1 The distribution

I want to focus on a particular distribution on $[k]^n$ which I'll ultimately call the "coalescence distribution", but which for now I'll call the "uniform ordered k-partition distribution". It is the same as the "equal-slices" distribution on $[k]^n$, except that we throw out the "degenerate slices" — those that are completely missing at least one character from $[k]$. (Hence $n \geq k$ is necessary.) I would also like to henceforth declare a combinatorial line in $[k]^n$ to be "degenerate" if its fixed coordinates fail to include at least one of each character in $[k]$. This is because it's convenient to think of a line in $[k]^n$ via its "template" in $([k] \cup \{\star\})^n$, and I would like to exclude the degenerate slices in $([k] \cup \{\star\})^n$. I believe that it's trivial to show that the usual DHJ(k) implies DHJ(k) with this new notion of degenerate lines.

There are many equivalent ways to think about this "uniform ordered k-partition distribution". One we've already mentioned: you pick a "nondegenerate slice" of $[k]^n$ uniformly at random, and then you draw uniformly from that slice. More concretely, recall that a "k-composition" of the number n is a way of writing it as $n = a_1 + \cdots + a_k$, where each a_i is a positive integer. There are precisely $\binom{n-1}{k-1}$ k-compositions of n, and choosing one of these randomly is what we mean by choosing a "uniform nondegenerate slice". To see that there are indeed $\binom{n-1}{k-1}$ k-compositions, think of n balls being lined up. There are $n-1$ "slots" between them. The k-compositions are in 1-1 correspondence with the ways of placing $k-1$ indistinguishable "walls" into the slots (with each slot being able to hold at most one wall). Given a wall-ball arrangement, a_i is the number of balls in the the ith walled-off segment.

Using the same image, we can understand the final distribution on strings in $[k]^n$. Imagine the lined-up balls are labelled, 1 to n. If we first randomly permute the balls and *then* place the walls, we effectively get an "*ordered* k-partition of $[n]$". The word "ordered" here refers to the fact that the order of the k parts is significant; the word "partition" here refers to the fact that the order *within* the parts is not significant. These ordered k-partitions are in 1-1 correspondence with the "nondegenerate" strings in $[k]^n$ — i.e., those strings in which each character from $[k]$ appears at least once. The correspondence is the obvious one: the numbers on the balls correspond to the indices in the string, and the ith walled-off segment corresponds to indices where the string has the character i.

I also like to think of these walls and balls with a different image: I imagine we take all the balls in the ith walled-off segment and package them into a cardboard box, which we label by i. Now the balls in a box are free to roll around, which illustrates that the order within the box is not significant; and, we are free to move the boxes around because the labels on them "save" their position in $[k]$. The reason for this

imagery may become clear when we recurse (as we will do shortly). For the more notationally minded, I also like to think of these configurations as being written as, say,

$$\langle\{2,6,8\},\{1,4,7,9,10\},\{3,5\}\rangle.$$

Here k is 3 and n is 10; we have balls 2, 6, and 8 in the first box, etc.

There's nothing too special about the final number of boxes here, k (except that it's at least n). Indeed, I would like to think about iterating this boxing-up process. Let m be another number between k and n (inclusive). Imagine we choose a uniformly random m-partition of $[n]$; i.e., we randomly pack up the n balls into m boxes as described above. (This is like choosing a random m-dimensional subspace with 0 free coordinates, in our other mode of thinking.) Next, we apply a uniformly random k-partition to these m... "items". I.e., we take the m boxes, line them up in a random order, plunk down $k-1$ walls into the $m-1$ slots, and then pack all the boxes in the ith walled segment into a *bigger* box, labelled i.

Proposition 1 *Doing a random ordered m-partition followed by a random ordered k-partition is "equivalent" to just doing a random ordered k-partition.*

Well, it's not *quite* "equivalent" per se; in the former case you have labelled boxes containing labelled boxes containing labelled balls, whereas in the latter case you just have labelled boxes containing labelled balls. For example, if you were to do a random labelled 2-partition of the labelled 3-partition shown above, you might get

$$\langle\{\{1,4,7,9,10\},\{3,5\}\},\{\{2,6,8\}\}\rangle.$$

Our convention, though, will be to treat "inner" set braces as nonexistent, so the above really means

$$\langle\{1,3,4,5,7,9,10\},\{2,6,8\}\rangle;$$

AKA the string 1211121211. In particular, the "inner" box-labels are not significant; given a "final" big box labelled i we just rip open everything inside it without regard to label, and treat all the balls we find as i's for the string.

Proof: (of Proposition 1.) Let's consider the two-step process: first an $n \to m$ ordered partition and then the $m \to k$ ordered partition. The first observation in the proof is that in the walls/balls imagery, the random permutation of the items and the random placing of the walls *commute*. So we think of the first step as randomly permuting the n balls, randomly placing $m-1$ walls, boxing up the m segments of balls, and then removing the walls. But we think of the *second* step as first placing $k-1$ walls at random in the $m-1$ inter-box slots, *then* randomly permuting the k segments of boxes.

The second observation is that the random *wall-placings* commute. What we're doing is first choosing $m-1$ out of $n-1$ slots at random, then involving some cardboard, then choosing $k-1$ out of the $m-1$ chosen slots at random. But this is clearly equivalent to simply choosing $k-1$ out of $n-1$ slots at random. So the whole process has simplified to permuting the n balls, walling them uniformly at random into k segments, and then permuting the segments. But now we again commute the first permutation and the walling; it's equivalent to first randomly placing $k-1$ walls into $n-1$ slots, then randomly permuting the n balls, then randomly permuting the k segments of balls. Finally, these two permutations can be combined because for *any* fixed k-segmenting, it's clear that doing a random permutation of all the balls and then doing a random partition of the segments is equivalent to just doing the first random partition. $\square$

2 Random coalescences

From now on, for brevity, I would like to call the uniform distribution on ordered k-partitions of $[n]$, the "(k-)coalescence distribution".

For $n \geq m$, define an "$n \to m$ coalescence" κ to be the info needed to recreate a random ordered m-partition of n as described in the previous section; specifically, κ is a permutation π on $[n]$ together with an element of the set $\binom{n-1}{m-1}$. Proposition 1 tells us that if κ_1 is a random $n \to m$ coalescence and κ_2 is a random $m \to k$ coalescence, then $\kappa_2\kappa_1$ is a random $n \to k$ coalescence (i.e., is distributed according to the coalescence distribution). I write it in this order because I think of these as κ's as "operators" on lists of sets; e.g.,

$$\kappa\langle\{1,4,7,9,10\},\{3,5\},\{2,6,8\}\rangle = \langle\{1,3,4,5,7,9,10\},\{2,6,8\}\rangle$$

if κ is the $3 \to 2$ coalescence defined by arranging 3 items in the order $\langle 2,1,3\rangle$ and then placing the wall in the second slot. (Note that this is not the unique κ achieving this transformation; the one that ordered as $\langle 1,2,3\rangle$ and placed the wall in the second slot would do the same.)

These random coalescences are our replacements for random restrictions/subspaces, which worked so nicely with product distributions on $[k]^n$. Proposition 1 says coalescences have the same "independence" property: we can imagine fixing randomly a particular $m \to k$ coalescence κ_0 (this is like fixing an $(n-(m-k))$-dimensional subspace) and then considering the space of random $n \to m$ coalescences κ. The resulting distribution on *strings* $\kappa_0\kappa$ is our desired "k-coalescence distribution". (Note: when we omit the list that a coalescence like $\kappa_0\kappa$ is operating on, it is assumed to be

$\langle\{1\},\{2\},\dots,\{n\}\rangle$.)

Proposition 1 of course extends to doing multiple coalescences: for any $n \geq m_1 \geq m_2 \geq \cdots \geq m_t$ we could make a random $n \to m_1$ coalescence, then a random $m_1 \to m_2$ coalescence, etc., down to a random $m_2 \to m_t$ coalescence, and this would be equivalent to just doing a single random $n \to m_t$ coalescence.

Indeed, if we want we can imagine all $n \to k$ coalescences as occurring via $n-k$ consecutive "single coalescences", $n \to (n-1) \to (n-2) \to \cdots (k+1) \to k$. It's of course natural to think of a random $(m+1) \to m$ coalescence according to which slot *doesn't* get a wall, as opposed to which m slots get the $m-1$ walls. I.e., a single coalescence like this involves randomly permuting the $m+1$ items, then merging two randomly chosen *adjacent* items.

Actually, when we come (finally) to DHJ, it will be useful to think not of merging two random adjacent items, but rather the *last* item with a random other item. This is because we like to identify $[k+1] \cong ([k] \cup \star)$, and we think of a combinatorial line as coming from replacing all the $\star$'s by 1', by 2's, ..., by k's. I.e., we will often think of $k+1$ labeled boxes of indices as a "combinatorial line template", and we can get a random one of the k points on this line by merging the contents of the $(k+1)$th box with the contents of a random box from $1 \dots k$. To this end, we make the following definition:

Definition 2 *Given an ordered set of $m+1$ boxes, a "coagulation" means merging the contents of the box labelled $m+1$ into the jth box, for some $j \in [m]$. A "random coagulation" means doing this with $j \in [m]$ chosen uniform at random.*

The final proposition for this document is that in a single coalescence $(m+1) \to m$, we can do coagulations instead of adjacent-merges. In this proposition we will really start to think of coalescences not so much as ordered partitions, but as sequences of operations which combine to form an "operator":

Proposition 3 *Consider an $(m+1) \to m$ coalescence κ defined by randomly drawing "$\kappa \sim C\Pi$"; this means, "permute" then "coagulate". Then κ is the same as a random $(m+1) \to m$ coalescence with the usual distribution.*

Proof: It's clear that starting with $m+1$ ordered items, the following are all equivalent:

(a) Permuting the $m+1$ items, then merging the $(m+1)$th into the jth, for $j \in [m]$ random.

(b) Choosing two out of $m+1$ items uniformly at random to be merged, choosing which of the m positions the merged pair will go into at random, then placing the other $m-1$ items randomly into the remaining $m-1$ positions.

(c) Permuting the $m+1$ items, then merging a random pair of adjacent items. $\square$

3 Densities, Lines, Subspaces, and Operator Soup

Let's start to introduce more notation. As we saw in the last section, a random single coalescence can be replaced by a random permutation followed by a random coagulation. We also saw that a random $n \to k$ coalescence was equivalent to $n-k$ consecutive independent random single coalescences. We may write this as follows: If we draw $\kappa \sim (C\Pi)^{n-k}$ randomly, then κ is distributed according to the coalescence distribution on $[k]^n$.

I confess this notation is a bit weird, so let me explain it. First, $(C\Pi)^{n-k}$ is shorthand for $C\Pi C\Pi C\Pi \cdots C\Pi$ ($C\Pi$ repeated $n-k$ times). Second, as before Π stands for permutation and C for coagulation. Saying $\kappa \sim (C\Pi)^{n-k}$ is saying that κ is formed by:

(i) First doing a random permutation of the n indices (again, when the "operator" κ is considered without an operand, it's assumed to be $\langle\{1\},\dots,\{n\}\rangle$);

(ii) then doing a random coagulation (merging the nth index in the permutation into a box with the jth, for $j \in [n-1]$ uniformly random);

(iii) then doing a random permutation of the resulting $n-1$ items;

(iv) then doing a random coagulation, etc.

In the end we get a string-in-$[k]^n$/ordered-k-partition with the desired distribution.

We now proceed with some additional notation. Since we are working on DHJ(k), we will typically have some set $A \subseteq [k]^n$ or perhaps some function $f : [k]^n \to \mathbb{R}$. Let κ_0 be some fixed $n \to m$ coalescence. We will write $f_{\kappa_0} : [k]^m \to \mathbb{R}$ for the function on $m \to k$ coalescences κ (i.e., strings in $[k]^m$) defined by $f_{\kappa_0}(\kappa) = f(\kappa\kappa_0)$. If f is the indicator of the set A, we'll write A_{κ_0} for the subset which f_{κ_0} indicates; i.e., the set of κ such that $\kappa\kappa_0 \in A$.

Let's discuss DHJ(k) in this notation. The "ur"-density for us is the standard coalescence density. In particular, we'll usually start by assuming

$$\mathop{\mathbf{E}}_{\kappa\sim(C\Pi)^{n-k}}[f] \geq \delta.$$

(If you really want to start with the assumption that the uniform density of A is at least δ, you can do it via our standard method of passing to a subspace of dimension $o(\delta\sqrt{n})$.) Now what about the conclusion, that A contains a combinatorial line? As mentioned before, we can think of the "template" of combinatorial line as some string in $[k+1]^n \cong (k \cup \{\star\})^n$; i.e., some $n \to (k+1)$ coalescence λ. Further, coagulations are defined so that a random coagulation of λ gives a random point on the associated line (uniformly, among the k). Hence,

$$\mathop{\mathbf{E}}_{\kappa\sim C}[f_\lambda(\kappa)] = \Pr_{\kappa\sim C}[\kappa\lambda \in A] = \text{fraction of points on the line } \lambda \text{ in } A.$$

In particular, the entire line is in A iff this quantity is 1 iff this quantity is $> 1 - 1/k$.

4 Some theorems

Finally, with all the notation set up, we can prove some theorems. Throughout, we use the following **convention:** If we write $\mathbf{E}_\kappa$ and the distribution κ is drawn from is omitted, it is assumed to be the standard k-coalescence distribution on the appropriate number of input items.

Let f be the indicator function of some set A.

Statement of DHJ(k): If $\mathbf{E}_\kappa[f(\kappa)] \geq \delta_1$ and $n \geq M_1 = M_1(\delta_1, k)$, then there exists a combinatorial line in A; i.e., with *positive* probability (albeit potentially depending on n) over the choice of $\lambda \sim (C\Pi)^{n-(k+1)}$ we have $\mathbf{E}_{\kappa\sim C}[f_\lambda(\kappa)] = 1$.

Statement of Varnavides-DHJ(k): If $\mathbf{E}_\kappa[f(\kappa)] \geq \delta_1$ and $n \geq M_2 = M_1(\delta_1/2, k)$, then with probability at least $\delta_2(M_2, \delta_1, k) = \delta_1/(2{M_2}^{k/2}(k+1)^{M_2})$ over the choice of the line $\lambda \sim (C\Pi)^{n-(k+1)}$, we have $\mathbf{E}_{\kappa\sim C}[f_\lambda(\kappa)] = 1$.

Proof of DHJ(k) $\Rightarrow$ Varnavides-DHJ(k): Let $\kappa_1 \sim (C\Pi)^{n-M_2}$ and let $\kappa_2 \sim (C\Pi)^{M_2-k}$. Since $\kappa_2\kappa_1$ is the standard coalescence distribution, we have $\mathbf{E}_{\kappa_1}\mathbf{E}_{\kappa_2}[f_{\kappa_1}(\kappa_2)] \geq \delta_1$. Hence for at least a $\delta_1/2$ fraction of κ_1's (call them the "good" κ_1's) we have $\mathbf{E}_{\kappa_2}[f_{\kappa_1}(\kappa_2)] \geq \delta_1/2$. Recall that $f_{\kappa_1} : [k]^{M_2} \to \{0,1\}$ denotes the indicator of A_{κ_1}. By DHJ(k) then, for each good κ_1 there exists some combinatorial line in A_{κ_1}; hence if we choose a *random* line $\lambda_2 \sim (C\Pi)^{M_2-(k+1)}$ it will be in A_{κ_1} with probability at least some quantity depending only on M_2 and k. This quantity is the least probability of any outcome of λ_2, which I believe is something like $1/(M_2^{k+2}(k+1)^{M_2})$, although I'll do the counting later. In any case, we've established that if we choose

$\kappa_1 \sim (C\Pi)^{n-M_2}$ and $\lambda_2 \sim (C\Pi)^{M_2-(k+1)}$, then $\mathbf{E}_{\kappa\sim C}[f_{\lambda_2\kappa_1}(\kappa)] = 1$ with probability at least $(\delta_1/2) \cdot (1/(M_2^{k+2}(k+1)^{M_2}))$, completing the proof. $\square$