

Recently it has been reported to us that when a different implementation of Chebyshev polynomials is applied to the same data sets used in the paper, the results produced by Chebyshev polynomials are not as good as indicated in the paper. We took this report seriously and started our investigation.

The first part of the investigation was to assess whether there was any deliberate fraudulent activities. We restored the Chebyshev polynomial code from backup tapes dated back as early as we could. Visual inspection of the code did not identify any suspicious statements in the code. More importantly, we reran the data sets with the restored code. Below is a summary of the reruns.

The table shows results of five data sets (all used in the paper) with dimensionality ranging from 1 to 4. The number of Chebyshev coefficients, k , was arbitrarily chosen from among the ones reported in [CN04]. As discussed in the paper, the pruning power of an indexing technique, which is the main metric for comparisons, is based on averaging 10 randomly chosen query time series. In the table below, we show the pruning power figures obtained using the restored Chebyshev code, and the corresponding figures reported in [CN04].

	Average pruning power (restored code)	Standard deviation (restored code)	Average pruning power reported in [CN04]
Stocks (1D, $k = 4$)	35%	11%	32%
Stocks (1D, $k = 10$)	59%	13%	50%
Stocks (1D, $k = 18$)	74%	11%	73%
ERP (1D, $k = 6$)	43%	22%	30%
ERP (1D, $k = 12$)	50%	20%	35%
Hockey (2D, $k = 4$)	95%	6%	90%
Kungfu (3D, $k = 4$)	71%	7%	74%
Kungfu (3D, $k = 8$)	76%	8%	78%
Angle (4D, $k = 10$)	53%	30%	55%

In most cases, the rerun figures were very similar to the ones reported in [CN04]. In the few cases when there were significant differences, the rerun results are even better than those reported in [CN04]. Thus, based on the

rerun results, it seems logical to conclude that the results reported in the paper were genuinely produced by the restored code.

Part 2 of our investigation focused on whether the restored code, written in C++, contained any bugs. We, thus, re-implemented the Chebyshev polynomial code with Matlab. Then we applied the Matlab version to all the data sets shown above and compared the results with the C++ version. Alas, we found that the Chebyshev coefficients were not identical, leading to a reduction in pruning power. Thus, it is clear that the C++ version contained a bug. We apologize for any inconvenience caused.

The discrepancies between the two versions vary significantly from one data set to another. The following table shows the “confusion” matrix, which is basically a histogram with relative errors as the bins. The columns show the percentages of the corresponding coefficients with a relative error of 0-2%, 2%-5%, 5%-10% and greater than 10%. The rows show the differences for the data sets.

	0-2%	2%-5%	5%-10%	> 10%
Stocks	0%	5%	25%	70%
ERP	21%	8%	8%	63%
Hockey	75%	13%	0%	12%
Kungfu	70%	14%	6%	10%
Angle	81%	5%	0%	14%

For the Stocks data set, around 70% of the Chebyshev coefficients produced by the two versions differ by more than 10%. Very similar situation is observed for the ERP data. Oddly enough, the situation is significantly different for the Hockey, Kungfu and Angle data sets. Over 70% of the coefficients are within 2% of each other; this can be attributed to the precision differences between C++ and Matlab.

Finally, the table above translates to pruning power, i.e., the more similar the Chebyshev coefficients between the two versions, the more similar are the corresponding pruning power results. Thus, for some data sets, Chebyshev (based on the Matlab version) can still be better than APCA, while for others, APCA could be more superior. In any event, our recommendation is that in future comparisons, it is imperative to include both APCA and Chebyshev polynomials, and even the Discrete Cosine and Fourier Transform methods. (*Written by Raymond Ng on March 24, 2006*)