♦ Thème 3.5. L'intelligence artificielle

I. — Traitement automatique de l'information

1) Les origines

Les premières machines à effectuer automatiquement une tâche donnée sont d'une part des « machines à calculer », ancêtre des calculatrices et, d'autre part, des métiers à tisser.

1642-1645	Invention de la Pascaline, première « machine à calculer » par Pascal.
1671-1694	Leibniz construit une machine pouvant réaliser des multiplications.
1725	Bouchon invente un métier à tisser programmable à l'aide d'un ruban.
1728	Falcon remplace le ruban par des cartes perforées.
1745-1755	Vaucanson remplace les cartes perforées par un cylindre métallique.
1801	Jacquard crée son métier à tisser reprenant les machines précédentes

Au début du XIXe siècle, le mathématicien anglais Charles Babbage entreprend la construction d'une machine à calculer appelée *machine aux différences* ou *machine analytique* capable d'effectuer des calculs exacts sans erreurs et de les imprimer. Pour cela, il s'inspire des machines construites par Pascal et Leibniz et a l'idée, en 1834, d'y ajouter un système de cartes perforées comme celui utilisé par Jacquard pour ses métiers à tisser. En 1843, Ada Lovelace publie la traduction d'un article sur la machine de Babbage augmentée de sept notes personnelles. La note G, notamment, contient un algorithme permettant de calculer des nombres particuliers (les nombres de Bernoulli). Cette note contient deux nouveautés majeures : tout d'abord, elle introduit une boucle conditionnelle et, ensuite, son algorithme est écrit, non pas de façon abstraite et générale, mais dans une forme destinée à être appliquée directement à la machine analytique de Babbage. En ce sens, il est considéré comme le premier programme informatique de l'histoire.

À la fin du XIXe siècle, Herman Hollerith, employé au bureau du recensement américain, met au point une machine permettant de lire et de trier des cartes perforées. Ce système sera employé pour le recensement de 1890 qui ne prendra « que » 6 ans, contre 10 ans auparavant. C'est le début de la mécanographie, ancêtre de l'informatique. Par la suite, Hollerith quittera l'administration américaine pour fonder sa propre société qui deviendra, en 1917, IBM.

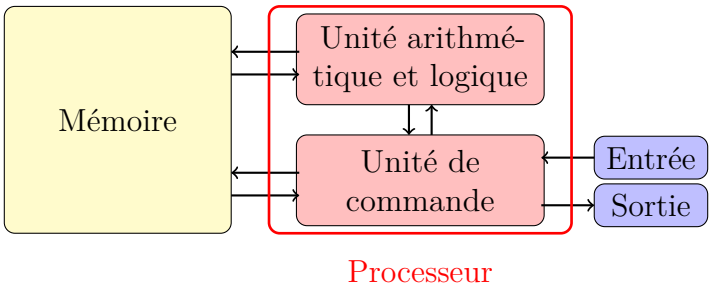
2) Invention de l'ordinateur

Jusqu'au début du XXe siècle, les machines pouvaient exécuter des calculs simples ou des tâches déterminées (comme les métiers à tisser) mais guère plus. En 1936, le mathématicien anglais Alan Turing publie un article dans lequel il définit une machine théorique, appelée depuis *machine de Turing*, qui est plus un concept qu'une machine réelle et qui permet de donner une définition de ce qui est « effectivement calculable » ou plus généralement ce qu'est un algorithme. L'article de Turing comporte deux idées majeures : d'abord, la nécessité de mémoriser des informations et donc de disposer d'une mémoire et, ensuite, le fait d'avoir une machine (théorique) qui peut effectuer n'importe quelle action. En effet, Turing montre l'existence d'*une machine universelle* c'est-à-dire d'une machine de Turing capable de simuler le comportement de n'importe quelle machine de Turing.

Cette vision, bien que purement théorique, est à l'origine du concept d'ordinateur.

En 1945, le mathématicien et physicien américano-Hongrois John von Neumann pose les bases de l'architecture des ordinateurs. Selon ses principes, un ordinateur est « une machine universelle contrôlée par un programme dont les instructions sont codées sous forme numérique (binaire) et enregistrés en mémoire » [(1)]. L'architecture de von Neumann se caractérise par 4 types d'éléments principaux :

- l'unité arithmétique et logique (UAL) chargée des opérations arithmétiques élémentaires ;
- l'unité de commande chargée du « séquençage » des opérations,
- la mémoire qui contient à la fois les données et le programme chargé d'indiquer à l'unité de commande les opérations à faire sur ces données,
- les unités d'entrée et de sortie qui permettent à la machine de communiquer avec le monde extérieur.



Le modèle de von Neumann

Les premières machines opérationnelles construites selon l'architecture de von Neumann voient le jour au Royaume-Uni avec l'EDSAC (Electronic Delay Storage Automatic Calculator) achevé en mai 1949 à l'université de Cambridge et le Manchester Mark 1 mis au point à la Victoria University de Manchester.

3) Développement des ordinateurs

Les deux premiers ordinateurs à usage commercial furent le Ferranti Mark 1, basé sur le Manchester Mark 1, fabriqué par Ferranti à partir de février 1951 et l'UNIVAC (UNIVersal Automatic Computer) basé sur les travaux d'Eckert et Mauchly à la suite de l'ENIAC et fabriqué par Remington Rand à partir de mars 1951. Ce dernier fut le premier ordinateur à avoir une « large » distribution (1500 exemplaires vendus). Il contenait 5 200 tubes et pesait 13 tonnes.

Par la suite, diverses innovations technologiques vont permettre de rendre les ordinateurs de plus en plus fiables, légers, bon marché et simples d'utilisation pour aboutir à une utilisation généralisée et quotidienne.

1949	Invention du transistor qui remplacera les tubes à vide.
années 1950	Ordinateurs de seconde génération.
1958	Invention du circuit intégré.
années 1960	Ordinateurs de troisième génération.
1969	Invention du microprocesseur.
années 1970	Ordinateurs de quatrième génération.
1975	2nde loi de Moore : la capacité des microprocesseurs double tous les 2 ans.
années 1980	Développement des systèmes d'exploitation et des logiciels.
années 1990	Développement des réseaux de communication.
années 2000	Invention des smartphones.

(1). P. Zanella, Y. Ligier et E. Lazard, *Architecture et technologie des ordinateurs - 5ème édition : Cours et exercices corrigés*, Dunod, 2013, p. 16.

II. — Stockage des données

1) Unité de mémoire

Dans le modèle de von Neumann, qui correspond aujourd'hui encore à la structure globale des ordinateurs, les programmes et les données sont stockés dans un espace mémoire.

Un tel espace est composé de milliards de dispositifs électroniques pouvant être dans 2 états possibles qu'on peut interpréter comme 0 ou 1. Cette unité élémentaire de mémoire à deux états est appelée un *bit* (contraction de *binary digit*). Un agglomérat de 8, 16, 32 ou plus de bits constituent ce qu'on appelle des *cases mémoires*. Le nombre de cases mémoires définit la taille de la mémoire de l'ordinateur.

Une autre unité utilisée est l'octet. Un octet est composé de 8 bits. Comme chaque bit peut prendre deux valeurs (0 ou 1), un octet peut coder $2^8 = 256$ valeurs différentes. Généralement, la taille d'une mémoire est donnée en octet ou dans une puissance de dix d'octets :

unité	kilooctet (ko)	mégaoctet (Mo)	gigaoctet (Go)	teraoctet (To)	pétaoctet (Po)
en octets	10^3	10^6	10^9	10^{12}	10^{15}

2) Support de stockage

L'information a été stockée au cours du temps sur différents types de supports internes ou externes.

La première génération de supports était constituée par les ruban perforés et les cartes perforées. La deuxième génération est constituée par les supports magnétiques. D'abord utilisés sous forme de bandes magnétiques à partir de 1928, ils se sont ensuite miniaturisés et démocratisés sous forme de disques durs à partir des années 1950 puis de cassettes et de disquettes dans les années 1960. La troisième génération est constituée par les supports optiques. Parmi ceux-ci, on trouve le disque compact (CD) développé au début des années 1980, le DVD apparu au milieu des année 1990 et le Blu-Ray commercialisé à partir du milieu des années 2000. La quatrième génération, enfin, est constituée par les mémoires flash. Cette technologie développée à la fin des années 1980 est au début coûteuse et peu efficace pour le stockage. Elle va cependant être améliorée pour devenir le mode standard de nos jours avec les clé USB, les cartes SD et micro SD et les disques SSD.

- 1928 — Bande magnétique : 50 octets par cm.
- 1967 — Disquette 8' : environ 82 ko.
- 1981 — Disquette 5'¼ : environ 369 ko.
- 1984 — Disquette 5'¼ (2nde génération) : environ 1,2 Mo.
- 1987 — Disquette 3'½ : environ 1,47 Mo
- 1990 — CD-ROM : environ 682 Mo
- 1995 — DVD : entre 4,7 et 17 Go.
- 2000 — Mémoire flash : 16 Go.
- 2007 — SSD : 1 To.
- 2013 — SSD : 6 To.
- 2015 — SSD : 10 To.
- 2019 — SSD : 16 To.

bande magnétique

disquettes 8', 5'¼ et 3'½

cassette

CDRom

Clé USB

Disque dur SSD

3) Format des données stockées

Un ordinateur peut traiter des données de natures très différentes (texte, image, son, vidéo, logiciel, programme...) une fois que celles-ci ont été numérisées c'est-à-dire transformées en une série de bits. Il existe différentes façons de coder des données qui constituent ce qu'on appelle le format d'un fichier numérique. Le format se reconnaît à l'extension présente dans le nom du fichier c'est-à-dire la chaîne de caractère préfixée par un point qu'on trouve à la fin de ce nom (par exemple, un_texte.txt, un_document_word.docx, un_fichier_son.mp3, un_fichier_video.avi). On peut distinguer deux types de fichiers. D'une part, les fichiers exécutables (comme les logiciels) qui sont écrits dans un langage compréhensible par la machine ou sous une forme interprétable directement par l'ordinateur (c'est-à-dire traduit en langage machine à la volée au fur et à mesure de l'exécution). D'autre part, les fichiers non exécutables qui ne peuvent pas être lus directement par la machine et qui demandent en général de disposer d'un logiciel spécifique pour être lus ou modifiés. Pour une même nature de donnée, il existe souvent plusieurs formats qui nécessitent parfois des logiciels différents pour être ouverts. Le tableau ci-dessous regroupe quelques extensions courantes (mais il y en beaucoup d'autres).

extension	.txt	.docx	.odt	.jpeg	.png	.mp3	.wav	.avi	.mp4
nature	texte	texte	texte	image	image	son	son	vidéo	vidéo

La taille d'un fichier (c'est-à-dire l'espace mémoire occupé par le fichier) est très variable. Les différents formats utilisent souvent des procédés de numérisation qui permettent d'économiser de la place. Par exemple, le format mp3 n'encodent pas les fréquences que l'oreille humaine ne peut pas entendre ou certains formats n'encodent qu'une seule fois une élément redondant (un même mot utilisé plusieurs fois dans un texte, une même image présente plusieurs fois dans une vidéo...). À titre indicatif, on peut retenir les ordres de grandeurs suivants : un fichier texte est de l'ordre du ko, un fichier son ou un fichier image est de l'ordre du Mo et une vidéo est de l'ordre de quelques centaines de Mo au Go.

4) L'exemple du code ASCII

Le format .txt utilise un code, appelé code ASCII, dans lequel chaque « caractère » (lettres, espaces, ponctuation, retours à la ligne, etc) correspond à un nombre entre entre 0 et $2^7 - 1 = 127$ dans sa première version et entre 0 et $2^8 - 1 = 255$ dans sa version étendue qui contient les caractères accentués. Par exemple, les lettres majuscules correspondent aux nombres de 65 à 90 et les lettres minuscules aux nombres de 97 à 122. Ces nombres sont ensuite codés en binaire sur un octet (ce qui est possible puisqu'on utilise 2^8 nombres). Par exemple, la lettre `f` correspond au nombre 102 et l'écriture binaire de 102 est 01100110 donc, dans le format .txt, la lettre `f` est codé par l'octet 01100110.

Ce qu'il faut retenir, c'est qu'un caractère est codé par un octet donc pour connaître la taille d'un fichier .txt, il suffit de compter le nombre de caractères. Par exemple, un fichier .txt contenant le texte :

```
Quelle est la taille de ce texte ?
```

a une taille de 34 octets car il contient 34 caractères (espaces et ponctuation compris).

III. — L'intelligence artificielle

Si l'idée de construire des machines imitant le comportement humain est ancienne (comme les automates de Vaucanson au milieu du XVIIIe), on peut situer la naissance de l'intelligence artificielle dans les année 1950 avec l'article fondateur d'Alan Turing intitulé *Computing Machinery and Intelligence* et la conférence organisée au Dartmouth College (Hanover, New Hampshire) en 1956 sur le thème des machines pensantes durant laquelle le chercheur américain John McCarthy introduit pour la première fois le terme *intelligence artificelle*.

1) Définition

Il n'est pas simple de clairement définir ce qu'on appelle une machine « intelligente ». Est-ce une machine capable d'apprendre ? d'utiliser des connaissances ? d'interagir avec son environnement ? de planifier une succession de tâches ?

Le notion d'**intelligence artificielle** (abrégé IA) possède des contours encore assez flous. On peut cependant adopter la définition suivante, due au chercheur français Yann Le Cun :

On pourrait dire que l'IA est un ensemble de techniques permettant à des machines d'accomplir des tâches et de résoudre des problèmes normalement réservés à des humains et à certains animaux.

Ainsi, l'IA a pour but des rendre des machines capables de reproduire des activités humaines, que ces activités soient de l'ordre de la compréhension, de la perception ou de la décision. On peut citer par exemple :

- la reconnaissance d'objets, d'animaux ou de personnes sur une photo ;
- l'apprentissage de jeux (comme les échecs) ;
- le pilotage de voiture ;
- la traduction de texte ;
- l'établissement de diagnostique médical.

Pour atteindre ce but, une des techniques utilisées (mais à laquelle ne se réduit pas pour autant l'IA) est l'apprentissage machine.

2) Apprentissage machine

L'**apprentissage machine** (appelé aussi apprentissage artificiel ou apprentissage automatique) est un domaine de recherche commun à l'IA et aux statistiques. Il consiste à élaborer des programmes dont le comportant peut évoluer en fonction de données dites *données d'entraînement*.

Le principe général est le suivant : on fournit à la machine un très grand nombre de données à partir desquelles la machine s'entraîne (phase d'apprentissage ou d'entraînement) afin de déterminer le comportement qu'elle adoptera ultérieurement sur de nouvelles données (phase d'inférence).

Imaginons, par exemple, qu"on souhaite apprendre à une machine à reconnaître des images de chats. Plutôt que d'essayer d'écrire des algorithmes complexes permettant d'identifier différentes caractéristiques de l'animal, on fournit à la machine un grand nombre d'images dont certaines sont des images de chat et d'autres non. La machine va alors déterminer des paramètres qui vont permettre de distinguer les photos contenant des chats des autres photos : c'est la phase d'apprentissage. À la fin de cette phase, la machine à développer ses propres critères de choix et elle peut les appliquer à des nouvelles photos : c'est la phase d'inférence. Notons ici une grande différence dans la performance de l'apprentissage machine par rapport à l'apprentissage humain : là où moins de 10 photos suffisent à un enfant pour savoir reconnaître un chat, il en faut plusieurs milliers pour une machine !

En utilisant les données d'entraînement, la machine va construire une fonction d'apprentissage destinée à répondre à une situation donnée. Cette fonction peut être une fonction de décision (comme dans le cas des photos de chats) mais elle peut aussi consister à déterminer un (ou plusieurs) nombres.

On distingue trois grands type d'apprentissages machine en fonction des modalités de la phase d'entraînement.

- l'**apprentissage supervisé** : dans ce type d'apprentissage, la machine s'entraîne sur des données qui ont été au préalablement étiquetées par l'homme. Si on reprend l'exemple précédent, on fournira à la machine des images en indiquant celles qui représentent des chats et celles qui n'en représentent pas.

  Parmi les techniques d'apprentissage supervisé, on peut signaler l'apprentissage profond qui, schématiquement, consiste en une succession de modules tels que les résultats produits par les uns sont utilisés comme données d'entrée par les autres. Ce principe s'inspire directement de l'architecture du cerveau humain sous la forme de réseaux de neurones artificiels.
- l'**apprentissage non supervisé** : dans ce type d'apprentissage, on fournit à la machine des données non étiquetées et on la laisse repérer des régularités, des proximités, des corrélations pour construire elle-même la fonction d'apprentissage.
- l'**apprentissage par renforcement** : dans ce type d'apprentissage, la machine va faire des essais et bénéficier d'un système de récompenses/punitions en fonction du succès de ses actions. C'est, par exemple, le cas des programmes qui apprennent à jouer à des jeux du type jeu d'échecs ou jeu de go. Dans ce cas, la machine va jouer (contre-elle même par exemple) et, grâce au système de récompenses, créer une fonction d'apprentissage qui lui permettra de trouver l'action optimale en fonction de la situation de jeu. Ainsi, ici, la machine crée ses propres données d'entrée en jouant des parties d'entraînement.

Les différentes méthodes s'appuient en général sur des outils mathématiques pour construire la fonction d'apprentissage.

On peut en donner quelques exemples dans des cas particulièrement simples.

- **Utilisation de courbes d'ajustement** : lorsqu'on dispose de données reliant deux paramètres (population en fonction du temps, prix de l'immobilier en fonction du temps, risque de développer une maladie en fonction de la présent d'un certain marqueur, etc.), on peut essayer de trouver une fonction dont la courbe s'ajuste le mieux possible aux données. Cette courbe peut être une droite (comme on l'a vu dans le thème 3.4 pour le modèle linéaire) mais peut prendre d'autres formes. Les données d'entraînement permettent dans ce cas de déterminer un modèle qui sera d'autant plus pertinent que le nombre de données sera grand.
- **Méthode du (ou des) plus proche(s) voisin(s)** : en reprenant la situation précédente, dans le cas où on souhaite classer les données en deux groupes A et B, on peut placer les données d'entraînement sous la forme de points dans un repère et, lorsqu'on doit traiter une nouvelle donnée, on place le point correspondant dans un repère et on cherche son plus proche voisin, c'est-à-dire le point issu des données d'entraînement le plus proche du nouveau point. Si ce plus proche voisin fait partie du groupe A, on classe la nouvelle donnée dans le groupe A et, sinon, on la classe dans le groupe B.

  Une autre possibilité et de considérer non pas seulement le plus proche voisin mais les k plus proches voisins (où k est un entier impair). Si, parmi ces k plus proches voisins, une majorité appartient au groupe A, on classe la nouvelle donnée dans le groupe A et, sinon, on la classe dans le groupe B.
- **Inférence bayésienne** : il s'agit d'une méthode de détermination des causes à partir des conséquences basées sur des calculs de probabilités. On détaillera cette technique dans la paragraphe suivant.

IV. — L'inférence bayésienne

1) Principe

L'inférence bayésienne est une méthode de calcul permettant de déterminer les probabilités des causes à partir de probabilités de leurs effets. L'adjectif « bayésienne » provient du nom d'un révérend et mathématicien anglais du XVIIIe siècle, Thomas Bayes, a qui l'on doit notamment un théorème très utilisé en probabilité.

L'inférence bayésienne est utilisée en apprentissage automatique, notamment dans le cas de prise de décision : une personne ayant un test positif pour une maladie est-elle effectivement malade ? un courrier électronique ayant un contenu suspect doit-il être considéré comme un spam ?

L'inférence bayésienne va permettre, à partir d'un grand nombre de données, de fournir une réponse probabiliste à ces questions.

2) Exemple d'un test de dépistage

Pour un test de dépistage, on définit les deux caractéristiques suivantes :

- sa **sensibilité** qui représente la probabilité qu'une personne malade effectuant ce test ait un résultat positif ;
- sa **spécificité** qui représente la probabilité qu'une personne non malade effectuant ce test ait un résultat négatif.

Cependant, lorsqu'une personne a un test positif, ce qui importe est de savoir si elle est vraiment malade.

On va voir que cela dépend essentiellement de la **prévalence** de la maladie c'est-à-dire la proportion de personnes atteintes de la maladie dans la population.

Considérons, par exemple, une population de 100 000 habitants et une maladie dont la prévalence est 0,1%. Supposons qu'on dispose d'un test dont la sensibilité est 95% et la spécificité est 98%. Ainsi, dans cette population, il y a :

-
-
-
-

On peut rassembler ces valeurs dans le tableau de contingence suivant :

	Test positif	Test négatif	Total
Personnes malades			
Personnes non malades			
Total			100 000

Ainsi,

- la **valeur prédictive positive**, c'est-à-dire la probabilité qu'une personne ayant un test positif soit effectivement malade est égale à
- la **valeur prédictive négative**, c'est-à-dire la probabilité qu'une personne ayant un test négatif soit effectivement non malade est égale à

Ainsi, on constate que, même si la sensibilité et la spécificité sont très bonnes, la valeur prédictive positive est faible c'est-à-dire une personne testée positive a peu de chance d'être effectivement malade.

Ce paradoxe apparent s'explique par le fait que les valeurs prédictives dépendent en fait grandement de la prévalence de la maladie.

La formule de Bayes permet de montrer que, si un test a une sensibilité Se et une spécificité Sp alors la valeur prédictive positive de ce test est

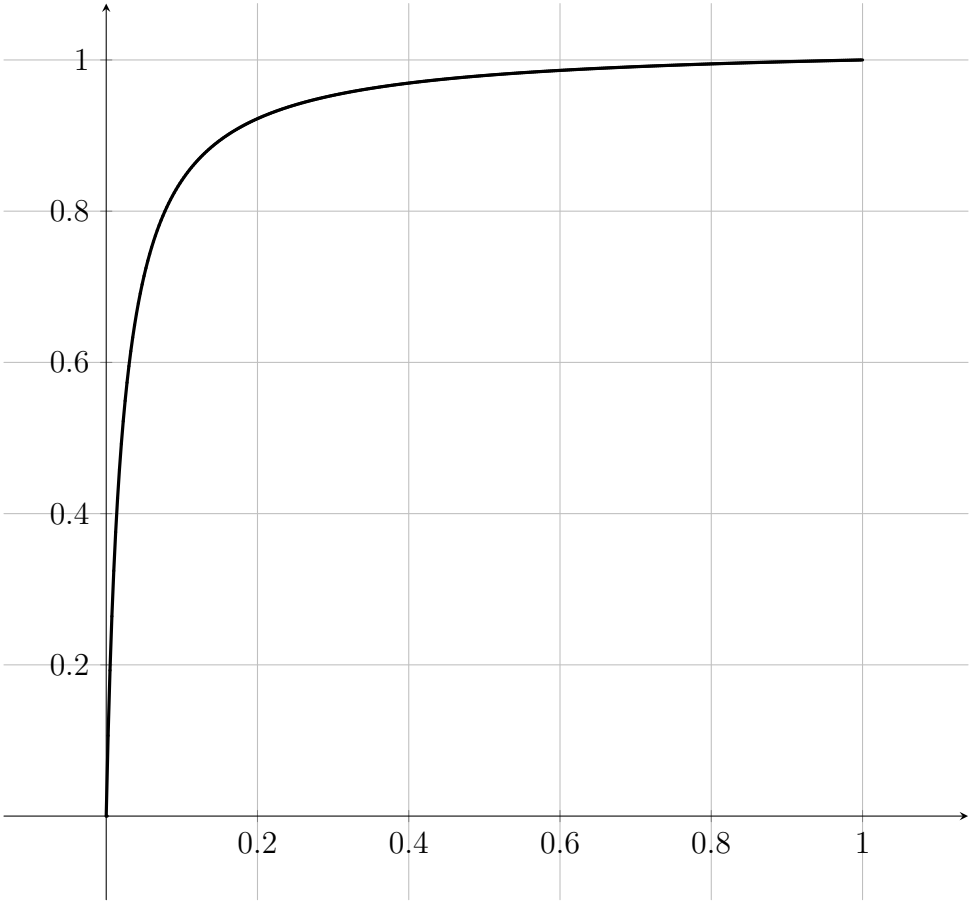
$$V = \frac{p \times \text{Se}}{p \times \text{Se} + (1-p)(1-\text{Sp})}$$

où p désigne la prévalence de la maladie dans la population.

Ainsi, si on considère le test précédent pour lequel $\text{Se} = 0{,}95$ et $\text{Sp} = 0{,}98$, on obtient une valeur prédictive positive en fonction de p égale à

$$\text{V}(p) = \frac{p \times 0{,}95}{p \times 0{,}95 + (1-p) \times 0{,}02} = \frac{0{,}95p}{0{,}93p + 0{,}02}$$

On a tracé ci-dessous la courbe représentative de la fonction V.



On constate que pour des prévalences faibles, la valeur prédictive positive est également faible, et ceci pour un sensibilité et une spécificité constantes.

3) Détection de spams

Un des premiers programmes de filtrage bayésien du courrier électronique a été le programme iFile conçu par Jason Rennie et publié en 1996. Le principe, analogue à celui du diagnostic médical, repose sur le fait que les mots du dictionnaire ont des probabilités différentes d'apparaître dans les spams et dans les courriers légitimes. Le filtre de détection des spams ne connaît pas à l'avance les probabilités d'apparition de ces mots, c'est pourquoi il lui faut une phase d'apprentissage pour les évaluer.

L'apprentissage se fait à partir de l'observation du comportement des utilisateurs, qui doivent indiquer manuellement si un message est un spam ou non. Le filtre ajuste les probabilités de rencontrer un mot M donné dans un spam ou dans un courrier légitime à l'aide des messages d'entraînement. Ensuite, en utilisant la formule de Bayes, la machine calcule la probabilité que le message soit un spam sachant qu'il contient le mot M. Cette probabilité est enfin comparée à un seuil : si elle est supérieure au seuil, le filtre classera ce message dans les spams.