Advantages and drawbacks of the Batch Kohonen algorithm

Jean-Claude Fort[1], Patrick Letremy[2], Marie Cottrell[2]

[1] Institut Elie Cartan et SAMOS-MATISSE
Université Nancy 1, F-54506 Vandoeuvre-Lès-Nancy, France
fortjc@iecn.u-nancy.fr

[2] Université Paris I, SAMOS-MATISSE, UMR CNRS 8595
90 rue de Tolbiac,
F-75634 Paris Cedex 13, France
pley,cottrell@univ-paris1.fr

Abstract :
The Kohonen algorithm (SOM) was originally defined as a stochastic algorithm which works in an on-line way and which was designed to model some plastic features of the human brain. In fact it is nowadays extensively used for data mining, data visualization, and exploratory data analysis. Some users are tempted to use the batch version of the Kohonen algorithm (KBATCH) since it is a deterministic algorithm which can go faster in some cases. After [7], which tried to elucidate the mathematical nature of the Batch variant, in this paper, we give some elements of comparison for both algorithms, using theoretical arguments, simulated data and real data.

1. Introduction

The Self-Organizing Map (SOM) of Teuvo Kohonen ([9], [10]) are used nowadays through numerous domains where it found effective applications by itself or coupled with other data analysis devices (classical factorial data analysis, source separation algorithm, filtering for signal processing, multilayer perceptrons for speech recognition, etc.). See for example [8], [5], [12], [3] etc. for definitions and numerous applications.

But it is well known that SOM appears to be a very useful extension (see [1]) of the classical Simple Competitive Learning algorithm (SCL) by adding neighborhood relations between the code-vectors. It is very interesting to keep in mind this property when studying the SOM algorithm.

Both SOM algorithm and SCL algorithm are on-line stochastic algorithms, which means they update the values of the code-vectors (or weight vectors) at each step, that is the arrival or presentation of a new observation. These modifications are instantaneously taken into account through the variations of the distribution and of the statistics along the observed data series. But both algorithms have their deterministic Batch equivalents, which use all the data at each step. If we have already observed N

data, then we use at one go these *N* values. This is the case for the algorithms known as Kohonen Batch algorithm (KBATCH, [11]) when the neighborhoods are taken into account or as Forgy algorithm or *k*-means algorithm ([6]) when there is no neighborhood relations.

2. On-line algorithms

Let us define the main notations :

- if the data are *d*-dimensional, the initial value of the *d*-dimensional code-vectors are $X_0(i)$, where *i* belongs to *I*, the set of units;
- the sequence of the observed data is denoted by ω_t, $t \geq 1$, they are also *d*-dimensional vectors;
- the symmetric neighborhood function $\sigma(i,j)$ measures the link between units *i* and *j*, $\sigma(i,i) = 1$ and decreases with the distance between *i* and *j*;
- the gain (or adaptation parameter) is $\varepsilon(t)$, $t \geq 1$, constant or decreasing.

When $\sigma(i,j) = 0$ as soon as $i \neq$ j, and = 1 for $i = j$, it is the SCL case.

Then the algorithm works in 2 steps. At time *t*+1,

- Choose a winner unit $i_0(t+1)$ defined by

$$i_0(t+1) = Arg\min_i \left\| \omega_{t+1} - X_t(i) \right\|$$

- Modify the code-vectors (or weight vectors) according to a reinforcement rule: the closer to the winner, the stronger is the change given by

$$X_{t+1}(i) = X_t(i) + \varepsilon(t)\sigma(i_0(t+1),i)\left(\omega_{t+1} - X_t(i)\right)$$

Actually few results are known about the mathematical properties of these algorithms (see [4]) except in the one dimensional case. The good framework of study, as for almost all the neural networks learning algorithms, is the theory of stochastic approximation.

For any set of code vectors $x = (x(i))$, $i \in I$, we put

$$C_i(x) = \left\{ \omega \,/\, \left\| x(i) - \omega \right\| = \min{}_i \left\| x(j) - \omega \right\| \right\}$$

It is the set of data for which unit *i* is the winner. The set of the $(C_i(x))$ is called the Voronoï tessellation defined by *x*.

We know that both SCL algorithm in general and SOM algorithm **for finite data and fixed size of the neighborhood function** can be considered as stochastic gradient algorithms associated to respectively the classical distortion and the extended distortion, see [13]. These distortion functions are given by

$$D(x) = \sum_{i \in I} \int_{C_i(x)} \left\| x(i) - \omega \right\|^2 \mu(d\omega)$$

and

$$D_{ext}(x) = \sum_{i\in I}\sum_{j}\sigma(i,j)\int_{C_j(x)}\|x(i)-\omega\|^2\mu(d\omega)$$

where μ is the distribution of the data. It can be continuous or discrete for SCL algorithm, but has to be discrete with finite support for the SOM algorithm.

Even when *D* is an actual energy function, it is not everywhere differentiable and we only get local information which is not what is expected from such a function. In particular, the existence of this energy function is not sufficient to rigorously prove the convergence! In any case, if they would be convergent, both algorithms (SOM and SCL) would converge toward an equilibrium of the associated ordinary differential equation (ODE), verifying

$$x^*(i) = \frac{\sum_j \sigma(i,j)\int_{C_j(x^*)}\omega\mu(d\omega)}{\sum_j \sigma(i,j)\mu(C_j(x^*))}.$$

In the SCL case, it means that all the *x*(i)* are the centers of gravity of their Voronoï tiles. More generally, in the SOM case, *x*(i)* is the center of gravity (for the weights which are given by the neighborhood function) of the union composed by their Voronoï tiles and the neighbor tiles. From this remark, the definitions of the batch algorithms can be derived.

3. The batch algorithms

We immediately derive the definitions of the batch algorithms. The aim is to find a deterministic iterative procedure to compute the *x*(i)*. This procedure is given by

$$x^{k+1}(i) = \frac{\sum_j \sigma(i,j)\int_{C_j(x^k)}\omega\mu(d\omega)}{\sum_j \sigma(i,j)\mu(C_j(x^k))}.$$

This expression defines the Forgy algorithm (so-called *k*-means algorithm) when there is no neighbor, that is when $\sigma(i,j) = 0$ as soon as $i \neq$ j, and = 1 for $i = j$.

It has been noticed in [7] that KBATCH is nothing else but a "quasi-Newtonian" algorithm (second-order gradient algorithm) which minimizes the extended distortion D_{ext}. It is a "quasi-Newtonian" algorithm because it uses only the diagonal part of the Hessian matrix and not the full matrix. Unfortunately, there are many disjoint sets where D_{ext} is differentiable and in each of them there is a local minimum of D_{ext}. However this fact gives a solid theoretical foundation to the Batch algorithm.

On another hand, it is well known that in practice a Newton algorithm is not exactly a descent algorithm, it can happen that the extended distortion function increases for some steps, even if the algorithm is deterministic. For example, Figure 1 shows the evolution of the Extended Distortion for KBATCH in a real case.

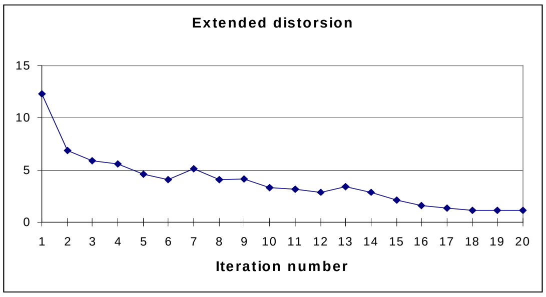

Figure 1 : Evolution of an extended distortion for the Batch Kohonen algorithm

But even this drawback has one advantage, since, after increasing, the distortion can decrease to a better minimum (by going through a "wall" of the distortion function).

Another problem is that KBATCH does not organize well the data. Figure 2 shows the result of a multidimensional scaling projection applied to the code-vectors in two cases. At left, the code-vectors result from the on-line SOM algorithm, at right they result from KBATCH. We see that in the first case, the code-vectors are displayed without crossing, while there are crossings in the second case.

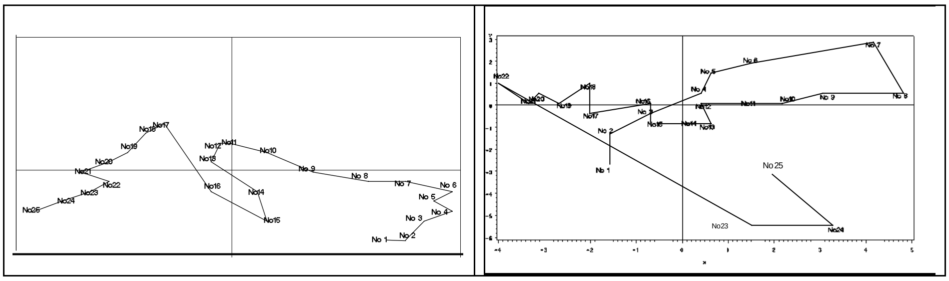

Figure 2 : Multidimensional scaling projection of the code-vectors for SOM and for KBATCH

In the two following sections, we present examples of simulated and real data, in order to compare the results that we get by using SOM and KBATCH.

4. Simulated data

The data are distributed along a noisy **W**-shape in 2 and 3 dimensions. The structure of the map is one-dimensional, with 50 units. The neighborhood function is constant in time, but exponentially decreasing with the distance between the units. The ε function is slowly decreasing. The number of iterations is the same (one iteration for KBATCH is equivalent to N iterations for SOM).

Figures 3 and 4 show the results of SOM and KBATCH, starting from the same initial situation.

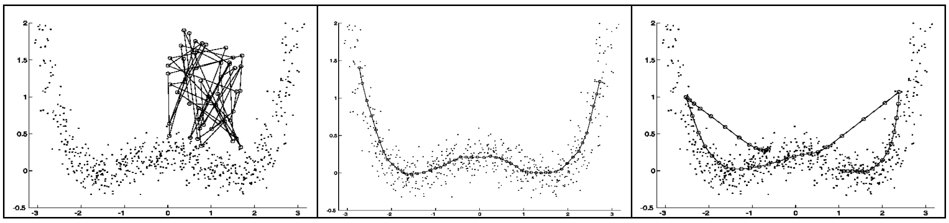

Figure 3 : Initial state, SOM result, KBATCH result, for 2D case.

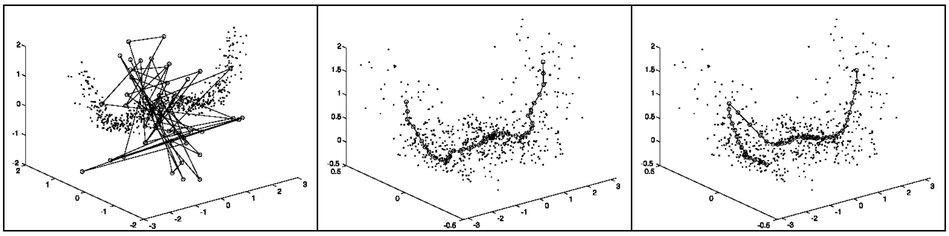

Figure 4 : Initial state, SOM result, KBATCH result, for 3D case.

We can observe that KBATCH does not succeed to well organize the code-vectors. A part of the string remains folded. Another examples can be found in [7].

5. Real data

We consider 1783 communes (districts) in the Rhône valley, in the south of France. This valley is situated on the two banks of the river Rhône. It includes some big cities (Marseille, Avignon, Arles, ...), some small towns, many rural villages. A large part is situated in medium mountains, in very depopulated areas since the so-called drift from the land. At the same time, in the vicinity of the large or small towns, the communes have attracted a lot of people who are working in urban employment. See [12], chapter 1, for the complete study.

The data table contains the current numbers of working population, distributed among six professional categories (farmers, craftsmen, managers, intermediate occupations, clerks, workers). The values are transformed into percentages and endowed with the χ^2 distance. See one example in Figure 5, that is the data concerning one district (it seems to be a working-class district).

We use a Kohonen one-dimensional network (a string) with 50 units to cluster into 50 classes the communes described by the professional composition data, using both SOM and KBATCH. Then in both cases, we use an ascending hierarchical classification to group the 50 Kohonen classes into 6 macro-classes. For this step, we work only with the 50 codes-vectors resulting from the previous classification, so this step is very easy to do and is not time consuming.

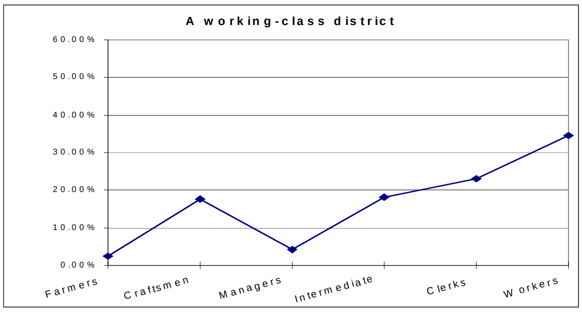

Figure 5 : The data for one commune

Results for the SOM algorithm (50 units, 10 000 iterations, neighborhood size decreasing from 7 to 1, 1 being the 0 neighbor case).
Figures 6 and 7 show the code-vectors along the 50-units string (numbered by column, from 1 to 10, 11 to 20, etc.) and the contents of the 6 macro-classes

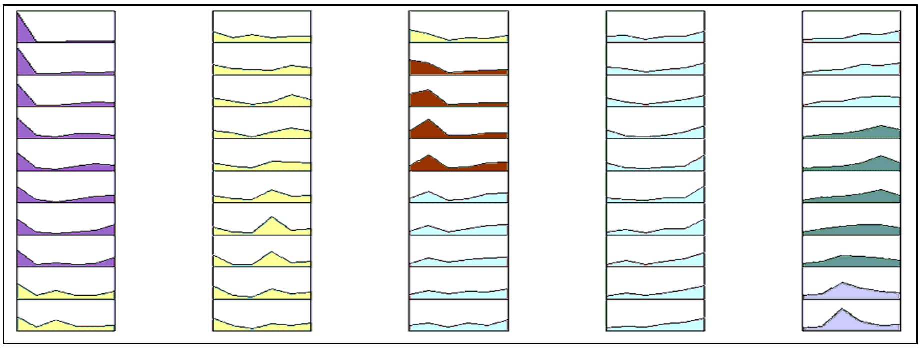

Figure 6 : The code-vectors, and the 6 macro-classes

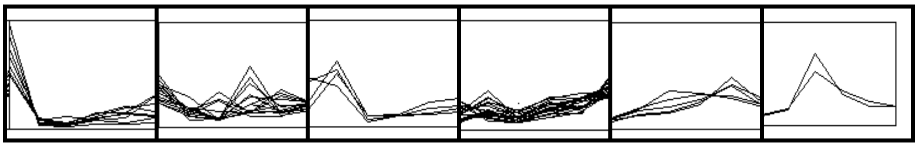

Figure 7 : The code-vectors grouped into 6 macro-classes

We immediately see that SOM is able to perfectly organize the data, as well in the Kohonen classes as in the macro-classes. The macro-classes group only contiguous classes[1]. They are almost totally ordered according to the percent of farmers, which decreases from 0.61, to 0.07 for a mean value of 0.22 and the percent of managers which increases from 0.009 to 0.30, for a mean value of 0.05. The final distortion is 0.1117.

[1] Macro-classes: I = 1 to 8, II = 9 to 21, III = 22 to 25, IV = 26 to 43, V = 44 to 48, VI = 49-50

Results for the KBATCH algorithm (50 units, 20 iterations, neighborhood size decreasing from 7 to 1, 1 being the 0 neighbor case).

Figures 8 and 9 show the code-vectors along the 50-unit string (numbered by column, from 1 to 10, 11 to 20, etc.) and the contents of the 6 macro-classes.

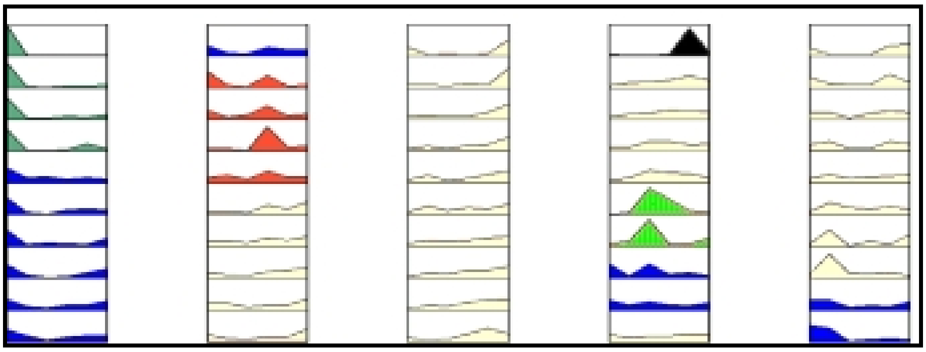

Figure 8 : The code-vectors, and the 6 macro-classes

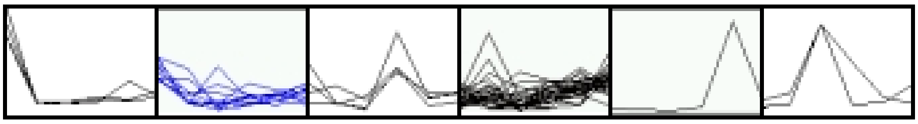

Figure 9 : The code-vectors grouped into 6 macro-classes

It is clear that KBATCH does not succeed to perfectly organize the data. Some neighbor code-vectors can be very different. The macro-classes group not contiguous classes[2]. The order between the classes is more difficult to perceive, even if in this case also, the farmers percent defines the macro-classes, it varies from 0.80 to 0.01, for a mean value of 0.22. However the final distortion is 0.08, that is smaller than the final distortion for the SOM algorithm. This means that the discrimination is good, as well as the homogeneity of the classes. But the organization fails.

One can also compare the number of communes by classes. See figure 10.

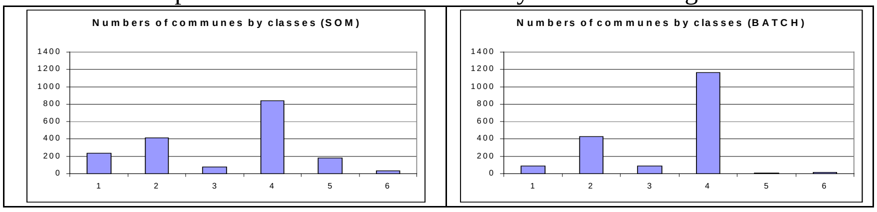

Figure 10 : Repartition of the communes according to both methods

The BATCH algorithm provides very unbalanced classes, that is not satisfactory, even if from the nature of the data, the classes have to be unbalanced.

[2] Macro-classes: 1 = 1 to 4, II = 5 to 11, 38-39, 49-50, III = 12 to 15, IV = 16 to 30, 32 to 35, 40 to 48, V = 31, VI = 36-37.

6. Conclusion

We have shown on some examples the advantages and the drawbacks of KBATCH.
Advantages : simplicity of the computation, quickness, better final distortion, no adaptation parameter to tune, deterministic reproducible results.
Drawbacks : bad organization, bad visualization, too unbalanced classes, strong dependence of the initialization.
Let us comment this last point. It is well known that the stochastic SOM algorithm is more or less insensitive to the initialization as we shown in [2], at least from the point of view of organization and neighborhood relations . The stochastic evolution of the code-vectors borrows the influence of the initialization. Contrarily, as it was shown in [7], for the KBATCH algorithm, the initialization is essential. This fact has two aspects, sometimes it allows to reach better minimum of the distortion function, but it can also conduce to very bad organization.

References

[1] E.de Bodt, M.Cottrell et M.Verleysen: Using the Kohonen algorithm for quick initialization of Simple Competitive Learning, *Proc. of ESANN'99, April 1999, Brugge*, M.Verleysen Ed., Editions D Facto, Bruxelles, p. 19-26, 1999.

[2] E.de Bodt, M.Cottrell: Bootstrapping Self-Organising Maps to Assess the Statistical Significanc of Local proximity, *Proc. of ESANN'2000, Avril 2000, Brugge*, M.Verleysen Ed., Editions D Facto, Bruxelles, p. 245-254, 2000.

[3] M.Cottrell, P.Rousset: The Kohonen algorithm: A Powerful Tool for Analysing and Representing Multidimensional Quantitative and Qualitative Data, *Proc. IWANN'97*, 1997.

[4] M.Cottrell, J.C.Fort, G.Pagès: Theoretical aspects of the SOM Algorithm, WSOM'97, Helsinki 1997, *Neurocomputing* 21, 119-138, 1998.

[5] G.Deboeck, T.Kohonen: *Visual Explorations in Finance with Self-Organization Maps*, Springer, 1998.

[6] E.W. Forgy: Cluster analysis of multivariate data: efficiency versus interpretability of classifications, *Biometrics*, 21, 3, 1965, p. 768.

[7] J.C.Fort, P.Letremy, M.Cottrell: Stochastic on-line algorithm versus batch algorithm for quantization and Self Organizing Maps, *Second NNSP Conference*, Falmouth, September 2001.

[8] S.Kaski: Data Exploration Using Self-Organizing Maps, *Acta Polytechnica Scandinavia*, 82, 1997.

[9] T.Kohonen: *Self-Organization and Associative Memory*, (3rd edition 1989), Springer, Berlin, 1984.

[10] T.Kohonen: *Self-Organizing Maps*, Springer, Berlin, 1995.

[11] T. Kohonen: Comparison of SOM Point Densities Based on Different Criteria, *Neural Computation*, 11,1999, p. 2081-2095.

[12] E.Oja and S.Kaski: *Kohonen Maps*, Elsevier, 1999.

[13] H. Ritter and T. Martinetz and K. Shulten : *Neural Computation and Self-Organizing Maps, an Introduction*, Addison-Wesley, Reading, 1992.

Annex

SOM classification

Macro-Class	Kohonen classes	Nb of districts	% farmers	Description
I	1 to 8	241	0.61	Rural districts, important proportion of farmers, three time the mean value. Very small villages, but using the χ^2 distance restores their importance
II	9 to 21	412	0.30	More farmers than in the total population, significant proportion of intermediate occupations, less workers than the mean.
III	22 to 25	76	0.30	Craftsmen, few intermediate and workers
IV	26 to 43	835	0.11	Few farmers and large proportion of workers. It is the largest class.
V	44 to 48	184	0.06	More intermediate occupations and clerks
VI	49 to 50	35	0.07	More managers and intermediate occupations

Table 1 : Description of the SOM classes

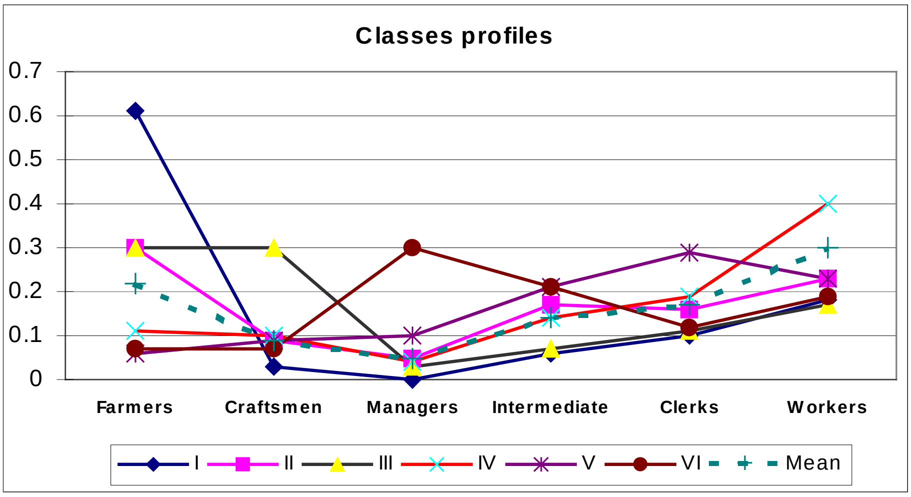

Figure A : Classes means, and total mean

The first variable is the more discriminant (with a Fisher value equal to 899), but all the Fisher are significant.

BATCH classification

Macro-class	Kohonen classes	Nb of districts	% farmers	Description
I	1 to 4	86	0.80	Rural districts, very important proportion of farmers, four time the mean value.
II	5 to 11, 38-39, 49-50	427	0.41	More farmers than in the total population, less intermediate, clerks and workers.
III	12 to 15	86	0.28	More farmers and intermediate, less clerks and workers
IV	16 to 30, 32 to 35, 40 to 48	1164	0.11	Very similar to the total population, with a little less farmers and a little more workers.
V	31	5	0.01	A very large proportion of clerks, but the class is very small.
VI	36 to 37	15	0.05	Essentially managers, but it is a small class

Table 2 : Description of the BATCH classes

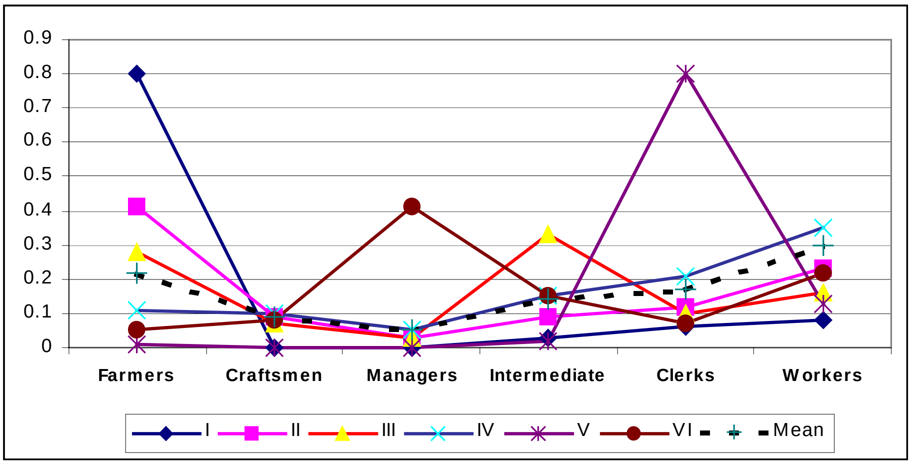

Figure B : Classes means, and total mean

In this case also, the first variable is the more discriminant, with a Fisher value of 1149, but all the Fisher are significant.