图像分布 \tilde{y} 中, 则从图像恢复的意义上来讲, 这不能与具有频率为 α 的正弦曲线的附加分量的原始对象分布相区分开。由于 ϵ 可以是任意小的, 因此无法将平凡的扰动与原始对象分布中的有限非平凡扰动区分开来。

预备知识



次梯度和次微分

次梯度与次微分

假设 $J : X \rightarrow \mathbb{R}$ 是一个凸函数, 并且 $u \in X$ 。如果向量 $p \in X^*$ (X^* 为 X 的对偶空间) 对任意的 $\forall v \in X^*$, 有

$$J(v) - J(u) - \langle p, v - u \rangle \geq 0.$$

则称 p 为向量值函数 $J(x)$ 在 u 处的次梯度, 所有在 u 处 J 的次梯度的集合称为 J 在 u 处的次微分, 它由 $\partial J(u)$ 表示。

请注意, 次微分扩展了函数的渐变概念。例如, L_1 范数不可微。但是存在次梯度, 即是次微分的。 $\frac{\partial}{\partial \mathbf{x}} \|\mathbf{x}\|_1 = \text{sign}(\mathbf{x})$ 。

次梯度和次微分

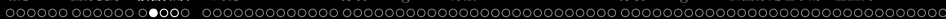
次梯度与次微分

假设 $J : X \rightarrow \mathbb{R}$ 是一个凸函数, 并且 $u \in X$ 。如果向量 $p \in X^*$ (X^* 为 X 的对偶空间) 对任意的 $\forall v \in X^*$, 有

$$J(v) - J(u) - \langle p, v - u \rangle \geq 0.$$

则称 p 为向量值函数 $J(x)$ 在 u 处的次梯度, 所有在 u 处 J 的次梯度的集合称为 J 在 u 处的次微分, 它由 $\partial J(u)$ 表示。

请注意, 次微分扩展了函数的渐变概念。例如, $L1$ 范数不可微。但是存在次梯度, 即是次微分的。 $\frac{\partial}{\partial \mathbf{x}} \|\mathbf{x}\|_1 = \text{sign}(\mathbf{x})$ 。



Bregman 距离

本文讨论的所有算法都依赖于 L.M.Bregman 在 1966 年提出的 Bregman 距离。

Bregman 距离

Bregman 距离

假设 $J : X \rightarrow \mathbb{R}$ 是一个凸函数, $u, v \in X$ 和 $\xi \in \partial J(v)$ 。那么点 u 和 v 之间的 Bregman 距离由下式定义

$$D^\xi(u, v) \equiv D_J^\xi(u, v) = J(u) - J(v) - \langle \xi, u - v \rangle.$$

Bregman 距离

Bregman 距离

假设 $J : X \rightarrow \mathbb{R}$ 是一个凸函数, $u, v \in X$ 和 $\xi \in \partial J(v)$ 。那么点 u 和 v 之间的 Bregman 距离由下式定义

$$D^\xi(u, v) \equiv D_J^\xi(u, v) = J(u) - J(v) - \langle \xi, u - v \rangle.$$

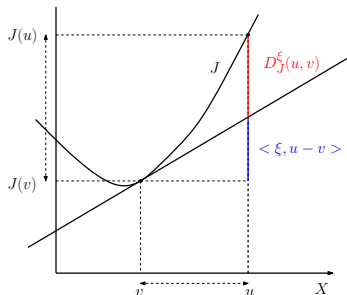


图 1: Bregman 距离 $D_J^\xi(u, v)$

Bregman 距离

Bregman 距离有几个很好的特性, 使它成为解决 L1 正则化问题的有效工具, 例如,

性质

对于 $u, v \in X$ 和 $\xi \in \partial J(v)$, $D_J^\xi(u, v) \geq 0$

性质

$D_J^\xi(v, v) = 0$



约束优化

一般约束优化问题可以表示为：

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} h_i(x) = 0 & i = 1, 2, \dots, m \\ g_j(x) \leq 0 & j = 1, 2, \dots, n \end{cases}$$



约束优化

一般约束优化问题可以表示为：

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} h_i(x) = 0 & i = 1, 2, \dots, m \\ g_j(x) \leq 0 & j = 1, 2, \dots, n \end{cases}$$

我们知道对于等式约束的最小化问题可以使用拉格朗日法进行求解, 对于不等式约束优化问题, 只要满足 KKT 条件, 也可用拉格朗日法求解。



约束优化

一般约束优化问题可以表示为：

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} h_i(x) = 0 & i = 1, 2, \dots, m \\ g_j(x) \leq 0 & j = 1, 2, \dots, n \end{cases}$$

我们首先定义上式问题的拉格朗日函数为：

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_{i=1}^m \alpha_i h_i(x) + \sum_{j=1}^n \beta_j g_j(x)$$



约束优化

一般约束优化问题可以表示为：

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} h_i(x) = 0 & i = 1, 2, \dots, m \\ g_j(x) \leq 0 & j = 1, 2, \dots, n \end{cases}$$

我们首先定义上式问题的拉格朗日函数为：

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_{i=1}^m \alpha_i h_i(x) + \sum_{j=1}^n \beta_j g_j(x)$$

然后可行解 x^* 需要满足的 KKT 条件如下所述：

$$\nabla_x \mathcal{L}(x^*, \alpha, \beta) = 0$$

$$h_i(x^*) = 0$$

$$g_j(x^*) \leq 0$$

$$\beta_j g_j(x^*) = 0$$

$$\beta_j \geq 0$$



约束优化

一般约束优化问题可以表示为：

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to} \quad \begin{cases} h_i(x) = 0 & i = 1, 2, \dots, m \\ g_j(x) \leq 0 & j = 1, 2, \dots, n \end{cases}$$

我们首先定义上式问题的拉格朗日函数为：

$$\mathcal{L}(x, \alpha, \beta) = f(x) + \sum_{i=1}^m \alpha_i h_i(x) + \sum_{j=1}^n \beta_j g_j(x)$$

KKT 条件结论

对于一般问题而言, KKT 条件是一组解成为最优解的必要条件, 而对于凸问题而言 KKT 条件是一组解成为最优解的必要条件。



复原质量评价

客观评价图像质量有很多指标, 常见的有均方误差、归一化均方误差、信噪比、峰值信噪比等, 其中的一个重要指标是峰值信噪比 (PSNR), 它是信号最大可能功率与噪声功率的比值, 单位是分贝, 它在文献中有许多不同的定义。给定 u 是原始图像, u_0 是恢复的图像, 本文使用的 PSNR 的一个流行定义是

峰值信噪比 PSNR

$$PSNR = 10 \times \log_{10} \frac{\sum_{i=1}^N 255^2}{\frac{1}{MN} \sum_{i=1}^N (u(i) - u_0(i))^2}$$

通常, 较大的 PSNR 表明图像重建质量较高。

复原质量评价

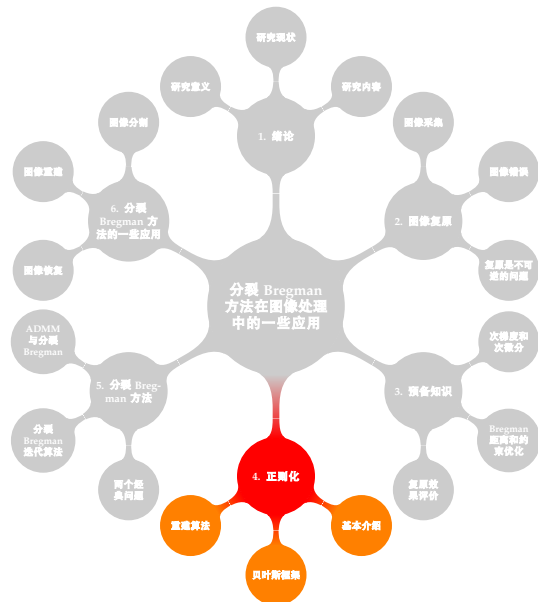
客观评价图像质量有很多指标, 常见的有均方误差、归一化均方误差、信噪比、峰值信噪比等, 其中的一个重要指标是峰值信噪比(PSNR), 它是信号最大可能功率与噪声功率的比值, 单位是分贝, 它在文献中有许多不同的定义。给定 u 是原始图像, u_0 是恢复的图像, 本文使用的 PSNR 的一个流行定义是

峰值信噪比 PSNR

$$PSNR = 10 \times \log_{10} \frac{\sum_{i=1}^N 255^2}{\frac{1}{MN} \sum_{i=1}^N (u(i) - u_0(i))^2}$$

通常, 较大的 PSNR 表明图像重建质量较高。

正则化





基本介绍

我们来看一个欠定问题的例子。具有两个方程和三个未知数 x_1, x_2, x_3 的方程组

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

该问题的通解是

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} t$$

显然该问题有无数组解, 它可以是 $x_1 = 0, x_2 = 1, x_3 = 0 (t = 0)$ 或 $x_1 = 1, x_2 = -1, x_3 = 1 (t = 1)$ 。



基本介绍

我们来看一个欠定问题的例子。具有两个方程和三个未知数 x_1, x_2, x_3 的方程组

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

该问题的通解是

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} t$$

显然该问题有无数组解, 它可以是 $x_1 = 0, x_2 = 1, x_3 = 0 (t = 0)$ 或 $x_1 = 1, x_2 = -1, x_3 = 1 (t = 1)$ 。



基本介绍

正则化是基于某些条件、信息、先验知识等将解决方案限制为唯一的。下面我们介绍两种如何在不添加新方程的情况下调整欠定方程的方法。

L2 最小长度解

从 n 维空间的原点到点 $x \in \mathbb{R}^n$ 的欧几里德距离可以测量为 ℓ_2 范数 $\|x\|_2$ 。找到欠定方程解中具有与原点的最小欧几里德距离 (最小 ℓ_2 范数) 的最小长度解的问题可以描述为以下最小化问题。

$$\min_x \|x\|_2^2 \quad \text{subject to} \quad Ax = b.$$

上式的拉格朗日函数为 $L(x, \lambda) = \frac{1}{2}x^\top x + \lambda^\top (Ax - b)$, 唯一最小解 x^* 如下

$$x^* = A^\top (AA^\top)^{-1}b$$

上面例子的 L2 最小长度解为 (0.33, 0.33, 0.33)



基本介绍

正则化是基于某些条件、信息、先验知识等将解决方案限制为唯一的。下面我们介绍两种如何在不添加新方程的情况下调整欠定方程的方法。

L2 最小长度解

从 n 维空间的原点到点 $x \in \mathbb{R}^n$ 的欧几里德距离可以测量为 l_2 范数 $\|x\|_2$ 。找到欠定方程解中具有与原点的最小欧几里德距离 (最小 l_2 范数) 的最小长度解的问题可以描述为以下最小化问题。

$$\min_x \|x\|_2^2 \quad \text{subject to} \quad Ax = b.$$

上式的拉格朗日函数为 $L(x, \lambda) = \frac{1}{2}x^\top x + \lambda^\top (Ax - b)$, 唯一最小解 x^* 如下

$$x^* = A^\top (AA^\top)^{-1}b$$

上面例子的 L2 最小长度解为 (0.33, 0.33, 0.33)



基本介绍

正则化是基于某些条件、信息、先验知识等将解决方案限制为唯一的。下面我们介绍两种如何在不添加新方程的情况下调整欠定方程的方法。

L2 最小长度解

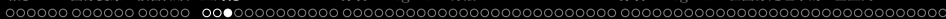
从 n 维空间的原点到点 $x \in \mathbb{R}^n$ 的欧几里德距离可以测量为 l_2 范数 $\|x\|_2$ 。找到欠定方程解中具有与原点的最小欧几里德距离 (最小 l_2 范数) 的最小长度解的问题可以描述为以下最小化问题。

$$\min_x \|x\|_2^2 \quad \text{subject to} \quad Ax = b.$$

上式的拉格朗日函数为 $L(x, \lambda) = \frac{1}{2}x^\top x + \lambda^\top (Ax - b)$, 唯一最小解 x^* 如下

$$x^* = A^\top (AA^\top)^{-1}b$$

上面例子的 L2 最小长度解为 (0.33, 0.33, 0.33)



基本介绍

L2 正则化

l_2 正则化与 l_2 最小长度解相似, l_2 正则化最小化了联立方程偏离的量(误差)和解的 l_2 范数的平方 $\|x\|_2^2$ 。

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_2^2$$

其中 λ 为正则化参数, l_2 正则化在统计领域被称为岭回归。该式确定的唯一解 x^* 表示为

$$x^* = (A^T A + 2\lambda I)^{-1} A^T b$$

由上式确定的最小值解 x^* 不再精确地满足 $Ax = b$ 。 Ax 和 b 之间的偏移量的大小 $\|Ax - b\|_2^2$ 称为损失函数。正常数 λ 起到平衡的作用, 使得目标的损失函数和 l_2 范数都变小, 并且有必要预先设定适当的值。将选择具有损失函数小(近似良好满足联立方程)的解。



基本介绍

L2 正则化

l_2 正则化与 l_2 最小长度解相似, l_2 正则化最小化了联立方程偏离的量(误差)和解的 l_2 范数的平方 $\|x\|_2^2$ 。

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_2^2$$

其中 λ 为正则化参数, l_2 正则化在统计领域被称为岭回归。该式确定的唯一解 x^* 表示为

$$x^* = (A^T A + 2\lambda I)^{-1} A^T b$$

由上式确定的最小值解 x^* 不再精确地满足 $Ax = b$ 。 Ax 和 b 之间的偏移量的大小 $\|Ax - b\|_2^2$ 称为损失函数。正常数 λ 起到平衡的作用,使得目标的损失函数和 l_2 范数都变小,并且有必要预先设定适当的值。将选择具有损失函数小(近似良好满足联立方程)的解。

贝叶斯框架——最大似然函数

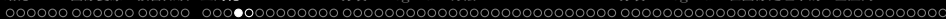
使用前面的符号, 设 f 表示检测到的图像, 在高斯噪声的情况下, f 是随机变量 F 的实现, 其元素 F_i 是具有均值 $(Hx + b)_i$ 和方差 σ_i^2 的高斯随机变量。



贝叶斯框架 —— 最大似然函数

假设与每个像素相关的随机变量在统计上是独立的, 那么在给定 H , x 和 b 的情况下, F 的概率分布为:

$$p_F(f; x) = \prod_{i \in S} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(f_i - (Hx+b)_i)^2}{2\sigma_i^2}}$$



贝叶斯框架 —— 最大似然函数

假设与每个像素相关的随机变量在统计上是独立的, 那么在给定 H , x 和 b 的情况下, F 的概率分布为:

$$p_F(f; x) = \prod_{i \in S} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(f_i - (Hx + b)_i)^2}{2\sigma_i^2}}$$

最大似然法 (ML) 估计就是这种函数的最大值。在高斯噪声的情况下, 如果我们对上式取负对数, 然后可以得到

$$\phi_0(x) = \sum_{i \in S} \left\{ -\ln \frac{1}{\sqrt{2\pi\sigma_i^2}} + \frac{(f_i - (Hx + b)_i)^2}{2\sigma_i^2} \right\}$$



贝叶斯框架 —— 最大似然函数

假设与每个像素相关的随机变量在统计上是独立的, 那么在给定 H , x 和 b 的情况下, F 的概率分布为:

$$p_F(f; x) = \prod_{i \in S} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(f_i - (Hx + b)_i)^2}{2\sigma_i^2}}$$

最大似然法 (ML) 估计就是这种函数的最大值。在高斯噪声的情况下, 如果我们对上式取负对数, 然后可以得到

$$\phi_0(x) = \sum_{i \in S} \left\{ -\ln \frac{1}{\sqrt{2\pi\sigma_i^2}} + \frac{(f_i - (Hx + b)_i)^2}{2\sigma_i^2} \right\}$$

由于 $\ln \frac{1}{\sqrt{2\pi\sigma_i^2}}$ 是常数, 上式的本质是最小化加权最小二乘逼近

$$\min \frac{1}{2} \left\| \frac{f_i - (Hx + b)_i}{\sigma_i} \right\|_2^2$$



贝叶斯框架 —— 最大后验估计

在贝叶斯方法中, 未知对象 x 也被认为是一个多值随机变量的实现, 并且先验信息被编码成该随机变量的给定概率分布 $p_X(x)$, 即所谓的先验信息。

贝叶斯公式表示为:

$$p_X(x|f) = \frac{p_X(x)p_F(f|x)}{p_F(x)}$$

如果在上面等式中加入 f 的检测值, 我们得到 X 的后验概率分布

$$P_f^X(x) = L_f^F(x) \frac{p_X(x)}{p_F(f)}$$



贝叶斯框架 —— 最大后验估计

在贝叶斯方法中, 未知对象 x 也被认为是一个多值随机变量的实现, 并且先验信息被编码成该随机变量的给定概率分布 $p_X(x)$, 即所谓的先验信息。

贝叶斯公式表示为:

$$p_X(x|f) = \frac{p_X(x)p_F(f|x)}{p_F(x)}$$

如果在上面等式中加入 f 的检测值, 我们得到 X 的后验概率分布

$$P_f^X(x) = L_f^F(x) \frac{p_X(x)}{p_F(f)}$$



重建算法

我们首先从基追踪 (Basis Pursuit) 问题出发

$$\operatorname{argmin}_x \|x\|_1 \quad \text{such that } Ax = b$$

LASSO 回归

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1$$

岭回归

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_2^2$$

最佳子集回归

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_0$$

重建算法

下面我们简单分析下选择 L1 正则项和 L2 正则项之间的区别和联系。

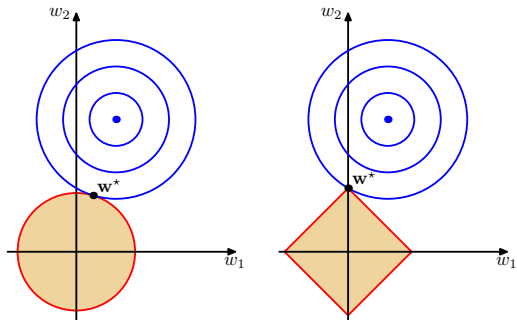


图 2: 误差函数与 L2 正则化(左)和 LASSO 正则化(右)的约束区域

因此, L1 范数可以导致稀疏解, L2 范数可以导致稠密解。

重建算法

下面我们简单分析下选择 L1 正则项和 L2 正则项之间的区别和联系。

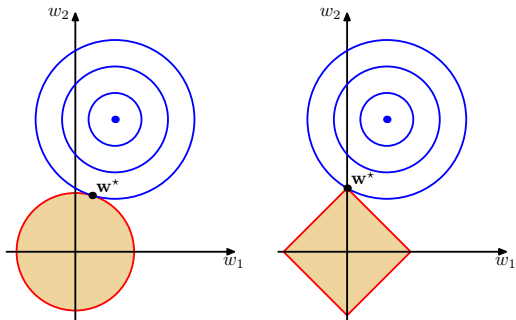


图 2: 误差函数与 L2 正则化(左)和 LASSO 正则化(右)的约束区域

因此, L1 范数可以导致稀疏解, L2 范数可以导致稠密解。

其他正则化的例子

其他正则化的例子：

- Elastic-net: 这种估计是 Lasso 和 Ridge 的结合, 通过找到最小化:

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_2^2 + \mu \|x\|_1$$

该估计具有 Ridge 和 Lasso 回归的好处: 从 LASSO 中进行特征选择, 以及从 Ridge 回归中对数值稳定性进行正则化。

- TV 正则化:

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|\nabla x\|_1$$

这种类型的正则化用于保留边缘, 同时去除噪声信息。它广泛用于图像处理。

其他正则化的例子

其他正则化的例子：

- Elastic-net: 这种估计是 Lasso 和 Ridge 的结合, 通过找到最小化:

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_2^2 + \mu \|x\|_1$$

该估计具有 Ridge 和 Lasso 回归的好处: 从 LASSO 中进行特征选择, 以及从 Ridge 回归中对数值稳定性进行正则化。

- TV 正则化:

$$\operatorname{argmin}_x \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|\nabla x\|_1$$

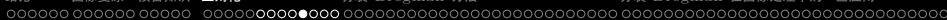
这种类型的正则化用于保留边缘, 同时去除噪声信息。它广泛用于图像处理。



重建算法总结

广泛的重建算法分为三类:凸优化,贪心方法和贝叶斯方法。

- 凸优化方法提供了更好的性能,以重构精度来衡量,代价是计算复杂度更高。
- 贪婪方法比较简单,但没有凸算法的重构保证。
- 贝叶斯方法提供了最佳的重建保证,以及重建质量的不确定性估计,但具有相当大的计算复杂性。



贝叶斯理论看正则化

接下来我们将从贝叶斯的观点来看常见的一些回归方法, 首先给出我们的结论, 从概率论的角度来说:

- 最小二乘的解析解可以从高斯分布以及最大似然估计求得
- Ridge 回归可以用高斯分布和最大后验估计获得
- LASSO 回归可以用拉普拉斯分布和最大后验估计获得



从贝叶斯观点看岭回归

我们对参数 θ 引入方差为 α^2 的零均值高斯先验 $\theta_i \sim \mathcal{N}(0, \alpha^2)$.

对参数 θ 引入高斯先验后, $y^{(i)} - \theta^\top x^{(i)}$ 的似然函数 $L(\theta)$ 为

$$\begin{aligned} L(\theta) &= p(y|X; \theta)p(\alpha) = \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta)p(\alpha) \\ &= \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^\top x^{(i)})^2}{2\sigma^2}\right) \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\alpha} \exp\left(-\frac{(\theta^{(j)})^2}{2\alpha^2}\right) \end{aligned}$$

取对数有:

$$l(\theta) = \ln L(\theta)$$

$$= m \ln \frac{1}{\sqrt{2\pi}\sigma} + n \ln \frac{1}{\sqrt{2\pi}\alpha} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})^2 - \frac{1}{\alpha^2} \cdot \frac{1}{2} \sum_{j=1}^n (\theta^{(j)})^2$$



从贝叶斯观点看岭回归

我们对参数 θ 引入方差为 α^2 的零均值高斯先验 $\theta_i \sim \mathcal{N}(0, \alpha^2)$.

对参数 θ 引入高斯先验后, $y^{(i)} - \theta^\top x^{(i)}$ 的似然函数 $L(\theta)$ 为

$$\begin{aligned} L(\theta) &= p(y|X; \theta)p(\alpha) = \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta)p(\alpha) \\ &= \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^\top x^{(i)})^2}{2\sigma^2}\right) \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\alpha} \exp\left(-\frac{(\theta^{(j)})^2}{2\alpha^2}\right) \end{aligned}$$

取对数有:

$$l(\theta) = \ln L(\theta)$$

$$= m \ln \frac{1}{\sqrt{2\pi}\sigma} + n \ln \frac{1}{\sqrt{2\pi}\alpha} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})^2 - \frac{1}{\alpha^2} \cdot \frac{1}{2} \sum_{j=1}^n (\theta^{(j)})^2$$



从贝叶斯观点看岭回归

我们对参数 θ 引入方差为 α^2 的零均值高斯先验 $\theta_i \sim \mathcal{N}(0, \alpha^2)$.

对参数 θ 引入高斯先验后, $y^{(i)} - \theta^\top x^{(i)}$ 的似然函数 $L(\theta)$ 为

$$\begin{aligned}
 L(\theta) &= p(y|X; \theta)p(\alpha) = \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta)p(\alpha) \\
 &= \prod_{i=1}^m \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^\top x^{(i)})^2}{2\sigma^2}\right) \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\alpha} \exp\left(-\frac{(\theta^{(j)})^2}{2\alpha^2}\right)
 \end{aligned}$$

取对数有:

$$\begin{aligned}
 l(\theta) &= \ln L(\theta) \\
 &= m \ln \frac{1}{\sqrt{2\pi}\sigma} + n \ln \frac{1}{\sqrt{2\pi}\alpha} - \frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})^2 - \frac{1}{\alpha^2} \cdot \frac{1}{2} \sum_{j=1}^n (\theta^{(j)})^2
 \end{aligned}$$

从贝叶斯观点看岭回归

得到引入高斯先验后的 θ 值的最大后验估计为

$$\theta_{MAP_{Guassian}} = \underset{\theta}{\operatorname{argmin}} \left(\frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})^2 + \frac{1}{\alpha^2} \cdot \frac{1}{2} \sum_{j=1}^n (\theta^{(j)})^2 \right)$$

这等价于：

$$J_R(\theta) = \frac{1}{2} \|y - \theta^\top X\|_2^2 + \lambda \|\theta\|_2^2$$

最终公式就是岭回归计算公式。最大后验估计就是在最大似然估计公式后乘以高斯先验, 现在我们知道在参数中引入高斯先验等价于 L2 正则化。

从贝叶斯观点看岭回归

得到引入高斯先验后的 θ 值的最大后验估计为

$$\theta_{MAP_{Guassian}} = \operatorname{argmin}_{\theta} \left(\frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})^2 + \frac{1}{\alpha^2} \cdot \frac{1}{2} \sum_{j=1}^n (\theta^{(j)})^2 \right)$$

这等价于：

$$J_R(\theta) = \frac{1}{2} \|y - \theta^\top X\|_2^2 + \lambda \|\theta\|_2^2$$

最终公式就是岭回归计算公式。最大后验估计就是在最大似然估计公式后乘以高斯先验, 现在我们知道在参数中引入高斯先验等价于 L2 正则化。



从贝叶斯观点看岭回归

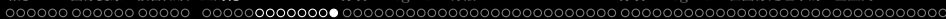
得到引入高斯先验后的 θ 值的最大后验估计为

$$\theta_{MAP_{Guassian}} = \underset{\theta}{\operatorname{argmin}} \left(\frac{1}{\sigma^2} \cdot \frac{1}{2} \sum_{i=1}^m (y^{(i)} - \theta^\top x^{(i)})^2 + \frac{1}{\alpha^2} \cdot \frac{1}{2} \sum_{j=1}^n (\theta^{(j)})^2 \right)$$

这等价于：

$$J_R(\theta) = \frac{1}{2} \|y - \theta^\top X\|_2^2 + \lambda \|\theta\|_2^2$$

最终公式就是岭回归计算公式。最大后验估计就是在最大似然估计公式后乘以高斯先验, 现在我们知道在参数中引入高斯先验等价于 L2 正则化。

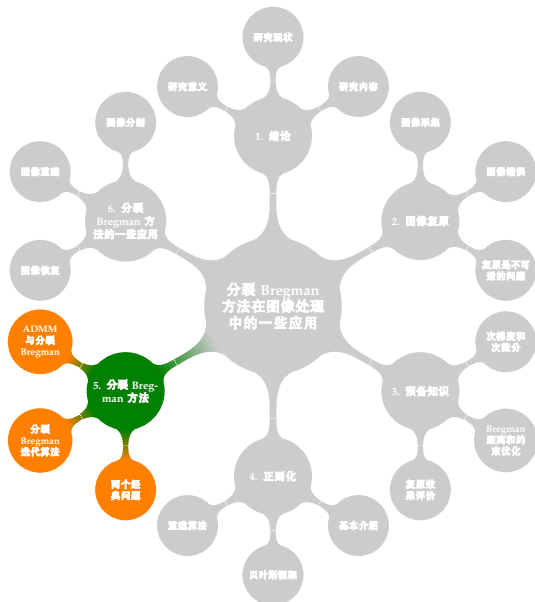


从贝叶斯观点看正则化

从贝叶斯观点看正则化

正则化参数相当于在参数中引入先验分布,降低了模型的复杂度(减少了解空间),增强了模型对噪声的鲁棒性(泛化能力)。从贝叶斯角度看,整个优化问题是一种贝叶斯最大后验估计,损失函数对应于后验估计中的似然函数,正则项对应于后验估计中的先验信息,两者的乘积对应于贝叶斯最大后验估计的形式。

分裂 Bregman 方法

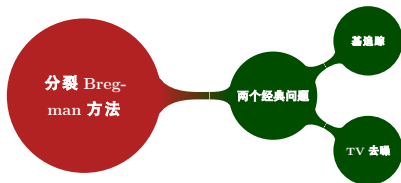


本章导图

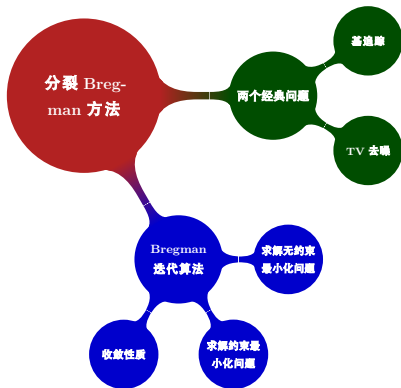


分裂 Breg-
man 方法

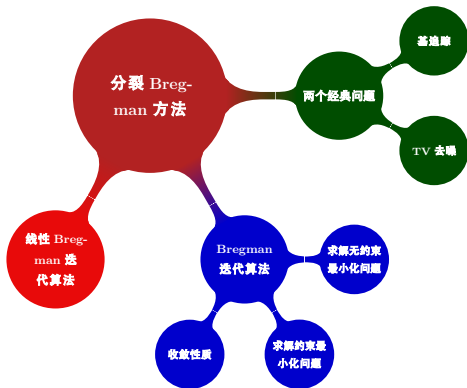
本章导图



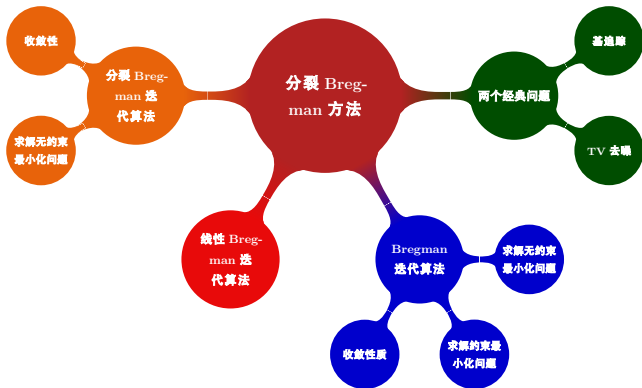
本章导图



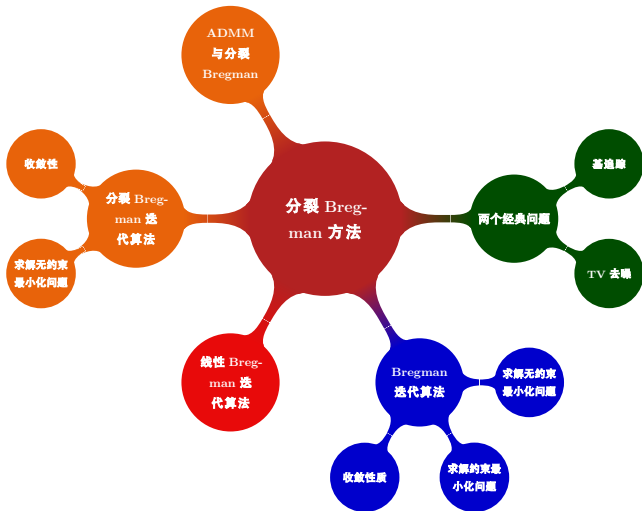
本章导图



本章导图



本章导图





两个经典问题 —— 基追踪问题

在基追踪问题中, 我们寻找具有最小 L_1 范数的线性方程的解, 即我们想解决约束问题

$$\min \|u\|_1 \quad \text{such that } Au = f \quad (5.1)$$

其中

$$\|u\|_1 = \sum_{i=1}^n |u_i|.$$

(5.1)中的约束导致了一些困难, 我们放松了约束, 需要解决无约束的基追踪问题

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} \|Au - f\|_2^2 + \mu \|u\|_1 \quad (5.2)$$

其中 $\mu \in \mathbb{R}$ 是一个正的常数。可以证明, 当 $\mu \rightarrow \infty$ 时, 无约束问题的解收敛于约束问题的解。

两个经典问题 —— 基追踪问题

在基追踪问题中, 我们寻找具有最小 L_1 范数的线性方程的解, 即我们想解决约束问题

$$\min \|u\|_1 \quad \text{such that } Au = f \quad (5.1)$$

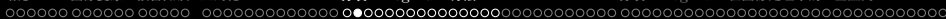
其中

$$\|u\|_1 = \sum_{i=1}^n |u_i|.$$

(5.1)中的约束导致了一些困难, 我们放松了约束, 需要解决无约束的基追踪问题

$$\min_{u \in \mathbb{R}^n} \frac{1}{2} \|Au - f\|_2^2 + \mu \|u\|_1 \quad (5.2)$$

其中 $\mu \in \mathbb{R}$ 是一个正的常数。可以证明, 当 $\mu \rightarrow \infty$ 时, 无约束问题的解收敛于约束问题的解。



两个经典问题 —— TV 去噪问题

我们想要恢复受噪声影响的图像, 这导致了如下形式的问题

$$\min_{u \in BV(\Omega)} \|u\|_{BV} + H(u). \quad (5.3)$$

其中

$$\|u\|_{BV} = \int_{\Omega} |\nabla u| dx = \|\nabla u\|_1.$$

我们尝试恢复原始图像通过最小化

$$\min_{u \in BV(\Omega)} \frac{\mu}{2} \|u - f\|_2^2 + \|\nabla u\|_1, \quad (5.4)$$

其中 $\mu > 0$ 是一个常数



两个经典问题 —— TV 去噪问题

我们想要恢复受噪声影响的图像, 这导致了如下形式的问题

$$\min_{u \in BV(\Omega)} \|u\|_{BV} + H(u). \quad (5.3)$$

其中

$$\|u\|_{BV} = \int_{\Omega} |\nabla u| dx = \|\nabla u\|_1.$$

我们尝试恢复原始图像通过最小化

$$\min_{u \in BV(\Omega)} \frac{\mu}{2} \|u - f\|_2^2 + \|\nabla u\|_1, \quad (5.4)$$

其中 $\mu > 0$ 是一个常数

Bregman 迭代算法

我们将介绍 Bregman 迭代方法, 并证明它可以用来解决各种各样的约束优化问题。

S.Osher 等人在 2005 年提出了 Bregman 迭代算法作为解决如下问题的有效算法

$$\min_u J(u) + H(u, f), \quad (5.5)$$

Osher 等人定义的 Bregman 迭代算法如下

算法 1 Bregman 迭代算法

initialize: $k = 0, u^0 = 0, p^0 = 0$.

while “ u^k not converge” **do**

$$u^{k+1} = \operatorname{argmin}_u D_J^{p^k}(u, u^k) + H(u)$$

$$p^{k+1} = p^k - \nabla H(u^{k+1}) \in \partial J(u^{k+1})$$

$$k = k + 1$$

end while

Bregman 迭代算法

很容易得到第一次迭代产生的结果为

$$u^1 = \min_{u \in X} J(u) + H(u). \quad (5.6)$$

因此该算法的第一次迭代解决了(5.5)。然而为了解决我们的初始问题,我们需要剩余项最小,这就是为什么 Bregman 迭代算法一直持续到剩余项收敛。

在处理基追踪问题时,我们让 $J(u) = \|u\|_1$ 和 $H(u) = \frac{1}{2}\|Au - f\|_2^2$ 。在研究图像去噪时,我们让 $J(u) = \|u\|_{BV}$ 或 $J(u) = \|\nabla u\|_1$, $H(u) = \frac{1}{2}\|u - f\|_2^2$ 。

Bregman 迭代算法

很容易得到第一次迭代产生的结果为

$$u^1 = \min_{u \in X} J(u) + H(u). \quad (5.6)$$

因此该算法的第一次迭代解决了(5.5)。然而为了解决我们的初始问题,我们需要剩余项最小,这就是为什么 Bregman 迭代算法一直持续到剩余项收敛。

在处理基追踪问题时,我们让 $J(u) = \|u\|_1$ 和 $H(u) = \frac{1}{2}\|Au - f\|_2^2$ 。在研究图像去噪时,我们让 $J(u) = \|u\|_{BV}$ 或 $J(u) = \|\nabla u\|_1$, $H(u) = \frac{1}{2}\|u - f\|_2^2$ 。

Bregman 迭代求解无约束最小化问题

我们考虑以下问题：

$$\min_u J(u) + \lambda H(u) \quad (5.7)$$

其中 λ 是正数。

我们可以修改这个问题, 迭代地解决：

$$\begin{aligned} u^{k+1} &= \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \lambda H(u) \\ &= \operatorname{argmin}_u J(u) - \langle p^k, u - u^k \rangle + \lambda H(u) \end{aligned} \quad (5.8)$$

有 $0 \in \partial \left(D_J^{p^k}(u, u^k) + \lambda H(u) \right)$, 如果 $p^{k+1} \in \partial J(u^{k+1})$, 那么

$$p^{k+1} = p^k - \lambda \nabla H(u^{k+1}) \quad (5.9)$$

为了简化, 令 $q^{k+1} = \lambda \nabla H(u^{k+1})$, 我们有 $p^{k+1} = p^k - q^{k+1}$.

Bregman 迭代求解无约束最小化问题

我们考虑以下问题：

$$\min_u J(u) + \lambda H(u) \quad (5.7)$$

其中 λ 是正数。

我们可以修改这个问题, 迭代地解决：

$$\begin{aligned} u^{k+1} &= \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \lambda H(u) \\ &= \operatorname{argmin}_u J(u) - \langle p^k, u - u^k \rangle + \lambda H(u) \end{aligned} \quad (5.8)$$

有 $0 \in \partial \left(D_J^{p^k}(u, u^k) + \lambda H(u) \right)$, 如果 $p^{k+1} \in \partial J(u^{k+1})$, 那么

$$p^{k+1} = p^k - \lambda \nabla H(u^{k+1}) \quad (5.9)$$

为了简化, 令 $q^{k+1} = \lambda \nabla H(u^{k+1})$, 我们有 $p^{k+1} = p^k - q^{k+1}$.

Bregman 迭代求解无约束最小化问题

我们考虑以下问题：

$$\min_u J(u) + \lambda H(u) \quad (5.7)$$

其中 λ 是正数。

我们可以修改这个问题, 迭代地解决：

$$\begin{aligned} u^{k+1} &= \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \lambda H(u) \\ &= \operatorname{argmin}_u J(u) - \langle p^k, u - u^k \rangle + \lambda H(u) \end{aligned} \quad (5.8)$$

有 $0 \in \partial \left(D_J^{p^k}(u, u^k) + \lambda H(u) \right)$, 如果 $p^{k+1} \in \partial J(u^{k+1})$, 那么

$$p^{k+1} = p^k - \lambda \nabla H(u^{k+1}) \quad (5.9)$$

为了简化, 令 $q^{k+1} = \lambda \nabla H(u^{k+1})$, 我们有 $p^{k+1} = p^k - q^{k+1}$.

Bregman 迭代求解无约束最小化问题

我们考虑以下问题：

$$\min_u J(u) + \lambda H(u) \quad (5.7)$$

其中 λ 是正数。

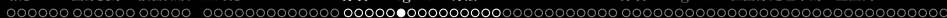
我们可以修改这个问题, 迭代地解决：

$$\begin{aligned} u^{k+1} &= \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \lambda H(u) \\ &= \operatorname{argmin}_u J(u) - \langle p^k, u - u^k \rangle + \lambda H(u) \end{aligned} \quad (5.8)$$

有 $0 \in \partial \left(D_J^{p^k}(u, u^k) + \lambda H(u) \right)$, 如果 $p^{k+1} \in \partial J(u^{k+1})$, 那么

$$p^{k+1} = p^k - \lambda \nabla H(u^{k+1}) \quad (5.9)$$

为了简化, 令 $q^{k+1} = \lambda \nabla H(u^{k+1})$, 我们有 $p^{k+1} = p^k - q^{k+1}$.



Bregman 迭代求解无约束最小化问题

然后, 如果 $p^0 = 0$, 我们有:

$$\begin{aligned} p^1 &= p^0 - q^1 = -q^1 \\ p^2 &= p^1 - q^2 = -q^1 - q^2 \\ &\vdots \end{aligned} \tag{5.10}$$

$$p^{k+1} = -\sum_{j=1}^{k+1} q^j$$

在文献 [3] 中, 作者分析了这个 Bregman 迭代格式的收敛性, 结果表明, 在 J 和 H 相当弱的假设下, 我们有

$$\lim_{k \rightarrow \infty} H(u^k) = 0 \tag{5.11}$$

Bregman 迭代求解无约束最小化问题

然后, 如果 $p^0 = 0$, 我们有:

$$\begin{aligned} p^1 &= p^0 - q^1 = -q^1 \\ p^2 &= p^1 - q^2 = -q^1 - q^2 \\ &\vdots \end{aligned} \tag{5.10}$$

$$p^{k+1} = -\sum_{j=1}^{k+1} q^j$$

在文献 [3] 中, 作者分析了这个 Bregman 迭代格式的收敛性, 结果表明, 在 J 和 H 相当弱的假设下, 我们有

$$\lim_{k \rightarrow \infty} H(u^k) = 0 \tag{5.11}$$

Bregman 迭代求解约束最小化问题

我们需要解决以下问题

$$\min_u J(u) \quad \text{such that} \quad Au = b \quad (5.12)$$

令 $H(u) = Au - b$, 使用二次惩罚函数将它变成一个无约束的问题, 因此有

$$\min_u \left\{ J(u) + \frac{\lambda}{2} \|Au - b\|_2^2 \right\} \quad (5.13)$$

这个问题的 Bregman 迭代是

$$\begin{aligned} u^{k+1} &= \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{\lambda}{2} \|Au - b\|_2^2 \\ &= \operatorname{argmin}_u J(u) - \langle p^k, u - u^k \rangle + \frac{\lambda}{2} \|Au - b\|_2^2 \end{aligned} \quad (5.14)$$

$$\begin{aligned} p^{k+1} &= p^k - \lambda \nabla H(u^{k+1}) \\ &= p^k - \lambda A^\top (Au^{k+1} - b) \end{aligned}$$

Bregman 迭代求解约束最小化问题

我们需要解决以下问题

$$\min_u J(u) \quad \text{such that} \quad Au = b \quad (5.12)$$

令 $H(u) = Au - b$, 使用二次惩罚函数将它变成一个无约束的问题, 因此有

$$\min_u \left\{ J(u) + \frac{\lambda}{2} \|Au - b\|_2^2 \right\} \quad (5.13)$$

这个问题的 Bregman 迭代是

$$\begin{aligned} u^{k+1} &= \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{\lambda}{2} \|Au - b\|_2^2 \\ &= \operatorname{argmin}_u J(u) - \langle p^k, u - u^k \rangle + \frac{\lambda}{2} \|Au - b\|_2^2 \end{aligned} \quad (5.14)$$

$$\begin{aligned} p^{k+1} &= p^k - \lambda \nabla H(u^{k+1}) \\ &= p^k - \lambda A^\top (Au^{k+1} - b) \end{aligned}$$

Bregman 迭代求解约束最小化问题

我们需要解决以下问题

$$\min_u J(u) \quad \text{such that} \quad Au = b \quad (5.12)$$

令 $H(u) = Au - b$, 使用二次惩罚函数将它变成一个无约束的问题, 因此有

$$\min_u \left\{ J(u) + \frac{\lambda}{2} \|Au - b\|_2^2 \right\} \quad (5.13)$$

这个问题的 Bregman 迭代是

$$\begin{aligned} u^{k+1} &= \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{\lambda}{2} \|Au - b\|_2^2 \\ &= \operatorname{argmin}_u J(u) - \langle p^k, u - u^k \rangle + \frac{\lambda}{2} \|Au - b\|_2^2 \\ p^{k+1} &= p^k - \lambda \nabla H(u^{k+1}) \\ &= p^k - \lambda A^\top (Au^{k+1} - b) \end{aligned} \quad (5.14)$$

Bregman 迭代求解约束最小化问题

在这种情况下, 由于 $H(u) = Au - b$, H 相对于 u 是线性的, 我们有

$$q^k = \lambda \nabla H(u^k) = \lambda A^\top (Au^k - b) \quad (5.15)$$

如果 $p^0 = 0, u^0 = 0$, 我们令 $p^k = \lambda A^\top (b^k - b)$, 且 $b^0 = b$, 我们有

$$\begin{aligned} p^{k+1} &= p^k - \lambda A^\top (Au^{k+1} - b) \\ \lambda A^\top b^{k+1} &= \lambda A^\top (b^k - Au^{k+1} + b) \end{aligned}$$

然后有

$$b^{k+1} = b^k + b - Au^k \quad (5.16)$$

Bregman 迭代求解约束最小化问题

在这种情况下, 由于 $H(u) = Au - b$, H 相对于 u 是线性的, 我们有

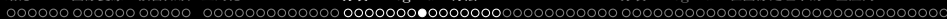
$$q^k = \lambda \nabla H(u^k) = \lambda A^\top (Au^k - b) \quad (5.15)$$

如果 $p^0 = 0, u^0 = 0$, 我们令 $p^k = \lambda A^\top (b^k - b)$, 且 $b^0 = b$, 我们有

$$\begin{aligned} p^{k+1} &= p^k - \lambda A^\top (Au^{k+1} - b) \\ \lambda A^\top b^{k+1} &= \lambda A^\top (b^k - Au^{k+1} + b) \end{aligned}$$

然后有

$$b^{k+1} = b^k + b - Au^k \quad (5.16)$$



Bregman 迭代求解约束最小化问题

在这种情况下, 由于 $H(u) = Au - b$, H 相对于 u 是线性的, 我们有

$$q^k = \lambda \nabla H(u^k) = \lambda A^\top (Au^k - b) \quad (5.15)$$

如果 $p^0 = 0, u^0 = 0$, 我们令 $p^k = \lambda A^\top (b^k - b)$, 且 $b^0 = b$, 我们有

$$\begin{aligned} p^{k+1} &= p^k - \lambda A^\top (Au^{k+1} - b) \\ \lambda A^\top b^{k+1} &= \lambda A^\top (b^k - Au^{k+1} + b) \end{aligned}$$

然后有

$$b^{k+1} = b^k + b - Au^k \quad (5.16)$$

Bregman 迭代求解约束最小化问题

在(5.14)中我们有：

$$\begin{aligned} & -p^k \top (u - u^k) + \frac{\lambda}{2} \|Au - b\|_2^2 \\ & = \lambda (b^k - b) \top (Au^k - Au) + \frac{\lambda}{2} \|Au - b\|_2^2 \end{aligned}$$

在梯度下, 我们有

$$\begin{aligned} & \nabla \left(\lambda (b^k - b) \top (Au^k - Au) + \frac{\lambda}{2} \|Au - b\|_2^2 \right) \\ & = \nabla \left(\frac{\lambda}{2} \|Au - b^k\|_2^2 \right) \end{aligned}$$

然后 Bregman 迭代可以写成

$$\begin{aligned} u^{k+1} & = \underset{u}{\operatorname{argmin}} J(u) + \frac{\lambda}{2} \|Au - b^k\|_2^2 \\ b^{k+1} & = b^k + b - Au^k \end{aligned} \tag{5.17}$$

Bregman 迭代求解约束最小化问题

在(5.14)中我们有：

$$\begin{aligned} & -p^k{}^\top (u - u^k) + \frac{\lambda}{2} \|Au - b\|_2^2 \\ & = \lambda (b^k - b)^\top (Au^k - Au) + \frac{\lambda}{2} \|Au - b\|_2^2 \end{aligned}$$

在梯度下, 我们有

$$\begin{aligned} & \nabla \left(\lambda (b^k - b)^\top (Au^k - Au) + \frac{\lambda}{2} \|Au - b\|_2^2 \right) \\ & = \nabla \left(\frac{\lambda}{2} \|Au - b^k\|_2^2 \right) \end{aligned}$$

然后 Bregman 迭代可以写成

$$\begin{aligned} u^{k+1} & = \underset{u}{\operatorname{argmin}} J(u) + \frac{\lambda}{2} \|Au - b^k\|_2^2 \\ b^{k+1} & = b^k + b - Au^k \end{aligned} \tag{5.17}$$

Bregman 迭代求解约束最小化问题

在(5.14)中我们有：

$$\begin{aligned} & -p^k{}^\top (u - u^k) + \frac{\lambda}{2} \|Au - b\|_2^2 \\ & = \lambda (b^k - b)^\top (Au^k - Au) + \frac{\lambda}{2} \|Au - b\|_2^2 \end{aligned}$$

在梯度下, 我们有

$$\begin{aligned} & \nabla \left(\lambda (b^k - b)^\top (Au^k - Au) + \frac{\lambda}{2} \|Au - b\|_2^2 \right) \\ & = \nabla \left(\frac{\lambda}{2} \|Au - b^k\|_2^2 \right) \end{aligned}$$

然后 Bregman 迭代可以写成

$$\begin{aligned} u^{k+1} & = \underset{u}{\operatorname{argmin}} J(u) + \frac{\lambda}{2} \|Au - b^k\|_2^2 \\ b^{k+1} & = b^k + b - Au^k \end{aligned} \tag{5.17}$$

Bregman 迭代求解约束最小化问题

由于 $H(u) = Au - b$, 无约束问题中的(5.11)现在变成

$$\lim_{k \rightarrow \infty} Au^k = b \quad (5.18)$$

换句话说, 对于较大的 k , 迭代 u^k 满足约束条件到任意高度的准确度。

以下定理显示了通过(5.17)得到的 $Au = b$ 的解 u^* 是原始约束问题(5.12)的解。

Theorem ([3])

让 $J: \mathbb{R}^n \rightarrow \mathbb{R}$ 是凸的, $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是线性的。考虑算法(5.17), 并假设一些迭代 u^k 满足 $Au^k = b$ 。那么 u^k 是原始约束问题(5.12)的解。

这个定理表明, 如果算法在(5.18)的意义上收敛, 则迭代 u^k 将任意地接近原始约束问题的解。

Bregman 迭代求解约束最小化问题

由于 $H(u) = Au - b$, 无约束问题中的(5.11)现在变成

$$\lim_{k \rightarrow \infty} Au^k = b \quad (5.18)$$

换句话说, 对于较大的 k , 迭代 u^k 满足约束条件到任意高度的准确度。

以下定理显示了通过(5.17)得到的 $Au = b$ 的解 u^* 是原始约束问题(5.12)的解。

Theorem ([3])

让 $J: \mathbb{R}^n \rightarrow \mathbb{R}$ 是凸的, $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是线性的。考虑算法(5.17), 并假设一些迭代 u^k 满足 $Au^k = b$ 。那么 u^k 是原始约束问题(5.12)的解。

这个定理表明, 如果算法在(5.18)的意义上收敛, 则迭代 u^k 将任意地接近原始约束问题的解。

Bregman 迭代求解约束最小化问题

由于 $H(u) = Au - b$, 无约束问题中的(5.11)现在变成

$$\lim_{k \rightarrow \infty} Au^k = b \quad (5.18)$$

换句话说, 对于较大的 k , 迭代 u^k 满足约束条件到任意高度的准确度。

以下定理显示了通过(5.17)得到的 $Au = b$ 的解 u^* 是原始约束问题(5.12)的解。

Theorem ([3])

让 $J: \mathbb{R}^n \rightarrow \mathbb{R}$ 是凸的, $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是线性的。考虑算法(5.17), 并假设一些迭代 u^k 满足 $Au^k = b$ 。那么 u^k 是原始约束问题(5.12)的解。

这个定理表明, 如果算法在(5.18)的意义上收敛, 则迭代 u^k 将任意地接近原始约束问题的解。

Bregman 迭代求解约束最小化问题

由于 $H(u) = Au - b$, 无约束问题中的(5.11)现在变成

$$\lim_{k \rightarrow \infty} Au^k = b \quad (5.18)$$

换句话说, 对于较大的 k , 迭代 u^k 满足约束条件到任意高度的准确度。

以下定理显示了通过(5.17)得到的 $Au = b$ 的解 u^* 是原始约束问题(5.12)的解。

Theorem ([3])

让 $J: \mathbb{R}^n \rightarrow \mathbb{R}$ 是凸的, $A: \mathbb{R}^n \rightarrow \mathbb{R}^m$ 是线性的。考虑算法(5.17), 并假设一些迭代 u^k 满足 $Au^k = b$ 。那么 u^k 是原始约束问题(5.12)的解。

这个定理表明, 如果算法在(5.18)的意义上收敛, 则迭代 u^k 将任意地接近原始约束问题的解。

Bregman 迭代算法的收敛性质

算法生成的序列在 H 中单调递减。

性质 (H 单调减少 [6])

$$H(u^{k+1}) \leq H(u^{k+1}) + D_J^{p^k}(u^{k+1}, u^k) \leq H(u^k)$$

残差项 $\{H(u^k)\}$ 的序列收敛于 H 的最小值。

性质 ([6])

如果 \tilde{u} 最小化 $H : X \rightarrow \mathbb{R}$ 和 $J(\tilde{u}) < \infty$, 那么

$$H(u^k) \leq H(\tilde{u}) + J(\tilde{u})/k.$$

Bregman 迭代算法的收敛性质

算法生成的序列在 H 中单调递减。

性质 (H 单调减少 [6])

$$H(u^{k+1}) \leq H(u^{k+1}) + D_J^{p^k}(u^{k+1}, u^k) \leq H(u^k)$$

残差项 $\{H(u^k)\}$ 的序列收敛于 H 的最小值。

性质 ([6])

如果 \tilde{u} 最小化 $H : X \rightarrow \mathbb{R}$ 和 $J(\tilde{u}) < \infty$, 那么

$$H(u^k) \leq H(\tilde{u}) + J(\tilde{u})/k.$$

Bregman 迭代算法的收敛性质

只要剩余项 $H(u^k; f) > \delta^2$, 那么原始信号或图像与当前迭代之间的 Bregman 距离就会减小。

性质 ([6])

对于一个开集 X , 令 $H : X \rightarrow \mathbb{R}$ 假设 $H(\tilde{u}; f) < \delta^2$ 且 $H(\tilde{u}; g) = 0$ (f, g, \tilde{u}, δ 分别表示噪声数据, 无噪数据, 完全恢复, 噪音水平)。那么只要 $H(u^{k+1}) > \delta^2$ 就有 $D_J^{p^{k+1}}(\tilde{u}, u^{k+1}) < D_J^{p^k}(\tilde{u}, u^k)$ 。

线性 Bregman 迭代算法

Bregman 算法是解决基追踪问题的好工具。然而, 算法在每一步都需要最小化

$$D_J^{p^k}(u, u^k) + H(u), \quad (5.19)$$

这在计算上可能是昂贵的。Wotao Yin 和 Osher 等人在 [6] 中介绍了 Bregman 迭代算法的线性化版本。根据矩阵函数的泰勒公式, 我们首先对 $H(u)$ 进行线性化, 给定 u^k 通过下式与 $H(u)$ 近似

$$H(u) = H(u^k) + \langle \nabla H(u^k), u - u^k \rangle. \quad (5.20)$$

由于这个近似只对 u 接近 u^k 准确, 因此 Wotao Yin 和 Osher 等人增加了惩罚项 $\frac{1}{2\delta} \|u - u^k\|_2^2$, 原来的问题被替换为

$$u^{k+1} = \underset{u}{\operatorname{argmin}} D_J^{p^k}(u, u^k) + H(u^k) + \langle \nabla H(u^k), u - u^k \rangle + \frac{1}{2\delta} \|u - u^k\|_2^2. \quad (5.21)$$

线性 Bregman 迭代算法

Bregman 算法是解决基追踪问题的好工具。然而, 算法在每一步都需要最小化

$$D_J^{p^k}(u, u^k) + H(u), \quad (5.19)$$

这在计算上可能是昂贵的。Wotao Yin 和 Osher 等人在 [6] 中介绍了 Bregman 迭代算法的线性化版本。根据矩阵函数的泰勒公式, 我们首先对 $H(u)$ 进行线性化, 给定 u^k 通过下式与 $H(u)$ 近似

$$H(u) = H(u^k) + \langle \nabla H(u^k), u - u^k \rangle. \quad (5.20)$$

由于这个近似只对 u 接近 u^k 准确, 因此 Wotao Yin 和 Osher 等人增加了惩罚项 $\frac{1}{2\delta} \|u - u^k\|_2^2$, 原来的问题被替换为

$$u^{k+1} = \underset{u}{\operatorname{argmin}} D_J^{p^k}(u, u^k) + H(u^k) + \langle \nabla H(u^k), u - u^k \rangle + \frac{1}{2\delta} \|u - u^k\|_2^2. \quad (5.21)$$

线性 Bregman 迭代算法

Bregman 算法是解决基追踪问题的好工具。然而, 算法在每一步都需要最小化

$$D_J^{p^k}(u, u^k) + H(u), \quad (5.19)$$

这在计算上可能是昂贵的。Wotao Yin 和 Osher 等人在 [6] 中介绍了 Bregman 迭代算法的线性化版本。根据矩阵函数的泰勒公式, 我们首先对 $H(u)$ 进行线性化, 给定 u^k 通过下式与 $H(u)$ 近似

$$H(u) = H(u^k) + \langle \nabla H(u^k), u - u^k \rangle. \quad (5.20)$$

由于这个近似只对 u 接近 u^k 准确, 因此 Wotao Yin 和 Osher 等人增加了惩罚项 $\frac{1}{2\delta} \|u - u^k\|_2^2$, 原来的问题被替换为

$$u^{k+1} = \underset{u}{\operatorname{argmin}} D_J^{p^k}(u, u^k) + H(u^k) + \langle \nabla H(u^k), u - u^k \rangle + \frac{1}{2\delta} \|u - u^k\|_2^2. \quad (5.21)$$

线性 Bregman 迭代算法

请注意, 惩罚项也使得目标函数有下界, 并且

$$\|(u - u^k) + \delta \nabla H(u^k)\|_2^2 = \|u - u^k\|_2^2 + 2\delta \langle \nabla H(u^k), u - u^k \rangle + \delta^2 \|\nabla H(u^k)\|_2^2.$$

此外, 观察到 $\|\nabla H(u^k)\|_2^2$ 和 $\|\nabla H(u^k)\|_2$ 是关于 u 的常数。因此, 迭代(5.21)等价于迭代

$$u^{k+1} = \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{1}{2\delta} \|u - (u^k - \delta \nabla H(u^k))\|_2^2. \quad (5.22)$$

我们将这种方法应用于基追踪问题(5.2), 并让 $H(u) = \frac{1}{2} \|Au - f\|_2^2$. 把这个 $H(u)$ 代入到(5.22)我们可以得到

$$u^{k+1} = \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{1}{2\delta} \|u - (u^k - \delta A^\top (Au^k - f))\|_2^2. \quad (5.23)$$

线性 Bregman 迭代算法

请注意, 惩罚项也使得目标函数有下界, 并且

$$\|(u - u^k) + \delta \nabla H(u^k)\|_2^2 = \|u - u^k\|_2^2 + 2\delta \langle \nabla H(u^k), u - u^k \rangle + \delta^2 \|\nabla H(u^k)\|_2^2.$$

此外, 观察到 $\|\nabla H(u^k)\|_2^2$ 和 $\|\nabla H(u^k)\|_2$ 是关于 u 的常数。因此, 迭代(5.21)等价于迭代

$$u^{k+1} = \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{1}{2\delta} \|u - (u^k - \delta \nabla H(u^k))\|_2^2. \quad (5.22)$$

我们将这种方法应用于基追踪问题(5.2), 并让 $H(u) = \frac{1}{2} \|Au - f\|_2^2$. 把这个 $H(u)$ 代入到(5.22)我们可以得到

$$u^{k+1} = \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{1}{2\delta} \|u - (u^k - \delta A^\top (Au^k - f))\|_2^2. \quad (5.23)$$

线性 Bregman 迭代算法

请注意, 惩罚项也使得目标函数有下界, 并且

$$\|(u - u^k) + \delta \nabla H(u^k)\|_2^2 = \|u - u^k\|_2^2 + 2\delta \langle \nabla H(u^k), u - u^k \rangle + \delta^2 \|\nabla H(u^k)\|_2^2.$$

此外, 观察到 $\|\nabla H(u^k)\|_2^2$ 和 $\|\nabla H(u^k)\|_2$ 是关于 u 的常数。因此, 迭代(5.21)等价于迭代

$$u^{k+1} = \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{1}{2\delta} \|u - (u^k - \delta \nabla H(u^k))\|_2^2. \quad (5.22)$$

我们将这种方法应用于基追踪问题(5.2), 并让 $H(u) = \frac{1}{2} \|Au - f\|_2^2$. 把这个 $H(u)$ 代入到(5.22)我们可以得到

$$u^{k+1} = \operatorname{argmin}_u D_J^{p^k}(u, u^k) + \frac{1}{2\delta} \|u - (u^k - \delta A^\top (Au^k - f))\|_2^2. \quad (5.23)$$

线性 Bregman 迭代算法

为了便于记忆, 定义收缩函数如下^[16]: 对于 $\mu \geq 0$

$$\text{shrink}(x, \mu) = \begin{cases} x - \mu, & x \in (\mu, \infty) \\ 0, & x \in [-\mu, \mu] \\ x + \mu, & x \in (-\infty, -\mu) \end{cases}$$

它遵循

$$u_i^{k+1} = \delta \text{shrink}(v_i^k, \mu).$$

综上所述, 基追踪问题的线性 Bregman 迭代算法可以写成

算法 2 基追踪问题的线性 Bregman 迭代算法

initialize: $u = 0, v = 0.$

while “ $\|Au - f\|$ does not converge” **do**

$$v^{k+1} = v^k + A^\top (f - Au^k)$$

$$u^{k+1} = \delta \text{shrink}(v_i^{k+1}, \mu)$$

end while

分裂 Bregman 迭代算法

在 [9] 中, T.Goldstein 和 S.Osher 介绍了分裂 Bregman 方法来解决如下形式的一般优化问题

$$\min_{u \in X} |\Phi(u)| + H(u) \quad (5.24)$$

他们方法的关键是解耦(5.24)中能量的 L1 和 L2 部分。然后我们考虑这个约束最小化问题

$$\min_{u,d} \{|d| + H(u)\} \quad \text{such that} \quad d = \Phi(u) \quad (5.25)$$

像基追踪问题一样, 我们放宽约束条件, 并将约束优化问题(5.25)转换为无约束问题

$$\min_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \quad (5.26)$$

分裂 Bregman 迭代算法

在 [9] 中, T.Goldstein 和 S.Osher 介绍了分裂 Bregman 方法来解决如下形式的一般优化问题

$$\min_{u \in X} |\Phi(u)| + H(u) \quad (5.24)$$

他们方法的关键是解耦(5.24)中能量的 L1 和 L2 部分。然后我们考虑这个约束最小化问题

$$\min_{u,d} \{|d| + H(u)\} \quad \text{such that} \quad d = \Phi(u) \quad (5.25)$$

像基追踪问题一样, 我们放宽约束条件, 并将约束优化问题(5.25)转换为无约束问题

$$\min_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \quad (5.26)$$

分裂 Bregman 迭代算法

在 [9] 中, T.Goldstein 和 S.Osher 介绍了分裂 Bregman 方法来解决如下形式的一般优化问题

$$\min_{u \in X} |\Phi(u)| + H(u) \quad (5.24)$$

他们方法的关键是解耦(5.24)中能量的 L1 和 L2 部分。然后我们考虑这个约束最小化问题

$$\min_{u,d} \{|d| + H(u)\} \quad \text{such that} \quad d = \Phi(u) \quad (5.25)$$

像基追踪问题一样, 我们放宽约束条件, 并将约束优化问题(5.25)转换为无约束问题

$$\min_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \quad (5.26)$$

分裂 Bregman 迭代算法

让 $J(u, d) = |d| + H(u)$, $A(u, d) = d - \Phi(u)$, 可以把(5.26)改写为

$$\min_{u, d} J(u, d) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \quad (5.27)$$

因此, 我们可以通过使用迭代 Bregman 算法来求解(5.27), 该算法产生以下序列:

$$\begin{aligned} (u^{k+1}, d^{k+1}) &= \operatorname{argmin}_{u, d} D_J^{p^k} (u, u^k, d, d^k) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \\ &= \operatorname{argmin}_{u, d} J(u, d) - \langle p_u^k, u - u^k \rangle - \langle p_d^k, d - d^k \rangle + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \\ p_u^{k+1} &= p_u^k - \lambda (\nabla \Phi)^\top (\Phi(u^{k+1}) - d^{k+1}) \\ p_d^{k+1} &= p_d^k - \lambda (d^{k+1} - \Phi(u^{k+1})) \end{aligned} \quad (5.28)$$



分裂 Bregman 迭代算法

让 $J(u, d) = |d| + H(u)$, $A(u, d) = d - \Phi(u)$, 可以把(5.26)改写为

$$\min_{u, d} J(u, d) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \quad (5.27)$$

因此, 我们可以通过使用迭代 Bregman 算法来求解(5.27), 该算法产生以下序列:

$$\begin{aligned} (u^{k+1}, d^{k+1}) &= \operatorname{argmin}_{u, d} D_J^{p^k} (u, u^k, d, d^k) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \\ &= \operatorname{argmin}_{u, d} J(u, d) - \langle p_u^k, u - u^k \rangle - \langle p_d^k, d - d^k \rangle + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2 \\ p_u^{k+1} &= p_u^k - \lambda (\nabla \Phi)^\top (\Phi(u^{k+1}) - d^{k+1}) \\ p_d^{k+1} &= p_d^k - \lambda (d^{k+1} - \Phi(u^{k+1})) \end{aligned} \quad (5.28)$$



分裂 Bregman 迭代算法

注意到

$$p_u^{k+1} = p_u^k - \lambda(\nabla\Phi)^\top (\Phi(u^{k+1}) - d^{k+1}) \quad (5.29)$$

$$= -\lambda(\nabla\Phi)^\top \sum_{i=1}^{k+1} (\Phi(u^i) - d^i), \quad (5.30)$$

和

$$p_d^{k+1} = p_d^k - \lambda(d^{k+1} - \Phi(u^{k+1})) = \lambda \sum_{i=1}^{k+1} (\Phi(u^i) - d^i). \quad (5.31)$$

此外, (5.30)和(5.31)有一个公共的求和项, 把它重新定义如下:

$$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1}) = \sum_{i=1}^{k+1} (\Phi(u^i) - d^i). \quad (5.32)$$



分裂 Bregman 迭代算法

注意到

$$p_u^{k+1} = p_u^k - \lambda(\nabla\Phi)^\top (\Phi(u^{k+1}) - d^{k+1}) \quad (5.29)$$

$$= -\lambda(\nabla\Phi)^\top \sum_{i=1}^{k+1} (\Phi(u^i) - d^i), \quad (5.30)$$

和

$$p_d^{k+1} = p_d^k - \lambda(d^{k+1} - \Phi(u^{k+1})) = \lambda \sum_{i=1}^{k+1} (\Phi(u^i) - d^i). \quad (5.31)$$

此外, (5.30)和(5.31)有一个公共的求和项, 把它重新定义如下:

$$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1}) = \sum_{i=1}^{k+1} (\Phi(u^i) - d^i). \quad (5.32)$$



分裂 Bregman 迭代算法

然后我们可以得到

$$p_u^{k+1} = -\lambda(\nabla\Phi)^\top b^{k+1} \quad (5.33)$$

$$p_d^{k+1} = \lambda b^{k+1} \quad (5.34)$$

应用(5.33)和(5.34)到(5.28), 我们有

$$\begin{aligned} (u^{k+1}, d^{k+1}) &= \operatorname{argmin}_{u,d} J(u, d) + \lambda \langle b^k, \Phi u - \Phi u^k \rangle - \lambda \langle b^k, d - d^k \rangle + \frac{\lambda}{2} \|d - \Phi u\|_2^2 \\ &= \operatorname{argmin}_{u,d} J(u, d) + \frac{\lambda}{2} \|d - \Phi u - b^k\|_2^2 + C_2 \end{aligned}$$

现在我们就有了 Bregman 迭代:

$$(u^{k+1}, d^{k+1}) = \operatorname{argmin}_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u) - b^k\|_2^2 \quad (5.35)$$

$$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1}) \quad (5.36)$$



分裂 Bregman 迭代算法

然后我们可以得到

$$p_u^{k+1} = -\lambda(\nabla\Phi)^\top b^{k+1} \quad (5.33)$$

$$p_d^{k+1} = \lambda b^{k+1} \quad (5.34)$$

应用(5.33)和(5.34)到(5.28), 我们有

$$\begin{aligned} (u^{k+1}, d^{k+1}) &= \operatorname{argmin}_{u,d} J(u, d) + \lambda\langle b^k, \Phi u - \Phi u^k \rangle - \lambda\langle b^k, d - d^k \rangle + \frac{\lambda}{2} \|d - \Phi u\|_2^2 \\ &= \operatorname{argmin}_{u,d} J(u, d) + \frac{\lambda}{2} \|d - \Phi u - b^k\|_2^2 + C_2 \end{aligned}$$

现在我们有 Bregman 迭代:

$$(u^{k+1}, d^{k+1}) = \operatorname{argmin}_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u) - b^k\|_2^2 \quad (5.35)$$

$$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1}) \quad (5.36)$$

分裂 Bregman 迭代算法

然后我们可以得到

$$p_u^{k+1} = -\lambda(\nabla\Phi)^\top b^{k+1} \quad (5.33)$$

$$p_d^{k+1} = \lambda b^{k+1} \quad (5.34)$$

应用(5.33)和(5.34)到(5.28), 我们有

$$\begin{aligned} (u^{k+1}, d^{k+1}) &= \operatorname{argmin}_{u,d} J(u, d) + \lambda\langle b^k, \Phi u - \Phi u^k \rangle - \lambda\langle b^k, d - d^k \rangle + \frac{\lambda}{2} \|d - \Phi u\|_2^2 \\ &= \operatorname{argmin}_{u,d} J(u, d) + \frac{\lambda}{2} \|d - \Phi u - b^k\|_2^2 + C_2 \end{aligned}$$

现在我们有 Bregman 迭代:

$$(u^{k+1}, d^{k+1}) = \operatorname{argmin}_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u) - b^k\|_2^2 \quad (5.35)$$

$$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1}) \quad (5.36)$$



分裂 Bregman 迭代算法

为了实现这个算法, 我们必须能够解决问题(5.35), 已经分解了这个函数的 L1 和 L2 分量, 我们可以通过迭代地最小化 u 和 d 来有效地执行这个最小化:

$$\text{Step 1: } u^{k+1} = \underset{u}{\operatorname{argmin}} H(u) + \frac{\lambda}{2} \|d^k - \Phi(u) - b^k\|_2^2 \quad (5.37)$$

$$\text{Step 2: } d^{k+1} = \underset{d}{\operatorname{argmin}} |d| + \frac{\lambda}{2} \|d - \Phi(u^{k+1}) - b^k\|_2^2 \quad (5.38)$$

该方案 (5.37-5.38-5.36) 被称为分裂 Bregman 迭代^[9]。

将 u 从 L1 问题解耦后, 步骤 1 现在是不同的。我们可以使用收缩算子明确计算 d 的最优值:

$$d_j^{k+1} = \operatorname{shrink} \left(\Phi(u)_j + b_j^k, \frac{1}{\lambda} \right)$$

其中

$$d = \operatorname{shrink}(u, \gamma) = \frac{z}{|z|} \max(|z| - \gamma, 0) \quad (5.39)$$



分裂 Bregman 迭代算法

为了实现这个算法, 我们必须能够解决问题(5.35), 已经分解了这个函数的 L1 和 L2 分量, 我们可以通过迭代地最小化 u 和 d 来有效地执行这个最小化:

$$\text{Step 1: } u^{k+1} = \underset{u}{\operatorname{argmin}} H(u) + \frac{\lambda}{2} \|d^k - \Phi(u) - b^k\|_2^2 \quad (5.37)$$

$$\text{Step 2: } d^{k+1} = \underset{d}{\operatorname{argmin}} |d| + \frac{\lambda}{2} \|d - \Phi(u^{k+1}) - b^k\|_2^2 \quad (5.38)$$

该方案 (5.37-5.38-5.36) 被称为分裂 Bregman 迭代^[9]。

将 u 从 L1 问题解耦后, 步骤 1 现在是不同的。我们可以使用收缩算子明确计算 d 的最优值:

$$d_j^{k+1} = \operatorname{shrink} \left(\Phi(u)_j + b_j^k, \frac{1}{\lambda} \right)$$

其中

$$d = \operatorname{shrink}(u, \gamma) = \frac{z}{|z|} \max(|z| - \gamma, 0) \quad (5.39)$$

分裂 Bregman 迭代算法

为了实现这个算法, 我们必须能够解决问题(5.35), 已经分解了这个函数的 L1 和 L2 分量, 我们可以通过迭代地最小化 u 和 d 来有效地执行这个最小化:

$$\text{Step 1: } u^{k+1} = \underset{u}{\operatorname{argmin}} H(u) + \frac{\lambda}{2} \|d^k - \Phi(u) - b^k\|_2^2 \quad (5.37)$$

$$\text{Step 2: } d^{k+1} = \underset{d}{\operatorname{argmin}} |d| + \frac{\lambda}{2} \|d - \Phi(u^{k+1}) - b^k\|_2^2 \quad (5.38)$$

该方案 (5.37-5.38-5.36) 被称为分裂 Bregman 迭代^[9]。

将 u 从 L1 问题解耦后, 步骤 1 现在是不同的。我们可以使用收缩算子明确计算 d 的最优值:

$$d_j^{k+1} = \operatorname{shrink} \left(\Phi(u)_j + b_j^k, \frac{1}{\lambda} \right)$$

其中

$$d = \operatorname{shrink}(u, \gamma) = \frac{z}{|z|} \max(|z| - \gamma, 0) \quad (5.39)$$

分裂 Bregman 迭代算法

算法 3 分裂 Bregman 迭代算法

initialize: $k = 0, u^0 = 0, b^0 = 0.$

while $\|u^k - u^{k-1}\|_2 > tol$ **do**

for $n = 1$ to N **do**

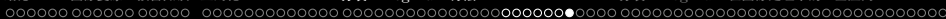
$$u^{k+1} = \underset{u}{\operatorname{argmin}} \left\{ H(u) + \frac{\lambda}{2} \|d^k - \Phi(u) - b^k\|_2^2 \right\}$$

$$d^{k+1} = \underset{d}{\operatorname{argmin}} \left\{ |d| + \frac{\lambda}{2} \|d - \Phi(u^{k+1}) - b^k\|_2^2 \right\}$$

end for

$$b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1})$$

end while



分裂 Bregman 算法的收敛性

令 (u^*, d^*, b^*) 为 (5.35-5.36) 的一个不动点。不动点满足 $b^* = b^* + \Phi(u^*) - d^*$, 这意味着 $d^* = \Phi(u^*)$ 。这个结果与(5.35)相结合, 满足 theorem 1 的条件, 表明 (u^*, d^*) 是约束问题(5.25)的一个解。

ADMM 与分裂 Bregman —— 增广拉格朗日法和乘子法

设有如下优化问题：

$$\min_x f(x) \quad \text{s.t. } Ax = b \quad (5.40)$$

(5.40)的增广拉格朗日函数为

$$L_\rho(x, y) = f(x) + y^\top (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2$$

其中 ρ 是一个惩罚参数。惩罚项 $\frac{\rho}{2} \|Ax - b\|_2^2$ 的好处是使得对偶函数 $g_\rho(y) = \inf_x L_\rho(x, y)$ 在更一般的条件下可导, 且 $\rho \rightarrow 0$ 对应于原始问题的解决方案。那么该问题的迭代公式为

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, y^k) \\ y^{k+1} &= y^k + \rho (Ax^{k+1} - b) \end{aligned} \quad (5.41)$$

该算法被称为解决问题(5.40)的乘子法(Method of Multipliers)。

ADMM 与分裂 Bregman —— 增广拉格朗日法和乘子法

设有如下优化问题：

$$\min_x f(x) \quad \text{s.t. } Ax = b \quad (5.40)$$

(5.40)的增广拉格朗日函数为

$$L_\rho(x, y) = f(x) + y^\top (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2$$

其中 ρ 是一个惩罚参数。惩罚项 $\frac{\rho}{2} \|Ax - b\|_2^2$ 的好处是使得对偶函数 $g_\rho(y) = \inf_x L_\rho(x, y)$ 在更一般的条件下可导, 且 $\rho \rightarrow 0$ 对应于原始问题的解决方案。那么该问题的迭代公式为

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, y^k) \\ y^{k+1} &= y^k + \rho (Ax^{k+1} - b) \end{aligned} \quad (5.41)$$

该算法被称为解决问题(5.40)的乘子法(Method of Multipliers)。

ADMM 与分裂 Bregman —— 增广拉格朗日法和乘子法

设有如下优化问题：

$$\min_x f(x) \quad \text{s.t. } Ax = b \quad (5.40)$$

(5.40)的增广拉格朗日函数为

$$L_\rho(x, y) = f(x) + y^\top (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2$$

其中 ρ 是一个惩罚参数。惩罚项 $\frac{\rho}{2} \|Ax - b\|_2^2$ 的好处是使得对偶函数 $g_\rho(y) = \inf_x L_\rho(x, y)$ 在更一般的条件下可导, 且 $\rho \rightarrow 0$ 对应于原始问题的解决方案。那么该问题的迭代公式为

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, y^k) \\ y^{k+1} &= y^k + \rho (Ax^{k+1} - b) \end{aligned} \quad (5.41)$$

该算法被称为解决问题(5.40)的乘子法(Method of Multipliers)。

ADMM 与分裂 Bregman —— 增广拉格朗日法和乘子法

设有如下优化问题：

$$\min_x f(x) \quad \text{s.t. } Ax = b \quad (5.40)$$

(5.40)的增广拉格朗日函数为

$$L_\rho(x, y) = f(x) + y^\top (Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2$$

其中 ρ 是一个惩罚参数。惩罚项 $\frac{\rho}{2} \|Ax - b\|_2^2$ 的好处是使得对偶函数 $g_\rho(y) = \inf_x L_\rho(x, y)$ 在更一般的条件下可导, 且 $\rho \rightarrow 0$ 对应于原始问题的解决方案。那么该问题的迭代公式为

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, y^k) \\ y^{k+1} &= y^k + \rho (Ax^{k+1} - b) \end{aligned} \quad (5.41)$$

该算法被称为解决问题(5.40)的乘子法(Method of Multipliers)。

ADMM 与分裂 Bregman —— ADMM 算法概述

交替方向乘子法 (ADMM) 解决了如下形式的约束优化问题

$$\operatorname{argmin}_x f(x) + g(z) \quad \text{s.t. } Ux + Vz = c \quad (5.42)$$

ADMM 由迭代最小化增广拉格朗日函数构成

$$L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ux + Vz - c) + \frac{\rho}{2} \|Ux + Vz - c\|_2^2 \quad (5.43)$$

y 是一个拉格朗日乘子, 增广拉格朗日参数 ρ 是我们可以选择使 $g(z)$ 平滑的一个参数。

那么 ADMM 的迭代方式为

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, z^k, y^k) \\ z^{k+1} &= \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &= y^k + \rho (Ux^{k+1} + Vz^{k+1} - c) \end{aligned} \quad (5.44)$$

ADMM 与分裂 Bregman —— ADMM 算法概述

交替方向乘子法 (ADMM) 解决了如下形式的约束优化问题

$$\operatorname{argmin}_x f(x) + g(z) \quad \text{s.t. } Ux + Vz = c \quad (5.42)$$

ADMM 由迭代最小化增广拉格朗日函数构成

$$L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ux + Vz - c) + \frac{\rho}{2} \|Ux + Vz - c\|_2^2 \quad (5.43)$$

y 是一个拉格朗日乘子, 增广拉格朗日参数 ρ 是我们可以选择使 $g(z)$ 平滑的一个参数。

那么 ADMM 的迭代方式为

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, z^k, y^k) \\ z^{k+1} &= \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &= y^k + \rho (Ux^{k+1} + Vz^{k+1} - c) \end{aligned} \quad (5.44)$$

ADMM 与分裂 Bregman —— ADMM 算法概述

交替方向乘子法 (ADMM) 解决了如下形式的约束优化问题

$$\operatorname{argmin}_x f(x) + g(z) \quad \text{s.t. } Ux + Vz = c \quad (5.42)$$

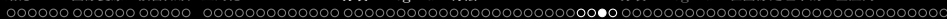
ADMM 由迭代最小化增广拉格朗日函数构成

$$L_\rho(x, z, y) = f(x) + g(z) + y^\top (Ux + Vz - c) + \frac{\rho}{2} \|Ux + Vz - c\|_2^2 \quad (5.43)$$

y 是一个拉格朗日乘子, 增广拉格朗日参数 ρ 是我们可以选择使 $g(z)$ 平滑的一个参数。

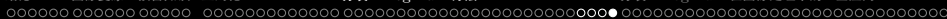
那么 ADMM 的迭代方式为

$$\begin{aligned} x^{k+1} &= \operatorname{argmin}_x L_\rho(x, z^k, y^k) \\ z^{k+1} &= \operatorname{argmin}_z L_\rho(x^{k+1}, z, y^k) \\ y^{k+1} &= y^k + \rho (Ux^{k+1} + Vz^{k+1} - c) \end{aligned} \quad (5.44)$$



ADMM 与分裂 Bregman —— ADMM 算法概述

交替最小化由于目标函数的可分解性而起作用： x 最小化步骤独立于 z 最小化步骤，反之亦然。后面我们可以看到这种拆分思想非常适合统计学习中的 L1 范数等问题：损失函数 + 正则项。

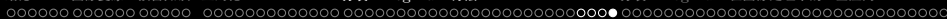


ADMM 与分裂 Bregman

Bregman 迭代是一类与交替方向乘子法 (ADMM) 相关的算法以及类似的基于拉格朗日方法的算法, 该方法包含受等式约束的凸函数求和^[14]。

已经证明, 当约束是线性时, 经典的 Bregman 迭代类似于增广拉格朗日方法^[6], 分裂 Bregman 算法被证明与 ADMM 算法密切相关^[14], 在有些情况下等价。

特别地, HUN NIEN 等人还指出, 对于 L1 范数正则化的最小二乘拟合问题, 分裂 Bregman 就是 ADMM 的特例^[37]。

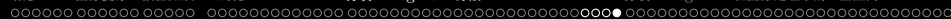


ADMM 与分裂 Bregman

Bregman 迭代是一类与交替方向乘子法 (ADMM) 相关的算法以及类似的基于拉格朗日方法的算法, 该方法包含受等式约束的凸函数求和^[14]。

已经证明, 当约束是线性时, 经典的 Bregman 迭代类似于增广拉格朗日方法^[6], 分裂 Bregman 算法被证明与 ADMM 算法密切相关^[14], 在有些情况下等价。

特别地, HUN NIEN 等人还指出, 对于 L1 范数正则化的最小二乘拟合问题, 分裂 Bregman 就是 ADMM 的特例^[37]。



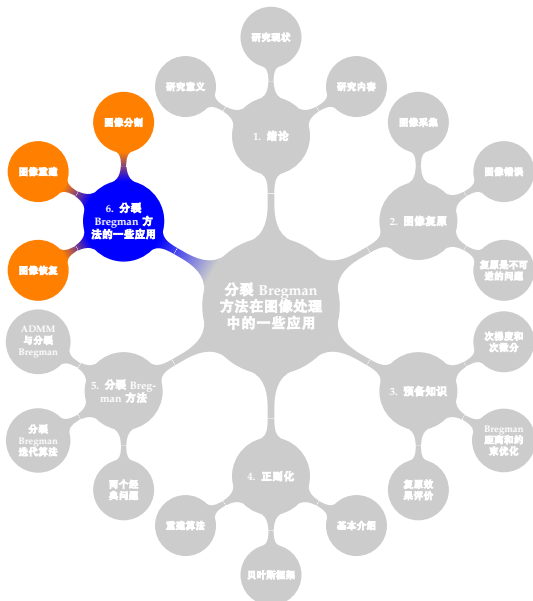
ADMM 与分裂 Bregman

Bregman 迭代是一类与交替方向乘子法 (ADMM) 相关的算法以及类似的基于拉格朗日方法的算法, 该方法包含受等式约束的凸函数求和^[14]。

已经证明, 当约束是线性时, 经典的 Bregman 迭代类似于增广拉格朗日方法^[6], 分裂 Bregman 算法被证明与 ADMM 算法密切相关^[14], 在有些情况下等价。

特别地, HUN NIEN 等人还指出, 对于 L1 范数正则化的最小二乘拟合问题, 分裂 Bregman 就是 ADMM 的特例^[37]。

分裂 Bregman 在图像处理中的一些应用



分裂 Bregman 应用广泛





稀疏信号重建 —— 基本原理

在这个应用中, 我们希望从被污染的信号 f 中恢复 u , 假设 u 被一个已知的算子 A 所改变。那么恢复 u 的一个有用的方法是使用 L1 最小化方案:

$$\operatorname{argmin}_u |u| + \frac{\mu}{2} \|Au - f\|_2^2$$

通过设置 $\Phi(u) = I$ (恒等式), $H(u) = \frac{\mu}{2} \|Au - f\|_2^2$, 应用分裂 Bregman 算法有

$$\begin{cases} u^{k+1} = \operatorname{argmin}_u \frac{\mu}{2} \|Au - f\|_2^2 + \frac{\lambda}{2} \|d^k - u - b^k\|_2^2 \\ d^{k+1} = \operatorname{argmin}_d |d| + \frac{\lambda}{2} \|d - u^{k+1} - b^k\|_2^2 \\ b^{k+1} = b^k + u^{k+1} - d^{k+1} \end{cases}$$

解决上式第一个关于 u 的子问题:

$$\begin{aligned} \partial_u \left(\frac{\mu}{2} \|Au - f\|_2^2 + \frac{\lambda}{2} \|d^k - u - b^k\|_2^2 \right) &= 0 \\ \Leftrightarrow \mu A^\top (Au^{k+1} - f) - \lambda (d^k - u^{k+1} - b^k) &= 0 \\ \Leftrightarrow u^{k+1} &= (\mu A^\top A + \lambda I)^{-1} (\mu A^\top f + \lambda (d^k - b^k)) \end{aligned}$$

稀疏信号重建 —— 基本原理

在这个应用中, 我们希望从被污染的信号 f 中恢复 u , 假设 u 被一个已知的算子 A 所改变。那么恢复 u 的一个有用的方法是使用 L1 最小化方案:

$$\operatorname{argmin}_u |u| + \frac{\mu}{2} \|Au - f\|_2^2$$

通过设置 $\Phi(u) = I$ (恒等式), $H(u) = \frac{\mu}{2} \|Au - f\|_2^2$, 应用分裂 Bregman 算法有

$$\begin{cases} u^{k+1} = \operatorname{argmin}_u \frac{\mu}{2} \|Au - f\|_2^2 + \frac{\lambda}{2} \|d^k - u - b^k\|_2^2 \\ d^{k+1} = \operatorname{argmin}_d |d| + \frac{\lambda}{2} \|d - u^{k+1} - b^k\|_2^2 \\ b^{k+1} = b^k + u^{k+1} - d^{k+1} \end{cases}$$

解决上式第一个关于 u 的子问题:

$$\begin{aligned} \partial_u \left(\frac{\mu}{2} \|Au - f\|_2^2 + \frac{\lambda}{2} \|d^k - u - b^k\|_2^2 \right) &= 0 \\ \Leftrightarrow \mu A^\top (Au^{k+1} - f) - \lambda (d^k - u^{k+1} - b^k) &= 0 \\ \Leftrightarrow u^{k+1} &= (\mu A^\top A + \lambda I)^{-1} (\mu A^\top f + \lambda (d^k - b^k)) \end{aligned}$$



稀疏信号重建 —— 基本原理

在这个应用中, 我们希望从被污染的信号 f 中恢复 u , 假设 u 被一个已知的算子 A 所改变。那么恢复 u 的一个有用的方法是使用 L1 最小化方案:

$$\operatorname{argmin}_u |u| + \frac{\mu}{2} \|Au - f\|_2^2$$

通过设置 $\Phi(u) = I$ (恒等式), $H(u) = \frac{\mu}{2} \|Au - f\|_2^2$, 应用分裂 Bregman 算法有

$$\begin{cases} u^{k+1} = \operatorname{argmin}_u \frac{\mu}{2} \|Au - f\|_2^2 + \frac{\lambda}{2} \|d^k - u - b^k\|_2^2 \\ d^{k+1} = \operatorname{argmin}_d |d| + \frac{\lambda}{2} \|d - u^{k+1} - b^k\|_2^2 \\ b^{k+1} = b^k + u^{k+1} - d^{k+1} \end{cases}$$

解决上式第一个关于 u 的子问题:

$$\begin{aligned} \partial_u \left(\frac{\mu}{2} \|Au - f\|_2^2 + \frac{\lambda}{2} \|d^k - u - b^k\|_2^2 \right) &= 0 \\ \Leftrightarrow \mu A^\top (Au^{k+1} - f) - \lambda (d^k - u^{k+1} - b^k) &= 0 \\ \Leftrightarrow u^{k+1} &= (\mu A^\top A + \lambda I)^{-1} (\mu A^\top f + \lambda (d^k - b^k)) \end{aligned}$$

稀疏信号重建 —— 基本原理

解决第二个关于 d 的子问题通过收缩算子：

$$d^{k+1} = \text{shrink}(u^{k+1} + b^k, 1/\lambda)$$

算法 4 分裂 Bregman 用于稀疏信号重建

initialize: $k = 0, u^0 = 0, b^0 = 0.$

while $\|u^k - u^{k-1}\|_2 > \text{tol}$ **do**

for $n = 1$ to N **do**

$$u^{k+1} = (\mu A^\top A + \lambda I)^{-1}(\mu A^\top f + \lambda(d^k - b^k))$$

$$d^{k+1} = \text{shrink}(u^{k+1} + b^k, 1/\lambda)$$

end for

$$b^{k+1} = b^k + u^{k+1} - d^{k+1}$$

end while



稀疏信号重建 —— 数值结果

根据上面的理论知识和算法, 我们首先建立了一个 500×1 的稀疏信号 f , 然后将一个 500×500 的随机算子 A 作用到 f 上得到噪声信号 f_0 , 我们得到的原始信号 f 和噪声信号 f_0 如图 3 所示。

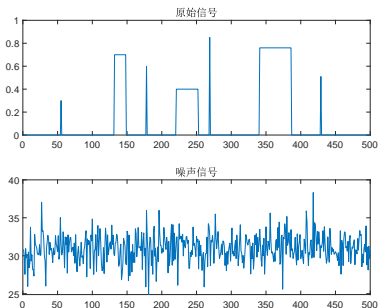


图 3: 原始稀疏信号和噪声信号

稀疏信号重建 —— 数值结果

原始信号和重建信号的对比如图 4 所示

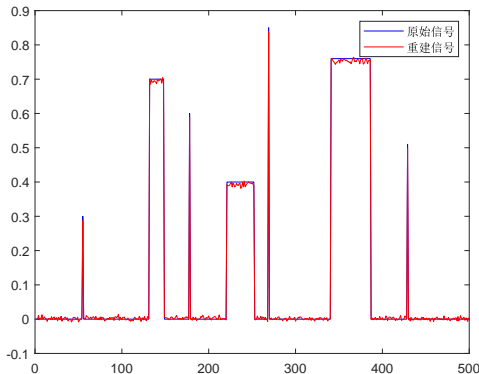


图 4: 分裂 Bregman 用于稀疏信号重建

表 1: 系数信号重建

PSNR		
重建误差	原始信号	重建信号
0.0062	18.3097	18.30971

有关稀疏信号重建方面进一步的文献可以参考 Osher 等人的工作 [3].

稀疏信号重建 —— 数值结果

原始信号和重建信号的对比如图 4 所示

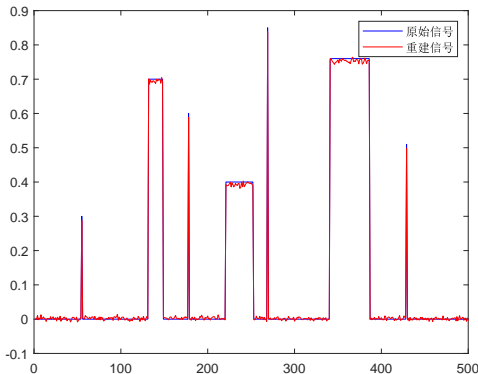


图 4: 分裂 Bregman 用于稀疏信号重建

表 1: 系数信号重建

PSNR		
重建误差	原始信号	重建信号
0.0062	18.3097	18.30971

有关稀疏信号重建方面进一步的文献可以参考 Osher 等人的工作 [3].

TV 去噪 —— 各向异性 TV 去噪

考虑各向异性 Rudin-Osher-Fatemi 去噪模型 [1], 我们想要从 f 中恢复原始图像 u , 他们提出的模型是

$$\min_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2} \|u - f\|_2^2$$

为了应用分裂 Bregman 算法, 我们首先分别用 d_x 和 d_y 替代 $\nabla_x u$ 和 $\nabla_y u$ 。有以下约束优化问题

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 \quad \text{such that} \quad d_x = \nabla_x u \quad \text{and} \quad d_y = \nabla_y u$$

为了弱化这个公式中的约束条件, 我们添加二次罚函数项:

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u\|_2^2$$

TV 去噪 —— 各向异性 TV 去噪

考虑各向异性 Rudin-Osher-Fatemi 去噪模型 [1], 我们想要从 f 中恢复原始图像 u , 他们提出的模型是

$$\min_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2} \|u - f\|_2^2$$

为了应用分裂 Bregman 算法, 我们首先分别用 d_x 和 d_y 替代 $\nabla_x u$ 和 $\nabla_y u$ 。有以下约束优化问题

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 \quad \text{such that} \quad d_x = \nabla_x u \quad \text{and} \quad d_y = \nabla_y u$$

为了弱化这个公式中的约束条件, 我们添加二次罚函数项:

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u\|_2^2$$

TV 去噪 —— 各向异性 TV 去噪

考虑各向异性 Rudin-Osher-Fatemi 去噪模型 [1], 我们想要从 f 中恢复原始图像 u , 他们提出的模型是

$$\min_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2} \|u - f\|_2^2$$

为了应用分裂 Bregman 算法, 我们首先分别用 d_x 和 d_y 替代 $\nabla_x u$ 和 $\nabla_y u$ 。有以下约束优化问题

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 \quad \text{such that} \quad d_x = \nabla_x u \quad \text{and} \quad d_y = \nabla_y u$$

为了弱化这个公式中的约束条件, 我们添加二次罚函数项:

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u\|_2^2$$

TV 去噪 —— 各向异性 TV 去噪

最后, 我们通过应用 Bregman 迭代来严格执行约束:

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y^k\|_2^2$$

其中 b_x^k 和 b_y^k 是通过 Bregman 迭代选择的适当值, 所得到的方案如下^[9]:

$$u^{k+1} = \operatorname{argmin}_u \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2$$

$$d_x^{k+1} = \operatorname{argmin}_{d_x} |d_x| + \frac{\lambda}{2} \|d_x - \nabla_x u^{k+1} - b_x^k\|_2^2$$

$$d_y^{k+1} = \operatorname{argmin}_{d_y} |d_y| + \frac{\lambda}{2} \|d_y - \nabla_y u^{k+1} - b_y^k\|_2^2$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

TV 去噪 —— 各向异性 TV 去噪

最后, 我们通过应用 Bregman 迭代来严格执行约束:

$$\min_{u, d_x, d_y} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y^k\|_2^2$$

其中 b_x^k 和 b_y^k 是通过 Bregman 迭代选择的适当值, 所得到的方案如下^[9]:

$$u^{k+1} = \operatorname{argmin}_u \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2$$

$$d_x^{k+1} = \operatorname{argmin}_{d_x} |d_x| + \frac{\lambda}{2} \|d_x - \nabla_x u^{k+1} - b_x^k\|_2^2$$

$$d_y^{k+1} = \operatorname{argmin}_{d_y} |d_y| + \frac{\lambda}{2} \|d_y - \nabla_y u^{k+1} - b_y^k\|_2^2$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

TV 去噪 —— 各向异性 TV 去噪

算法 5 分裂 Bregman 方法用于各向异性 TV 去噪

Initialize: $u^0 = f, d_x^0 = d_y^0 = b_x^0 = b_y^0 = 0$

while $\|u^k - u^{k-1}\|_2 > tol$ **do**

$$u^{k+1} = G^k$$

$$d_x^{k+1} = \text{shrink}(\nabla_x u^{k+1} + b_x^k, 1/\lambda)$$

$$d_y^{k+1} = \text{shrink}(\nabla_y u^{k+1} + b_y^k, 1/\lambda)$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

end while

TV 去噪 —— 各向同性 TV 去噪

分裂 Bregman 技术也可以用于具有高斯噪声的各向同性 TV 模型,除了我们考虑各向同性的总体变化之外,这种情况与之前的情况类似,在这种情况下,我们希望解决

$$\min_u \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} + \frac{\mu}{2} \|u - f\|_2^2 \quad (6.1)$$

就像我们之前做的那样,我们通过设置 $d_x = \nabla_x u$ 和 $d_y = \nabla_y u$ 来分割 L1 和 L2 分量,这个问题的分裂 Bregman 公式就变成了

$$\min_{u, d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\mu}{2} \|u - f\|_2^2 \quad \text{such that } d_x = \nabla_x u \quad \text{and} \quad d_y = \nabla_y u$$

其中

$$\|(d_x, d_y)\|_2 = \sum_{i,j} \sqrt{d_{x,i,j}^2 + d_{y,i,j}^2}$$

TV 去噪 —— 各向同性 TV 去噪

分裂 Bregman 技术也可以用于具有高斯噪声的各向同性 TV 模型,除了我们考虑各向同性的总体变化之外,这种情况与之前的情况类似,在这种情况下,我们希望解决

$$\min_u \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} + \frac{\mu}{2} \|u - f\|_2^2 \quad (6.1)$$

就像我们之前做的那样,我们通过设置 $d_x = \nabla_x u$ 和 $d_y = \nabla_y u$ 来分割 L1 和 L2 分量,这个问题的分裂 Bregman 公式就变成了

$$\min_{u, d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\mu}{2} \|u - f\|_2^2 \quad \text{such that } d_x = \nabla_x u \quad \text{and} \quad d_y = \nabla_y u$$

其中

$$\|(d_x, d_y)\|_2 = \sum_{i,j} \sqrt{d_{x,i,j}^2 + d_{y,i,j}^2}$$

TV 去噪 —— 各向同性 TV 去噪

请注意, d_x 和 d_y 与各向异性情况不同, 它不会解耦, 这改变了必须处理这些变量的方式。像我们在各向异性情况下一样强化约束, 添加惩罚函数项并应用 Bregman 迭代, 我们有

$$\min_{u, d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y\|_2^2$$

应用 Bregman 迭代方法, 得到各向同性情况下的分裂 Bregman 方案如下:

$$u^{k+1} = \operatorname{argmin}_u \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2$$

$$(d_x^{k+1}, d_y^{k+1}) = \operatorname{argmin}_{d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\lambda}{2} \|d_x - \nabla_x u^k - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u^k - b_y^k\|_2^2$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$



TV 去噪 —— 各向同性 TV 去噪

请注意, d_x 和 d_y 与各向异性情况不同, 它不会解耦, 这改变了必须处理这些变量的方式。像我们在各向异性情况下一样强化约束, 添加惩罚函数项并应用 Bregman 迭代, 我们有

$$\min_{u, d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y\|_2^2$$

应用 Bregman 迭代方法, 得到各向同性情况下的分裂 Bregman 方案如下:

$$u^{k+1} = \operatorname{argmin}_u \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2$$

$$(d_x^{k+1}, d_y^{k+1}) = \operatorname{argmin}_{d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\lambda}{2} \|d_x - \nabla_x u^k - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u^k - b_y^k\|_2^2$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$



TV 去噪 —— 各向同性 TV 去噪

在各向同性问题中, d_x 和 d_y 会耦合在一起。我们首先考虑关于 d_x 的最小化问题

$$0 = \frac{d_x^{k+1}}{\|(d_x^k, d_y^k)\|_2} + \lambda(d_x^{k+1} - \nabla_x u^k - b_x^k).$$

定义

$$s^k = \sqrt{|\nabla_x u^k + b_x^k|^2 + |\nabla_y u^k + b_y^k|^2} \quad (6.2)$$

我们通过 s^k 来近似 $\|(d_x^k, d_y^k)\|_2$, 得到

$$0 = \frac{d_x^{k+1}}{s^k} + \lambda(d_x^{k+1} - \nabla_x u^k - b_x^k).$$

变形得到

$$d_x^{k+1} \left(\lambda + \frac{1}{s^k} \right) = \lambda(\nabla_x u^k + b_x^k)$$

TV 去噪 —— 各向同性 TV 去噪

在各向同性问题中, d_x 和 d_y 会耦合在一起。我们首先考虑关于 d_x 的最小化问题

$$0 = \frac{d_x^{k+1}}{\|(d_x^k, d_y^k)\|_2} + \lambda(d_x^{k+1} - \nabla_x u^k - b_x^k).$$

定义

$$s^k = \sqrt{|\nabla_x u^k + b_x^k|^2 + |\nabla_y u^k + b_y^k|^2} \quad (6.2)$$

我们通过 s^k 来近似 $\|(d_x^k, d_y^k)\|_2$, 得到

$$0 = \frac{d_x^{k+1}}{s^k} + \lambda(d_x^{k+1} - \nabla_x u^k - b_x^k).$$

变形得到

$$d_x^{k+1} \left(\lambda + \frac{1}{s^k} \right) = \lambda(\nabla_x u^k + b_x^k)$$

TV 去噪 —— 各向同性 TV 去噪

化简得

$$d_x^{k+1} = \frac{s^k \lambda (\nabla_x u^k + b_x^k)}{s^k \lambda + 1}$$

同理可得

$$d_y^{k+1} = \frac{s^k \lambda (\nabla_y u^k + b_y^k)}{s^k \lambda + 1}$$

因此, 尽管 d_x 和 d_y 不像各向异性情况下那样解耦, 但我们仍然可以使用广义收缩公式明确地解决关于 d 的最小化问题。

$$d_x^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_x u^k + b_x^k}{s^k} \quad (6.3)$$

$$d_y^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_y u^k + b_y^k}{s^k} \quad (6.4)$$

TV 去噪 —— 各向同性 TV 去噪

化简得

$$d_x^{k+1} = \frac{s^k \lambda (\nabla_x u^k + b_x^k)}{s^k \lambda + 1}$$

同理可得

$$d_y^{k+1} = \frac{s^k \lambda (\nabla_y u^k + b_y^k)}{s^k \lambda + 1}$$

因此, 尽管 d_x 和 d_y 不像各向异性情况下那样解耦, 但我们仍然可以使用广义收缩公式明确地解决关于 d 的最小化问题。

$$d_x^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_x u^k + b_x^k}{s^k} \quad (6.3)$$

$$d_y^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_y u^k + b_y^k}{s^k} \quad (6.4)$$

TV 去噪 —— 各向同性 TV 去噪

化简得

$$d_x^{k+1} = \frac{s^k \lambda (\nabla_x u^k + b_x^k)}{s^k \lambda + 1}$$

同理可得

$$d_y^{k+1} = \frac{s^k \lambda (\nabla_y u^k + b_y^k)}{s^k \lambda + 1}$$

因此, 尽管 d_x 和 d_y 不像各向异性情况下那样解耦, 但我们仍然可以使用广义收缩公式明确地解决关于 d 的最小化问题。

$$d_x^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_x u^k + b_x^k}{s^k} \quad (6.3)$$

$$d_y^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_y u^k + b_y^k}{s^k} \quad (6.4)$$

TV 去噪 —— 各向同性 TV 去噪

算法 6 分裂 Bregman 方法用于各向同性 TV 去噪

Initialize: $u^0 = f, d_x^0 = d_y^0 = b_x^0 = b_y^0 = 0$

while $\|u^k - u^{k-1}\|_2 > tol$ **do**

$$u^{k+1} = G^k$$

$$d_x^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

end while

TV 去噪 —— 数值结果

对于我们的实验, 我们使用 $\mu = 0.05$, $\lambda = 0.1$, $c = 25$, 停止准则是 $\frac{\|u^k - u^{k-1}\|_2}{\|f\|_2} < 10^{-5}$ 。相对误差由 $\frac{\|u - X\|_2}{\|X\|_2}$ 来度量。

根据以上参数设置, 我们首先将其作用在 512×512 的经典图像 Lena 上, 分别用各向异性 TV 去噪算法和各向同性 TV 去噪算法进行图像恢复, 得到的结果如图 5 所示。



TV 去噪 —— 数值结果

对于我们的实验, 我们使用 $\mu = 0.05$, $\lambda = 0.1$, $c = 25$, 停止准则是 $\frac{\|u^k - u^{k-1}\|_2}{\|f\|_2} < 10^{-5}$ 。相对误差由 $\frac{\|u - X\|_2}{\|X\|_2}$ 来度量。

根据以上参数设置, 我们首先将其作用在 512×512 的经典图像 Lena 上, 分别用各向异性 TV 去噪算法和各向同性 TV 去噪算法进行图像恢复, 得到的结果如图 5 所示。

TV 去噪 —— 数值结果

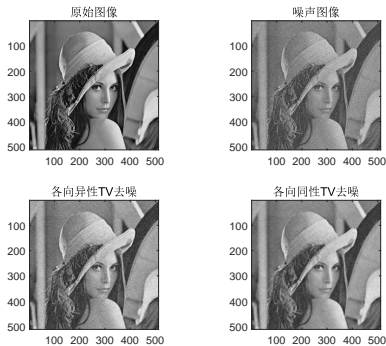


图 5: Lena 图像的分裂 Bregman 去噪结果

表 2: Lena 图像 TV 去噪

噪声图像	PSNR		各向异性 TV 去噪			各向同性 TV 去噪		
	各向异性 TV 去噪	各向同性 TV 去噪	迭代次数	时间 (s)	相对误差	迭代次数	时间 (s)	相对误差
20.190063	27.919886	29.189392	22	7.83	0.010686	19	6.73	0.010143

TV 去噪 —— 数值结果

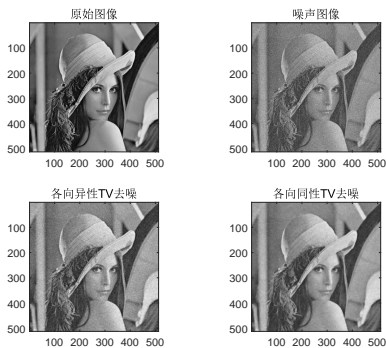


图 5: Lena 图像的分裂 Bregman 去噪结果

表 2: Lena 图像 TV 去噪

噪声图像	PSNR		迭代次数	各向异性 TV 去噪		各向同性 TV 去噪		相对误差
	各向异性 TV 去噪	各向同性 TV 去噪		时间 (s)	相对误差	时间 (s)	相对误差	
20.190063	27.919886	29.189392	22	7.83	0.010686	19	6.73	0.010143

TV 去噪 —— 数值结果

为了说明分裂 Bregman 算法误差的收敛情况, 图 6 展示了 Lena 图像在两种去噪方法下的相对误差的对数值随着迭代次数的变化情况。

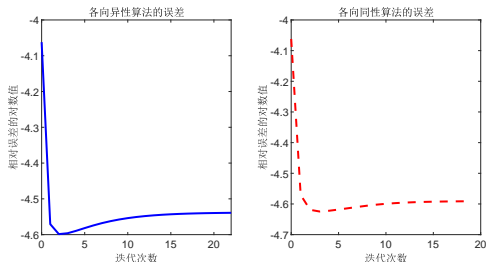


图 6: Lena 图像的分裂 Bregman 去噪结果

从图 6 中可以看出相对误差随迭代次数的增加并不是单调递减的, 那么我们有理由猜测这个结论是否正确。

TV 去噪 —— 数值结果

为了说明分裂 Bregman 算法误差的收敛情况, 图 6 展示了 Lena 图像在两种去噪方法下的相对误差的对数值随着迭代次数的变化情况。

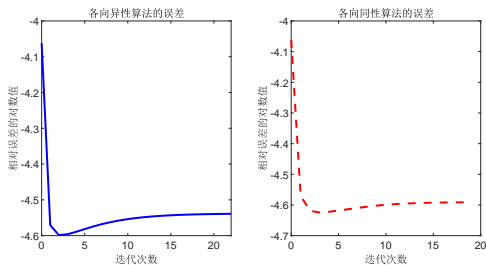


图 6: Lena 图像的分裂 Bregman 去噪结果

从图 6 中可以看出相对误差随迭代次数的增加并不是单调递减的, 那么我们有理由猜测这个结论是否正确。

TV 去噪 —— 数值结果

从表 3 中我们可以看到, 与各向异性 TV 去噪相比, 各向同性 TV 去噪速度更快, 迭代次数更少, 结果更精确。

表 3: 分裂 Bregman 算法 TV 去噪结果

图像	规模		各向异性 TV 去噪			各向同性 TV 去噪		
	m	n	迭代次数	相对误差	时间(s)	迭代次数	相对误差	时间(s)
Fluid Jet	400	300	29	0.082906	3.117048	24	0.081180	2.358589
Bone	367	490	28	0.067295	4.763319	23	0.064862	3.928985
Gatlinburg	480	640	26	0.045119	8.339040	23	0.043952	7.204275
Durer	648	509	25	0.024920	8.886522	21	0.025723	7.353789
Durer Detail	359	371	30	0.078010	3.705711	25	0.083189	2.890617
Cape Cod	360	360	29	0.086020	3.335583	24	0.082384	2.703716
Clown	200	320	31	0.087864	1.933677	26	0.087427	1.400082
Earth	257	250	27	0.073270	1.605807	23	0.072721	1.394439
Mandrill	480	500	26	0.031280	6.447517	23	0.031230	5.681188

TV 去噪 —— 数值结果

此外,我们还设置噪声方差 c 分别 15, 25, 35, 以此根据不同噪声强度下进行 TV 去噪,并计算相应的峰值信号比 (PSNR), 计算结果如表 4 所示。

表 4: 不同噪声强度的峰值信号比

图像	$c = 15, \text{PSNR}$			$c = 25, \text{PSNR}$			$c = 35, \text{PSNR}$		
	Noisy	ATV	ITV	Noisy	ATV	ITV	Noisy	ATV	ITV
Fluid Jet	24.6	33.9	37.2	20.2	29.4	32.4	17.3	26.6	29.1
Bone	24.6	34.0	37.7	20.2	29.5	32.7	17.3	26.6	29.1
Gatlinburg	24.6	33.6	36.3	20.2	29.5	32.3	17.3	26.7	29.2
Durer	24.6	28.3	28.1	20.2	26.6	27.2	17.2	24.9	26.0
Durer Detail	24.6	27.3	27.1	20.2	25.9	26.3	17.2	24.4	25.3
Cape Cod	24.7	33.5	36.4	20.2	29.3	32.1	17.2	26.4	28.8
Clown	24.6	30.2	31.1	20.2	27.7	29.2	17.3	26.6	29.1
Earth	24.6	26.5	26.6	20.2	25.2	25.8	17.2	23.9	24.8
Mandrill	24.6	20.0	19.9	20.2	19.7	19.7	17.2	19.3	19.4

TV 去噪 —— 数值结果

图 7 表明 Fluid Jet 图像去噪结果的相对误差单调下降。

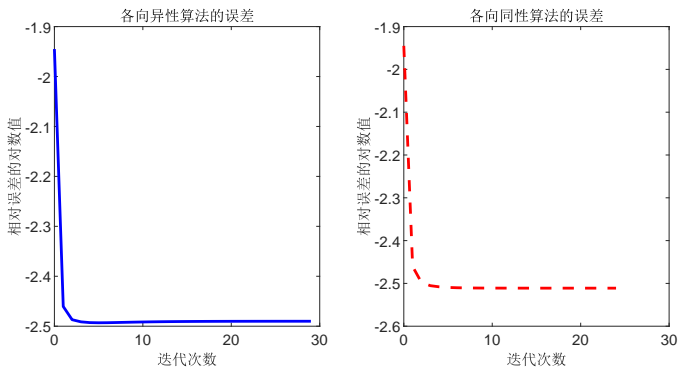


图 7: Fluid Jet 图像的分裂 Bregman 去噪误差结果

TV 去噪 —— 数值结果

图 8 表明 Durer Detail 图像去噪结果在第一次迭代中误差减小, 之后误差单调增加并收敛到比原始噪声图像的相对误差更高的点。

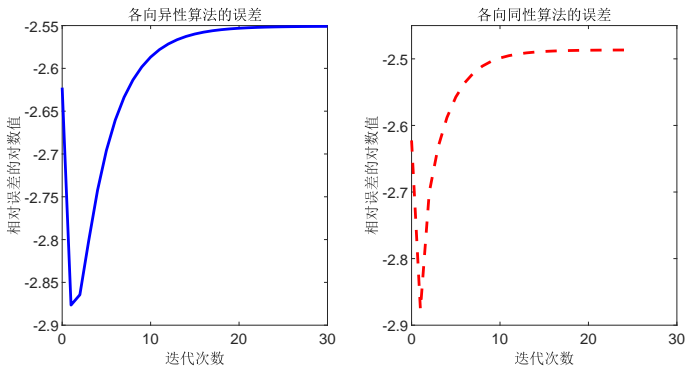


图 8: Durer Detail 图像的分裂 Bregman 去噪误差结果

TV 去噪 —— 数值结果

图 9 表明 Clown 图像的去噪结果在迭代过程中的相对误差的值是震荡的，没有一个明显的变化规律。

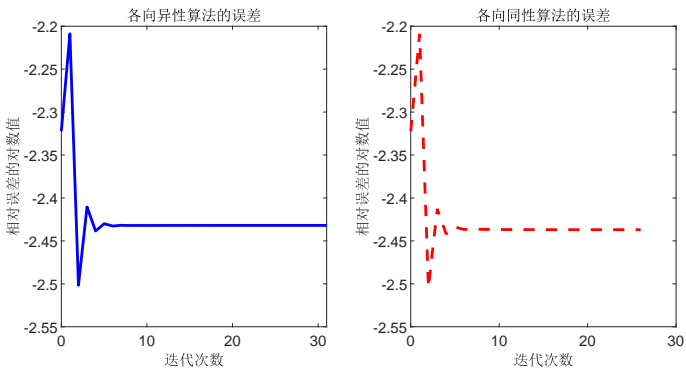


图 9: Clown 图像的分裂 Bregman 去噪误差结果

TV 去噪 —— 数值结果

图 10 表明 Mandrill 图像去噪结果的相对误差会随着迭代次数一直增加下去。

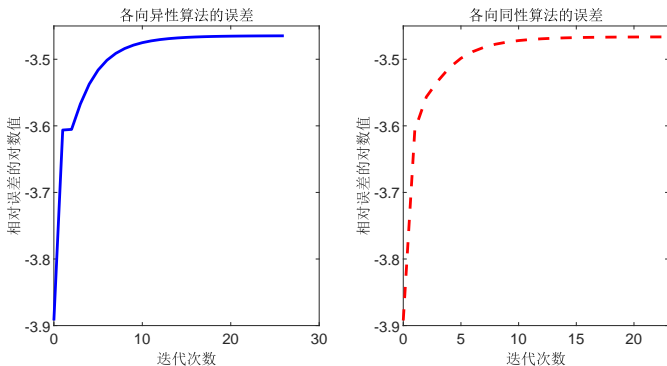


图 10: Mandrill 图像的分裂 Bregman 去噪误差结果



TV 去噪 —— 数值结果

我们在前面展示了 Bregman 迭代算法的相对误差是单调下降的。然而，最后三幅图 (图 8、图 9 和图 10) 表明，具有各向异性和各向同性的分裂 Bregman 算法的相对误差不一定是单调收敛的。

目前我们对这种行为还没有一个令人满意的解释，并且进一步探索它会很有趣。我们未来还想看看 Bregman 迭代算法中讨论的其他性质是否适用于分裂 Bregman 算法。

TV 去噪 —— 数值结果

我们在前面展示了 Bregman 迭代算法的相对误差是单调下降的。然而，最后三幅图 (图 8、图 9 和图 10) 表明，具有各向异性和各向同性的分裂 Bregman 算法的相对误差不一定是单调收敛的。

目前我们对这种行为还没有一个令人满意的解释，并且进一步探索它会很有趣。我们未来还想看看 Bregman 迭代算法中讨论的其他性质是否适用于分裂 Bregman 算法。



非盲 TV 去卷积 —— 各向异性非盲 TV 去卷积

模糊图像问题可表示为 $g = K * u - f$, 我们希望从模糊图像 g 中恢复原始图像 u 。除了出现卷积核之外, 该模型与前面的去噪问题相同, 我们需要解决问题

$$\operatorname{argmin}_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2} \|K * u - f\|_2^2$$

根据以前相同的处理办法, 令 $d_x = \nabla_x u$, $d_y = \nabla_y u$, $H(u) = \frac{\mu}{2} \|Au - f\|_2^2$, 然后相应的模型如下

$$\left\{ \begin{array}{l} u^{k+1} = \operatorname{argmin}_u H(u) + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2 \\ d_x^{k+1} = \operatorname{argmin}_{d_x} |d_x| + \frac{\lambda}{2} \|d_x - \nabla_x u^{k+1} - b_x^k\|_2^2 \\ d_y^{k+1} = \operatorname{argmin}_{d_y} |d_y| + \frac{\lambda}{2} \|d_y - \nabla_y u^{k+1} - b_y^k\|_2^2 \\ b_x^{k+1} = b_x^k + \nabla_x u^{k+1} - d_x^{k+1} \\ b_y^{k+1} = b_y^k + \nabla_y u^{k+1} - d_y^{k+1} \end{array} \right.$$



非盲 TV 去卷积 —— 各向异性非盲 TV 去卷积

模糊图像问题可表示为 $g = K * u - f$, 我们希望从模糊图像 g 中恢复原始图像 u 。除了出现卷积核之外, 该模型与前面的去噪问题相同, 我们需要解决问题

$$\operatorname{argmin}_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2} \|K * u - f\|_2^2$$

根据以前相同的处理办法, 令 $d_x = \nabla_x u$, $d_y = \nabla_y u$, $H(u) = \frac{\mu}{2} \|Au - f\|_2^2$, 然后相应的模型如下

$$\left\{ \begin{array}{l} u^{k+1} = \operatorname{argmin}_u H(u) + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2 \\ d_x^{k+1} = \operatorname{argmin}_{d_x} |d_x| + \frac{\lambda}{2} \|d_x - \nabla_x u^{k+1} - b_x^k\|_2^2 \\ d_y^{k+1} = \operatorname{argmin}_{d_y} |d_y| + \frac{\lambda}{2} \|d_y - \nabla_y u^{k+1} - b_y^k\|_2^2 \\ b_x^{k+1} = b_x^k + \nabla_x u^{k+1} - d_x^{k+1} \\ b_y^{k+1} = b_y^k + \nabla_y u^{k+1} - d_y^{k+1} \end{array} \right.$$



非盲 TV 去卷积 —— 各向异性非盲 TV 去卷积

整理得到对应的各向异性非盲 TV 去卷积算法 7

算法 7 分裂 Bregman 方法用于各向异性非盲 TV 去卷积

Initialize: $u^0 = f, d_x^0 = d_y^0 = b_x^0 = b_y^0 = 0$

while $\|u^k - u^{k-1}\|_2 > tol$ **do**

$$u = \mathcal{F}^{-1} \left[\left(\frac{\mu}{\lambda} |\mathcal{F}(A)|^2 - \overline{\mathcal{F}(\Delta)} \right)^{-1} \frac{\mu}{\lambda} \overline{\mathcal{F}(A)} \mathcal{F}(F) - \mathcal{F}(\nabla(d^k - b^k)) \right]$$

$$d_x^{k+1} = \text{shrink}(\nabla_x u^{k+1} + b_x^k, 1/\lambda)$$

$$d_y^{k+1} = \text{shrink}(\nabla_y u^{k+1} + b_y^k, 1/\lambda)$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

end while

非盲 TV 去卷积 —— 各向同性非盲 TV 去卷积

各项同性非盲 TV 去卷积需要解决的问题如下：

$$\operatorname{argmin}_u \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} + \frac{\mu}{2} \|K * u - f\|_2^2$$

需要注意的是，各向同性非盲 TV 去卷积与各向异性非盲 TV 去卷积之间的区别依然是 d_x 和 d_y 的计算。根据前文各向同性 TV 去噪的推导过程，我们可以得到各向同性非盲 TV 去卷积的 d_x 和 d_y 的计算公式如下

$$d_x^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_y u^k + b_y^k}{s^k}$$

其中

$$s^k = \sqrt{|\nabla_x u^k + b_x^k|^2 + |\nabla_y u^k + b_y^k|^2}$$

非盲 TV 去卷积 —— 各向同性非盲 TV 去卷积

各项同性非盲 TV 去卷积需要解决的问题如下：

$$\operatorname{argmin}_u \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} + \frac{\mu}{2} \|K * u - f\|_2^2$$

需要注意的是，各向同性非盲 TV 去卷积与各向异性非盲 TV 去卷积之间的区别依然是 d_x 和 d_y 的计算。根据前文各向同性 TV 去噪的推导过程，我们可以得到各向同性非盲 TV 去卷积的 d_x 和 d_y 的计算公式如下

$$d_x^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_y u^k + b_y^k}{s^k}$$

其中

$$s^k = \sqrt{|\nabla_x u^k + b_x^k|^2 + |\nabla_y u^k + b_y^k|^2}$$

非盲 TV 去卷积 —— 各向同性非盲 TV 去卷积

那么对应的各向同性非盲 TV 去卷积算法如算法 8 所示。

算法 8 分裂 Bregman 方法用于各向同性非盲 TV 去卷积

Initialize: $u^0 = f, d_x^0 = d_y^0 = b_x^0 = b_y^0 = 0$

while $\|u^k - u^{k-1}\|_2 > tol$ **do**

$$u = \mathcal{F}^{-1} \left[\left(\frac{\mu}{\lambda} |\mathcal{F}(A)|^2 - \overline{\mathcal{F}(\Delta)} \right)^{-1} \frac{\mu}{\lambda} \overline{\mathcal{F}(A)} \mathcal{F}(F) - \mathcal{F}(\operatorname{div}(d^k - b^k)) \right]$$

$$d_x^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

end while

非盲 TV 去卷积 —— 数值结果

我们还是以 Lena 作为测试图像,设置的模糊核是一个 $9 \times 9, \sigma = 1.5$ 的高斯核,停止准则是 $\frac{\|u^k - u^{k-1}\|_2}{\|f\|_2} < 10^{-5}$, 设置参数 $\mu = 100000, \lambda = 50, Niter = 20$, 执行程序可以得到图 11。

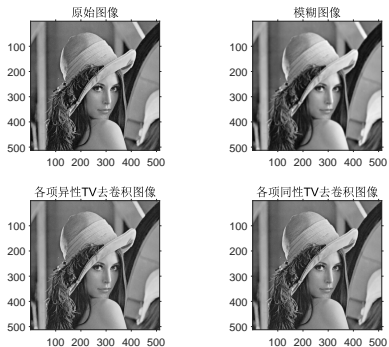


表 5: 非盲 TV 去卷积

PSNR		
模糊图像	各项异性	各向同性
30.206851	36.310557	36.451356

图 11: Lena 图像非盲 TV 去卷积

非盲 TV 去卷积 —— 数值结果

我们还是以 Lena 作为测试图像,设置的模糊核是一个 $9 \times 9, \sigma = 1.5$ 的高斯核,停止准则是 $\frac{\|u^k - u^{k-1}\|_2}{\|f\|_2} < 10^{-5}$, 设置参数 $\mu = 100000, \lambda = 50, Niter = 20$, 执行程序可以得到图 11。

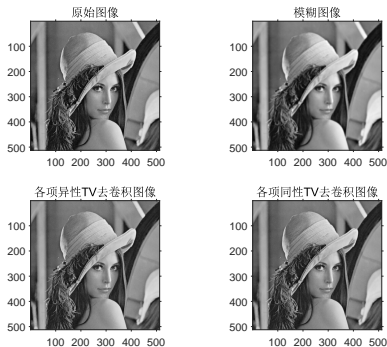


表 5: 非盲 TV 去卷积

PSNR		
模糊图像	各项异性	各向同性
30.206851	36.310557	36.451356

图 11: Lena 图像非盲 TV 去卷积

CS 图像重建 —— 基本原理

稀疏 MRI 重建问题的一般形式如下

$$\min_u J(u) \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2 \quad (6.5)$$

在 [6] 中表明, 使用 Bregman 迭代技术, 问题(6.5)可以被简化为一系列无约束的形式问题

$$u^{k+1} = \min_u J(u) + \frac{\mu}{2} \|R\mathcal{F}u - f^k\|_2^2 \quad (6.6)$$

$$f^{k+1} = f^k + f - R\mathcal{F}u^{k+1} \quad (6.7)$$

我们用分裂 Bregman 技术来解决这个无约束的问题。

CS 图像重建 —— 基本原理

稀疏 MRI 重建问题的一般形式如下

$$\min_u J(u) \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2 \quad (6.5)$$

在 [6] 中表明, 使用 Bregman 迭代技术, 问题(6.5)可以被简化为一系列无约束的形式问题

$$u^{k+1} = \min_u J(u) + \frac{\mu}{2} \|R\mathcal{F}u - f^k\|_2^2 \quad (6.6)$$

$$f^{k+1} = f^k + f - R\mathcal{F}u^{k+1} \quad (6.7)$$

我们用分裂 Bregman 技术来解决这个无约束的问题。



CS 图像重建 —— 基本原理

我们得到无约束的 CS 优化算法：

算法 9 无约束的 CS 优化算法

initialize: $u^0 = \mathcal{F}^{-1}f$, and $d_x^0 = d_y^0 = w^0 = b_x^0 = b_y^0 = b_w^0 = 0$

while $\|u^k - u^{k-1}\|_2 > tol$ do

$$u^{k+1} = \mathcal{F}^{-1}K^{-1}\mathcal{F}rhs^k$$

$$d_x^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$w^{k+1} = \text{shrink}(Wu^{k+1} + b_w^k, 1/\lambda)$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

$$b_w^{k+1} = b_w^k + (Wu^{k+1} - w^{k+1})$$

end while

请注意, 该算法仅解决无约束 CS 问题(6.6)。为了解决约束问题(6.5), 我们必须上面的算法中用 f^k 代替 f 。在近似求解每个无约束问题之后, 我们必须应用 Bregman 更新规则 $f^{k+1} = f^k + f - RFu^{k+1}$ 。

CS 图像重建 —— 基本原理

我们得到无约束的 CS 优化算法：

算法 10 无约束的 CS 优化算法

initialize: $u^0 = \mathcal{F}^{-1}f$, and $d_x^0 = d_y^0 = w^0 = b_x^0 = b_y^0 = b_w^0 = 0$

while $\|u^k - u^{k-1}\|_2 > tol$ **do**

$$u^{k+1} = \mathcal{F}^{-1}K^{-1}\mathcal{F}rhs^k$$

$$d_x^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$w^{k+1} = \text{shrink}(Wu^{k+1} + b_w^k, 1/\lambda)$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

$$b_w^{k+1} = b_w^k + (Wu^{k+1} - w^{k+1})$$

end while

请注意, 该算法仅解决无约束 CS 问题(6.6)。为了解决约束问题(6.5), 我们必须上面的算法中用 f^k 代替 f 。在近似求解每个无约束问题之后, 我们必须应用 Bregman 更新规则 $f^{k+1} = f^k + f - R\mathcal{F}u^{k+1}$ 。

CS 图像重建 —— 基本原理

当我们在这个无约束算法内部的外面嵌入 Bregman 更新时, 我们得到以下

算法 11:

算法 11 约束的 CS 优化算法

initialize: $u^0 = \mathcal{F}^{-1}f$, and $d_x^0 = d_y^0 = w^0 = b_x^0 = b_y^0 = b_w^0 = 0$

while $\|R\mathcal{F}u^k - f\|_2^2 > \sigma^2$ **do**

for $i = 1$ to N **do**

$$u^{k+1} = \mathcal{F}^{-1}K^{-1}\mathcal{F}r_h s^k$$

$$d_x^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$w^{k+1} = \text{shrink}(Wu^{k+1} + b_w^k, 1/\lambda)$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

$$b_w^{k+1} = b_w^k + (Wu^{k+1} - w^{k+1})$$

end for

$$f^{k+1} = f^k + f - R\mathcal{F}u^{k+1}$$

end while

CS 图像重建 —— 合成图像 CS 重建

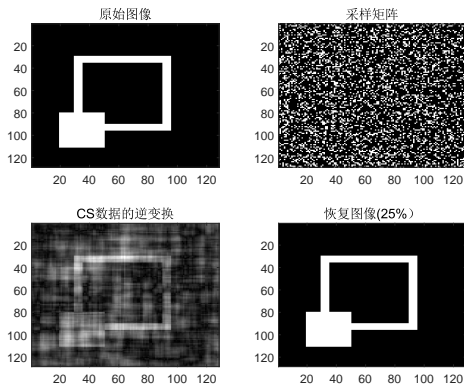
图 12 展示了从 25% 的 k 空间数据恢复合成图像

表 6: 合成图像重建结果

重建误差	PSNR	
	噪声图像	重建图像
6.6972×10^{-9}	8.715005	81.74106

图 12: 从 25% 的 k 空间数据恢复合成图像

CS 图像重建 —— 合成图像 CS 重建

图 12 展示了从 25% 的 k 空间数据恢复合成图像

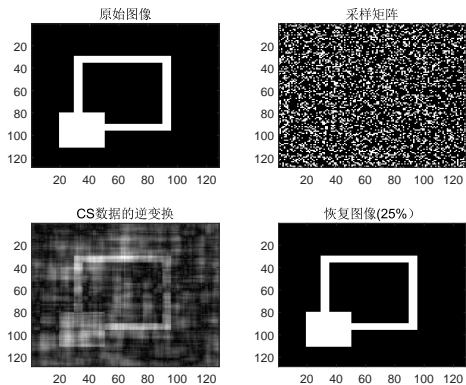


表 6: 合成图像重建结果

PSNR		
重建误差	噪声图像	重建图像
6.6972×10^{-9}	8.715005	81.74106

图 12: 从 25% 的 k 空间数据恢复合成图像

CS 图像重建 —— 合成图像 CS 重建

图 13 展示了不同 k 空间数据恢复合成图像的情况。

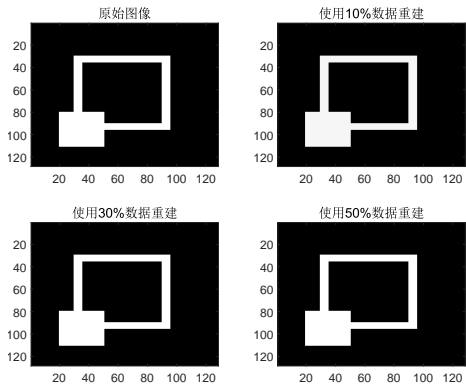


表 7: 合成图像重建对比

k 数据	重建误差	PSNR(噪声:8.715005)
10%	9.8333×10^{-6}	50.073021
30%	2.4951×10^{-10}	96.029067
50%	1.3391×10^{-13}	128.731772

图 13: 不同 k 空间数据恢复合成图像比较

CS 图像重建 —— 合成图像 CS 重建

图 13 展示了不同 k 空间数据恢复合成图像的情况。

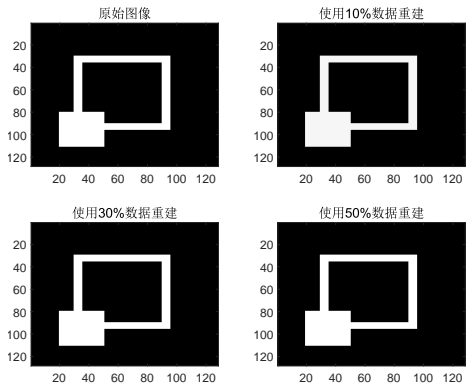


表 7: 合成图像重建对比

k 数据	重建误差	PSNR(噪声:8.715005)
10%	9.8333×10^{-6}	50.073021
30%	2.4951×10^{-10}	96.029067
50%	1.3391×10^{-13}	128.731772

图 13: 不同 k 空间数据恢复合成图像比较

CS 图像重建 —— 真实图像 CS 重建

图 14 展示了从 25% 的 k 空间数据恢复 Lena 图像

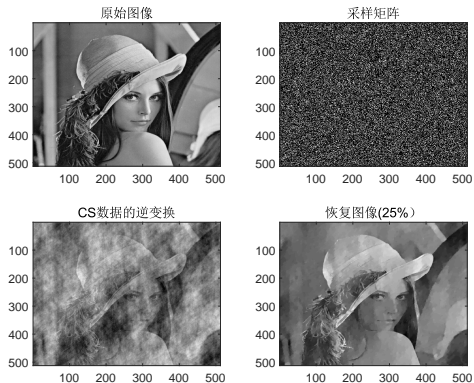


表 8: Lena 图像重建结果

PSNR		
重建误差	噪声图像	重建图像
0.0123	5.671858	19.109545

图 14: 从 25% 的 k 空间数据恢复 Lena

CS 图像重建 —— 真实图像 CS 重建

图 14 展示了从 25% 的 k 空间数据恢复 Lena 图像

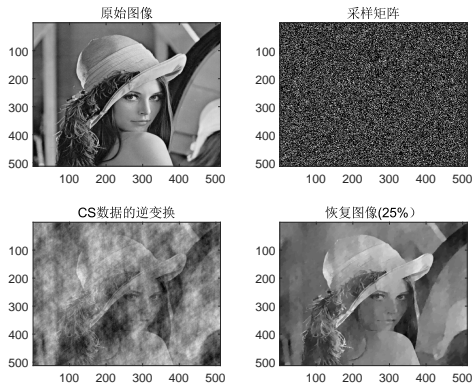


表 8: Lena 图像重建结果

PSNR		
重建误差	噪声图像	重建图像
0.0123	5.671858	19.109545

图 14: 从 25% 的 k 空间数据恢复 Lena

CS 图像重建 —— 真实图像 CS 重建

分别在 10%、30%、50% 的 k 空间数据情况下的重建图像如图 15 所示。

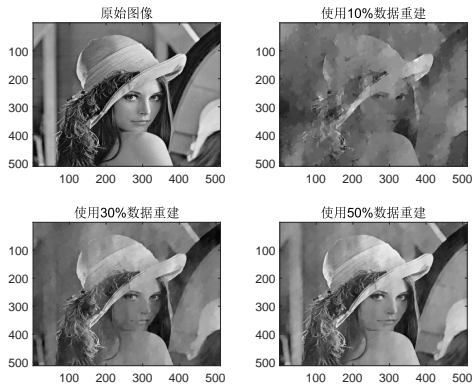


表 9: Lena 图像重建对比

k 数据	重建误差	PSNR(噪声:5.671858)
10%	0.0218	16.619690
30%	0.0105	19.789812
50%	0.0139	24.084200

图 15: 不同 k 空间数据恢复 Lena 图像比较

CS 图像重建 —— 真实图像 CS 重建

分别在 10%、30%、50% 的 k 空间数据情况下的重建图像如图 15 所示。

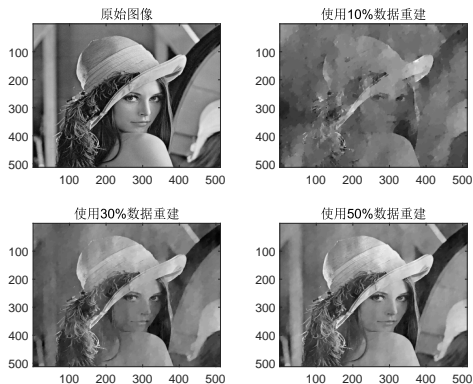


表 9: Lena 图像重建对比

k 数据	重建误差	PSNR(噪声:5.671858)
10%	0.0218	16.619690
30%	0.0105	19.789812
50%	0.0139	24.084200

图 15: 不同 k 空间数据恢复 Lena 图像比较



MRI 图像重建 —— 基本原理

压缩感知制定了以下约束优化问题：

$$\min_u J(u) \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2, \quad (6.8)$$

基于图像获取中帧之间的固有相似性，我们扩展(6.8)中的公式以最小化空间和时间上的 TV。使用各向同性时空 TV 模型，我们获得以下约束优化问题^[17]：

$$\min_u \|(\nabla_x u, \nabla_y u)\|_2 + \|\nabla_t u\|_1 \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2 \quad (6.9)$$

其中 ∇_t 是时间导数， $\|\nabla_t u\|_1$ 是时间 TV。

根据前面的方法，我们让 $d_x \leftarrow \nabla_x u$ 、 $d_y \leftarrow \nabla_y u$ 和 $d_t \leftarrow \nabla_t u$ ，通过应用 Bregman 迭代，将问题(6.8)转换为无约束优化问题

$$\begin{aligned} \min_u & \| (d_x, d_y) \|_2 + \| d_t \|_1 + \frac{\mu}{2} \| R\mathcal{F}u - f^k \|_2^2 + \frac{\lambda}{2} \| d_x^k - \nabla_x u - b_x^k \|_2^2 \\ & + \frac{\lambda}{2} \| d_y^k - \nabla_y u - b_y^k \|_2^2 + \frac{\lambda}{2} \| d_t^k - \nabla_t u - b_t^k \|_2^2 \end{aligned} \quad (6.10)$$

MRI 图像重建 —— 基本原理

压缩感知制定了以下约束优化问题：

$$\min_u J(u) \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2, \quad (6.8)$$

基于图像获取中帧之间的固有相似性，我们扩展(6.8)中的公式以最小化空间和时间上的 TV。使用各向同性时空 TV 模型，我们获得以下约束优化问题^[17]：

$$\min_u \|(\nabla_x u, \nabla_y u)\|_2 + \|\nabla_t u\|_1 \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2 \quad (6.9)$$

其中 ∇_t 是时间导数， $\|\nabla_t u\|_1$ 是时间 TV。

根据前面的方法，我们让 $d_x \leftarrow \nabla_x u$ 、 $d_y \leftarrow \nabla_y u$ 和 $d_t \leftarrow \nabla_t u$ ，通过应用 Bregman 迭代，将问题(6.8)转换为无约束优化问题

$$\begin{aligned} \min_u \| (d_x, d_y) \|_2 + \| d_t \|_1 + \frac{\mu}{2} \| R\mathcal{F}u - f^k \|_2^2 + \frac{\lambda}{2} \| d_x^k - \nabla_x u - b_x^k \|_2^2 \\ + \frac{\lambda}{2} \| d_y^k - \nabla_y u - b_y^k \|_2^2 + \frac{\lambda}{2} \| d_t^k - \nabla_t u - b_t^k \|_2^2 \end{aligned} \quad (6.10)$$

MRI 图像重建 —— 基本原理

压缩感知制定了以下约束优化问题：

$$\min_u J(u) \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2, \quad (6.8)$$

基于图像获取中帧之间的固有相似性，我们扩展(6.8)中的公式以最小化空间和时间上的 TV。使用各向同性时空 TV 模型，我们获得以下约束优化问题^[17]：

$$\min_u \|(\nabla_x u, \nabla_y u)\|_2 + \|\nabla_t u\|_1 \quad \text{such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2 \quad (6.9)$$

其中 ∇_t 是时间导数， $\|\nabla_t u\|_1$ 是时间 TV。

根据前面的方法，我们让 $d_x \leftarrow \nabla_x u$ 、 $d_y \leftarrow \nabla_y u$ 和 $d_t \leftarrow \nabla_t u$ ，通过应用 Bregman 迭代，将问题(6.8)转换为无约束优化问题

$$\begin{aligned} \min_u \| (d_x, d_y) \|_2 + \| d_t \|_1 + \frac{\mu}{2} \| R\mathcal{F}u - f^k \|_2^2 + \frac{\lambda}{2} \| d_x^k - \nabla_x u - b_x^k \|_2^2 \\ + \frac{\lambda}{2} \| d_y^k - \nabla_y u - b_y^k \|_2^2 + \frac{\lambda}{2} \| d_t^k - \nabla_t u - b_t^k \|_2^2 \end{aligned} \quad (6.10)$$

MRI 图像重建 —— 基本原理

解决问题(6.9)中约束优化问题的完整算法如算法 12 所示。

算法 12 约束的时空 TV 重建算法

initialize: $u^0 = \mathcal{F}^{-1}f$, and $d_x^0 = d_y^0 = b_x^0 = b_y^0 = b_t^0 = 0$

while $\|R\mathcal{F}u^k - f\|_2^2 > \sigma^2$ **do**

for $i = 1$ to N **do**

$u^{k+1} = \mathcal{F}^{-1}K^{-1}\mathcal{F}r^k$

$$d_x^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - 1/\lambda, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$d_t^{k+1} = \text{shrink}(\nabla_t u^{k+1} + b_t^k, 1/\lambda)$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

$$b_t^{k+1} = b_t^k + (\nabla_t u^{k+1} - d_t^{k+1})$$

end for

$f^{k+1} = f^k + f - R\mathcal{F}u^{k+1}$

end while

MRI 图像重建 —— 数值结果

我们得到了大脑图像的两重重建结果如图 16 所示。

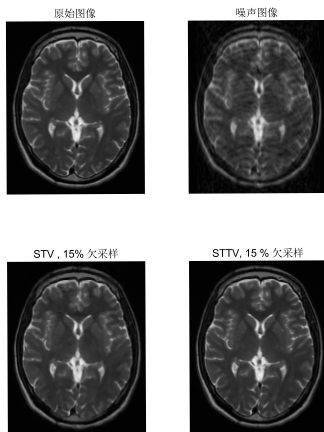


表 10: STV 和 STTV 数值结果

	噪声图像	STV	STTV
误差	—	0.1275	0.0573
PSNR	26.0627	31.26238	38.23819

图 16: 两种方法重建结果

MRI 图像重建 —— 数值结果

我们得到了大脑图像的两重重建结果如图 16 所示。

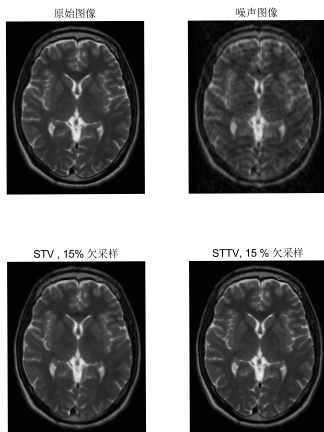


图 16: 两种方法重建结果

表 10: STV 和 STTV 数值结果

	噪声图像	STV	STTV
误差	—	0.1275	0.0573
PSNR	26.0627	31.26238	38.23819

MRI 图像重建 —— 数值结果

为了比较两种不同算法在迭代时间上的差异性, 我们做出了两种算法的相对误差随迭代次数之间的变化情况如图 17 所示。

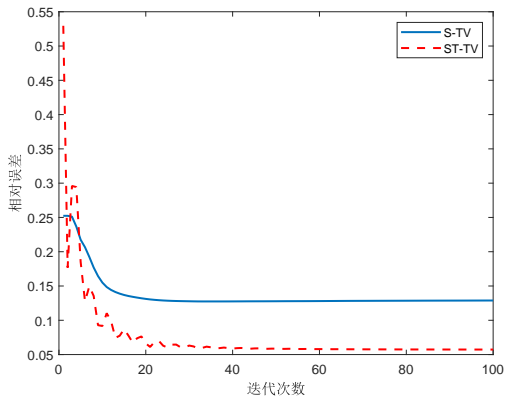


图 17: 重建误差与外循环迭代次数的关系

图像分割 —— 基本原理

分割和对象提取是图像处理和计算机视觉中最重要的任务之一。凸分割可以表现为如下形式^[13]

$$\min_{0 \leq u \leq 1} |\nabla u|_g + \mu \langle u, r \rangle$$

我们应用分裂 Bregman 方法, 像之前的问题一样, 引入辅助变量 $d \leftarrow \nabla u$ (这里 $d = (d_x, d_y)$, $\nabla u = (\nabla_x u, \nabla_y u)$)。为了执行弱等式约束, 我们添加一个二次罚函数。我们得到以下无约束优化问题^[13]

$$(u^*, d^*) = \operatorname{argmin}_{0 \leq u \leq 1, d} |d|_g + \frac{\mu}{2} \langle u, r \rangle + \frac{\lambda}{2} \|d - \nabla u\|_2^2 \quad (6.11)$$



图像分割 —— 基本原理

分割和对象提取是图像处理和计算机视觉中最重要的任务之一。凸分割可以表现为如下形式^[13]

$$\min_{0 \leq u \leq 1} |\nabla u|_g + \mu \langle u, r \rangle$$

我们应用分裂 Bregman 方法, 像之前的问题一样, 引入辅助变量 $d \leftarrow \nabla u$ (这里 $d = (d_x, d_y)$, $\nabla u = (\nabla_x u, \nabla_y u)$)。为了执行弱等式约束, 我们添加一个二次罚函数。我们得到以下无约束优化问题^[13]

$$(u^*, d^*) = \operatorname{argmin}_{0 \leq u \leq 1, d} |d|_g + \frac{\mu}{2} \langle u, r \rangle + \frac{\lambda}{2} \|d - \nabla u\|_2^2 \quad (6.11)$$



图像分割 —— 基本原理

为了严格执行约束条件, 我们使用 Bregman 迭代来解决问题, 由此产生的优化问题序列是

$$(u^{k+1}, d^{k+1}) = \underset{0 \leq u \leq 1, d}{\operatorname{argmin}} |d|_g + \frac{\mu}{2} \langle u, r \rangle + \frac{\lambda}{2} \|d - \nabla u - b^k\|_2^2$$

$$b^{k+1} = b^k + \nabla u^k - d^k$$

分裂 Bregman 用于图像分割的算法如下

算法 13 分裂 Bregman 方法用于图像分割

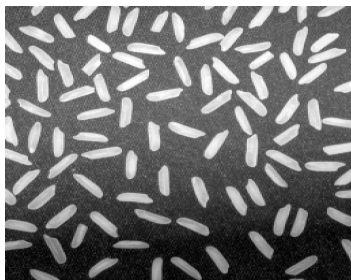
```

while  $\|u^k - u^{k-1}\|_2 > tol$  do
  定义  $r^k = (c_1^k - f)^2 - (c_2^k - f)^2$ 
   $u^{k+1} = \max\{\min\{\beta, 1\}, 0\}$ 
   $d^{k+1} = \operatorname{shrink}_g(b^k + \nabla u^{k+1}, \lambda)$ 
   $b^{k+1} = b^k + \nabla u^k - d^k$ 
  Find  $\Omega^k = \{x : u^k(x) > \mu\}$ 
  更新  $c_1^{k+1} = \int_{\Omega^k} f \, dx, c_2^{k+1} = \int_{(\Omega^k)^c} f \, dx$ 
end while

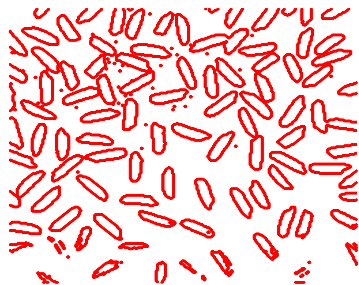
```

图像分割 —— 数值结果

首先将其运用在 256×256 图像上, 分割结果如图 18 所示。



(a) 原始图像



(b) 分割结果

图 18: 大米图像分割

图像分割 —— 数值结果

然后将其运用在经典的 512×512 Lena 图像上, 分割图像如图 19 所示。



(a) 原始图像



(b) 分割结果

图 19: Lena 图像分割

图像分割 —— 数值结果

我们比较了几种边缘检测算子对 Lena 图像的分割情况, 图 20 展示了五种不同的边缘检测算子对 Lena 图像的边缘检测情况。

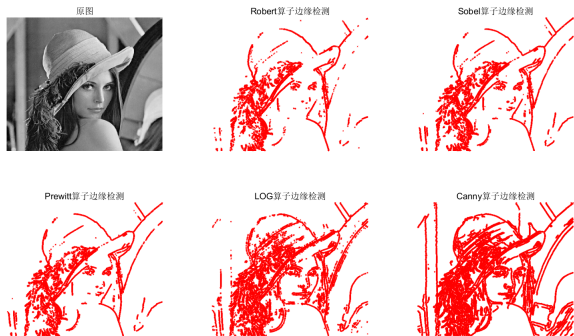


图 20: Lena 图像的边缘检测结果



主要研究内容

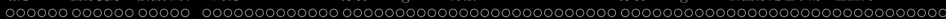
研究了分裂 Bregman 方法在图像处理中的一些应用, 主要研究内容如下:

- 研究整理了图像处理的相关数学原理和基础知识, 包括凸和下半连续、次梯度和次微分、Bregman 距离等重要概念, 并研究了岭回归、Lasso 和最佳子集回归基础上的正则化技术。
- 研究了分裂 Bregman 方法与增广拉格朗日乘子法及交替方向乘子法的内在关联, 及其在无约束优化和约束优化问题中的求解应用。
- 针对稀疏信号重建模型、全变差去噪模型、非盲 TV 去卷积模型、图像重建模型以及图像分割模型研究其分裂 Bregman 求解算法, 并通过数值实验验证其有效性。

主要研究内容

研究了分裂 Bregman 方法在图像处理中的一些应用,主要研究内容如下:

- 研究整理了图像处理的相关数学原理和基础知识,包括凸和下半连续、次梯度和次微分、Bregman 距离等重要概念,并研究了岭回归、Lasso 和最佳子集回归基础上的正则化技术。
- 研究了分裂 Bregman 方法与增广拉格朗日乘子法及交替方向乘子法的内在关联,及其在无约束优化和约束优化问题中的求解应用。
- 针对稀疏信号重建模型、全变差去噪模型、非盲 TV 去卷积模型、图像重建模型以及图像分割模型研究其分裂 Bregman 求解算法,并通过数值实验验证其有效性。



主要研究内容

研究了分裂 Bregman 方法在图像处理中的一些应用, 主要研究内容如下:

- 研究整理了图像处理的相关数学原理和基础知识, 包括凸和下半连续、次梯度和次微分、Bregman 距离等重要概念, 并研究了岭回归、Lasso 和最佳子集回归基础上的正则化技术。
- 研究了分裂 Bregman 方法与增广拉格朗日乘子法及交替方向乘子法的内在关联, 及其在无约束优化和约束优化问题中的求解应用。
- 针对稀疏信号重建模型、全变差去噪模型、非盲 TV 去卷积模型、图像重建模型以及图像分割模型研究其分裂 Bregman 求解算法, 并通过数值实验验证其有效性。

主要研究内容

研究了分裂 Bregman 方法在图像处理中的一些应用, 主要研究内容如下:

- 研究整理了图像处理的相关数学原理和基础知识, 包括凸和下半连续、次梯度和次微分、Bregman 距离等重要概念, 并研究了岭回归、Lasso 和最佳子集回归基础上的正则化技术。
- 研究了分裂 Bregman 方法与增广拉格朗日乘子法及交替方向乘子法的内在关联, 及其在无约束优化和约束优化问题中的求解应用。
- 针对稀疏信号重建模型、全变差去噪模型、非盲 TV 去卷积模型、图像重建模型以及图像分割模型研究其分裂 Bregman 求解算法, 并通过数值实验验证其有效性。



下一步的工作

主要从理论和应用两个方向上再进行深入研究, 未来的一些展望如下:

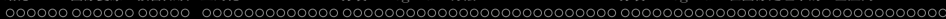
- 分裂 Bregman 方法与传统的一些方法有密切的联系, 如对偶法、拉格朗日乘子法、ADMM、贝叶斯方法, 需要作进一步的研究, 才能对现有算法进行改进和延伸。
- 分裂 Bregman 方法在计算效率上还有很大的改进空间, 这也是进一步研究的方向。
- 分裂 Bregman 方法对正则化参数比较敏感, 目前国内外还没有一种通用的选择方法, 这也是本文需要进一步研究的方向。
- 在图像分割方面我们做的工作比较少, 仅仅是实现了分裂 Bregman 算法用于 GCS 图像分割, 未来可以考虑实现图像修复和字符提取等方面的内容。



下一步的工作

主要从理论和应用两个方向上再进行深入研究, 未来的一些展望如下:

- 分裂 Bregman 方法与传统的一些方法有密切的联系, 如对偶法、拉格朗日乘子法、ADMM、贝叶斯方法, 需要作进一步的研究, 才能对现有算法进行改进和延伸。
- 分裂 Bregman 方法在计算效率上还有很大的改进空间, 这也是进一步研究的方向。
- 分裂 Bregman 方法对正则化参数比较敏感, 目前国内外还没有一种通用的选择方法, 这也是本文需要进一步研究的方向。
- 在图像分割方面我们做的工作比较少, 仅仅是实现了分裂 Bregman 算法用于 GCS 图像分割, 未来可以考虑实现图像修复和字符提取等方面的内容。



下一步的工作

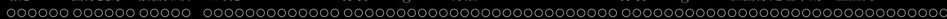
主要从理论和应用两个方向上再进行深入研究, 未来的一些展望如下:

- 分裂 Bregman 方法与传统的一些方法有密切的联系, 如对偶法、拉格朗日乘子法、ADMM、贝叶斯方法, 需要作进一步的研究, 才能对现有算法进行改进和延伸。
- 分裂 Bregman 方法在计算效率上还有很大的改进空间, 这也是进一步研究的方向。
- 分裂 Bregman 方法对正则化参数比较敏感, 目前国内外还没有一种通用的选择方法, 这也是本文需要进一步研究的方向。
- 在图像分割方面我们做的工作比较少, 仅仅是实现了分裂 Bregman 算法用于 GCS 图像分割, 未来可以考虑实现图像修复和字符提取等方面的内容。

下一步的工作

主要从理论和应用两个方向上再进行深入研究, 未来的一些展望如下:

- 分裂 Bregman 方法与传统的一些方法有密切的联系, 如对偶法、拉格朗日乘子法、ADMM、贝叶斯方法, 需要作进一步的研究, 才能对现有算法进行改进和延伸。
- 分裂 Bregman 方法在计算效率上还有很大的改进空间, 这也是进一步研究的方向。
- 分裂 Bregman 方法对正则化参数比较敏感, 目前国内外还没有一种通用的选择方法, 这也是本文需要进一步研究的方向。
- 在图像分割方面我们做的工作比较少, 仅仅是实现了分裂 Bregman 算法用于 GCS 图像分割, 未来可以考虑实现图像修复和字符提取等方面的内容。



下一步的工作

主要从理论和应用两个方向上再进行深入研究, 未来的一些展望如下:

- 分裂 Bregman 方法与传统的一些方法有密切的联系, 如对偶法、拉格朗日乘子法、ADMM、贝叶斯方法, 需要作进一步的研究, 才能对现有算法进行改进和延伸。
- 分裂 Bregman 方法在计算效率上还有很大的改进空间, 这也是进一步研究的方向。
- 分裂 Bregman 方法对正则化参数比较敏感, 目前国内外还没有一种通用的选择方法, 这也是本文需要进一步研究的方向。
- 在图像分割方面我们做的工作比较少, 仅仅是实现了分裂 Bregman 算法用于 GCS 图像分割, 未来可以考虑实现图像修复和字符提取等方面的内容。

参考文献 I



Rudin L I, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms[J]. Physica D: nonlinear phenomena, 1992, 60(1-4): 259-268.



Bregman L M. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming[J]. USSR computational mathematics and mathematical physics, 1967, 7(3): 200-217.bregman1967relaxation



Osher S, Burger M, Goldfarb D, et al. An iterative regularization method for total variation-based image restoration[J]. Multiscale Modeling & Simulation, 2005, 4(2): 460-489.



Wang Y, Yang J, Yin W, et al. A new alternating minimization algorithm for total variation image reconstruction[J]. SIAM Journal on Imaging Sciences, 2008, 1(3): 248-272.



Darbon J, Osher S. Fast discrete optimization for sparse approximations and deconvolutions[J]. preprint, 2007.



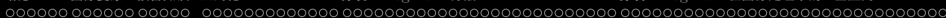
Yin W, Osher S, Goldfarb D, et al. Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing[J]. SIAM Journal on Imaging sciences, 2008, 1(1): 143-168.



Osher S, Mao Y, Dong B, et al. Fast linearized Bregman iteration for compressive sensing and sparse denoising[J]. arXiv preprint arXiv:1104.0262, 2011.



Cai J F, Osher S, Shen Z. Convergence of the linearized Bregman iteration for ℓ_1 -norm minimization[J]. Mathematics of Computation, 2009, 78(268): 2127-2136.



参考文献 II



Goldstein T, Osher S. The split Bregman method for L1-regularized problems[J]. SIAM journal on imaging sciences, 2009, 2(2): 323-343.



Giusti E. Minimal surfaces and functions of bounded variation[J]. Monogr. Math., 1984, 80.



Ekeland I, Temam R. Convex analysis and variational problems[M]. Siam, 1999.



Brègman L M. Relaxation method for finding a common point of convex sets and its application to optimization problems[C]//Doklady Akademii Nauk. Russian Academy of Sciences, 1966, 171(5): 1019-1022.



Goldstein T, Bresson X, Osher S. Geometric applications of the split Bregman method: segmentation and surface reconstruction[J]. Journal of Scientific Computing, 2010, 45(1-3): 272-293.



Esser E. Applications of Lagrangian-based alternating direction methods and connections to split Bregman[J]. CAM report, 2009, 9: 31.



Dacorogna B. Introduction to the Calculus of Variations[M]. World Scientific Publishing Company, 2008.



Donoho D L. Compressed sensing[J]. IEEE Transactions on information theory, 2006, 52(4): 1289-1306.



Montesinos P, Abascal J F P J, Cussó L, et al. Application of the compressed sensing technique to selfgated cardiac cine sequences in small animals[J]. Magnetic resonance in medicine, 2014, 72(2): 369-380.



Chan T F, Golub G H, Mulet P. A nonlinear primal-dual method for total variation-based image restoration[J]. SIAM journal on scientific computing, 1999, 20(6): 1964-1977.



参考文献 III



Chambolle A. An algorithm for total variation minimization and applications[J]. Journal of Mathematical imaging and vision, 2004, 20(1-2): 89-97.



Wang Y, Yang J, Yin W, et al. A new alternating minimization algorithm for total variation image reconstruction[J]. SIAM Journal on Imaging Sciences, 2008, 1(3): 248-272.



Tai X C, Wu C. Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model[C]//International Conference on Scale Space and Variational Methods in Computer Vision. Springer, Berlin, Heidelberg, 2009: 502-513.



Cai J F, Osher S, Shen Z. Linearized Bregman iterations for compressed sensing[J]. Mathematics of Computation, 2009, 78(267): 1515-1536.



Cai J F, Osher S, Shen Z. Linearized Bregman iterations for frame-based image deblurring[J]. SIAM Journal on Imaging Sciences, 2009, 2(1): 226-252.



Yin W. Analysis and generalizations of the linearized Bregman method[J]. SIAM Journal on Imaging Sciences, 2010, 3(4): 856-877.



Huang B, Ma S, Goldfarb D. Accelerated linearized Bregman method[J]. Journal of Scientific Computing, 2013, 54(2-3): 428-453.



Wu C, Tai X C. Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models[J]. SIAM Journal on Imaging Sciences, 2010, 3(3): 300-339.

参考文献 IV



Cai J F, Osher S, Shen Z. Split Bregman methods and frame based image restoration[J]. Multiscale modeling & simulation, 2009, 8(2): 337-369.



Zhang X, Burger M, Osher S. A unified primal-dual algorithm framework based on Bregman iteration[J]. Journal of Scientific Computing, 2011, 46(1): 20-46.



樊启斌, 焦雨领. 变分正则化图像复原模型与算法综述 [J]. 数学进展, 2012, 41(5): 531-546.



万保成. 分裂 Bregman 方法及其在图像处理中的应用 [D]. 吉林大学, 2009.



李磊. 分裂 Bregman 方法在图像处理中的应用 [D]. 大连理工大学, 2010.



易丽娅, 鲁晓磊, 王进军, 等. 图像复原的 Bregman 迭代双正则化方法 [J]. 中国图象图形学报, 2011, 16(3): 350-356.



张慧. Bregman 迭代方法及其在稀疏问题中的应用 [D]. 国防科学技术大学, 2009.



高路. 基于 Bregman 的 CT 稀疏角度迭代重建研究 [D]. 西安电子科技大学, 2014.



何泊. 全变分型图像去噪与去模糊问题 [D]. 河南大学, 2014.



李文斌. 求解全变分去噪问题的双松弛分裂 Bregman 迭代算法 [D]. 吉林大学, 2014.



Nien H, Fessler J A. A convergence proof of the split Bregman method for regularized least-squares problems[J]. arXiv preprint arXiv:1402.4371, 2014.

致谢



- 感谢指导老师周彬副教授，感谢所有任课老师，感谢评委老师，先生学高为吾师，身正为吾范，誓当饮水思源，结草衔环以报，兵终生不忘。
- 感谢父母，感谢同窗，感谢学院。
- 最后感谢岁月，于世间之欢愉与悲苦，许以宽宏。

Thanks for Listening.

欢迎评委老师批评指正

分裂 Bregman 方法在图像处理中的一些应用

Some Applications of Split Bregman Method in Image Processing

指导老师：周彬 副教授

学 生：谭兵