Interpolation Jack Polynomials

Group 2: Havi Ellers and Xiaomin Li
Mentor: Dr. Hadi Salmasian

Fields Institute for Research in Mathematical Sciences

August 28, 2019

Outline

Outline

Interpolation Jack Polynomials

Interpolation Jack Polynomials

Interpolation Jack polynomials are certain polynomials P_λ, indexed by partitions λ, in n variables $x_1, \dots, x_n$ and with coefficients in the field $\mathbb{Q}(k)$.

Example: when $n = 3$

The interpolation Jack polynomial associated with the partition $\lambda = (2, 0, 0)$ is

$$P_\lambda(x, y, z) = x^2 + y^2 + z^2 + \left(\frac{2k}{k+1}\right) xy + \left(\frac{2k}{k+1}\right) yz + \left(\frac{2k}{k+1}\right) xz - x - \left(\frac{3k+1}{k+1}\right) y - \left(\frac{5k+1}{k+1}\right) z$$

Interpolation Jack Polynomials

Where do these polynomials come from?

$$P_\lambda(x_1, \ldots, x_n) = \sum_{\substack{T \text{ a reverse} \\ \text{tableau} \\ \text{of shape } \lambda}} \psi_T(k) \prod_{s \in T} (x_{T(s)} - a'(s) + l'(s)k)$$

Interpolation Jack Polynomials

Why are these polynomials interesting?

- If you set $k = 1$, you get a *factorial Schur polynomial*.
- The highest degree homogeneous part of a factorial Schur polynomial is a regular Schur polynomial.
- The Schur polynomial associated with the partition $\lambda = (\lambda_1, \ldots, \lambda_n)$ is defined to be

$$S_\lambda = \frac{\det\left(x_i^{\lambda_j + n - j}\right)}{\det\left(x_i^{n-j}\right)}$$

These are well-studied polynomials and have connections to the representation theory of $\mathrm{gl}(n, \mathbb{C})$.

Our Goal:

We would like to find rational functions of k, $c_\mu^\lambda(k)$, such that

$$\frac{\partial}{\partial k} P_\lambda(x, y, z) = \sum_\mu c_\mu^\lambda(k) P_\mu(x, y, z)$$

where the sum ranges all partitions μ such that $|\mu| \leq |\lambda|$.

Example

For $\lambda = (2,0,0)$ we can write:

$$\frac{\partial}{\partial k} P_{(2,0,0)} = \left(\frac{5k+3}{k+1}\right) P_{(0,0,0)} + \left(\frac{-6k-4}{k+1}\right) P_{(1,0,0)} + \left(\frac{2}{k^2+2k+1}\right) P_{(1,1,0)}$$

Outline

The Case $\lambda = (0, 0, 0)$

If $\lambda = (0, 0, 0)$ then $\frac{\partial}{\partial k} P_\lambda = 0$, so all coefficients are zero.

The Case $\lambda = (D_1, D_2, D_3)$

$\lambda = (D_1, D_2, D_3)$

If $\lambda = (D_1, D_2, D_3)$ where $D_1, D_2, D_3 \neq 0$, then

$$c_{(\mu_1,\mu_2,\mu_3)}^{(D_1,D_2,D_3)} = \begin{cases} 0 & \text{if } \mu_3 - D_3 < 0 \\ c_{(\mu_1-D_3,\mu_2-D_3,\mu_3-D_3)}^{(D_1-D_3,D_2-D_3,0)} & \text{otherwise} \end{cases}$$

So we only need to look at λ with at most two non-zero parts!

The Case $\lambda = (D_1, 0, 0)$

For $\lambda = (D_1, 0, 0)$:

- Complete characterization!
- Formula for the coefficient of P_μ depends on the number of non-zero parts in μ.

Falling Factorial

Notational note: Since the formulae often involve products of consecutive terms, we give the following definition to make the formulae more succinct.

Falling Factorial

Define the *falling factorial notation* for $b > 0$:

$$a^{\overline{b}} := (a+0)(a+1)\cdots(a+b-1)$$

Remark:

- In our research, a is often a polynomial of k, such as $k+\mu_1$, $2k+\mu_1$, etc.
- Notice that a is the "start" of the consecutive product, and b is the number of terms in this product.

Example:

$$(2k+\mu_1)^{\overline{D-\mu_1}} := (2k+\mu_1)(2k+\mu_1+1)\cdots(2k+D-1)$$

The Case $\lambda = (D_1, 0, 0)$

Formula 1: Conjecture for the coefficient of $(\mu_1, 0, 0)$ in $(D_1, 0, 0)$, where $\mu_1 \geq 0$:

$$\frac{(-1)^{D_1-\mu_1} \cdot \frac{D_1!}{(D_1-\mu_1)\ \mu_1!} \times [(2k+\mu_1)^{\overline{D_1-\mu_1}} + (k+\mu_1)^{\overline{D_1-\mu_1}}]}{(k+\mu_1)^{\overline{D_1-\mu_1}}}$$

Formula 2: Conjecture for the coefficient of $(\mu_1, \mu_2, 0)$ in $(D_1, 0, 0)$, where $\mu_1, \mu_2 > 0$:

$$\frac{(-1)^{D_1-\mu_1+\mu_2} \cdot \frac{D_1!\ (D_1-\mu_1-1)!}{(D_1-\mu_1-\mu_2)!\ (\mu_1-\mu_2)!\ \mu_2!}}{(k+\mu_1-\mu_2+1)^{\overline{\mu_2}} \cdot (k+D_1-\mu_2)^{\overline{\mu_2}}}$$

The Case $\lambda = (D_1, 0, 0)$

Formula 3: Conjecture for the coefficient of (μ_1, μ_2, μ_3) in $(D_1, 0, 0)$, where $\mu_1, \mu_2, \mu_3 > 0$:

$$\frac{(-1)^{D_1-\mu_1+\mu_2} \cdot \frac{D_1!(\mu_2-1)!}{(\mu_1-\mu_2)!(\mu_2-\mu_3)!(\mu_3)!}(k-\mu_3+1)^{\overline{\mu_3}} \cdot (k+\mu_1-\mu_3+1)^{\overline{\mu_3-1}} \times S}{(k+\mu_1-\mu_2+1)^{\overline{D_1-\mu_1+\mu_2-1}} \cdot (k+\mu_2-\mu_3+1)^{\overline{\mu_3}} \cdot (2k+\mu_1-\mu_3+1)^{\overline{\mu_3}}}$$

where S is

$$\sum_{j=0}^{D_1-\mu_1-\mu_2-\mu_3} \binom{D_1-\mu_1-\mu_2-1-j}{\mu_3-1}\binom{j+\mu_2-1}{\mu_2-1}(2k+\mu_1)^{\overline{D_1-\mu_1-\mu_2-\mu_3-j}} \cdot (k+D_1-\mu_2-j)^{\overline{j}}$$

The Case $\lambda = (D_1, D_2, 0)$

For $\lambda = (D_1, D_2, 0)$:

- Partial characterization.
- Formula for the coefficient of P_μ depends on the number of non-zero parts in μ.

The Case $\lambda = (D_1, D_2, 0)$

Formula 4: Conjecture for the coefficient of $(\mu_1, 0, 0)$ in $(D_1, D_2, 0)$:
If $D_1 - \mu_1 > 0$:

$$\frac{(-1)^{D_1-\mu_1}\left[\frac{(D_1-D_2)!}{\mu_1!}\right][(D_2-1)!]\,[(k-1)(k)]\,[(2k+\mu_1)^{\overline{D_1-\mu_1-D_2}}]\cdot S}{(k+\mu_1-D_2)^{\overline{D_1-\mu_1}}\cdot(2k+D_1-D_2)^{\overline{D_2-(D_1-\mu_1)}}}$$

where S is a polynomial of degree $2(D_2-1)$ with:

$$\begin{aligned}\text{leading term } &= \binom{D_1-\mu_1-1}{D_2-1}k^{2(D_2-1)}\\ \text{constant term } &= (-1)_2^{D_2-1}\cdot(\mu_1-D_2+1)^{\overline{D_2-1}}\cdot(D_1-D_2+1)^{\overline{D_2-1}}\end{aligned}$$

If $D_1 - \mu_1 = 0$:

$$(-1)^{D_2}\cdot(D_2-1)!$$

The Case $\lambda = (D_1, D_2, 0)$

Formula 5: Conjecture for the coefficient of $(\mu_1, \mu_2, 0)$ in $(D_1, D_2, 0)$ where $\mu_1, \mu_2 > 0$:
If $\mu_2 > D_2$:

$$\frac{(-1)^{D_1+D_2-\mu_1+\mu_2} \cdot \frac{(D_1-D_2)!\ (D_1-\mu_1-1)!}{(\mu_1-\mu_2)!\ (\mu_2-D_2)!\ (D_1+D_2-\mu_1-\mu_2)!}}{(k+\mu_1-\mu_2+1)^{\overline{\mu_2-D_2}} \cdot (k+D_1-\mu_2)^{\overline{\mu_2-D_2}}}$$

Outline

Overview

We wrote all our code in Sagemath, occasionally suplemented by mathematica. There were three main parts of our code:

- Generating the polynomials
  - Construction of polynomials is completely algorithmic
  - Main function "getAnswer(λ, plot)" outputs the polynomial for λ.
- Computing the linear combination
  - Also algorithmic
  - Main function "getLinearComb(P_λ)" outputs the linear combination for $\frac{\partial}{\partial k} P_\lambda$
- Various helper functions to help us find the formulae

The Polynomials

Step 1: Generate the polynomials

Main function is a nested function of about 25 auxiliary functions. We used tuples to represent partitions and lists of lists to represent tableaux.

- *fill(tList, unfilledPositions, allFilledLists)*: Pop one box from "unfilledPositions", call *fillEntry()* to fill this box with one of 1,2, or 3 (if possible) and recurse.
- *getAllReverseT(allLambda)*: Return tableaux of all possible reverse fillings of shape "allLambda".
- *getPHelper(allFilledLists, plot)*: Return final polynomial P_λ by adding the polynomials for all reverse fillings in "allFilledLists".
- *getAnswer(allLambda, plot)*: Given a valid partition "allLambda" $= \lambda = (D_1, D_2, D_3)$, return the polynomial P_λ by calling *getPHelper()*.

The Linear Combination

Step 2: Compute the linear combination

Pseudo Code for *getLinearComb(P)*:

Algorithm 1 Get the Linear Combination for $\frac{\partial}{\partial k} P_\lambda$

$dP \leftarrow$ *partial derivative of* P_λ *with respect to k*
linComb $= \emptyset$
while $dP \neq 0$ **do**
    $lm \leftarrow$ *leading monomial of dP*
    $lc \leftarrow$ *leading coefficient of dP*
    $u1, u2, u3 \leftarrow$ *degree of x,y,z in lm respectively*

    $Pu \leftarrow$ *get Pu for* $u = (u1, u2, u3)$
    $PuCoef \leftarrow$ *coefficient of lm in Pu*
    coef $= (lc \,/\, PuCoef)$
    $dP \leftarrow dP - coef * Pu$

    linComb.append(coef, $(u1, u2, u3)$*)*
return linComb

Bugs! (I.e. Comedic Relief)

1. With our first version of code, this algorithm never terminated! Fix:

$$P_\lambda = \sum_T \prod_{s \in T} \psi_T(k)(\dots) \quad \rightarrow \quad P_\lambda = \sum_T \psi_T(k) \prod_{s \in T} (\dots)$$

2. Then we thought every linear combination contained at most 41 polynomials!
$\rightarrow$ Our code had been written to terminate after 40 steps.

Outline

Split Into Cases

Technique 1: Split into cases.
Example: Coefficient of $(1, 0, 0)$ in $(D, 1, 0)$:

$$(2,1,0) \to -(k-1)$$

$$(3,1,0) \to \frac{2(k-1)(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-12(k-1)(2k+1)}{k+2}$$

$$(5,1,0) \to \frac{48(k-1)(2k+1)(2k+3)}{(k+3)(k+2)}$$

$$(6,1,0) \to \frac{-480(k-1)(2k+1)(2k+3)}{(k+4)(k+3)}$$

$$\vdots$$

Notice Common Factors

Technique 2: Notice common factors.
Example: Coefficient of $(1, 0, 0)$ in $(D, 1, 0)$:

$$(2,1,0) \to -(k-1)$$

$$(3,1,0) \to \frac{2(k-1)(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-12(k-1)(2k+1)}{k+2}$$

$$(5,1,0) \to \frac{48(k-1)(2k+1)(2k+3)}{(k+3)(k+2)}$$

$$(6,1,0) \to \frac{-480(k-1)(2k+1)(2k+3)}{(k+4)(k+3)}$$

Get Rid of Common Factors

Technique 2: Notice common factors.
Example: Coefficient of $(1,0,0)$ in $(D,1,0)$:

$$(2,1,0) \to -1$$

$$(3,1,0) \to \frac{2(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-12(2k+1)}{k+2}$$

$$(5,1,0) \to \frac{48(2k+1)(2k+3)}{(k+3)(k+2)}$$

$$(6,1,0) \to \frac{-480(2k+1)(2k+3)}{(k+4)(k+3)}$$

Add Factors to Numerator and Denominator

Technique 3: Add factors to numerator and denominator
Example: Coefficient of $(1, 0, 0)$ in $(D, 1, 0)$:

$$(2,1,0) \to -1$$

$$(3,1,0) \to \frac{2(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-12(2k+1)\color{red}{(k+1)}}{(k+2)\color{red}{(k+1)}}$$

$$(5,1,0) \to \frac{48(2k+1)(2k+3)\color{red}{(k+1)}}{(k+3)(k+2)\color{red}{(k+1)}}$$

$$(6,1,0) \to \frac{-480(2k+1)(2k+3)\color{red}{(k+2)(k+1)}}{(k+4)(k+3)\color{red}{(k+2)(k+1)}}$$

Look at Constant Factor

Technique 4: Look at constant factor
Example: Coefficient of $(1, 0, 0)$ in $(D, 1, 0)$:

$$(2,1,0) \to -1$$

$$(3,1,0) \to \frac{2(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-2 \cdot 6(2k+1)(k+1)}{(k+2)(k+1)}$$

$$(5,1,0) \to \frac{2 \cdot 24(2k+1)(2k+3)(k+1)}{(k+3)(k+2)(k+1)}$$

$$(6,1,0) \to \frac{-2 \cdot 2 \cdot 120(2k+1)(2k+3)(k+2)(k+1)}{(k+4)(k+3)(k+2)(k+1)}$$

Look at Constant Factor

Technique 4: Look at constant factor

Example: Coefficient of $(1, 0, 0)$ in $(D, 1, 0)$:

$$(2,1,0) \to -1$$

$$(3,1,0) \to \frac{2(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-6(2k+1)(2k+2)}{(k+2)(k+1)}$$

$$(5,1,0) \to \frac{24(2k+1)(2k+2)(2k+3)}{(k+3)(k+2)(k+1)}$$

$$(6,1,0) \to \frac{-120(2k+1)(2k+2)(2k+3)(2k+4)}{(k+4)(k+3)(k+2)(k+1)}$$

Look at Constant Factor

Technique 4: Look at constant factor
Example: Coefficient of $(1,0,0)$ in $(D,1,0)$:

$$(2,1,0) \to -1$$

$$(3,1,0) \to \frac{2(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-6(2k+1)(2k+2)}{(k+2)(k+1)}$$

$$(5,1,0) \to \frac{24(2k+1)(2k+3)(2k+2)}{(k+3)(k+2)(k+1)}$$

$$(6,1,0) \to \frac{-120(2k+1)(2k+3)(2k+4)(2k+2)}{(k+4)(k+3)(k+2)(k+1)}$$

Look at Constant Factor

Technique 4: Look at constant factor
Example: Coefficient of $(1, 0, 0)$ in $(D, 1, 0)$:

$$(2,1,0) \to -1!$$

$$(3,1,0) \to \frac{2!(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-3!(2k+1)(2k+2)}{(k+2)(k+1)}$$

$$(5,1,0) \to \frac{4!(2k+1)(2k+2)(2k+3)}{(k+3)(k+2)(k+1)}$$

$$(6,1,0) \to \frac{-5!(2k+1)(2k+2)(2k+3)(2k+4)}{(k+4)(k+3)(k+2)(k+1)}$$

Look at Constant Factor

Technique 4: Look at constant factor
Example: Coefficient of $(1,0,0)$ in $(D,1,0)$:

$$(2,1,0) \to -1!$$

$$(3,1,0) \to \frac{2!(2k+1)}{k+1}$$

$$(4,1,0) \to \frac{-3!(2k+1)(2k+2)}{(k+2)(k+1)}$$

$$(5,1,0) \to \frac{4!(2k+1)(2k+2)(2k+3)}{(k+3)(k+2)(k+1)}$$

$$(6,1,0) \to \frac{-5!(2k+1)(2k+2)(2k+3)(2k+4)}{(k+4)(k+3)(k+2)(k+1)}$$

Notice Pattern!

Notice pattern!

Example: Coefficient of $(1, 0, 0)$ in $(D, 1, 0)$:

$$(D, 1, 0) \to \frac{(-1)^{D-1}(D-1)![(2k+1)\cdots(2k+D-2)]}{(k+1)\cdots(k+D-2)}$$

Doesn't work for $(2, 1, 0) \to$ Multiply by $\frac{k}{k}$:

$$(D, 1, 0) \to \frac{(-1)^{D-1}(D-1)![k][(2k+1)\cdots(2k+D-2)]}{(k)\cdots(k+D-2)}$$

Add Back in Divided Factors

Final step: Add back in divided factors
Example: Coefficient of $(1,0,0)$ in $(D,1,0)$:

$$(D,1,0) \to \frac{(-1)^{D-1}(D-1)![(k)(k-1)][(2k+1)\cdots(2k+D-2)]}{(k)\cdots(k+D-2)}$$

Technique 4: **Pascal's Identity and Binomial Coefficients**

Example:

Coefficients of $(D-i, \mu_2, \mu_3)$ in $(D, 0, 0)$ $(\mu_2, \mu_3 > 0)$. Fix $\mu_3 = 1$. Write the leading coefficient as a function $f(\mu_2, i)$ depending on i and μ_2.

	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$	...
$D-i,1,1$	1	$3k$	$7k^2$	$15k^3$	$31k^4$	$63k^5$	$127k^6$	$255k^7$	...
$D-i,2,1$		1	$4k$	$11k^2$	$26k^3$	$57k^4$	$120k^5$	$247k^6$	...
$D-i,3,1$			1	$5k$	$16k^2$	$42k^3$	$99k^4$	$219k^5$	...
...	...	...	...	...	...	...	...	...	...

Pascal's Identity and Binomial Coefficients

If we look at the first two rows of the table above, notice $3+1=4, 7+4=11, 15+11=26, 31+26=57, \ldots$ The key observation is that

$$f(\mu_2, i) = f(\mu_2, i-1) + f(\mu_2 - 1, i-1).$$

This should remind us of Pascal's identity

$$\binom{n}{k} = \binom{n}{k-1} + \binom{n-1}{k-1}.$$

Moreover, the leading coefficients of the first row has a clear pattern: $2^{i-1} - 1$. Using this as an initial condition and the recursive formula above, we could write out the formula for $f(\mu_2, i)$ as a function of i for each fixed μ_2.

Pascal's Identity and Binomial Coefficients

Using the pattern $2^{i-1} - 1$, we first conjecture:

$$(2k + \dots) \cdots (2k + \dots) - (k + \dots) \cdots (k + \dots).$$

However, it does not work! Later we apply the formula for geometric sum:

$$\begin{aligned} & 2^{i-1} - 1 \\ = & 2^{i-2} + 2^{i-3} + \cdots + 2^1 + 2^0, \end{aligned}$$

and change the conjecture to be:

$$\sum_{j=0}^{i-2} (2k + \dots)^{\overline{(i-2)-j}}.$$

Finally we conjecture the unfactored part of coefficient of $(D - i, \mu_2, \mu_3)$ in $(D, 0, 0)$ $(\mu_2, \mu_3 > 0)$:

$$\sum_{j=0}^{D-\mu_1-\mu_2-\mu_3} \binom{D - \mu_1 - \mu_2 - 1 - j}{\mu_3 - 1} \binom{j + \mu_2 - 1}{\mu_2 - 1} (2k + \mu_1)^{\overline{D-\mu_1-\mu_2-\mu_3-j}} \cdot (k + D - \mu_2 - j)^{\overline{j}}$$

Outline

Slow Run Time

Slow run time was a problem...

Storing polynomials and linear combinations helps a bit, but run time is still noticeable.

Example: To check the conjecture for $P_{(31,0,0)}$ took 14 hours, and as D increases the run time increases.

Outline

Future Directions

Future Directions:

- Complete the cases where we only have partial formulae.
- Try to prove our conjectures.
- Try to generalize to *n* variable case!

Thank you!

We would like to thank the Fields Institute for admitting us to this program to do research, and Professor Salmasian for mentoring us throughout the summer.