Indexing (n, s)-combinations

Paul E. Dunne

February 24, 2015

Abstract

In this paper, a method to index an ordered set of length s from a set of n different integers (that are not necessarily in sequence order), is detailed. Additionally the inverse method is given, that converts an index into an ordered set.

1 Converting an Ordered Set into an Index

Given $(\underline{n}, s)$ with $1 \leq s \leq |\underline{n}| = n$, and an ordered set $\underline{C} = \langle c_1, c_2, \ldots, c_s \rangle$ of s elements from an ordered set $\underline{n} = \{ag_1, ag_2, ag_3, \ldots, ag_n\}$, where $c_i < c_{i+1}$ for each $1 \leq i < s$ and $ag_j < ag_{j+1}$ for each $1 \leq j < |\underline{n}|$, we can define:

$$\underline{C} = \langle c_1, c_2, ..., c_s \rangle \text{ where } c_i = j \text{ if } c_i = ag_j$$

Additionally we can define a bijective mapping:

$$posn(\underline{C}, \underline{n}, s) \quad \leftrightarrow \quad \left[1, 2, \ldots, \left(\begin{array}{c} n \\ s \end{array} \right)\right]$$

In this mapping $posn(\underline{C}, \underline{n}, s)$ is the index of $\underline{C}$ under the lexicographic ordering of s-tuples from $\{ag_1, ag_2, ag_3 \ldots, ag_n\}$. That is the ordering $<_{lex}$ under which

$$\langle c_1, c_2, \ldots, c_s \rangle <_{lex} \langle d_1, \ldots, d_s \rangle \ \text{ if } \ \left\{ \begin{array}{ll} c_1 < d_1 & \text{or} \\ c_1 = d_1 & \text{and } \langle c_1, c_2, \ldots, c_s \rangle <_{lex} \langle d_2, \ldots, d_s \rangle \end{array} \right.$$

To map $\underline{C} = \langle c_1, c_2, \ldots, c_s \rangle$ to its correct position we simply extend the observation that if $c_1 = 1$ then under lexicographic ordering, $\underline{C}$ must be among the first $\left(\begin{array}{c} n-1 \\ s-1 \end{array} \right)$ elements. In general, if $c_1 = m$, then $\underline{C}$ occurs *after* all choices in which $c_1 < m$, so that

$$\sum_{r=1}^{m-1} \left(\begin{array}{c} n-r \\ s-1 \end{array} \right) \quad < \quad posn(\underline{C}, n, s) \quad \leq \quad \sum_{r=1}^{m} \left(\begin{array}{c} n-r \\ s-1 \end{array} \right)$$

In total, noting that $posn(\langle ag_i \rangle, \underline{n}, 1) = i$, we obtain

$$posn(\langle c_1, c_2, \ldots, c_s \rangle, \underline{n}, s) \quad = \quad \sum_{i=1}^{s} \sum_{r=1}^{c_i - \left(\sum_{j=1}^{i-1} c_j\right) - 1} \left(\begin{array}{c} n - \left(\sum_{j=1}^{i-1} c_j\right) - r \\ s-i \end{array} \right)$$

2 Converting an Index into an Ordered Set

To compute $posn^{-1}(m, \underline{n}, s)$, that is the s-tuple occurring at position m within the lexicographic ordering, the process is, in effect, "reversed".

Given $(m, \underline{n}, s)$:

1. Find the *maximum* value, t, for which

$$m \;>\; \sum_{r=1}^{t-1} \binom{n-r}{s-1}$$

2. $c_1 \;:=\; t$;
3. $$\langle d_1, \ldots, d_{s-1} \rangle \;:=\; posn^{-1}\left(m - \sum_{r=1}^{c_1-1} \binom{n-r}{s-1}, n - c_1, s-1 \right)$$
4. $\underline{e} = \langle e_1, e_2, ..., e_s \rangle = \langle c_1, d_1 + c_1, \ldots, d_{s-1} + c_1 \rangle$
5. return $\langle ag_{e_1}, ag_{e_2}, \ldots, ag_{e_s} \rangle$