Computationally Efficient Algorithms for the Two-Dimensional Kolmogorov-Smirnov Test.

Raul HC Lopes, Peter R Hobson and Ivan D Reid
School of Engineering and Design,
Brunel University, Uxbridge UB8 3PH, United Kingdom

Brunel UNIVERSITY WEST LONDON

Introduction The classical one-dimensional Kolmogorov-Smirnov (KS) test compares two empirical distributions by defining the largest absolute difference D_{max} between their cumulative probability distribution functions (CDF) as a measure of disagreement between them. Adapting this test to more than one dimension is a challenge because there are 2^d-1 independent ways of defining a CDF when d dimensions are involved.

However there are many applications in experimental physics where comparing two-dimensional data sets is important. For example, monitoring the performance of high-energy physics detectors such as the Compact Muon Solenoid (CMS)[1] will involve periodic comparison of collected data to reference histograms, a task for which machine assistance must be sought.

Data As an example, Fig. 1 shows data obtained from reconstructing simulated tracks of muon pairs, originating from ***Z***-particle decay, within the CMS Silicon Tracker sub-detector, using the CMSSW software framework[1]. The histograms show the relationship between the reduced χ^2 of the track fit and the track pseudorapidity η (a function of the track angle from the normal to the beam-line). The first histogram gives results for perfect detector alignment, while the second was obtained after introducing small misalignments to individual detector modules, representative of probable initial errors[2]. Subsets of these data are used for comparisons in this work.

Peacock's Test Peacock[3] proposed defining a statistic independent of ordering in two dimensions by finding the largest difference between the CDFs under any ordering. For n points this means calculating the CDF in the $4n^2$ quadrants of the plane defined by all pairs of combinations (X_i, Y_j) of the points' coordinates. A brute-force implementation takes one step of complexity $\mathbf{\Theta}(n^2)$ for each point, for an overall complexity $\mathbf{\Theta}(n^3)$. This can be reduced by using a range-counting tree algorithm to give a time upper bound of $\mathbf{O}(n^2 \lg n)$ and a lower bound $\mathbf{\Omega}(n^2 \lg n)$[4].

Parallel Implementation We have implemented the Peacock test on a Linux cluster in both the brute-force and range-counting tree variants. Results for varying sample size and CPU numbers are given in Table 1. All tests on the same samples yielded identical results, and the speed-up was linear in the number of CPUs for both methods. As predicted, the methods scale as n^3 and $n^2 \lg n$, respectively.

Fasano and Franceschini's Variation Fasano and Franceschini introduced a variation[5] on Peacock's test with a greatly reduced lower bound, using n quadrants centred on the n points in the sample – a brute-force algorithm for this is presented in *Numerical Recipes*[6]. However, a range-counting algorithm can index the n points in $\mathbf{O}(n \lg n)$ time, followed by n two-sided range queries of $\mathbf{O}(\lg n)$ giving a lower bound of $\mathbf{O}(n \lg n)$[4]. Results for serial computations of both methods are given in Table 2, compared with Peacock's test. Again, performance scales with prediction.

Cooke's Algorithm A reportedly efficient implementation of Peacock's test has been introduced by Cooke[7]. The claim that this test runs in $\mathbf{O}(n \lg n)$ has been proven optimistic and differences to Peacock's test clearly demonstrated[4]. Table 3 shows comparisons between the Peacock and Cooke methods for various data samples. Cooke's method gives a finite distance (implying a difference) when identical samples are compared. As well, when the samples have repetitive data the method shows a greatly increased running time.

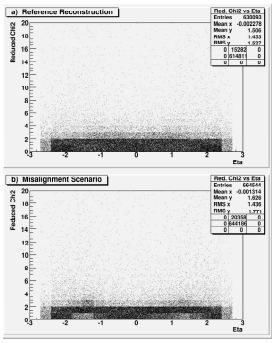
ROOT's Implementation of the 2D-KS Test ROOT[8] implements a 2D-KS test for comparison of *histogrammed* data. This means that it cannot give a true metric like Peacock's test. Instead it takes two cumulative 1D sums over the histogram bins, in rows and columns, and reports the average of D_{max} for the two cases. At finer binnings, these pseudo-CDFs asymptote to the CDFs of the X and Y coordinates separately (see Table 4), which can lead to scenarios where markedly different histograms compare as if they came from the same distribution[4].

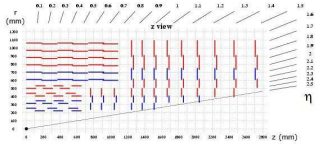
Conclusion These comparisons show that a proper comparison of 2D data can be difficult and expensive to achieve. For unbinned data the Fasano and Franceschini method has a clear advantage over Peacock's method, especially when implemented with range-counting trees. If the performance boost afforded to Peacock's method by a parallel implementation can also be carried over to Fasano and Franceschini's, then high-speed multi-CPU analyses would be available for datasets of 10^5 points and more. Cooke's algorithm, while undoubtedly fast, appears to have some problems in living up to its promise and would need to be evaluated on a case-by-case basis.

On the other hand, when comparisons need to be made on already histogrammed data, there is currently no substitute for the ROOT 2D-KS test, despite its shortcomings. It especially wins out on speed when normal binning ranges are used. An alternative test within the ROOT framework is under development, and we hope to provide this test in the near future.

Acknowledgements We would like to thank the CMS Tracker community and CMSSW developers for providing the tools used to produce the data sets. This work has been funded by the Science and Technology Facilities Council, UK.

Figure 1. 2D data used in this study; reduced χ^2 of track fit *vs.* pseudorapidity η for reconstructed muon-pair tracks in the CMS Silicon Tracker. a) Ideal detector geometry. b) Small "short-term scenario" misalignments of the detector modules. (Below: view of a quadrant of the Tracker, showing how η varies with track angle from the normal to the z axis.)





Sample Size	CPUs	Brute Force D_{max}	t (s)	Range-Count D_{max}	t (s)
4096	8	0.075195	2399	0.075195	127
	16	0.075195	1203	0.075195	63.2
	32	0.075195	601		
	64			0.075195	16.0
5120	8	0.078320	4699	0.078320	214
	16	0.078320	2349	0.078320	106
	32	0.078320	1175	0.078320	53.5
6144	8	0.084147	8117	0.084147	327
	16	0.084147	4060	0.084147	166
	32	0.084147	2029	0.084147	82.0
	64			0.084147	41.1
7168	8	0.086356	12872	0.086356	463
	16	0.086356	6445	0.086356	235
	32	0.086356	3223	0.086356	117
	64			0.086356	58.8
8192	8	0.083374	19232	0.083374	650
	16	0.083374	9615	0.083374	322
	32	0.083374	4810	0.083374	162
	64			0.083374	82.2

Table 1. Parallel inplementations of the brute-force and range-counting tree algorithms of Peacock's 2D-KS test (see text). *MPI C; Suse 9.3 64-bit cluster; 1.8 GHz AMD Opteron 265 worker nodes.*

Sample Size	Peacock				Fasano and Franceschini			
	BF D_{max}	t (s)	RCT D_{max}	t (s)	BF D_{max}	t (s)	RCT D_{max}	t (s)
1024	0.096680	110	0.096680	35.5	0.094747	0.15	0.094747	0.04
2048	0.084961	969	0.084961	180	0.081055	0.60	0.081055	0.13
3072	0.072591	7357	0.072591	502	0.069824	1.38	0.069824	0.24
4096	0.075195	19143	0.075195	1130	0.073364	2.46	0.073364	0.38
5120	0.078320	37591	0.078320	1970	0.075684	3.84	0.075684	0.52
6144	0.084147	64887	0.084147	2613	0.082601	5.53	0.082601	0.67
7168	0.086356	103208	0.086356	3758	0.085658	8.51	0.085658	0.83
8192	0.083374	153833	0.083374	5144	0.082520	11.7	0.082520	1.11
65536					0.075539	840	0.075539	20.1
131072					0.074738	1318	0.074738	52.4

Table 2. Serial inplementations of the brute-force (BF) and range-counting tree (RCT) algorithms of Peacock's 2D-KS test, and Fasano and Franceschini's variation. *C; Ubuntu; 2.0 GHz AMD Athlon XP.*

Samples	Size	Cooke D_{max}	t (s)	Peacock D_{max}	t (s)
aeta8 *vs.* aeta8	256	0.031250	0.0	0.0	0.06
meta9 *vs.* meta 9	512	0.015625	0.0	0.0	0.29
r7m4 *vs.* r7m4	2048	0.066406	0.43	0.0	12.41
r8m4 *vs.* r8m4	4096	0.077637	1.75	0.0	1023
r9m4 *vs.* r9m4	8192	0.061890	5.41	0.0	5259
aeta13 *vs.* meta13	8192	0.083374	0.17	0.083374	5259

Table 3. Comparisons between data sub-samples using the Cooke and Peacock methods. Samples: aeta8 – 256 tracks from aligned distribution; meta9 – 512 tracks from misaligned distribution; r7m4 – 128 repeats of 16 misaligned tracks; r8m4 – 256 repeats of 16 misaligned tracks; r9m4 – 512 repeats of 16 misaligned tracks; aeta13 – 8192 aligned tracks; meta13 – 8192 misaligned tracks. *C; Ubuntu; 2.0 GHz AMD Athlon XP.*

Histogram Size	$D_{max}(\chi^2)$	$D_{max}(\eta)$	t (s)
25 x 25	0.078755	0.009555	<0.01
50 x 50	0.082351	0.007044	<0.01
100 x 100	0.085978	0.005911	<0.01
200 x 200	0.083166	0.005745	0.01
500 x 500	0.083219	0.006118	0.09
1000 x 1000	0.084688	0.005947	0.36
RPy 1D-KS	0.081797	0.006348	0.46

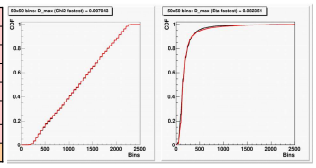


Table 4. The ROOT 2D-KS comparison test applied to 8192-track samples of data (aeta13 and meta13 from Table 3). The maximum differences in the two pseudo-CDFs obtained by scanning the histogram bins in two directions are shown; all tests were consistent with the samples' being from different populations. Results from the discrete 1D-KS test using the ***RPy*** stastistics package on the χ^2 and η data separately are also given. (Right: the two pairs of pseudo-CDFs in the 50x50 case.) *ROOT 5.13 and RPy libraries called from python; Scientific Linux CERN 3.08; 2.8 GHz Intel Pentium D.*

1) CMS Physics TDR: Volume I (PTDR1), *Detector Performace and Software*. http://cmsdoc.cern.ch/cms/cpt/tdr/-ptdr1_final_colour.pdf
2) L Barbone, N De Filippis, O Buchmueller, FP Schilling, T Speer and P Vanlaer, *Nucl. Instr. and Meth. A* **566**, 49 (2006).
3) JA Peacock, *Mon. Not. R. Astron. Soc.* **202**, 615 (1983).
4) RHC Lopes, I Reid and PR Hobson, http://bura.brunel.ac.uk/bitstream/2438/1166/1/acat2007.pdf.
5) G Fasano and A Franceschini, *Mon. Not. R. Astron. Soc.* **225**, 155 (1987).
6) WH Press, SA Teukolsky, WT Vetterling and BP Flannery, *Numerical Recipes in C*. Cambridge Univ. Press, 2002.
7) A Cooke, http://www.acooke.org/jara/muac/algorithm.html.
8) See http://root.cern.ch.

[http://people.brunel.ac.uk/~eesridr/CHEP_2DKS.ppt]