1 Boundary matrix

In order to calculate the Betti numbers in an algorithmic way, we must develop an formal method. The boundary matrix provides us such an approach.

Given a simplicial complex \( K \), we have its \( p \)th chain group, \( C_p \), which is the group of all possible formal sums of \( p \)-simplices in \( K \). More strictly speak, \( C_p = \mathbb{Z}^{z_p}_p \), where \( z_p \) is the number of \( p \)-simplices in \( K \) and \( \mathbb{Z} \) is the cyclic group \( \mathbb{Z}/2\mathbb{Z} \). With the \( p \)th boundary operator \( \partial_p : C_p \to C_{p-1} \) we defined previously, we have \( Z_p = \text{Ker}(\partial_p) \) and \( B_{p-1} = \text{Im}(\partial_p) \). Similarly to the abelian group representation of \( C_p \), we also have that \( Z_p = \mathbb{Z}^{z_p}_p \) and \( B_p = \mathbb{Z}^{b_p-1}_2 \), where \( z_p, b_p \) is called the rank of \( Z_p \) and \( B_p \) respectively. By the Rank-nullity theorem in linear algebra, we have

\[
C_p \cong Z_p \oplus B_{p-1} \text{ or } \mathbb{Z}^{z_p}_2 \cong \mathbb{Z}^{b_p-1}_2 \oplus \mathbb{Z}^{b_{p-1}-1}_2
\]

which indicates the important result

\[
n_p = z_p + b_{p-1}
\]

Since we also know that \( p \)th Betti number \( \beta_p = z_p - b_p \), we can calculate the \( \beta_p \) for each \( p \) as long as we can decompose \( n_p \) as \( \[1\] \).

Now let us give the way to decompose \( n_p \) with the boundary matrix. Let \( C_p = \{\alpha_1, \alpha_2, \ldots, \alpha_{n_p}\} \) and \( C_{p-1} = \{\tau_1, \tau_2, \ldots, \tau_{n_{p-1}}\} \). Define

\[
M_p = \begin{pmatrix}
\alpha_1 & \alpha_2 & \ldots & \alpha_{n_p} \\
\tau_1 & \alpha_1 & \ldots & \alpha_{n_p} \\
\tau_2 & \alpha_2 & \ldots & \alpha_{n_p} \\
\vdots & \vdots & \ddots & \vdots \\
\tau_{n_{p-1}} & \alpha_{n_{p-1}} & \ldots & \alpha_{n_{p-1}}
\end{pmatrix}
\]

where \( a_{ij} = 1 \) if and only if \( i \)th \( p-1 \) simplex \( \tau_i \) is a face of the \( j \)th \( p \) simplex \( \alpha_j \), otherwise, \( a_{ij} = 0 \). Then we have

\[
\partial_p \alpha_j = \sum_{i=1}^{n_{p-1}} a_{ij}^p \tau_i
\]

From an algebraic point of view, if we use \( \{\tau_1, \tau_2, \ldots, \tau_{n_{p-1}}\} \) as the basis of \( B_{p-1} \), the \( j \)th column of \( M_p \) is just the coordinate of \( \partial_p \alpha_j \) under that basis. Thus, the \( p \)th boundary matrix actually encodes all the possible relationship between \( p \)th simplices and their boundaries.

Given a \( p \)th chain \( c \) in \( K \), we have \( c = \sum_{j=1}^{n_p} c_j \alpha_j \) and its boundary

\[
\partial_p c = \partial_p \sum_{j=1}^{n_p} c_j \alpha_j = \sum_{j=1}^{n_p} c_j \partial_p \alpha_j = \sum_{j=1}^{n_p} c_j \sum_{i=1}^{n_{p-1}} a_{ij}^p \tau_i = \sum_{i=1}^{n_{p-1}} \left( \sum_{j=1}^{n_p} a_{ij}^p c_j \right) \tau_i
\]
Thus, under the basis of \( \{ \tau_1, \tau_2, \ldots, \tau_{n_{p-1}} \} \), the coordinate of \( \partial_p c \) is
\[
\begin{bmatrix}
a_1^1 & a_1^2 & \cdots & a_1^{n_p} \\
a_2^1 & a_2^2 & \cdots & a_2^{n_p} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n_{p-1}}^1 & a_{n_{p-1}}^2 & \cdots & a_{n_{p-1}}^{n_p}
\end{bmatrix}
\begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_{n_p}
\end{bmatrix}
\]
Since it is true for any \( p \) chain in \( K \), we have \( B_{p-1} = \text{Im}(\partial_p) = \text{SPAN}\{\text{col}_1, \text{col}_2, \ldots, \text{col}_{n_p}\} \), where \( \text{col}_j \) is the \( j \)th column of boundary matrix \( M_p \). With the knowledge of linear algebra, we have \( b_{p-1} = \text{Dim}(B_{p-1}) = \text{rank}(M_p) \). Now we can see that we have related \( b_{p-1} \) to rank of the boundary matrix, which can be calculated through the well-known linear algebra technique, Gaussian elimination.

Before we move forward, the following example will help to illustrate the concept we discuss so far.

**Example 1.1** Consider the 2-dimensional faces of the tetrahedron with four points \( a, b, c, d \) and the boundary operator \( \partial_2 \). Then we have

\[
M_2 = \begin{pmatrix}
abc & abd & acd & bcd \\
\begin{pmatrix}
ab \\
ac \\
ad \\
bc \\
bd \\
\end{pmatrix} & \begin{pmatrix} 1 & 1 \\
1 & 1 \\
1 & 1 \\
1 & 1 \\
\end{pmatrix}
\end{pmatrix}
\]

*For example, the 3 simplex abc has 3 faces, ab, ac, bc. Thus the column of abc has 1s on the corresponding entries. And the 2 chain, abc + abd, has faces, ac, ad, bc, bd, and the sum of the columns corresponding to abc and abd has 1s on its four faces.*

### 2 Smith Normal Form

As we discussed in the previous section, we can obtain the Betti numbers \( \beta_p \) for any \( p \) as long as we can find the rank for \( M_p \) for each \( p \). Now let us introduce an algorithmic method to decompose \( n_p \) as (1).

First of all, let us introduce two basic operations of columns on the boundary matrix, which don’t change the rank of the matrix.

1. Exchanging two column, say \( j_1 \)th and \( j_2 \)th column, let \( \hat{a}_{i}^{j_1} = a_{i}^{j_2} \) and \( \hat{a}_{i}^{j_2} = a_{i}^{j_1} \) for \( i = 1, 2, \ldots, n_{p-1} \), where \( \hat{a}_{i}^{j} \) is the entry of new matrix.

2. Adding one column to another, let \( \hat{a}_{i}^{j_1} = a_{i}^{j_1} + a_{i}^{j_2} \).

And similar operations can be define on the rows of boundary matrix. Moreover, all these operations can be view as multiplying \( M_p \) by one elementary matrix\(^2\). And we know that these operations on boundary matrix won’t change its rank. Using the two basic operations we mentioned before, we can reduce the
boundary matrix $M_p$ into the following form, or **Smith Normal Form** with Gaussian elimination.

$$S_p = \begin{bmatrix}
1 & 1 & \cdots & 0 \\
0 & 1 & 0 & \cdots \\
0 & 0 & \ddots & \ddots \\
0 & 0 & \cdots & 0
\end{bmatrix}$$

After reducing the boundary matrix into this form, we can easily find $\text{rank}(M_p) = b_{p-1}$ by finding the number of 1s on the diagonal of the $S_p$. Now let us describe the process of Gaussian elimination, which is basic in linear algebra.

Here, we just use the process introduced in [1]. Let $T = M_p$ initially. Using at most two exchange operations, we can move a 1 to the upper left position of $T$. And with at most $n_p - 1$ column and $n_{p-1} - 1$ row additions, we can cancel out all the 1s in the first row and first column. Then let $T$ be its submatrix obtained by removing the first row and first column and do the same thing recursively. Let $x$ be the row and column number of upper left element of the submatrix we consider. Initially, $x = 1$.

**Algorithm 1** Reduce(x)

```plaintext
if $\exists k \geq x, l \geq x$ with $a_{kl}^x = 1$ then
  exchange rows $x$ and $k$; exchange columns $x$ and $l$;
  for $i = x + 1$ to $n_{p-1}$ do
    if $a_{xi}^x = 1$ then
      add row $x$ to row $i$
    end if
  end for
  for $i = x + 1$ to $n_p$ do
    if $a_{ij}^x = 1$ then
      add column $x$ to column $j$
    end if
  end for
  Reduce($x + 1$)
end if
```

With this procedure, we have at most $n_{p-1}^2$ row operation and at most $n_p^2$ column operations and $O(n_{p-1}n_p(n_p + n_{p+1}))$ for total complexity.

### 3 Basis of Boundary group $B_{p-1}$ and cycle group $Z_p$

Besides the interest on Betti number, if we are also interested in the basis of the Boundary group $B_{p-1}$ and cycle group $Z_p$. Gaussian elimination will help us as long as we also record the change on the representative $p$-chains for the columns. In particular, we associate with the $j$-th column also a $p$-chain $\gamma_j$. Initially, $\gamma_j = \alpha_j$; note that at this point, $\text{col}_j = \partial \gamma_j$. We now reduce the matrix so that each remaining columns are linearly independent using only the following operations.

1. For exchange operation of columns, we also exchange the representatives of columns.

\[
\hat{\gamma}_{j_1} \leftarrow \gamma_{j_2}, \hat{\gamma}_{j_2} \leftarrow \gamma_{j_1}
\]
2. For addition of $j_1$ column to $j_2$ column

$$\tilde{\gamma}_{j_2} \leftarrow \gamma_{j_1} + \gamma_{j_2}$$

Let $\gamma^k_j$ and $\text{col}^k_j$ denote the $j$-th column and its associated $p$-chain after $k$ operations. Note that none of the two operations above change the following fact, which means that we maintain the following invariance during the reduction:

Invariance $\text{col}^k_j = \partial \gamma^k_j$.

In the end, after the initial boundary matrix $M_p$ is reduced to $R_p$ where all columns are linearly independent, it is easy to see that the remaining non-zero columns form a basis of $B_{p-1}$.

To obtain a basis for $Z_p$, collect the set of $p$-chains $\Pi = \{ \gamma_i \mid \text{col}_i \text{ is all zero} \}$. Obviously, each $p$-chain in $\Pi$ is a $p$-cycle, as its corresponding column (which is its boundary due to the invariance that we maintain) is empty. There are exactly $z_p$ number of cycles in $\Pi$. Furthermore, it is easy to verify that all these $p$-chains are independent. This is because at the beginning, all $\gamma^0_i$ are independent. The column operations we perform do not change this fact. Hence the set of $\gamma^k_i$ at any stage $k$ are always independent. It then follows that $p$-cycles in $\Pi$ form a basis for the $p$-th cycle group $Z_p$.

Now let us use these two operations to compute the basis of boundary for the example (1.1).

**Example 3.1** As we can see in the matrix in example (1.1), in order to cancel the 1s in first row, we need to add 1st column to 2nd column. After that, we got the following matrix

$$\begin{pmatrix}
ab & abc & abd & acd & bcd \\
ab & 1 & abc \\
ab & 1 & 1 & 1 \\
ab & 1 & 1 \\
ab & 1 & 1 & 1 \\
ab & 1 & 1 & 1 \\
ab & 1 & 1 & 1
\end{pmatrix}$$

We continue this way to eliminate, for each diagonal 1, all entries to its right in its corresponding column. In the end, we obtain the following reduced matrix $R_2$.

$$\begin{pmatrix}
ab & abc & abd & abd & abd \\
ab & 1 & abc & abc & abc \\
ab & 1 & 1 & 1 \\
ab & 1 & 1 \\
ab & 1 & 1 & 1 \\
ab & 1 & 1 & 1 \\
ab & 1 & 1 & 1
\end{pmatrix}$$

From this reduced matrix $R_2$, we can conclude that the columns $c_1 = abc, c_2 = abd + abc, c_3 = abc + abd + acd$ for the basis for $B_1$ for tetrahedron. The only column with all 0s is the last one, and the associated $p$-chain $c_4 = abc + abd + acd + bcd$ is the only non-trivial element in $Z_2$ of tetrahedron.

**References**
