# Markov degree of the Birkhoff model

Takashi Yamaguchi · Mitsunori Ogawa · Akimichi Takemura

Received: 22 April 2013 / Accepted: 14 November 2013 / Published online: 3 December 2013 © Springer Science+Business Media New York 2013

Abstract We prove the conjecture by Diaconis and Eriksson (J. Symbolic Comput. 41(2):182–195, 2006) that the Markov degree of the Birkhoff model is three. In fact, we prove the conjecture in a generalization of the Birkhoff model, where each voter is asked to rank a fixed number, say r, of candidates among all candidates.

Keywords Algebraic statistics  $\cdot$  Markov basis  $\cdot$  Normality of semigroup  $\cdot$  Ranking model

# **1** Preliminaries

Diaconis and Eriksson [4] conjectured that the Markov degree of the Birkhoff model is three, i.e., the toric ideal associated with the Birkhoff model is generated by binomials of degree at most three. In this paper we give a proof of this conjecture in a generalization of the Birkhoff model, where each voter is asked to rank a fixed number of most preferred candidates among all candidates. Our proof is based on arguments of Jacobson and Matthews [6] for Latin squares. The set of Latin squares is a particular fiber in our setting and our result is also a generalization of [6]. See [1] for terminology of algebraic statistics and toric ideals used in this paper.

Consider an election, where there are *n* candidates and *N* voters. Each voter is asked to give r  $(1 \le r \le n)$  preferred candidates and to rank them. For example,

Graduate School of Information Science and Technology, University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

T. Yamaguchi e-mail: takashi\_yamaguchi@mist.i.u-tokyo.ac.jp

A. Takemura e-mail: takemura@stat.t.u-tokyo.ac.jp

T. Yamaguchi · M. Ogawa (🖂) · A. Takemura

e-mail: mitsunori\_ogawa@mist.i.u-tokyo.ac.jp

let n = 5, r = 3 and let the candidates be a, b, c, d, e. A vote (a, c, d) by a voter means that he/she ranks a first, c second and d third. For a positive integer m, denote  $[m] = \{1, ..., m\}$ . When the candidates are labeled as 1, ..., n, the set of possible votes is

$$S_{n,r} = \left\{ \sigma = \left( \sigma(1), \dots, \sigma(r) \right) \mid \sigma : \text{ injection from } [r] \text{ to } [n] \right\}, \quad |S_{n,r}| = \frac{n!}{(n-r)!}$$

where  $\sigma(j)$  denotes the candidate chosen in the *j*th position in the vote  $\sigma = (\sigma(1), \ldots, \sigma(r))$ . Let  $\psi_{jk}$ ,  $j \in [r]$ ,  $k \in [n]$ , be positive parameters and define a probability distribution over  $S_{n,r}$  by

$$p(\sigma) = \frac{1}{Z} \prod_{j=1}^{r} \psi_{j\sigma(j)}, \quad Z = \sum_{\sigma \in S_{n,r}} \prod_{j=1}^{r} \psi_{j\sigma(j)}.$$
 (1)

If  $\psi_{jk}$  is large, then the candidate *k* is likely to be ranked in the *j*th position. When r = n, this model is the Birkhoff model [4, 8]. In this paper we call (1) an (n, r)-Birkhoff model. The sufficient statistic of the (n, r)-Birkhoff model consists of numbers of times the candidate *k* is ranked in the *j*th position,  $j \in [r], k \in [n]$ . We denote the sufficient statistic as  $(t_{jk})_{j \in [r], k \in [n]}$ .

Define a 0-1 matrix  $A = A_{n,r}$  of size  $rn \times (n!/(n-r)!)$ , called a configuration matrix for the (n, r)-Birkhoff model, whose columns are labeled by  $\sigma \in S_{n,r}$  and rows are labeled by (j, k) = (position, candidate), such that the  $((j, k), \sigma)$ -element of A is one if and only if  $\sigma(j) = k$ . For example, for n = 4, r = 3, the configuration matrix  $A_{4,3}$  with labels for its rows and columns is

	$\overline{\mathfrak{S}}$	<del>4</del> i	ନ	<del>4</del> (	ରି ଜ	$\widehat{\mathbf{n}}$	$\widehat{\mathbf{e}}$	<del>4</del> ;	<u>с</u> ;	<del>4</del> ;	Ę,	ଳ	ନ	<del>4</del> ;	Ę	<del>4</del> ;	Ę,	ରି	ลิ	$\widehat{\mathbf{e}}$	<u>-</u>	$\widehat{\mathbf{e}}$	<u>–</u> (	6		
	<u> </u>	(17	(13	<u>[]</u>	<u> </u>	<u>4</u>	$\overline{5}$	[5]	63	63	7	5	(31)	(31)	(32	32	<u>6</u>	( <u>3</u> 4	( <del>4</del>	<del>[</del> 4]	64	(42	(43	(43		
(1, 1)	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
(1, 2)	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0		
(1, 3)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0		
(1, 4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1		
(2, 1)	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0		
(2, 2)	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	(	(2)
(2, 3)	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1		
(2, 4)	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0		
(3, 1)	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0		
(3, 2)	0	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	1		
(3, 3)	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0		
(3, 4)	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0		

Let  $x(\sigma) \in \mathbb{N} = \{0, 1, ...\}$  be the frequency of voters choosing a vote  $\sigma \in S_{n,r}$  and let  $\mathbf{x} = \{x(\sigma) \mid \sigma \in S_{n,r}\}$  be the vector of frequencies. Then  $\mathbf{t} = A_{n,r}\mathbf{x}$  is the sufficient statistic vector. For a given  $\mathbf{t}, \mathcal{F}_{\mathbf{t}} = \{\mathbf{x} \in \mathbb{N}^{|S_{n,r}|} \mid A\mathbf{x} = \mathbf{t}\}$  is the  $\mathbf{t}$ -fiber.

Let *K* be any field and let  $K[\{p(\sigma), \sigma \in S_{n,r}\}]$  be the polynomial ring in the indeterminates  $p(\sigma), \sigma \in S_{n,r}$ . Similarly let  $K[\{\psi_{jk}, j \in [r], k \in [n]\}]$  be the polynomial

ring in the indeterminates  $\psi_{jk}$ ,  $j \in [r]$ ,  $k \in [n]$ . Let

$$\pi_{n,r}: K\big[\big\{p(\sigma) \mid \sigma \in S_{n,r}\big\}\big] \to K\big[\big\{\psi_{jk}, j \in [r], k \in [n]\big\}\big]$$

be a homomorphism defined by

$$\pi_{n,r}: p(\sigma) \mapsto \prod_{j=1}^r \psi_{j\sigma(j)}$$

Then the toric ideal  $I_A = I_{A_{n,r}}$  for the (n, r)-Birkhoff model is the kernel of  $\pi_{n,r}$ . Moves for  $A_{n,r}$  are the elements of the integer kernel ker $\mathbb{Z} A_{n,r} = \{z \in \mathbb{Z}^{|S_{n,r}|} | Az = 0\}$  of  $A_{n,r}$ .

Note that if a voter ranks r = n - 1 most preferred candidates, then he/she automatically ranks the last candidate. It is easy to see that the configuration matrix  $A_{n,n-1}$  for the (n, n - 1)-Birkhoff model and the configuration matrix  $A_{n,n}$  for the Birkhoff model have the same number of columns and their integer kernels are the same: ker<sub>Z</sub>  $A_{n,n-1} = \ker_Z A_{n,n}$ .

### 2 Main result and its proof

The main result of this paper is the following theorem.

**Theorem 1** For  $r \ge 2$  and  $n \ge 3$ , the toric ideal  $I_A$  for the (n, r)-Birkhoff model is generated by binomials of degree two and three.

For r = 1 or r = n = 2, the toric ideal  $I_A$  is trivial. For  $r \ge 2$  and  $n \ge 3$ , any set of generators for  $I_A$  contains a binomial of degree three. In the terminology of algebraic statistics, Theorem 1 states that the Markov degree of the (n, r)-Birkhoff model is three for  $r \ge 2$  and  $n \ge 3$ .

The rest of this section is devoted to a proof of this theorem. We define some notation and terminology for our proof, mainly following [6]. Candidates are denoted either by letters a, b, c, ... or by numbers 1, ..., n. The set of n candidates is denoted by [n], using numbers.

First we give the definition for "valid" votes and definitions for two kinds of "invalid votes." Our proof will be based on the idea of swapping candidates between two votes.

**Definition 1** An  $r \times n$  integer matrix  $V = (v_{jk})$  is a *proper vote* if  $v_{jk} \in \{0, 1\}, \forall j, k$ , every row sum is one, and every column sum is zero or one. A *proper data set* of N votes is the multiset of N proper votes.

**Definition 2** An  $r \times n$  integer matrix  $V = (v_{jk})$  is an *improper vote* if every row sum is one, every column sum is zero or one, and there exists a unique cell  $(j^*, k^*) \in [r] \times [n]$  such that

$$v_{j^*k^*} = -1, \quad v_{jk} \in \{0, 1\}, \quad \forall (j, k) \neq (j^*, k^*).$$

An *improper data set* of N votes is the multiset of  $r \times n$  integer matrices  $V^{(1)} = (v_{jk}^{(1)}), \ldots, V^{(N)} = (v_{jk}^{(N)})$  such that one of them is an improper vote, the others are proper votes, and  $\sum_{i=1}^{N} v_{jk}^{(i)} \ge 0, \forall j, k$ .

**Definition 3** An  $r \times n$  integer matrix  $V = (v_{jk})$  is a vote with collision if  $v_{jk} \in \{0, 1\}, \forall j, k$ , every row sum is one and there exists a unique candidate  $k^* \in [n]$  such that

$$\sum_{j=1}^{r} v_{jk^*} = 2, \qquad \sum_{j=1}^{r} v_{jk} \in \{0, 1\}, \quad \forall k \neq k^*.$$

In this case we also say that the vote V contains a *collision* or the candidate  $k^*$  *collides* in V.

**Definition 4** An  $r \times n$  integer matrix  $V = (v_{jk})$  is an *improper vote with collision* if every row sum is one, there exists a unique cell  $(j^*, k^*) \in [r] \times [n]$  such that

$$v_{j^*k^*} = -1, \quad v_{jk} \in \{0, 1\}, \quad \forall (j, k) \neq (j^*, k^*),$$

and there exists a unique candidate  $k^{**} \in [n]$  such that

$$\sum_{j=1}^{r} v_{jk^{**}} = 2, \qquad \sum_{j=1}^{r} v_{jk} \in \{0, 1\}, \quad \forall k \neq k^{**}.$$

We call a multiset D of  $r \times n$  integer matrices a *data set* if each matrix in D is one of the votes defined in Definitions 1–4.

As in Sect. 1, we often denote votes by row vectors. For proper votes and votes with collision, we denote them by *r*-dimensional row vectors whose *j*th entry is the candidate ranked in the *j*th position for each  $j \in [r]$ . For improper votes we define their row vector representation as follows. Let  $V = (v_{jk})$  be an improper vote with  $v_{j^*k} = v_{j^*k'} = 1$ ,  $v_{j^*k''} = -1$ . We denote *V* by an *r*-dimensional row vector whose *j*th entry is the candidate ranked in the *j*th position for each  $j \in [r]$  with  $j \neq j^*$  and the *j*\*th entry is k + k' - k''. Here, k + k' - k'' is just a symbol and we call it an *improper element*. The following vectors are examples of a proper vote, a vote with collision and an improper vote, respectively:

$$(b, a, c, d),$$
  $(b, a, c, a),$   $(a, d, a, b + c - a)$ 

We also define the row vector representation for improper votes with collision in the similar manner.

Several kinds of data sets were defined as the multiset of integer matrices above. For these data sets we use their matrix representation. The matrix representation  $\overline{D}$  for a data set  $D = \{V^{(1)}, \ldots, V^{(N)}\}$  is an  $N \times r$  matrix whose *i*th row is the row vector representation of vote  $V^{(i)}$  for each  $i \in [N]$ . Although the order of the rows of  $\overline{D}$  is arbitrary, this matrix representation is convenient for our proof. When there is no confusion,  $\overline{D}$  is also called a data set. Latin squares are also of this form. They are tables with N = n = r such that each candidate appears exactly once in each row and column. An example of improper data set I and its matrix representation  $\overline{I}$  is as follows:

$$I = \left\{ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\},\$$
$$\bar{I} = \begin{bmatrix} c & c & a+b-c \\ b & a & c \\ a & b & c \end{bmatrix}.$$

In the following, when we display a data set, we mainly use its matrix representation.

We now introduce some operations for data sets. Let  $D = \{V^{(1)}, \dots, V^{(N)}\}$  be a data set. Consider a pair of distinct votes in D, say  $V^{(i_1)} = (v_{jk}^{(i_1)})$  and  $V^{(i_2)} = (v_{jk}^{(i_2)})$ . A *swap*  $\{i_1, i_2\} : k_1 \stackrel{j^*}{\leftrightarrow} k_2$  for D is an operation transforming D into D', where

$$D' = \left(D \setminus \left\{V^{(i_1)}, V^{(i_2)}\right\}\right) \cup \left\{\tilde{V}^{(i_1)}, \tilde{V}^{(i_2)}\right\}, \quad \tilde{V}^{(i_1)} = \left(\tilde{v}_{jk}^{(i_1)}\right), \tilde{V}^{(i_2)} = \left(\tilde{v}_{jk}^{(i_2)}\right),$$
$$\tilde{v}_{jk}^{(i)} = \begin{cases} v_{jk}^{(i)} + 1, & (i, j, k) = (i_1, j^*, k_2), (i_2, j^*, k_1), \\ v_{jk}^{(i)} - 1, & (i, j, k) = (i_1, j^*, k_1), (i_2, j^*, k_2), \\ v_{jk}^{(i)}, & \text{otherwise.} \end{cases}$$

In general, D' is not a data set, because  $\tilde{V}^{(i)}$ ,  $i = i_1, i_2$ , may not be a vote defined above. If D' is also a data set, the swap is called an *applicable swap*. Since only applicable swaps appear in our proof, applicable swaps are called merely swaps, hereafter. The swap operation does not alter the sufficient statistic of the data set. Note that the swap may cause a new collision or a new improper element in  $\tilde{V}^{(i)}$ ,  $i = i_1, i_2$ .

The matrix representation of data sets helps intuitive understanding and manipulation of the swap operation. The definition of the swap operation shows that the  $j^*$ th row of  $\tilde{V}^{(i_1)}$  is the sum of the  $j^*$ th row of  $V^{(i_1)}$  and a row vector with 0 entries except -1 for the  $k_1$ th entry and 1 for the  $k_2$ th entry. Similarly, the  $j^*$ th row of  $\tilde{V}^{(i_2)}$  is the sum of the  $j^*$ th row of  $V^{(i_2)}$  and a row vector with 0 entries except 1 for the  $k_1$ th entry and -1 for the  $k_2$ th entry. Hence, the matrix representation  $\bar{D}'$  of D' is the sum of the matrix representation  $\bar{D}$  of D and the matrix with 0 entries except  $k_2 - k_1$  for the  $(i_1, j^*)$ -entry and  $k_1 - k_2$  for the  $(i_2, j^*)$ -entry where  $k_2 - k_1$  and  $k_1 - k_2$  are symbols.

For illustration, we show an example of swap in the matrix representation of data sets. Let  $\overline{P} = (p_{ij})$  be a proper data set in its matrix representation with  $p_{11} = a, p_{21} = b, a \neq b$  and consider a swap  $\{1, 2\} : a \stackrel{1}{\leftrightarrow} b$ . By adding

$$\begin{bmatrix} b-a\\a-b \end{bmatrix}$$

to the submatrix  $\begin{bmatrix} a \\ b \end{bmatrix}$  we interchange two candidates a and b as

$\begin{bmatrix} a & * & \cdots & * \end{bmatrix} \begin{bmatrix} b - a & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} b \\ b & \cdots & 0 \end{bmatrix}$	* ··· *]
$\begin{vmatrix} b & * & \cdots & * \end{vmatrix}  \begin{vmatrix} a - b & 0 & \cdots & 0 \end{vmatrix}  \begin{vmatrix} a \end{vmatrix}$	* ··· *
$  * * \cdots *   +   0 0 \cdots 0   =   *$	* ··· * . (3)
	: : :

where candidates denoted by \* are not changed. In this case *a* and *b* may collide after the swap. In order to simplify the notation, we sometimes denote the swap by  $a \stackrel{j}{\leftrightarrow} b$ or  $a \leftrightarrow b$ .

Let us discuss a sequence of swaps. Consider swapping a and b in two different positions j, j' in the same *i*th and *i'*th votes. In our proof below, we often perform these two swaps sequentially, i.e., we swap a and b in the *j*th position first and then in the *j'*th position. We denote this operation by

$$a \stackrel{j}{\leftrightarrow} b \stackrel{j'}{\leftrightarrow} a \quad \text{or} \quad \{i, i'\} : a \stackrel{j}{\leftrightarrow} b \stackrel{j'}{\leftrightarrow} a$$

and call this a *double swap*. The double swap corresponds to the basic move for no three-factor interaction model (cf. [2]). As an example, a double swap  $a \stackrel{1}{\leftrightarrow} b \stackrel{2}{\leftrightarrow} a$ , where the second swap causes an improper element, is written as

$$\begin{bmatrix} a & b \\ b & c \end{bmatrix} + \begin{bmatrix} b-a & a-b \\ a-b & b-a \end{bmatrix} = \begin{bmatrix} b & a \\ a & b+c-a \end{bmatrix}.$$
 (4)

More generally, we consider a sequence of m swaps in positions  $j_1, \ldots, j_m$ , such that two consecutive swaps involve a common candidate, and denote it as

$$a_1 \stackrel{j_1}{\leftrightarrow} a_2 \stackrel{j_2}{\leftrightarrow} \cdots \stackrel{j_{m-1}}{\leftrightarrow} a_m \stackrel{j_m}{\leftrightarrow} a_{m+1} \tag{5}$$

or indicating the votes as

$$\{i, i'\}: a_1 \stackrel{j_1}{\leftrightarrow} a_2 \stackrel{j_2}{\leftrightarrow} \cdots \stackrel{j_{m-1}}{\leftrightarrow} a_m \stackrel{j_m}{\leftrightarrow} a_{m+1}.$$

$$\tag{6}$$

We call (5) (or (6)) a *chain swap* of length *m* (even when  $a_1 = a_{m+1}$ , i.e., we do not make a distinction between a chain and a loop). A chain swap of length one is just a swap.

Suppose that we perform several chain swaps for the same two votes and ignore the order of swaps. An even number of swaps on two proper elements at the same position results in no swap and an odd number swaps on two proper elements at the same position results in a single swap.

On the other hand, we need to be careful for swaps involving an improper element. Let b + c - a be an improper element in the *j*th position in an improper data set  $\overline{I}$ . Since the elements of the sufficient statistic of *I* are assumed to be nonnegative, there is a vote of  $\overline{I}$  containing a in the same position as b + c - a. If we make a swap  $a \leftrightarrow b$  between these two elements, then b + c - a becomes c and a becomes b:

$$\begin{bmatrix} b+c-a\\a \end{bmatrix} + \begin{bmatrix} a-b\\b-a \end{bmatrix} = \begin{bmatrix} c\\b \end{bmatrix}.$$
(7)

Similarly  $a \leftrightarrow c$  results in  $\begin{bmatrix} b \\ c \end{bmatrix}$ . Note that  $\begin{bmatrix} c \\ b \end{bmatrix}$  and  $\begin{bmatrix} b \\ c \end{bmatrix}$  are swaps of each other. Hence the result of several swaps can be regarded as a single swap  $a \leftrightarrow b$  or  $a \leftrightarrow c$ . Although there is an ambiguity between  $a \leftrightarrow b$  or  $a \leftrightarrow c$ , the result of a swap between these votes at the *j*th position is either  $\begin{bmatrix} b \\ c \end{bmatrix}$  or  $\begin{bmatrix} c \\ b \end{bmatrix}$ .

Furthermore we consider a swap between two candidates in  $\binom{b+c-a}{d}$ ,  $d \neq a, b, c$ . We allow  $b \leftrightarrow d$  or  $c \leftrightarrow d$  between these two elements. After  $b \leftrightarrow d$  we have

$$\begin{bmatrix} b+c-a\\d \end{bmatrix} \to \begin{bmatrix} d+c-a\\b \end{bmatrix}$$
(8)

and to  $\begin{bmatrix} d+c-a \\ b \end{bmatrix}$  we can make further swaps  $c \leftrightarrow b$  or  $d \leftrightarrow b$ . The end result of several swaps is one of the following three cases:

$$\begin{bmatrix} c+d-a\\ b \end{bmatrix}$$
,  $\begin{bmatrix} b+d-a\\ c \end{bmatrix}$  or  $\begin{bmatrix} b+c-a\\ d \end{bmatrix}$ .

These three cases correspond to single swaps  $b \leftrightarrow d$ ,  $c \leftrightarrow d$  and to no swap to  $\begin{bmatrix} b+c-a \\ d \end{bmatrix}$ .

Although there is an ambiguity on the result of chain swaps involving an improper element, the end result of several chain swaps is a set of simultaneous swaps of a subset of positions among the two votes. We call this a swap operation for a subset of positions among two votes, or simply a *swap operation among two votes*. When we apply a swap operation to a proper or an improper data set *D* for a subset *J* of positions among two votes  $R = \{i, i'\}$  and the result is *D'*, we denote the operation by a long double-sided arrow:

$$D \stackrel{R}{\longleftrightarrow} D',$$

where we omit J, because it is often cumbersome to specify J. In this notation we denote a proper data set by P and an improper data set by I, when we want to clarify the kinds of data sets, instead of D.

We now give a proof of Theorem 1 in a series of lemmas. Let P and P' be two proper data sets with the same sufficient statistic. Our strategy for a proof is to perform swap operations to P, involving at most three votes of P at each step, to increase the number of the common elements in  $\overline{P}$  and  $\overline{P'}$ . In each operation, elements at the same position of the three votes of  $\overline{P}$  are permuted. This corresponds to a move of degree at most three. In fact, each operation will be further decomposed into a series of swap operations among two votes, which involve intermediate improper data sets.

For the *i*th vote of  $\overline{P}$  and the *i*'th vote of  $\overline{P}'$ ,

$$(p_{i1},\ldots,p_{ir}), (p'_{i'1},\ldots,p'_{i'r}),$$

let

$$C = C_{i,i'} = \left| \left\{ j \mid p_{ij} = p'_{i'j} \right\} \right|$$
(9)

be the number of the same candidates in the same positions in these two votes. We call *C* the number of *concurrences*. If C = r, then we can remove these two votes from  $\overline{P}$  and  $\overline{P'}$  and consider other N - 1 votes. On the other hand, we will show that if C < r, then we can always increase *C* by a series of swap operations involving at most three votes of  $\overline{P}$ . The *i*th vote of  $\overline{P}$  will eventually coincide with the *i*'th vote of  $\overline{P'}$ . Then, Theorem 1 is proved by induction on *N*.

Our first lemma concerns resolving collisions.

**Lemma 1** Let  $\overline{D}$  be a data set without any improper element and suppose that at least one of the *i*th and *i'*th votes contains a collision. If each candidate appears at most twice in these two votes in total, we can resolve all the collisions by a swap operation among these two votes.

*Remark 1* We can prove this lemma based on the normality of the semigroup generated by the configuration matrix  $A_{n,r}$  such as  $A_{4,3}$  in (2). The normality follows from results in [7, 9] and [3]. We will discuss this point again in Sect. 3.2. However, we give our own proof of Lemma 1, because we will use similar arguments for improper data sets. Arguments based on the normality cannot be applied to improper data sets.

*Proof* We may assume i = 1 and i' = 2 and at least the first vote contains a collision. We first consider the case that there is only one collision in the two votes. Let *a* denote the colliding candidate. Relabeling the positions, without loss of generality, the two votes are displayed as

$$\begin{bmatrix} a & a & d & * & \cdots & * \\ b & c & * & * & \cdots & * \end{bmatrix},$$

where  $b \neq c$ . We choose one of the two *a*'s arbitrarily, say in the second position, and make a swap  $a \stackrel{2}{\leftrightarrow} c$  with the following result:

$$\begin{bmatrix} a & c & d & * & \dots & * \\ b & a & * & * & \cdots & * \end{bmatrix}.$$
 (10)

Since *a* appears at most twice in these two votes in total, *a* does not collide in the second vote. However, *c* might again collide in the first vote, e.g.,

$$\begin{bmatrix} a & c & d & c & * & \dots & * \end{bmatrix}$$

We then make a swap for c, which was in the first vote from the beginning (in this example  $c \stackrel{4}{\leftrightarrow} *$ ). If we continue this process, we always have collisions in the first vote. If this process ends in finite number of steps, then by a chain swap we resolve the collisions of a and subsequent collisions due to swaps. We claim that this process indeed ends in finite number of steps. Actually we show a stronger result that no candidate appears twice in this process of resolving collisions.

Suppose otherwise. Then there is a candidate, say  $\alpha$ , which is swapped twice for the first time. We consider two cases:  $\alpha = a$  and  $\alpha \neq a$ .

Consider the case  $\alpha = a$ . The process of swaps is displayed as follows:

$$a \leftrightarrow c \leftrightarrow s_1 \leftrightarrow \cdots \leftrightarrow s_{l-1} \leftrightarrow a \leftrightarrow \cdots$$

Since the collision always occurs in the first vote, the candidate *a* was moved from the second vote to the first vote in the swap  $s_{l-1} \leftrightarrow a$ . By (10) we have  $c = s_{l-1}$ , which contradicts the assumption that  $\alpha = a$  is the first candidate colliding twice.

Consider the case  $\alpha \neq a$ . The process of swaps is displayed as follows:

$$a \leftrightarrow c \leftrightarrow s_1 \leftrightarrow \dots \leftrightarrow s_{l-1} \leftrightarrow \alpha \leftrightarrow s_{l+1} \leftrightarrow \dots \leftrightarrow s_{m-1} \leftrightarrow \alpha \leftrightarrow \dots$$
(11)

Considering the subprocess of (11) which starts from the first  $\alpha$ , we can apply the discussion for the  $\alpha = a$  and confirm that there exists a contradiction. We have shown the lemma for the case that there is only one collision.

Now suppose that there are *m* colliding candidates  $a_1, a_2, \ldots, a_m$ . Each of these candidates appears in one of the votes twice. Temporarily, we assign different labels, say  $a'_l, a''_l, l = 2, \ldots, m$ , to candidates except for  $a_1$ , namely, we ignore collisions of  $a_2, \ldots, a_m$ . By the above procedure we resolve the collision of  $a_1$  and subsequent collisions. When this procedure is finished, we restore the labels  $a'_l, a''_l \rightarrow a_l, l = 2, \ldots, m$ . Then some collisions of  $a_2, \ldots, a_m$  may have been already resolved, but we do not have any new collision. Hence, by the above procedure we decrease the number of collisions. As long as there is a remaining collision, we can repeat this procedure and resolve all the collisions.

So far we discussed resolving collisions. We now consider resolving an improper element by a swap operation among two votes.

**Lemma 2** Let *I* be an improper data set containing an element b + c - a. By a swap operation among two votes,  $\overline{I}$  can be transformed to a proper data set.

*Proof* Without loss of generality, assume that the first vote contains b + c - a and the second vote contains a. We can then make a swap  $\{1, 2\} : a \leftrightarrow b$ , as in (7). Here a may collide in the first vote and b may collide in the second vote. However, both a and b appear at most twice in these two votes. Hence we can now resolve these possible collisions by Lemma 1 by a swap operation among these two votes.  $\Box$ 

The operation of Lemma 2 is denoted by

$$I \stackrel{R}{\longleftrightarrow} P, \tag{12}$$

where R is a set of two votes of I.

At this point we make the following two definitions.

**Definition 5** We call two votes in Lemma 2 of the form

 $i_{\text{im}} \begin{bmatrix} * \cdots & * & b+c-a & * \cdots & * \\ * & \cdots & * & a & * & \cdots & * \end{bmatrix}$ 

a *resolvable pair*. Here  $i_{im}$  is an improper vote and  $i_{pr}$  is a proper vote. A resolvable pair is denoted as  $[i_{im}, i_{pr}]$ .

Note that any improper data set  $\overline{I}$  contains a resolvable pair  $[i_{im}, i_{pr}]$  and R in (12) is the set of votes of a resolvable pair.

**Definition 6** A swap operation among two votes  $R = \{i, i'\}$  in  $I \xleftarrow{R} I'$  is *compatible* with improper data sets  $\overline{I}$  and  $\overline{I'}$  if there exists a common resolvable pair  $[i_{im}, i_{pr}]$  of  $\overline{I}$  and  $\overline{I'}$  such that  $R \cap \{i_{im}, i_{pr}\} \neq \emptyset$ , or equivalently  $|R \cup \{i_{im}, i_{pr}\}| \leq 3$ .

**Lemma 3** Let P, P' be two proper data sets with the same sufficient statistic. Suppose that the *i*th vote of P and the *i*'th vote of P' are different, i.e.,  $V^{(i)} \neq V'^{(i')}$ , and let C < r in (9) be the number of concurrences in these two votes. Then, C can be increased by at most three steps of swap operations among two votes of P, where (1) each intermediate swap operation between two consecutive improper data sets is compatible with them, and (2) if the resulting data set is improper then its improper vote and the *i*th vote form a resolvable pair.

*Proof* Without loss of generality we consider the first votes of P and P'. We consider two disjoint cases.

Case 1 The same candidate appears in distinct positions in the two votes.

Let *b* be the candidate appearing in the distinct positions in the two votes. Relabeling the positions, without loss of generality, let  $p_{11} = a$ ,  $p'_{11} = b$ ,  $a \neq b$ , and  $p_{12} = b$ . Since *P'* contains *b* in the first position and the sufficient statistic of *P* and *P'* are the same, *P* has to contain *b* in the first position, say  $p_{21} = b$ . We now perform a double swap  $a \stackrel{1}{\Leftrightarrow} b \stackrel{2}{\Leftrightarrow} a$  to *P*:

Γ	а	b	*	•••	*		$\lceil b \rceil$	а	*	•••	*	
	b	С	*	•••	*		a	b + c - a	*	•••	*	
	*	*	*	•••	*	$\rightarrow$	*	*	*	• • •	*	Ι,
	÷	÷	÷	÷	÷		:	÷	÷	÷	÷	Í
L	*	*	*	•••	*_		_*	*	*		*_	

where \*'s are not changed. By this double swap *C* is increased. If c = a, the swap results in a proper data set. Otherwise, the swap results in an improper data set, where [2, 1] forms a resolvable pair. Therefore, *C* is increased by a process of the form  $P \stackrel{\{1,2\}}{\longleftrightarrow} P$  or  $P \stackrel{\{1,2\}}{\longleftrightarrow} I$ .

Case 2 Every candidate appearing twice in the two votes appears in the same position.

Again, let  $p_{11} = a$ ,  $p'_{11} = b$ ,  $a \neq b$ . The candidate *b* does not appear in the first vote of *P* and *a* does not appear in the first vote of *P'*. As in Case 1, we can assume  $p_{21} = b$ . Since the first vote of *P'* does not contain *a*, the total frequency of the candidate *a* in *P'* is less than *N*. Since the sufficient statistic is common, it follows that there is a vote of *P* which does not contain *a*.

If the second vote does not contain *a*, we can make a swap  $a \stackrel{1}{\leftrightarrow} b$  among the first two votes and increase *C* without causing collision. This process is of the form  $P \stackrel{\{1,2\}}{\longleftrightarrow} P$ .

If the second vote contains *a*, without loss of generality, let  $p_{22} = a$  and also assume that the third vote of *P* does not contain *a*. Let  $p_{32} = c \neq a$ . Since *a* is chosen in the second vote and not chosen in the third vote and both votes have the same number *r* of candidates, there is a candidate *d*, who is chosen in the third vote but is not chosen in the second vote. If *d* is in the position j > 2, then by relabeling of positions we assume that  $p_{33} = d$ . Then  $\overline{P}$  looks like

$$\begin{bmatrix} a & * & * & \cdots & * \\ b & a & * & \cdots & * \\ d & c & * & \cdots & * \\ * & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \cdots & * \end{bmatrix}$$
 or 
$$\begin{bmatrix} a & * & * & \cdots & * \\ b & a & * & \cdots & * \\ * & d(=c) & * & \cdots & * \\ * & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \cdots & * \end{bmatrix}$$
 or 
$$\begin{bmatrix} a & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \cdots & * \\ * & * & * & \cdots & * \end{bmatrix}$$

We perform a swap  $\{2, 3\}$ :  $a \stackrel{2}{\leftrightarrow} d$  to the second position of the second and third votes:

_				_						_	
a	*	*	•••	*		a	*	*	•••	*	
b	d	*	•••	*		b	d	*	•••	*	
d	a + c - d	*		*		*	а	*	•••	*	
*	*	*	•••	*	or	*	*	*	•••	*	or
:	:	÷		:		:	÷	÷		:	
-*	*	*		*		L*	*	*	•••	*	
Гa	*	*		*]							
b	d	*		*							
*	a + c - d	d		*							
*	*	*	•••	*	•						
:	:	÷		:							
L*	*	*	•••	*							

After the swap the second vote does not contain *a*. The result is proper if c = d (the middle case) and improper if  $c \neq d$ .

Now we apply a swap  $\{1, 2\} : a \stackrel{1}{\leftrightarrow} b$  for the first position of the first and the second votes and increase V. In the case  $c \neq d$ , the last swap was performed on

an improper data set, but it is compatible with the data sets. Furthermore we can resolve the improper element a + c - d by Lemma 2, since [3, 2] is a resolvable pair. The process in this case is summarized as  $P \stackrel{\{2,3\}}{\longleftrightarrow} P \stackrel{\{1,2\}}{\longleftrightarrow} P$  or  $P \stackrel{\{2,3\}}{\longleftrightarrow} P$ 

This proves the lemma.

**Lemma 4** Let I be an improper data set and P' be a proper data set with the same sufficient statistic. Consider the i'th vote from P' and choose any resolvable pair  $[i_{im}, i_{pr}]$  of I. Then, by at most three swap operations among two votes of I, we can (1) increase the number of concurrences  $C_{i_{pr},i'}$ , or (2) make I proper without changing the  $i_{pr}$ th vote of I. Furthermore, if the resulting data set is improper, then its improper vote and the  $i_{pr}$ th vote form a resolvable pair, and each intermediate swap operation between two consecutive improper data sets, is compatible with them.

To prove Lemma 4, we need the following two lemmas. For an improper data set *I*, we denote  $\overline{I} = \{t_{ij}\}$ .

**Lemma 5** Let I be an improper data set with  $\iota_{i_{\min}j} = b + c - a$ . Suppose that  $\iota_{ij'} = a$  where  $i \neq i_{i_{\min}}, j' \neq j$ . Then, letting  $\iota_{ij} = d \neq a$ , I can be transformed by a swap operation among two votes  $R = \{i_{i_{\min}}, i\}$  to another improper data set I' containing the improper  $i_{i_{\min}}$  th vote where  $\iota'_{i_{\min}j'} = a$  and  $\iota'_{i_{\min}j}$  is either of b + c - a, b + d - a or of c + d - a.

*Proof* Without loss of generality, assume that  $i_{im} = j = 1$  and i = j' = 2. Let  $\iota_{12} = e \neq b, c$ . We first make a swap of  $\{1, 2\} : a \stackrel{2}{\leftrightarrow} e$  to I:

b+c-a	е	*		*		b+c-a	а	*		*	
d	a	*	•••	*		d	е	*	•••	*	
*	*	*	• • •	*	$\rightarrow$	*	*	*	•••	*	
•	÷	÷	÷	÷		÷	÷	÷	:	÷	
*	*	*		*		*	*	*		*	

After the swap, a may collide in the first vote and e may collide in the second vote. If there is no collision, the claim of this lemma is proved. Otherwise, we can resolve these possible collisions in the following way.

We try to resolve the collision of e in the second vote as in Lemma 1 considering a swap process:

$$\{1,2\}: e \leftrightarrow s_1 \leftrightarrow s_2 \leftrightarrow \cdots . \tag{13}$$

In this process the collisions always occur in the second vote.

Consider the case that *d* is equal to *b* or *c*, say d = b. Since the first and the second votes contain *b* only in the first position, *b* does not collide in (13), which implies that *c* does not collide in (13). Since no *a* is in the second vote at the beginning of (13), *a* does not collide in (13). Therefore, there is no swap involving the first position in (13), which implies that the collision of *e* can be resolved as in Lemma 1.

Consider the case  $d \neq b, c$ . The difference of this case from Lemma 1 is that the process (13) may hit the first position. This happens when *d* appears in (13) for the first time as  $s_l \stackrel{j}{\leftrightarrow} d$ ,  $j \neq 1$ , and  $d = \iota_{1j}$  in the first vote is swapped down to the second vote in the *j*th position. Then we need to choose *b* or *c* and make the swap  $d \leftrightarrow b$  or  $d \leftrightarrow c$  in the first position. By symmetry, without loss of generality, we perform  $d \stackrel{1}{\leftrightarrow} c$ :

$$\begin{bmatrix} b+c-a \\ d \end{bmatrix} \rightarrow \begin{bmatrix} b+d-a \\ c \end{bmatrix}.$$

This amounts to ignoring b and -a and we look at the improper element b + c - a just as a proper element c in resolving the collision of e. We leave b - a in the (1, 1)element of  $\overline{I}$  as it is during the sequence in (13). Then just as in Lemma 1 it follows that no candidate appears twice in (13). Note that b and -a which were left in the (1, 1)-element cause no trouble, because collision occurs always in the second vote. Indeed, b causes no trouble because it does not leave the first vote. The candidate a causes no trouble because the second vote does not initially contain a and when a is swapped from the first vote to the second vote, then the process in (13) ends at that point.

After the collision of *e* is resolved, *a* may still collide in the first vote. Let  $j_1$  and  $j_2$ ,  $j_1 \neq j_2$ , be the labels of positions containing *a* in the first vote other than the first position. To resolve this collision we consider the following two swap processes:

$$\{1,2\}: a \stackrel{j_1}{\leftrightarrow} s_1 \leftrightarrow s_2 \leftrightarrow \cdots, \tag{14}$$

$$\{1,2\}: a \stackrel{j_2}{\leftrightarrow} s'_1 \leftrightarrow s'_2 \leftrightarrow \cdots,$$
(15)

where no swap in the  $j_2$ th position is involved in (14) and no swap in the  $j_1$ th position is involved in (15). Since every candidate in the first and second votes except a appears in at most two positions, the common candidate involved both in (14) and in (15) is a only. Then one of (14) and (15), say (14), involves neither b nor c, or involves b and no c. Therefore, ignoring c, -a and a in the  $j_2$ th column, we see that the swap process (14) ends in finite number of steps as in Lemma 1.

**Lemma 6** Let I be an improper data set with an improper element  $\iota_{i_{\min}j} = b + c - a$ . Let  $\iota_{ij} = d, i \neq i_{im}$ , and suppose that  $d \neq a, b, c$ . Then I can be transformed to another improper data set I' by a swap operation among two votes  $R = \{i_{im}, i\}$  such that either  $\iota'_{i_{\min}j} = b + d - a, \iota'_{ij} = c$  or  $\iota'_{i_{\min}j} = c + d - a, \iota'_{ij} = b$ .

*Proof* Without loss of generality assume  $i_{im} = j = 1$  and i = 2. Then the upper-left  $2 \times 1$  submatrix of I is  $\begin{bmatrix} b+c-a \\ d \end{bmatrix}$ . Note that [1, 2] is not a resolvable pair because  $d \neq a$ .

We begin by considering two swaps of  $\{1, 2\} : d \stackrel{1}{\leftrightarrow} b$  and  $\{1, 2\} : d \stackrel{1}{\leftrightarrow} c$ . If  $\{1, 2\} : d \stackrel{1}{\leftrightarrow} c$ . If  $\{1, 2\} : d \stackrel{1}{\leftrightarrow} b$  is applied to *I*, *b* may collide in the second vote and *d* may collide in the first vote. If  $\{1, 2\} : d \stackrel{1}{\leftrightarrow} c$  is applied to *I*, *c* may collide in the second vote and *d* 

may collide in the first vote. Considering the resolution of possible collisions in the second vote for each swap, the following two swap processes are obtained:

$$\{1,2\}: d \stackrel{1}{\leftrightarrow} b \leftrightarrow s_1 \leftrightarrow s_2 \leftrightarrow \cdots,$$
(16)

$$\{1,2\}: d \stackrel{1}{\leftrightarrow} c \leftrightarrow s'_1 \leftrightarrow s'_2 \leftrightarrow \cdots . \tag{17}$$

Since the number of positions which contains a in the first or second vote is at most three, one of (16) and (17), say (16), contains a at most once. Note that each candidates other than a appears in the first and second votes at most twice. If (16) does not contain a, we see that (16) ends in finite number of steps as in Lemma 1. If (16) contains one a, the finiteness of (16) is proved by applying the similar discussion of Lemma 1 for the subprocess of (16) which starts from a.

After resolving the collision of b in the second vote, d may still collide in the first vote. At this point the second vote contains at most one a. Consider a swap process

$$\{1,2\}: d \leftrightarrow s_1'' \leftrightarrow s_2'' \leftrightarrow \cdots .$$
<sup>(18)</sup>

Since *b* has already been involved in (16), no  $s_i''$  is equal to *b*. If some  $s_i''$  is *c*, the chain swap

$$\{1,2\}: d \leftrightarrow s_1'' \leftrightarrow s_2'' \leftrightarrow \cdots \leftrightarrow c$$

resolves the collisions in the first vote. Since *a* appears in the second vote at most once, the process (18) contains *a* at most once. If *a* does not appear in (18), the process does not hit the first position and we see that (16) ends in finite number of steps as in Lemma 1. If *a* appears in (18), the finiteness of (16) is proved by applying the similar discussion of Lemma 1 for the subprocess of (16) which starts from *a*.  $\Box$ 

Using Lemmas 5 and 6 we shall prove Lemma 4.

*Proof of Lemma 4* Without loss of generality, let i' = 1,  $[i_{im}, i_{pr}] = [2, 1]$ ,  $\iota_{11} = a$ , and  $\iota_{21} = b + c - a$ . Then  $\overline{I}$  looks like

$$\begin{bmatrix} a & * & \cdots & * \\ b + c - a & * & \cdots & * \\ & * & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots \\ & * & * & \cdots & * \end{bmatrix}$$

In the cases below, where a resulting data set is improper, [2, 1] will be a resolvable pair.

Case 1  $p'_{11} = a$ .

In this case in P' and hence in I, the candidate a appears at least once in the first position. Therefore a is in the first position in some vote i > 2 in I. Let i = 3. Then the votes [2, 3] of I form a resolvable pair and I can be transformed to a proper data set by Lemma 2. This corresponds to (2) of the lemma and is summarized as  $I \stackrel{\{2,3\}}{\longleftrightarrow} P$ .

Case 2  $p'_{11} \neq a$ , but *a* appears in the first vote of *P'*. Without loss of generality let  $p'_{12} = a$ . Let  $d = \iota_{12}$ .

Case 2-1  $\iota_{22} = a$ .

We perform the double swap  $a \stackrel{1}{\leftrightarrow} d \stackrel{2}{\leftrightarrow} a$  to the first two votes

$\int a$	d	*		*		[ d	a	*		*	
b+c-a	a	*	•••	*		b+c-d	d	*		*	
*	*	*	•••	*	$\rightarrow$	*	*	*	•••	*	Ι.
	÷	÷	÷	:		:	÷	÷	÷	:	
*	*	*		*		*	*	*		*_	

This increases  $C_{11}$ . This corresponds to (1) of the lemma and is summarized as  $I \stackrel{\{1,2\}}{\longleftrightarrow} I$ .

Case 2-2  $\iota_{22} \neq a$ .

Since  $p'_{12} = a$ , *a* has to appear in the second position of *I*. Without loss of generality, let  $\iota_{32} = a$ . Let  $e = \iota_{31}$  and  $f = \iota_{22}$ . Then  $\overline{P}$  looks like

а	d	*		*]
b + c - a	f	*	•••	*
е	а	*	•••	*
*	*	*	•••	*
:	÷	÷	÷	:
*	*	*		*

From Lemma 5 applied to votes {2, 3}, this case is reduced to Case 2-1. This case together with the subsequent operation of Case 2-1 is summarized as  $I \stackrel{\{2,3\}}{\longleftrightarrow} I$ .

Case 3 *a* does not appear in the first vote of P'.

Let  $d = p'_{11}$ ,  $d \neq a$ . If d = b or d = c, we directly go to the Cases 3-1 or Case 3-2 below. If  $d \neq b$ , c, we need an extra step as follows. Let  $\iota_{31} = d$  without loss of generality. Then  $\overline{I}$  looks like

$\int a$	*		*]
b+c-a	*	•••	*
d	*	•••	*
*	*		*
:	÷	÷	:
*	*		*

By Lemma 6 applied to votes  $\{2, 3\}$ , we move *d* to the second vote resolving the possible collisions. At this point the (2, 1)-element of *I* may be b + d - a or c + d = a

d-a. We consider the former case without loss of generality. Then  $\overline{I}$  looks like

$$\begin{bmatrix} a & * & \cdots & * \\ b + d - a & * & \cdots & * \\ c & * & \cdots & * \\ * & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots \\ * & * & \cdots & * \end{bmatrix}.$$
(19)

Case 3-1 d appears in the first vote of I.

Let  $d = \iota_{12}$  without loss of generality. We apply a double swap  $a \stackrel{1}{\leftrightarrow} d \stackrel{2}{\leftrightarrow} a$ :

a	d	*	•••	*		d	а	*	•••	*]	
b+d-a	е	*	•••	*		b	d + e - a	*	• • •	*	
*	*	*	• • •	*	$\rightarrow$	*	*	*	• • •	*	
	•				,						
:	:	:	:	:		:	:	:	:	:	
*	*	*		*_		*	*	*		*	

This case is summarized as  $I \stackrel{\{1,2\}}{\longleftrightarrow} I$  or  $I \stackrel{\{2,3\}}{\longleftrightarrow} I \stackrel{\{1,2\}}{\longleftrightarrow} I$ , where  $I \stackrel{\{2,3\}}{\longleftrightarrow} I$  is needed for the case  $d \neq b, c$ . We do not repeat this comment for the other cases below. Case 3-2 d does not appear in the first vote of I.

If *a* appears only once in the second vote, say in the *j*th position, j > 1, then we can apply the swap  $\{1, 2\} : a \stackrel{1}{\leftrightarrow} d$  to make *I* proper, which is summarized as  $I \stackrel{\{1,2\}}{\longleftrightarrow} P$  or  $I \stackrel{\{2,3\}}{\longleftrightarrow} I \stackrel{\{1,2\}}{\longleftrightarrow} P$ .

Hence we consider the case that *a* appears in two positions labeled by  $j_1$ ,  $j_2$ ,  $1 < j_1 < j_2$  of the second vote of *I*. Since *P'* does not contain *a* in the first vote, *I* has a vote not containing *a*.

Case 3-2-1 The third vote of (19) contains *a*.

Without loss of generality, suppose the fourth vote of *I* does not contain *a*. Denote  $e = \iota_{41}$ . Interpreting two *a*'s in the second vote as a collision, we try to resolve the collision by swapping *a* down to the fourth vote. Then we have two processes of swaps

$$\{2,4\}: a \stackrel{j_1}{\leftrightarrow} s_1 \leftrightarrow s_2 \leftrightarrow \cdots,$$
<sup>(20)</sup>

$$\{2,4\}: a \stackrel{j_2}{\leftrightarrow} s'_1 \leftrightarrow s'_2 \leftrightarrow \cdots .$$

$$(21)$$

During these processes the collisions occur in the second vote. Only one of these two can contain d. Then we can choose a process, say (20), which does not contain d. As in Lemma 1 no candidate appears twice in (20). Hence, (20) is a finite chain swap resolving the collisions.

At this stage  $\overline{I}$  looks like

ſ	- a	*		*		a	*		*	
	b + d - a	*		*		e+d-a	*		*	
I	с	*	• • •	*		С	*	• • •	*	
	е	*		*	or	b	*	• • •	*	
	*	*		*	01	*	*	• • •	*	
	:	÷	÷	÷		:	÷	÷	÷	
l	*	*		*_		*	*	• • •	*	

In either case, the swap  $\{1, 2\}$ :  $a \stackrel{1}{\leftrightarrow} d$  increases *C* and makes *I* proper. The whole process for this case is summarized as  $I \stackrel{\{2,4\}}{\longleftrightarrow} I \stackrel{\{1,2\}}{\longleftrightarrow} P$  or  $I \stackrel{\{2,3\}}{\longleftrightarrow} I \stackrel{\{2,4\}}{\longleftrightarrow} I \stackrel{\{1,2\}}{\longleftrightarrow} P$ .

Case 3-2-2 The third vote of (19) does not contain *a*.

We can just use the third vote of (19) as the fourth vote of the previous case. Hence  $I \stackrel{\{2,4\}}{\longleftrightarrow} I$  is replaced by  $I \stackrel{\{2,3\}}{\longleftrightarrow} I$  and this case is summarized as  $I \stackrel{\{2,3\}}{\longleftrightarrow} I \stackrel{\{1,2\}}{\longleftrightarrow} P$ .

We now summarize what we have proved so far. We will again discuss the following result in Sect. 3.1.

Let *P* and *P'* be two proper data sets with the same sufficient statistic, respectively. Suppose that the *i*th vote of *P* and the *i'*th vote of *P'* are different, i.e.,  $V^{(i)} \neq V^{(i')}$ . If we allow improper data sets, then by a sequence of swap operations among two votes of *P*, we can make the *i*th vote of *P* identical with the *i'*th vote of *P'*. Then we throw away this common vote from the two data sets and repeat the procedure. It should be noted that *P* may have been transformed to an improper data set *I* when two votes coincide, but *I* contains a resolvable pair  $[i_{im}, i_{pr}]$  with  $i_{pr} \neq i$ . Hence we can continue this process until *P* is fully transformed to *P'*.

In order to finish our proof of Theorem 1, we have to show that each intermediate improper data set can be temporarily transformed to a proper data set and the consecutive proper data sets are connected by operations among three votes.

We decompose the whole process of transforming P to P' into segments that consist of transformations from a proper data set to another proper data set with improper intermediate steps. One segment is depicted as follows:

$$P_1 \longleftrightarrow I_1 \longleftrightarrow \cdots \longleftrightarrow I_i \longleftrightarrow I_{i+1} \longleftrightarrow \cdots \longleftrightarrow I_m \longleftrightarrow P_m, \qquad (22)$$

where each  $\longleftrightarrow$  (omitting *R*) denotes a swap operation among two votes in Lemmas 3 and 4. By these lemmas, the number of concurrences in  $P_m$  is larger than in  $P_1$ . We claim that for any consecutive improper data sets  $I_i$ ,  $I_{i+1}$ , we can find proper data sets  $P_i$ ,  $P'_i$ ,  $P'_{i+1}$  satisfying

$$P_i \longleftrightarrow I_i \longleftrightarrow I_{i+1} \longleftrightarrow P'_{i+1}, \tag{23}$$

$$P'_i \longleftrightarrow I_i \longleftrightarrow P_i.$$
 (24)

The swap operation for  $I_i \leftrightarrow I_{i+1}$  is compatible with both data sets. Hence if we choose a common resolvable pair for  $I_i$  and  $I_{i+1}$ , then (23) for transforming  $P_i$  to  $P'_i$  involves three votes. On the other hand, since both  $P'_i \leftrightarrow I_i$  and  $I_i \leftrightarrow P_i$  involve an improper vote, the operation of transforming  $P'_i$  to  $P_i$  involves three votes. This completes the proof of Theorem 1.

## **3** Discussion

In this section we discuss some topics related to our main result.

3.1 Extension of fibers by allowing one negative element

As discussed after the proof of Lemma 4, we have shown the following result by our proof of Theorem 1 (cf. (22)).

**Proposition 1** Let P and P' be any two proper data sets with the same sufficient statistic. If we allow improper data sets as the intermediate states, P and P' are connected by swap operations among two votes, whose single operation is used at each step.

Since an improper data set has one -1, Proposition 1 seems to suggest that every fiber  $\mathcal{F}_t$  for the configuration  $A_{n,r}$  becomes connected by degree two moves if we extend  $\mathcal{F}_t$  by allowing one negative element  $x(\sigma) = -1$  in x which satisfies t = Ax. However, this is incorrect. In fact, allowing -1 in data set and allowing -1 in  $\mathcal{F}_t$  are two different things.

For example, consider the case of n = 3 and r = 2 with candidates labeled as a, b, c. It is easy to see that dim ker  $A_{3,2} = 1$  and  $I_{A_{3,2}}$  is a principal ideal generated by a single binomial p(ab)p(bc)p(ca) - p(ac)p(cb)p(ba). Hence there is no degree two move in ker<sub>Z</sub>  $A_{3,2}$ . Yet, we can connect two data sets

$$P = \begin{bmatrix} a & b \\ b & c \\ c & a \end{bmatrix}, \qquad P' = \begin{bmatrix} a & c \\ c & b \\ b & a \end{bmatrix}$$

by applying  $\{2, 3\}$ :  $a \stackrel{1}{\leftrightarrow} b$ ,  $\{1, 2\}$ :  $b \stackrel{2}{\leftrightarrow} c$  and  $\{2, 3\}$ :  $a \stackrel{1}{\leftrightarrow} c$  in this order:

$$\begin{bmatrix} a & b \\ b & c \\ c & a \end{bmatrix} \rightarrow \begin{bmatrix} a & b \\ a & c \\ b + c - a & a \end{bmatrix} \rightarrow \begin{bmatrix} a & c \\ a & b \\ b + c - a & a \end{bmatrix} \rightarrow \begin{bmatrix} a & c \\ c & b \\ b & a \end{bmatrix}.$$

Note that the middle two data sets can be interpreted either as

adding 
$$(ab), (ac), (bc), (ca)$$
 and subtracting  $(ac)$ 

or as

adding 
$$(ab)$$
,  $(ac)$ ,  $(ba)$ ,  $(cb)$  and subtracting  $(ab)$ .

However the middle two data sets do not correspond to an element of a fiber for  $A_{3,2}$ .

#### 3.2 Normality

It is natural to ask if the semigroup generated by  $A_{n,r}$  is normal. Consider the set Q of  $n \times r$  real matrices  $X = \{x_{ij}\}$  satisfying

$$0 \le x_{ij} \le 1, \ \forall i, j, \quad \sum_{j=1}^r x_{ij} \le 1, \forall i, \qquad \sum_{i=1}^n x_{ij} = 1, \forall j.$$

The set Q is the Birkhoff polytope in  $\mathbb{R}^{n \times r}$ , which is a special case of transportation polytopes [5]. By [3] the set of vertices of Q is exactly the same as the set of columns of  $A_{n,r}$ . Then by the results of [7] and [9] the semigroup generated by  $A_{n,r}$  is normal. Lemma 1 is a consequence of this normality, because by the normality there exist two valid votes with the same sufficient statistic as the *i*th and the *i*'th votes in Lemma 1. These two proper votes can be obtained from the two votes of a swap operation in Lemma 1. However, the normality is not useful in proving Lemmas 5 and 6.

#### 3.3 Generation of moves for running a Markov chain

Based on Theorem 1 we can run a Markov chain for general r as follows. We randomly generate two or three proper votes of r candidates out of n candidates. Once these votes are obtained, we randomly perform permutations of candidates in the same position. We do this for each position. If no collision occurs, then we have two or three proper votes. If the obtained set of votes is different from the initial set, then the difference is a move. In this way, we obtain a random move of degree two or three and then run a Markov chain over a given fiber. Extensive computational results on Markov bases of (n, r)-Birkhoff models for  $r \le 5$  and arbitrary n are available from the authors.

Acknowledgements This work is partially supported by Grant-in-Aid for JSPS Fellows (No. 12J07561) from Japan Society for the Promotion of Science (JSPS). The authors thank the anonymous referees for their valuable comments.

## References

- Aoki, S., Hara, H., Takemura, A.: Markov Bases in Algebraic Statistics. Springer Series in Statistics. Springer, New York (2012)
- 2. Aoki, S., Takemura, A.: Minimal basis for a connected Markov chain over  $3 \times 3 \times K$  contingency tables with fixed two-dimensional marginals. Aust. N. Z. J. Stat. **45**(2), 229–249 (2003)
- Brualdi, R.A., Dahl, G.: An extension of the polytope of doubly stochastic matrices. Linear Multilinear Algebra 61(3), 393–408 (2013)
- Diaconis, P., Eriksson, N.: Markov bases for noncommutative Fourier analysis of ranked data. J. Symb. Comput. 41(2), 182–195 (2006)
- Haase, C., Paffenholz, A.: Quadratic Gröbner bases for smooth 3 × 3 transportation polytopes. J. Algebr. Comb. 30(4), 477–489 (2009)
- Jacobson, M.T., Matthews, P.: Generating uniformly distributed random Latin squares. J. Comb. Des. 4(6), 405–437 (1996)
- Ohsugi, H., Hibi, T.: Convex polytopes all of whose reverse lexicographic initial ideals are squarefree. Proc. Am. Math. Soc. 129(9), 2541–2546 (2001) (electronic)
- 8. Sturmfels, B., Welker, V.: Commutative algebra of statistical ranking. J. Algebra 361, 264–286 (2012)
- Sullivant, S.: Compressed polytopes and statistical disclosure limitation. Tohoku Math. J. 58(3), 433– 445 (2006)