PROCEEDINGS OF SCIENCE

The two-dimensional Kolmogorov-Smirnov test

Raul H.C. Lopes*
School of Engineering & Design, Brunel University
E-mail: raul.lopes@brunel.ac.uk

Ivan Reid
School of Engineering & Design, Brunel University
E-mail: Ivan.Reid@brunel.ac.uk

Peter R. Hobson
School of Engineering & Design, Brunel University
E-mail: Peter.Hobson@brunel.ac.uk

Goodness-of-fit statistics measure the compatibility of random samples against some theoretical probability distribution function. The classical one-dimensional Kolmogorov-Smirnov test is a non-parametric statistic for comparing two empirical distributions which defines the largest absolute difference between the two cumulative distribution functions as a measure of disagreement. Adapting this test to more than one dimension is a challenge because there are $2^d - 1$ independent ways of defining a cumulative distribution function when d dimensions are involved. In this paper three variations on the Kolmogorov-Smirnov test for multi-dimensional data sets are surveyed: Peacock's test [1] that computes in $O(n^3)$; Fasano and Franceschini's test [2] that computes in $O(n^2)$; Cooke's test that computes in $O(n^2)$.

We prove that Cooke's algorithm runs in $O(n^2)$, contrary to his claims that it runs in $O(n \lg n)$. We also compare these algorithms with ROOT's version of the Kolmogorov-Smirnov test.

XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research
April 23-27 2007
Amsterdam, the Netherlands

*Speaker.

1. Introduction

Let X and Y be two independent stochastic variables whose cumulative distribution functions F and G are unknown. A classical two-sample problem consists of testing the null hypothesis

$$H_0 : F(x) = G(x), \text{ for every } x \in R^d$$

against the general alternative

$$H_1 : F(x) \neq G(x), \text{ for some } x \in R^d$$

This is the kind of problem that could arise in a context where, given an observed sample $X_1, \ldots, X_n$ and a control (or maybe a guess) sample $Y_1, \ldots, Y_m$, one must determine whether they come from the same distribution function. In the end we are asking if we can (dis)prove to a certain level of significance the null hypothesis that the two sets come from the same population.

The nature of the sets is, however, important in defining the kind of test available. When comparing whether educational patterns in London and Edinburgh are the same we are dealing with tables of numbers binned in discrete categories. Searching for neutrinos detected from the Supernova 1987A [3] demands dealing with points measured by pairs of time and amount of energy, both seen as continuous variables. A well accepted test for binned distributions is based on the χ^2 statistic.Continuous data can always be binned by grouping the events into ranges, but comes usually at the price of losing information.

For the one-dimensional continuous data, tests based on differences in the cumulative distribution functions are in general seen as very effective. The most widely used of these is the Kolmogorov-Smirnov test, which uses the maximum absolute difference between the distribution functions of the samples. This is in general an attractive test because it is distribution-free, it makes use of each individual data point in the samples, and it is independent of direction of ordering of the data.

Adapting goodness-of-fit tests to multi-dimensional space is generally seen as a challenge. Tests based on binning face the hurdle of what is called in the literature "the curse of dimensionality": a high dimensional space is mostly empty, and binning tests can only start to be effective when the data sets are very large [4].

Adapting the Kolmogorov-Smirnov test on the other hand demands defining a probability function that is independent of the direction of ordering, which does not seem to be possible given that there are $2^d - 1$ ways of defining a cumulative distribution function in a d dimensional space.

In this paper we discuss the correctness and complexity of four variations of the Kolmogorov-Smirnov test for comparing two two-dimensional data sets. In section 2, we discuss the test introduced by Peacock in [1], providing a lower-bound for its computation. In section 3, the variation of Peacock's test introduced by Fasano and Franceschini in [2] is presented, together with a lower-bound for computing it. Section 4 discusses Cooke's test which is introduced in [5] as an implementation of Peacock's test. We show that his test is not a faithful variation of Peacock's test and that the upper-bound for computing it is incorrectly stated in [5]. Section 5 discusses the ROOT implementations of the Kolmogorov-Smirnov test. We claim that ROOT's is not a Kolmogorov-Smirnov test even in the one-dimensional space. Section 6 presents experiments, and then a conclusion section follows.

2. Peacock's variation on the Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test is applicable to continuous, unbinned, one-dimensional data samples. It assumes that a list of data points can be easily converted to a cumulative distribution function. The test uses the maximum absolute difference between two cumulative distribution functions. When comparing one data set $F(x)$ against a known cumulative distribution function $P(x)$ the Kolmogorov-Smirnov statistic (K-S statistic) is

$$D_{KS} = max|F(x) - P(x)|$$

When comparing two samples with cumulative distribution functions $F(x)$ and $G(x)$ the statistic is defined as

$$D_{KS} = max|F(x) - G(x)|$$

Extending the K-S statistic to multi-dimensional space is a challenge. In one-dimensional space the statistic is independent of the direction of ordering of data because $P(> x) = 1 - P(< x)$. In an d dimensional space, however, there are $2^d - 1$ independent ways of defining a cumulative distribution function. In [1], Peacock introduced the idea of making the statistic independent of any particular ordering by finding the largest difference between the cumulative distribution functions under any possible ordering. Given n points in a two-dimensional space, that amounts to calculating the cumulative distribution functions in the $4n^2$ quadrants of the plane defined by all pairs (X_i, Y_j), X_i and Y_j being coordinates of any pairs of points in the given samples.

Peacock's test demands partitioning the n points in $4n^2$ quadrants and then computing the maximum absolute difference between cumulative distribution functions in all quadrants. The counting step can be performed by a brute force algorithm that for each point in a sample sweeps through each quadrant, deciding whether the point is in it. That gives n steps, one for each point, with complexity Θn^2, giving the final complexity of Θn^3.

The counting steps can be dramatically improved by using a range-counting algorithm. A set of n points in two dimensions can be indexed by a range-count tree in $\mathcal{O}(n \lg n)$ [6]. Given such range-counting tree, any two-sided range-counting query can be answered in $\mathcal{O}(n \lg n)$ time. An efficient algorithm for Peacock's test indexes the two samples of points in two range-counting trees, and then performs one two-sided query for each quadrant defined by the test, giving a time upper-bound of $\mathcal{O}(n^2 \lg n)$.

Fact: The lower-bound for finding the maximum of $\mathcal{O}(n^2)$ quantities is ωn^2 [7].

Claim: The lower-bound for performing Peacock's test on n points is the $\mathcal{O}(4n^2)$ quadrants and that is $\omega n^2 \lg n$ [8].

Peacock's test is very demanding. Performing the test on 2^{18} points with the brute force algorithm running on a 4GHz processor, would demand several days. Even the range-counting tree based algorithm would demand days to perform the test on a sample with a million points, see Table 1.

3. Fasano and Franceschini test

Fasano and Franceschini introduced in [2] a variation on Peacock's test that greatly reduces the lower-bound for its computation. Their heuristic consists in considering quadrants centred in

Set Size	BFP bound	BFP time (hours)	RCP bound	RCP time (hours)	BFFF bound	BFFF time (hours)	RCFF bound	RCFF time (hours)
2^8	2^{24}	2^{-20}	2^{19}	2^{-25}	2^{16}	2^{-28}	2^{19}	2^{-15}
2^{12}	2^{36}	2^{-8}	2^{27}	2^{-17}	2^{24}	2^{-20}	2^{27}	2^{-17}
2^{16}	2^{48}	2^{4}	2^{36}	2^{-8}	2^{32}	2^{-12}	2^{36}	2^{-8}
2^{20}	2^{60}	2^{16}	2^{44}	2^{0}	2^{40}	2^{-4}	2^{24}	2^{0}
2^{24}	2^{72}	2^{28}	2^{52}	2^{8}	2^{48}	2^{4}	2^{28}	2^{8}

Table 1: Peacock versus FF complexity

each point of the given samples. An sample of n points would define n quadrants. A brute force algorithms performs, for each given point, a sweep through all quadrants to decide whether the points is and must be counted in it. This algorithm is presented for example in [9].

An algorithm based on a range-counting tree can index the n points in $\mathcal{O}(n \lg n)$ time. After that n two-sided range queries of $\mathcal{O}(lgn)$ can be used two compute cumulative distribution functions differences in all quadrants.

Claim: The lower-bound for computing the Fasano and Franceschini test is the lower-bound for sorting n points in a two-dimensional plane, which is $\mathcal{O}(n \lg n)$ [8].

Table 1 compares upper-bounds for performing these two tests using both brute force and optimal algorithms. The identification in the columns are:

- BFFF: brute-force algorithm for Fasano and Franceschini test, as presented in [9].
- RCFF: range-counting tree version of Fasano and Franceschini test.
- RCP: range-counting tree version of Peacock test.
- BFP: brute-force algorithm for Peacock test.

The *upper-bound* columns present the asymptotic upper-bounds. The *time* columns present number of hours to compute in the hypothetical world where we have a 4GHz processor that can perform $\lg n$ range queries in $\lg n$ cycles.[1]

The Table indicates that for sets with up to a thousand (2^{10}) points any of the algorithms is fast enough. For more than one million points (2^{20}), only the last Fasano and Franceschini test implemented on top of a range-counting tree can give us times in the range of minutes.

4. Cooke's test

In [5], Cooke introduces an efficient implementation for Peacock's test. He describes the algorithm, and presents implementations for it in both Python and C. Chan in [10], discusses a parallel implementation for Cooke's algorithm.

Cooke's algorithm is evaluated here because it is the fastest of all tests based on the 1-D Kolmogorov test, even if, as we show below, the claim that the test runs in $\mathcal{O}(n \lg n)$ is false. The

[1] In isolation each of these assumptions is clearly false. Together they might resist Moore's law for a few years.

algorithm uses two binary trees each containing all points from both samples, each point marked to identify its source. The trees are ordered by x coordinate. In the first tree, points are inserted in increasing order of they y coordinate. As a result, points with lower y coordinate will be allocated next to the root. By sweeping the tree from leaves to root, the algorithm performs a sweep from top to bottom in all quadrants defined by the tree. The second tree reverts the order of insertion of points, locating points with higher y coordinates next to the root. That produces a sweep from bottom to top in the quadrants when the tree is swept from leaves to root. The number of points in each quadrant is updated during the sweeps, by counting the number of points in each subtree, and updating the maximum difference.

Cooke's unproved assumptions about his algorithm are:

- The dominating time is in the construction of the tree, which he believes is done in $\mathcal{O}(n \lg n)$.
- That by sweeping all quadrants defined by the nodes from top to bottom and reverse he is computing over all $4n^2$ quadrants defined by Peacock's algorithm.

The first assumption is clearly false. The algorithnm uses an unbalanced binary tree and the upper-bound to build such tree is $\mathcal{O}(n^2)$[11, 12]. Moreover the algorithm depends on ordering the tree level by y coordinate. As such it would not help using a balanced tree instead because the process of balancing would disturb the ordering in the levels. However, the algorithm is still efficient. In the end, the expected time to construct an unbalanced binary tree is $\mathcal{O}(n \lg n)$.

His second assumption is also false: his algorithm does not implement Peacock's test. Indeed it is not possible to select the maximum of $4n^2$ unknown quantities, something prescribed by Peacock's test, in less that $4n^2$ comparisons [7]. If Cooke's algorithm computed Peacock's test, one of the following would be true:

- Cooke would have invented a new algorithm for selecting the maximum of a random sequence of numbers,
- or some of the quadrants defined by Peacock are redundant, which would make sweeping them all unnecessary.

Tests performed with a few data sets clearly show that Cooke's test is not Peacock's test. Table 2 shows a set of tests. Each file F_i contains 1024 muon-simulated events, all samples from the same distribution. The table shows that Cook's test is definitely different from Peacock's.

5. The Kolmogorov-Smirnov test in ROOT

ROOT [13] implements a 2-D Kolmogorov-Smirnov test using an extension of its 1D Kolmogorov-Smirnov test. The two-dimensional test suffers from at least two serious limitations: it uses binned data, a limitation already found in the one-dimensional test, and and it computes the statistic as an average of two one-dimensional statistics.

ROOT's 2-D Kolmogorov-Smirnov test computes two one-dimensional Kolmogorov-Smirnov test over two given histograms. The Kolmogorov-Smirnov test is applied to find the probability that the x coordinate in the two histograms comes from the same distribution. Then a similar test

File	Cooke distance	Peacock distance
F_0 x F_0	0.00195312	0
F_1 x F_1	0.00195312	0
F_2 x F_2	0.00195312	0
F_3 x F_3	0.00195312	0
F_0 x F_1	0.0595703	0.0595703
F_1 x F_2	0.0957031	0.0957031

Table 2: Comparing Cooke's and Peacock's tests

File	40 bins	400 bins	4000 bins
F_0 x F_1	0.999999999999	0.999999999999	0.999996452676
F_1 x F_2	0.000192577442914	0.00316115427651	$5.08375684453e-06$
F_2 x F_3	0.0640695497946	0.110550802587	0.390164713988
F_3 x F_4	0.653380519844	0.631485586153	0.230296596904

Table 3: ROOT's test under different binning

compare the *y* coordinate in the two samples. ROOT then computes the average of those two probabilities. It is quite easy to define distributions that will completely fool this test. For example, two samples that are equal in the *x* coordinate, and whose *y* coordinates are almost permutations of each other will fool ROOT into determining that the samples have a great chance of being equal.

In addition, ROOT's 1-D Kolmogorov-Smirnov test takes as input a pair of histograms with binned data. This goes against the most fundamental principle of the Kolmogorov-Smirnov test, a test defined to be applied to continuous unbinned data [14, 9]. Indeed ROOT seems to be the only package to implement the Kolmogorov-Smirnov test over binned data, see for example [15, 16]. Table 3 shows results for two-sample comparisons under different binning. The files are all different and they have 4096 points each. Results report probability that the files are different, next to one indicating that compared samples are probably different. Notice that the third test shows an increase in the probability (of likelihood) with the increase in the number of bins. The last one shows exactly the opposite: a decrease in the probability (of likelihood) with the increase in the number of bins.

6. Experimental results

We performed tests with simulated data from Compact Muon Solenoid (CMS) experiment at CERN to evaluate the quality of the four tests we discussed. Two sets of data were used, representing reconstructed muon tracks from simulated Z-particle decay within the CMS experiment at CERN. The data pairs chosen were the reduced-χ^2 goodness-of-fit parameter for the track fit and Φ, the azimuthal angle of the reconstructed track around the beam-line axis. One set of data were produced using the ideal detector positions, the second after introduction of small misalignments representing the probable errors in positioning of the detectors when CMS first starts operation[17].

We used the bootstrap method for hypothesis testing, algorithm 16.1 of [18]. As described earlier in this paper, we worked with two-sample tests. Given two samples, our target is to find out

Files	Theta	ASL
F_0 x F_0	0	1
F_1 x F_1	0	1
F_2 x F_2	0	1
F_3 x F_3	0	1
F_0 x F_1	0.0595703	1
F_1 x F_2	0.0957031	0.961538
F_2 x F_3	0.0976562	0.961538
F_3 x F_0	0.0400391	1
F_0 x F_4	0.0375977	1
F_1 x F_5	0.0317383	1
F_2 x F_6	0.0463867	0.980769
F_3 x F_7	0.0283203	1
F_4 x F_4	0	1
F_5 x F_5	0	1
F_6 x F_6	0.0664062	0.961538
F_7 x F_7	0	1

Table 4: Peacock: H_0 acceptance

if the null hypothesis (that they come from the same distribution) is true. The bootstrap method here consists in running the statistics being tested once for the original samples to obtain one statistic, say $\widehat{\theta}$, and then run again by resampling the original sets.

In our tests, $\widehat{\theta}$ was obtained for the original samples of sizes m and n. Then fifty tests were performed over samples obtained by permutation with replacement from the union of the original samples.

It is important to keep in mind that Peacock, Fasano and Franceschini, and Cooke tests compute distance between samples. Lower distances indicate greater chances that the samples come from the same population. For these statistics, we are computing

$$Prob(\widehat{\theta}^* \geq \widehat{\theta})$$

ROOT's results show probabilities that the compared samples do not come from the same distribution. Again in ROOT's case, we are computing

$$Prob(\widehat{\theta^*} \geq \widehat{\theta})$$

Tables 4, 5, and 6 show results of runs for Peacock, Fasano and Franceschini, and Cooke tests, using pairs of samples from the same distribution. Tables 7, 8, and 9 show results for comparing samples from the same distribution using ROOT with 40, 400 and 4000 bins respectively. In each Table, the first column identifies the samples, the others give the distance $\widehat{\theta}$ computed by the respective test, and the significance level obtained. Files F_0 through F_3 have 1024 points each. Files F_4 through F_7 have 2048 points each. It should be noticed that samples from the same distribution are being compared, with small distances between pairs of samples being expected.

Files	Theta	ASL
F_0 x F_0	0.000976562	1
F_1 x F_1	0.000976562	1
F_2 x F_2	0.000976562	1
F_3 x F_3	0.000976562	1
F_0 x F_1	0.0527344	1
F_1 x F_2	0.09375	0.943396
F_2 x F_3	0.0927734	0.90566
F_3 x F_0	0.0361328	1
F_0 x F_4	0.0341797	1
F_1 x F_5	0.0288086	1
F_2 x F_6	0.0444336	0.962264
F_3 x F_7	0.0253906	1
F_4 x F_4	0.000488281	1
F_5 x F_5	0.000488281	1
F_6 x F_6	0.0625	0.924528
F_7 x F_7	0.0458984	0.962264

Table 5: Fasano and Franceschini: H_0 acceptance

Files	Theta	ASL
F_0 x F_0	0.00195312	1
F_1 x F_1	0.00195312	1
F_2 x F_2	0.00195312	1
F_3 x F_3	0.00195312	1
F_0 x F_1	0.0595703	0.169811
F_1 x F_2	0.0957031	0.0566038
F_2 x F_3	0.0976562	0.0188679
F_3 x F_0	0.0400391	0.90566
F_0 x F_4	0.0380859	0.943396
F_1 x F_5	0.0322266	0.981132
F_2 x F_6	0.046875	0.698113
F_3 x F_7	0.0288086	1
F_4 x F_4	0.000976562	1
F_5 x F_5	0.000976562	1
F_6 x F_6	0.0664062	0.0188679
F_7 x F_7	0.0507812	0.0377358

Table 6: Cooke: H_0 acceptance

Files	Theta	ASL
F_0 x F_0	0.0	1.0
F_1 x F_1	0.0	1.0
F_2 x F_2	0.0	1.0
F_3 x F_3	0.0	1.0
F_0 x F_1	0.081412658923	0.452830188679
F_1 x F_2	0.947605449092	0.0188679245283
F_2 x F_3	0.969351430782	0.0188679245283
F_3 x F_0	2.9058913591e-05	0.962264150943
F_0 x F_4	1.0	0.0188679245283
F_1 x F_5	1.0	0.0188679245283
F_2 x F_6	1.0	0.0188679245283
F_3 x F_7	1.0	0.0188679245283
F_4 x F_4	0.0	1.0
F_5 x F_5	0.0	1.0
F_6 x F_6	0.935443883178	0.0377358490566
F_7 x F_7	0.199500407284	0.415094339623

Table 7: ROOT: same distribution (40 bins)

Files	Theta	ASL
F_0 x F_0	0.0	1.0
F_1 x F_1	0.0	1.0
F_2 x F_2	0.0	1.0
F_3 x F_3	0.0	1.0
F_0 x F_1	0.0584077905556	0.660377358491
F_1 x F_2	0.832456544994	0.0188679245283
F_2 x F_3	0.894883937055	0.0377358490566
F_3 x F_0	9.32213365805e-05	1.0
F_0 x F_4	1.0	0.0188679245283
F_1 x F_5	1.0	0.0188679245283
F_2 x F_6	1.0	0.0188679245283
F_3 x F_7	1.0	0.0188679245283
F_4 x F_4	0.0	1.0
F_5 x F_5	0.0	1.0
F_6 x F_6	0.857287521801	0.0566037735849
F_7 x F_7	0.208282378927	0.320754716981

Table 8: ROOT: same distribution (400 bins)

Files	Theta	ASL
F_0 x F_0	0.0	1.0
F_1 x F_1	0.0	1.0
F_2 x F_2	0.0	1.0
F_3 x F_3	0.0	1.0
F_0 x F_1	0.248590402425	0.433962264151
F_1 x F_2	0.912996158571	0.0188679245283
F_2 x F_3	0.877199278419	0.0754716981132
F_3 x F_0	0.000599746915281	1.0
F_0 x F_4	1.0	0.0188679245283
F_1 x F_5	1.0	0.0188679245283
F_2 x F_6	1.0	0.0188679245283
F_3 x F_7	1.0	0.0188679245283
F_4 x F_4	0.0	1.0
F_5 x F_5	0.0	1.0
F_6 x F_6	0.749179307943	0.11320754717
F_7 x F_7	0.110671664737	0.679245283019

Table 9: ROOT: Same distributions (4000 bins)

Statistic	$\overline{asl}$	σ_{asl}
Peacock	0.9915864375	0.000245013276662
FF	0.981132	0.0009968039872
Cooke	0.6768867125	0.190904241164
ROOT(40 bins)	0.498820754717	0.221714429809
ROOT(400 bins)	0.510613207547	0.224918416993
ROOT(4000 bins)	0.524764150943	0.217596712946

Table 10: Mean and Standard Error for ASL for each test

Table 10 shows the means and standard errors for the ASL obtained for all tests. Least standard error is for Peacock's test, followed by Fasano and Franceshini's. Cooke's test and ROOT's test both present standard errors more than a hundred times higher than the other two tests.

Table 11 shows results of tests for samples comparing aligned against misaligned data. The first column describes the file. Then come the distances obtained for each of the tests: **P** for Peacock, **FF** for Fasano and Franceschini, **C** for Cooke, **R40** for ROOT with 40 bins, **R4000** for ROOT with 4000 bins. Files G_i have data from the simulation of misaligned tracks. Files G_0 through G_3 have 1024 points each. Files G_4 through G_7 have 2048 points each.

ROOT's results point to much larger differences between the samples than the other tests, results that seem unlikely given that we are comparing aligned against misaligned data and differences are not expected to be so big. In addition, ROOT shows large differences in results for the same pair of samples when the binning is changed.

Files	P	FF	C	R40	R4000
F_0 x G_0	0.0957031	0.0908203	0.0957031	0.0	2.08980157956e-07
F_1 x G_1	0.150391	0.147461	0.150391	0.999999544256	0.999998609916
F_2 x G_2	0.150391	0.139648	0.150391	0.980636391814	0.930160784516
F_3 x G_3	0.114258	0.107422	0.114258	0.257610913545	0.845952035138
F_0 x G_1	0.100098	0.0966797	0.100098	1.0	1.0
F_1 x G_2	0.138672	0.134766	0.138672	1.0	1.0
F_2 x G_3	0.128906	0.125977	0.128906	1.0	1.0
F_3 x G_0	0.110352	0.108887	0.110352	1.0	1.0
F_0 x G_4	0.0834961	0.0805664	0.0834961	1.14419584918e-12	3.5473241915e-06
F_1 x G_5	0.134766	0.131836	0.134766	0.999807422557	0.999994916243
F_2 x G_6	0.121582	0.119141	0.121582	0.935930450205	0.609835286012
F_3 x G_7	0.0961914	0.0942383	0.0961914	0.346619480156	0.769703403096

Table 11: H_0 rejection

7. Conclusion

In this paper, we tested four variations of the Kolmogorov-Smirnov test for two-dimensional data sets. We compared Peacock's, Fasano and Franceschini's , Cooke's, and ROOT's tests. We established precise computing bounds for the first three of them. We have shown that Cooke's test, contrary to what is stated in [5], is not an implementation of Peacock's test. Tests comparing samples from the same distribution indicate that Peacock's and Fasano and Franceschini's tests are more stable than the others. Experiments with ROOT have shown results with large discrepancies when the size of bins is changed. We are now extending these tests to incorporate three new statistics for multi-dimensional tests, including statistics based on minum energy and graph matching.

Acknowledgement

We would like to acknowledge financial support from the Science and Technology Facilities Council, UK.

References

[1] J. A. Peacock, *Two-dimensional goodness-of-fit testing in astronomy*, *Monthly Notices Royal Astronomy Society* **202** (1983) 615–627.

[2] G. Fasano and A. Franceschini, *A multidimensional of the Kolmogorov-Smirnov test*, *Monthly Notices Royal Astronomy Society* **225** (1987) 155–170.

[3] D. N. Sperger, T. Piran, A. Loeb, J. Goodman, and J. N. Bahcall, *A simple model for neutrino cooling of the large magellanic cloud supernova*, *Science* **237** (1987) 1471–1473.

[4] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualisation.* John Wiley & Sons Inc, 2007.

[5] A. Cooke, "The muac algorithm for solving 2d ks test." http://www.acooke.org/jara/muac/algorithm.html.

[6] L. Arge, G. S. Brodal, R. Fragerberg, and M. Laustsen, *Cache-oblivious planar orthogonal range searching and counting*, in *Proceedings of the twenty-first annual symposium on Computational geometry*, pp. 160–169, 2005.

[7] A. V. Aho, J. E. Holpcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[8] D. E. Knuth, *The Art of Computer Programmimg, volume 3: Sorting and Searching*. Addison-Wesley, 1998.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. Cambridge University Press, 2002.

[10] I. Chan, "Parallelizing a 2d kolmogorov-smirnov statistic." Download from http://beowulf.lcs.mit.edu/18.337-2002/projects-2002/ianchan/KS2D/ProjectPage.htm, July 2008.

[11] D. E. Knuth, *The Art of Computer Programmimg, volume 1: Fundamental Algorithms*. Addison-Wesley, 1998.

[12] T. H.Cormen, C. E.Leiserson, R. L.Rivest, and C. Stein, *Introduction to Algorithms*. MIT Press, 2001.

[13] R. Brun and F. Rademakers, *Root: An object-oriented data analysis framework*, in *Proceedings AIHENP'96 Workshop*, 1996. See also http://root.cern.ch/, July 2007.

[14] NIST/SEMATECH, "e-handbook of statistical methods." See http://www.itl.nist.gov/div898/handbook/, July 2007.

[15] W. N. Venables, "An introduction to R." See also http://www.r-project.org, July 2007.

[16] J. W. Eaton, *GNU Octave manual*. Network Theory Limited, 2002. See also http://www.gnu.org/software/octave/, July 2007.

[17] L. Barbone, N. D. Filippis, O. Buchmueller, F. Schilling, T. Speer, and P. Vanlaer, *Impact of cms silicon tracker misalignment on track and vertex reconstruction*, *Nuclear Instruments and Methods in Physics Research Section A* **566** (2006) 45–49.

[18] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.