**Data Mining SofameHack project**

Quentin Rivollat et Guy-Raphaël Stauffer

# Introduction

Le but de ce travail est, à partir de donnée récoltée par des capteurs, sur des patients en train de marcher, d’apprendre à prédire quand un patient touche et quitte le sol avec un de ses pieds.

# Données

## Présentation des données

Les données fournies sont récoltées par des capteurs placés sur toute la corp d’un patient. Les capteurs transmettent leurs positions tandis que le patient marche quelques pas. Les données contiennent aussi quatre types d’événements possédant deux étiquettes indiquant si l’événement correspond au pied droite ou gauche et s’il touche ou quitte le sol.

Les événements ne sont cependant présents que sur une partie du trajet effectué par le patient, et ne sont pas toujours enregistrés. Le but du projet est d’apprendre à prédire où se trouve les événements sur le trajet du patient où ils ne sont pas enregistrés.

Les patients sont regroupés en trois pathologies étiquetées « CP », « FD », et « ITW » et regroupant 45, 25 et 20 patients respectivement.

## Traitement des données

Pour pouvoir utiliser ces données, il faut dans un premiers temps les traités pour extraire les informations pertinentes pour la prédiction des événements.

Pour commencer, nous avons décidé de nous concentrer exclusivement sur les données relatives à la position vertical des capteurs, en ignorant leur positionnement horizontal. Il nous semblait effectivement plus pertinent d’utiliser la position verticale pour prédire les moments où le pied touche ou quitte le sol.

Pour obtenir des points précis dans l’ensemble des données de l’axe verticale, nous avons aussi fait le choix de nous concentrer sur les extremums locaux des courbes. En effet les extremums sont des points faciles à identifiés, et certainement significatif pour prédire les événements.

Il est aussi facilement possible de retirer des paires d’extremums parasites apparaissant au milieu d’une courbe ascendante ou descendante.

Finalement nous avons décidé de nous concentrer seulement sur certains capteurs dont les extremums étaient assez réguliers pour être fiables. Les capteurs de talons par exemple possèdent un maximum et un minimum par pas de manière très fiable.

Les capteurs retenus dépendent de la pathologie du patient, car certaines pathologies rendent certains capteurs extrêmement irréguliers.

# Approche 1

Prédire les événements directement à partir d’un capteur, puis apprendre les poids pour rassembler les capteurs

# Résultat 1

# Réseaux de Neurones

Une deuxième approche que nous avons essayée est d’utilisé un réseaux de Neurone.

En réalité nous avons entrainer des réseaux de neurones différents pour chaque type d’événements et pour chaque pathologie, pour un total de 12 réseaux de neurones.

## Traitement des données :

Pour commencer, il faut définir les données qui vont être transmises au réseau de neurone. Comme chaque réseau de neurone veut prédire un événement par pas, nous avons cherchés à définir un pas.

Pour cela nous avons utilisé, comme dit plus haut les extremums de l’axe z des données récoltée par les capteurs. En particulier pour définir les limites d’un pas, nous avons utilisé le capteur de la cheville, dont les maximums et les minimums sont très réguliers.

Ainsi donc, un réseau de neurone cherchant à prédire un événement lié au pied gauche va utiliser pour définir le début et la fin d’un pas deux maximums consécutifs du capteur du talon du pieds gauche. Les données des chevilles étant régulières, on peut être certains que l’événement concerné doit avoir lieux entre le début et la fin du pas.

Il nous reste encore, pour pouvoir injecter ces données dans un réseaux de neurones, à extraire les caractéristiques d’un pas dans une forme standard. Les caractéristiques que nous avons décidé d’utiliser sont les frames où l’on peut observer les maximaux et les minimaux de l’axe z des capteurs.

Pour chaque pathologie, nous avons sélectionner un certain nombre de capteurs dont les maximums et minimums étaient assez régulier pour pouvoir être raisonnablement certains qu’ils posséderont un maximum et un minimum dans chaque pas.

Nous avons aussi rajouté une certaine tolérance à la définition d’un pas afin de palier au fait que certains maximums se situent toujours près des limites d’un pas, parfois juste avant, parfois juste après. Le maximum de la cheville droit se situe par exemple toujours plus ou moins au même moment que celle du talon droit, qui marque la limite du pas. En ajoutant cette tolérance, nous nous assurons de toujours inclure le maximum de la cheville dans le pas, qu’il se trouve juste avant ou juste après le maximum du talon. Cette tolérance a été traitée comme un hyper paramètre lors de l’entrainement des réseaux de neurones.

Malgré l’ajout de la tolérance, il peut arriver qu’on ne trouve pas sur certaines instances un des extremums recherchés pour former un pas standard. Pour satisfaire à l’exigence d’une forme standard nous avons alors choisis de remplacer la donnée manquante par une valeur arbitraire de « 0 » en laissant le soin aux réseaux de neurones de gérer ces données corrompues.

Certains pas sont cependant trop corrompus, et, pour ne pas trop perturber l’apprentissage des pas non corrompus, nous avons décidé de supprimer les pas possédant plus d’un quart données manquantes.

Finalement les données sont normalisées pour être adaptée au réseau de neurones. On soustrait à chaque frame représentant un extremum (ou en événement) la frame de début du pas, avant de diviser le tout par la longueur du pas. Les données sont donc comprises entre 0 et 1. Il est facile à partir des frames de début et de fin du pas de dénormalisé les résultats pour obtenir la prédiction d’une frame précise.

## Paramètres des réseaux de Neurones

Les réseaux de neurones prennent en entrées un nombre différent de données par instance pour chacune des pathologies en fonction du nombre de capteurs fiables utilisés pour extraire les caractéristiques du pas. Ces données sont ensuite transmises à la première couche de neurone. Le nombre de neurones de cette première couche est traitée comme un hyper paramètre et déterminé par la méthode de la cross-validation. La fonction d’activation utilisée et ReLu.

La deuxième couche de neurones ne comporte qu’un neurone, dont la sortie représente le moment prédit pour l’évènement. Une sigmoïde est utilisée comme fonction d’activation.

Les données étant normalisée, on peut utiliser l’erreur moyenne des erreurs au carrés comme fonction de coût, sans obtenir des gradients disproportionnés par rapports aux poids du réseau.

Pour éviter le phénomène d’overfitting, qui nous guète particulièrement à cause du peu de données disponibles, nous avons rajouté à la fonction de coût un regularizer.

Comme nous soupçonnons que certains extremums, sont trop variable où pas assez significatifs pour être utilisés pour prédire l’événement, nous voulions encourager le modèle à ignorer les données qu’il jugeait inutile, nous avons donc opter pour un regularizer L1.

## Hyper paramètres

Comme mentionné plus haut nous avons utilisé la méthode de la cross validation pour estimer les meilleures valeurs pour nos hyper paramètres.

Le choix de cette méthode est dû au peu de données disponibles pour l’entrainement des réseaux de neurones. Non seulement nous avons peu d’instances par pathologies, mais s’ajoute à ça le fait que certaines instances ne comportent pas tout les types d’événements et ne peuvent donc pas être utilisées pour l’entrainement et le test du réseau de neurones cherchant à prédire les événements manquants.

Pour chaque réseau de neurones nous avons donc sélectionner les instances de la pathologie visée ayant l’événement visé, et nous avons divisés ces instances en deux ensembles. L’ensemble de teste comprenant un quart des instances et l’ensemble d’entrainement comprenant le reste des instances.

Nous avons ensuite appliqué la cross-validation à l’ensemble de teste pour estimer les meilleures valeurs des hyper paramètres avant de tester notre précision sur l’ensemble de teste.

# Résultats

|  |  |  |
| --- | --- | --- |
| Pathologie | Événement | Erreur moyenne sur l’ensemble de teste. |
| CP | 'Left', 'Foot\_Strike\_GS' | 4 |
| CP | 'Left', 'Foot\_Off\_GS' | 4.3 |
| CP | 'Right', 'Foot\_Strike\_GS' | 10 |
| CP | 'Right', 'Foot\_Off\_GS' | 8.5 |
| FD | 'Left', 'Foot\_Strike\_GS' | 2 |
| FD | 'Left', 'Foot\_Off\_GS' | 1 |
| FD | 'Right', 'Foot\_Strike\_GS' | 9 |
| FD | 'Right', 'Foot\_Off\_GS' | 3.25 |
| ITW | 'Left', 'Foot\_Strike\_GS' | 2 |
| ITW | 'Left', 'Foot\_Off\_GS' | 1 |
| ITW | 'Right', 'Foot\_Strike\_GS' | 1 |
| ITW | 'Right', 'Foot\_Off\_GS' | 2.25 |

Les résultats présentés ici sont à prendre ave du recule. Comme nous l’avons dit plus haut le nombre de données étant très réduit, les ensembles de teste et d’entrainement sont aussi très petit. Ils ne représentent donc pas nécessairement la réelle distribution des données, et les hyper paramètres sélectionnés par la méthode de la cross-validation peuvent beaucoup varier en fonction de quelles instances se trouvent dans l’ensemble de teste ou dans l’ensemble d’entrainement. En conséquence les précisions observées peuvent être amenés à beaucoup varier en fonction de la répartition dans les différents ensembles. Ce problème disparaitrait cependant si l’on disposait d’un plus grand nombre de données.

Malgré cela on observe tout de même que les résultats obtenus varient passablement selon la pathologie concernée. Ainsi la précision d’une ou deux frames observés pour tous les cas de la pathologie ITW peut être considérée comme satisfaisante. Les résultats observés pour la pathologie FD, bien que plus variables peuvent encore être considérés comme acceptables.

En ce qui concerne la pathologie CP, les résultats sont bien moins bon. Ceci est sûrement dû à la plus grande difficulté que nous avons rencontré à trouver des capteurs réguliers. Il est aussi possible que la méthode que nous avons choisie de se concentrer sur les extremums de l’axe z des capteurs ne soient pas adapté à cette pathologie.

# Conclusion