Distributions with Fixed Marginals and Related Topics
IMS Lecture Notes – Monograph Series Vol. 28, 1996

Isotonic Regression on Permutations

By H. W. Block[1], S. Qian, and A. R. Sampson[1]
University of Pittsburgh

Motivated by an approach to qualifying potential judges, we study isotonic regression problems on a partially ordered set of permutations. We consider the partial orders discussed in Block, Chhetry, Fang and Sampson (1990) which are used for comparing the dependence of bivariate empirical distributions with fixed marginals. We give a method to generate permutations and their inversion numbers, and develop a technique to input these orders. We discuss methods of finding predecessors and immediate predecessors in the sense of these orders. Then, we develop an algorithm to search for isotonic regressions on a set of permutations under these orders.

1. Introduction and Motivation. This paper presents the algorithms and programs necessary to solve isotonic regression problems involving partial orders on permutations. Our solution depends on some approaches to identifying predecessors for three partial orders given in Block, Chhetry, Fang and Sampson (1990) and utilizes results of Block, Qian and Sampson (1994) for computing isotonic regressions over partially ordered sets. The partial orders of Block et al. (1990) are used for comparing the dependence of bivariate empirical distributions. These distributions have fixed marginals putting mass $1/n$ at $1, \ldots, n$ where n is the sample size.

One motivation for considering the isotonic regression problem of this paper is a new approach for qualifying potential judges by utilizing one known expert's rating of k distinct objects according to some criteria. While we present the necessary computations for implementing this approach, we do not present any formal statistical modeling.

Suppose that we wish to evaluate a number of different possible judges who will be expected to rank individuals in a given setting, e.g., wine tastings,

---

[1]Supported in part by National Security Agency Grant No. MDA-904-90-H-4036 and in part by National Science Foundation Grant No. DMS-9203444.

AMS 1991 Subject Classification: Primary 62G05; Secondary 20B99

Key words and phrases: Partial orders, permutation, inversion numbers, isotonic regression.

athletic competitions or "beauty" contests. To test potential judges, we suppose that we have a single known expert's ranking of k distinct objects from worst to best according to some qualitative criteria. For example, suppose that we have a wine expert who provides a rank ordering for the quality of eight 1989 Bordeaux wines from worst to best with no ties permitted. For convenience, we label each wine by its expert ranking, i.e. $1 \equiv$ worst, $\ldots, 8 \equiv$ best. The evaluator of the potential judges now picks T distinct reorderings of the expert's order ($T \leq 8!$). Each of these T reorderings can be described by the corresponding permutation. These reorderings are then presented, one at a time, to a potential judge, who is asked to provide, according to his opinion, a percentage score for correctness of the presented reorderings. This process using the T reorderings is repeated for each of the potential judges.

The evaluator develops the T reorderings of the expert's evaluation by following one of three partial orderings on the set of permutations. That is, one particular partial ordering is selected and the evaluator takes T reorderings of the expert's ranking according to the rules of the partial ordering. We now describe in detail how the evaluator would utilize each of these partial orderings to obtain the T reorderings.

As an example of our approach, we consider the wine tasting setting. Initially, the bottles of the eight wines are lined up in the expert's order from worst (1) to best (8). Then the bottles of wine are moved around according to the rules of the selected partial order until a reordering is obtained. A photograph of this reordering is taken and this is one of the T reorderings presented to the subject. This process is repeated to obtain the other $T-1$ reorderings. Each of these photographs is then presented to a potential judge who is told that this is a ranking from worst to best (physically ordered from left to right) and is asked to give a grade for how good this ranking is. If the judge has good ability, we would expect high scores to be given to rankings similar to the expert's rankings and low scores to those which are quite different from the expert's ranking. Moreover, if one ordering is "closer" to the experts order than another, then the former score should be higher than the latter.

We now describe the three partial orders (designated b_1, b_2 or b_3) which would be used to obtain the reorderings. The b_1 ordering involves a finite sequence of switches, where the evaluator may switch out of order any two wine bottles among the eight, which are in the expert's original order, (e.g., (13546278) may be switched to (13546872)). The second ordering, the b_2 ordering, only permits the evaluator to sequentially switch wine bottles which were originally adjacent in the expert's original order (e.g., (13546278) may be switched to (13547268)). This allows for the reordering of the wine where changes are very subtle, and is of use in discerning a potential judge's ability to make fine discriminations. The third (b_3) ordering that we consider permits the evaluator to sequentially switch neighboring bottles out of the expert's order (e.g., (13546278) can be switched to (13564278)). This latter ordering can be viewed as an ordering for switching convenience, i.e., moving neighboring bottles. A more rigorous treatment of these three partial orderings can be found in Block, Chhetry, Fang and Sampson (1990).

Let $\mathbf{i}$ and $\mathbf{j}$ be two of the evaluator's T reorderings which were arranged according to one of the three orderings b_1, b_2 or b_3. A potential judge will be in *concordance with the expert* according to a particular partial ordering if $\mathbf{i}$ is better ordered than $\mathbf{j}$ implies that the potential judge scores the ordering $\mathbf{i}$ at least as high as that of $\mathbf{j}$. Note that $\mathbf{i}$ is better ordered than $\mathbf{j}$ if $\mathbf{i}$ is in some sense closer to the expert's rating than is $\mathbf{j}$, according to that partial order.

If the evaluator has chosen many test permutations, then it is unrealistic to expect the potential judge to be in perfect concordance with all these permutations with respect to the fixed partial ordering under consideration. To measure each potential judge's degree of discrepancy, we use the measure

$$\min \sum (s(\mathbf{i}) - f(\mathbf{i}))^2$$

where the sum is over all T permutations and the minimum is taken over all functions f which are isotonic with respect to the ordering, that is, $\mathbf{i}$ better ordered than $\mathbf{j}$ implies $f(\mathbf{i}) \geq f(\mathbf{j})$, and where $s(\mathbf{i})$ is the potential judge's score for permutation $\mathbf{i}$ subject to $s(\mathbf{i})$ being constrained in some way so that the scores are comparable across potential judges. From this measure, we can see how far the potential judge is from the closest scoring which is in perfect concordance with the given ordering. To compute this measure, we need to find the isotonic regression of the judge's score function on the set of T given permutations with respect to the selected partial order. One could then use this measure, computed for each judge, to decide if each judge should be qualified or not.

We note that we motivate and apply our results in the context of qualifying judges. However, in the spirit of Block et al. (1990) one could consider any function defined on the set of bivariate empirical rank distributions and one of the four orderings for positive dependence and then isotonize that function with respect to the given ordering. Our methods would apply to such a problem.

In Section 2, we formulate the problem and in Section 3 study methods of finding an immediate predecessor of a permutation in S_n with respect to certain partial orders. Sections 4 and 5 prove further computational details. In Section 6 specific computations are given for various choices of the function s. The program for our algorithm is given in the Appendix.

2. Problem Formulation. Let S_n be the set of all permutations of the n integers $\{1, 2, \ldots, n\}$. Partial orders on S_n have been studied in statistics, computer science, discrete mathematics and other areas. Block, Chhetry, Fang and Sampson (henceforth BCFS) (1990) gave a unified approach to three well known partial orders on S_n and introduced a new one. They called these partial orders the b_1, b_2, b_3 and b_4 orders. BCFS (1990) showed that the b_1, b_2, b_3 and b_4 orders correspond to the more concordant, more row regression, more column regression and more associated orders on the class of bivariate empirical rank distributions, respectively. While our motivation was based on the three orders b_1, b_2 and b_3, we include results for b_4 for completeness.

Let $\mathbf{i} = (i_1, i_2, \ldots, i_n) \in S_n$. An inversion of $\mathbf{i}$ is a pair of indices (k, l) of $\mathbf{i}$ with $k < l$ and $i_k > i_l$. An inversion (k, l) of $\mathbf{i}$ is said to be of type 2 if $i_k - i_l = 1$. An inversion (k, l) of $\mathbf{i}$ is said to be of type 3 if $l = k + 1$, i.e., i_k and i_l are adjacent elements in the permutation. The inversion number of a permutation $\mathbf{i}$ is the number of inversions contained in $\mathbf{i}$, and is denoted as $m(\mathbf{i})$. It is well known that $0 \leq m(\mathbf{i}) \leq n(n-1)/2$, for any $\mathbf{i} \in S_n$. An interchange of two components i_k and i_l of a permutation $\mathbf{i}$ is said to be a correction if (k, l) is an inversion of $\mathbf{i}$.

A permutation $\mathbf{i}$ is said to be better ordered than $\mathbf{j}$ in the sense of b_1-order, written as $\mathbf{i} \geq_1 \mathbf{j}$, if $\mathbf{i} = \mathbf{j}$ or $\mathbf{i}$ is obtainable from $\mathbf{j}$ in a finite number of steps, each of which consists of a correction of an inversion. A permutation $\mathbf{i}$ is said to be better ordered than $\mathbf{j}$ in the sense of b_2-order, written as $\mathbf{i} \geq_2 \mathbf{j}$, if $\mathbf{i} = \mathbf{j}$ or $\mathbf{i}$ is obtainable from $\mathbf{j}$ in a finite number of steps, each of which consists of interchanging an inversion of type 2. A permutation $\mathbf{i}$ is said to be better ordered than $\mathbf{j}$ in the sense of b_3-order, written as $\mathbf{i} \geq_3 \mathbf{j}$, if $\mathbf{i} = \mathbf{j}$ or $\mathbf{i}$ is obtainable from $\mathbf{j}$ in a finite number of steps, each of which consists of interchanging an inversion of type 3. A permutation $\mathbf{i}$ is said to be better ordered than $\mathbf{j}$ in the sense of b_4-order, written as $\mathbf{i} \geq_4 \mathbf{j}$, if $\mathbf{i} = \mathbf{j}$ or $\mathbf{i}$ is obtainable from $\mathbf{j}$ in a finite number of steps, each of which consists of interchanging an inversion of type 2 or type 3.

BCFS (Theorem 2.5, 1990) showed that the b_2- and b_3-orders are not equivalent and each implies the b_4-order; and that the b_4-order implies the b_1-order.

A real valued function f on S_n is said to be isotonic with respect to a b_t-order, if $f(\mathbf{i}) \geq f(\mathbf{j})$, whenever $\mathbf{i} \geq_t \mathbf{j}$, where $t = 1, 2, 3$ or 4. The class of all isotonic functions on S_n with respect to a b_t-order is denoted as I_t. A real valued function s^* on S_n is said to be an isotonic regression of a given function s with nonnegative weights w, if s^* is the solution of the following problem:

$$\min \sum_{x \in S_n} (s(x) - f(x))^2 w(x) \qquad \text{subject to } f \in I_t. \tag{2.1}$$

For any given function s on S_n with positive weights $w(\cdot)$, the objective functional is continuous and strictly convex. Hence, there exists a unique isotonic regression of s with weights w.

Problems of the form (2.1) are called isotonic regression problems on the set of permutations subject to the b_t-orders. These problems can arise in the evaluation of ranking problems and evaluating disorder in computer sorting algorithms. A comprehensive reference for isotonic regression is Robertson, Wright and Dykstra (1988). Recently, Block, Qian and Sampson (henceforth, BQS) (1992, 1994) gave a unified approach to a wide class of algorithms for isotonic regressions and developed some new efficient algorithms, especially for partial orders.

A partial order on a finite set X can be represented as a directed graph without cycles, but this representation is not unique. The representation with a minimum number of edges is called a Hasse diagram. In this diagram, each edge is a pair of elements in X such that one of them is an immediate predecessor of the other. The advantage of this representation is that the depiction of the partial order is compact and easy to handle. Additionally, in computation, this representation saves a significant amount of computer memory. Consequently, it is commonly used in algorithms for partial orders. The IBCR algorithm developed by BQS (1994), which searches for an isotonic regression on a partially ordered set is able to use this representation to facilitate the computation of various partial orders. In order to utilize this representation, we must know all immediate predecessors of each element in X. Therefore, it is basic to find the immediate predecessors of each element in S_n, in order to apply the IBCR algorithm to problem (2.1).

3. Immediate Predecessors Under the b_t-Orders. In this section we study methods for finding immediate predecessors of permutations in S_n with respect to the b_t-orders.

Let $t = 1, 2, 3$ or 4, and let $\mathbf{j}, \mathbf{i} \in S_n$. The permutation $\mathbf{i}$ is said to be a predecessor of $\mathbf{j}$ in the sense of b_t-order, if $\mathbf{j} \geq_t \mathbf{i}$ and $\mathbf{j} \neq \mathbf{i}$; the permutation $\mathbf{k}$ is said to be an immediate predecessor of $\mathbf{j}$ in the sense of b_t-order, if $\mathbf{k}$ is a predecessor of $\mathbf{j}$ and no permutation is between $\mathbf{k}$ and $\mathbf{j}$ in the sense of the same b_t-order. Recall that $m(\mathbf{i})$ is the inversion number of the permutation $\mathbf{i}$.

LEMMA 3.1. *Let $\mathbf{i}$ be a predecessor of $\mathbf{j}$ in S_n in the sense of b_t-order with $t \in \{1, 2, 3, 4\}$. Then $m(\mathbf{i}) > m(\mathbf{j})$.*

PROOF. When $\mathbf{j}$ is obtained from $\mathbf{i}$ by correcting an inversion, it is well known that $m(\mathbf{i}) > m(\mathbf{j})$. If $\mathbf{i}$ is a predecessor of $\mathbf{j}$ in the sense of b_1-order, by transitivity, we have $m(\mathbf{i}) > m(\mathbf{j})$. The proposition is true for other b_t-orders, because each of them implies the b_1-order. ∎

COROLLARY 3.2. *Let $t = 1, 2, 3$ or 4, and let $\mathbf{i}$ be a predecessor of $\mathbf{j}$ in S_n in the sense of b_t-order. Then $m(\mathbf{i}) - m(\mathbf{j}) = 1$ implies that $\mathbf{i}$ is an immediate predecessor of $\mathbf{j}$ in the sense of the b_t-order.*

LEMMA 3.3. *If $\mathbf{i}$ is an immediate predecessor of $\mathbf{j}$ in S_n in the sense of one of the b_t-orders, then $\mathbf{j}$ is obtained from $\mathbf{i}$ by a correction of an inversion.*

LEMMA 3.4. *A permutation $\mathbf{j}$ is obtained from a permutation $\mathbf{i}$ by a correction of an inversion if and only if $m(\mathbf{i}) > m(\mathbf{j})$ and $\mathbf{i}$ and $\mathbf{j}$ differ by exactly two different components.*

PROOF. The necessity of the condition is proved by Lemma 3.1. If $\mathbf{i}$ and $\mathbf{j}$ differ by exactly two different components, k and l, then $\mathbf{j}$ is obtained from $\mathbf{i}$ by interchanging the k-th and l-th components of $\mathbf{i}$. Since $m(\mathbf{i}) > m(\mathbf{j})$, we have $(k - l)(i_k - i_l) < 0$. Hence, $\mathbf{j}$ is obtained from $\mathbf{i}$ by a correction of an inversion (k, l). ∎

LEMMA 3.5. *Let $\mathbf{j}$ be obtained from $\mathbf{i}$ by interchanging the k-th and l-th components of $\mathbf{i}$ with $k < l$ and $i_k > i_l$, i.e., $\mathbf{j}$ is obtained from $\mathbf{i}$ by a correction of an inversion. If for each index u between k and l, $i_u > i_k$, or $i_u < i_l$, then $m(\mathbf{i}) - m(\mathbf{j}) = 1$.*

PROOF. Assume for u between k and l that $i_u > i_k$. Since $i_u > i_k > i_l, i_u$ is responsible for only one inversion with respect to i_k and i_l in $\mathbf{i}$. Similarly in $\mathbf{j}, i_l$ and i_u are ordered, and i_u and i_k are disordered, so i_k causes only one inversion in $\mathbf{j}$. If $i_u < i_l$, the argument is similar. Consequently the net change from $\mathbf{i}$ to $\mathbf{j}$ is one, i.e., the correction of the inversion in the k and l positions. ∎

COROLLARY 3.6.

(1) *If $\mathbf{j}$ is obtained from $\mathbf{i}$ by a correction of an inversion of type 2, then $m(\mathbf{i}) - m(\mathbf{j}) = 1$;*

(2) *If $\mathbf{j}$ is obtained from $\mathbf{i}$ by a correction of an inversion of type 3, then $m(\mathbf{i}) - m(\mathbf{j}) = 1$.*

THEOREM 3.7. *A permutation $\mathbf{i}$ is an immediate predecessor of a permutation $\mathbf{j}$ in the sense of b_1-order on S_n, if and only if,*

(1) *$m(\mathbf{i}) - m(\mathbf{j}) = 1$; and*

(2) *there are exactly two different components between $\mathbf{i}$ and $\mathbf{j}$.*

PROOF. Sufficiency follows from Corollary 3.2. Let $\mathbf{i}$ be an immediate predecessor of $\mathbf{j}$. By Lemma 3.3, $\mathbf{j}$ is obtained from $\mathbf{i}$ by a correction of an inversion. Hence, by Lemma 3.4, $m(\mathbf{i}) > m(\mathbf{j})$ and (2) is satisfied. Assume there exists an index u between the two different components k and l with $k < l$, otherwise $m(\mathbf{i}) - m(\mathbf{j}) = 1$. Now we assume $m(\mathbf{i}) - m(\mathbf{j}) > 1$. By Lemma 3.5, $i_l < i_u < i_k$. Let $\mathbf{j^1}$ be obtained by interchanging k-th and u-th components of $\mathbf{j^0} = \mathbf{i}$; let $\mathbf{j^2}$ be obtained by interchanging u-th and l-th components of $\mathbf{j^1}$; then $\mathbf{j^3} = \mathbf{j}$ is obtained by interchanging k-th and u-th components of $\mathbf{j^2}$. Because $i_l < i_u < i_k$, each $\mathbf{j^t}$ is obtained from $\mathbf{j^{t-1}}$ by a correction of an inversion. Thus $\mathbf{j^1}$ and $\mathbf{j^2}$ are between $\mathbf{i}$ and $\mathbf{j}$ in the sense of b_1-order. This contradicts $\mathbf{i}$ being an immediate predecessor of $\mathbf{j}$. ∎

If we know the inversion numbers for all permutations in S_n, we can easily find all immediate predecessors for each element in S_n in the sense of the b_1-order by Theorem 3.7. For any permutation $\mathbf{j}$, each immediate predecessor of $\mathbf{j}$ has inversion number $m(\mathbf{j}) + 1$. If a permutation with the inversion number $(m(\mathbf{j}) + 1)$ has exactly two different components from the permutation $\mathbf{j}$, it is an immediate predecessor of $\mathbf{j}$. All immediate predecessors of $\mathbf{j}$ in the sense of the b_1-order can be found in this way. Conditions 1) and 2) in Theorem 3.7 can be easily implemented in a program.

For b_2, b_3 and b_4 orders, we have similar theorems for their immediate predecessors. We summarize the results below.

THEOREM 3.8. *(a) For $t = 2, 3, 4$, a permutation $\mathbf{i}$ is an immediate predecessor of a permutation $\mathbf{j}$ in the sense of the b_t-order on S_n if and only if $\mathbf{j}$ is obtained from $\mathbf{i}$ by a correction of an inversion of type t.*

(b) Necessary and sufficient conditions for $\mathbf{i}$ to be an immediate predecessor of $\mathbf{j}$ are:

(1) $m(\mathbf{i}) - m(\mathbf{j}) = 1$;

(2) $\mathbf{i}$ and $\mathbf{j}$ differ in exactly two components;

(3) for the two components k and l of 2);

$$|i_k - i_l| = 1 \text{ (for } b_2\text{-orderings)} ,$$

$$|k - l| = 1 \text{ (for } b_3\text{-orderings)} ,$$

$$|i_k - i_l| = 1 \text{ , or } |k - l| = 1 \text{ (for } b_4\text{-orderings)} .$$

4. Generating the Elements of S_n and Their Inversion Numbers. From Section 3, it is clear that the inversion numbers of the permutations play an important role in searching for immediate predecessors of an element in S_n in the sense of b_t-orders. The inversion number of a permutation can be computed by its definition, but we develop in this section an efficient way of finding the inversion numbers for all elements in S_n, utilizing the structure of S_n. While there are many algorithms to generate permutations we know of none to find inversion numbers. The set of permutations, S_n, has $n!$ elements, and the range of the inversion numbers for $\mathbf{i} \in S_n$ is $\{0, 1, \ldots, b\}$ with $b = n(n-1)/2$. We begin by using the inversion numbers to divide S_n into $b+1$ subsets. The subset containing all the permutations with inversion number u is called the u-th layer of S_n and is denoted as $S_{n,u}$. For example, for $S_2 = \{(1\ 2), (2\ 1)\}, \{(1\ 2)\}$ is the 0-th layer and $\{(2\ 1)\}$ is the 1st layer of S_2.

We use a recursive method to generate S_n from S_{n-1}, where $a = (n-1)(n-2)/2$ is the maximal inversion number of S_{n-1}. Assume that we have obtained S_{n-1} with layers $S_{n-1,0}, S_{n-1,1}, \ldots, S_{n-1,a}$, and assume the k-th layer $S_{n-1,k}$ has $w_{n-1,k}$ elements. For each permutation $\mathbf{i}$ in S_n, we can view $\mathbf{i}$ as obtained by inserting the integer n into a permutation $\mathbf{j}$ of order $(n-1)$. For each permutation $\mathbf{j} = (j_1, j_2, \ldots, j_{n-1})$, there are n locations available for inserting the integer n. Let h be an integer between 1 and n. We define an inserting function ϕ_h on S_{n-1} to S_n as follows:

$$\phi_h(j) = (j_1, \ldots, j_{h-1}, n, j_h, \ldots, j_{n-1}) .$$

Let A be a subset of S_n, and $\phi_h(A)$ denote the range of ϕ_h on A. Obviously the function ϕ_h is a one to one correspondence from S_{n-1} onto $\phi_h(S_{n-1}) = \{i \in S_n : i_h = n\}$. The set S_n is a union of $\phi_h(S_{n-1}), h = 1, 2, \ldots, n$, i.e., $S_n = \cup\{\phi_h(S_{n-1}) : h = 1, 2, \ldots, n\}$.

THEOREM 4.1. *Let h be an integer between 1 and n inclusive, and let $\mathbf{j} \in S_{n-1}$. Then*

$$m(\phi_h(\mathbf{j})) = m(\mathbf{j}) + (n-h) .$$

PROOF. The function ϕ_h inserts n into the h-th location of $\mathbf{j}$, which generates $n-h$ inversions for integer n, and the inversions of other integers do not change after the insertion. Hence, $m(\phi_h(\mathbf{j})) = m(\mathbf{j}) + (n-h)$. ∎

COROLLARY 4.2. *Let $w_{n,x}$ be the numbers of elements in $S_{n,x}$, the x-th layer of S_n and $a = (n-1)(n-2)/2$.*

(1) For $x = 0, 1, \ldots, n-1$,

$$S_{n,x} = \phi_n(S_{n-1,x}) \cup \phi_{n-1}(S_{n-1,x-1}) \cup \ldots \cup \phi_{n-x}(S_{n-1,0}), \quad \text{and}$$
$$w_{n,x} = w_{n-1,x} + w_{n-1,x-1} + \ldots + w_{n-1,0} .$$

(2) For $x = n, n+1, \ldots, a$,

$$S_{n,x} = \phi_n(S_{n-1,x}) \cup \phi_{n-1}(S_{n-1,x-1}) \cup \ldots \cup \phi_1(S_{n-1,x-n+1}) \text{ , and}$$
$$w_{n,x} = w_{n-1,x} + w_{n-1,x-1} + \cdots + w_{n-1,x-n+1} \ .$$

(3) For $x = a, a+1, \ldots, n(n-1)/2$,

$$S_{n,x} = \phi_{n-x+a}(S_{n-1,a}) \cup \phi_{n-x+a-1}\ (S_{n-1,a-1}) \cup \ldots \cup \phi_1(S_{n-1,x-n+1}) \ ,$$
and
$$w_{n,x} = w_{n-1,a} + w_{n-1,a-1} + \cdots + w_{n-1,x-n+1} \ .$$

By the above analysis we can generate S_n with layers by our recursion method. We can search for immediate predecessors of a permutation $\mathbf{j}$ in the $(m(\mathbf{j})+1) - st$ layer of S_n. This reduces the number of candidates for immediate predecessors of $\mathbf{j}$. Based upon the results in Sections 3 and 4, we develop a program called IBCRb to generate S_n with layers, find immediate predecessors in the sense of b_t-orders, and search for the isotonic regression on S_n for a given function g with non-negative weights w. This program is described in the Appendix.

5. The b_t-Orders on a Subset of S_n. Let X be a subset of S_n with only a few permutations. In order to increase the efficiency of computing the isotonic regression, we do not want to generate the whole permutation set S_n. In this situation, we cannot utilize the structure of S_n to find immediate predecessors in the sense of b_t-orders. The inversion numbers of permutations still play an important role in this case. We calculate the inversion number of a permutation by counting its inversions. In order to find immediate predecessors of a permutation in X in the sense of b_t-orders, we have to find its predecessors in X. By Lemma 3.1, we know that $\mathbf{i}$ is a predecessor of $\mathbf{j}$ implies $m(\mathbf{i}) > m(\mathbf{j})$.

Let $\mathbf{i} = (i_1, i_2, \ldots, i_n) \in S_n$. If we sort the first $l (\leq n)$ elements $i_1, i_2, \ldots, i_l$ of $\mathbf{i}$, then the resulting sequence is called the increasing rearrangement of the first l components of $\mathbf{i}$, denoted as $i(1,l) \leq i(2,l) < \ldots < i(l,l)$.

THEOREM 5.1. *A permutation* $\mathbf{i}$ *is a predecessor of* $\mathbf{j}$ *in the sense of* b_1*-order, if and only if,* $m(\mathbf{i}) > m(\mathbf{j})$ *and for each* $l = 1, 2, \ldots, n$, *we have,* $j(k,l) \leq i(k,l)$, $k = 1, 2, \ldots, l$, *where* $j(k,l)$, $k = 1, 2, \ldots, l$ *and* $i(k,l)$, $k = 1, 2, \ldots, l$ *are the increasing arrangements of the first* l *components of* $\mathbf{j}$ *and* $\mathbf{i}$, *respectively.*

Theorem 5.1 is due to Yanagimoto and Okamoto (1969).

THEOREM 5.2. *A permutation* $\mathbf{i}$ *is a predecessor of* $\mathbf{j}$ *in the sense of* b_2*-order, if and only if,* $m(\mathbf{j}\mathbf{i}^{-1}) = m(\mathbf{i}) - m(\mathbf{j}) > 0$, *where* $\mathbf{i}^{-1}$ *is the inverse of* $\mathbf{i}$, that is, $\mathbf{i}\mathbf{i}^{-1} = (1, 2, \ldots, n)$.

THEOREM 5.3. *A permutation* $\mathbf{i}$ *is a predecessor of* $\mathbf{j}$ *in the sense of* b_3*-order, if and only if,* $m(\mathbf{j}^{-1}\mathbf{i}) = m(\mathbf{i}) - m(\mathbf{j}) > 0$.

Proofs of Theorems 5.2 and 5.3 can be found in BCFS (1993). According to these theorems, we can easily check whether a permutation $\mathbf{i}$ is a predecessor of $\mathbf{j}$ in X in the sense of b_1, b_2 and b_3 orders.

It seems to us that identifying a predecessor of a permutation in the sense of b_4-order is not as easy as the other b_t-orders. For a permutation $\mathbf{j}$ in X, we know that predecessors of $\mathbf{j}$ in the sense of b_2 or b_3 order are predecessors of a $\mathbf{j}$ in the sense of b_4-order, but there are some predecessors of $\mathbf{j}$ in the sense of the b_4-order that are not predecessors in the sense of the b_2 or b_3 orders. In order to find all predecessors of a $\mathbf{j}$ in the sense of b_4-order, we can use the predecessors of $\mathbf{j}$ in the sense of the b_1-order as the candidates for predecessors of a $\mathbf{j}$ in the sense of the b_4-order, and then check these using the definition of the b_4-order. It is easy to find a candidate to be a predecessor of $\mathbf{j}$ in the sense of the b_1-order, but it is difficult to identify whether or not this candidate is a predecessor of $\mathbf{j}$ in the sense of the b_4-order. We must try every possibility before we say that a candidate is not a predecessor of $\mathbf{j}$ in the sense of b_4-order.

EXAMPLE 5.4. Let $\mathbf{i} = (4, 3, 5, 1, 2)$, $\mathbf{j} = (3, 1, 5, 4, 2)$ and $X = \{\mathbf{i}, \mathbf{j}\}$. It is easy to see that $m(\mathbf{i}) = 7$ and $m(\mathbf{j}) = 5$. Thus $m(\mathbf{i}) - m(\mathbf{j}) = 2$. Since $\mathbf{i}^{-1} = (4, 5, 2, 1, 3)$ and $\mathbf{j}^{-1} = (2, 5, 1, 4, 3)$, we have, $\mathbf{j}\mathbf{i}^{-1} = (4, 2, 1, 3, 5)$ and $m(\mathbf{j}\mathbf{i}^{-1}) = 4$; $\mathbf{i}\mathbf{j}^{-1} = (4, 1, 3, 2, 5)$ and $m(\mathbf{i}\mathbf{j}^{-1}) = 4$. Therefore, by Theorems 5.2 and 5.3, $\mathbf{i}$ is not a predecessor of $\mathbf{j}$ in the sense of the b_2 or b_3-order. It is easy to see that $\mathbf{i}$ is a predecessor of $\mathbf{j}$ in the sense of the b_1-order by Theorem 5.1. Now we check to see whether $\mathbf{i}$ is a predecessor of $\mathbf{j}$ in the sense of b_4-order. In S_5, $\mathbf{i}$ has 3 immediate successors in the sense of b_4, $(3, 4, 5, 1, 2)$, $(4, 3, 1, 5, 2)$ and $(4, 2, 5, 1, 3)$. For two of these elements, there are more than two components differing from those of $\mathbf{j}$, and the difference of the inversion number is exactly 1. For the third, correction of the inversion to reach $\mathbf{j}$ is not of type 2 or 3. Therefore, $\mathbf{i}$ is not a predecessor of $\mathbf{j}$ in the sense of the b_4-order.

After we find predecessors for all elements in X, we can use the definition to find immediate predecessors, that is, $\mathbf{i}$ is an immediate predecessor of $\mathbf{j}$ if and only if $\mathbf{i}$ is a predecessor of $\mathbf{j}$ and there is no other permutation in X between $\mathbf{i}$ and $\mathbf{j}$.

6. Comparison of Potential Judges. The preceding sections provided both the motivation and the techniques necessary to utilize our measure of discrepancy to evaluate a potential judge. However, as noted in Section 2, to compare two or more potential judges by the proposed measure of discrepancy, we need to provide some type of constraints on the scores that the potential judges may utilize. The reason for this is that the measure of discrepancy is sensitive to the scaling that the potential judges would be using in assigning their scores, $s(\mathbf{i})$. For instance, two judges assigning scores in the same order pattern, but with one using a narrow range of scores and the other using a large range of scores, would have different degrees of discrepancies although their scores would have the same order pattern. There are several techniques for standardizing the scores to take into account the variability of the potential judges' scores. In the example presented in this section, we consider the following approach in order to avoid this scaling problem. When asking a potential judge to assign scores to T orderings, we require that the judge use one of T scores pre-specified by the evaluator. Furthermore, once one of these pre-specified scores is used by a potential judge that score is removed from further possible usage by that judge. In application, the judge would be given a box of T chips with percentages marked on the chips and the judge would pick and assign one of these chips to each of the presented T orderings. Moreover, we would allow the judge to see all T orderings before assigning the prescribed scores.

For our example, we consider S_8 and judiciously select a subset $\mathcal{S}$ consisting of 30 permutations chosen with respect to the b_3-ordering. A schematic of these 30 permutations as well as descriptions of each of their immediate predecessors in $\mathcal{S}$ is given in Table 6.1 and in Hasse diagram format in Table 6.2. Intuitively, one can describe the choice of these 30 permutations or reorderings as being along four "strings" with each string beginning at the perfect order and ending at the complete reverse order. Moreover, there are levels along each of these strings, where these levels correspond to the inversion number of each reordering. On each of the four strings there is one reordering or permutation at each level or inversion number.

Motivated by the structure of these 30 reorderings, the preassigned scores we allow the judges to chose from are the following: 1, 2, 2, 2, 2, 3, 3, 3, 3, 4,. . .,4,. . .,8, 8, 8, 8, 9. These thirty scores can be viewed as one score of 10%, four scores of 20%, . . ., four scores of 80%, and one score of 90%.

Our example concerns two types of potential judges, "good" judges and "bad" judges. The good judges have differing levels of variability by which they assign their scores, as described below, and the bad judge is one without any discriminating ability. Specifically we simulated the bad judge's chosing of the scores by randomly assigning the 30 scores $1, \ldots, 9$ to each of the thirty orderings under consideration. The good judges were simulated in the following manner. To each of the thirty orderings depicted in Table 6.1 we assigned a random variable by adding a normal error quantity to the inversion number assigned to each permutation. The normal error had mean 0, and three possible standard deviations, $\sigma = 1$, 1.5 and 2. For example, to the permutation 36241857 the assigned random quantity was $10 + \epsilon$, where ϵ is distributed according to a normal with mean 0 and standard deviation σ. For each of the three chosen values of σ, we then preceded in the following manner. We rank ordered these thirty random quantities. The permutation corresponding to the highest ranked random quantity was assigned the score of 9, the next highest four ranked random variables were assigned the score of $8, \ldots$, and the lowest random quantity was assigned the score of 1. Tables 6.3 b), c), and d) provide outcomes of a single simulation of assigned scores corresponding, respectively, to σ =1, 1.5, and 2, along with the isotonized version (s^*) of these scores. The format of the Table corresponds to the diagramatic structure of the thirty chosen permutations. Also given for these three tables is the discrepancy measure. Not surprisingly the discrepancy measure increases as σ increases. An example of a score assignment for the bad potential judge is given in Table 6.3a, along with its isotonized version and the corresponding discrepancy score.

As a matter of interest, we simulated 10,000 assignments of random scores, i.e., bad judge's scores. Table 6.4 provides a frequency histogram of the 10,000 discrepancy measures corresponding to these 10,000 simulations. Another viewpoint of this histogram is seeing it as the null hypothesis distribution corresponding to a totally uninformed potential judge. According to this distribution the three good potential judges are obviously seen as "nonguessers."

Appendix
PROGRAM IBCRb

A program for finding isotonic regressions on a set of permutations

NAME: IBCRb.EXE

LANGUAGE: C

DESCRIPTION AND PURPOSE:

Positive dependence orderings have been studied extensively in recent years. Block *et al* (1990) pointed out that many dependence orderings can be modeled using partial orderings on a set of permutations. These orderings are called the b_1, b_2, b_3 and b_4 orderings. Here we present an algorithm for calculating the least squares regression function which is restricted to be isotonic with respect to one of the b_t orderings on the permutation set S_n.

THEORY: See Sections 2 and 3 in this paper.

The IBCRb algorithm generates the permutation set S_n with a partition by inversion numbers for a given order. Then for each b_t-ordering, it finds immediate predecessors for each element of S_n. If a function $g(x)$ on S_n is given with weights w, the IBCRb algorithm provides the solution to the following problem:

$$\min \sum_{x \in S_n} (g(x) - f(x))^2 w(x) \qquad \text{subject to} \qquad f \in I_t,$$

where I_t is the class of all functions on S_n such that $\mathbf{i} \geq_t \mathbf{j}$ implies $f(\mathbf{i}) \geq f(\mathbf{j})$, for $t = 1, 2, 3$ or 4. In the current program IBCRb, the function $g(x)$ is defined as

$$g(x) = \sum_{k=1}^{n} |x(k) - k| \qquad \text{for each } x \in S_n .$$

The function generating $g(x)$ is implemented as function gef( ). Users can easily change the function gef( ) for other definitions of $g(x)$.

SYNTAX: ibcrb outputfile

INPUT:

There is only one input data value, nn, the order of the permutation. After the screen displays "Enter the order of permutation," use the keyboard to input the integer; then press the return key.

SCREEN OUTPUT:

After entering the order of the permutations, "*The order of permutations is nn.*" is displayed on the screen.

After finishing the calculation of the four isotonic regressions, it prints "*Success.*" on the screen.

FILE OUTPUT:

1. Accumulation number of elements in each level.
2. The permutations with codes in each level.
3. Four tables. Each table is for a b_t-ordering, and has more than 4 columns.

    Column 1: code of each permutation;

    Column 2: the original function $g(x)$;

    Column 3: the weight functions $w(x)$;

    Column 4: the isotonic regression $g^*(x)$ of g with weights w;

    Column 5 and up: the codes of immediate predecessors of each permutation.

Table 6.1

	Permutations	Inversion Numbers	Immediate Predecessors
0	87654321	28	
1	86745231	25	0
2	87634521	25	0
3	86475321	25	0
4	68745321	25	0
5	86452713	21	1 3
6	86345721	21	2 3
7	68347521	21	2 3 4
8	68453271	21	3 4
9	82645173	17	5
10	86234571	17	1 6
11	63824751	17	7
12	36845271	17	1 6 7 8
13	26458173	13	4 9
14	82631457	13	10
15	63248517	13	6 8 11
16	36814527	13	12
17	26145873	10	13
18	26381457	10	8 11 14
19	36241857	10	12 15
20	13684527	10	16
21	26143587	7	8 17
22	21638457	7	9 18
23	32416857	7	19
24	13645287	7	20
25	21436587	4	21
26	12364857	4	19 20 22
27	23416578	4	10 23
28	13456278	4	24
29	12345678	0	25 26 27 28

Table 6.2
Hasse Diagram Corresponding To Table 6.1

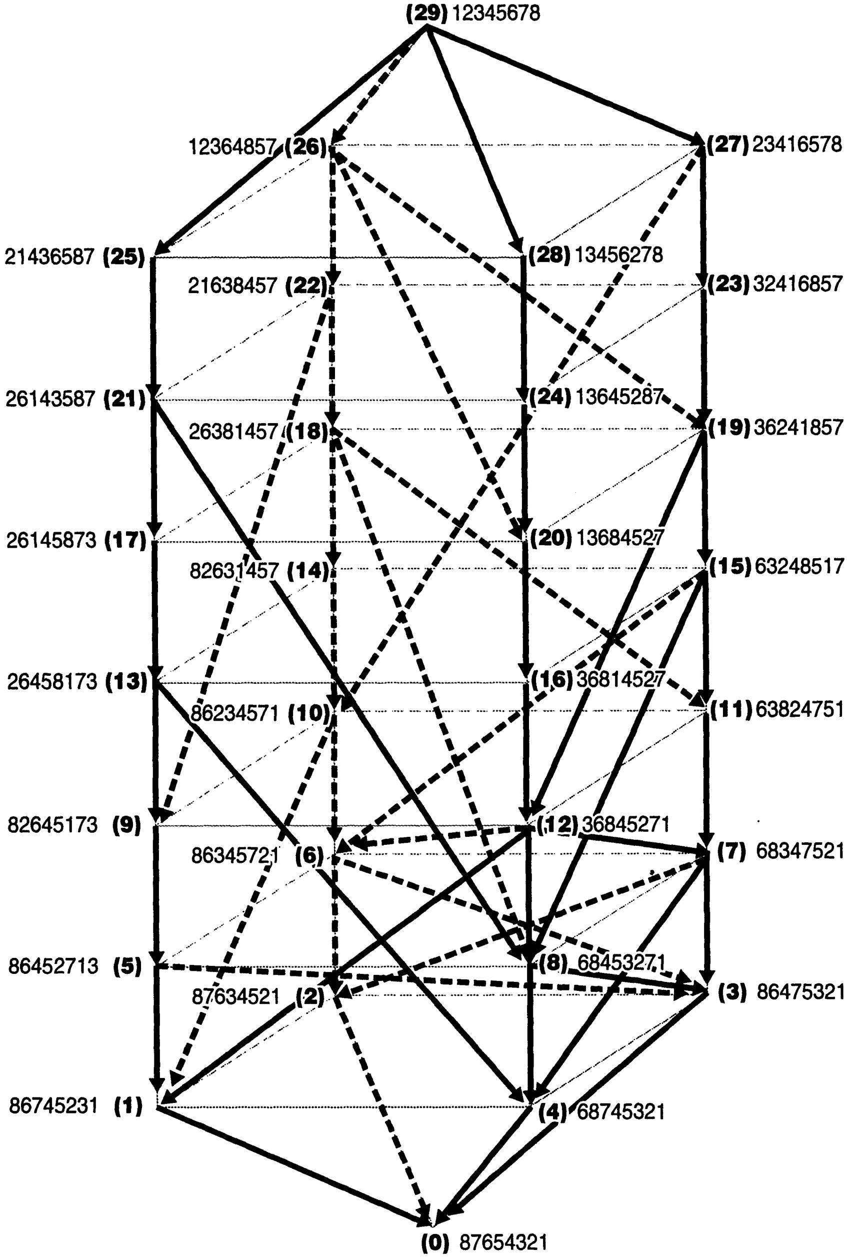


Table 6.3

$s(x)$: original score

$s^*(x)$: the isotonic regression of $g(x)$

a) "Bad" Potential Judge

$s(x)$				$s^*(x)$			
	7				7		
4	6	3	2	$5\frac{2}{3}$	6	$5\frac{1}{6}$	$5\frac{1}{2}$
7	6	4	9	$5\frac{2}{3}$	6	$5\frac{1}{6}$	$5\frac{1}{2}$
6	6	3	3	$5\frac{2}{3}$	6	$5\frac{1}{6}$	$5\frac{1}{6}$
1	5	8	5	$4\frac{1}{3}$	$5\frac{1}{6}$	$5\frac{1}{6}$	$5\frac{1}{6}$
2	4	4	8	$4\frac{1}{3}$	$5\frac{1}{6}$	$5\frac{1}{6}$	$5\frac{1}{6}$
8	7	8	3	$4\frac{1}{3}$	$5\frac{1}{6}$	$5\frac{1}{6}$	$4\frac{1}{3}$
5	5	2	7	$4\frac{1}{3}$	5	2	$4\frac{1}{3}$
	2				2		

$$\Sigma(s(x) - s^*(x))^2 = 114\ \tfrac{1}{6}$$

(b) "Good" Potential Judge $(\sigma = 1)$

$s(x)$				$s^*(x)$			
	9				9		
8	8	7	8	8	8	7	8
7	8	6	6	7	8	6	6
5	7	6	4	$5\frac{1}{3}$	7	6	$5\frac{1}{3}$
5	4	5	5	$5\frac{1}{3}$	4	5	$5\frac{1}{3}$
6	4	4	7	$5\frac{1}{3}$	4	4	$5\frac{1}{3}$
2	3	3	2	$2\frac{1}{2}$	3	3	$2\frac{1}{2}$
3	2	2	3	$2\frac{1}{2}$	2	2	$2\frac{1}{2}$
	1				1		

$$\Sigma(s(x) - s^*(x))^2 = 6\ \tfrac{1}{3}$$

(c) "Good" Potential Judge ($\sigma = 1.5$)

$s(x)$				$s^*(x)$			
	7				8		
7	7	9	5	$7\frac{1}{2}$	$7\frac{2}{3}$	8	$6\frac{2}{3}$
8	8	6	8	$7\frac{1}{2}$	$7\frac{2}{3}$	6	$6\frac{2}{3}$
6	8	5	7	6	$7\frac{2}{3}$	5	$6\frac{2}{3}$
6	5	5	6	6	5	5	6
2	4	3	3	3	4	3	3
4	4	2	3	3	4	3	3
2	4	3	2	2	3	3	2
	1				1		

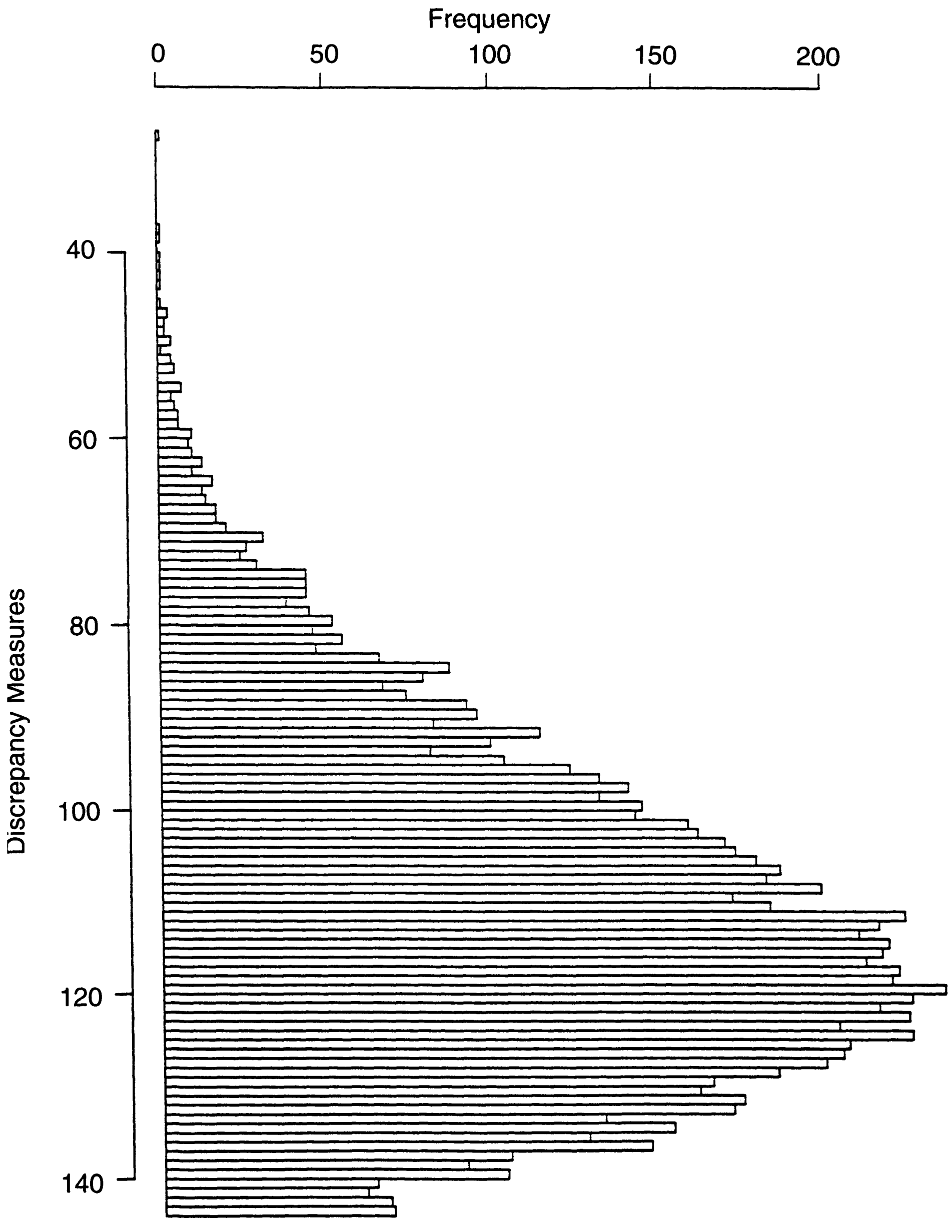
$$\Sigma(s(x) - s^*(x))^2 = 11\ \tfrac{5}{6}$$

(d) "Good" Potential Judge ($\sigma = 2$)

$s(x)$				$s^*(x)$			
	9				9		
6	8	8	7	7	8	8	7
8	6	4	7	7	7	$5\frac{1}{3}$	7
7	7	6	4	7	7	$5\frac{1}{3}$	4
5	8	6	4	5	7	$5\frac{1}{3}$	4
5	2	1	4	5	3	3	4
3	2	5	2	3	3	3	$2\frac{1}{2}$
3	5	3	2	3	3	$2\frac{1}{2}$	$2\frac{1}{2}$
	3				$2\frac{1}{2}$		

$$\Sigma(s(x) - s^*(x))^2 = 21\ \tfrac{2}{3}$$

Table 6.4
Frequency Histogram of 10,000 Discrepancy Measures



References

BLOCK, H. W., CHHETRY, D., FANG, Z. and SAMPSON, A. (1993). Metrics on permutations based on partial orderings. University of Pittsburgh, Technical Report 87-11.

BLOCK, H. W., CHHETRY, D., FANG, Z. and SAMPSON, A. (1990). Partial orders on permutations and dependence orderings on bivariate empirical distributions. *Ann. Statist.* **18** , 1840–50.

BLOCK, H. W., QIAN, S. and SAMPSON, A. (1992). Structure algorithms for isotonic regression. University of Pittsburgh, Technical Report.

BLOCK, H. W., QIAN, S. and SAMPSON, A. (1994). Structure algorithms for partially ordered isotonic regression. *J. Comput. Graph. Statist.* **3**, 285–300.

ROBERTSON, T., WRIGHT, F. T. and DYKSTRA, R. L. (1988). *Order Restricted Statistical Inference.* J. Wiley & Sons, Chichester.

YANAGIMOTO, T. and OKAMOTO, M. (1969). Partial orderings of permutations and monotonicity of rank correlation statistics. *Ann. Inst. Statist. Math.* **21**, 489–506.

DEPARTMENT OF MATHEMATICS AND STATISTICS
UNIVERSITY OF PITTSBURGH
PITTSBURGH, PA 15260