A CODE-COMPARISON OF STUDENT ASSIGNMENTS BASED ON NEURAL VISUALISATION MODELS

David Martín, Emilio Corchado and Raúl Marticorena
Department of Civil Engineering, University of Burgos, Spain
C/ Francisco de Vitoria s/n. 09006 Burgos
{dmartin, escorchado, rmartico}@ubu.es

Keywords: Teaching of programming, Project evaluation, Programming metrics, Principal component analysis, Unsupervised learning.

Abstract: In this present multidisciplinary work, measurements taken from source-code comparisons of practical assignments completed by students of computer programme are analysed and visually represented, and conclusions are drawn so as to gain insight into the situation and the progress of the group. This representation is compared with another one generated by conventional code metrics, and the scope and meaning of the results are assessed in each case. These analyses use various statistical and neural dimensionality-reduction techniques for sets of multidimensional data.

1 INTRODUCTION

Analytical and multidimensional data visualization techniques are often applied in a range of professional contexts. They provide tools that are intended to facilitate the interpretation of results, and thus improve the effectiveness of decision-making that might affect the progress of a business. It appears reasonable for computing professionals involved in teaching tasks to take advantage of those same improvements.

A teacher's awareness of students, of the socio-educational context, and of the inherent dynamics within classroom groups is important in the definition of contents and in curricular development and design. The timely identification of structures, hierarchies and subgroups in a group of students means the teacher can focus follow up work and make individual or group changes so as to optimize the learning/teaching process. It is not an easy task, especially with large groups and with study modules that have few teaching hours. As an objective contribution to that awareness, conventional assessment tools are available to the teacher, which are complemented by subjective observations based on professional experience and "wisdom" (classroom time, personal consultation, tutorials, etc.). Quality improvement systems are fundamentally based on objective measurements generated by conventional assessment models or generalizations drawn from student satisfaction surveys. All of these are conducive to positive outcomes in the teaching/learning process, but lack an immediacy that is desirable for decision-making in the classroom.

Also within that same quality perspective, indicators are used in programming development methodologies to follow up projects. Programming languages can easily incorporate the application of measurement systems or metrics given that they use reduced grammars. Practical programming assignments performed by students of Computer Science could be candidates for this type of objective measurement.

Thus, in this study, projection techniques have been applied to multivariate data to obtain a 2D representation, simplifying the dataset but looking for the "most interesting" directions, in so far as those directions highlight specific aspects in the dataset. Principal Component Analysis (PCA) (Hotelling, 1933), (Friedman & Tukey, 1974) was used, as well as a neuronal model of Exploratory Projection Pursuit (EPP), Maximum-Likelihood Hebbian Learning (MLHL), which is described in (Fyfe & Corchado, 2002), (Corchado & Fyfe, 2003), (Corchado et al., 2004).

The analyses done target discovering of student groupings, based on the source code from their assignments, which may not be easily perceivable by means of quotidian contact in the classroom nor conventional assessment techniques. These observations may reveal individual or group non-desirable discordant practices so that teachers could focus on them and determine different adaptive teaching strategies based on their own experience. In the studied case it was also checked if the observed groupings might have an academic origin, with negative results. A comparative study was done on the results obtained from classic code metrics and no valuable observation was obtained from those graphs.

The rest of this paper is organized as follows. The high-dimensionality data analysis techniques applied in this study are discussed in Section 2. In Section 3, the source and the data collection methods are described. Section 4 presents the data processing and the results. The main conclusions are presented in Section 5 as well as proposals for future lines of work.

2 DIMENSIONALITY REDUCTION VISUALIZATION FOR DATA ANALYSIS

Projection methods project high-dimensional data points onto lower dimensions in order to identify "interesting" directions in terms of any specific index or projection. Such indexes or projections are, for example, based on the identification of directions that account for the largest variance of a dataset (such as Principal Component Analysis (PCA) (Hotelling, 1933), (Pearson, 1901), (Oja, 1989)) or the identification of higher order statistics such as the skew or kurtosis index, as in the case of Exploratory Projection Pursuit (EPP) (Friedman & Tukey, 1974). Having identified the interesting projections, the data is then projected onto a lower dimensional subspace plotted in two or three dimensions, which makes it possible to examine its structure with the naked eye. The remaining dimensions are discarded as they mainly relate to a very small percentage of the information or the dataset structure. In that way, the structure identified through a multivariable dataset may be visually analysed with greater ease.

A combination of these types of techniques together with the use of scatter plot matrixes constitute a very useful visualization tool to investigate the intrinsic structure of multidimensional datasets, allowing experts to study the relations between different components, factors or projections, depending on the technique that is used.

2.1 The Unsupervised Connectionist Model

The standard statistical EPP method (Friedman & Tukey, 1974) provides a linear projection of a dataset, but it projects the data onto a set of basic vectors which best reveal the interesting structure in data; interestingness is usually defined in terms of how far the distribution is from the Gaussian distribution.

One neural implementation of EPP is Maximum-Likelihood Hebbian Learning (MLHL) (Corchado et al., 2004), (Corchado & Fyfe, 2003), (Fyfe & Corchado, 2002), which identifies interestingness by maximising the probability of the residuals under specific probability density functions that are non-Gaussian.

Considering an N-dimensional input vector (x), and an M-dimensional output vector (y), with W_{ij} being the weight (linking input j to output i), then MLHL can be expressed (Corchado & Fyfe, 2003), (Corchado et al., 2003) as:

1. Feed-forward step:

$$y_i = \sum_{j=1}^{N} W_{ij} x_j, \forall i \qquad (1)$$

2. Feedback step:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \qquad (2)$$

3. Weight change:

$$\Delta W_{ij} = \eta . y_i . sign(e_j) | e_j |^{p-1} \qquad (3)$$

Where: η is the learning rate and p a parameter related to the energy function (Corchado et al., 2004), (Fyfe & Corchado, 2002), (Corchado & Fyfe, 2003).

3 COMPARISON AND MEASUREMENT OF PROGRAMMING ASSIGNMENTS

The objective of the study is to classify practical computer programming assignments completed by university students. It seeks to facilitate the identification of divergent or non-desirable situations in the educational process. Students following the "Programming Methods" study module in the 2nd year of *Ingeniería Técnica en Informática de Gestión* [Technical Engineering in Computer Science] complete practical assignments using the programming language Java. Throughout the four months of the study module, students have to develop two assignments - P1 and P2 - either individually or in pairs, following the design specifications as proposed by the teachers.

The first, P1, is collected in December and the second, P2, at the end of the first four months, in January. The second assignment entails making improvements to the first, includes new functions and applies the techniques learnt during the final stages of the study module. The practical assignment for the study module consists in the partial implementation of games.

3.1 Comparison of Practical Assignments

The primary datasets were constituted by comparisons of source code written in Java that were extracted by the "JDup" tool (Marticorena et al., 2008). The JDup tool generated the relevant comparisons, crossed by pairs from the 60 P1 and the 50 P2 practical assignments (1800 and 1250 respectively). JDup comparisons are made by establishing a minimum match length of 7 tokens.

The software tool compares tokens, snippets of code, and evaluates their percentage similarity. It was designed to detect plagiarisms (measured similarity in the region of 100%). The analysis of the entire spectrum of values of the set of comparisons was attempted in this work. Although in the first sample examined (December 2007), duplicate practical assignments could be identified, and the results were corroborated by direct checks (reviewing the code, personal interviews, etc.), neither the validity of the method nor the validity of the possible approximations used in the tool to improve the performance of the algorithm were formally tested. As opposed to the trivial possibility of a normal distribution, the detection and reiteration of clear groupings in the present work was taken as proof of the tools effectiveness.

3.2 Code Metrics

There are a series of measures that are widely used as evaluation indicators of software programmes. In this work, code metrics taken from a freeware tool called SourceMonitor were used (Campwood Software, 2007). SourceMonitor values are actively and effectively used for the characterisation and quantification of development effort in Computer System projects in the final year of Computer Engineering; projects that are much more diverse and very different. They allow objective comparisons, even between student intakes over recent years.

Table 1: Metrics calculated by SourceMonitor.

Statements
Percent Branch Statements
Method Call Statements
Percent Lines with Comments
Classes and Interfaces
Methods per Class
Average Statements per Method
Maximum Method Complexity
Maximum Block Depth
Average Block Depth
Average Complexity

The metrics, listed in Table 1, assess the size, the structure and the complexity of the code, although in our case, as the students all work on a shared design set by the teachers, some of the above metrics did not initially appear relevant. It was expected that the measurements of branch statements and complexity, or even the total number of lines, would be the most discriminatory when distinguishing between the practical assignments and the programming models proposed by the students.

These measurements were used alongside the representations generated by the comparison of the projects, and at the same time were independently treated with the same analytical techniques.

3.3 Data Preparation

The first group of practical assignments was corrected in December 2007, after the students had handed them in. The list generated by the JDup tool from the 60 assignments generated a longer list of 1800 comparisons which were ordered by degree of similarity. The reference solution prepared by the course teachers was included in the analysis. Having detected cases of plagiarism subject to sanctions, which appeared at the top of the list, the rest of the data were not directly interpretable by the teachers.

In a search for analogies with other datasets under study, the list was transformed into a symmetrical matrix. The comparisons were arranged by pairs as a Cartesian product, forming a symmetrical matrix that constitutes the dataset to be treated. By lines, each input variable may be understood as a distance from a practical concrete model, with values in the interval [0, 1].

These datasets, along with the corresponding labels, were recorded in a CSV format text file that was used as input data in the programme that applies the previously described reduction treatment and that generates the graphic representations. Alternative labels were also included in the file as well as other comparative or contrasting values that were solely used, after processing, in the representation and the final colouring of the graphs. Data preparation, performed on a conventional spread sheet, was a time consuming task, as a great amount of data had to be reordered and associated with academic management information taken from various sources: names, number of students completing the practical assignments, qualifications, etc.

3.4 Labelling of the samples

With a view to facilitating the interpretation of the graphs, each assignment was identified by a label. The use of the full names of each pair of students that performed the practical work would have taken up too much space and produced overlaps, without forgetting that the publication of student data of a personal nature should be subject to rigorous guidelines. Accordingly, the real names were delinked, and a two-letter code was assigned to the student that allowed the name to be easily localized between the two different data treatment stages and that also enabled a more compact on-screen visualization of the graph. It should be remembered that cases arise of one or more students that leave the course, in which case the student code that remains on file can also be quickly found. It is in the case of plagiarism where overlap is inevitable and reading is made more difficult; but it was assumed that these cases had been urgently investigated and sanctioned at an earlier stage.

Table 2 shows the codes assigned to the first assignment P1. The assignation of a special code "xx" to the teachers' reference solution proved very effective when observing and attempting to interpret the graphs.

The actual index of the assignment could be made to appear on the data table, although it is not especially relevant as the order of the table roughly corresponds to the order in which the assignments were handed in, which differed on both occasions.

Table 2: Labels used for the first practical assignment.

ey	af	fr	cq	bp+dd	fk
bb+cf	ax+bs	cv+du	dx+ft	ds	aw
ay+ef	cb	ar	aq+dk	cw	bw+cx
cp+fy	an+er	by	fd+fm	ee+eg	bd
ak+ap	ev	cc+eq	cu	bv+ep	dw+ek
cd+eu	bu+dn	cy+fw	bg+db	be+ch	ec+fb
dm	bc	bf+ew	dr	dp	ba+fe
cm+dg	br+en	as+et	bt	av+fs	ce+fc
ca+fh	cr	dh+ff	at+fa	au	bk
ac+ag	cs+eb	bm+fq	cg+dq	dc+dv	ad
xx					

Processing and labelling was repeated when analyzing and comparing the second practical assignment at the end of the four months, maintaining the same codes even though students might have changed partners.

4 DATA ANALYSIS

The model described in section 3.3 emerged due solely to other coinciding academic works along with the production of an extensive report on plagiarism. When the data corresponding to the first practical assignment had become available, PCA analysis identified the two clearly separate groups in Figure 1 that prompted ongoing study of the data gathered in this way. The position in the central band, towards the edge of the graph that is occupied by the teacher's reference solution was also significant ("xx" in Figure 1).

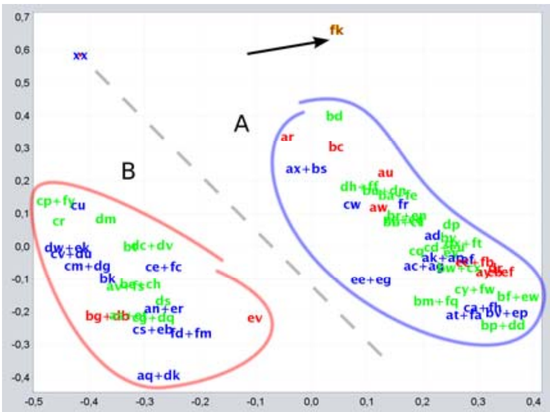

Figure 1: PCA analysis of P1 (academic marks are colour-coded).

Regardless of the researcher's discipline, the graph appears to awaken some concern from a teaching perspective. The evident polarization (the two large groups marked out as A and B in Figure 1 and Figure 2) might reflect some weakness in the teaching process, for example:

- Different approaches between the two teachers responsible for the practical assignments.
- Insurmountable weaknesses in half of the group.
- Students repeating the course, from different years.
- Class timetabling.

The possibility that these groupings were simply due to social relations in the group that leads to different influences or styles of programming, was also evaluated without this being of concern from an educational perspective. Whatever the cause might have been, it was thought that the study should continue to find out whether it could lead to some corrections or improvements in the teaching/learning process.

4.1 Initial Projection

The first observation was made using an earlier development applying both PCA and MLHL techniques. Codification of the students' names took place at a later point in time. Figure 1 (PCA) and Figure 2 (MLHL) were subsequently recreated using the same analytical techniques and labels already described. Shading (in grey on the printed graph) represents the marks awarded for each assignment.

A non-uniform, random distribution was observed (Figure 1 and Figure 2) regardless of which technique was used:

- The practical assignments that were copied occupy the same position.
- The assignments are distributed in two large, very different, separate groups. A further two subgroups could be identified within these two large groups.
- The reference solution offered by the teachers is found outside the groups, at an equidistant point some distance from them both.
- Some students may be seen in situations that are isolated from the groups. The most prominent is the case of a student on an international exchange programme (indicated with an arrow Figure 1 and Figure 2)

The two groupings may be clearly appreciated with both techniques. There is a notable separation and the definition of the two subgroups improves in the MLHL projection.

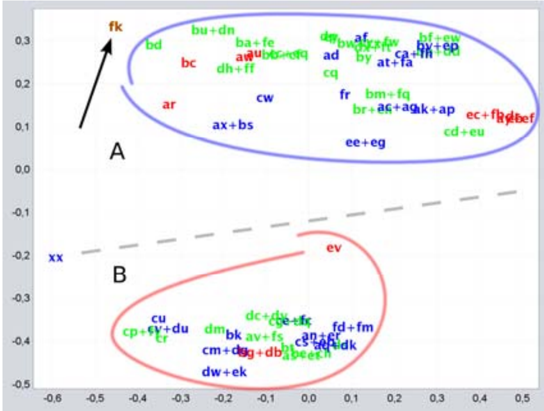

Figure 2: MLHL neural networking analysis of P1 (academic marks are colour-coded).

4.2 Variables in the Local Setting

It was subsequently investigated whether the polarization observed in the P1 projection (A and B in Figure 1 and Figure 2) might be due to some known and "non-desirable" cause. Possible causes of an academic origin are:

- Teacher.
- Group/timetable of the practical classes.
- Individual work or work in pairs.
- Students repeating the module.
- Mark awarded for the practical assignment.

The corresponding values were introduced into the CSV file and applied to the final graphs as colour-coded points and as text labels. In no case was a conclusive relation appreciated between the two visible groupings.

4.3 Treatment of the Second Assignment

Following treatment of the first assignments, the analysis of the second assignments was awaited, in which possible ratification and evolution of the pattern would be observable.

Whatever the case, two determining factors should be considered prior to arriving at any conclusion:

- It was not a matter of separate exercises, as the second practical assignment was an extension or an improvement of the first. As their starting point, each student began with the code handed in for the first practical and a major part of the entire code would remain unchanged or have only minimal modifications.

- Students were aware of the monitoring being carried out and had been informed of the sanctions for plagiarism. Logically enough, greater honesty might be expected in terms of reasonable collaboration between students throughout the development of their assignment.
- Students abandoned the course. This in turn meant lower numbers of samples and practical assignments in pairs being transformed into individual assignments.

During the break between assignments, improvements were made to the usability of the tool by incorporating a configuration window that facilitated repetition and adjustment of the tests. The same general treatment as in the case of P1 was repeated.

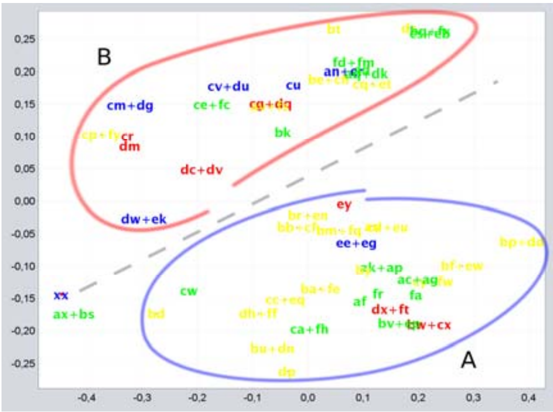

Figure 3: PCA analysis of P2 (academic marks are colour-coded).

In Figure 3, the persistence of two groups, although less separately, may be seen. The composition of the groups appears to be the same.

- Within the two groups, movements may be seen between the subgroups that are to some extent relevant.
- Greater dispersion. The number of practical assignments in peripheral positions increased, and notably some practices may be seen in zones close to the reference solution.

Although practical assignments are found in the four marking bands in both groups, a greater density of higher marks appears to be evident in one of the groups and in the other, a greater density of lower marks. It is not considered conclusive.

3.2 Code Metrics

The composition of the two groupings was contrasted with information on the groups and from students; likewise, the SourceMonitor code-metric measurements, described in section 3.2, were applied. No conclusive relation was observed, but these data were used to perform an independent analysis that generated the representation in Figure 4. The distribution was much closer to a normal distribution than that obtained from the comparisons and the only significant factor was the presence of samples at the extreme periphery. These “extreme” practices also occupied the prominent positions in the representations that were evident in the earlier analyses.

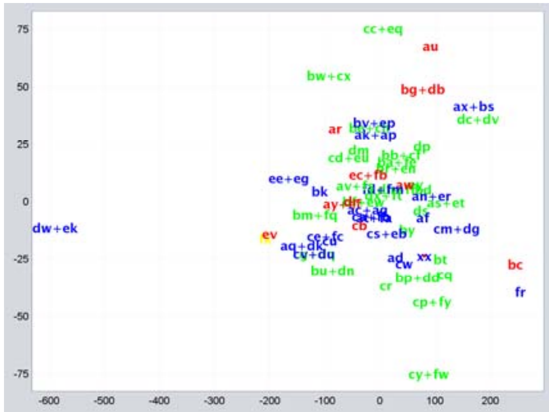

Figure 4: PCA analysis of P1 Metrics.

4.5 Evolution of the Distribution

Having conducted a separate, parallel analysis of the two datasets, the evolution between both was closely examined, and the observable groups and subgroups were defined. Tracing a line around the groupings (P1, B1, etc.) may be done on sight as shown in Figure 5.

Two large zones (A and B) are identifiable and various subzones may be perceived in each one. A1 and B2 correspond to the densest areas of the two groupings. Separated from these groups, an individual, isolated practical assignment (belonging to the exchange student, labelled “fk”) and the reference solution “xx” may be identified, at some distance, though on the axis of the representation. Some points approach that zone (such as "ax+bs" or "bd") in Figure 6 that corresponds to the second of the practical assignments. The assignment completed by the exchange student does not appear in this image, as he did not hand in this second assignment.

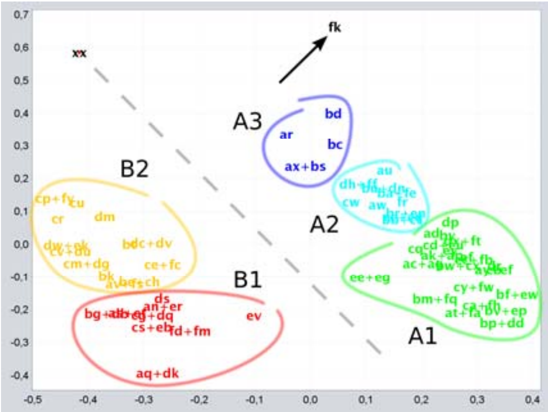

Figure 5: Classification of groupings in P1.

A separation of the groups may also be seen. Let us remember that the students were by that point aware of the analysis that was underway and had probably modified some of their practices relating to an occasional exchange of code. The closest points to “xx” are marked in Figure 6, as well as a unique case in which a clear change was detected between groups A and B. Informal contact was made with this student who explained that she had made significant transformations in order to resolve an important error discovered after handing in the first assignment. Another case of movement was also detected, but in this case it was associated with a change of partner.

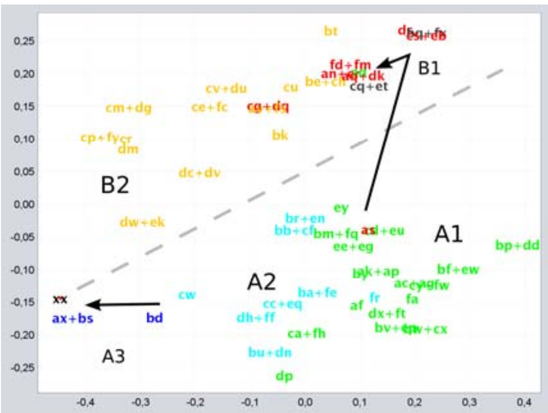

Figure 6: Classification of groupings in P2.

4.6 Experience Gained

Applying different dimension reduction techniques to the dataset produced by the JDUP tool, two clear groups were observed as well as some individuals in peripheral positions. The observation was mostly reproduced in a second dataset from a second assignment to the same students and some evolutions were observed. No coincidence was found to common academic settings, but the case of a foreign exchange student.

Same techniques were applied to code metrics obtained using the SourceMonitor tool to the same assignments in order to compare both, but results were quite poor. A common centred distribution was plotted.

These representations are not intended as a conclusive categorization and in no case are they proposed as evaluation tools. However, it is considered that they might be a valid tool to provide the teacher with insight into the group of students. Peripheral situations or pronounced changes can centre attention on certain students, whom the teacher might try to observe more closely during the practical sessions, and where necessary, a proposal for more personalized attention, provision of support, providing support and adaption of the proposed assignments.

5 CONCLUSIONS AND FUTURE LINES OF WORK

The set of values obtained by the JDup tool is considered a valid means of characterizing a set of practical assignments developed in separate ways on the basis of a common design. This was not the case of the values corresponding to the shared metrics obtained with SourceMonitor, which were shown to have a much more limited discriminatory capacity.

A model based on differential data is proposed, which is more easily generalizable than other theoretical measurements (metrics), the representative nature of which will vary according to the problem under study. The crossed-comparisons model contributed a rich description of the dataset, and allowed its dynamic to be observed, but did not allow us to identify the factors that caused these structures.

PCA and MLHL dataset visualization allowed an important polarization to be detected in the group of students under study. A search was made for matching elements, although it was not possible to associate this polarization with any defect or failing in the academic organization of the course, in the teaching methods, or even with the resulting set of marks.

The impression formed by the teachers was corroborated; students had learnt about the use of the JDup tool to detect plagiarism in the first mandatory practical assignment, had commented on it, and had taken it into account. It is believed that this is the reason for greater diversity and dispersion in the second assignment, without forgetting the logical and expected impact of the group's progress in the subject matter.

The use of statistical (PCA) and neuronal (MLHL) models applied to the work developed by students studying computer programming allowed information to be obtained on group dynamics in the classroom and its evolution over time; something that is difficult to achieve by direct observation and that might be useful for planning timely changes to teaching methods.

This work has sought greater knowledge of teaching/learning processes in the context of computing, thereby highlighting the spirit of improvement and the interest that form part of everyday teaching tasks; continuous improvement with a view to training qualified professionals.

The following future lines of work are proposed:

- Apply the method to other groups and subjects.
- Apply the comparisons model to other fields and to evaluation techniques where the representation generated by the model may be objectively contrasted with the curricular competence under evaluation.
- Propose improvements that facilitate portability of the JDup tool data.
- Improve the user interface of the analysis tool or integrate it into other tools.
- Apply other classification techniques that can improve the definition of the graphs and the automatic generation of groupings.

ACKNOWLEDGEMENTS

This research has been partially funded through project BU006A08 of the JCyL.

REFERENCES

Campwood Software. (2007). SourceMonitor Version 2.4 [Computer software]. Last accessed November 2008. Available from http://www.campwoodsw.com

Corchado, E., MacDonald, D., Fyfe, C. (2004). Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. In *Data Mining and Knowledge Discovery 8(3)* (pp. 203-225).

Corchado, E. and Fyfe, C. (2003). Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. In *International Journal of Pattern Recognition and Artificial Intelligence 17(8)* (pp. 1447-1466).

Corchado, E., Han, Y., Fyfe, C. (2003). Structuring Global Responses of Local Filters Using Lateral Connections. In *Journal of Experimental & Theoretical Artificial Intelligence 15(4)* (pp 473-487).

Friedman, J.H. and Tukey, J.W. (1974). A Projection Pursuit Algorithm for Exploratory Data-Analysis. In *IEEE Transactions on Computers 23(9)* (pp. 881-890).

Fyfe, C. and Corchado, E. (2002). Maximum likelihood Hebbian rules. In *ESANN 2002, 10th European Symposium on Artificial Neural Networks, Proceedings* (pp. 143-148).

Hotelling, H. (1933). Analysis of a Complex of Statistical Variables Into Principal Components. *In Journal of Educational Psychology 24* (pp. 417-444).

Marticorena, R., Bernal, I., Rebollo, J., López, C. (2008). Detección de la Copia de Prácticas de Programación con JDup. In *Actas de las XIV Jornadas de Enseñanza Universitaria de Informática, Jenui 2008* (pp. 579-586).

Oja, E. (1989). Neural networks, principal components, and subspaces. In *International Journal of Neural Systems 1*, (pp. 61-68).

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. In *Philosophical Magazine 2(6)*, (pp. 559-572).