Package ‘BosonSampling’

September 4, 2017

Type Package
Title Classical Boson Sampling
Version 0.1.1
Date 2017-08-23
Author Peter Clifford and Raphaël Clifford
Maintainer Raphaël Clifford <clifford@cs.bris.ac.uk>
Description Classical Boson Sampling using the algorithm of Clifford and Clifford (2017) <arXiv:1706.01260>. Also provides functions for generating random unitary matrices, evaluation of matrix permanents (both real and complex) and evaluation of complex permanent minors.
License GPL-2
Imports Rcpp (>= 0.12.12)
LinkingTo Rcpp, RcppArmadillo
Encoding latin1
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-09-04 17:48:50 UTC

R topics documented:

BosonSampling-package ........................................ 2
bosonSampler .................................................. 2
Permanent-functions .......................................... 4
randomUnitary .................................................. 5

Index 6
BosonSampling-package Classical Boson Sampling

Description

Classical Boson Sampling using the algorithm of Clifford and Clifford (2017) <arXiv:1706.01260>. Also provides functions for generating random unitary matrices, evaluation of matrix permanents (both real and complex) and evaluation of complex permanent minors.

Details

Index of help topics:

<table>
<thead>
<tr>
<th>BosonSampling-package</th>
<th>Classical Boson Sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permanent-functions</td>
<td>Functions for evaluating matrix permanents</td>
</tr>
<tr>
<td>bosonsampler</td>
<td>Function for independently sampling from the Boson Sampling distribution</td>
</tr>
<tr>
<td>randomUnitary</td>
<td>Random unitary</td>
</tr>
</tbody>
</table>

Author(s)

Peter Clifford and Raphaël Clifford
Maintainer: Raphaël Clifford <clifford@cs.bris.ac.uk>

bosonsampler Function for independently sampling from the Boson Sampling distribution

Description


Usage

bosonsampler(A, sampleSize, perm = FALSE)

Arguments

A the first n columns of an (m x m) random unitary matrix, see randomUnitary
sampleSize the number of independent sample values required for given A
perm TRUE if the permanents and pmfs of each sample value are required
Details

When the matrix $A$ is formed from the first $n$ columns of an $(m \times m)$ random unitary matrix, $X \leftarrow \text{bosonSampler}(A, \text{sampleSize} = N, \text{perm} = \text{TRUE})$ provides $X$values, $X$perms and $X$pmfs where $X$values is a list of $N$ independent sample values from the Boson Sampling distribution. Each sample value is a vector of $n$ output modes in random order. The elements of the vector can be sorted in increasing order to provide a multiset representation of the sample value.

The outputs $X$perms and $X$pmfs are lists of the permanents and probability mass functions (pmfs) associated with the sample values. The permanent associated with a sample value $v = (v_1, \ldots, v_n)$ is the permanent of an $(n \times n)$ matrix constructed with rows $v_1, \ldots, v_n$ of $A$. Note the constructed matrix, $M$, may have repeated rows since $v_1, \ldots, v_n$ are not necessarily distinct. The pmf is calculated as $\text{Mod}(\text{pm})^2/\text{prod}(\text{factorial}(\text{tabulate}(c)))$ where pm is the permanent of $M$.

Value

$X = \text{bosonSampler}(A, \text{sampleSize} = N, \text{perm} = \text{TRUE})$ provides $X$values, $X$perms and $X$pmfs.

References


Examples

```r
set.seed(7)
n <- 20 # number of photons
m <- 200 # number of output modes
A <- randomUnitary(m)[,1:n]
# sample of output vectors
valueList <- \text{bosonSampler}(A, \text{sampleSize} = 10)$values
valueList
# sample of output multisets
apply(valueList,1,sort)
#
set.seed(7)
n <- 12 # number of photons
m <- 30 # number of output modes
A <- randomUnitary(m)[,1:n]
# sample of output vectors
valueList <- \text{bosonSampler}(A, \text{sampleSize} = 1000)$values
# Compare frequency of output modes at different
# positions in the output vectors
matplot(1:m,apply(valueList,1,tabulate), pch =20, t = "p",
xlab = "output modes", ylab = "frequency")
```
Permanent-functions  

Functions for evaluating matrix permanents

Description

These three functions are used in the classical Boson Sampling problem.

Usage

cxperm(a)  
reperm(b)  
cxpermminors(c)

Arguments

A  
a square complex matrix.
B  
a square real matrix.
C  
a rectangular complex matrix where nrow(C) = ncol(C) + 1.

Details

Permanents are evaluated using Glynn’s formula (equivalently that of Nijenhuis and Wilf (1978)).

Value

cxperm(A) returns the permanent of the complex matrix A.
reperm(B) returns the permanent of the real matrix B.
cxpermminors(C) returns the vector of permanents of all ncol(C)-dimensional square matrices constructed by removing individual rows from C.

References


Examples

```r
set.seed(7)
n <- 20
A <- randomUnitary(n)
cxPerm(A)  
#  
B <- Re(A)
rePerm(B)  
#  
C <- A[,-n]
```
randomUnitary

v <- cxPermMinors(C)
# Check Laplace expansion by sub-permanents
C(cxPerm(A),sum(v*A[,n]))

Description
Generates a random unitary matrix (square)

Usage
randomUnitary(size)

Arguments
size      dimension of matrix

Examples
m <- 25 # size of matrix (m x m)
set.seed(7)
U <- randomUnitary(m)
#
# n <- 5 # First n columns
A <- U[,1:n]
Index

bosonSampler, 2
BosonSampling-package, 2

cxPerm (Permanent-functions), 4
cxPermMinors (Permanent-functions), 4

Permanent-functions, 4

randomUnitary, 2, 5
rePerm (Permanent-functions), 4