第5章 基于矩阵机制的差分隐私连续数据发布

5.1 引言

随着数字技术的发展，数据越来越多地出现于现实生活当中。数据给人们的生活带来的好处不言而喻。人们不仅可以利用数据进行评估、分析和预测，还可以从中寻找有价值的结论，如“啤酒与尿布”的故事。然而，在享受数据带来的好处的同时，也应该注意到数据中包含的个人隐私信息可能存在泄露的风险。特别是当攻击者怀有恶意时，他就有可能利用已掌握的知识分析用户所发布的数据，并从中挖掘出数据所对应的用户的隐私信息。例如，只需根据4个时空点就能使95%的人泄露其位置信息[1]。因此，如何在发布数据的同时避免数据中包含的隐私被泄露是数据时代亟待解决的问题之一。针对这一问题，各种隐私保护模型被提出。其中，以提供严格数据保护为特点的差分隐私模型[2-4]得到了广泛的认可。该模型被提出后，人们基于该模型开展了很多研究工作。内容涉及直方图发布[5-8]、空间划分发布[9,10]、智能数据分析[11,12]等，有效克服了该模型基于k-匿名[13]和划分[14]的隐私保护方法需要事先对攻击做出假设的不足。差分隐私数据发布研究的关键问题在于如何在保证差分隐私的前提下提高发布数据的可用性。

现有关于差分隐私的数据发布方法大多关注静态发布问题，而现实应用中更多情况下需要发布方法具有连续数据发布的能力。然而，研究表明，这些方法无法应用于连续数据发布问题。为此，本章对差分隐私下的连续数据发布问题展开研究。例如，某医疗数据库中记录了每个月的入院病人的信息，其中病人感染HIV的情况为敏感信息。表5.1展示了其中3个月的数据示例。同时，出于某研究目的，医院将按月统计并公布入院的HIV病人数。公布的数据形如表5.2所示，医院累计当前入院并且感染HIV的病人并于当月发布最新数据。与数据静态发布不同，在医院发布完每个月的统计信息后，该数据并非不再改变，而是在下个月将得到更新。更重要的是，在发布每一次数据的过程中，以后需要发布的数据是无法被预知的。该问题的核心是在满足差分隐私的条件下，寻找更精确、更高效的连续数据发布方法。

表 5.1　病人感染 HIV 情况

Name	HIV+	Name	HIV+	Name	HIV+
Alice	Yes	Alan	No	Andy	No
Bob	No	Ben	Yes	Bill	No
Carol	Yes	Cari	No	Chen	Yes
⋮	⋮	⋮	⋮	⋮	⋮

表 5.2　入院的 HIV 病人数

Month	HIV+	Sum of HIV+
1	531	531
2	392	923
3	426	1349
⋮	⋮	⋮

以上连续发布问题的一种朴素解决方案[15]是直接将前一个月发布的 HIV 病人数与本月新增的 HIV 病人数相加，然后再添加噪声使其满足差分隐私。该方案导致每一次发布数据的噪声的均方误差线性累加，最终使发布的数据失去可用性。文献[15]针对该问题提出了一种基于二叉树的发布方法。然而，此方法仅仅引入二叉树模拟发布，并未对精确性提出有效优化方法。为此，本章以提高发布数据的精确性为主要目标，将矩阵机制引入差分隐私连续数据发布问题中，以期设计出高效的基于矩阵机制的差分隐私连续数据发布方法，可有效满足大规模连续数据发布的要求。

5.2　基础知识与问题提出

在差分隐私的数据发布中，为提高数据发布的精度，Hay 等人[7]和 Xiao 等人[8]分别提出基于一致性调节的区间树方法和小波变换方法，实现了较高精度的数据发布。然而，上述两种方法只适用于差分隐私下的数据静态发布，无法应用于差分隐私下的连续数据发布。Chan 等人[15]提出了两种利用二叉树结构进行连续数据发布的方法。第一种方法是构建一棵叶节点数量为 2^m 的完全二叉树，然后利用模拟二叉树统计发布的过程进行连续数据发布。第二种方法是第一种方法的改进版本，它试图通过调整二叉树各层节点的隐私预算分配来达到无限发布的效果。研究表明，虽然第一种方法相比于朴素方法，数据发布的精确性有显著的提升，然而该方法仅仅引入二叉树结构来模拟发布过程，并未做进一步改进，因此数据发布的精确性仍有较大的提升空间；第二种方法的隐私预算分配并不合理，导致发布数据的误差远大于第一种方法。

为了解决差分隐私下的线性查询问题，Li 等人[16]提出了基于矩阵机制的批量查询方法。其基本思想是通过寻找策略矩阵对线性查询进行优化，进而提高发布数据的精确性。然而，该文献提出的矩阵机制仅能满足小规模数据集和查询负载的要求。此外，它还很容易产生次优化的查询策略，使得结果往往并不理想。为此，Yuan 等人[17]利用了负载矩阵低秩

的性质进行优化，提出低秩矩阵机制，在一定程度上改善了原有矩阵机制的不足，提升了数据发布效率与精确性。然而，该文献提出的优化查询使用半正定规划算法，同样只能适用于小规模数据集和查询负载的要求。由于本章研究的连续数据发布问题本质上也是线性查询问题，因此拟利用矩阵机制，结合连续数据发布问题本身具有的一些特性，设计出精确性更高的高效算法，使之具有大规模连续数据发布的能力。

矩阵机制是一种针对差分隐私下线性查询问题的优化方法。它通过将查询集 $\boldsymbol{Q}$ 转换成负载矩阵 $\boldsymbol{W}$，然后寻找最优策略矩阵 $\boldsymbol{M}$ 来实现差分隐私下线性查询的优化。其中，查询集 $\boldsymbol{Q}$ 是一组线性查询的集合，满足 $\boldsymbol{Q}=\{q_1,q_2,\cdots,q_n\}$。每个线性查询表示如下：

$$q=\sum_{1}^{m}w_i x_i$$

其中，$\boldsymbol{X}=(x_1,x_2,\cdots,x_m)^{\mathrm{T}}$ 为数据向量，w_i 为该查询在分量 x_i 上的权重。负载矩阵 $\boldsymbol{W}$ 由每组线性查询的权重组成，并满足

$$\boldsymbol{Q}=\boldsymbol{W}\boldsymbol{X}=\left(\sum_{j=1}^{m}W_{1j}x_j,\sum_{j=1}^{m}W_{2j}x_j,\cdots,\sum_{j=1}^{m}W_{nj}x_j\right)^{\mathrm{T}}$$

原始的矩阵机制[16]通过直接寻找策略矩阵 $\boldsymbol{M}$ 的方式求解问题，这种做法的效率和优化效果都不够理想。而低秩矩阵机制[17]则是采用分解负载矩阵的方法来寻找优化策略。在该机制下，它将 $\boldsymbol{W}$ 分解成两个矩阵 $\boldsymbol{B}$、$\boldsymbol{M}$。其中 $\boldsymbol{M}$ 表示低秩矩阵下的策略矩阵，且满足 $\boldsymbol{W}=\boldsymbol{B}\boldsymbol{M}$。通过对中间结果 $\boldsymbol{M}\boldsymbol{X}$ 添加噪声的方式减少误差。其形式化表示如下：

$$A(\boldsymbol{W},\boldsymbol{X})=\boldsymbol{B}\left(\boldsymbol{M}\boldsymbol{X}+\frac{\Delta_{\boldsymbol{M}}}{\varepsilon}\widetilde{L}_n\right) \tag{5.1}$$

文献[17]指出，式(5.1)的敏感度 $\Delta_{\boldsymbol{M}}$ 与策略矩阵的列范式相等，即 $\Delta_{\boldsymbol{M}}=\boldsymbol{M}_1$。而低秩矩阵的均方误差由下式求得：

$$\frac{2}{\varepsilon^2}\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})\Delta_{\boldsymbol{M}}{}^2 \tag{5.2}$$

研究表明，差分隐私下的数据连续发布问题能够被转换成基于矩阵机制的优化问题。只需将每一次发布视为一个查询，然后将所有发布过程视为查询负载，并转换成相应的矩阵，利用矩阵机制进行求解。

5.3 基于矩阵机制的差分隐私连续数据发布

考虑一个随着时间增长会不断产生记录并被添加进来的记录流。该记录流是数据发布的来源。记录流的每条记录都有需要保护的属性 σ，满足 $\sigma\in\{0,1\}$，并假设记录集 $\mathbb{A}_i$ 表示第 $i-1$ 次和第 i 次发布之间记录流所中被添加的记录的集合。由 $\mathbb{A}_i$ 可求出第 i 次发布的数据增量 $a_i=|\{\sigma|\sigma\in\mathbb{A}_i\text{且}\sigma=1\}|$。

定义 5.1(连续数据发布) 对于记录流，数据发布者随着记录流中的记录增长按照某种规则多次发布当前记录流中满足 $\sigma=1$ 的记录数的行为即为连续数据发布。假设第 i 次发布的累计数据为 s_i，那么 s_i 满足

$$s_i=s_{i-1}+a_i=\sum_{j=1}^{i}a_j \tag{5.3}$$

差分隐私下的连续数据发布行为即通过某种隐私算法 $A(*)$ 发布添加了噪声的累计数据 $\tilde{s}_i$，从而使数据连续发布的结果满足 ε-差分隐私。同时，在此基础上，本节还要求提出的算法能够精确而且高效地发布数据。

为了避免数据连续发布的敏感度过高而影响数据发布精度的问题，本节主要考虑了满足 ε-差分隐私的次数受限的数据连续发布算法。

定义 5.2（次数受限的数据连续发布算法）　如果隐私算法 $A(*)$ 至多接受并发布 N 次满足 ε-差分隐私的统计数据，则称该算法为次数受限的数据连续发布算法。

上述问题能够转换成线性查询问题，而矩阵机制能够对线性查询问题进行优化。为了提出更精确的数据连续发布算法，本章将这一问题与矩阵机制相结合，并在此基础上寻找快速发布算法。而且，由于矩阵机制是成熟的且经过严格检验的满足 ε-差分隐私[17]的隐私保护机制。因此，基于矩阵机制的线性查询算法只要符合式(5.1)的形式就保证了该算法满足 ε-差分隐私。

利用矩阵机制优化前，需要将式(5.3)转换成负载矩阵 $\boldsymbol{W}$：

$$\boldsymbol{W}=\begin{bmatrix}1 & 0 & 0 & \cdots\\ 1 & 1 & 0 & \cdots\\ 1 & 1 & 1 & \cdots\\ \vdots & \vdots & \vdots & \ddots\end{bmatrix} \tag{5.4}$$

可以看出，数据连续发布下的负载矩阵 $\boldsymbol{W}$ 为下三角矩阵，且满足 $W_{ij}=1, i<j$。

根据矩阵机制的特征及一般研究过程[16,17]。本章将通过以下步骤对该问题展开研究：

(1) 寻找初始的策略矩阵 $\boldsymbol{M}$，将 $\boldsymbol{W}$ 分解为矩阵 $\boldsymbol{B}$、$\boldsymbol{M}$，使矩阵机制能够较为精确地发布数据。

(2) 对策略矩阵 $\boldsymbol{M}$ 进行研究，寻找优化策略以优化数据发布的精确性。

(3) 综合分析矩阵 $\boldsymbol{B}$、$\boldsymbol{M}$ 以及优化策略的性质，在保证数据发布的精确性得到优化的前提下，提出高效的优化算法。

5.4　隐私连续数据发布算法

5.4.1　策略矩阵的构建

由于数据随着发布过程动态产生，未来数据无法得知，因此只能根据当前以及过往的数据进行优化。这一特征反映到矩阵机制时，就要求矩阵机制所应用的策略矩阵为下三角矩阵。通过各种数据结构的对比研究，发现树状数组[19]更加适合构造基于矩阵机制的数据连续发布方法的策略矩阵。它能够自然并且快速求解数列的前 k 项和，符合连续数据发布的基本特征。同时，初步研究表明，将它与差分隐私结合而提出的隐私保护模型能够达到与基于二叉树的连续数据发布方法[15]相当的精确性。同时，深入研究表明，结合树状数组的差分隐私模型在实现的巧妙性以及进一步优化的潜力方面均比后者略胜一筹。

树状数组主要针对以下问题：给定 N 个实数，记为 $a_1, a_2, \cdots, a_n$，要求快速求出前 k 项的和。记前 k 项的和为 s_k，则 $s_k=\sum_{j=1}^{k}a_j$。

针对该问题，利用树状数组提出如下解决方法。该方法计算了中间统计量 c_i。而 c_i 由以下公式求得：

$$c_i = \sum_{j=i-\text{lowbit}(i)+1}^{i} a_j \tag{5.5}$$

其中，函数 lowbit(x)表示将正整数 x 写成二进制形式后，将该二进制数值为 1 的最低位的数置为 1，其余位均置为 0。例如，当 $x=10$ 时，其对应的二进制数为$(1010)_2$。从低往高，2^0 位为 0，2^1 位为 1，那么，lowbit(x)的输出值就将 2^1 位置为 1，其他位均置为 0，即$(0010)_2=2$。再将其转成十进制数，就得到 lowbit(10)＝2。

该函数的算法如下。

算法 5.1 lowbit(x)

输入：正整数 x

输出：x 对应的 lowbit 值

1. $p \leftarrow 0, y \leftarrow 1$;
2. while x mod 2＝0
3. $\quad y \leftarrow 2 * y; x \leftarrow \left\lfloor \frac{x}{2} \right\rfloor$;
4. wend
5. return y

结合算法 5.1 以及式(5.5)，按照树状数组求解中间统计量的方式构造策略矩阵 M，算法如下。

算法 5.2 求解策略矩阵 $\boldsymbol{M}$

输入：发布次数 N

输出：策略矩阵 $\boldsymbol{M}$

1. $\boldsymbol{M} \leftarrow \boldsymbol{0}_{N\times N}$; //初始化为零矩阵
2. for p＝1 to N do
3. pt$\leftarrow p$;
4. while pt$<N$
5. $\boldsymbol{M}_{\text{pt},p} \leftarrow 1$; //更新矩阵元素
6. $\quad$ pt$\leftarrow$pt＋lowbit(pt);
7. wend
8. end for
9. return $\boldsymbol{M}$

下面讨论矩阵 $\boldsymbol{B}$ 的求解。由 $\boldsymbol{W}=\boldsymbol{B}\boldsymbol{M}$ 以及 $\boldsymbol{M}_N$ 的可逆性可得 $\boldsymbol{B}=\boldsymbol{W}_N\boldsymbol{M}_N^{-1}$。因此，只需根据树状数组的性质就能够快速地求解 $\boldsymbol{B}$。而树状数组的求和操作按照以下公式进行求解：

$$s_i = c_i + s_{i-\text{lowbit}(i)} \tag{5.6}$$

算法 5.3 求解矩阵 $\boldsymbol{B}$

输入：发布次数 N

输出：矩阵 $\boldsymbol{B}$

1. $\boldsymbol{B}\leftarrow\boldsymbol{0}_{N\times N}$；　　//初始化为零矩阵
2. for $p=1$ to N do
3. pt$\leftarrow p$；
4. while pt>0
5. 　　$\boldsymbol{B}_{\mathrm{pt},p}\leftarrow 1$　　//更新矩阵元素
6. 　　pt$\leftarrow$pt-lowbit(pt)；
7. wend
8. end for
9. return $\boldsymbol{B}$

上述算法结合低秩矩阵的表达式，即可得到基于树状数组的数据连续发布的表达式：

$$A(\boldsymbol{W},\boldsymbol{D})=\boldsymbol{B}\left(\boldsymbol{M}_N\boldsymbol{X}+\frac{\Delta_{\boldsymbol{M}_N}}{\varepsilon}\widetilde{\boldsymbol{L}}_n\right) \tag{5.7}$$

接下来分析该策略矩阵所产生的均方误差的情况。关于 $\boldsymbol{M}_N$ 和 $\boldsymbol{B}$ 有如下两个相关定理。

定理 5.1　$\|\boldsymbol{M}_N\|_1=\|\boldsymbol{M}_N(:,1)\|_1\geqslant\|\boldsymbol{M}_N(:,j)\|_1(j>1)$，且 $\|\boldsymbol{M}_N\|_1=\lfloor\log_2 N\rfloor+1$。

证明：通过算法 5.2 研究矩阵 $\boldsymbol{M}_N(:,1)$的构造情况。可得当第一次迭代($p=1$)时，有 pt$=p=1=2^0$。设第 t 次迭代时有 pt$=2^{t-1}$，由更新表达式有 pt$=$pt$-$lowbit(pt)$=2^{t-1}+2^{t-1}=2^t$。而根据步骤 6 的判断条件，pt$>N$ 时，该列构造结束。因此有，$2^{t-1}\leqslant N\Rightarrow t\leqslant\log_2 N+1$。此时 $\boldsymbol{M}_{\mathrm{pt},p}=1$ 更新了$\lfloor\log_2 N\rfloor+1$ 次，因此 $\|\boldsymbol{M}_N(:,1)\|_1=\lfloor\log_2 N\rfloor+1(j>1)$。

对于 $j>1$ 的情况，假设第 t 次迭代时 pt$>2^{t-1}$且 lowbit(pt)$\geqslant 2^{t-1}$，由第一次迭代有 pt$=j>1$ 可知满足该条件。根据 lowbit 函数的性质，有 lowbit(pt')$=$lowbit(pt$+$lowbit(pt))$\geqslant$lowbit(2lowbit(pt))$=$2lowbit(pt)$\geqslant 2^t$，从而 pt$'=$pt$+$lowbit(pt)$>2^t$。很显然，根据 $pt>2^{t-1}$可推得，$j>1$ 时的更新次数不大于 $j=1$ 时。因此，$\|\boldsymbol{M}_N(:,1)\|_1\geqslant\|\boldsymbol{M}_N(:,j)\|_1(j>1)$，$\|\boldsymbol{M}_N\|_1=\max_j\|\boldsymbol{M}_N(:,j)\|_1=\lfloor\log_2 N\rfloor+1$。

定理 5.1 得证。

定理 5.2　构造矩阵 $\boldsymbol{B}$ 的第 p 行的迭代次数为将 p 表示为二进制数$(p)_2$ 时$(p)_2$ 中包含的 1 的个数。

证明：根据算法 5.3 的步骤 6 有操作 pt$=$pt$-$lowbit(pt)，该操作的结果是将$(\mathrm{pt})_2$ 中值为 1 的最低位置为 0。而由步骤 3，pt 由 p 进行初始化。说明$(p)_2$ 中包含多少个 1，迭代次数就进行了多少次。

定理 5.2 得证。

由式(5.2)推出该策略矩阵所产生的均方误差为

$$\frac{2}{\varepsilon^2}\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})\Delta^2_{\boldsymbol{M}_N}=\frac{2}{\varepsilon^2}\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B})(\lfloor\log_2 N\rfloor+1)^2$$

同时，结合定理 5.2 可知，$\boldsymbol{B}$ 的每一行至多有 $O(\log_2 N)$个元素为 1，其余均为 0。因此，$\boldsymbol{B}$ 中有 $O(N\log_2 N)$个元素为 1。即 trace($\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}$)的数值复杂度也为 $O(N\log_2 N)$。又由定理 5.1 可知，$\boldsymbol{M}_N$ 的列范数$\lfloor\log_2 N\rfloor+1$ 的复杂度为 $O(\log_2 N)$。因而，根据式(5.2)可以求得总体的均方误差为 $O(N\log_2^3 N)$，而每条查询的均方误差则为 $O(\log_2^3 N)$。

综上所述，由树状数组构造的策略矩阵满足低敏感性以及均方误差复杂度低的特点，初步具备了在差分隐私下较为精确地进行连续数据发布的能力。然而，仅仅利用树状数组构造的策略矩阵并不能达到本章的精确性要求。接下来，将在此基础上寻找更精确的数据发布算法。

一般而言，可以用发布数据的一致性调节[7]、策略矩阵的权重系数调节等方法提高数据发布的精确性。然而，研究表明该问题已经满足线性一致性，具体分析如下。

定义 5.3（线性一致性） 对于负载矩阵 $\boldsymbol{W}$，记未加噪的查询结果为 $\boldsymbol{Y}=\boldsymbol{W}Q(\boldsymbol{D})$，通过低秩矩阵机制获得的查询结果为 $\boldsymbol{Y}'=A(\boldsymbol{W},\boldsymbol{D})$。$A(\boldsymbol{W},\boldsymbol{D})$满足线性一致性当且仅当对任意可以表示成 $z=v\boldsymbol{Y}$ 的行向量 v 都有 $v\boldsymbol{Y}'$为定值，其中 z 为可以用 $\boldsymbol{Y}$ 线性表示的统计量。

定理 5.3 当式(5.1)中的矩阵 $\boldsymbol{M}$ 为行满秩矩阵时，低秩矩阵机制满足线性一致性。

证明：当矩阵 $\boldsymbol{L}$ 为行满秩矩阵时，求得其右逆矩阵 $\boldsymbol{M}^{+}=\boldsymbol{M}^{\mathrm{T}}(\boldsymbol{M}\boldsymbol{M}^{\mathrm{T}})^{-1}$，满足 $\boldsymbol{M}\boldsymbol{M}^{+}=\boldsymbol{I}$。由于 $\boldsymbol{W}=\boldsymbol{B}\boldsymbol{M}$，因此 $\boldsymbol{B}=\boldsymbol{W}\boldsymbol{M}^{+}$。

任取统计查询 z，有多个 v_i 满足 $z=v_i\boldsymbol{W}\boldsymbol{X}$（其中 v_i 之间互不相等）。

将其代入式(5.1)中，可得统计后的结果，令 z_i'表示由 v_i 求得的查询结果：

$$z_i'=v_i\boldsymbol{B}\left(\boldsymbol{M}\boldsymbol{X}+\frac{\Delta_{\boldsymbol{M}}}{\varepsilon}\widetilde{L}_n\right)=v_i\boldsymbol{W}\boldsymbol{M}^{+}\left(\boldsymbol{M}\boldsymbol{X}+\frac{\Delta_{\boldsymbol{M}}}{\varepsilon}\widetilde{L}_n\right)=z\boldsymbol{M}^{+}\left(\boldsymbol{M}\boldsymbol{X}+\frac{\Delta_{\boldsymbol{M}}}{\varepsilon}\widetilde{L}_n\right)$$

经过化简可以看出，z_i'是与 v_i 无关的噪声统计量。因此，z_i'的值是相等的。

定理 5.3 得证。

而矩阵 $\boldsymbol{M}_N$ 可知为可逆矩阵。结合定理 5.3，可得出推论：由树状数组构造基于矩阵机制的数据连续发布方法满足线性一致性，因此无法从线性一致性的角度提高数据发布的一致性。本章将在 5.4.2 节对策略矩阵的权重系数调节问题作进一步研究。

5.4.2 查询均方误差的降低

5.4.1 节分析了差分隐私下的连续数据发布的性质，并通过树状数组构造出基于矩阵机制的策略矩阵。本节在 5.3 节所描述的步骤的基础上进行优化。进一步研究矩阵 $\boldsymbol{M}_N$，可发现该矩阵未饱和。以 $\boldsymbol{M}_3$ 为例，表示如下：

$$\boldsymbol{M}_3=\begin{pmatrix}1&0&0\\1&1&0\\0&0&1\end{pmatrix}\Rightarrow\boldsymbol{M}_3'=\begin{pmatrix}1&0&0\\1&1&0\\0&0&2\end{pmatrix}$$

计算可得 $\|\boldsymbol{M}_3\|_1=2$。若将 $\boldsymbol{M}_3$ 的第三行乘以 2，得到 $\boldsymbol{M}_3'$，依旧满足 $\|\boldsymbol{M}_3'\|_1=2$，并不会影响整体的敏感度。同时，矩阵 $\boldsymbol{B}$ 也应转换为 $\boldsymbol{B}'$。

$$\boldsymbol{B}=\begin{pmatrix}1&0&0\\0&1&0\\0&1&1\end{pmatrix}\Rightarrow\boldsymbol{B}'=\begin{pmatrix}1&0&0\\0&1&0\\0&1&0.5\end{pmatrix}$$

根据式(5.2)，可以直接求出转换前后两者之间的均方误差。转换前为$\frac{32}{\varepsilon^2}$，转换后为$\frac{26}{\varepsilon^2}$。经过转换，均方误差降低了，这说明直接由树状数组构造的矩阵 $\boldsymbol{M}_N$ 优化得还不够彻底。经研究发现，可以通过在 $\boldsymbol{M}_N$ 前面乘一个对角阵的方式提高精确性。

令 $\boldsymbol{\Sigma}_N=\begin{pmatrix}\lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3\end{pmatrix}$ 表示 $N\times N$ 的系数对角阵，则可将式(5.7)拓展如下：

$$A(\boldsymbol{W},\boldsymbol{D})=\boldsymbol{B}\boldsymbol{\Sigma}_N^{-1}\left(\boldsymbol{\Sigma}_N\boldsymbol{M}_N X+\frac{\Delta_{\boldsymbol{\Sigma}_N\boldsymbol{M}_N}}{\varepsilon}\widetilde{L}_n\right) \tag{5.8}$$

式(5.8)即为添加系数对角阵后的隐私保护机制。当 $\boldsymbol{\Sigma}_N=\boldsymbol{I}_N$ 时，该公式与式(5.7)等价。对应的均方误差公式如下：

$$\frac{2}{\varepsilon^2}\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\Sigma}_N^{-2})\Delta_{\boldsymbol{\Sigma}_N\boldsymbol{M}_N}{}^2 \tag{5.9}$$

根据文献[17]的结论，令 $\boldsymbol{B}'=\alpha\boldsymbol{B}\boldsymbol{\Sigma}_N^{-1}$，$L'=\alpha^{-1}\boldsymbol{\Sigma}_N\boldsymbol{M}_N$，则有 $\frac{2}{\varepsilon^2}\mathrm{trace}(\boldsymbol{B}'^{\mathrm{T}}\boldsymbol{B}')\Delta_{L'}^2=\frac{2}{\varepsilon^2}\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\Sigma}_N^{-2})\Delta_{\boldsymbol{\Sigma}_N\boldsymbol{M}_N}{}^2$。因此，可将 $\Delta_{\boldsymbol{\Sigma}_N\boldsymbol{M}_N}$ 限制为 $|\boldsymbol{\Sigma}_N\boldsymbol{M}_N|_1\leqslant 1$，最小化 $\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\Sigma}_N^{-2})$，则该优化问题可表示为如下形式：

$$\begin{aligned}&\min_{\boldsymbol{\Sigma}_N} f(\boldsymbol{\Sigma}_N)=\frac{2}{\varepsilon^2}\mathrm{trace}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\Sigma}_N^{-2})\\ &\text{s.t.}\quad |\boldsymbol{\Sigma}_N\boldsymbol{M}_N|_1\leqslant 1\end{aligned}$$

当上式取得最优解时，即等价于式(5.10)取得最优解。为简化推理过程，在式(5.10)中忽略了常数$\frac{2}{\varepsilon^2}$，实际计算时加上该常数即可。

$$\begin{aligned}&\min_{\boldsymbol{\Sigma}_N} f(\boldsymbol{\Sigma}_N)=\sum_{i=1}^{N}\frac{\boldsymbol{B}(:,i)^{\mathrm{T}}\boldsymbol{B}(:,i)}{\lambda_i^2}\\ &\text{s.t.}\quad \boldsymbol{M}_N^{\mathrm{T}}\begin{pmatrix}\lambda_1\\ \vdots\\ \lambda_N\end{pmatrix}\leqslant\boldsymbol{I}_{N\times 1}\\ &\lambda_i>0\end{aligned} \tag{5.10}$$

其中 $\boldsymbol{B}(:,i)$表示矩阵 $\boldsymbol{B}$ 的第 i 列。

由分析可知，式(5.10)是一个线性约束下的凸优化问题。针对该问题，可直接采用 SQP 方法[20]求得最优解。然而 SQP 方法是一种时间复杂度很高的算法，无法满足大规模数据的要求。实验表明，对于一般计算机，该方法最多只能满足 $N<1000$ 的求解规模。因此，需要进一步研究更快速的方法求得式(5.10)的最优解。

5.4.3　最小误差的快速求解

由于 SQP 方法的时间复杂度很高，对于大规模数据，对角阵 $\boldsymbol{\Sigma}_N$ 求解是无法完成的。因此，有必要对 $\boldsymbol{\Sigma}_N$ 的求解进一步优化，并提出高效的解决方案。利用 $\boldsymbol{M}_N$ 和 $\boldsymbol{B}$ 之间的特殊性质，本节提出一种高效的求解最小误差的算法——快速对角阵优化算法(Fast Diagonal Matrix Optimization Algorithm, FDA)。当 $N=2^m-1$ 时，该算法可以在 $O(\log N)$的时间复杂度下求解 $\boldsymbol{\Sigma}_N$ 的任意系数值λ_i。该算法与未使用 $\boldsymbol{\Sigma}$ 前的方法有相当的求解效率，因此它保证了在不影响算法时间复杂度的前提下提高了隐私数据发布的精确性。该算法是基于以下定理提出的。

定理 5.4 令 $\boldsymbol{\Sigma}_N^*$ 表示 $\boldsymbol{\Sigma}_N$ 最优系数矩阵，则存在待定系数 α 使得 $\boldsymbol{\Sigma}_{2^m-1}^*$ 和 $\boldsymbol{\Sigma}_{2^{m-1}-1}^*$ 之间满足以下递推关系：

$$\boldsymbol{\Sigma}_{2^m-1}^* = \begin{pmatrix} \alpha\boldsymbol{\Sigma}_{2^m-1}^* & & \\ & 1-\alpha & \\ & & \boldsymbol{\Sigma}_{2^m-1}^* \end{pmatrix} \tag{5.11}$$

证明：对于矩阵 $\boldsymbol{B}_{2^m-1}$ 进一步分析，可发现它满足以下特性：令 $a<2^{m-1}$，$b=2^{m-1}+a$。将其写成二进制形式可以描述为：$a=(x_{m-2}x_{m-3}\cdots x_0)_2$ 和 $b=(1x_{m-2}x_{m-3}\cdots x_0)_2$。根据算法 5.2，不难发现 $\boldsymbol{B}_{2^m-1}(a,t)=1(t>0)$ 当且仅当 $\boldsymbol{B}_{2^m-1}(b,2^{m-1}+t)=1$。同时，由于 b 的 2^{m-1} 位为 1，因此 $\boldsymbol{B}_{2^m-1}(b,2^{m-1})=1$。

通过以上分析，可将 $\boldsymbol{B}_{2^m-1}$ 写成如下形式：

$$\boldsymbol{B}_{2^m-1} = \begin{pmatrix} \boldsymbol{B}_{2^{m-1}-1} & \boldsymbol{O}_{(2^{m-1}-1)\times 1} & \boldsymbol{O}_{(2^{m-1}-1)\times(2^{m-1}-1)} \\ \boldsymbol{O}_{1\times(2^{m-1}-1)} & 1 & \boldsymbol{O}_{1\times(2^{m-1}-1)} \\ \boldsymbol{O}_{(2^{m-1}-1)\times(2^{m-1}-1)} & \boldsymbol{I}_{(2^{m-1}-1)\times 1} & \boldsymbol{B}_{2^{m-1}-1} \end{pmatrix} \tag{5.12}$$

通过式(5.12)得

$$\begin{aligned} \boldsymbol{B}_{2^m-1}(:,b)^{\mathrm{T}}\boldsymbol{B}_{2^m-1}(:,b) &= \begin{pmatrix} \boldsymbol{O}_{(2^{m-1})\times 1} \\ \boldsymbol{B}_{2^{m-1}-1}(:,b-2^{m-1}) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \boldsymbol{O}_{(2^{m-1})\times 1} \\ \boldsymbol{B}_{2^{m-1}-1}(:,b-2^{m-1}) \end{pmatrix} \\ &= \boldsymbol{B}_{2^{m-1}-1}(:,a)^{\mathrm{T}}\boldsymbol{B}_{2^{m-1}-1}(:,a) \end{aligned}$$

下面分析 $\boldsymbol{M}_{2^m-1}$ 与 $\boldsymbol{M}_{2^{m-1}-1}$ 之间的关系。

根据算法 5.4，有 $\boldsymbol{M}_{2^m-1}(t,a)=1(1\leqslant t\leqslant 2^{m-1}-1)$ 当且仅当 $\boldsymbol{M}_{2^m-1}(2^{m-1}+t,b)=1$，满足 $\forall\, 1\leqslant t\leqslant 2^{m-1}-1$，$\boldsymbol{M}_{2^m-1}(2^{m-1},t)=1$。

因此，将 $\boldsymbol{M}_{2^m-1}$ 与 $\boldsymbol{M}_{2^{m-1}-1}$ 写成如下递推关系：

$$\boldsymbol{M}_{2^m-1} = \begin{pmatrix} \boldsymbol{M}_{2^{m-1}-1} & \boldsymbol{O}_{(2^{m-1}-1)\times 1} & \boldsymbol{O}_{(2^{m-1}-1)\times(2^{m-1}-1)} \\ \boldsymbol{I}_{1\times(2^{m-1}-1)} & 1 & \boldsymbol{O}_{1\times(2^{m-1}-1)} \\ \boldsymbol{O}_{(2^{m-1}-1)\times(2^{m-1}-1)} & \boldsymbol{O}_{(2^{m-1}-1)\times 1} & \boldsymbol{M}_{2^{m-1}-1} \end{pmatrix} \tag{5.13}$$

令 $\boldsymbol{R}_N$ 表示 $\boldsymbol{\Sigma}_N$ 的对角线元素组成的列向量，$\boldsymbol{R}_N=(\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_N)^{\mathrm{T}}$。当 $N=2^m-1$ 时，可将 $\boldsymbol{R}_{2^m-1}$ 拆分成 3 个部分：

$$\boldsymbol{R}_{2^m-1} = (\boldsymbol{R}_{2^m-1}^{(1)\mathrm{T}} \quad \lambda_{2^{m-1}} \quad \boldsymbol{R}_{2^m-1}^{(2)\mathrm{T}})^{\mathrm{T}} \tag{5.14}$$

其中，$\boldsymbol{R}_{2^m-1}^{(1)}=(\lambda_1 \quad \lambda_2 \quad \cdots \quad \lambda_{2^{m-1}-1})^{\mathrm{T}}$，$\boldsymbol{R}_{2^m-1}^{(2)}=(\lambda_{2^{m-1}+1} \quad \lambda_{2^{m-1}+2} \quad \cdots \quad \lambda_{2^m-1})^{\mathrm{T}}$。

对于式(5.10)，也可将其拆分为以下 3 个子部分：

$$f(R_{2^m-1}) = \sum_{i=1}^{2^{m-1}-1} \frac{\boldsymbol{B}_{2^m-1}^{\mathrm{T}}(:,i)\boldsymbol{B}_{2^m-1}(:,i)}{\lambda_i^2} \tag{①}$$

$$+ \frac{\boldsymbol{B}_{2^m-1}^{\mathrm{T}}(:,2^{m-1})\boldsymbol{B}_{2^m-1}(:,2^{m-1})}{\lambda_{2^{m-1}}{}^2} \tag{②}$$

$$+ \sum_{i=1}^{2^{m-1}-1} \frac{\boldsymbol{B}_{2^m-1}^{\mathrm{T}}(:,i)\boldsymbol{B}_{2^m-1}(:,i)}{\lambda_{2^{m-1}+i}{}^2} \tag{③}$$

令 $f^{(i)}(*)$ 分别表示这 3 个子部分，从而将上述表达式转换为 3 个子部分的和：

$$f(\boldsymbol{R}_{2^m-1}) = f^{(1)}(\boldsymbol{R}_{2^m-1}^{(1)}) + f^{(2)}(\lambda_{2^{m-1}}) + f^{(3)}(\boldsymbol{R}_{2^m-1}^{(2)})$$

而对于其限制条件，有 $\boldsymbol{M}_{2^m-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1}\leqslant\boldsymbol{I}_{(2^m-1)\times 1}$。依照式(5.13)和式(5.14)展开得

$$\begin{pmatrix} \boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1}^{(1)}+\lambda_{2^{m-1}}\boldsymbol{I}_{(2^{m-1}-1)\times 1} \\ \lambda_{2^{m-1}} \\ \boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1}^{(2)} \end{pmatrix} \leqslant \boldsymbol{I}_{(2^m-1)\times 1} \tag{5.15}$$

由式(5.15)可将限制条件分解成以下3个子条件：

$$\begin{cases} \boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1}^{(1)} \leqslant (1-\lambda_{2^{m-1}})\boldsymbol{I}_{(2^{m-1})\times 1} & ① \\ \lambda_{2^{m-1}} \leqslant 1 & ② \\ \boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1}^{(2)} \leqslant \boldsymbol{I}_{(2^{m-1}-1)\times 1} & ③ \end{cases}$$

通过以上3个子限制条件，可知子条件①受限于子条件②中的$\lambda_{2^{m-1}}$的取值。因此，先假设$\lambda_{2^{m-1}}$为待定系数，令$\lambda_{2^{m-1}}=1-\alpha(0<\alpha<1)$。

式(5.10)取最优时，子部分①满足：

$$f(\boldsymbol{R}_{2^m-1}^{*})=\min_{\boldsymbol{\Sigma}_{2^m-1}} f(\boldsymbol{R}_{2^m-1}) \Rightarrow f^{(1)}(\boldsymbol{R}_{2^m-1}^{*(1)})=\min_{\Sigma_{2^m-1}} f^{(1)}(\boldsymbol{R}_{2^m-1}^{(1)})$$

$$\boldsymbol{M}_{2^m-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1} \leqslant \boldsymbol{I}_{(2^m-1)\times 1} \Rightarrow \boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1}^{(1)} \leqslant (1-\lambda_{2^{m-1}})\boldsymbol{I}_{(2^{m-1})\times 1}$$

由于

$$\boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}\boldsymbol{R}_{2^m-1}^{(1)} \leqslant \alpha\boldsymbol{I}_{(2^{m-1})\times 1} \Leftrightarrow \boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}\left(\frac{1}{\alpha}\boldsymbol{R}_{2^m-1}^{(1)}\right) \leqslant \boldsymbol{I}_{(2^{m-1})\times 1}$$

因此，令$\mu_i=\frac{1}{\alpha}\lambda_i$，$\boldsymbol{Q}_N=\frac{1}{\alpha}\boldsymbol{R}_N=(\mu_1 \quad \mu_2 \quad \cdots \quad \mu_N)$，并将其代入式(5.10)的子部分①后有

$$f^{(1)}(\boldsymbol{R}_{2^m-1}^{(1)})=\sum_{i=1}^{2^{m-1}-1}\frac{\boldsymbol{B}_{2^m-1}^{\mathrm{T}}(:,i)\boldsymbol{B}_{2^m-1}(:,i)}{{\lambda_i}^2}=\frac{1}{\alpha^2}\sum_{i=1}^{2^{m-1}-1}\frac{\boldsymbol{B}_{2^m-1}^{\mathrm{T}}(:,i)\boldsymbol{B}_{2^m-1}(:,i)}{(\mu_i)^2}$$

$$=\frac{1}{\alpha^2}f^{(1)}(\boldsymbol{Q}_{2^m-1}^{(1)})$$

通过以上分析，可将式(5.10)的子部分①的问题描述如下：

$$\text{opt}:\min_{\boldsymbol{Q}_{2^m-1}^{(1)}}\frac{1}{\alpha^2}f^{(1)}(\boldsymbol{Q}_{2^m-1}^{(1)}) \Leftrightarrow \text{opt}:\min_{\boldsymbol{Q}_{2^m-1}^{(1)}} f^{(1)}(\boldsymbol{Q}_{2^m-1}^{(1)})$$

$$\text{s.t.} \quad \boldsymbol{M}_{2^{m-1}-1}^{\mathrm{T}}(\boldsymbol{Q}_{2^m-1}^{(1)}) \leqslant \boldsymbol{I}_{(2^{m-1})\times 1}$$

$$\mu_i > 0$$

将$\boldsymbol{Q}_{2^m-1}^{(1)}$用$\boldsymbol{R}_{2^{m-1}-1}$代入，则问题等价于求解$\boldsymbol{R}_{2^{m-1}-1}^{*}$，即$\boldsymbol{\Sigma}_{2^{m-1}-1}^{*}$。由此可得$\boldsymbol{Q}_{2^m-1}^{*(1)}=\boldsymbol{R}_{2^{m-1}-1}^{*}=\frac{1}{\alpha}\boldsymbol{R}_{2^m-1}^{*(1)}$，即$\boldsymbol{R}_{2^m-1}^{*(1)}=\alpha\boldsymbol{R}_{2^{m-1}-1}^{*}$。

而式(5.10)的子部分③可看成是$\alpha=1$的特殊情况。因此，可以得出$\boldsymbol{R}_{2^m-1}^{*(2)}=\boldsymbol{R}_{2^{m-1}-1}^{*}$。

综上所述，式(5.11)成立。

定理5.4得证。

由定理5.4，可得形如式(5.10)的$\boldsymbol{\Sigma}_N^{*}$递推关系，从而可由$\boldsymbol{\Sigma}_{2^{m-1}-1}^{*}$的结果来求解关于$\boldsymbol{\Sigma}_{2^m-1}^{*}$的最优结果。

假设已经求得$N=2^{m-1}-1$下的最小均方误差$\text{err}_{m-1}=\min_{\boldsymbol{\Sigma}_{2^{m-1}-1}} f(\boldsymbol{\Sigma}_{2^{m-1}-1})$及$\boldsymbol{\Sigma}_{2^{m-1}-1}^{*}$。将上述问题转化为关于$\alpha$的最优化问题：

$$\text{opt}:\text{h}(\alpha)=\min_{\alpha}\left(\frac{\text{err}_{m-1}}{\alpha^2}+\frac{2^{m-1}}{(1-\alpha)^2}\right)+\text{err}_{m-1}$$