

Algorithmes évolutionnaires et reconnaissance vocale

11 septembre 2005

Table des matières

1	Enjeux	2
2	Les algorithmes évolutionnaires	2
2.1	Généralités	2
2.2	Représentation	3
2.3	Evolution	4
2.4	Sélection	4
2.5	Reproduction & Mutation	4
3	La reconnaissance vocale	4
3.1	Difficultés majeures	4
3.2	Solutions abordées	4
4	Résultats	4

1 Enjeux

Le monde des systèmes de reconnaissance automatique de la parole devient de plus en plus intime du grand public. Les méthodes d'apprentissage des logiciels se basent sur la reconnaissance d'éléments caractéristique dans le spectre vocal de l'utilisateur. Cela permet ensuite d'isoler ce que l'on appelle des **phonèmes**¹. Les plus connus et utilisés actuellement sont : **ViaVoice** de IBM et **Naturally Speaking** de Dragons Systems Inc. L'inconvénient majeur qui les lie provient du fait qu'ils sont adaptés à un cadre précis d'utilisation. Ils ne fonctionnent pour la plupart que pour un utilisateur unique (cela provenant du fait que la phase d'apprentissage est spécifique à chaque voix, et qu'il est nécessaire de la répéter lors de tout changement d'utilisateur), et ils ne sont fiables (robustesse du logiciel) que lors d'un enregistrement en intérieur (les bruits parasites paralysant l'analyse des données).

Ces logiciels souffrent donc d'une lacune de conception qu'il pourrait être utile de combler. Nous nous proposons, au fil de cette étude, d'étudier l'utilité que pourraient avoir des algorithmes évolutionnaires sur l'amélioration de la phase d'apprentissage. Le dossier se décompose en deux parties majeures. La première est introduction à la théorie des algorithmes évolutionnaires, et la seconde aborde les procédures d'apprentissage d'un logiciel de reconnaissance vocale, ainsi que les méthodes que nous proposons pour simplifier cet apprentissage.

2 Les algorithmes évolutionnaires

2.1 Généralités

Certains problèmes scientifiques sont plus délicats à traiter que d'autres. C'est le cas en particulier des problèmes qui dépendent de nombreux paramètres. Les problèmes d'optimisation de systèmes font partie de ces "casses-tête" de l'ingénieur. Comment, en effet, être certain d'avoir choisi les meilleurs paramètres afin d'optimiser un système donné? La difficulté du problème grandit avec le nombre de degrés de liberté du problème. Un système d'une demi-douzaine de degrés de liberté soumet déjà les ingénieurs à un échec technique (il n'existe en effet pas de méthodes précises capables de résoudre ces problèmes). C'est sur ce constat que se sont développés les algorithmes évolutionnaires. Le principe reprend la théorie darwinienne de l'évolution sélective, orientée par la pression qu'exerce l'environnement sur les populations.

Principes fondamentaux : Le but principal de ces algorithmes est d'atteindre la meilleure adaptation possible à un ensemble de contraintes données. Le problème auquel est soumis un ingénieur est composé d'une part de données/paramètres, et d'autre part de contraintes/attentes que l'on impose. Par exemple si l'on cherche à obtenir le meilleur rapport vitesse/autonomie d'un avion, il faudra faire intervenir différents paramètres comme la forme de l'avion, le mode de propulsion, les altitudes de vol etc. Il existe souvent des liens entre chacun de ces paramètres comme le mode de propulsion qui impose en quelque sorte les altitudes de vol. Différentes stratégies ont été employées pour résoudre ces problèmes d'optimisation. La plus simple d'entre elles est la méthode d'essai et d'erreur. Un grand nombre de solutions potentielles sont mises à l'essai. Les solutions inadéquates sont éliminées, jusqu'à ce qu'un essai se révèle satisfaisant. On sélectionne ensuite parmi les solutions retenues celle qui correspond le mieux au but recherché. Cette méthode, bien sur, est automatisée par l'utilisation d'algorithmes. Cependant, le problème du nombre des solutions potentielles reste toujours présent : pour un problème avec de nombreux paramètres, le test de toutes les solutions possibles est fastidieux et peut nécessiter des ressources matérielles impressionnantes.

¹Pour en savoir plus sur les principes de ces logiciels visiter la page <http://www.vieartificielle.com/article/?id=191>

Cependant, cette démarche peut être améliorée par une analyse rapide du problème. On fixe tout d'abord les variables du problème. Leur nombre influe directement sur la multiplicité des solutions cherchées. On note ces variables x_1, \dots, x_n . Dans notre exemple : x_1 est la forme aérodynamique, x_2 est le mode de propulsion envisagé et x_3 est l'altitude de vol. On définit ensuite l'espace de recherche. Ce dernier correspond simplement à l'intervalle sur lequel peuvent varier les variables. Pour notre avion il serait absurde de lui donner une forme carrée dont nous connaissons pertinemment les médiocres résultats aérodynamique. Une étude superficielle du problème par les ingénieurs suffit à restreindre convenablement cet espace. L'algorithme va donc évoluer dans l'espace de recherche délimité par : $\forall k \in [1; n] \quad x_{kmin} \leq x_k \leq x_{kmax}$.

Il faut ensuite établir clairement la fonction d'adaptation que cherche à atteindre l'algorithme. L'algorithme aura pour but d'atteindre un optimum de cette fonction, c'est pour cela qu'il est nécessaire de la définir avec précision. Cette fonction prend bien sûr comme variables les x_k définis plus haut. Il se peut que la définition pose problème. Deux cas peuvent se présenter :

1. Le problème possède un objectif unique : la fonction correspond alors à la valeur de cet objectif. Si l'on décidait juste d'obtenir la meilleure autonomie pour notre avion, il suffirait de tester le temps de vol des solutions proposées. La plus longue serait alors sélectionnée, comme maximum de la fonction d'adaptation.
2. Le problème possède plusieurs objectifs : il est alors nécessaire de pondérer ces différents objectifs en fonction de leur importance. La fonction d'adaptation correspondra alors à la somme de chacun des objectifs attendus, associés à des poids. Choisir ces poids n'est pas tâche aisée, bien qu'il soit facile de classer par ordre d'importance les objectifs, les poids ne sont pas forcément des nombres entiers. Choisir de bons poids entraîne une convergence plus rapide des solutions vers la solution optimale, c'est pourquoi ce choix est décisif, et comme il n'existe aucune étude dans ce domaine, la meilleure solution est souvent le tâtonnement.

Les algorithmes évolutionnaires sont là pour éviter un travail de recherche non exhaustif et aveugle. Leur principe se fonde sur les théories de l'évolution énoncées par Darwin. L'idée fondamentale est qu'une génération initiale d'algorithmes est lâchée dans l'espace de recherche. La pression exercée par l'environnement ne s'exerce alors plus sur l'état de santé des individus, mais sur leurs performances vis-à-vis de la fonction d'adaptation. Ainsi, les individus ayant un meilleur taux de réussite (ceux s'approchant le plus d'une solution optimale) auront de meilleures chances de se reproduire dans la génération suivante. Cette reproduction artificielle s'accompagne elle aussi des mutations génétiques observées dans la nature. Ainsi au fil des générations, les individus (les algorithmes) s'adapteront de mieux en mieux (s'approcheront de plus en plus d'un extremum de la fonction d'adaptation) à leur environnement (les contraintes apposées).

Donc, quel que soit le problème abordé, les algorithmes évolutionnaires sont créés et se développent selon les étapes ci-dessous :

1. Une population aléatoire d'individus est initialisée.
2. Une sélection est effectuée parmi eux en fonction de leur performance
3. Une nouvelle génération est créée grâce à des croisements et des mutations

Les algorithmes génétiques sont ceux qui ont été le plus étudiés ² et dont nous nous proposons de faire usage ici.

2.2 Représentation

Un individu est codé par un génotype, composé de gènes correspondant aux valeurs des paramètres du problème à traiter. Le génotype de l'individu correspond à une solution potentielle au problème posé. On choisit alors de coder les gènes des individus grâce à des séquences binaires qui sont censées

²Lire pour cela les ouvrages de DE. Goldberg et de T.Bäck disponibles à la BMI de l'ENS rue d'Ulm

représenter les paramètres possibles du problème. Un gène peut donc, dans la grande majorité des cas, représenter un entier (ou réel) qu'il est immédiat de coder binnairement. Ainsi le nombre 7 sera-t-il 0100 en binaire.

2.3 Evolution

2.4 Sélection

2.5 Reproduction & Mutation

3 La reconnaissance vocale

3.1 Difficultés majeures

3.2 Solutions abordées

4 Résultats