Computing the Hypergeometric Function of a Matrix Argument

Plamen Koev
Massachusetts Institute of Technology

Joint work with Alan Edelman
Why $pF_q(\cdot, \cdot, X)$?

- Distributions of
  \[ \lambda_{\text{min}}, \lambda_{\text{max}}, \det, \text{ etc.} \]
  of
  Wishart, Jacobi, Laguerre
  expressed in terms of $pF_q(\cdot, \cdot, X)$

- The distributions useful in:
  - Hypothesis testing (e.g., $\Sigma = I$, etc.)
  - Parameter estimation: $A \sim W_m(n, \sigma^2 I)$, $\sigma = ?$

- Applications in:
  - Population classification
  - Automatic target classification
  - Wireless communications

- Computing $pF_q(\cdot, \cdot, X)$: 40-year-old open problem
  - Notorious complexity and slow convergence
  - Empirical methods inefficient
Distribution of $\lambda_{\text{max}}$ of $4 \times 4$ Wishart with 7 DOF, $\Sigma = I$

- Exact vs Empirical with 20,000 replications
If $A \sim W_n(m, \Sigma)$ then

$$P(\lambda_{\text{max}}(A) < x) \sim x^{\frac{m}{2}} \cdot _1F_1 \left(\frac{m}{2}; \frac{n+m+1}{2}; -\frac{1}{2}x\Sigma^{-1}\right)$$

$$= x^{\frac{m}{2}} \cdot \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} p_{\kappa} \cdot x^{k} \cdot C_{\kappa}(\Sigma^{-1})$$

- Slow convergence $\Rightarrow \infty \sim 50, 100, 150$
- $C_{\kappa}(X)$ – Zonal Polynomial – Really hard: $O(n|\kappa|)$ terms in each!
- Our Contribution: $O(n)$
- Impossible until now
  - Previous best algorithm ($n = 5$): 8 days
    (Gutiérrez, Rodriguez, Sáez, 2000)
  - New algorithm: $\frac{1}{100}$ second
Computing $pF_q(\cdot, \cdot, X)$ is really hard!
Computing $pF_q(\cdot, \cdot, X)$ is really hard!
Computing $pF_q(\cdot, \cdot, X)$ is really hard!

Quadratic forms can be reduced to complex noncentral Wishart matrices. In general, results for noncentral Wishart matrices contain hypergeometric functions of a matrix argument which are typically expressed in terms of zonal polynomials, and are therefore difficult to evaluate. In this paper, we derive an alternate scalar representation for a particular hypergeometric function (10) and simplifying.

Unfortunately (10) is extremely difficult to evaluate numerically due to the hypergeometric function of a matrix argument. However, in this paper we are interested in the case when $M$ is rank-1, and can simplify (10) using...
We can compute now in fractions of a second

The Annals of Statistics

ON THE DISTRIBUTION OF THE LARGEST EIGENVALUE IN PRINCIPAL COMPONENTS ANALYSIS

BY IAIN M. JOHNSTONE

Stanford University

Let $x_{(1)}$ denote the square of the largest singular value of an $n \times p$ matrix $X$, all of whose entries are independent standard Gaussian variates. Equivalently, $x_{(1)}$ is the largest principal component variance of the covariance matrix $X'X$, or the largest eigenvalue of a $p$-variate Wishart distribution on $n$ degrees of freedom with identity covariance.

Consider the limit of large $p$ and $n$ with $n/p = \gamma \geq 1$. When centered by $\mu_p = (\sqrt{n-1} + \sqrt{p})^2$ and scaled by $\sigma_p = (\sqrt{n-1} + \sqrt{p})(1/\sqrt{n-1} + 1/\sqrt{p})^{1/3}$, the distribution of $x_{(1)}$ approaches the Tracy–Widom law of order 1, which is defined in terms of the Painlevé II differential equation.
\( A_p \sim W_p(n, I); \quad n/p = 5; \quad (\lambda_{\text{max}}(A_p) - \mu_p)/\sigma_p \to TW_1 \)
Jacobi: $A_p \sim W_p(q, \Sigma), \ B_p \sim W_p(n, \Sigma), \ A_p - \lambda (A_p + B_p)$

$p = 4k + 2, \ n/q = 2, \ n/p = 3,$

$\lambda_{\text{max}}(A_p (A_p + B_p)^{-1}) \rightarrow \lambda_{\text{max}}(A_\infty (A_\infty + B_\infty)^{-1}) \sim TW_1$
Open Problems

- The work of Gross and Richards (in the complex $\beta = 2$ case)
- New FFT-type algorithm
- Butler–Wood Laplace approximations
- The work of William Chen (IRS)
- Alternative formulas
- Finite implications of the Tracy-Widom laws (Johnstone)
The Work of Gross and Richards

\[ pF_q^{(1)}(a_1:p; b_1:q; X, Y) \equiv \sum_{k=0}^{\infty} \sum_{\kappa|k} \frac{(a_1)^{(1)}_{\kappa} \cdots (a_p)^{(1)}_{\kappa}}{k!(b_1)^{(1)}_{\kappa} \cdots (b_q)^{(1)}_{\kappa}} \cdot \frac{C^{(1)}_{\kappa}(X)C^{(1)}_{\kappa}(Y)}{C^{(1)}_{\kappa}(I)} \]

\[ = B \cdot \frac{\det (pF_q(a'_1:p; b'_1:q; x_iy_j))_{i,j=1}^m}{V(X)V(Y)}, \]

where

\[ a'_i = a_i - m + 1, \quad i = 1, 2, \ldots, p, \]
\[ b'_i = b_i - m + 1, \quad i = 1, 2, \ldots, q, \]
\[ V(Z) = \prod_{i>j}(z_i - z_j), \]
\[ B = \text{scalar constant} \]
Laplace approximations

Butler and Wood:

\[ 1F_1^{(2)}(a; c; X) = B \cdot \int_{0 < Y < I} e^{tr(XY)}(\det Y)^{a-n+\frac{1}{2}} \det(I - Y)^{c-a-n+\frac{1}{2}} (dY), \]

\[ 2F_1^{(2)}(a, b; c; X) = B \cdot \int_{0 < Y < I} \det(I - XY)^{-b}(\det Y)^{a-n+\frac{1}{2}} \det(I - Y)^{c-a-n+\frac{1}{2}} (dY), \]

where \( B = \frac{\Gamma_n^{(2)}(c)}{(\Gamma_n^{(2)}(a)\Gamma_n^{(2)}(c-a))} \).

Unclear when these formulas deliver good accuracy
The work of William Chen (IRS)

Upper percentage points of \( \theta_{0.8} \) of \( \theta(p,m,n) \), the largest eigenvalue of \( |B-\theta(W+B)| = 0 \), when \( s=7 \)

<table>
<thead>
<tr>
<th>m</th>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>.8774</td>
<td>.8961</td>
<td>.9097</td>
<td>.9202</td>
<td>.9284</td>
<td>.9351</td>
<td>.9407</td>
<td>.9453</td>
<td>.9493</td>
<td>.9531</td>
<td>.9559</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>.8342</td>
<td>.8576</td>
<td>.8750</td>
<td>.8886</td>
<td>.8995</td>
<td>.9084</td>
<td>.9158</td>
<td>.9222</td>
<td>.9275</td>
<td>.9325</td>
<td>.9369</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>.7930</td>
<td>.8201</td>
<td>.8408</td>
<td>.8571</td>
<td>.8703</td>
<td>.8812</td>
<td>.8904</td>
<td>.8983</td>
<td>.9051</td>
<td>.9112</td>
<td>.9167</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>.7543</td>
<td>.7844</td>
<td>.8077</td>
<td>.8263</td>
<td>.8415</td>
<td>.8542</td>
<td>.8650</td>
<td>.8743</td>
<td>.8824</td>
<td>.8895</td>
<td>.8957</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>.7183</td>
<td>.7507</td>
<td>.7761</td>
<td>.7966</td>
<td>.8136</td>
<td>.8278</td>
<td>.8401</td>
<td>.8506</td>
<td>.8598</td>
<td>.8680</td>
<td>.8752</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>.6850</td>
<td>.7191</td>
<td>.7462</td>
<td>.7683</td>
<td>.7867</td>
<td>.8023</td>
<td>.8158</td>
<td>.8274</td>
<td>.8377</td>
<td>.8468</td>
<td>.8549</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>.6542</td>
<td>.6896</td>
<td>.7180</td>
<td>.7414</td>
<td>.7610</td>
<td>.7778</td>
<td>.7923</td>
<td>.8049</td>
<td>.8161</td>
<td>.8261</td>
<td>.8350</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>.6258</td>
<td>.6621</td>
<td>.6915</td>
<td>.7159</td>
<td>.7365</td>
<td>.7543</td>
<td>.7697</td>
<td>.7832</td>
<td>.7952</td>
<td>.8059</td>
<td>.8155</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>.5995</td>
<td>.6365</td>
<td>.6666</td>
<td>.6918</td>
<td>.7133</td>
<td>.7318</td>
<td>.7480</td>
<td>.7623</td>
<td>.7750</td>
<td>.7864</td>
<td>.7966</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>.5752</td>
<td>.6125</td>
<td>.6432</td>
<td>.6691</td>
<td>.6912</td>
<td>.7104</td>
<td>.7273</td>
<td>.7422</td>
<td>.7555</td>
<td>.7675</td>
<td>.7783</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>.5526</td>
<td>.5902</td>
<td>.6213</td>
<td>.6476</td>
<td>.6703</td>
<td>.6900</td>
<td>.7074</td>
<td>.7229</td>
<td>.7368</td>
<td>.7492</td>
<td>.7606</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>.5316</td>
<td>.5693</td>
<td>.6006</td>
<td>.6273</td>
<td>.6504</td>
<td>.6706</td>
<td>.6885</td>
<td>.7044</td>
<td>.7187</td>
<td>.7317</td>
<td>.7434</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>.5121</td>
<td>.5497</td>
<td>.5812</td>
<td>.6082</td>
<td>.6315</td>
<td>.6521</td>
<td>.6704</td>
<td>.6867</td>
<td>.7014</td>
<td>.7148</td>
<td>.7269</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>.4940</td>
<td>.5314</td>
<td>.5629</td>
<td>.5900</td>
<td>.6137</td>
<td>.6345</td>
<td>.6531</td>
<td>.6697</td>
<td>.6848</td>
<td>.6985</td>
<td>.7110</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>.4770</td>
<td>.5142</td>
<td>.5457</td>
<td>.5729</td>
<td>.5967</td>
<td>.6177</td>
<td>.6366</td>
<td>.6535</td>
<td>.6689</td>
<td>.6828</td>
<td>.6957</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>.4611</td>
<td>.4980</td>
<td>.5294</td>
<td>.5566</td>
<td>.5805</td>
<td>.6018</td>
<td>.6208</td>
<td>.6380</td>
<td>.6536</td>
<td>.6678</td>
<td>.6809</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>.4462</td>
<td>.4828</td>
<td>.5141</td>
<td>.5412</td>
<td>.5652</td>
<td>.5865</td>
<td>.6057</td>
<td>.6231</td>
<td>.6389</td>
<td>.6533</td>
<td>.6666</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>.4322</td>
<td>.4684</td>
<td>.4995</td>
<td>.5266</td>
<td>.5506</td>
<td>.5720</td>
<td>.5913</td>
<td>.6083</td>
<td>.6247</td>
<td>.6394</td>
<td>.6529</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>.4166</td>
<td>.4421</td>
<td>.4727</td>
<td>.4995</td>
<td>.5234</td>
<td>.5449</td>
<td>.5643</td>
<td>.5820</td>
<td>.5982</td>
<td>.6131</td>
<td>.6268</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>.3838</td>
<td>.4184</td>
<td>.4485</td>
<td>.4750</td>
<td>.4987</td>
<td>.5201</td>
<td>.5395</td>
<td>.5573</td>
<td>.5736</td>
<td>.5887</td>
<td>.6027</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>.3634</td>
<td>.3972</td>
<td>.4266</td>
<td>.4527</td>
<td>.4761</td>
<td>.4973</td>
<td>.5167</td>
<td>.5345</td>
<td>.5509</td>
<td>.5660</td>
<td>.5802</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>.3451</td>
<td>.3779</td>
<td>.4067</td>
<td>.4323</td>
<td>.4554</td>
<td>.4765</td>
<td>.4957</td>
<td>.5134</td>
<td>.5298</td>
<td>.5450</td>
<td>.5592</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>.3285</td>
<td>.3604</td>
<td>.3885</td>
<td>.4137</td>
<td>.4364</td>
<td>.4572</td>
<td>.4762</td>
<td>.4938</td>
<td>.5102</td>
<td>.5254</td>
<td>.5396</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>.3031</td>
<td>.3329</td>
<td>.3594</td>
<td>.3733</td>
<td>.3951</td>
<td>.4151</td>
<td>.4336</td>
<td>.4507</td>
<td>.4668</td>
<td>.4818</td>
<td>.4959</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>.2786</td>
<td>.3064</td>
<td>.3285</td>
<td>.3423</td>
<td>.3620</td>
<td>.3803</td>
<td>.3980</td>
<td>.4146</td>
<td>.4302</td>
<td>.4447</td>
<td>.4582</td>
<td></td>
</tr>
</tbody>
</table>
Future work: Cooley–Tukey–type algorithm

- \((\text{DFT})_{ij}\) — characters of \(\mathbb{Z}/n\mathbb{Z}\) ↔ \(C_\lambda\) — characters of \(\text{GL}_n(\mathbb{C})\)

- Main identity

\[
C_\kappa(x_1, x_2, \ldots, x_n) = \sum_{\lambda < \kappa} C_\lambda(x_1, x_2, \ldots, x_{n-1}) \cdot x^n_{|\kappa|-|\lambda|} \cdot f_{\lambda\kappa}
\]

In matrix form:

\[
C_n = C_{n-1} \cdot Y_n(x_n)
\]

where for example

\[
Y_2(x) = \begin{bmatrix} 1 & x & x^2 & x^3 \\ 1 & x & x^2 & x^3 & x^4 \\ 1 & x & x^2 & x^3 & x^4 & x^5 \\ 1 & x & x^2 & x^3 & x^4 & x^5 & x^6 \end{bmatrix}
\]

- \(Y_n\) structured ... MVM takes linear time

- Cost(New Alg) \(\approx \sqrt{\text{Cost(Current Alg)}}\) ... just like FFT
Conclusions

- New efficient algorithm for $pF_q$: Takes seconds
- Works on $\Sigma = I$ and $\Sigma \neq I$
- Solves a 40-year-old problem

- Future Work:
  - Cooley–Tukey–like algorithm
    \[ \text{Cost} = O(\sqrt{\text{Current cost}}) \]
  - Toolbox

- Paper in Math. Comp., MATLAB software, slides, all available from:
  
  http://math.mit.edu/~plamen
Computing \( pFq(\cdot; \cdot; X) \)

\[
P(\lambda_{\text{max}}(A) < x) \sim x^{\frac{m}{2}} \cdot _1 F_1 \left( \frac{m}{2}; \frac{n+m+1}{2}; -\frac{1}{2}x\Sigma^{-1} \right)
\]

\[
= x^{\frac{m}{2}} \cdot \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} p_{\kappa} \cdot x^k \cdot C_{\kappa}(\Sigma^{-1})
\]

- Means computing zonal polynomials \( C_{\kappa}(\Sigma^{-1}) \)
- \( C_{\kappa}(\Sigma^{-1}) \) depends only on the eigenvalues \( x_1, x_2, \ldots, x_n \) of \( \Sigma^{-1} \)
- Illustrate \( \beta = 2 \) (complex); general \( \beta \) analogous

<table>
<thead>
<tr>
<th>Partition ( \kappa )</th>
<th>( C_{\kappa} )</th>
<th>Number of terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>( x_1 + \cdots + x_n )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>(2) ( \sum_{i \leq j} x_i x_j )</td>
<td>( O(n^2) )</td>
<td></td>
</tr>
<tr>
<td>(1, 1, 1) ( \sum_{i &lt; j &lt; k} x_i x_j x_k )</td>
<td>( O(n^3) )</td>
<td></td>
</tr>
<tr>
<td>( \kappa ) ( \sum_T x^T )</td>
<td>( O(n^{\mid \kappa \mid}) )</td>
<td></td>
</tr>
</tbody>
</table>
Computing $C_\kappa(X)$

**IDEA:** $C_\kappa$ are $\chi(\mathrm{GL}_n(\mathbb{C}))$; $\chi(\mathrm{GL}_{n-1}(\mathbb{C}))$ induce $\chi(\mathrm{GL}_n(\mathbb{C}))$

**Example:**

$$C_{(1,1)}(X) = \sum_{i<j} x_ix_j$$

$$= x_1x_2 + (x_1 + x_2)x_3 + \cdots + (x_1 + \cdots + x_{n-1})x_n$$

**Algorithm:**

$$s_1 = x_1$$
$$s_2 = s_1 + x_2 \quad (= x_1 + x_2)$$
$$s_3 = s_2 + x_3 \quad (= x_1 + x_2 + x_3)$$
$$\vdots$$
$$s_{n-1} = s_{n-2} + x_{n-1} \quad (= x_1 + x_2 + \cdots + x_{n-1})$$

$$C_{(1,1)}(X) = s_1x_2 + s_2x_3 + \cdots + s_{n-1}x_n$$

- **Cost:** $O(n)$ versus $O(n^2)$
- **In general:** $O(n)$ versus $O(n^{\left|\kappa\right|})$