REVUE DE STATISTIQUE APPLIQUÉE

Y. MISITI
G. OPPENHEIM
J. M. POGGI

Représentation des connaissances dans les systèmes à base de règles et d'objets pour la statistique

Revue de statistique appliquée, tome 40, n° 2 (1992), p. 99-108
<http://www.numdam.org/item?id=RSA_1992__40_2_99_0>

© Société française de statistique, 1992, tous droits réservés.

L'accès aux archives de la revue « Revue de statistique appliquée » (http://www.sfds.asso.fr/publicat/rsa.htm) 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/

Rev. Statistique Appliquée, 1992, **XXXX** (2), 99-108

REPRÉSENTATION DES CONNAISSANCES DANS LES SYSTÈMES À BASE DE RÈGLES ET D'OBJETS POUR LA STATISTIQUE

Y. MISITI (1), G. OPPENHEIM (1 & 2), J.M. POGGI (1 & 3)

(1) Université Paris Sud, URA D 0743, Lab. Stat. Appli. Mathématique, bat. 425, 91405 ORSAY cedex
(2) Université Paris V
(3) Université Paris X

RÉSUMÉ

Cet article propose quelques outils méthodologiques pour la représentation des connaissances dans les systèmes experts en statistique. Ce travail s'appuie sur l'expérience acquise lors de la réalisation du système expert GEPETTO en conception de lois de commande optimale en automatique . On examine ici la validité de l'approche mise en oeuvre pour un problème statistique de modélisation de séries chronologiques ; le logiciel MANDRAKE est analysé.

Mots-clés : *système expert, représentation des connaissances, objets, règles, CAO, statistique.*

1. Introduction

L'intérêt pour les systèmes experts en statistique se généralise aujourd'hui, comme en témoignent la mise en place du programme européen DOSES et la variété des thèmes abordés. On peut citer : la régression ([GALE 86]), les modèles linéaires ([WOLSTENBOLME 88]), l'analyse de données ([DAMBROISE 86], [KLOSGEN 87]), la statistique non paramétrique ([HAND 87]), la modélisation de séries chronologiques ([AZENCOTT 88]).

L'objet de cet article est de proposer quelques outils méthodologiques pour la représentation des connaissances en statistique.

Ce travail s'appuie sur l'expérience acquise lors de la réalisation d'un système expert en conception de lois de commande optimale en automatique. Une première version de ce système a été réalisée pour l'Aérospatiale (*cf.* [MISITI 87]) avec Prolog et Fortran. Une deuxième version a été mise au point avec des outils plus récents de l'intelligence artificielle (*cf.* [MISITI 91]). Pour ce faire, nous avons recensé, analysé, typé et classifié les connaissances utilisées pour mettre au point une loi de commande. Dans le système GEPETTO, les modes de représentation

utilisés sont ceux du générateur à base de règles et d'objets NEXPERT-OBJECT (*cf.* [NEXPERT]) pour les parties symboliques et du logiciel scientifique MATLAB (*cf.* [MATLAB]) pour les algorithmes numériques.

Le problème du domaine statistique étudié est la modélisation de séries chronologiques par le logiciel MANDRAKE (*cf.* [AZENCOTT 88], [MANDRAKE]).

Les logiciels conventionnels privilégient, en général, les connaissances d'origine mathématique. Le mode de représentation adéquat est alors l'algorithme numérique pour lequel des techniques sont disponibles et bien connues.

Néanmoins, des connaissances indispensables pour la résolution du problème sont souvent absentes ou insuffisamment dégagées, dans ce type de système. En particulier les connaissances concernant :
– le savoir-faire et l'expérience du domaine d'analyse ;
– le choix des paramètres d'entrée des algorithmes ;
– la façon d'enchaîner les algorithmes, *i.e.* la démarche générale de résolution (non déterministe) ou les dépendances d'informations numériques (déterministe) ;
– la façon d'interpréter les résultats d'un algorithme. Peu de travaux publiés analysent la connaissance incluse dans un problème statistique (*cf.* [VAN DEN BERG 90], [WITTKOWSKI 90]). Morcelé, non intégré à son domaine d'étude, celui-ci est restreint à son algorithmique. Sous cette forme, le problème est mutilé et les difficultés d'application de la statistique proviennent sans doute de cet éclatement. Nous cherchons par ce travail à rassembler les morceaux habituellement épars.

La première partie du texte précise, pour chacun des deux problèmes : la conception de lois de commande et la modélisation de séries chronologiques, la nature de la tache à accomplir par le système informatique, et à dégager puis structurer les connaissances utiles. La seconde partie du texte présente les outils sur lesquels est basée la représentation des connaissances. Sont utilisés les modes de représentation suivants : règles de production (*cf.* [FARRENY 87]) et objets (*cf.* [MASINI 89]).

2. Deux problèmes : la conception de lois de commande et la modélisation de séries chronologiques

2.1 Traits communs aux deux problèmes

Les deux problèmes ont plusieurs traits communs :
– ce sont des problèmes de conception restreinte, consistant à choisir la valeur d'un nombre fini de paramètres dits de réglage ;
– la démarche de résolution comporte, dans chacun des cas, des phases de choix de ces paramètres, des phases de propagation de ces choix, des phases de diagnostic et un aspect itératif de la mise au point ;
– enfin, la mise au point requiert des connaissances de nature et d'origine variées : théorèmes, savoir-faire expérimental, stratégies de résolution, heuristiques, algorithmes.

Nous nous concentrerons donc dans ce paragraphe sur la méthode de résolution du problème en hiérarchisant les connaissances. Dans ce cadre, il est alors indispensable de ne pas centrer l'analyse sur les algorithmes qui, dans ces domaines, préexistent.

2.2 Conception de lois de commande

2.2.1 Le problème

Le problème consiste à concevoir (hors ligne) un régulateur de type LQG basé sur un modèle d'ordre réduit, pour contrôler l'état d'un système physique (par exemple un lanceur).

L'évolution de ce dernier est modélisée par un système d'équations différentielles stochastiques linéaires. L'état est seulement partiellement observé et est commandé par un retour d'état estimé, optimal vis-à-vis d'un critère quadratique.

Par conséquent, les paramètres de réglage d'une telle loi de commande sont :
– le pas d'échantillonnage : d, les calculs en ligne sont effectués par des ordinateurs numériques ;
– le modèle d'ordre réduit : MS, afin que le calcul d'une commande soit suffisamment rapide ;
– les matrices de covariance des bruits d'état et de mesure : Se et Sm ;
– les matrices de pondération de l'état et de la mesure dans le critère quadratique : Qe et Qm.

Les critères de qualité d'une loi de commande sont fondés sur :
– le respect des contraintes du système physique ;
– les performances dynamiques du système commandé ;
– la robustesse de la stabilité vis-à-vis de perturbations de modèle ou de perturbations externes.

2.2.2 La démarche de résolution

La démarche de résolution utilisée, dans le système GEPETTO, consiste d'abord à fixer des valeurs initiales des paramètres de réglage de la meilleure qualité possible puis à procéder par améliorations successives.

La stratégie conduit donc à décomposer le problème en quatre sous problèmes hiérarchisés et traités successivement : le choix de d, le choix de MS, le choix de Qe et Qm associé à un problème de commande purement déterministe, le choix de Se et Sm associé à un problème purement stochastique (conception d'un filtre de Kalman). Sur la base de descripteurs permettant de valider une loi de commande, des phases d'amélioration raffinent ensuite le choix des paramètres.

Par exemple, pour le problème d'estimation de l'état par un filtre de Kalman, les valeurs initiales de Se et Sm sont choisies sur les bases suivantes :
– tenir compte des bruits connus et modélisés (bruits de capteurs ou équations estimées par des modèles stochastiques) ;

– assurer la convergence de l'estimateur *i.e.* la stabilité du filtre ;
– fixer le niveau du rapport signal/bruit.

Les améliorations intègrent la modélisation stochastique des méconnaissances paramétriques et utilisent des procédures connectant les problèmes d'estimation et de commande.

2.3 Modélisation de séries chronologiques

2.3.1 Le problème

Le problème consiste, en vue de la prévision et de la dessaisonnalisation, à ajuster une série chronologique $Y = (Y(t);\ t = 1, \ldots, T)$, par un modèle ARIMA de la forme suivante :

$$P(B)PS(B)F(B)Y(t) = Q(B)QS(B)W(t),$$

avec $F(B) = (I - B)^d(I - B^s)^D$ tel que $F(B)$ soit stationnaire et où :

P, PS, Q, QS sont des polynômes de degré p, ps, q et qs respectivement ; B est l'opérateur retard et W une innovation.

Les paramètres de réglage de la modélisation sont donc les degrés d, s, D, p, ps, q, qs.

Les critères de validité d'une modélisation sont fondés sur :
– la qualité des estimations de W ;
– les degrés des polynômes P, PS, Q, QS qui doivent être faibles ;
– la robustesse des estimateurs des coefficients de ces polynômes.

2.3.2 La démarche de résolution

MANDRAKE est un logiciel développé à l'aide d'outils conventionnels, mais est étudié ici car sa démarche de résolution est originale.

D'une part, les trois phases classiques de la modélisation : stationnarisation (choix de d, s et D), identification (choix de p, q, ps, qs) et estimation, sont complètement séquentialisées, contrairement à une modélisation «à la main» où de fréquents retours en arrière sont indispensables. La stratégie conduit à surexploiter la phase d'identification, peu coûteuse en temps de calcul, de sorte que la phase d'estimation fine des paramètres du modèle (maximisation de la vraisemblance, très onéreuse) soit effective seulement pour les modèles qui se sont révélés les plus prometteurs. Mentionnons qu'une estimation grossière, mais suffisante dans cette phase, des modèles candidats est obtenue par un algorithme de moindres carrés.

D'autre part, l'heuristique choisie : «générer et tester», consiste à créer progressivement des modèles et à tester leur admissibilité en phase d'identification, ce qui revient à énumérer implicitement tous les modèles.

3. Représentation et organisation des connaissances

3.1 Principe d'organisation des connaissances

Dans cette famille de problèmes de type CAO, l'une des principales difficultés réside dans le fait qu'une règle de l'expert engendre, en général, un volumineux ensemble d'objets et de traitements, secondaires pour lui mais indispensables au fonctionnement correct du système informatique.

Les principes de représentation retenus sont les suivants :

P1 : structurer et typer les connaissances ;
P2 : séparer les connaissances de haut niveau conceptuel de celles de bas niveau ;
P3 : séparer les connaisances symboliques des connaissances numériques ;
P4 : assigner aux connaissances stratégiques un rôle prééminent dans la dynamique du système informatique ;
P5 : la vision externe des connaissances, par l'expert ou l'utilisateur, doit être un sous-ensemble convenablement choisi de la représentation interne, et non comme souvent, le produit d'un interface ad-hoc difficile à maintenir.

L'outil de base pour la représentation des connaissances est une classification de celles-ci sur trois critères :

– domaine/problème ;

ce premier critère distingue les connaissances propres au domaine de base (par exemple l'automatique, la statistique) de celles liées au problème étudié (par exemple : la commande d'un lanceur, la modélisation d'une série de consommation d'électricité pour la prévision).

– rationnelle/empirique ;

– passive/active/stratégique ;

ce dernier critère, classique en intelligence artificielle, induit une hiérarchie des connaissances par puissance déductive croissante.

Pour respecter ces principes, nous disposons des modes de représentation suivants : les règles de production, les objets et les algorithmes.

Les relations entre les modes de représentation et les classes de connaissances s'en déduisent :

– les connaissances stratégiques sont représentées par des règles qui assurent le pilotage du système informatique ;
– les connaissances passives sont représentées par des classes et structurées en un graphe d'objets sur lequel opère l'héritage ;
– les connaissances actives, non intégrées à des algorithmes numériques, sont représentées par des règles.

Les algorithmes numériques sont représentés (en plus de leur code) par les démons et les sources d'information attachés aux attributs des objets sur lesquels opère l'envoi de message.

Précisons maintenant la représentation des connaissances.

3.2 Représentation par règles

Parmi les connaissances représentées par des règles de production, nous distinguerons les stratégiques des actives. Les premières sont essentiellement liées à une décomposition en sous-problèmes dont le traitement est purement symbolique. Les secondes, au contraire, font le lien entre les traitements symbolique et numérique.

3.2.1 Les connaissances stratégiques

Elles représentent la stratégie de conception, par niveaux d'abstraction décroissants, et sont codées dans le langage de l'expert. La démarche de résolution est donc codée par un ensemble de règles de réécriture autonome et lisible, dirigeant la dynamique d'exécution.

Par exemple, pour la modélisation de séries chronologiques, la stratégie de MANDRAKE peut être représentée par les règles suivantes (ceci est déduit de la documentation du logiciel *cf.* [MANDRAKE]), figurées par un arbre (se lisant de gauche à droite, puis de haut en bas) :

Modélisation ARIMA de Y (1)

Identification ARMA de $F(B)Y$ (10)

Stationnarisation de Y (2)

Choix de $F : d, s, D$ (3)

Choix de p, q, ps, qs (11)

Générer des filtres (4)

Tester les filtres (7)

Choix de p, q (12)

Améliorer modèles, choix de ps, qs. (15)

Détecter tendance et saisonnalités (5)

Générer valeurs de d, s, D (6)

Evaluer filtres (8)

Choisir filtres (9)

Générer AR purs MA purs, ARMA (13)

Tester modèles (14)

Générer, corriger modèles (16)

Tester modèles (17)

Classer modèles (18)
Sélectionner modèles (19)
Interclasser modèles utilisateur (20)

Estimer modèles (21)
Classer modèles (22)
Sélectionner modèles (23)

3.2.2 Les connaissances actives

Celles-ci correspondent en fait à la réécriture des hypothèses terminales du graphe engendré par les règles stratégiques. Elles ont pour rôle de valuer les paramètres de réglage ou de demander l'évaluation d'un objet. Elles ont donc pour effet de transmettre momentanément le contrôle aux objets qui, en particulier, propageront les modifications induites, et exécuteront les algorithmes nécessaires puis rendront le contrôle aux règles chargées de résoudre les sous-problèmes en attente.

Par exemple, dans le graphe précédent l'hypothèse 14 : tester modèles se réécrit ainsi :
- effectuer une estimation rapide ;
- tester l'admissibilité ;
- étudier la variance résiduelle ;
- étudier la blancheur des résidus ;
- retenir modèles.

Ces hypothèses, non réécrites, activent donc les sources d'information associées à chacune d'elles, ce mécanisme est détaillé en 3.3.2.

3.3 Représentation par objets

Dans ce paragraphe la façon de structurer la base d'objets est décrite ainsi que la manière de définir leur comportement.

3.3.1 Structures et types de données

La base d'objets est constituée de deux ensembles d'entités : les objets et les classes. La définition des classes reflète les types de données du domaine considéré, les objets sont simplement les instances des classes.

La base est structurée en un graphe à l'aide de trois relations : est une partie de, est une instance de, est un composant de. A tout objet ou classe, sont associés des slots définissant autant d'aspects ou de propriétés de l'entité. Ainsi les instances de la classe «série» auront pour slots : «valeur», «longueur», «unité», et pour sous-objet le «corrélogramme» avec les slots : «valeur», «pics», entre autres. Ceci permet de représenter les connaissances passives.

Pour définir le comportement de ces slots, à chacun d'eux est attaché un meta-slot. Le meta-slot définit d'une part, les sources d'information prescrivant les moyens de valuer le slot (par exemple un algorithme pour le slot «valeur» du «corrélogramme») et d'autre part, les démons donnant les actions à éxécuter en cas de modification de la valeur du slot (par exemple l'invalidation du corrélogramme d'une série en cas de changement de sa valeur).

L'héritage et l'envoi de message sont les deux mécanismes d'inférence liés à cette structuration.

Formulons une dernière remarque sur le type des slots et les supports physiques des informations. A chaque connaissance passive est associée un objet ayant au moins un slot valeur dont le domaine est défini ainsi :

– si la valeur est traitée symboliquement, elle a le type primitif désiré ;

– si la valeur est traitée uniquement numériquement, elle est de type complexe non primitif (matrice, système, modèle) et le support physique des informations est une base de données externe, commune aux traitements symboliques et numériques. Le type du slot est dans ce cas booléen indiquant si la valeur externe actuelle est correcte. Le mécanisme de maintenance de la vérité est précisé dans le paragraphe suivant.

3.3.2 Messages et méthodes

Il reste à décrire la programmation des meta-slots. Celle-ci concerne, pour l'essentiel, les connaisances passives traitées numériquement.

En effet, les connaissances passives qui ne sont traitées que symboliquement sont aisées à valider puisque codées dans des règles. Les outils logiques de haut niveau sont donc utilisables : réseau des règles (statique) et traces (dynamique).

En revanche, une masse très importante de connaissances passives sont traitées par de nombreux algorithmes. Il faut donc définir des outils permettant d'assurer la calculabilité de toute information, d'obtenir une modularité maximale et de maintenir la cohérence chronologique de la base d'objets.

Introduisons pour cela un graphe.

3.3.2.1 Un graphe de dépendance

Le graphe de dépendance choisi est défini ainsi :

– X l'ensemble des sommets : les slots d'objets dont la valeur est traitée numériquement ;

– les arcs sont donnés par la relation successeur (notée succ) suivante : si $x = f(x_1, \ldots, x_n)$ alors x appartient à succ (x_i) pour $i = 1, \ldots, n$, où f est un algorithme numérique choisi de façon «modulaire» *i.e.* en ne tenant compte ni de la décomposabilité des x_i (prise en compte par le réseau d'objets), ni de leur calculabilité (assurée par d'autres dépendances).

La manière la plus naturelle de définir les dépendances dans ce contexte consiste à centrer l'analyse sur un x de X, et lui faire correspondre deux sous-ensembles de X :

– pred (x) : celui des éléments nécessaires au calcul de x (en notant pred la relation prédecesseur associée à succ) ;

– succ (x) : celui des éléments dont la valeur dépend de x.

3.3.2.2 Représentation du graphe

Un codage efficace du graphe s'en déduit aisément en exploitant un des principes clés de la programmation par objets.

Le meta-slot de chaque x de X est défini ainsi :

1) Les sources d'information, déclenchées lorsque la valeur de x est inconnue et demandée, contiennent :
- les envois de message aux éléments de pred (x) afin qu'ils se calculent;
- l'évaluation de f(pred (x)) où f est un algorithme;
- l'écriture éventuelle sur une base de données externe;
- la valuation de x;

2) Les démons, déclenchés lorsque la valeur de x est modifiée, contiennent la remise à «inconnue» des éléments de succ (x).

Ceci assure la fiabilité et la cohérence de la base d'objets puisque toute modification d'une donnée est immédiatement propagée dans le graphe. En outre, cette programmation conduit à gérer de façon économique les recalculs. En effet, la modification d'une donnée entraîne que ses successeurs sont à réévaluer mais, le calcul effectif n'aura lieu que si un message de demande de valeur est reçu.

Conclusion

Ce papier illustre l'apport des modes de représentation actuels à l'élaboration de systèmes experts en statistique. Ils permettent de représenter l'ensemble des connaissances incluses dans un problème statistique, y compris le savoir-faire et l'expérience du domaine d'étude.

En particulier, nous préconisons que les connaissances codées sous forme de règles ou structurées dans la base d'objets correspondent au niveau d'abstraction de l'expert ou de l'utilisateur. Ce choix utilise à plein les interfaces graphiques disponibles dans la plupart des générateurs de systèmes experts. A l'inverse, les connaissances subalternes, négligées par les experts, par économie de pensée, sont «cachées» dans les meta-slots et les messages.

Bibliographie

[AZENCOTT 88] R. Azencott, Y. & B. Girard, R. Astier, P. Jakubowicz, M. Baudin, M.M. Martin, *MANDRAKE : Logiciel expert d'analyse de séries chronologiques*, International Symposium on Forecasting - Amsterdam, june 1988

[DAMBROISE 86] E. Dambroise, P. Massote, *MUSE : An expert system in statistics*, Proceedings of the 7th COMPSTAT, Rome, 1986

[FARRENY 87] H. Farreny, M. Ghallab, *Eléments d'intelligence artificielle*, Hermès, 1987

[GALE 86] W.A. Gale ed., *Artificial intelligence and statistics*, Reading, Addison-Wesley, 1986

[HAND 87] D.J. Hand, *A statistical knowledge enhancement system*, J. R. Statist. Soc. A, 150, p. 334-345, 1987

[KLOSGEN 87] W. Klösgen, G. Wenzel, *On the representation of expert-knowledge in data analysis systems*, Proceedings of the DOSES seminar, p. 316-334, EUROSTAT, Luxembourg, dec. 1987

[MANDRAKE] Universités Paris I et XI, EDF, CNET, *MANDRAKE, Manuel d'utilisation, mars 1990*, distribution CEMS

[MASINI 89] G. Masini, A. Napoli, D. Colnet, D. Léonard, K. Tombre *Les langages à objets*, InterEditions, 1989

[MATLAB] Mathworks Inc. *Matlab reference manual*, 1990

[MISITI 87] Y. Misiti, J.M. Poggi, R. Astier, J. Dupont, G. Oppenheim, F.Y. Villemin *Un système expert en conception de lois de pilotage*. Actes des 7 èmes journées internationales : systèmes experts et applications, p. 1053-1075, Avignon, juin 1987

[MISITI 91] Y. Misiti, J.M. Poggi *GEPETTO : An expert system for Computer Aided Control System Design*, Proceedings of the European Control Conference, Grenoble, july 91.

[NEXPERT OBJECT] Neuron Data Inc. *Nexpert Object and Nexpert Forms reference manuals, 1990*

[VAN DEN BERG 90] G.M. Van den Berg, R.A. Visser *Knowledge modelling for statistical consultation systems; two empirical studies*. Proceedings of the 9 th COMPSTAT, p. 75-80, Dubrovnik, 1990

[WITTKOWSKI 90] K.M. Wittkowski *Statistical knowledge-based systems – Critical remarks and requirements for approval* Proceedings of the 9 th COMPSTAT, p. 49-56, Dubrovnik, 1990

[WOLSTENBOLME 88] Wolstenbolme, C.M. O'Brien, J.A. Nelder *GLIMPSE : a knowledge-based front end for statistical analysis* Knowledge-Based Systems, 1, p. 173-178, 1988