

Approches symboliques et sous-symboliques en représentation de connaissances

Synthèse de lectures

de

Benoit Lavoie

benoit@benoit-lavoie.ca

Programme de Doctorat en Informatique Cognitive

Université du Québec à Montréal

8 février 2007

Table des matières

1. DESCRIPTION DE LA PROBLÉMATIQUE	1
2. INTRODUCTION.....	1
3. NOTIONS DE BASE.....	1
4. APPROCHES SYMBOLIQUES EN REPRÉSENTATION DE CONNAISSANCES	2
5. APPROCHES SOUS-SYMBOLIQUES EN REPRÉSENTATION DE CONNAISSANCES.....	5
6. AVANTAGES ET LIMITES DES APPROCHES SYMBOLIQUES ET SOUS-SYMBOLIQUES EN REPRÉSENTATION DE CONNAISSANCES.....	7
7. APPROCHES HYBRIDES SYMBOLIQUES/SOUS-SYMBOLIQUES EN REPRÉSENTATION DE CONNAISSANCES.....	15
8. REMERCIEMENTS.....	18
9. RÉFÉRENCES.....	18

1. Description de la problématique

On distingue deux principaux types d'approches en représentation de connaissances: les approches symboliques d'une part et les approches sous-symboliques d'autre part. Ce document décrit ces approches et identifie leurs avantages et limites. Ce document décrit également comment les approches symboliques et sous-symboliques peuvent être conjuguées à l'aide d'un exemple pratique.

Ce document est une synthèse de lectures préparée dans le cadre du programme de Doctorat en Informatique Cognitive de l'Université du Québec à Montréal (UQAM).

2. Introduction

Pour répondre à la problématique nous décrivons préalablement quelques notions qui nous aideront à situer notre réponse dans un contexte général: la notion de représentation de connaissances, les notions d'approches symboliques et sous-symboliques et la classification des approches selon cette dichotomie.

Les références utilisées dans ce document consistent en partie en notes de cours présentées dans le cadre du programme de Doctorat en Informatique Cognitive de l'UQAM: les notes de deux cours en particulier (introduction à l'informatique cognitive et séminaire en apprentissage automatique) nous sont apparues être particulièrement pertinentes pour répondre à la problématique. Plusieurs références concernent des documents que nous avons consultés en ligne pour nos recherches. Des liens accompagnant les références donnent accès en ligne à plusieurs des références citées.

La section 3 décrit la notion de représentation de connaissances en informatique et introduit les approches en représentation de connaissances. La section 4 décrit les approches symboliques en représentation de connaissances. La section 5 décrit les approches sous-symboliques en représentation de connaissances. La section 6 décrit les avantages et limites des approches symboliques et sous-symboliques de manière générale d'abord, puis de manière détaillée. La section 7 décrit les approches hybrides symboliques/sous-symboliques et présente un exemple particulier de système hybride. Finalement la section 8 contient la liste des principales références.

3. Notions de base

Notion de connaissances en informatique

D'un point de vue informatique, la représentation de connaissances est liée principalement aux domaines de l'intelligence artificielle et de l'ingénierie des connaissances (Wikipedia, 2006a). Les *connaissances* se définissent (Information Society Technology, 2001) comme des relations, faits, hypothèses, heuristiques, modèles, etc. qui sont dérivés par analyse (formelle ou informelle) ou interprétation de *données* (valeurs, symboles, nombres, ... sans structure et contexte). On distingue les connaissances selon différents critères: connaissances vs méta-connaissances; connaissances explicites vs implicites (tacites); connaissances complètes vs partielles; connaissances certaines vs incertaines; connaissances simples vs complexes; connaissances statiques vs dynamiques; connaissances dépendantes du domaine (situées) vs indépendantes du domaine; etc.

Notion de représentation en informatique

“La notion de représentation est fondamentale pour la modélisation” (Memmi, 2004a) des connaissances; une représentation permet de décrire et de manipuler les connaissances. *“Une représentation est un état partiel d’un système”* référant à des caractéristiques d’éléments (objets ou états) qui sont *“pertinentes pour effectuer une tâche ou résoudre un problème. Une représentation est elle-même un objet pour des mécanismes de traitement sensibles à la nature de cette représentation.”*

“Une représentation est un choix” (Memmi, 2004a) parmi les caractéristiques potentiellement illimitées des connaissances représentées. Le choix d’une représentation donnée détermine les inférences possibles à partir de celle-ci. Une représentation est spécifiée avec un formalisme (ensemble de conventions syntactiques et sémantiques).

On distingue les représentations selon différents critères: représentations locales vs distribuées (ou modulaires vs non modulaires); représentations déclaratives vs procédurales; représentations symboliques vs sous-symboliques (voir détails plus bas); etc.

Approches en représentation de connaissances

On distingue souvent les approches suivantes en représentation de connaissances (MacIntyre, 2000): (i) approches symboliques; (ii) approches sous-symboliques (parfois appelées non symboliques¹); et (iii) approches hybrides symboliques / sous-symboliques. Les trois prochaines sections décrivent ces approches et les avantages et limites des approches symboliques et des approches sous-symboliques au niveau des représentations de connaissances et de leurs traitements.

4. Approches symboliques en représentation de connaissances

Aperçu

Les approches symboliques en représentation de connaissances sont basées sur des représentations dont le formalisme représente les connaissances explicitement par un ensemble de symboles. Ces approches sont généralement caractérisées par des représentations de connaissances locales (i.e. les connaissances sont circonscrites à des éléments particuliers de représentation).

Un *symbole* est une entité atomique désignant un concept (objet ou état) (Memmi, 2004a). Un symbole peut être combiné avec d’autres symboles afin de former des expressions structurées complexes référant à des concepts plus complexes. Un symbole peut être manipulé par un système physique de symboles (e.g. ordinateur). Dans une représentation symbolique des connaissances, les connaissances sont explicitement reliées à la forme (syntaxe) et au sens (sémantique) des symboles: e.g. langages naturels, langages de logique formelle, etc.

¹ En informatique et en philosophie les termes *sous-symbolique* et *non symbolique* sont parfois employés comme synonymes: e.g. voir <http://adsabs.harvard.edu/abs/1994RSPTA.349...87W>
http://www.comp.lancs.ac.uk/computing/research/aai-aied/people/paulb/old243prolog/subsection3_1_2.html

Exemples d'approches symboliques

Les approches symboliques en représentation et traitement des connaissances sont variées (Ali, 1999; Memmi, 2004a); elles peuvent être basées sur les systèmes experts, les logiques formelles, les arbres de décisions², les ontologies, les schémas, les bases de cas, etc. Les approches symboliques sont souvent complémentaires: e.g. les systèmes experts peuvent utiliser des règles basées sur une logique formelle et qui sont dérivées d'arbres de décisions. Différentes approches symboliques sont décrites dans le reste de cette section. Les avantages et limites liés aux approches symboliques sont décrits à la section 6.

Représentation et traitement de connaissances avec les systèmes experts

Les connaissances dans un système expert consistent en un ensemble de faits et de règles (implications de type SI-ALORS; Memmi, 2004a). Ces connaissances sont généralement déclaratives et sont souvent regroupées dans une base de connaissances.

Ces connaissances sont traitées par un moteur d'inférence dans un contexte de questions et réponses entre l'utilisateur et le système. Le moteur d'inférence est souvent basé sur une logique formelle (e.g. logique classique ou logique floue) et/ou sur les arbres de décisions (voir plus bas). Le moteur d'inférence est souvent générique et directement adaptable à de nouveaux domaines. Il utilise le chaînage avant (partant des faits et règles afin de déduire des conclusions qui peuvent s'ajouter aux faits) et/ou le chaînage arrière (partant d'une conclusion, déterminer si on peut la prouver à partir des règles et faits). L'acquisition des faits et règles se fait généralement manuellement en analysant les questions et réponses d'experts dans un domaine donné. Les modifications aux faits et règles se font aussi généralement manuellement.

Représentation et traitement de connaissances avec les logiques formelles

Une logique formelle consiste en un langage et un mécanisme de raisonnement formel (Memmi, 2004a). Ceux-ci établissent des critères de consistance et de validation d'expressions à partir de leurs formes plutôt qu'à partir de leurs contenus. Les différents types de logique formelle (logique classique, logique floue, logiques non monotones, etc.) se distinguent par les propriétés de leurs langages. (Certains de ces langages sont discutés plus en détails aux sections 3 et 4 de ce document.)

Le mécanisme de raisonnement lié aux logiques formelles est basé sur des règles et techniques d'inférences. Il permet de valider des conclusions données (e.g. règle de modus tollens de la logique classique) ou d'inférer de nouvelles conclusions à partir des faits et règles connus (e.g. règle de modus ponens de la logique classique) (voir section 3).

Représentation et traitement de connaissances avec les arbres de décisions

Un arbre de décisions est une représentation arborescente où: (i) les nœuds internes sont utilisés pour des tests

² Les justifications pour considérer les arbres de décisions comme des représentations symboliques sont décrites plus bas. Cependant, certains auteurs semblent diverger d'opinion sur ce point (voir aussi plus bas).

discriminatoires sur les valeurs des attributs d'un problème; (ii) les branches sont utilisées pour représenter les valeurs alternatives résultant des tests sur les attributs; (iii) les noeuds terminaux représentent des solutions au problème donné (e.g. classes dans un problème de classification); et (iv) l'arbre correspond à un ensemble de règles en forme normale disjonctive (Lounis, 2006a; Lavoie 2006). Les arbres de décisions et les techniques associées aux arbres de décisions sont souvent classifiés parmi les approches symboliques car les arbres de décisions correspondent directement à un ensemble de règles symboliques (<http://www.cse.unsw.edu.au/~waleed/thesis/node36.html>; Lounis, 2006a). Les arbres de décisions sont d'ailleurs dits être des représentations très intuitives à comprendre à cause de cette correspondance avec les règles symboliques (Lounis, 2006a).

Un arbre de décisions est généralement induit à partir d'un ensemble de données, chacune étant représentée par un ensemble de valeurs d'attributs simples (nombre ou chaîne de caractères), et dont l'un des attributs correspond à la solution (classification). Chaque donnée est préalablement spécifiée par (i) un ensemble de valeurs d'attributs simples et (ii) la valeur d'une classe (solution). Plusieurs techniques d'induction sont disponibles. Une technique classique consiste à (i) former la racine de l'arbre à partir de l'attribut le plus discriminatif pour la classification, (ii) créer des sous-branches à partir des valeurs des données pour cet attribut, (iii) appliquer récursivement cette technique à chaque branche pour les attributs restant. Les arbres sont également élagués en acceptant une certaine marge d'erreurs de façon à pouvoir classer les données généralement sans être trop spécifique aux données. Les arbres peuvent être utilisés directement pour la classification ou ils peuvent être convertis de façon triviale en règles qui sont plus facilement manipulables.

Représentation et traitement de connaissances avec les ontologies

Une ontologie est une spécification explicite d'une conceptualisation (Gruber, 1993 – citation Gómez-Pérez et al., 2004). Une ontologie contient des spécifications de propriétés de concepts et de relations entre concepts (Gómez-Pérez et al., 2004; Bouchard & Obaid, 2005). Ces spécifications sont organisées dans une hiérarchie où celles de plus haut niveau sont définies de façon générique et consensuelle afin de faciliter la réutilisation et le partage d'une application à une autre, et d'un groupe de chercheurs à un autre. Selon les ontologies, les spécifications sont plus ou moins formelles et plus ou moins détaillées.

Différents traitements sont disponibles avec différents types d'outils (Gómez-Pérez, 2002): (i) outils de développement fournissant des fonctions d'édition, de navigation, d'importation, d'exportation, etc.; (ii) outils d'intégration fournissant des fonctions de fusion de plusieurs ontologies; (iii) outils d'évaluation fournissant des fonctions de vérification de la consistance des spécifications et de la validité avec des langages donnés; (iv) outils d'annotation fournissant des fonctions reliées directement aux annotations textuelles des ontologies; (v) outils de recherche (requête) fournissant des fonctions reliées à l'accès aux informations contenues dans les ontologies; (vi) outils d'apprentissage fournissant des fonctions reliées à l'acquisition semi-automatique d'ontologie à partir de textes; etc.

Représentation et traitement de connaissances avec les schémas

Un schéma est un ensemble de paires attribut/valeur (*slot/value*) décrivant explicitement une connaissance plus ou moins

complexe où la valeur de chaque attribut est spécifiée soit directement dans le schéma ou par des liens à d'autres schémas (Memmi, 2004a). Les schémas peuvent inclure des liens hiérarchiques et assumer des défauts (pairs attribut/valeur héritées de d'autres schémas). Contrairement aux ontologies, le contenu des schémas est généralement défini de façon ad-hoc, non consensuelle, pour des applications particulières et la spécification des liens hiérarchiques (hiérarchie de schéma) peut être manquante ou incomplète.

Le traitement des schémas inclut généralement des fonctionnalités d'unification et d'héritage. Les schémas sont généralement créés manuellement pour des domaines donnés. Les schémas sont beaucoup utilisés pour le traitement du langage car bien adaptés à la représentation des constituants des phrases et des formes utilisées dans les interfaces homme-machine.

Représentation et traitement de connaissances avec les bases de cas

Dans un système à base de cas, un cas représente comment un problème a été résolu dans le passé (Lounis, 2006b). Un cas possède deux parties: une description de problème et une solution de problème. La description du problème est basée sur la spécification d'un ensemble d'attributs.

Les cas forment une base de connaissances et sont utilisés dans un système à base de cas afin d'aider à résoudre de nouveaux problèmes. Les degrés de similarité entre la description d'un nouveau problème et celles des problèmes passés sont utilisés afin de déterminer la solution à un nouveau problème. Un degré de similarité est déterminé en comparant les attributs des problèmes, en considérant différents poids associés à différents attributs.

5. Approches sous-symboliques en représentation de connaissances

Aperçu

Les approches sous-symboliques en représentation de connaissances sont basées sur des représentations dont le formalisme ne représente pas les connaissances explicitement par un ensemble de symboles. Ces approches sont généralement caractérisées par des représentations de connaissances distribuées (i.e. les connaissances ne sont pas circonscrites à des éléments particuliers de représentation mais elles sont plutôt distribuées dans les représentations globalement).

Les approches sous-symboliques en représentation et traitement de connaissances incluent entre autres les réseaux de neurones artificiels, les algorithmes génétiques et les réseaux bayésiens (Ali, 1999). Ces approches sous-symboliques sont décrites plus en détails dans le reste de cette section. Les avantages et limites liés aux approches sous-symboliques sont décrits plus loin à la section 6.

Représentation et traitement de connaissances avec les réseaux de neurones artificiels

Les réseaux de neurones artificiels sont les représentations sous-symboliques les mieux connues (Ali, 1999) et ils ont certaines similarités avec les systèmes neuronaux biologiques (Memmi 2004b). Un réseau de neurones artificiels est un graphe orienté. À chaque nœud est associé une fonction d'activation prédéterminée (fonction à seuil, fonction linéaire ou

fonction continue). Cette fonction prend comme valeurs d'entrées un vecteur de données numériques pondérées par des poids associés aux arcs d'entrées du nœud. Cette fonction produit en sortie une valeur d'activation numérique servant possiblement de donnée d'entrée à d'autres nœuds. Il existe deux principaux types d'architectures: réseaux récurrents (avec cycles) et réseaux à couches distinctes (sans cycles). Des réseaux à architecture hybride existent également: e.g. les réseaux de Elman (Memmi 2004b). Les structures des réseaux (nombre de neurones artificiels et connexions entre ces neurones) sont généralement prédéterminées excepté dans certains types de réseaux à dynamique plus complexe tels que les réseaux de type ART (Memmi 2004b). L'état des connaissances est distribué dans les poids (ajustés par entraînement) et les valeurs d'activation du réseau.

Lors du fonctionnement du réseau, les données d'entrées reçues par les neurones sont pondérées par les poids d'activation. Les données pondérées sont traitées par les fonctions d'activation des neurones concernés qui produisent des valeurs d'activation servant de données d'entrées à d'autres neurones. C'est la propagation des valeurs d'activation qui produit le traitement global désiré (classification ou catégorisation en général).

Une phase d'apprentissage permet d'ajuster les poids d'activation du réseau: les poids initialement assignés avec des valeurs par défaut sont ajustés de façon itérative et adaptative. L'apprentissage supervisé consiste à donner des exemples de paires d'entrée-sortie pour permettre au système de corriger ses réponses initiales afin de se rapprocher de celles désirées. L'apprentissage non supervisé consiste à fournir des exemples de données d'entrée au système et à laisser le système déterminer les corrélations avec les données.

Représentation et traitement de connaissances avec les algorithmes génétiques

Les algorithmes génétiques utilisent des représentations de solutions appelées chromosomes; ce terme vient de la forme généralement linéaire de ces représentations ainsi que de leurs traitements qui sont inspirés sur des opérations de biologie évolutive (Wikipedia, 2006b). On distingue trois types de codage de représentation: (i) codage sur une chaîne binaire (codage peu naturel mais le plus fréquemment employé à cause qu'il offre plus de possibilité au niveau du croisement génétique); (ii) codage sur une chaîne de caractères à valeurs multiples (codage plus naturel que le codage binaire); et codage arborescent (utile lorsque les solutions n'ont pas de taille bien définie).

Le traitement est basé sur trois opérations de biologie évolutive: (i) la *sélection* qui détermine les chromosomes (individus) enclins à donner les meilleures solutions; (ii) le *croisement* où deux chromosomes s'échangent des parties de leurs chaînes pour donner de nouveaux chromosomes (la probabilité de croisement est typiquement entre 0.1 à 0.9); et (iii) la *mutation* qui substitue aléatoirement un gène d'un chromosome par un autre (la probabilité de mutation est typiquement entre 0,001 et 0,01). La mutation sert à éviter une convergence prématurée de l'algorithme vers une solution donnée

Le traitement débute avec une population de base qui se compose le plus souvent de chaînes de caractères correspondant chacune à un chromosome. Le contenu de cette population initiale est généré aléatoirement. On attribue à chacune des solutions une note qui correspond à son adaptation au problème. Ensuite, on effectue une sélection au sein de cette population (plusieurs techniques de sélection sont disponibles). Lorsque deux chromosomes ont été sélectionnés, on réalise

un croisement. On effectue ensuite des mutations sur une faible proportion d'individus choisis aléatoirement. Ce processus produit une nouvelle population. On réitère le processus un grand nombre de fois de manière à imiter le principe d'évolution, qui ne prend son sens que sur un nombre important de générations. On peut arrêter le processus au bout d'un nombre arbitraire de générations ou lorsque que le résultat d'évaluation d'une solution est satisfaisant.

Représentation et traitement de connaissances avec les réseaux bayésiens

Un réseau bayésien est un graphe orienté acyclique représentant la distribution des probabilités jointes de variables (Lounis, 2006c). Ce graphe indique explicitement les dépendances et indépendances conditionnelles; chaque variable est dépendante conditionnellement seulement des variables qui sont descendants dans le graphe. Les probabilités sont basées sur de l'information a priori non retrouvée dans les données.

Le calcul des probabilités repose sur le théorème de Bayes: $P(h|D) = (P(D|h) P(h)) / P(D)$ où D = évidence; h = hypothèse; $P(h)$ = probabilité a priori de h ; $P(h|D)$ = probabilité a posteriori de h étant donné D ; et $P(D|h)$ = probabilité conditionnelle de D étant donné h . La distribution de probabilité jointe d'un ensemble de variables à partir d'un réseau bayésien est donnée par: $P(x_1, x_2, \dots, x_n) = \prod_i P(x_i | Parents(x_i))$ où $Parents(x_i)$ est l'ensemble des prédécesseurs immédiats de x_i . Des algorithmes permettent d'induire la structure et/ou les probabilités conditionnelles des variables.

6. Avantages et limites des approches symboliques et sous-symboliques en représentation de connaissances

Aperçu

MacIntyre (2000) propose une liste des avantages et limites liées aux approches symboliques et sous-symboliques. Toutefois, les caractéristiques proposées s'appliquent seulement de façon très générale à ces approches et ne tient pas compte des cas particuliers. Negnevitsky (2002, Table 8.1, p. 259) propose une description plus raffinée mais pour seulement deux approches symboliques et deux approches sous-symboliques. Le reste de cette section décrit les descriptions proposées par MacIntyre (2000) et une extension que nous proposons à partir de la description de Negnevitsky (2002).

Sommaire des avantages et limites liés aux approches symboliques selon MacIntyre (2000)

Selon MacIntyre (2000) les avantages généralement associés aux approches symboliques de représentation de connaissances incluent: (i) facilité de compréhension à cause de représentations locales avec connaissances explicites; (ii) possibilité d'explication du raisonnement; (iii) grande variété de techniques de raisonnement et de recherche; (iv) facilité pour le développement incrémental par prototype; (v) possibilité de représenter des connaissances complexes avec riche structure interne (cependant les arbres de décisions, auxquels ne réfèrent pas MacIntyre, sont ici une exception car leurs structures font références aux attributs et valeurs de façon superficielle); (vi) possibilité de fonctionner avec peu (ou aucune) données d'entrées (car des connaissances a priori éditées manuellement peuvent souvent être suffisantes – mais les arbres de décisions sont une exception puisqu'ils demandent des données d'entraînement).

Selon MacIntyre (2000) les limites généralement associées aux approches symboliques de représentation de connaissances incluent: (i) connaissances spécifiques au domaine (i.e. ne se transférant pas bien d'un domaine à l'autre – mais les arbres de décisions sont une exception); (ii) approches généralement peu adaptées à certains domaines numériques (e.g. traitement d'images); (iii) performances ne s'améliorant pas avec l'expérience (exception: arbres de décisions et systèmes à base de cas); (iv) généralisation des solutions faible ou inexistante; (v) faible robustesse aux données bruitées (i.e. ne fonctionnant pas avec des erreurs dans les données ou impliquant une dégradation brusque des résultats en présence d'erreurs – mais les arbres de décisions et les systèmes à base de cas sont des exceptions); (vi) nécessité de l'implication d'experts pour la spécification de connaissances (les arbres de décisions font ici exception); et (vii) développement long et coûteux en terme de ressources humaines et non humaines (logiciels nécessaires pour édition ou élicitation des connaissances, etc. - mais les arbres de décisions font ici aussi exception).

Sommaire des avantages et limites liés aux approches sous-symboliques selon MacIntyre (2000)

Selon MacIntyre (2000) les avantages généralement associés aux approches sous-symboliques de représentation de connaissances incluent: (i) adaptabilité à des environnements changeant; (ii) capacité d'apprentissage par l'expérience (la performance peut s'accroître avec l'exposition à de nouvelles données); (iii) capacité de généralisation avec de nouvelles données; (iv) robustesse aux données bruitées (dégradation progressive des résultats avec les erreurs dans les données); (v) peu d'expertise requise pour spécifier l'entrée de données; (vi) développement initial rapide (le raffinement peut être ardu).

Selon MacIntyre (2000) les limites généralement associées aux approches sous-symboliques de représentation de connaissances incluent: (i) représentations distribuées implicites, difficiles à identifier; (ii) traitement opaque des données; (iii) peu de capacité d'explication des résultats (séquence du raisonnement difficile à traiter); (iv) requière souvent beaucoup de données d'entrées pour l'entraînement; (v) application limitée à des représentations de connaissances simples sans structure interne.

Description des avantages et limites basée sur une extension de la description de Negnevitsky (2002)

Negnevitsky (2002, Table 8.1, p. 259) décrit les avantages et limites de deux approches symboliques (systèmes experts et logique floue) et de deux approches sous-symboliques (réseaux de neurones artificiels et algorithmes génétiques). La description est basée sur quelques caractéristiques (décrites plus bas) dont l'évaluation repose sur l'échelle suivante: ++ (bon), + (plutôt bon), - (plutôt mauvais), - - (mauvais). La description proposée par Negnevitsky est intéressante mais elle reste incomplète, ne couvrant que quelques approches et un nombre assez limité de caractéristiques. Nous n'avons toutefois pas trouvé dans la littérature de description plus raffinée des avantages et limites des approches symboliques et sous-symboliques que celle proposée par Negnevitsky. Cependant, on retrouve dans la littérature certaines descriptions complémentaires des avantages et limites liés à certaines approches symboliques et sous-symboliques (Boukadoum, 2006; Memmi, 2004a; Memmi, 2004b ; etc.).

Plus bas nous proposons une description des avantages et limites des approches symboliques et sous-symboliques qui se

veut une tentative de synthèse de la description de Negnevitsky avec d'autres descriptions trouvées dans la littérature. Nous référons au terme *tentative* car la classification de caractéristiques selon une échelle allant de bon à mauvais est quelque chose de subjectif.

Le Tableau 1 décrit les acronymes utilisés plus bas pour référer aux différentes approches symboliques et sous-symboliques. Ce tableau décrit aussi la notation utilisée pour référer aux valeurs relatives d'évaluation des caractéristiques décrites plus bas. Nous ajoutons à la notation proposée par Negnevitsky la notation +/- afin de référer à une valeur d'évaluation variable selon les variantes de l'approche considérée (par exemple, un système expert basé sur la logique floue supporte bien l'imprécision alors qu'un système expert basé sur la logique classique ne la supporte pas).

Types d'approche			Évaluation	
SE: système expert	ON: ontologies	RN: réseau de neurones artificiels	- -: mauvais	+/- : variable selon variantes
LC: logique classique	SC: schémas	AG: algorithme génétique	- : plutôt mauvais	
LF: logique floue	BC: système à base de cas	RB: réseau bayésien	+ : plutôt bon	
AD: arbre de décisions			++ : bon	

Tableau 1. Notations utilisées pour décrire les avantages et limites des approches

Le Tableau 2 décrit la liste des caractéristiques que nous considérons plus bas pour référer aux avantages et limites des approches symboliques et sous-symboliques. Cette liste de caractéristiques est une légère extension de la liste utilisée par Negnevitsky pour décrire les représentations ou leurs traitements: alors que Negnevitsky réfère simplement à *représentation de connaissances*, nous distinguons *compréhensibilité des représentations* et *support de connaissances complexes*; nous ajoutons également les caractéristiques de *tolérance à l'erreur* et de *facilité de développement*.

Caractéristiques des représentations ou de leurs traitements	Description
Compréhensibilité des représentations	Facilité de compréhension des connaissances représentées
Tolérance à l'incertitude	Support pour les données manquantes
Tolérance à l'imprécision	Support pour les données imprécises
Tolérance à l'erreur	Support pour les données erronées
Adaptabilité	Support pour de nouveaux types de données et de nouveaux domaines
Capacité d'apprentissage	Support pour l'apprentissage automatique des connaissances
Capacité de découverte/fouille	Support pour découvrir des connaissances parmi beaucoup de données
Capacité d'explication	Support pour le traçage du raisonnement
Facilité de développement	Simplicité et rapidité du développement
Maintenabilité	Simplicité de la maintenance
Support de connaissances complexes	Support pour représenter des connaissances complexes

Tableau 2. Caractéristiques utilisées pour décrire les avantages et limites des approches

Le Tableau 3 décrit les valeurs relatives d'évaluation que nous proposons pour chacune des approches symboliques et sous-symboliques présentées précédemment par rapport à chaque caractéristique décrite dans le Tableau 2. Des justifications pour ces évaluations relatives des approches individuelles sont présentées plus bas. Les évaluations apparaissant dans les cellules ombragées □ sont celles proposées par Negnevitsky. Les évaluations manquantes sont celles que nous ne sommes pas en mesure de déterminer ou de justifier.

Caractéristiques	Approches symboliques							Approches sous-symboliques		
	SE	LC	LF	AD	ON	SC	BC	RN	AG	RB
Compréhensibilité des représentations	+	++	++	++	++	+	+	--	-	-
Tolérance à l'incertitude	+	++	++	++	+	+	+	++	++	++
Tolérance à l'imprécision	+/-	--	++	++	-	-	+	++	++	+
Tolérance à l'erreur	--	--	--	+	-	--	+	++	++	++
Adaptabilité	--	-	-	+	-	--	+/-	++	++	++
Capacité d'apprentissage	--	--	--	++	+/-	--	+	++	++	++
Capacité de découverte/fouille	--	-	-	+	+	-	--	++	+	
Capacité d'explication	++	++	++	+	+	+	+	--	-	+
Facilité de développement	--	-	-	++	-	--	--	-	-	++
Maintenabilité	--	+	+	++	+/-	-	-	++	+	+
Support de connaissances complexes	+	+	+	-	++	++	+	--	--	--

Tableau 3. Avantages et limites des approches individuelles (symboliques ou sous-symboliques)

Le Tableau 4 décrit une évaluation globale des approches symboliques et sous-symboliques pour chaque caractéristique à partir d'une synthèse des valeurs d'évaluation du Tableau 3.

Caractéristiques	Approches symboliques	Approches sous-symboliques
Compréhensibilité des représentations	Bonne ou plutôt bonne	Plutôt mauvaise ou mauvaise
Tolérance à l'incertitude	Plutôt bonne ou bonne	Bonne
Tolérance à l'imprécision	Variante de bonne (LF, AD) à mauvaise (LC)	Bonne ou plutôt bonne
Tolérance à l'erreur	Variante de mauvaise (SE, LC, LF, SC) à plutôt bonne (AD, BC)	Bonne
Adaptabilité	Variante de mauvaise (SE, SC) à plutôt bonne (AD, BC ?)	Bonne
Capacité d'apprentissage	Variante de mauvaise (SE, LC, LF, SC) à bonne (AD)	Bonne
Capacité de découverte/fouille	Généralement mauvaise ou plutôt mauvaise (exception: AD et ON plutôt bonne)	Plutôt bonne ou bonne (RB non-spécifiée)
Capacité d'explication	Plutôt bonne ou bonne	Variante de mauvaise (RN) à plutôt bonne (RB)
Facilité de développement	Généralement mauvaise ou plutôt mauvaise (exception: AD bonne)	Plutôt mauvaise (RN, AG) ou bonne (RB)
Maintenabilité	Variante de bonne (AD) à mauvaise (SE)	Plutôt bonne ou bonne
Support de connaissances complexes	Généralement bonne ou plutôt bonne (exception: AD plutôt mauvaise)	Mauvaise

Tableau 4. Évaluation globale des approches symboliques et des approches sous-symboliques pour chaque caractéristique

Les tableaux 5 à 14 contiennent des remarques justifiant les valeurs relatives d'évaluation apparaissant dans le Tableau 3.

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	+	les faits et règles déclaratives sont généralement compréhensibles mais pas le moteur d'inférence
Tolérance à l'incertitude	+	évaluation de Negnevitsky
Tolérance à l'imprécision	+ / -	plutôt bonne avec logique floue (ou AD), plutôt mauvaise autrement
Tolérance à l'erreur	--	des connaissances erronées ont un impact direct sur le raisonnement
Adaptabilité	--	dépendance au domaine
Capacité d'apprentissage	--	les connaissances sont créées manuellement
Capacité de découverte/fouille	--	évaluation de Negnevitsky
Capacité d'explication	++	raisonnement basé sur de bonnes fondations théoriques
Facilité de développement	--	requière l'implication intensive d'experts
Maintenabilité	--	grande interdépendance entre les règles
Support de connaissances complexes	+	limité à connaissances formalisables par des faits et règles

Tableau 5. Remarques sur l'évaluation des systèmes experts

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	++	les faits et règles déclaratives sont généralement compréhensibles
Tolérance à l'incertitude	++	l'inférence logique permet de raisonner avec des informations manquantes
Tolérance à l'imprécision	--	support de valeurs booléennes seulement
Tolérance à l'erreur	--	des connaissances erronées ont un impact direct sur les conclusions des inférences
Adaptabilité	-	dépendance au domaine
Capacité d'apprentissage	--	les connaissances sont créées manuellement
Capacité de découverte/fouille	-	même valeur que celle assignée par Negnevitsky pour la logique floue
Capacité d'explication	++	raisonnement basé sur de bonnes fondations théoriques
Facilité de développement	-	requière l'implication intensive d'experts
Maintenabilité	-	les règles sont assez indépendantes
Support de connaissances complexes	+	limité à connaissances formalisables par des faits et règles

Tableau 6. Remarques sur l'évaluation de la logique classique

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	++	les faits et règles déclaratives sont généralement compréhensibles
Tolérance à l'incertitude	++	l'inférence logique permet de raisonner avec des informations manquantes
Tolérance à l'imprécision	++	support des valeurs floues
Tolérance à l'erreur	--	des connaissances erronées ont un impact direct sur les conclusions des inférences
Adaptabilité	-	dépendance au domaine
Capacité d'apprentissage	--	les connaissances sont créées manuellement
Capacité de découverte/fouille	-	évaluation de Negnevitsky
Capacité d'explication	++	raisonnement basé sur de bonnes fondations théoriques
Facilité de développement	-	requière l'implication intensive d'experts
Maintenabilité	-	les règles sont assez indépendantes
Support de connaissances complexes	+	limité à connaissances formalisables par des faits et règles
Autres		(i) l'optimisation des règles peut être difficile, (ii) la performance peut être pauvre avec règles complexes, et (iii) la validation des règles peut être difficile

Tableau 7. Remarques sur l'évaluation de la logique floue

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	++	Les arbres et règles correspondantes sont généralement compréhensibles excepté pour des représentations très complexes
Tolérance à l'incertitude	++	support des valeurs manquantes (cas pour certaines techniques)
Tolérance à l'imprécision	++	les valeurs numériques réelles sont supportées
Tolérance à l'erreur	+	plusieurs techniques supportent des seuils d'erreurs mais relativement faibles
Adaptabilité	+	s'adapte bien aux domaines dont les données sont représentables par quelques attributs dont les valeurs sont non-structurées
Capacité d'apprentissage	++	plusieurs techniques d'induction d'arbres sont disponibles
Capacité de découverte/fouille	+	peuvent découvrir les attributs pertinents d'une solution mais les techniques ne supportent généralement qu'un nombre limité d'attributs
Capacité d'explication	+	l'arbre fournit une trace directe d'une décision mais la puissance d'explication est limité explicitement au contenu de l'arbre
Facilité de développement	++	les données d'entrées ne sont que des ensembles d'attributs/valeurs
Maintenabilité	++	les arbres sont généralement induits automatiquement et n'ont pas à être maintenus manuellement
Support de connaissances complexes	-	limité à des ensembles d'attributs simples
Autres		(i) représentations facilitant le calcul

Tableau 8. Remarques sur l'évaluation des arbres de décisions

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	++	les spécifications sont déclaratives et généralement compréhensibles
Tolérance à l'incertitude	+	les spécifications peuvent être incomplètes
Tolérance à l'imprécision	-	les spécifications de concepts et de relations peuvent être partielles ou faire références à des contraintes assez vagues sur leurs propriétés. Toutefois, les connaissances doivent être définies explicitement
Tolérance à l'erreur	-	les erreurs ont des impacts directement dans la représentation. Des techniques de validation peuvent détecter certains types d'erreurs
Adaptabilité	-	même les ontologies de niveau plus abstrait ont des restrictions sur les domaines de modélisation
Capacité d'apprentissage	+ / -	les connaissances ontologiques profondes ne sont jamais acquises automatiquement mais celles de surfaces le sont parfois
Capacité de découverte/fouille	+	il existe plusieurs techniques permettant l'extraction de connaissances à partir de grand corpus textuels mais les types de connaissances extraites sont assez limités
Capacité d'explication	+	les liens relationnels permettent les explications
Facilité de développement	-	requière l'implication très intensive d'experts mais permet le développement partagé
Maintenabilité	+ / -	les ontologies de haut niveau se maintiennent facilement (en principe) mais pas celles de bas niveau qui doivent souvent être mises à jour
Support de connaissances complexes	++	supporte des attributs et relations divers

Tableau 9. Remarques sur l'évaluation des ontologies

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	+	les spécifications sont déclaratives et compréhensibles mais elles sont très arbitraires
Tolérance à l'incertitude	+	les spécifications peuvent être incomplètes
Tolérance à l'imprécision	-	les spécifications supportent la représentation de valeurs imprécises
Tolérance à l'erreur	--	les erreurs ont des impacts directement dans la représentation
Adaptabilité	--	forte dépendance au domaine
Capacité d'apprentissage	--	les connaissances ne sont généralement pas acquises automatiquement
Capacité de découverte/fouille	-	les schémas peuvent être instanciés automatiquement à partir de corpus structurés et non structurés mais les techniques sont ad-hoc
Capacité d'explication	+	les liens relationnels permettent les explications
Facilité de développement	--	requière l'implication très intensive d'experts
Maintenabilité	-	la maintenance est compliquée par la dépendance au domaine mais est facilitée par la modularité des schémas
Support de connaissances complexes	++	attributs et relations divers

Tableau 10. Remarques sur l'évaluation des schémas

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	+	les spécifications sont déclaratives et compréhensibles mais sont arbitraires
Tolérance à l'incertitude	+	les spécifications peuvent être incomplètes
Tolérance à l'imprécision	+	les attributs peuvent avoir des valeurs réelles (floues)
Tolérance à l'erreur	+	la métrique de similarité offre une certaine robustesse
Adaptabilité	+ / -	forte dépendance au domaine mais la métrique de similarité peut offrir une certaine flexibilité
Capacité d'apprentissage	+	la similarité de cas permet de réutiliser des solutions
Capacité de découverte/fouille	--	les cas pourraient possiblement être instanciés à partir de corpus structurés et non structurés mais si de telles techniques existent (?) elles sont probablement ad-hoc
Capacité d'explication	+	explication limitée à la similarité des cas
Facilité de développement	--	expertise nécessaire pour le choix des attributs et la détermination des cas de base (parfois nombreux)
Maintenabilité	-	la maintenance est compliquée par la dépendance au domaine mais est facilitée par la modularité des cas
Support de connaissances complexes	+	chaque cas peut être défini par un ensemble d'attributs divers
Autres		(i) spécification arbitraire des poids des attributs; (ii) une bonne performance nécessite souvent l'élimination de cas spécifiques

Tableau 11. Remarques sur l'évaluation des systèmes à base de cas

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	- -	connaissances distribuées dans les poids et valeurs d'activation
Tolérance à l'incertitude	+ +	évaluation de Negnevitsky
Tolérance à l'imprécision	+ +	évaluation de Negnevitsky
Tolérance à l'erreur	+ +	l'importance des erreurs est amoindrie par le biais des statistiques
Adaptabilité	+ +	évaluation de Negnevitsky
Capacité d'apprentissage	+ +	poids ajustés par apprentissage (données nécessaires)
Capacité de découverte/fouille	+ +	évaluation de Negnevitsky
Capacité d'explication	- -	évaluation de Negnevitsky
Facilité de développement	-	les données d'entrées sont faciles à préparer mais définir et ajuster la structure du réseau pour le problème peut nécessiter un travail non négligeable
Maintenabilité	+ +	évaluation de Negnevitsky
Support de connaissances complexes	- -	données structurées ou séquences difficiles à représenter/traiter
Autres		(i) approche assez simple et assez générale; (ii) intéressant pour la simulation en neurobiologie; (iii) nécessite de grandes quantités de données d'entraînement

Tableau 12. Remarques sur l'évaluation des réseaux de neurones artificiels

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	-	connaissances distribuées dans les chromosomes
Tolérance à l'incertitude	+ +	évaluation de Negnevitsky
Tolérance à l'imprécision	+ +	évaluation de Negnevitsky
Tolérance à l'erreur	+ +	l'importance des erreurs est amoindrie par le biais des statistiques et des transformations aléatoires
Adaptabilité	+ +	évaluation de Negnevitsky
Capacité d'apprentissage	+ +	évaluation de Negnevitsky
Capacité de découverte/fouille	+	évaluation de Negnevitsky
Capacité d'explication	-	nombreuses transformations aléatoires
Facilité de développement	-	les paramètres comme la taille de la population ou le taux de mutation sont parfois difficiles à déterminer
Maintenabilité	+	évaluation de Negnevitsky
Support de connaissances complexes	- -	données structurées ou séquences difficiles à représenter / traiter
Autres		(i) intéressant pour la simulation en biologie évolutive; (ii) idéal pour trouver des solutions optimales pour des problèmes à plusieurs variables (traitement parallèle); mais une solution optimale est obtenue par chance et est non assurée; (iii) temps calcul élevé au niveau fonction d'évaluation: convergence lente (évalue solutions non optimales); (iv) nécessite de grandes quantités de données d'entraînement

Tableau 13. Remarques sur l'évaluation des algorithmes génétiques

Caractéristiques	Éval.	Remarques
Compréhensibilité des représentations	-	le graphe représente des dépendances explicites entre variables mais les distributions de probabilités sont difficiles à interpréter
Tolérance à l'incertitude	++	les données manquantes sont supportées
Tolérance à l'imprécision	+	les valeurs réelles des attributs doivent être préalablement discrétisées
Tolérance à l'erreur	++	l'importance des erreurs est amoindrie par le biais des statistiques
Adaptabilité	++	s'adapte bien aux problèmes reliés aux distributions de probabilités
Capacité d'apprentissage	++	la structure du graphe et les distributions de probabilités des variables peuvent être apprises à partir de données
Capacité de découverte/fouille		<i>indéterminée</i> : nous ignorons l'applicabilité des réseaux bayésiens pour la fouille
Capacité d'explication	+	le graphe représente explicitement les dépendances entre variables
Facilité de développement	++	principale motivation pour utiliser cette approche
Maintenabilité	+	une fois la structure et les relations du graphe définies (manuellement ou automatiquement), les statistiques du graphe sont acquises automatiquement et n'ont pas à être maintenues manuellement.
Support de connaissances complexes	--	méthode applicable qu'à des attributs discrets (ou discrétisés)
Autres		(i) simplicité de la méthode; (ii) résultats facilement vérifiables; (iii) l'utilisation d'information a priori est sensé, mais l'information a priori est arbitraire; (iv) l'information a priori peut être erronée et conduire à de fausses conclusions; (v) résultats subjectifs; (vi) nécessite de grandes quantités de données pour déterminer les distributions de probabilités

Tableau 14. Remarques sur l'évaluation des réseaux bayésiens

7. Approches hybrides symboliques/sous-symboliques en représentation de connaissances

Aperçu

Une approche hybride symbolique/sous-symbolique de représentation de connaissances combine au moins une approche symbolique de représentation de connaissances et une approche sous-symbolique de représentation de connaissances. Chacune des principales approches en représentation de connaissances a des avantages et des inconvénients. Le but d'une approche hybride est d'intégrer les approches de façon à combiner certains avantages de chacune des approches concernées (Negnevitsky, 2002, p.257).

Des exemples d'approches hybrides inclus: (i) systèmes experts neuronaux (connexionnistes) intégrant systèmes experts et réseaux de neurones artificiels (Negnevitsky, 2000, section 8.2); (ii) réseaux de neurones flous (réseaux connexionnistes flous) intégrant réseaux de neurones artificiels et logique floue (Negnevitsky, 2000, section 8.3); systèmes génétiques (évolutifs) flous intégrant algorithmes génétiques et logique floue (Negnevitsky, 2000, section 8.6); etc. Le reste de cette section décrit les systèmes experts neuronaux plus en détails.

Les systèmes experts neuronaux

Les caractéristiques des systèmes experts et celles des réseaux de neurones artificiels sont complémentaires à plusieurs niveaux (Negnevitsky, 2000, section 8.2). D'un côté, un système expert (i) permet de modéliser l'intelligence humaine par le biais du raisonnement, (ii) utilise des représentations locales de connaissances explicites (e.g. faits et règles symboliques),

(iii) permet l'explication du raisonnement, et (iv) utilise des connaissances initiales créées manuellement et qui ne sont pas modifiables par expérience. De l'autre côté, un réseau de neurones artificiels (i) permet de modéliser l'intelligence humaine par le biais du traitement parallèle et de l'apprentissage, (ii) utilise des représentations distribuées de connaissances implicites (poids, connexions et fonctions d'activation du réseau), (iii) ne permet pas l'explication du raisonnement et (iv) utilise des connaissances initiales modifiables par expérience.

Les caractéristiques générales d'un système expert neuronal incluent: (i) possibilité de modéliser l'intelligence humaine partiellement par le biais du raisonnement et partiellement par le biais du traitement parallèle et de l'apprentissage; (ii) utilisation des représentations partiellement distribuées (réseaux de neurones) et partiellement locales (règles symboliques); (iii) possibilité d'explication du raisonnement à partir des règles symboliques; et (iv) utilisation de connaissances initiales liées aux réseaux de neurones modifiables par expérience.

Un cas particulier de système expert neuronal

Negnevitsky (2000, section 8.2) présente un exemple de système expert neuronal. L'architecture de ce système est celui d'un système expert avec en plus (i) une composante de règles sous forme de réseau de neurones artificiels et (ii) une composante d'extraction de règles symboliques à partir du réseau de neurones artificiels. Ce système permet la modification des règles (poids des connexions) du réseau de neurones artificiels par apprentissage à partir d'un ensemble de données d'entraînement. Ce système permet également la création automatique de règles symboliques par l'extraction des connaissances correspondantes contenues dans le réseau de neurones artificiels. Le domaine d'application de ce système est celui de la décision sur la classification d'un objet (*oiseau/bird*, *avion/plane*, ou *planeur/glider*) à partir d'un ensemble de traits (*ailles/wings*, *queue/tail*, *bec/beak*, *plumes/feathers*, et/ou *moteur/engine*).

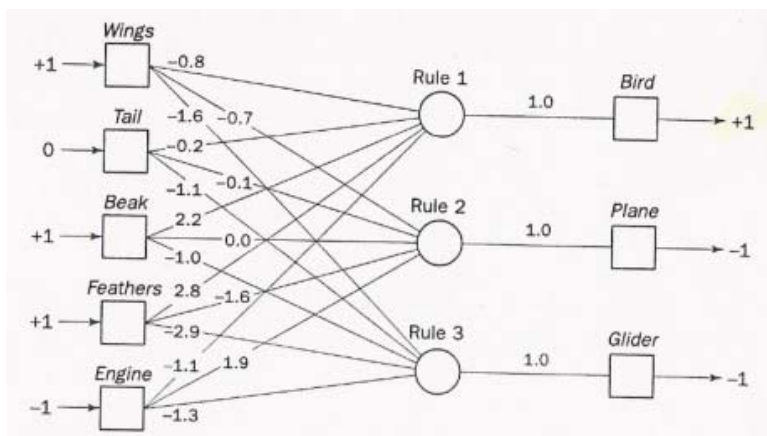


Figure 1. Exemple de représentation de règles dans un réseau de neurones artificiels: source (Negnevitsky 2000, p. 262)

La Figure 1 illustre la représentation de règles dans le réseau de neurones artificiels présenté par Negnevitsky. La structure du réseau correspond à un ensemble de règles, chacune associée à une classe (e.g. *oiseau/bird*). Le réseau se compose de trois couches: une première couche où les neurones représentent des traits caractéristiques utilisés pour les règles; une deuxième couche où les neurones représentent l'application des règles via les fonctions d'activation; et une troisième couche où les neurones représentent les conclusions des règles (classes associées aux données d'entrées). Les données

d'entrées consistent en un vecteur numérique dont chaque valeur représente l'absence (-1), l'incertitude (0) ou la présence (1) d'un trait correspondant. La structure du réseau est prédéfinie mais les poids sont ajustables par entraînement.

Les étapes d'extraction de règles symboliques à partir du réseau de neurones artificiels sont les suivantes (Negnevitsky, 2002, pp. 264--265):

1. À partir de données d'entrées et des valeurs de sorties du réseau de neurones, choisir une classification/conclusion (e.g. *Bird is true*, voir Figure 1) pour laquelle on veut produire une règle d'inférence symbolique.
1. Identifier le neurone associé à la règle pour cette classification dans le réseau (e.g. neurone *Rule 1* de la Figure 1) et déterminer les valeurs de contributions des connexions associées à ce neurone; une contribution est une valeur de même signe que la valeur de classification (e.g. contributions positives pour *Bird is true*). La valeur d'une contribution est déterminée par la valeur absolue du poids de la connexion.
2. Ordonner les valeurs de contribution des entrées par ordre décroissant. Par exemple, pour le neurone correspondant à la règle pour la classification *Bird is true* on a:

2.8 (contribution pour le trait *Feathers*)

2.2 (contribution pour le trait *Beak*)

1.1 (contribution pour le trait *Engine*)

0.2 (contribution pour le trait *Tail*)

3. Créer une règle symbolique dont la conclusion est formée des traits d'entrée contribuant à la règle correspondante dans le réseau de neurones et dont la conclusion est formée à partir de la valeur de classification précédemment choisie: e.g.

IF Feathers is true THEN Bird is true

4. Vérifier la règle courante en vérifiant l'heuristique suivante: *une inférence peut-être faite si la somme des poids connus vers le neurone correspondant à la règle est supérieure à la somme des autres poids*. La formalisation de cette heuristique est la suivante:

$$\sum_{i=1}^n x_i w_i > \sum_{j=1}^n |w_j| \quad \text{où } i \in \text{KNOWN} \text{ et } j \in \text{UNKNOWN}$$

Par exemple, en considérant la règle précédente et le réseau précédent à partir de la Figure 1 on a:

$$\text{KNOWN} = 1 \times 2.8 = 2.8$$

$$\text{UNKNOWN} = |-0.8| + |-0.2| + |2.2| + |-1.1| = 4.3$$

$$\text{KNOWN} < \text{UNKNOWN}$$

Donc, la règle ne peut pas être inférée.

5. Raffiner et valider itérativement la règle en considérant les prochains traits d'entrées selon leurs valeurs de contribution: e.g. raffinement de la règle précédente selon le réseau de la Figure 1

IF Feathers is true AND Beak is true THEN Bird is true

L'évaluation de cette règle donne:

$$\begin{aligned} \text{KNOWN} &= 1 \times 2.8 + 1 \times 2.2 = 5.0 \\ \text{UNKNOWN} &= |-0.8| + |-0.2| + |-1.1| = 2.1 \\ \text{KNOWN} &> \text{UNKNOWN} \end{aligned}$$

Donc, la règle peut être inférée.

Discussion sur les systèmes experts neuronaux

Les systèmes experts neuronaux peuvent faire des déductions avec une information incomplète. Le réseau de neurones permet de représenter des règles et d'acquérir des connaissances (poids des connexions) à propos de ces règles. Les connaissances du réseau de neurones peuvent être converties en règles symboliques afin de permettre l'explication d'un raisonnement fait à partir de ces règles. Les systèmes experts neuronaux basés sur la logique classique sont cependant mal adaptés pour les variables dont les valeurs sont continues; un nombre infini de règles symboliques pourrait potentiellement résulter dans ces cas (Negnevitsky, 2000, p. 266). L'utilisation de la logique floue à la place de la logique classique peut permettre traiter les cas où les valeurs sont continues.

8. Remerciements

Ce travail a été financé par une bourse doctorale du Conseil de Recherches en Sciences Humaines du Canada (CRSH).

9. Références

- Ali, Syed S. (1999). *Links*. Online document on the topic of knowledge, University of Wisconsin, Milwaukee. <http://tiger.cs.uwm.edu/~syali/links/9906/column.html>
- Bouchard, Lorne; & Abdel Obaid (2005). *Construction d'ontologie*. Notes de cours (Séminaire sur le Web Sémantique), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.
- Boukadoum, Mounir (2006). *Évaluation de variables hydro-électriques: une comparaison de différentes approches d'apprentissage automatique ou rien ne sert de courir, encore faut-il partir à point revisité*. Notes de séminaire en informatique cognitive, Université du Québec à Montréal.
- Gómez-Pérez, Asunción; Mariano Fernández-Lopez; & Oscar Corcho (2004, 2^{ème} édition). Theoretical Foundations of Ontologies. Chapter 1 of *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer-Verlag, pp. 1–45.
- Gómez-Pérez, Asunción (Editor; 2002). *Deliverable 1.3: A survey on ontology tools*. OntoWeb Consortium. http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_v1-0.zip
- Information Society Technology (2001). *Glossary*. Online document about terminology in information technology. <http://www.cordis.lu/ist/ka1/administrations/publications/glossary.htm>
- Lavoie, Benoit (2006). *Arbres de décisions*. Synthèse de lectures (Séminaire spécial sur l'apprentissage automatique), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal. 12p. http://www.benoit-lavoie.ca/public/docs/Arbres_de_decisions.pdf
- Lounis, Hakim (2006a). *Arbres de décision*. Notes de cours (Séminaire sur l'apprentissage automatique), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.
- Lounis, Hakim (2006b). *Apprentissage à partir de cas*. Notes de cours (Séminaire sur l'apprentissage automatique), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.
- Lounis, Hakim (2006c). *Apprentissage Bayésien*. Notes de cours (Séminaire sur l'apprentissage automatique), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.

- MacIntyre, John (2000). *Other AI Paradigms and Hybrid Systems*, Lectures notes (Knowledge Engineering), School of Computing and Technology, University of Sunderland. <http://osiris.sunderland.ac.uk/~cs0jma/kehybrid.ppt>
- Memmi, Daniel (2004a). *Méthodes symboliques*. Notes de cours (Introduction à l'informatique cognitive), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.
- Memmi, Daniel (2004b). *Réseaux de neurones artificiels*. Notes de cours (Introduction à l'informatique cognitive), Programme de Doctorat en Informatique Cognitive, Université du Québec à Montréal.
- Negnevitsky, Michael (2002). Hybrid intelligent systems. Chapter 8 of *Artificial Intelligence: A Guide to Intelligent Systems*, Addison Wesley.
- Wikipedia (2006a). *Knowledge representation*. Online article from Wikipedia encyclopedia. http://en.wikipedia.org/wiki/Knowledge_representation
- Wikipedia (2006b). *Algorithme génétique*. Online article from Wikipedia encyclopedia. http://fr.wikipedia.org/wiki/Algorithme_génétique