Correction to "Accurate Statistical Approaches for Generating Representative Workload Compositions"

Lieven Eeckhout*, Rashmi Sundareswara†, Joshua J. Yi‡, David J. Lilja§ and Paul Schrater†
* ELIS Department, Ghent University, Belgium
† CS Department, University of Minnesota, Twin Cities, MN
‡ Freescale Semiconductor, Austin, TX
§ ECE Department, University of Minnesota, Twin Cities, MN

***Abstract*— This note describes a correction to the paper entitled "Accurate Statistical Approaches for Generating Representative Workload Compositions" published at the 2005 IEEE International Symposium on Workload Characterization in October 2005 in Austin, TX. The above paper stated that Independent Components Analysis (ICA) is a better alternative to Principal Components Analysis (PCA) for generating representative workloads. The ICA algorithm that we used in our study performs PCA prior to an orthogonal rotation for maximizing independence. As a result, PCA and the ICA algorithm should in fact perform equally well for composing representative workloads. This is only the case for the ICA algorithm that we used though; this is not a general statement about ICA versus PCA. However, due to a misunderstanding of the ICA software, we took this erronous conclusion. This is rectified in this correction note and we discuss what the results in the previously published paper really showed, namely that an appropriate weighting of the input variables can have a beneficial impact on the workload composition quality.**

I. Statistical data analysis techniques

This section revisits principal components analysis (PCA), independent components analysis (ICA) and cluster analysis (CA). We will refer to the original data matrix as matrix X in which the rows are the benchmarks and the columns are the program characteristics.

A. Principal Components Analysis (PCA)

Principal Components Analysis (PCA) [1] transforms p variables $X_1, X_2, \ldots, X_p$ into p principal components $Z_1, Z_2, \ldots, Z_p$ with $Z_i = \sum_{j=1}^{p} a_{ij} \cdot X_j$. This transformation has the following important properties: (i) $Var[Z_1] \geq Var[Z_2] \geq \ldots \geq Var[Z_p]$; and (ii) $Cov[Z_i, Z_j] = 0, \forall i \neq j$. Note that the total variation in the data remains the same before and after the transformation, namely: $\sum_{i=1}^{p} Var[X_i] = \sum_{i=1}^{p} Var[Z_i]$. After PCA analysis, the principal components are normalized, *i.e.*, the mean of each principal component equals zero, and the standard deviation equals one. Mathematically speaking, PCA solves the eigenvalue problem over the correlation matrix. The PCA transformation can also be expressed in a matrix notation: $Z = A \cdot X$ such that $E\{Z \cdot Z^T\} = I$.

The software that we use for applying PCA, STATISTICA [2], allows for applying PCA on normalized data as well as on non-normalized data, i.e., the software allows for automatically normalizing the data set prior to analysis. The default option of the software is to normalize prior to PCA; in our work we use this default option. We refer to the normalized original data matrix as X_n, *i.e.*, in X_n the columns have a unit-variance and the mean is zero. So, in fact, the principal components that we obtain are computed as follows: $Z = A \cdot X_n$ such that $E\{Z \cdot Z^T\} = I$.

B. Independent Components Analysis (ICA)

Independent Components analysis (ICA) [3] assumes that the original data matrix is centered, *i.e.*, the data set must be transformed so that the means along the columns in the data matrix are zero. We refer to this matrix as X_{zm}; note that $X_{zm} \neq X_n$. Some ICA algorithms, including the one we used, first perform PCA on X_{zm}: $Z = A \cdot X_{zm}$ such that $E\{Z \cdot Z^T\} = I$. As a second step, these ICA algorithms then transform the data set Z obtained from PCA into S using an orthogonal transformation: $S = K \cdot Z$. This transformation maximizes the independence between the dimensions in the transformed space. Since we assume that the independent components S_i have unit variance, the transformation K is orthogonal because $E\{S \cdot S^T\} = K \cdot E\{Z \cdot Z^T\} \cdot K^T = K \cdot K^T = I$. In other words, the transformation done through the K matrix is an orthogonal rotation. Note that this is only the case for some ICA algorithms, including the one that we used in our experiments.

The software that we use for applying ICA, the FastICA package for Matlab[1], centers the original data matrix but does not normalize the data set, *i.e.*, the mean for each column is zero but has a non-unit variance. This is a subtle but important difference which is the root cause of our erroneous results in the published paper as will be made clear later in this note.

C. Cluster Analysis

The next step in our workload composition methodology is to apply cluster analysis [1] on the transformed data set—note that we can apply cluster analysis in both the PCA and ICA spaces. The final goal is to obtain a number of groups containing various benchmarks that exhibit 'similar' behavior. This clustering is done based on the Euclidean distance between data points.

[1] `http://www.cis.hut.fi/projects/ica/fastica/`

II. What went wrong?

A important step in our methodology is that the transformations (both PCA and ICA) start from a normalized data matrix due to the heterogeneity of the data set. Some program characteristics, such as ILP, vary in the range of tens, whereas other program characteristics vary in the range of fractions smaller than 1, such as the cache miss rates. Normalization puts all the program characteristics on a common scale. In the above example, measuring benchmark similarity through clustering using non-normalized data would give a higher weight to the ILP metric than to the cache miss rate metrics.

For PCA we used the (default) normalization as desired. For ICA on the other hand, we erroneously assumed that the software also normalizes the data (by default) prior to PCA. This turned out not to be true though; the software only centers the data set. As a result, PCA and ICA operated on different data sets which was not our intention and which skewed the results substantially.

III. What should the data have looked like?

If the analyses would have been done according to our intention, *i.e.*, both the input matrices for PCA and ICA are normalized, we would not have observed a difference in the final clustering results between PCA and ICA. The reason is that the ICA implementation that we used does an orthogonal rotation on the data set obtained from PCA. This orthogonal rotation preserves the Euclidean distance between data points, *i.e.*, the distance between two data points after PCA equals the distance between two data points after ICA. And since clustering works on the Euclidean distance between data points, there should not be a difference between the clustering result after PCA and ICA. After correcting our setup we indeed verified that PCA and ICA yield the same clustering results.

Note again that this conclusion applies only to the ICA algorithm that we used. For other ICA algorithms that do not apply PCA prior to an orthogonal rotation, other conclusions may be reached, *i.e.*, ICA and PCA may yield different clustering results.

IV. What did the data really show?

The results presented in the published paper showed improvements in terms of workload accuracy and workload size. The obvious question then is, what did these results really show? It turned out that the original data matrix was not normalized and that due to various analyses prior to this study, most variables in the data matrix showed a standard deviation around 1 except for the data stream characteristics. The data stream variables had a significantly smaller standard deviation ranging between 0.3 and 0.6. For PCA, this was rectified by the software which normalized the data prior to analysis. This was not the case for ICA. As a result, the data stream characteristics had a smaller impact on the overall clustering in ICA. Apparently, this yields better workload composition results for the given data set.

What we learn from this experience is that giving a higher weight to particular variables in the PCA and ICA analyses can have an important impact on the clustering result. In fact, the composed workload can be significantly better in terms of accuracy and size by giving higher weight to workload characteristics that (apparently) correlate stronger with overall performance. In conclusion, an interesting direction for future research would be to determine how weights should be assigned to the various workload characteristics to obtain a clustering of benchmark programs that satisfies a desired set of constraints.

Acknowledgements

The authors would like to thank Kenneth Hoste from Ghent University for his assistance on this correction note.

References

[1] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, 5th ed. Prentice Hall, 2002.

[2] StatSoft, Inc., STATISTICA for Windows. Computer program manual. 1999. `http://www.statsoft.com`, 1999.

[3] A. Hyvärinen, "Survey on independent component analysis," *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.