

# Data Mining SofameHack project

Quentin Rivollat et Guy-Raphaël Stauffer

## Introduction

Pour des personnes qui n'ont aucun handicap, détecter des événements lors d'une marche, c'est-à-dire poser le pied et le lever, est relativement simple. Quelques capteurs sur le pied suffisent, et les résultats restent néanmoins précis. Cependant, l'utilisation de si peu d'outils sur des personnes atteintes de handicaps moteurs est plus compliquée. Ces handicaps peuvent prendre plusieurs formes : cela va de simples déséquilibres pendant la marche, causé des mesures irrégulières, mais utilisables, jusqu'à des personnes nécessitant des aides pendant tout le déplacement, rendant les données plus difficilement exploitables. Pour avoir des mesures précises de ces événements, une façon possible serait la façon manuelle, mais cela serait extrêmement long, et assez fastidieux. Une solution à cela pourrait être de développer des algorithmes de détection automatique, en utilisant les principes utilisés fréquemment en Data Mining. En effet, ici, avec toutes les informations recueillies par les capteurs, ces algorithmes pourrait s'avérer, si ce n'est exact, plutôt efficaces.

Ainsi, le but de ce travail est, à partir de données récoltées par les capteurs situés à divers positions 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 le corps d'un patient. Les capteurs transmettent leurs positions dans l'espace par leur localisation sur les 3 axes, correspondant aux 3 dimensions, c'est-à-dire ici x, y et z, tandis que le patient marche quelques pas. Les données, situées dans des fichiers de type C3D, contiennent au maximum quatre types d'événements possédant deux étiquettes indiquant si l'événement correspond au pied droit ou gauche ('Right' ou 'Left') et s'il touche ou quitte le sol. ('Strike' ou 'Off').

Les événements ne sont cependant présents que sur une partie du trajet effectué par le patient, et ne sont pas toujours enregistrés, ce qui rend le travail un peu plus difficile. 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 traiter pour extraire les informations les plus pertinentes pour la meilleure prédiction possible des événements. Pour commencer, nous avons décidé de nous concentrer exclusivement sur les données relatives à la position verticale des capteurs, c'est-à-dire sur l'axe z, 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. Les données horizontales x changent très peu, car elles correspondent à la direction de la marche du patient (s'il va sur les côtés, droite/gauche), qui ne change quasiment pas. Et les données horizontales y correspondent au sens de la marche (avant / arrière), ce qui donnent des données très linéaires, et ainsi difficilement exploitables.

Pour obtenir des points précis dans l'ensemble des données de l'axe vertical, nous avons aussi fait le choix de nous concentrer sur les extremums locaux des courbes, que sont les maximums et minimums. En effet les extremums sont des points faciles à identifier, et certainement significatifs 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 et ainsi inutilisables

## Première approche

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

Une première approche que l'on a eu est la plus simpliste des deux. Une première étape, comme indiqué précédemment, fut de trouver d'abord les capteurs les plus fiables, les plus réguliers, et qui éviteront toute erreur grossière. Puis, une fois ces capteurs trouvés, on parcourt chaque événement, et on fait les opérations suivantes : on trouve l'extremum le proche (on indiquera si l'on préfère que ce soit un max ou un min), on mesure la distance, en frames, qui les sépare, et ce sur tous les fichiers trainings, afin de faire une moyenne, la distance moyenne event-extremum. Une fois cela fait pour tous les capteurs (et tous les fichiers d'entraînement, donc), chaque capteur est déjà capable de faire des prédictions des événements, en trouvant tous les extrémaux qui l'intéressent, et en ajoutant la distance moyenne, pour indiquer où les autres événements semblent se situer, puis on fait la moyenne de la prédiction de chaque capteur, pour obtenir une unique prédiction pour chaque pas. Avec cela, on obtient déjà une bonne base, avec des résultats autour de 2 frames d'erreur pour ITW, 3 frames pour FD, et 4 frames pour CP.

Ensuite, pour améliorer la précision, on va faire intervenir des poids. En effet, en faisant une simple moyenne des prédictions de tous les capteurs, on estime ainsi que tous ces capteurs ont autant d'importance les uns par rapport aux autres, ont un jugement égal. Or, ce n'est pas forcément le cas. Même si l'on a sélectionné les capteurs qui nous semblaient les plus fiables, certains font généralement de meilleurs prédictions que les autres. Il faut alors

mettre en avant ces capteurs, en leur donnant un poids plus important lors de la moyenne. De la même manière, on donnera un poids plus faible aux capteurs les moins précis. Ces modifications des poids se font après le parcours de chaque fichier training : après chaque prédiction, on fait un classement des capteurs, du plus précis ou moins précis, et on fait augmenter le poids, légèrement, du premier, et inversement on diminue le poids du dernier.

On peut faire cette opération une seule fois, car à chaque essai, les poids convergent très vite vers leur valeur finale.

## Résultats

Voici les résultats moyens obtenus avec cette première approche :

Pathologie	Événement	Erreur moyenne sur l'ensemble de teste.
CP	'Left', 'Foot_Strike_GS'	2.1
CP	'Left', 'Foot_Off_GS'	5.4
CP	'Right', 'Foot_Strike_GS'	3
CP	'Right', 'Foot_Off_GS'	7.6
FD	'Left', 'Foot_Strike_GS'	2.3
FD	'Left', 'Foot_Off_GS'	3.2
FD	'Right', 'Foot_Strike_GS'	3.6
FD	'Right', 'Foot_Off_GS'	1.2
ITW	'Left', 'Foot_Strike_GS'	1.5
ITW	'Left', 'Foot_Off_GS'	1.1
ITW	'Right', 'Foot_Strike_GS'	2.1
ITW	'Right', 'Foot_Off_GS'	1.2

Pour obtenir ces résultats, le même algorithme (présenté plus haut) a été utilisé. La seule différence est les capteurs utilisés pour faire ces prédictions.

Ces résultats sont globalement assez bons pour un modèle aussi peu complexe.

Néanmoins, il s'agit de moyenne. Ceci est à prendre en compte, car pour la détection de certains capteurs, on a obtenu des erreurs très importantes (situations rares, mais qui existent), ce qui signifie que les autres erreurs sont encore plus précises que la moyenne indiquée.

Les meilleurs résultats sont clairement obtenus avec la pathologie ITW, et cela pour la simple raison que les données captées sont les mieux exploitables, car plus nombreuses et régulières, évitant ainsi de fortes erreurs de prédiction. Avec des erreurs de l'ordre d'environ une frame et demi dans l'ensemble, nous sommes proches d'une excellente prédiction, pour des données allant jusqu'à 300 frames de long.

Vient ensuite le groupe FD : ici, les prédictions restent globalement assez bonnes, mais pas aussi précises que dans ITW. Ces 3 frames d'erreurs globales montrent la fébrilité de ce modèle lorsque la quantité de données diminue, tout comme sa précision.

Et finalement, les tests sur la pathologie CP : ici, on obtient des résultats assez mauvais en comparaison aux deux autres, Jusqu'à 7 frames d'erreur parfois. Cela montre clairement la limite de ce modèle lorsque les données manquent, et que beaucoup d'entre elles sont de "mauvaise qualité", c'est-à-dire entraînant beaucoup d'erreur, par leur manque de fiabilité.

## Réseaux de Neurones

Une deuxième approche que nous avons essayée est d'utiliser un réseau de neurones.

En réalité nous avons entraîné 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 neurones veut prédire un événement par pas, nous avons cherché à définir un pas.

Pour cela nous avons utilisé, comme dit plus haut, les extremums de l'axe z des données récoltées 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 neurones 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 pied gauche. Les données des chevilles étant régulières, on peut être certain que l'événement concerné doit avoir lieu entre le début et la fin du pas.

Il nous reste encore, pour pouvoir injecter ces données dans un réseau 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 sélectionné un certain nombre de capteurs dont les maximums et minimums étaient assez réguliers 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 pallier 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 droite 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'entraînement 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 choisi 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 de données manquantes.

Finalement les données sont normalisées pour être adaptées 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 dénormaliser 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 traité comme un hyper paramètre et déterminé par la méthode de la cross-validation. La fonction d'activation utilisée est 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 normalisées, 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 guette 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 variables ou pas assez significatifs pour être utilisés pour prédire l'évènement, nous voulons encourager le modèle à ignorer les données qu'il jugeait inutile, nous avons donc opté 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'entraînement 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 tous les types d'événements

et ne peuvent donc pas être utilisées pour l'entraînement et le test du réseau de neurones cherchant à prédire les événements manquants.

Pour chaque réseau de neurones nous avons donc sélectionné les instances de la pathologie visée ayant l'événement visé, et nous avons divisé ces instances en deux ensembles. L'ensemble de test comprenant un quart des instances et l'ensemble d'entraînement comprenant le reste des instances.

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

## Résultats

Pathologie	Événement	Erreur moyenne sur l'ensemble de test.
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 avec du recul. Comme nous l'avons dit plus haut le nombre de données étant très réduit, les ensembles de test et d'entraînement sont aussi très petits. 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 test ou dans l'ensemble d'entraînement. En conséquence les précisions observées peuvent être amenées à beaucoup varier en fonction de la répartition dans les différents ensembles. Ce problème disparaîtrait 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 observée 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 bons. Ceci est sûrement dû à la plus grande difficulté que nous avons rencontrée 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 adaptée à cette pathologie.

## Conclusion

En comparant les résultats de nos deux méthodes, on se rend compte que la première méthode, pourtant moins complexe que la méthode des réseaux de neurones, donne des résultats comparables voire meilleurs, à l'inverse de ce à quoi on pouvait s'attendre au départ.

Ceci peut être dû au manque de quantité de données nécessaires à une pleine exploitation du potentiel d'un réseau de neurone. Cette idée est appuyée par le fait que l'on obtient de meilleurs résultats lorsque l'on a plus de training.

Il est possible aussi que le problème consistant à prédire des événements à partir des extremums des capteurs ne nécessite pas une solution aussi complexe qu'un réseaux de neurones.

Le choix de se concentrer sur l'axe z et sur les extremums pour traiter les données a donné de bons résultats pour la pathologie ITW, car les données correspondantes étaient dans l'ensemble fiables et très régulières, nous permettant d'utiliser de nombreux capteurs, et d'éviter des erreurs importantes, comme ce fut le cas des deux autres pathologies

Pour la pathologie CP, les résultats obtenus sont médiocres avec les deux méthodes, ce qui est dû à la difficulté que nous avons rencontrée à trouver des capteurs réguliers et significatifs. Ainsi, la diminution du nombre de capteurs résulte logiquement en une perte nette de précision. Ceci indique que le traitement des données que nous avons choisies n'est pas très adapté aux données de cette pathologie.

Des observations similaires peuvent être faites avec FD, car les capteurs sont tout aussi irréguliers qu'avec CP, et donc tout aussi difficilement exploitables.