Introduction

Abstract: The algorithm constructs natural statistics of a spectral sample, by using convolutions on the symmetric group and the Weingarten function. These statistics are unbiased estimators of cumulants of traces.

Application Areas/Subject: Computational statistics
Keyword: Random matrix, cumulant of traces, polykays
\section*{Initialization}

> restart;
> with(combinat, Chi, partition, permute);
> with(group, invperm, mulperms);
> with(ListTools, Flatten);

\section*{Background}

\subsection*{Construction of Schur function}

The procedure \texttt{Sch} takes in input an integer partition and returns the Schur polynomial in \(N\) indeterminates all evaluated in 1.

\begin{verbatim}
Sch := proc(\lambda, N)
local \mu;
\mu := combinat[conjpart](\lambda);
Matrix(nops(\mu), nops(\mu), (i, j) -> `if`(`\mu[i] + i - j < 0, 0, binomial(N, \mu[i] + i - j)`));
expand(linalg[det](%));
end:
\end{verbatim}

Example: for the partition (1,2,3) of the integer 6

\begin{verbatim}
> Sch([1, 2, 3], N)
\frac{1}{45} N^6 - \frac{1}{9} N^4 + \frac{4}{45} N^2
\end{verbatim}

for the partition (1\,^2,2) of the integer 4

\begin{verbatim}
> Sch([1, 1, 2], N)
\frac{1}{8} N^4 - \frac{1}{4} N^3 - \frac{1}{8} N^2 + \frac{1}{4} N
\end{verbatim}

\subsection*{Weingarten function}

The procedure \texttt{Wg} takes in input an integer partition and returns the Weingarten function as a rational function in \(N\).

The algorithm makes use of Schur polynomials and the character of the symmetric group.

\begin{verbatim}
Wg := proc(\mu, N)
local q, uno;
q := add(x, x = \mu);
uno := [`$\backslash`q(1, q)];
\end{verbatim}
factor \left( \frac{\text{add} \left( \frac{\left( \text{Chi} \left( \lambda, \text{uno} \right) \right)^2 \cdot \text{Chi} \left( \lambda, \mu \right)}{\text{Sch} \left( \lambda, N \right)} \right)}{q^2} \right) \right) \right); \\
end proc:

Example: for the partition (1,2,3) of the integer 6 
\[ W_g(1, 2, 3, N) \]
\[ \frac{2 N^2 + 13}{N (N + 5) (N + 4) (N + 2) (N + 1)^2 (N - 1)^2 (N - 2) (N - 4) (N - 5)} \]  
(3.2.1)

Example: for the partition (1^2,2) of the integer 4 
\[ W_g(1, 1, 2, N) \]
\[ - \frac{1}{(N + 3) (N + 1) (N - 1) (N - 3) N} \]  
(3.2.2)

\section*{Spectral k-statistics}

\section*{The Maple routines}

\subsection*{Some details on secondary Maple routines}

The procedure \texttt{compldisjcyc} takes as input a permutation and returns its decomposition in disjoint cycles. In the output there are also the fixed points;

\begin{verbatim}
> compldisjcyc := proc(a)
   local v, S;
   v := convert(a, `disjcyc`);
   S := {op(a)} minus {seq(op(c), c = v)};
   [seq([i], i = S), op(v)];
end:
\end{verbatim}

Example: for the permutation which fix 1 and 4 and switch 2 and 3
\[ \texttt{compldisjcyc}([1, 3, 2, 4]) \]
\[ [[1], [4], [2, 3]] \]  
(4.1.1.1)

Example: for the permutation which sends 1 in 2, 2 in 3, 3 in 4 and 4 in 1
\[ \texttt{compldisjcyc}([2, 3, 4, 1]) \]
\[ [[1, 2, 3, 4]] \]  
(4.1.1.2)

Example: for the identity permutations
\[ \texttt{compldisjcyc}([1, 2, 3, 4]) \]
\[ [[1], [2], [3], [4]] \]  
(4.1.1.3)

The procedure \texttt{tipo} takes as input a permutation in disjoint cycles and returns its cycle type, that is how many cycles of each length are present in the cycle decomposition of the
permutation.

```plaintext
> tipo := proc( )
> local n, v;
> if nargs = 1 then n := max(seq(op(x), x = args1));
> else n := args2; fi;
> v := sort([[seq(nops(c), c = args1)]]);
> [1$ (n - add(x, x = v)), op(v)];
> end proc:

Examples: for the permutation which fix 1 and 4 and switch 2 and 3
> tipo([[1, 4], [2], [3]])

[1, 1, 2]  \hspace{1cm} (4.1.1.4)

Examples: for the permutation which sends 1 in 2, 2 in 3, 3 in 4 and 4 in 1
> tipo([[1, 2, 3, 4]])

[4] \hspace{1cm} (4.1.1.5)

Examples: for the identity permutation
> tipo([ ], 4)

[1, 1, 1, 1] \hspace{1cm} (4.1.1.6)
```

\section*{The master function}

The procedure CXX takes as input a permutation and returns the formula (5.6) in Theorem 5.2, see \cite{1}. This formula corresponds to the convolution between products of traces of a spectral sample \( X \) and the inverse of a function giving the spectral sample size powered by the number of disjoint cycles.

```plaintext
> CX := proc( )
> local b, n, binv;
> b := args1;
> binv := invperm(b);
> if nargs = 1 then n := max(seq(op(x), x = b));
> else n := args2; fi;
> add( Wg( tipo( mulperms(binv, convert(a, `disjcyc`)), n), N) \cdot E( mul( Tr( mul( X, i = c ) ), c = compldisjcyc(a) ) ), a = permute(n) );
> expand(%);
> end:

CXX := proc( )
> local n; n := max(op(Flatten([args])));
> eval(CX( args1, n ), [seq(X_i = X, i = 1..n)]);
> end:

Examples: for the permutation which fix 1 and 4 and switch 2 and 3
> CXX([[1, 4], [2], [3]])

(4.1.2.1)
```
\[
\begin{align*}
E(Tr(X)^2 Tr(X^2)) \\
\frac{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3)}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3)} \\
- \frac{10 \ E(Tr(X)^2)^2}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N} \\
- \frac{20 \ E(Tr(X) Tr(X^3))}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N} \\
- \frac{E(Tr(X)^4)}{(N + 3) (N + 1) (N - 1) (N - 3) N} \\
+ \frac{10 \ E(Tr(X^4))}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3)} \\
+ \frac{N^2 \ E(Tr(X)^2 Tr(X^2))}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N}
\end{align*}
\]

Examples: for the permutation which sends 1 in 2, 2 in 3, 3 in 4 and 4 in 1

\[CXX([[2, 3, 4, 1]])\]

\[
\begin{align*}
\frac{10 \ E(Tr(X)^2 Tr(X^2))}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3)} \\
- \frac{5 \ E(Tr(X)^2)^2}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N} \\
- \frac{20 \ E(Tr(X) Tr(X^3))}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N} \\
2 \frac{E(Tr(X)^2)^2}{(N + 3) (N + 1) (N - 1) (N - 3) N} \\
- \frac{4 \ E(Tr(X) Tr(X^3))}{(N + 3) (N + 1) (N - 1) (N - 3) N} \\
- \frac{5 \ E(Tr(X)^4)}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N} \\
+ \frac{E(Tr(X^4))}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3)} \\
+ \frac{N^2 \ E(Tr(X^4))}{(N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3)}
\end{align*}
\]

The procedure eTr takes as input the output of the procedure CXX and replaces traces with
power sums indexed by their powers.

\[ eTr := y \rightarrow S_{\text{degree}(y)}, \quad eE := y \rightarrow y^{1/2} \]

\[ eCXX := \text{proc(} \quad \text{local } Ris, \text{ var}; \]
\[ \quad \text{Ris} := \text{eval(CXX(args), [Tr = eTr, E = eE])}; \]
\[ \quad \text{Ris} := \text{simplify(Ris)}; \]
\[ \quad \text{var} := \text{`minus'(indets(Ris), \{N\})}; \]
\[ \quad \text{mul(factorial(nops(x) - 1), x = compldisjcyc(op(args))} \cdot \text{collect(numer(Ris), \text{var})} \cdot \left( \frac{1}{\text{factor(denom(Ris))}} \right) \]
\[ \text{end proc; } \]

Example: for the permutation which sends 4 in 1, with fixed 2 and 3

\[ eCXX([ [4, 2, 3, 1] ]) \]
\[ -5 S_1^4 + 10 S_1^2 S_2 N + (-4 N^2 - 4) S_3 S_1 + (3 - 2 N^2) S_2^2 + (N^3 + N) S_4 \]
\[ (N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N \]

(4.1.2.3)

Example: for the permutation which sends 1 in 2, 2 in 3, 3 in 4 and 4 in 1

\[ eCXX([ [2, 3, 4, 1] ]) \]
\[ 6 \left( -5 S_1^4 + 10 S_1^2 S_2 N + (-4 N^2 - 4) S_3 S_1 + (3 - 2 N^2) S_2^2 + (N^3 + N) S_4 \right) \]
\[ (N + 3) (N + 2) (N + 1) (N - 1) (N - 2) (N - 3) N \]

(4.1.2.4)

\[ > \]

\section*{Conclusions}

This algorithm extends the symmetric functions k-statistics and polykays to spectral sampling. Spectral samples are eigenvalues of freely randomized classical sample. The notion of freely randomized classical sample has been introduced for the first time in [1].

\section*{References}


Legal Notice: The copyright for this application is owned by the author(s). Neither Maplesoft nor the author are responsible for any errors contained within and are not liable for any damages resulting from the use of this material. This application is intended for non-commercial, non-profit use only. Contact the author for permission if you wish to use this application in for-profit activities.

\[ > \]