

J.-P. ASSELIN DE BEAUVILLE

F. DUMÉRY

Identification du microprocesseur destinataire d'un code-objet

Les cahiers de l'analyse des données, tome 13, n° 2 (1988),
p. 223-232

http://www.numdam.org/item?id=CAD_1988__13_2_223_0

© Les cahiers de l'analyse des données, Dunod, 1988, tous droits réservés.

L'accès aux archives de la revue « Les cahiers de l'analyse des données » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

IDENTIFICATION DU MICROPROCESSEUR DESTINATAIRE D'UN CODE-OBJET

[CODE MICROPROC.]

*J.-P. ASSELIN DE BEAUVILLE**

*F. DUMÉRY***

1 Le problème et les données

On sait qu'un programme-source, écrit en langage de haut niveau (Fortran, Pascal,...) ou en langage d'assemblage, doit être traduit en langage machine (ou code-objet) afin d'être exécuté par un microprocesseur donné. Du point de vue linguistique, ce code est une suite d'octets, chacun formé de huit chiffres binaires prenant les valeurs 0 ou 1.

Un observateur humain ne sait, au seul vu d'un code, déterminer sur quel microprocesseur il est destiné à être exécuté. Pourtant, la structure du jeu des instructions élémentaires et de leur représentation numérique ne peut manquer de se refléter dans la distribution des octets et des chiffres au sein de ceux-ci. L'objet du présent article est de montrer que des outils classiques d'analyse multidimensionnelle, analyse factorielle, analyse discriminante, classification, permettent, dans une large mesure de résoudre ce problème de reconnaissance des formes.

Il importe ici de noter qu'il ne s'agit pas d'un exercice purement gratuit: ceux qui manipulent des fichiers sur bandes ou autres supports savent bien que les informations manquent souvent pour relire correctement ces fichiers: un diagnostic fondé sur l'analyse des données pourrait, dans certains cas, tirer l'informaticien de l'embarras.

La collecte des données (octets de code) requiert quelque expérience, mais peut se faire à partir des systèmes d'exploitation et des utilitaires fournis avec les

(*) Laboratoire d'Informatique, U.F.R. des Sciences et Techniques, Parc de Gramont, 37200 TOURS.

(**) Étudiant en M.S.T. d'informatique appliquée à la Production Industrielle, Département d'Informatique, U.F.R. des Sciences et Techniques, Parc de Gramont, 37200 TOURS.

Identification du microprocesseur; Asselin de Beauville

A l'intersection de la ligne p et de la colonne Bx, on a, en dix-millièmes, la fréquence avec laquelle est mis à un le x-ème bit dans les octets du code du programme p ; le but est de distinguer les codes destinés à divers microproc, dont le nom est rappelé dans le sigle de p

	B7	B6	B5	B4	B3	B2	B1	B0
65a	5425	3774	4428	3014	3787	3266	2557	3573
65b	6370	4183	4535	3330	3656	3949	2884	3805
65c	6338	4488	4306	4079	3739	4209	3031	4367
65d	4038	3215	4713	2646	4262	3328	3457	3271
65e	4059	3473	4495	3131	4382	2984	2421	3346
65f	4713	3628	4761	3545	5259	3348	2051	3874
65g	3091	4299	5893	2559	3819	3546	2979	3581
68a	4456	3375	4006	3544	3550	5356	4750	4181
68b	4409	3608	5011	4878	3969	6203	5063	4722
68c	4000	3095	4148	3449	3416	5761	4774	4848
68d	3899	2528	5939	3793	4988	6376	5519	4682
68e	4053	2204	5684	2763	3941	5737	5211	4559
68f	3378	1909	5490	2956	3429	5439	5861	3716
68g	2904	1706	5990	3594	3398	5430	4896	4870
68h	4888	2220	4184	4373	4055	5972	5385	4608
68i	3293	2121	5721	3263	3978	5306	5132	4904
68j	4132	2251	5721	3263	3926	7093	6517	5892
68k	3914	2307	5218	3657	3958	6029	5343	5007
68l	4076	2332	4700	3946	4076	5729	5075	4790
80a	5711	6176	4760	3410	5231	4409	4814	5782
80b	3044	3588	3380	2049	3576	2998	3067	3461
80c	2681	3173	3424	4468	5367	4255	4548	5273
80d	3284	2953	3448	4611	5409	4790	5287	4909
80e	3022	3085	3763	3506	4316	4258	4001	5187
80f	2426	3076	3306	2706	3561	3972	3791	4005
80g	3875	4223	3086	2854	4385	4153	4223	5754
80h	2552	2396	3854	3385	4740	3750	4896	4401
80i	2947	2957	2500	2582	2899	2923	3260	3457
80j	3274	3576	3369	3162	4499	3895	4325	4164
80k	4355	4867	4560	4806	4980	4367	5097	5625
80l	4344	4651	4534	3873	4835	4197	4737	5662
80m	3180	2968	2980	2278	3645	3350	3833	3608
80n	4220	4139	4031	4139	4772	5142	4903	5039
80o	3111	3091	3087	2527	3204	3177	3259	3839
88a	4211	3854	4524	4673	4479	3914	4702	4539
88b	4499	4372	5199	3631	5165	4532	4561	4330
88c	4929	4267	3616	3989	4582	4387	5260	3936
88d	4385	3791	4845	3881	4946	4811	4837	4319
88e	3993	3162	3794	3824	4059	4853	5007	3971
88f	4073	3105	4526	4677	4647	5060	3952	4133
88g	4606	3424	4221	4719	4659	4511	5486	4230
88h	4581	3821	4375	3528	4153	4826	5119	4407
88i	4511	4260	4234	4703	4746	4040	5328	4064
88j	3963	3324	3987	3280	3648	4549	4482	3697
88k	3638	3429	3654	3702	4038	3205	3558	3878
88l	3393	3442	4841	4206	4097	4821	4673	4137
88m	4756	4795	4764	4380	4905	4325	4711	4301
88n	4654	4658	4105	4053	4994	3676	4810	4582
88o	4953	4126	4260	3772	4756	4200	4721	4370

micro-ordinateurs. En vue de la présente étude, des programmes ont été soumis à quatre machines, utilisant des microprocesseurs différents, énumérées ci-après:

Sil'z III, Leanord, système d'exploitation CPM, microprocesseur Z80;
Microprocess, système d'exploitation KDOS, microprocesseur 6809;
IBM-PC, système d'exploitation MS-DOS, microprocesseur 8088;
ORIC, sans DOS, microprocesseur 6502.

Le tableau de base, publié ici, croise un ensemble de segments de code (dont chacun est désigné par un sigle formé de deux chiffres évoquant le microprocesseur, suivis d'une lettre minuscule) avec l'ensemble des huit positions au sein d'un octet. On a, par exemple, $k(80c, B0) = 5273$, parce que le bit B0 est mis à 1 dans 52,73% des octets du troisième segment de code destiné au microprocesseur Z80 du Sil'z III.

Il eût été possible d'analyser le tableau de contingence lui-même, dénombrant pour chaque segment de code, le nombre des octets où est mis à un chacun des bits Bx; le calcul de pourcentage donne à tous les segments même poids, quelle qu'en soit la longueur. On peut encore envisager de créer un tableau à 256 colonnes dénombrant directement le nombre de fois que figure dans chaque segment chacune des valeurs numériques de l'octet: mais d'une part le tableau serait plus lourd à traiter et d'autre part pour avoir un profil significatif à 256 composantes, il faudrait sans doute traiter des segments de plusieurs kilo-octets.

Au §2, le tableau de base est soumis à l'analyse en composantes principales et à l'analyse factorielle discriminante. Au §3, ce même tableau, dédoublé et complété comme on l'expliquera, est soumis à l'analyse des correspondances.

2 Analyse en composantes principales et analyse factorielle discriminante

2.1 Analyse en composantes principales

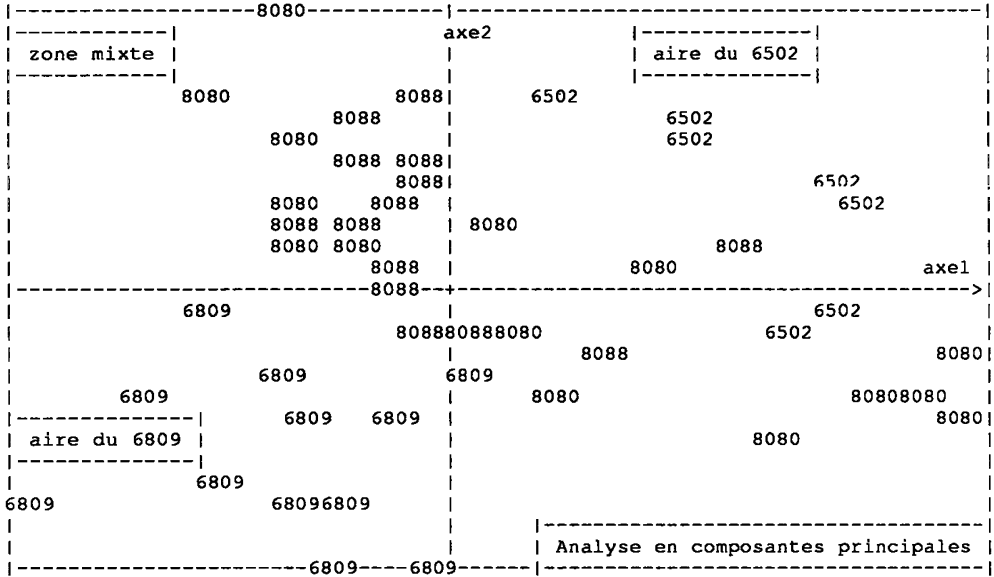
Nous avons utilisé le logiciel ADDAD.

Le plan des deux premiers facteurs explique environ 64% de l'inertie totale (38% et 28% pour les axes 1 et 2 respectivement).

Le premier axe peut s'interpréter par les variables B1 et B2 qui apportent la plus forte contribution à l'inertie de cet axe sur lequel elles sont assez bien représentées. Vers la gauche du premier axe sont les codes pour lesquels les bits 1 et 2 sont souvent à 1; tandis que vers la droite, sur ce même axe, se projettent les codes dont les bits 1 et 2 sont plus souvent à 0.

Le deuxième axe factoriel s'explique presque uniquement par B6; les valeurs maxima de F2 étant attribuées à des codes où le sixième bit de l'octet est souvent à 1; l'autre extrémité de l'axe 2 étant caractéristique des microprocesseurs pour lesquels le sixième bit est souvent à zéro.

Sur le nuage des individus, apparaît une nette séparation des codes destinés aux microprocesseurs 6809 et 6502; tandis qu'à partir d'un troisième



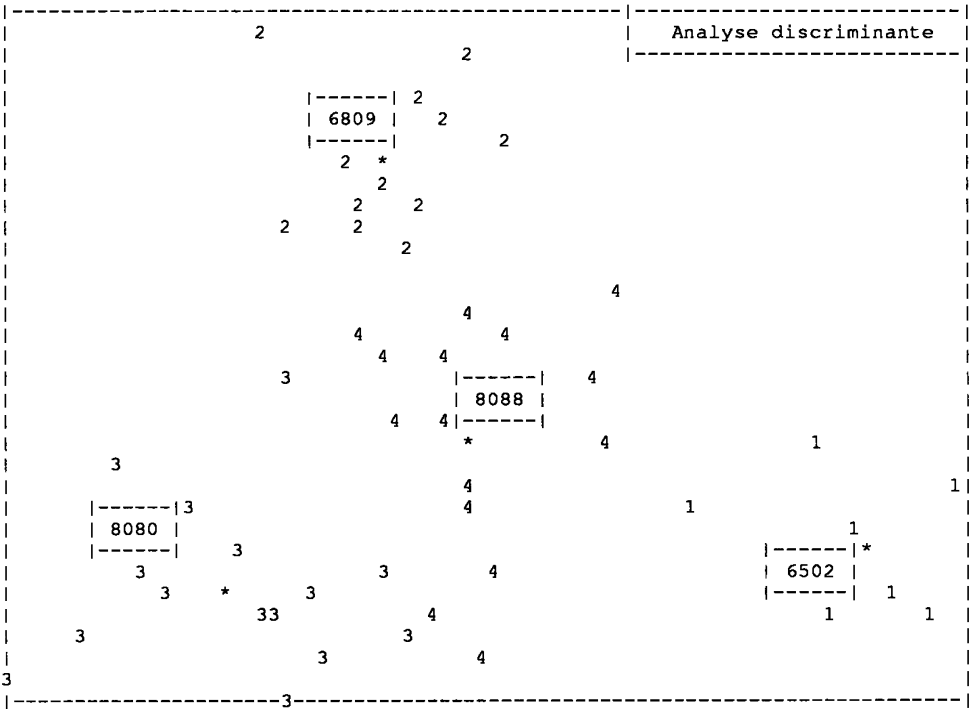
sous-nuage, le 8080 et le 8088 se mêlent quelque peu.

2.2 Analyse factorielle discriminante

Nous utilisons le programme MAHAL3 de J.-M. Romeder, programme de discrimination linéaire pas à pas fondé sur l'algorithme de Sebestyen. À chaque pas, on inclut dans l'analyse la variable maximisant le critère $\text{trace}(Tq^{-1}Bq)$, où Tq est la matrice déduite de la matrice de covariance totale T en ne considérant que les termes relatifs aux q variables prises en compte dans l'analyse; et Bq est la matrice déduite de façon identique de la matrice de covariance inter-classe B . Les individus sont affectés à chacun des groupes, ici quatre, en fonction de leur distance d'inertie (métrique T^{-1} , dite de Mahalanobis) aux centres de gravité de ces groupes.

Nous nous bornerons à publier les résultats les plus intéressants, obtenus au pas n° 6. Six variables ont été entrées dans l'ordre : B2, B1, B0, B7, B5, B4: seules B3 et B6 ne sont pas prises en compte. Le pourcentage d'individus bien classés est de 100%. Pour les trois premiers axes, les pouvoirs discriminants (définis comme les valeurs propres) sont respectivement de 0,84; 0,78 et 0,51. Ce résultat remarquable apparaît clairement sur le graphique d'A.F.D., où les individus (segments de code) afférents aux microprocesseurs 6502, 6809, 8080

et 8088 sont repérés respectivement par les chiffres 1, 2, 3 et 4. Le centre de chaque groupe est figuré par un astérisque. On constate qu'il n'y a plus de



chevauchement; L'axe horizontal oppose le 8080 (à gauche) au 6502 (à droite); l'axe vertical oppose le 6809 aux trois autres microprocesseurs.

Connaissant les composantes des deux premiers vecteurs propres sur les 6 variables retenues, on peut projeter sur le plan (1,2) le point figuratif d'un segment de code quelconque *a*, et d'après la place de ce point reconnaître à quel microprocesseur le code est destiné (à supposer qu'il s'agisse d'un des quatre microprocesseurs considérés ici). De façon précise on a:

F1 =
(-0,007.B2)+(-0,052.B1)+(-0,073.B0)+(0,046.B7)+(0,067.B5)+(0,031.B4);

F2 =
(0,099.B2)+(0,009.B1)+(-0,058.B0)+(0,001.B7)+(0,026.B5)+(-0,012.B4);

formules où on a noté B_x la fréquence avec laquelle le chiffre x est mis à 1 dans les octets du segment de code *a*. Éventuellement, on pourra procéder au calcul de la distance de Mahalanobis entre *a* et les centres des quatre classes.

Afin d'apprécier la sûreté de la discrimination, on a appliqué la méthode de l'échantillon d'épreuve. Pour cela, on a retiré du tableau des données les 10 individus suivants, prélevés dans toutes les classes:

{65f, 65g, 68k, 68l, 80m, 80n, 80o, 88m, 88n, 88o};

et les frontières discriminantes ont été recalculées d'après les 39 individus restants. Sans entrer dans le détail, résumons les résultats obtenus.

Jusqu'au 4-ème pas, les variables retenues sont les mêmes: B2, puis B1, B0, B7. Au cinquième pas, est introduite B6 (et non B5); finalement, on introduit successivement B4, B5 et B3. Au pas 6, le taux de bien classés est 92,3% dans l'échantillon de base et 80% dans l'échantillon d'épreuve. Au pas 8, on atteint respectivement 100% et 90%; c'est-à-dire que seul est mal classé un individu du 6502 affecté au 8088.

3 Analyse des correspondances et classification ascendante hiérarchique

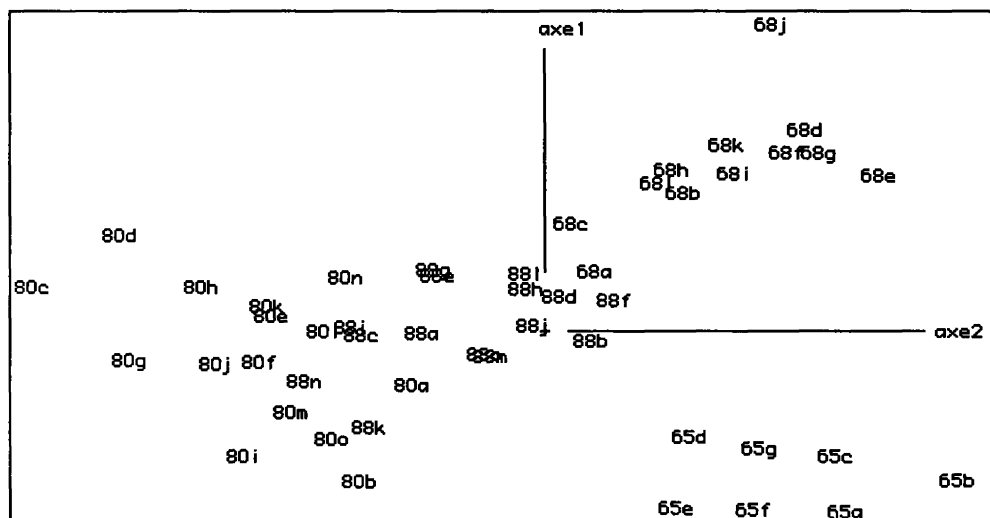
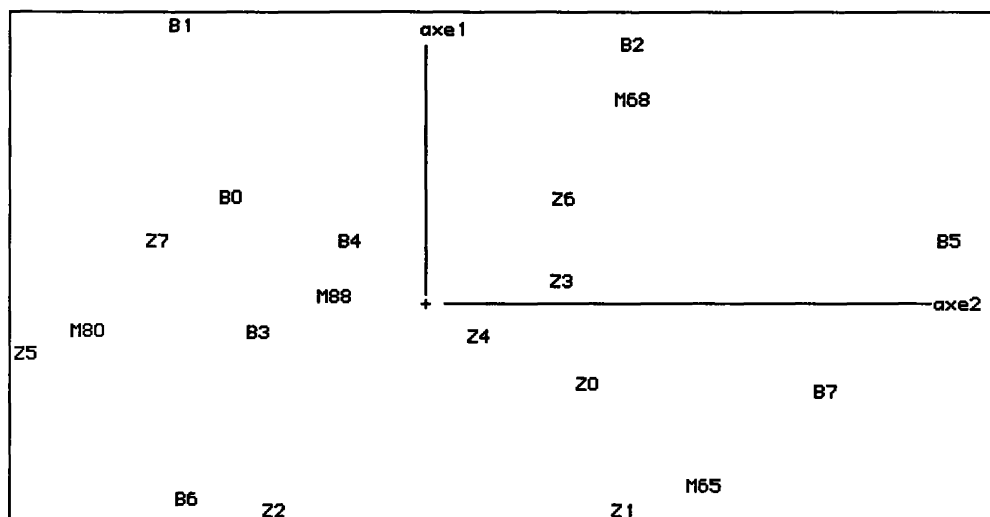
En traitant de plusieurs manières les mêmes données, on peut apprécier les difficultés du problème de discrimination que nous avons résolu avec un remarquable succès sur notre échantillon.

3.1 Analyse des correspondances

On a utilisé le programme qorils, écrit par J.-P. Benzécri pour le micro-ordinateur Macintosh.

Le tableau de base reste celui publié au §1; mais d'une part ce tableau est dédoublé, d'autre part on lui adjoint 4 nouvelles lignes afférentes chacune à un microprocesseur. De façon précise on associe à chaque colonne Bx une colonne complémentaire Zx, mesurant la fréquence du zéro en position x: $k(i, Zx) = 1 - k(i, Bx)$. Ceci est nécessaire car l'analyse des correspondances, ne considérant pas les valeurs brutes mais les profils, ne prendra en compte la fréquence des zéros que si leur sont affectées des colonnes. Les lignes M65, M68, M80 et M88 sont créées comme des moyennes des lignes des codes affectés aux processeurs respectifs.

On a fait une analyse avec en lignes principales les 49 lignes des codes et en supplémentaires les quatre lignes M65 à M88. Cette analyse a l'intérêt de montrer suivant quels axes se diversifient les codes de toute destination: en particulier le 1-er axe oppose les variables Bx aux variables Zx, donc les codes riches en 1 aux codes riches en 0 (en toute position dans l'octet). Mais pour séparer les codes suivant leur destination, il est plus efficace de prendre pour tableau principal le tableau des 4 lignes M, et de projeter en supplémentaire les 49 codes.

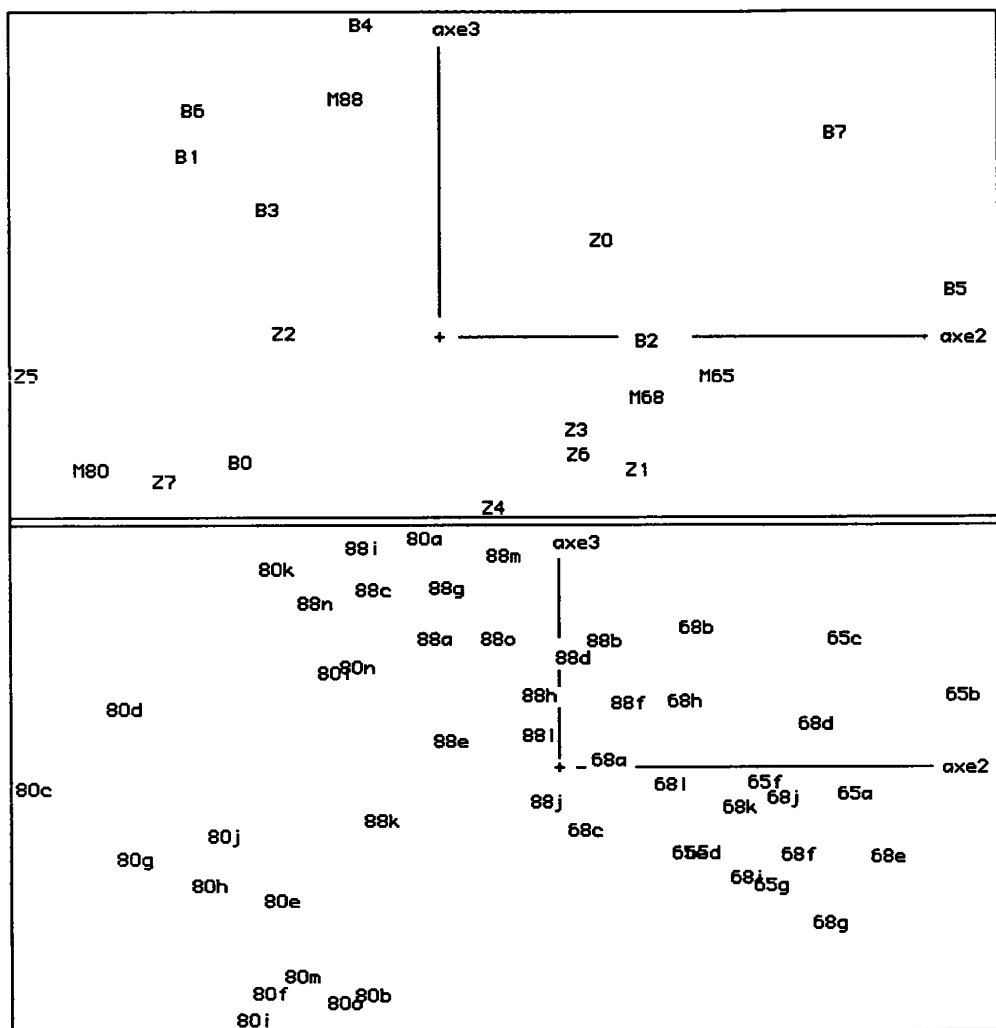


Asselin de Beauville : Identification du microprocesseur

Avec 4 lignes, on a seulement 3 facteurs, donnant une représentation du tétraèdre des 4 microprocesseurs M.

On a représenté les plans (1,2) et (1,3) en séparant, pour plus de clarté, le graphique des éléments supplémentaires de celui des éléments principaux. Le lecteur trouvera dans la représentation simultanée des points M et des variables Bx et Zx la caractérisation approchée de ceux-là par celles-ci.

Dans le plan (1,2) les codes du 6502 se séparent nettement; ceux du 6809 forment un amas qui touche à l'aire des codes du 8080 et du 8088.

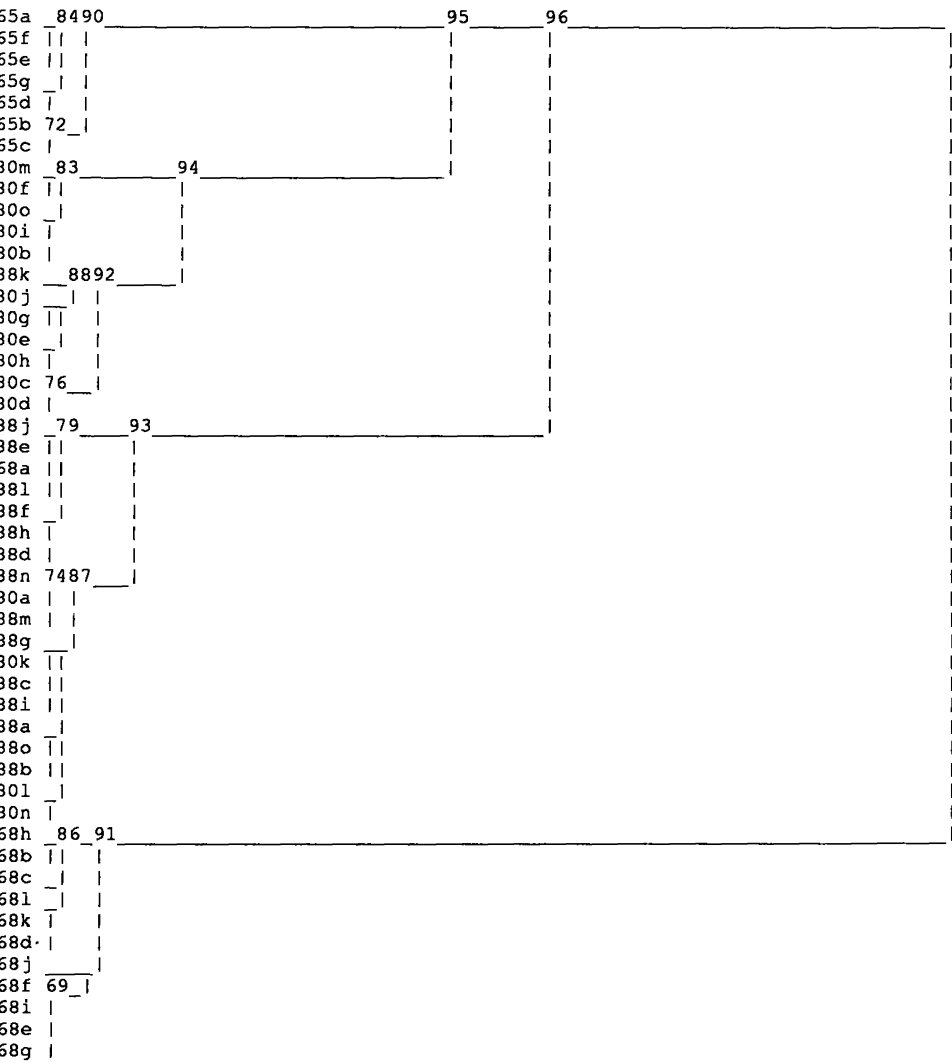


Asselin de Beauville : identification du microprocesseur

Dans le plan (1,3), 8080 et 8088 se séparent assez bien; mais non sans quelques échanges que la classification automatique permettra de retrouver et de préciser.

3.2 Classification ascendante hiérarchique

Ici, on utilise le programme CHRBZ de J.-P. Benzécry. Les 49 codes sont classés d'après leurs coordonnées sur les 3 axes factoriels du §3.1. Si l'on retient la partition en 6 classes {90, 83, 92, 79, 87, 91} définie par les 5 nœuds les plus hauts, on a une classe pure, 90, pour les 7 codes du 6502; une autre classe pure, 91, formée par 11 des 12 codes du 6809; et, ailleurs, quelques



mélanges. Les individus 68a, 80a, 80k, 80l, 80n, 88k se présentent comme des frontaliers susceptibles d'être mal classés, quelle que soit la méthode utilisée, si la composition de l'échantillon en donne l'occasion.

En revenant aux graphiques issus de l'analyse des correspondances, et, plus particulièrement, au plan (1,3), on voit comment des individus (codes) sont échangés entre les zones du 8080 et du 8088.

=====												
c	Partition en 6 classes : Sigles des individus de la classe numero c											
90		65a	65f	65e	65g	65d	65b	65c				
83		80m	80f	80o	80i	80b						
92		88k	80j	80g	80e	80h	80c	80d				
79		88j	88e	68a	88l	88f	88h	88d				
87		88n	80a	88m	88g	80k	88c	88i	88a	88o	88b	80l 80n
91		68h	68b	68c	68l	68k	68d	68j	68f	68i	68e	68g
=====												
90					95				96			
83					94							
92												
79					93							
87												
91												

ci dessus l arbre de la partition en 6 classes

4 Conclusion

Une excellente discrimination a été obtenue au §2.2 entre les codes destinés aux quatre microprocesseurs considérés. L'interprétation en terme de fréquence des 1 et des 0 dans les diverses positions de l'octet est claire et confirmée par toutes les analyses. Il vaudra la peine de poursuivre la recherche en tabulant d'avantage de données, (c'est-à-dire de segments de code, destinés éventuellement à d'autre processeurs), en faisant remonter l'interprétation aux jeux d'instructions, en utilisant non 8 ou 16 variables mais les 256 valeurs numériques possibles de l'octet.

Remerciements: Les auteurs tiennent à remercier le Pr. J.-P. Benzécri pour les améliorations qu'il a apportées à la première version de cette note.

Référence bibliographique

J.-M. Romeder : Méthodes et programmes d'analyse discriminante, DUNOD, Paris, 1973.