Cahier des Charges
IHM & Machine-Learning

Préparé pour : Bailly Gilles, Chercheur au LIP6.
Préparé par : Villerabel Mathias.
11 février 2017

TABLE DES MATIÈRES :

SPÉCIFICATIONS TECHNIQUE DES BESOINS ET EXIGENCES

Contexte et définition du problème

Aujourd'hui n'importe quels logiciels , navigateurs internet ou encore applications mobiles ne peuvent se passer de proposer de nouvelles fonctionnalités ou de revisiter leurs plus anciennes. Hors avant toute chose , peu importe les éléments que proposent ces fonctionnalités , peu importe qu'elles fonctionnent techniquement ou commercialement.
Il faut que les utilisateurs puissent y accéder aisément et de façon intuitive. Et cela repose sur une interface Homme-Machine. Les interactions Hommes-Machines sont tous les moyens et les outils mis en place pour permettre à un humain de contrôler ou communiquer avec une machine. L'objectif de ces interactions est de concevoir des plateformes ou des applications qui soient ergonomiques, efficaces et faciles à utiliser.
Et de nos jours on le voit bien ces interactions sont omniprésentes, au travail via des logiciels, dans la vie de tous les jours sur les smartphones ou les nouveaux objets connectés, à la maison via les nouvelles technologies de domotique. Souvent ces interactions sont rendues possibles par des actions listées dans un menu.

Ce menu devenu alors pierre angulaire de l'échange entre l'utilisateur et la machine doit être pratique, fonctionnel pour cet utilisateur. Comment le développeur peut il adapter ce menu ? Comment doit il ordonner les options de ce menu ?
Une approche serait de faire venir des utilisateurs d'enregistrer leurs choix et de tenter d'optimiser le menu en fonction de ces dits choix. Cependant cela reste couteux et long, et le devient encore plus si on souhaite prendre une plus grande portion d'utilisateurs pour être plus précis.

Il y ici deux problématiques, l'une est de modéliser efficacement le comportement d'un utilisateur face à un menu , et l'autre de comparer ce comportement face à un autre menu.
Cette comparaison par exemple en terme de temps de sélection donnerait une base pour le développeur pour améliorer son menu.

Objectifs

Nous souhaitons mettre en place un algorithme nous permettant d'apprendre le comportement d'utilisateurs sur différents menus ordonnés par ordre alphabétique , ou sémantique. Nous comptons ensuite comparer ce modèle face à d'autres menus et en déterminer son efficacité en temps pris pour sélectionner des éléments du menu.

Notre code à pour objectif d'être modulable, des éléments pourront facilement y être ajouter ou enlever (ajout de vecteurs dans les états, contraintes supplémentaires, etc …).

Condition d'utilisation

Les utilisateurs de cette application sont attendus avec un bon niveau de compréhension en implémentation d'algorithme, il s'agit d'un utilisateur de niveau expert.
Nous concentrerons sur la recherche visuelle d'un menu , c'est à dire les informations visualisées par un humain lors de son parcours d'un menu.
Nous ne prendrons pas en compte l'aspect de pointage dans la sélection du contenu d'un menu.

Description fonctionnelle

L'utilisateur pourra ajouter des données aux états (state vector).

- Ajouter des éléments (items).

  Permet l'ajout d'éléments au menus.

  Exemple : ajout de l'élément Minimise.

- Ajouter des informations supplémentaires.

  Permet l'ajout de vecteur aux éléments du menu.

  Exemple : ajout de l'information de la taille , ou encore si il est un synonyme.

L'utilisateur pourra lancer l'apprentissage d'un comportement en fonction d'un menu organisé par différents ordres.

- Lancer l'apprentissage du comportement sur un menu.

  Permet de rendre le comportement appris d'un utilisateur.

  Exemple : Comportement d'un utilisateur sur un menu trié par ordre Alphabétique.

- Afficher les résultats des fonctions d'évaluations au cour de l'apprentissage.

  Permet une représentation graphique des choix pris par l'algorithme

  Exemple : augmentation de la valeur d'une courbe si un statut est choisi

L'utilisateur pourra comparer son modèle sur différents types de menu

- Renvoyer le temps de recherche d'un modèle sur un menu

  Permet de comparer avec une valeur numérique de retour la vitesse de recherche d'éléments dans un menu.

  Exemple : un comportement appris sur un menu ordonné alphabétiquement appliqué à un menu ordonné alphabétiquement puis sémantiquement et inorganisé

ESQUISSE DE SOLUTIONS

Piste de solutions existantes

Nous avons décidé d'utiliser les processus de décision markovien lié avec du Machine-Learning permettraient de modéliser les états possibles dû au choix d'un utilisateur et d'en dresser son comportement.

Afin de modéliser ce comportement nous utiliserons des états de menu et les actions menant à ces états.

Un état est défini comme étant un vecteur de taille égale aux nombres d'items d'un menu.

Il peut prendre pour valeur Null, 0.0 , 0.33 , 0.66 , 1.0 pour chaque élément du vecteur.

Ces actions (fixer un élément ou le sélectionner, ou encore n'en choisir aucun) auront des valeurs allant de 0 à Nombre d'item +1.

L'état Initial serait donc pour 4 Items : [Null , Null , Null , Null]

Et prendre l'action 0 à cet état (fixé l'item 0) donnera l'état suivant :

[0.0 , Null , Null , Null]

Avec 0.0 la valeur indiquant à si l'item est proche de la cible. La valeur 1.0

étant celle indiquant que l'item considéré est celui cherché.

Ces valeurs seront apprises en utilisant une table d'apprentissage et les processus de décision markovien , ou apprentissage par renforcement.

Cet méthode d'apprentissage repose sur le principe de donné un score à une action donné.

Les chaines de Markov permettent une représentation d'un état et des actions considérées qui mènent à un état.

Architecture logicielle

Pour l'implémentation de notre algorithme, nous prendrons les choix suivants :

- Python version 3.** comme langage de programmation.
- Numpy, matplotlib comme bibliothèques.
- SQLITE comme moteur de base de donnée.