Automatically characterizing driving activities onboard a smart wheelchair from accelerometer data

HiuKim Yuen

Master of Science

Computer Science

McGill University

Montreal,Quebec

2014-06-20

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Master of Science

DEDICATION

This thesis is dedicated to my parents for their support and engagement for me to study abroad and pursue my dreams.

ACKNOWLEDGEMENTS

I would like to acknowledge the invaluable guidance provided by my supervisor Joelle Pineau. She is an insightful, supportive and patient mentor who helped me pick up the research work quickly. I was very graceful and lucky to be able to work with such an awesome professor throughout my Master degree, given my lack of prior research experiences.

I also feel proud to be a member of the Reasoning and Learning Lab in McGill University. Not only because of the many intelligent people around, but also the supportive environment in which I can always conduct valuable discussion for research problems.

Finally, I have to express my special thanks to my family and friends, who have kept me motivated throughout the whole time. Without them, my life in Montreal would not be fantastic.

ABSTRACT

Wheelchairs play an important role for people living with locomotor impairments. However, powered wheelchair users frequently report both minor and major accidents. The goal of this thesis is to advocate for the use of robotic technology, in particular sensor-based detection and automatic classification of activities, to track and characterize activities onboard smart wheelchairs.

This thesis presents an end-to-end pipeline for accurately detecting and classifying wheelchair activities from accelerometer data using signal processing and machine learning methods. In the first step, a datalogging platform is installed on a commercially available power wheelchair that records accelerations in 3 directions. After that, fast fourier transform, together with some other processing steps, are applied to produce a feature vector representing the activities. A classifier is then trained on top of the feature vector to categorize different activity types. Besides, we also explore the possibility of discovering hidden patterns of activities using unsupervised topic modeling methods.

Our methods are validated by empirical results; Experiments were conducted in a clinical setting, in which experienced wheelchair users were asked to conduct a set of typical wheelchair activities. In a 25-class classification problem, we achieved around 50% accuracy in categorizing activity types. We also qualitatively and quantitatively show that topic modeling provides good insights on characterizing wheelchair activities. Altogether, this work provides new tools and methods for characterizing the usage of smart wheelchairs, or potentially other mobile robots.

ABRÉGÉ

Les chaises roulantes jouent un rôle important pour les gens présentant des difficultés de locomotion. Cependant, les utilisateurs de chaises roulantes électriques rapportent des accidents autant mineurs que majeurs. Le but de cette thèse est de promouvoir l'utilisation de technologies robotiques, plus précisemment la détection à base de senseurs et la classification automatique d'activités dans le but de traquer et caractériser les activitées à bord de chaises roulantes intelligentes.

Cette thèse présente une méthode pour détecter et classifier précisemment les activités des chaises roulantes à partir de données d'accéléromètre en utilisant des méthodes d'apprentissage automatique et de traitement de signaux. D'abord, une plateforme de collecte de données enregistrant dans trois directions est installée sur un fauteuil roulant électrique commerciale. Ensuite, une transformation rapide de Fourier ainsi que quelques étapes de traitement sont appliquées pour produire un vecteur de caractéristiques représentant des activitées. Un classificateur est ensuite entraîné sur ces vecteurs afin de catégoriser différents types d'activités. Nous explorons aussi des moyens de découvrir des motifs cachés d'activités en utilisant des méthodes d'apprentissage non-supervisé de "topic modelling".

Nos méthodes sont validées par des résultats expérimentaux; les expériences ont été conduites dans un environnement clinique où des usagers de fauteuils roulants se sont vu demander de conduire un ensemble d'activités typiques pour une personne en fauteuil roulant. Dans un problème de classification avec 25 classes, nous avons atteint environ 50% de précision dans la caractérisation de types d'activités. Nous montrons aussi de façon qualitative et quantitative comment le "topic modelling" produit de bons indices dans la catégorisation d'activités de fauteuil roulant. Ce travail montre de nouveaux outils et de nouvelles méthodes pour caractériser l'usage de fauteuils intelligents ou, potentiellement, d'autres robots mobiles.

TABLE OF CONTENTS

LIST OF TABLES

Table page

LIST OF FIGURES

Chapter 1
Introduction

Mobility plays an important role in social participation and quality of life. For individuals who live with locomotor impairments, mobility can be facilitated by the optimal use of assistive devices such as powered wheelchairs (PW) [10]. However, PW users frequently report both minor accidents, such as colliding with people, furniture and walls, and major accidents such as tips and falls, which can lead to serious injuries [11]. In order to provide better assistance to this population, the design of intelligent powered wheelchairs using robotics and intelligent system technologies, has received significant attention from the robotics community in recent years [13].

During the last decade, significant research on intelligent wheelchairs has focused on the design and control aspects, including but not limited to human-machine interfaces and autonomous navigation. However, due to the fact that wheelchair-related accidents are not uncommon [19], we believe that monitoring is an equally important aspect in the development of intelligent wheelchairs, or assistive robots in general.

In fact, monitoring plays a very important role in the users' training process of PW. Given limited number of training sessions between clinicians and patients before a decision is made whether the patients are suitable for controlling the PW on their own, it is important for the clinicians to receive as much useful information as possible. In this regard, an automatic system to characterize driving activities would be helpful because it provides an objective and comprehensive summaries on the patients' driving experiences.

1.1 Problem Definition

With the goal of developing a full-fledged monitoring system that can characterize wheelchair activities and evaluate safety performances during the use of intelligent wheelchairs, this thesis presents an end-to-end pipeline, from capturing sensor data to automatic activity recognition, together with empirical validations. More specifically, in term of activity recognition, we have two branches, namely activity classification and pattern discovery.

1.2 Methods

In activity classification, we assume each activities are well defined in the sense that a clear activity type is associated with a known period of sensor input. The objective is to figure out the properties of the sensor input associated with each activity types, so an effective classifier can be built in order to categorize any unknown activities. We applied Fast Fourier Transform to convert the time series signals into frequency domain features. With some further manipulation, we built effective classifiers using support vector machine and nearest neighbour.

In pattern discovery, we get rid of the assumption of well defined activities. Therefore, instead of categorizing activities, we are trying to extract semantics out of a whole stream of sensor input. We applied topic modeling, which is originally used in text pattern recognition, on our sensor data. We tried to infer higher level hidden topics and then further analyse the properties and constituents of the topics. Topic modeling is powerful in the sense that no manual annotation is required.

1.3 Experimental Results

For the purpose of the studies, a dataset of driving activities carried out by seven participants were collected in a clinical settings. Each of the participants have conducted a series of tasks, extracted from the Wheelchair Skills Test from the Dalhousie University [32]. With a total of around 25 types of tasks, our classifier achieved an accuracy of around 50%, compared to 4% with a random classifier. We also successfully demonstrated a few potential uses of topic modeling with this dataset, including examples of story telling and hazard discovery.

1.4 Contributions

Since autonomous navigation and interactive command systems used to be the main focus on the research of intelligent wheelchairs, the number of works conducted on activity recognitions, on the contrary, is very rare. As per our knowledge, this is the first attempt to apply topic modeling in wheelchair activities. This is also one of the first premiers attempting to train an activity classifier as well as to give a thorough evaluation on the performance with participations from real users in a clinical settings. The main contribution of our work is an end-to-end pipeline which is not only straight forward and easy to implement, but also transferable and extendable to other mobile robotic applications. Our contribution is validated during experiments with real PW users.

There are mainly 3 contributors for this work. The experimental dataset is collected by professor Philippe Archambault from the School of Physical & Occupational Therapy Department from McGill University. I, myself, worked on the methodologies and experiments with the provided datasets. My supervisor, Joelle

Pineau, also from the Computer Science Department, provided advice, guidance and collaboration on the project.

1.5 Thesis Organization

The organization of this thesis is as follows: In chapter 2, we summarize the related works, mostly on the area of intelligent wheelchair and activity recognitions. In chapter 3, we explain in details the machine learning and signal processing technologies required to understand our work. We then present the activity recognition methodologies in chapter 4, followed by empirical studies in chapter 5. We conclude our work and give some directions for future work in the last chapter.

Chapter 2
Related Works

There are two areas of research, namely smart wheelchairs and activity recognitions, that are closely related to this thesis. In the smart wheelchair section, we give a brief roadmap of the development of smart wheelchairs since its early research prototypes. After that, we also mention a few recent smart wheelchair research projects/groups that are comparatively more related to our work. In the activity recognition section, we give a brief summary on research conducted in recent years, with a primary focus on using accelerometer data, which is what we used.

2.1 Smart Wheelchairs

Traditional Powered Wheelchairs (PW) are usually controlled by a joystick interface. However, to a lot of elderly and people with body disabilities, this could be a stressful and difficult task. In fact, surveys indicate that 85% of clinicians see some number of patients each year who cannot use a powered wheelchair because of the lack of motor skills, strength, or visual acuity [1]. As a result of these limitations, intelligent (smart) wheelchair evolved as an improved version since the early 1980s.

Early versions of smart wheelchair research prototypes are more like a mobile robot equipped with chairs, e.g. "Mister Ed" [52]. In "Mister Ed", users sit on an ordinary chair, which stick on top of a mobile robot, and the users control the robot via a hand-held joystick. The design focused on cooperative control between human and robot. By that, there were switches allowing the riders to authorize the robot to perform autonomous tasks like steering around obstacles, traversing hallways, turning at doors or following other moving objects.

Later, most research prototypes were developed by incorporating additional components on commercially available powered wheelchairs, e.g. NavChair [4]. The extra components incorporated in NavChair included a DOS-based computer, ultrasonic sensors and interface module interposed between the joystick and power module of the wheelchair. In order to accommodate a broad range of potential users with different disabilities, NavChair was designed to support a hierarchy of operating levels, each of which requires varying degrees of control from users. Simply put, it determines how autonomous the users want it to be. One of the biggest contributions of the NavChair project was their autonomous navigation module. Two obstacle avoidance methods, namely minimum vector field historgram (MVFH) and vector force field (VFF), were applied. The two methods were originally used in autonomous mobile robots, but modified to meet the specific needs of wheelchairs, for instance, to account for its' rectangular shape.

SMARTCHAIR is another early research prototype developed by the research group in GRASP laboratory [53]. Similar to NavChair, they also provided a shared-control paradigm with different levels of automation. In term of hardware, they included a omni-directional camera that allows users to view the surrounding on a display, on which the users can also give command via a visual interface. They had carried out experiments on the performance of their navigation systems, including navigation to targets, hallway navigation and doorway navigation. The empirical results mainly consisted of a comparison of navigation time and number of obstacle collisions between manual mode and autonomous mode. The main conclusion of their results was that their system provide faster response time to dynamic changes in the environments.

CALL [56], OMNI [55] and SCENARIO [54] were other early research projects on smart wheelchair, also focusing on semi-autonomous to fully-autonomous navigations. For further details on the early development of smart wheelchairs, we refer readers to a comprehensive literature review on smart wheelchairs before 2005 by Simpson [16]. Research on smart wheelchairs continue after that, focusing mainly on autonomous navigation and alternative command systems, which are however not the main focus of this thesis. So for the rest of this section, we will give a brief introduction to three other wheelchair research projects after 2005 which are more related to our work.

In 2008, T Carlson and Y Demiris designed an interactive smart wheelchair by predicting user intentions [57]. Unlike the prototypes mentioned previously, in which human-computer collaboration is accomplished based on providing different levels of automation, they approached the problem by predicting intentions and responding to those predictions with adaptable levels of assistance. The predictions were made by recognizing environments and movement properties like directions.

In 2009, another smart wheelchair project team called CanWheel was formed comprising of scientists, clinical researchers and trainees across Canada [58]. On a high level, CanWheel comprised of five main project areas. 1) Evaluating the Needs & Experiences of Older Adults using Power Wheelchairs. 2) The Natural History and Measurement of Power Mobility Outcomes. 3) Strategies and Platforms for Collaboratively-Controlled, Environmentally-Aware Wheelchair Innovation, 4) Activity and Status Monitoring System and 5) Evaluation of Safety, Efficacy and Impact of the Wheelchair. Unlike other project teams which focus more on the technology side, the CanWheel team pays more attention on user adaptability with the goal of bringing the smart wheelchair out of the laboratory into public use.

The work presented in this thesis is part of the smart wheelchair research group called the SmartWheeler [59]. The SmartWheeler project tries to tackle a range of challenging issues in mobile robotics, focusing on tasks pertaining to human-robot interaction and robust control like dialogue management. A wheelchair platform is built on top of a commercially available powered wheelchair Sunrise Quickie Freestyle, with the addition of a touch-sensitive graphical display, front and back laser-range finders and an onboard computer. The wheelchair platform is intended to be a test-bed for validating novel concepts and algorithms for automated decision making onboard assistive robots.

2.2 Activity Recognition

Activity recognition has experienced increased attention over the years due to its applications in numerous domains like medical diagnosis and video gaming. Pavan et al. has done a good survey summarizing different types of activity recognition methods [2]. For the purpose of introducing our works, we focus on activity recognition research that are based on accelerometer data because it is used in our methodologies.

2.2.1 Human Activities

Partly due to the advance of wearable sensor devices like smart phones, more research has been published on recognizing simple human activities using accelerometer data [27] [28]. Previous works mainly involved classifying simple activities like standing, walking and running, etc. by analysing the sensor data captured from accelerometers. The primary objectives are often on the evaluation of classification accuracies with different learning algorithms and features.

In contrast to simple activities as mentioned above, researchers are also interested in complex activities, which are usually defined as activities that by themselves are mixture of numerous simple activities, for instances, ‘going to work’ or ‘having lunch’. In general, recognizing high-level daily routines is a much harder problem, and per our knowledge, there are still no effective approaches published that could classify complex activities as good as the simple ones. In fact, the number of publications on complex activities recognition are relatively rare. In this thesis, we tried to tackle both simple and complex activities on wheelchairs.

Among those publications that attempted to tackle complex activities, T^am et al. proposed a novel approach in using topic modeling, which is originally used in text pattern discovery, to infer high-level daily routines as a probabilistic combination of activity patterns [29]. They modelled high-level daily routines as a combination of hidden topics (equivalent to hidden topics in the context of text documents). Their approach is powerful in the sense that user annotations is not required. In their paper, they gave some interesting qualitative results alongside the quantitative measures. Some of the idea in the pattern discovery section in this thesis is similar to their work.

2.2.2 Smart Wheelchair Activities

Moghaddam has done a comprehensive analysis on the classification of simple wheelchairs' activities using accelerometer data in her thesis [9]. She compared four different forms of features, including i) simple time domain features like mean and variance, ii) frequency domain features, computed from fast fourier transform, iii) wavelet transformed features, and iv) time delayed embedded features. Empirical results showed that frequency domain features and time delayed embedded features work best among the four. Her preliminary work on comparing different feature types provides a solid foundation for our choice of using fast fourier transform in this thesis.

The experimental dataset in her work consisted of a list of 35 activity types and 30 repeated trials of each type (i.e. 1050 activities in total). First of all, the 35 activity types were labelled as either 'Safe' or 'Unsafe' with prior knowledge, and the objective is to build a classifier that can identify an unseen activity being either 'Safe' or 'Unsafe', i.e a binary classification problem. Using 80% (24 trials out of 30) of data as training and the rest for testing, she achieved an error rate of less than 7% using frequency domain features.

In the activity classification section of this thesis, we are also working on a similar problem. However, the dataset we used is different in three main areas: First, the dataset used in our work is collected from real PW users in a clinical settings, whereas the dataset in [9] was collected by a healthy operator in a laboratory settings. Second, the activities carried out in our dataset are more varied and complex. For example, a complicated task ‘Go to local gym’ is included. Third, we only have 5 trials per each activity type per each participant in our dataset, compared to 30 trials in [9].

Chapter 3
Technical Background

In this chapter, we summarize the technical background required to understand the methodologies presented later in this thesis. We give a brief introduction on Machine Learning, together with four different Machine Learning tasks including Classification, Clustering, Dimensionality Reduction and Topic Modeling. Lastly, we discuss Faster Fourier Transform as a way to process accelerometer signals.

3.1 Machine Learning

As a branch of artificial intelligence, machine learning has shown successes in many different applications in recent years, including but not limited to robotic. [34]. The core of machine learning deals with generalization of data: given a subset of data instances within a larger pool, we would like to construct a summarization that could apply to the data instances outside the given subset [35]. This generalization is powerful in the sense that it allows us to gain insight on unseen data. In reality, it allows us to predict future events (unseen data instances) based on past experiences (given subsets). The ‘Learning’ aspect of Machine Learning comes from the idea that the generalization is learnt from data, and the process of learning the generalization is usually termed as ‘Training’.

Take Optical Character Recognition (OCR) as an example application [36]. The objective of OCR is to recognize characters or words in scanned or photographed images. This could be accomplished in many ways, but the Machine Learning approach does the work by observing samples. We feed the machines with sample images in which we know what characters they representing. With some learning algorithms (which we illustrate further in the following sections), we train a model that can take any unseen image as input and use the learner to predict the characters in the image.

To put it formally, define X to be the space of input variables and Y to be the space of output variables. Let's say X has k dimensions, in which we refer each dimension as a feature. The output variable Y, on the other hand, is usually one dimensional. Then $x_i \in X$ is an instantiation of input and it has a corresponding output y_i. We assume there is a true function which map inputs to outputs, i.e. $f(x_i) = y_i \quad \forall i$. However this true function is unknown. The objective of machine learning is then to find an approximation function $\hat{f}(x_i) = \hat{y_i}$ such that $\hat{y_i}$ and y_i are as close as possible in certain measurement metric. For example, squared error, i.e. $(\hat{y_i} - y_i)^2$ is a commonly used metric if Y is continuous. We often call this measurement metric loss function $L(\hat{y_i}, y_i)$ because it measures how deviate the predicted output is from the true output. The machine learning task is to learn this approximate function using a set of data in which both x and y are given.

3.2 Linear Regression

Let's consider linear regression to illustrate the above idea. Regression analysis is the statistical process to relate dependent variable and independent variables. In this context, dependent variable is output Y and dependent variables are input X (multiple variables because X is multi-dimensional) as mentioned before. In linear regression model, we assume y_i is a linear combination of the inputs x_i, and our objective is to find the set of parameters $\beta \in \mathbb{R}^{k+1}$ such that $\hat{y_i} = \beta_0 + \beta_1 x_1 + \beta_2 x2 +$

$... + \beta_k x_k$, $i = 1, 2, ..., n$. The parameters are evaluated using a loss function, in this case, the sum of squared error i.e. $\sum_i^n (\hat{y}_i - y_i)^2$. In the other words, the machine learning task is to find the set of β such that the loss is minimized [37]. It turns out that linear regression has a closed form solution. If we express the sum of squared error in matrix notations: i.e.

$$\sum_i^n (\hat{y}_i - y_i)^2 = (X\beta - Y)^T \cdot (X\beta - Y)$$

, where

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{pmatrix}, \qquad X = \begin{pmatrix} x_{1,1} & x_{1,2} & ... & x_{1,k} \\ x_{2,1} & x_{2,2} & ... & x_{2,k} \\ ... & & & \\ x_{n,1} & x_{n,2} & ... & x_{n,k} \end{pmatrix}, \qquad \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ ... \\ \beta_n \end{pmatrix} \tag{3.1}$$

Define cost function $J(\beta) = \frac{1}{2}(X\beta - Y)^T \cdot (X\beta - Y)$, and we can find the β which minimize this cost function by taking first derivative, i.e. $\frac{\partial}{\partial\beta} J(\beta) = 0$. With some algebra, $\beta = (X^T X)^{-1} X^T Y$.

3.3 Features Extraction

One big challenge of applying machine learning technologies on a particular problem is to gather useful features [38]. Features can be considered as certain representations of data, and they can be of any form imaginable. Take OCR as example, a feature could be intrinsic characteristic like the shape of characters, (e.g. 'A' contains sharp angle on top). On the other hand, a feature could also be extrinsic like the amount of space surrounding the character, (e.g. a large empty spaces on the left, which might indicates the beginning of a sentence). Often, figuring out good features requires prior knowledge on the application domains. For example in this case, the fact that we know a sentence is usually started with a capital letter inspires us to consider the amount of space as a feature.

Assuming we have a good set of features, then the next step is usually more straight forward, or more like trial-and-error. There are a variety of machine learning algorithms that we can plug in easily. However, in practice, we often need to go back and forth many times to refine the features. Although applying a well studied learning algorithm is straight forward, having good understanding on the mechanics and principles of the algorithms allows us to make more effective diagnosis.

3.4 Supervised Learning vs Unsupervised Learning

There are mainly two broad classes of machine learning algorithms, namely supervised learning and unsupervised learning. The major difference between the two is whether the data is labelled, meaning that the data is associated with a desired output value. One typical example of supervised learning is classification. Classification is the problem of identifying categories of an observation. Usually, in this problem settings, we are given a set of observations (training data), which we know their correct categories (labels). We train a model which can then be used to identify the categories of other unseen observations. In fact, classification is very similar to the regression problem mentioned in section 3.2 except for that fact that the output is categorical instead of continuous.

Unsupervised learning, on the other hand, does not require observations being labelled. One typical example of unsupervised learning is clustering. Clustering is the problem of grouping observations. We don't know the correct groups where they belong and most of the time, we don't even know how many groups there are. Usually, what we do is to measure the similarities between observations and put them into clusters (groups) accordingly. We extend the discussion of Classification and Clustering in the following sections because these two methods are used in our work.

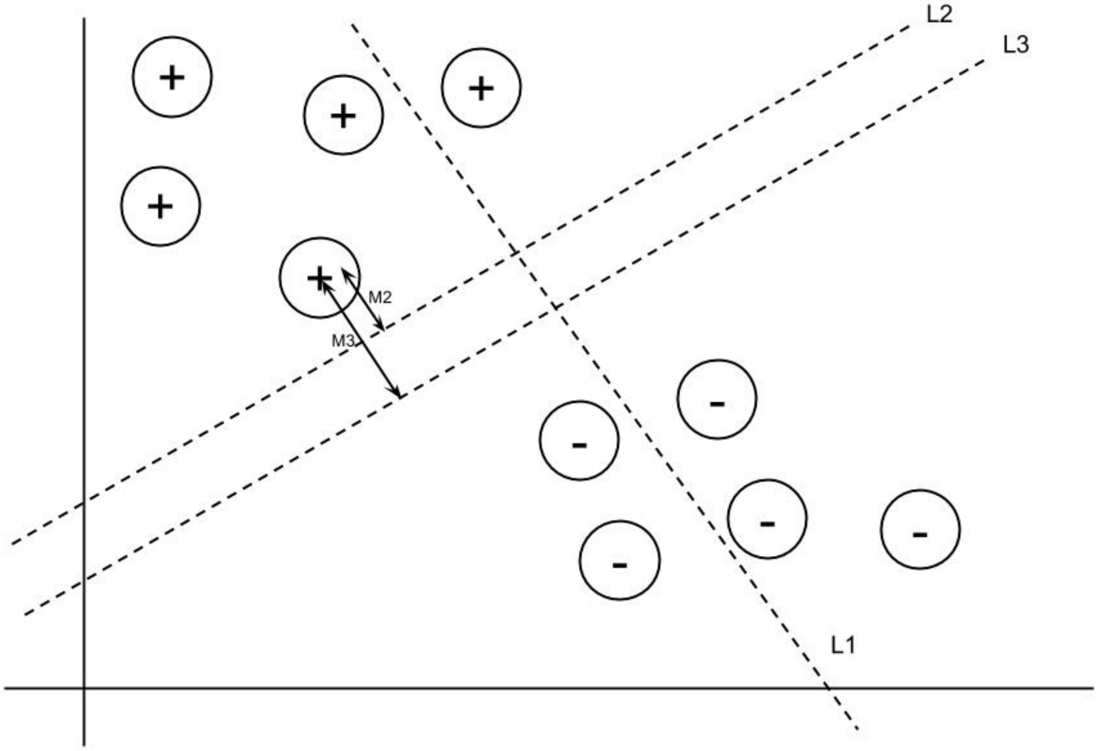
3.5 Supervised Learning: Classification

As mentioned, classification is the task of identifying categories of observations. In this section, we will look into two classification algorithms, namely Support Vector Machine (SVM) and Nearest Neighbours, which are based on two very different ideas. Classifiers are built and compared using both methods in Chapter 4.

3.5.1 Support Vector Machine

For simplicity, let's first consider cases where we only have two output categories (binary classification). The basic idea of an SVM is to draw a decision boundary to separate the two classes. Figure 3–1 illustrates the idea of drawing a decision boundary. Suppose your observations are two dimensional, each data point is labelled as either positive or negative, and the objective is to find a line that can separate the positives from the negatives. Obviously, line L1 is not a valid choice because it does not fulfill the requirement. To pick between L2 and L3, we need to introduce another concept called margin, which is defined as the smallest distance from any observations to the line. Intuitively, the larger the margin is, the more likely that a new observation (which we don't know the true category yet) will fall into the correct side. Therefore, we would like to find a line that is not only able to separate the positives and negatives perfectly, but at the same time maximizes the margin. In view of this, L3 would be a better separator than L2 because the margin M3 is larger than the margin M2. Once we have the decision boundary, the classifying step is pretty straight forward: if the new observation falls into the positive side of the boundary, then it is classified as positive, otherwise, it is negative.

Figure 3–1: SVM - Decision Boundary



Let us look at figure 3–2 for a formal definition of margin. Define w as the normal vector to the decision boundary, thus $\frac{w}{\|w\|}$ is the unit normal. w_0 is the constant, thus $y = wx + w_0$ is the line representing the decision boundary. A is the data point in consideration, and B is the corresponding point on the decision boundary that is nearest to A. γ represent the length of the distance between the data point to the decision boundary, and so the vector from B to A would be $\gamma\frac{w}{\|w\|}$. If we represent A as a vector x, then B would be $x - \gamma\frac{w}{\|w\|}$. Since B is on the decision boundary, so $w \cdot (x - \gamma\frac{w}{\|w\|}) + w_0 = 0$. By rearranging the equation, we solve γ as $\gamma = \frac{w}{\|w\|} \cdot x + \frac{w_0}{\|w\|}$

if data point is on the positive side of the decision boundary (direction of the line normal).

Now, we have N data points and we define γ_i as the distance between the i-th data point to the decision boundary. The margin of the decision boundary is then $\min_i \gamma_i$. The objective of SVM is to find the decision boundary which gives the largest margin, i.e.

$$\max_{w,w_0} \min_i \gamma_i$$

$$= \max_{w,w_0} \min_i y_i \cdot (\frac{w}{\|w\|} \cdot x + \frac{w_0}{\|w\|})$$

where $y_i = 1$ if the category of the data point is positive, and $y_i = -1$ otherwise.

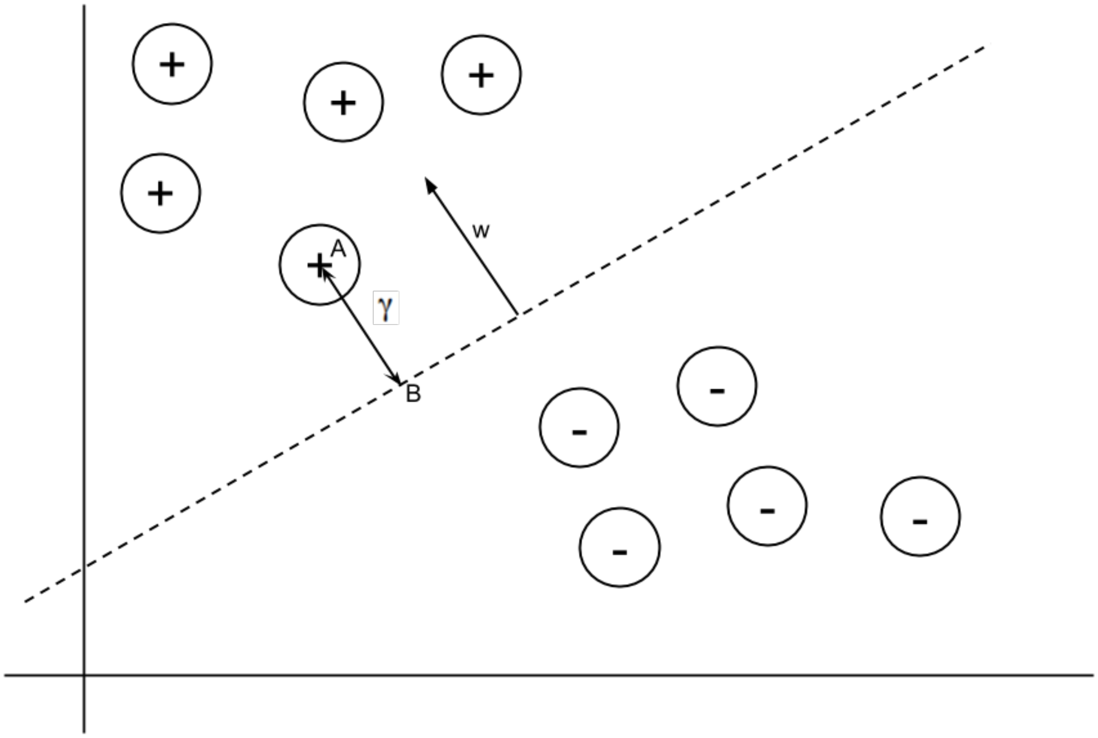
Solving for w and w_0 in the above formation turns out to be a constraint optimization problem. From the definition of margin, we have

$$M \leq y_i \cdot (\frac{w}{\|w\|} \cdot x + \frac{w_0}{\|w\|}) \quad \forall i$$

which suggests that we are maximizing M with respect to w and w_0 subject to $y_i \cdot (\frac{w}{\|w\|} \cdot x + \frac{w_0}{\|w\|}) \quad \forall i$. For further details on solving this optimization problem, we refer the reader to the reference [20].

In many cases, observations are not linearly separable, as shown in figure 3–3. To deal with this problem, we map the observations into a higher dimensional feature space, in which the observations are linearly separable with a higher dimensional

Figure 3–2: SVM - Margin



hyperplane. To illustrate the idea, consider an example of 1 dimension observation in figure 3–4. On the left hand side, you cannot draw any single straight line to separate the positive and negative points (You can draw a parabola though). However, we can map them into a 2 dimensional feature space, by introducing a second dimension $x_2 = (x_1 - C)^2$. In the new feature space, the observations are then linearly separable. This idea can be extended to higher dimensional observations.

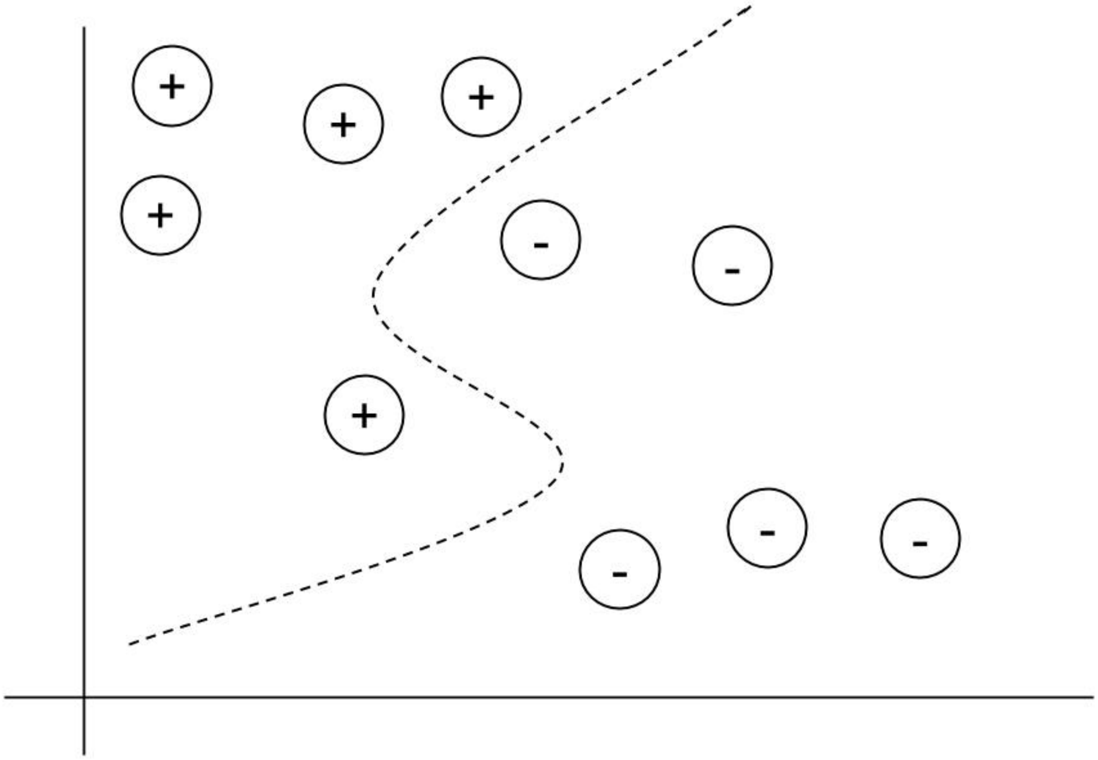
Normally, the computation of high dimensional features could be expensive. For example, if the observation space is an n-dimensional vector, i.e. $< x_1, x_2, ..., x_n >$, and to construct a feature vectors which consists of all second degree polynomial terms, i.e. $< {x_1}^2, {x_2}^2, ..., {x_n}^2, x_1x_2, x_1x_3, ..., x_{n-1}x_n >$, the asymptotic performance would be $O(N^2)$. Similarly, to construct a feature space of K-degree polynomial terms requires a computational cost of $O(N^K)$, which is not practically feasible most of the time. The beauty of SVM is that we are able to separate the observations in that high-dimensional (or even infinite dimensional) feature spaces without actually computing the features nor constructing the decision boundary, by making use of a mathematical trick called the kernel trick [20].

As you can see, a single decision boundary can only be drawn if there are only two classes. One common approach to handle multi-class classification problem is to reduce it into multiple binary classification problems. For example, we can build binary classifiers (called sub-classifiers) between each pair of classes. i.e. $\frac{N(N-1)}{2}$ classifiers for N classes. To classify a new observation, we first run it against all the sub-classifiers, and per each sub classifications, whichever class win get one vote. At the end, we take the class with the largest number of votes as the final output.

3.5.2 Nearest Neighbour

The Nearest Neighbour (NN) classifier makes use of a different idea from SVM. In NN, we need to define a distance metric between two data points, i.e. $D(x_i, x_j)$ Some examples of distance metric are Manhattan distance, i.e. $\sum\limits_k |x_{i,k} - x_{i,k}|$ [41] or Euclidean distance, i.e. $\sqrt{\sum\limits_k (x_{i,k} - x_{i,k})^2}$ [40] where $x_{i,k}$ is the k-th dimension of the data point i. With the distance metric defined, we can rearrange our training data in increasing order of their distances to the new data point x that we want to classify, i.e. $x_{(1)}, x_{(2)}, ..., x_{(N)}$ such that $D(x_{(1)}, x) \leq D(x_{(2)}, x) \leq ... \leq D(x_{(N)}, x)$. The nearest neighbour classifier is built upon this new ordering [39].

Figure 3–3: SVM - Non-linearly Separable



In the simplest version called the 1-nearest neighbour classifier, we simply assign the new data point the class of its closest training data point, i.e. $y_{(1)}$ Figure 3–5 illustrate the idea considering euclidean distance as distance metric. The grey point is the new observation that we want to identify, and points 1 to 3 are our training observations. We compute the euclidean distances D1, D2 and D3 from the new observation to each of the training observations, and pick the one with smallest distance, which is point 2. So we simply output the class of observation 2 as the predicted class of the new observation.

Figure 3–4: SVM - Mapping to Higher Dimension

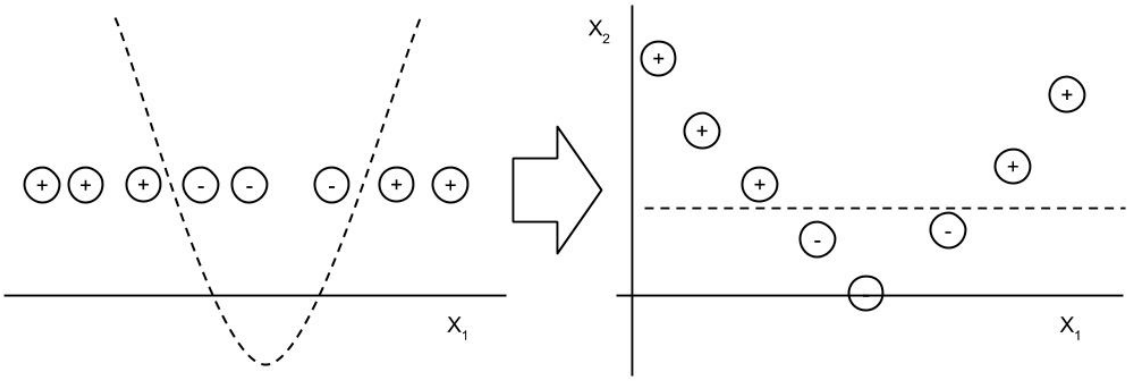
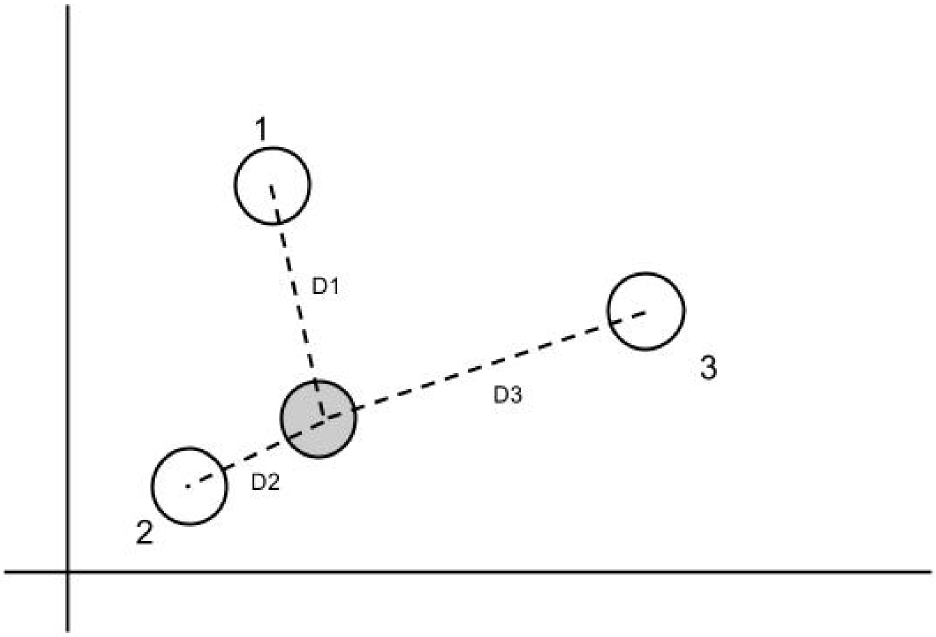


Figure 3–5: Nearest Neighbour Illustration



There are many types of variations of nearest neighbour classification, one of which is called the k-nearest neighbours. In k-nearest neighbours, instead of taking a single nearest neighbour $y_{(1)}$, we take K neighbours and then make a majority vote on the classes, i.e. the mode of set $(y_{(1)}, y_{(2)}, ..., y_{(K)})$.

3.5.3 Support Vector Machine vs Nearest Neighbour

There are some fundamental differences between the two classifiers. First, SVM can deal with very high dimensional, or even infinite dimensional data using kernel [20] whereas the computational cost of NN scales up proportionally with the dimensions. Second, NN is very intuitive and easy to implement while the mathematics behind SVM could be very involved. Third, SVM require a good choice of kernel while NN require a good choice of distance metric between data points. Finally, NN can handle multi-class classification naturally, whereas SVM can only handle it indirectly on top of the binary version.

3.6 Unsupervised Learning: Clustering

Clustering is the task of putting objects into groups (i.e. clusters), in which the objects within the same groups are more similar under some similarity measure. One popular notion of clusters is small distances within cluster members and large distances between different clusters. Clustering has been used widely in data mining and pattern recognition. Unlike classification problems, the observations usually do not have predefined classes and there are no absolute correct answers to the problems.

K-means Clustering is one instance of clustering algorithms with the aim to partition all the observations into K disjoint sets, so as to minimize the within-cluster sum of squares [42]. To put it mathematically, given a set of N observations, each of which is a vector of any dimension, the objective is to find the partitions $S = \{S_1, S_2, ..., S_k\}$ such that

$$\underset{S}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|\mathbf{x_j} - \mu_\mathbf{i}\|_2, where$$

$$\mu_i = \frac{1}{N} \sum_{x_j \in S_i} x_j$$

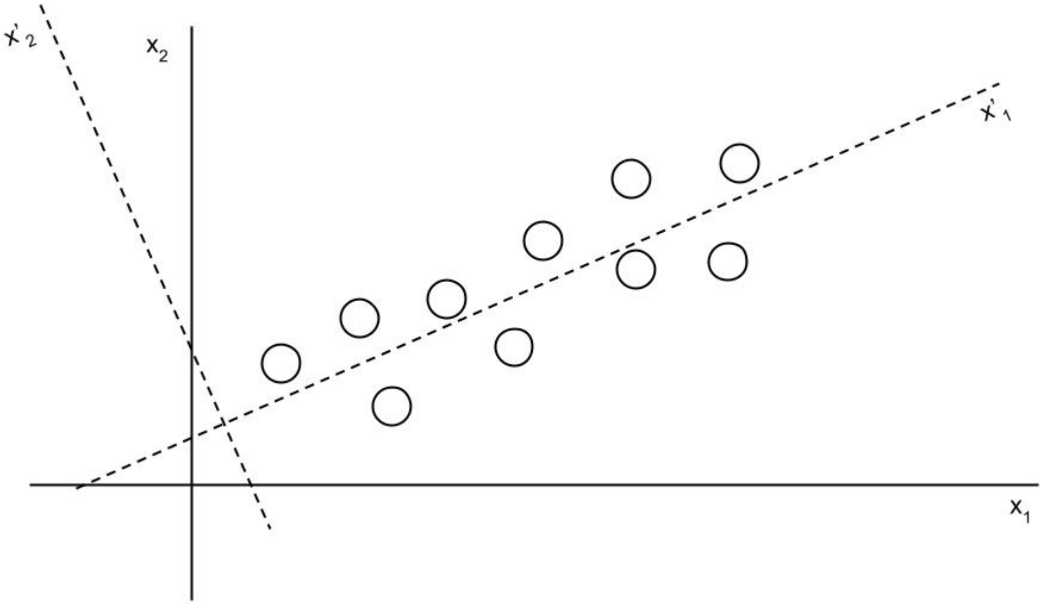
Solving this problem deterministically turns out to be NP-hard [43], however, in practice, there are efficient iterative algorithms that converge to local optimum which can achieve good solutions quickly [44].

3.7 Dimensionality Reduction: Principal Component Analysis

Dimensionality reduction is the technique of reducing the number of input features. Sometimes observations posse very high dimension, meaning that they have a lot of features, and a lot of time, dealing with high dimensional data is not good for the following reasons: First, it requires high computational costs and storage spaces. Second, some features are highly correlated, and they might all be influenced by a single underlying factor. Dimensionality reduction allows us to remove redundant features and recover underlying factors, which usually give better results because of the reduced noise.

Principal Component Analysis (PCA) [45] is a common dimensionality reduction algorithm in the machine learning community. The idea is to convert a set of correlated variables (the original features of the observations) into a set of linearly independent variables called principal components by doing a linear transformation. Other than being linearly independent to each other, the first principal component has to posses the highest possible variance, and after fixing the first component, the second principal component has be picked such that it has the highest possible variance, so and so. Figure 3–6 illustrates the idea of principal components. Consider the original observations having two dimensions, x_1 and x_2. We do a linear transform on the observations and represents them in terms of another two bases x'_1 and x'_2. Note that x'_1 gives us the highest possible variance among all the possible basis. x'_2 is linearly independent to x'_1 and obviously also posses the highest possible variance. Here, we define x'_1 and x'_2 as the principal components.

Figure 3–6: PCA - Principal Components



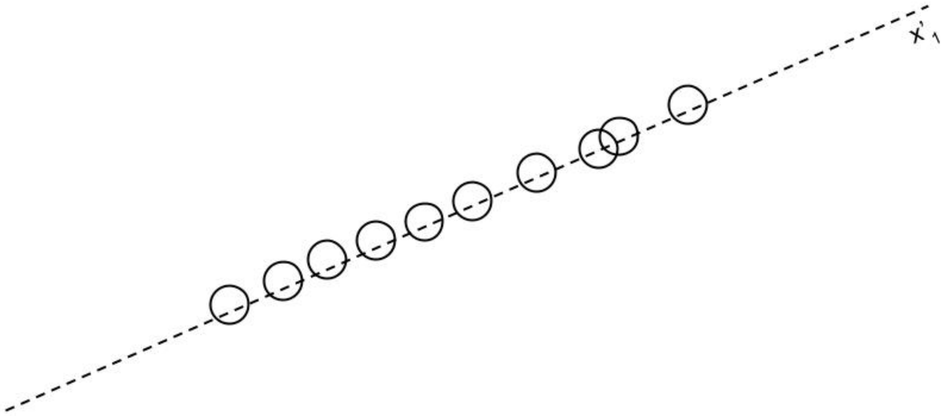
Algebraically, PCA can be solved by eigenvector decomposition [46]. First, we construct a matrix X in which each row corresponds to a data point, and each column corresponds to a particular features, i.e. $X_{i,j}$ indicates the j-th feature of the i-th data point. The goal of PCA is to find an orthonormal matrix P where $Y = PX$ such that the covariance matrix $\frac{1}{n-1}YY^T$ is diagonalized. The covariance matrix being diagonalized means that the components are linearly independent. By definition, P is our desired linear transformation matrix, and the rows of P are then our principal components.

It turns out that the rows of P are simply the eigenvectors of XX^T (detailed proof in reference [46]). Therefore, we can obtain the eigenvectors (principal components) by doing a singular value decomposition on XX^T, and sorting the components according to their respective eigenvalues (which are also the variances of X along the rows of P).

Continuing with our example, once we extract the principal components of the observations, we can do dimension reduction easily by simply dropping the least significant component(s), for example in this case, the x'_2 component. It means that the observations can now be represented as a single dimension in term of x'_1 , as shown in Figure 3–7. Intuitively, even though we only have one dimension left, it can still represent quite well the original observations. In fact, it is the best 1-dimensional representation we can have in term of capturing variances. Let's say instead of using the principal components, you simply drop the x_2 in the original observations. The x_1 component will give us less information than x'_1.

Usually the number of principal components to be dropped is decided by analysing the variances accounted for. Suppose the observations are in D dimension, and denote the variances of the observations in the i-th ($i \in D$) principal component be V_i. The total variances would be $\sum\limits_{i=1}^{D} V_i$, and the percentage of variances accounted for by the first k ($k \leq D$) components would be $\frac{\sum_{i=1}^{k} V_i}{\sum_{i=1}^{D} V_i}$. One common practice is to retain the first k principal components such that the percentage of variances accounted for reaches a certain threshold [45], like 90%. However, there is no absolute rule. Dimensionality reduction plays an crucial role in our work; we reduced our frequency

Figure 3–7: PCA - Dimension Reduction



domain features from 150 dimensions to around 10 dimensions, while retaining about 90% of variances. Without this, our experiments will take a lot more time to finish.

3.8 Topic Models

Topic Modeling is a type of statistical model learning for discovering hidden topic patterns from a set of observations. It was originally used in text pattern discovery, but later extended to other application domains [47]. Intuitively, given a text document, the observations are a sequence of words. We could imagine that the words are not chosen randomly, but more likely being chosen according to some underlying topics. For example, if the document is about "science", then we're more likely to see words like "experiments" and "discovery". The important concept here is that the topics are usually unobservable (hidden), and that's why we have to infer them from what we observe.

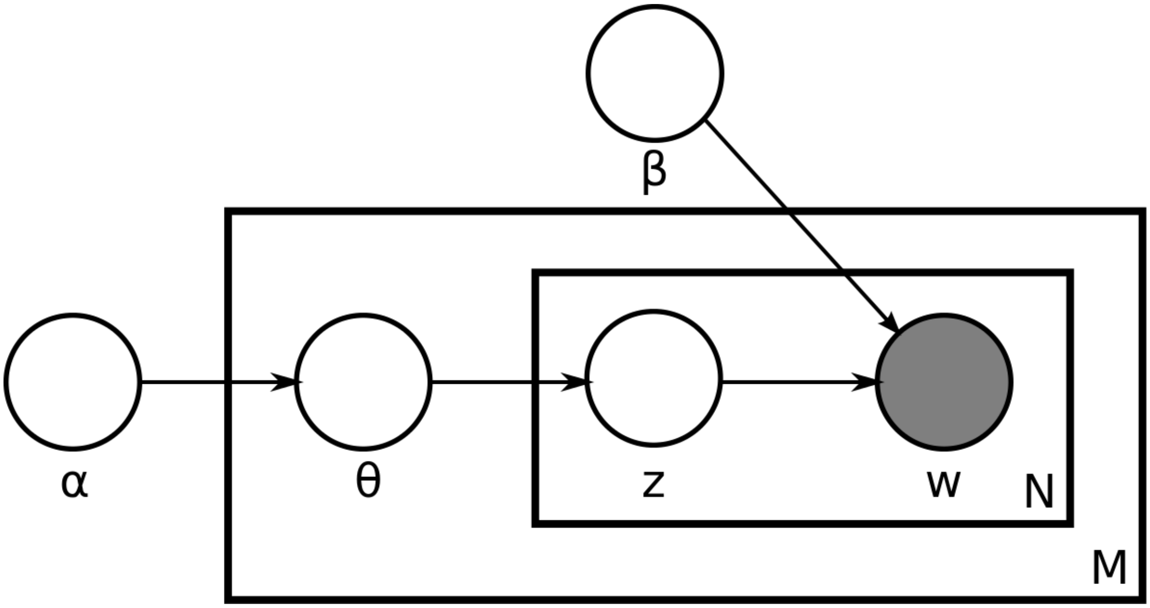
3.8.1 Bags-Of-Words Model

In reality, grammars and word orders play an important role in the context of the documents. However, for simplicity, a methodology called Bags-Of-Words is usually applied [47]. In Bags-of-Words model, we completely ignore the ordering of words and represent the document solely as frequencies of words from a dictionary. For example, a simple text document "Science is interesting. Everybody loves Science." is represented as {"Everybody": 1, "interesting": 1, "is": 1, "loves": 1, "Science": 2}. Note that the ordering in the dictionary is irrelevant.

3.8.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a more sophisticated instance of topic models [21]. The basic idea of LDA is that each document is represented as a mixture of latent topics, where each topic, on the other hand, is represented as a distribution over words. Figure 3–8 shows the plate notation (Plate notation is a method of representing variables that repeat in a graphical model) of LDA model. The outer box represents the duplicates of M documents, while the inner box represent the duplicates of N words in a document (Note that we assume all the documents have N words in the diagram for simplicity, but that is not a requirement). w is the only observable variable, and $w_{i,j}$ as the j-th word appeared in the i-th document. $z_{i,j}$ is the topic corresponding to $w_{i,j}$, i.e. j-th topic appeared in the i-th document.

Figure 3–8: Graphical Model representation of LDA [21]



Assuming there is a total of K topics and V words in the vocabularies, LDA assumes the following generative process for the documents

1. For each document i,
   (a) Choose $\theta_i \sim Dir(\alpha)$

   where $\theta_i \in \mathbb{R}^K$ and $Dir(\alpha)$ is the Dirichlet distribution for parameters α

   (b) For each word,

i. Choose a topic $z_{i,j} \sim Multinomial(\theta_i)$

where $Multinomial(\theta_i)$ is a multinomial distribution over parameter θ_i

ii. Choose a word $w_{i,j}$ probabilistically according to $P(w_{i,j}|z_{i,j}, \beta)$

where $P(w_{i,j}|z_{i,j}, \beta)$ is a multinomial probability conditioned on topic $z_{i,j}$ over parameter β

and $\beta \in \mathbb{R}^{KxV}$, in which $\beta_{i,j}$ indicates the probability of picking word i given that the topic is j

There are a couple of important properties of LDA. First, in LDA, each document is composed of multiple hidden topics, whereas in some other models, each document contains only a single topic. Second, LDA has a generative nature: each words of the document (observable variables) are modelled as if they are generated from an underlying topic (latent variables) in a probabilistic manner. In contrast, discriminative models do not care about how the observations are generated, but focus on the conditional probability of the latent topic given the observations, i.e. $P(topics|words)$. Third, LDA assume Dirichlet priors on the topic distribution of a documents.

The two parameters that we need to estimate in LDA is α and β. Basically, in the training phase, we wish to find $\alpha \in \mathbb{R}^K$ and $\beta \in \mathbb{R}^{K \times V}$ such that the likelihood of the our observations (i.e. $w_1, w_2, ..., w_M$) is maximized, i.e.

$$\underset{\alpha,\beta}{\operatorname{argmax}} \sum_{d=1}^{M} \log p(w_d|\alpha, \beta), \tag{3.2}$$

and this can be solved using a variational EM algorithm [21], or Gibbs Sampling [22].

As a by-product of the training, we will obtain the topic distribution of documents, i.e. $\theta_1, \theta_2, ..., \theta_M$. This topic distribution is the most important outcome for the purpose of our analyses in this thesis. In fact, we can also make inference on any unseen documents using α and β by computing the posterior distribution of the hidden variables, i.e.

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)}. \tag{3.3}$$

3.9 Training, Testing and Validation

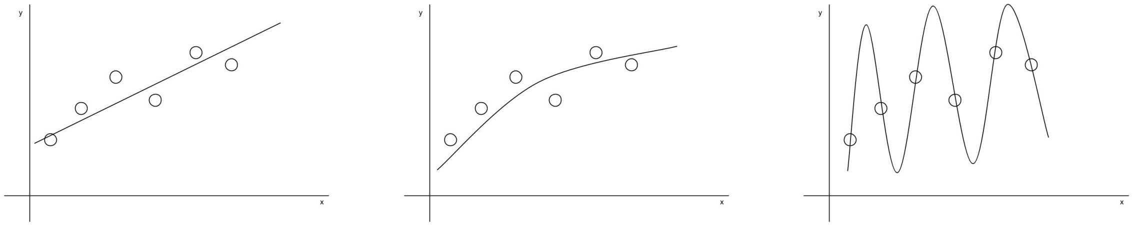
3.9.1 Testing Set

When we want to report how well the machine learning algorithms or models work on a particular dataset, one common approach is to split the dataset into two disjoint subsets, called training set and testing set. We construct and train our models and algorithms only with the training set, and after that we report the results on the testing set. The reason for leaving an untouched testing set is because you want to measure how your learned model is expected to generalize to unseen data.

3.9.2 Overfitting

Overfitting refers to the scenario that you are tuning the models too much in order to make it work well for the training data, with the consequence that it does not generalize to unseen data. Figure 3–9 illustrates the idea of overfitting. Suppose we are given a set of observations with a single independent variable x and a dependent variable y (i.e. a Regression problem). The objective is to make a good hypothesis on the relationship between x and y in order to make good prediction on y given a new observation x in the future. One way to do it is to fit a line to minimize the error in the training data. The three graphs from figure 3–9 from left to right show i) a straight line, ii) a quadratic curve, and iii) a 5-degree polynomial curve respectively. The training error improve from left to right, and in fact the curve fit perfectly (zero error) with the 5-degree polynomial curve. However, the problem is that the 5-degree polynomial curve, although working perfectly with the training data, is unlikely to make a good prediction on a new observation. In fact, the polynomial curve is highly likely to work better and so if our hypothesis goes beyond certain degree (too complicated), we suffer from overfitting.

Figure 3–9: Overfitting Illustration



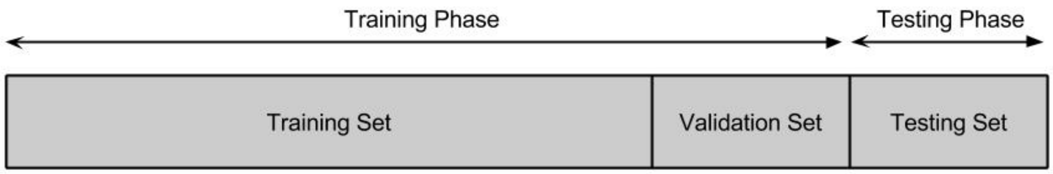
Left: Data fitted with linear function; Middle: Data fitted with quadratic function; Right: Data fitted with 5-degree polynomial function.

3.9.3 Validation Set

To fix the problem of overfitting, sometimes we leave out an additional portion of the training data as validation set, which is used to find the correct hypothesis as well as controlling the models' complexity. This left-out portion of data is called Validation Set. Continuing with the above example, imagine now you have one extra observation and you want to validate your hypothesis with this new observation. You will realize that your 5-degree polynomial curve does a very bad job. Indeed, you can find the best hypothesis that gives the most accurate prediction on this validation data point. Intuitively, if the hypothesis works well on the left-out validation data, it is more likely that it will also work well on the testing set because they are both unseen.

Note the difference between testing set and validation set: the testing set is never touched during the training process, and it is used purely for reporting results at the end, whereas the validation set is used during training for the purpose of getting a good model that generalizes well for unseen data. The splitting of dataset can be summarized in figure 3–10.

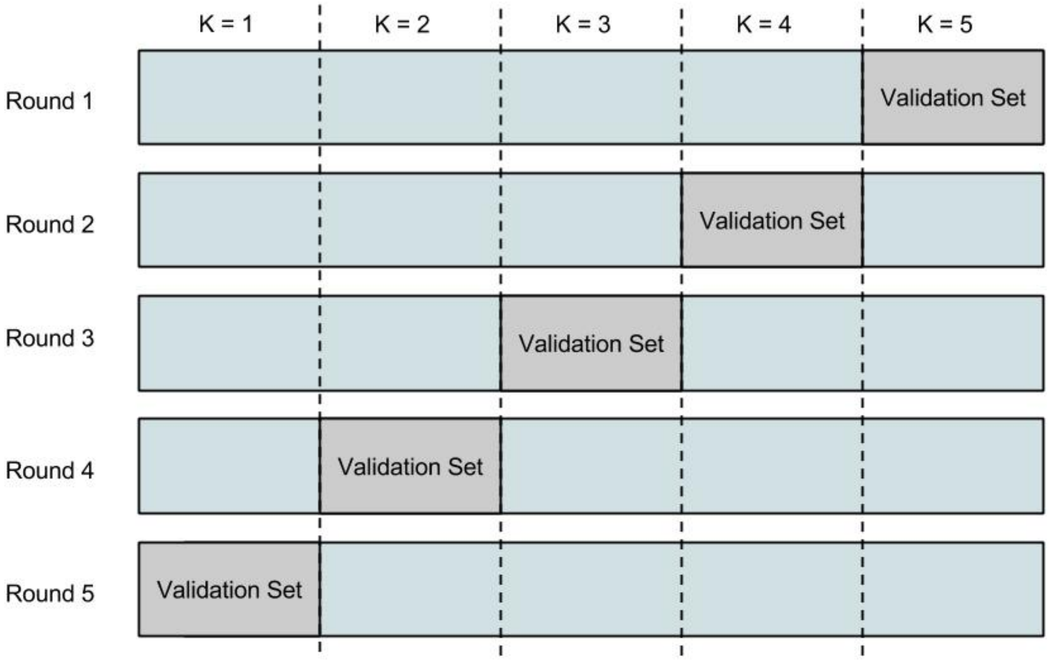
Figure 3–10: Dataset Composition



3.9.4 Cross Validation

One problem with the setting in figure 3–10 is that we didn't fully utilize the amount of data we have. Let's say if we set aside 20% of the data in the validation set, we are losing 20% amount of data for fitting the models' parameters. Therefore, a methodology called Cross Validation is usually applied [48]. In a K-fold cross validation, we split the training data into K disjoint subsets. We train our model for K times, each time using one of the subset as validation data and all other subsets as training. After that, we take the average of the K runs. Figure 3–11 illustrates the idea of a 5-fold cross validations: We split the whole training data into 5 subsets, and in each round, we leave out one subset as validation. Due to the limited amount of data in our work, cross validations have been used a lot to tune the parameters of our model.

Figure 3–11: Cross Validation Illustration



3.10 Signal Processing

Signal Processing is the area of Electrical Engineering that deals with the analysis of analog and digitized signals [60]. Some common signals include sounds, electromagnetic radiations and sensor readings. The particular signals of interest for the purpose of this thesis, is accelerometer sensor readings.

Signals usually appear as a sequence of quantitative measurements over time, therefore one mathematical way to represent signal is to consider it as a function across time, i.e. $s(t)$. Sometimes, this time-domain function doesn't reveal much useful information, or at least it's hard to interpret. For one thing, the readings in consecutive time steps posse temporal dependency, and because of this correlation, we cannot treat each data point independently and apply machine learning algorithms directly. Often, we transform the raw signals into higher level features, and Fourier Analysis is one of commonly used method.

3.10.1 Fourier Analysis

Fourier analysis of a time series is a process of decomposing a periodic function into a set of sinusoidal components, i.e. $s(t) = \sum_{n=1}^{N} a_n \sin(2\pi n f_o t + \phi_n)$ [49]. Each sinusoidal components have different period, i.e. $2\pi n f_o t$, phase shift ϕ_n, and amplitude a_n.

Sometimes, we refer to this set of sinusoidal components as fourier series. Intuitively, the way to look at the fourier transform is that we are converting the original signal from time domain to frequency domain because each sin curve simply represents a particular frequency. The coefficients $a_1, a_2, ..., a_n$, on the other hand, correspond to the magnitude of those frequencies.

3.10.2 Discrete Fourier Transform and Fast Fourier Transform

Discrete Fourier Transform (DFT) is a mathematical technique used to convert a series of equally spaced samples into a list of coefficients of sinusoids [23]. Obviously, the time series sensor data fall into the category of equally spaced samples, and therefore we can use DFT to compute the frequency coefficients from the raw signals. Naive approach to compute DFT takes an asymptotic performance of $O(N^2)$. Faster Fourier Transform (FFT) refer to the algorithms (there are more than one) that compute DFT efficiently, usually with an asymptotic performance of $O(N \log N)$ [24]. We will not go into the details of the actual implementation of FFT, so for the rest of the thesis, we will simply refer to the computed frequency coefficients as

‘FFT coefficients’. In fact, the FFT coefficients are the main extracted features we used in our studies.

Chapter 4
Methodology for characterizing smart wheelchairs' activities

The main contribution of this thesis is an end-to-end pipeline for wheelchairs activities recognition. To define activities recognition more specifically, we have two main goals: First, we would like to train a classifier from previously seen and labelled activities which can then be used to categorize future instances of these activities. Second we would like to explore the possibility of using topic modeling to infer hidden activity patterns in a totally unsupervised manner. Figure 4–1 shows our multi-layered model. We shall explain each components in this chapter.

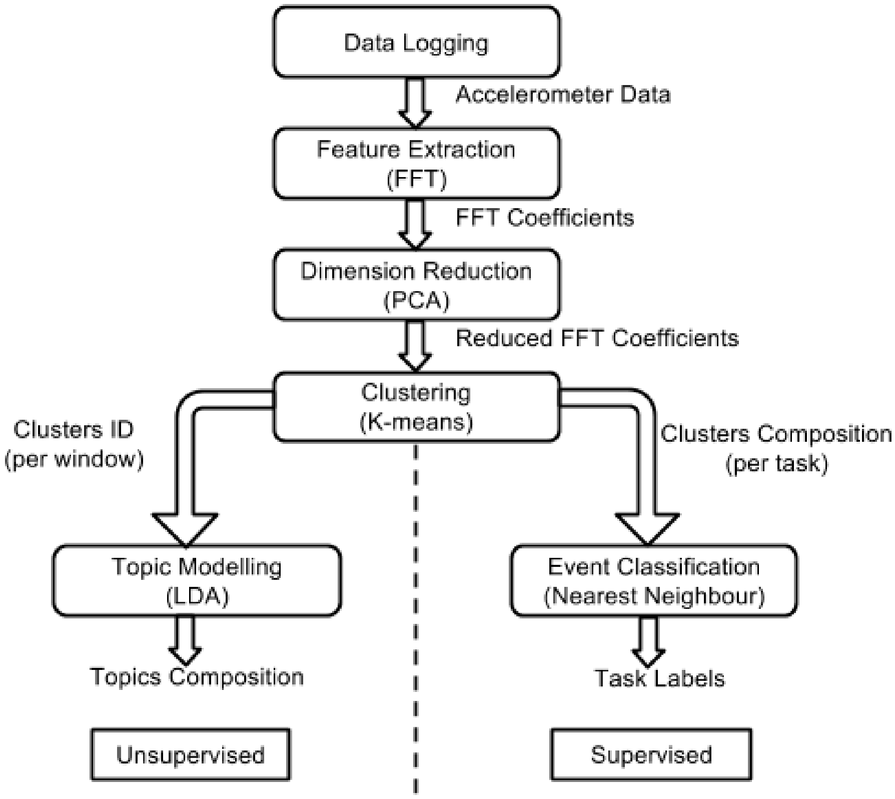
4.1 Data Logging

For the purposes of this study, a data-logging platform, called the Wireless Inertial Measurement Unit with GPS (WIMU-GPS) (Figure 4–2), was developed and installed on the powered wheelchair of the users. We record 3D accelerometer data, which captures the acceleration magnitude in x, y and z directions at a rate of 250 Hz. Figure 4–3 shows a sample accelerometer signals in 3 directions captured from our sensor device.

4.2 Data Collection

In a clinical setting, under the monitoring of therapists, 7 real powered wheelchair users (which we will refer to as participants for the rest of the text) were asked to perform a list of driving tasks, extracted from the Wheelchair Skills Test (WST) [32], using their own powered wheelchairs. The WST provides a training and testing

Figure 4–1: End-to-End pipeline overview



protocol developed to help clinicians assess and train wheelchair users. As such, it represents a rich and diverse set of wheelchair driving activities characteristic of everyday use.

Table 4–1 summarizes the types of tasks, together with the number of trials, carried out by each participant. There are a total of 743 trials with 29 different types, and the average duration of a single trial is around 18 seconds.

Two different sets of experiments are presented. In the first set of experiments, we treat each participant individually, and build a personalized event classifier. In this case, for each participant, we use the first trial of each type of tasks as testing

Figure 4–2: Overview of datalogging platform (from [25])

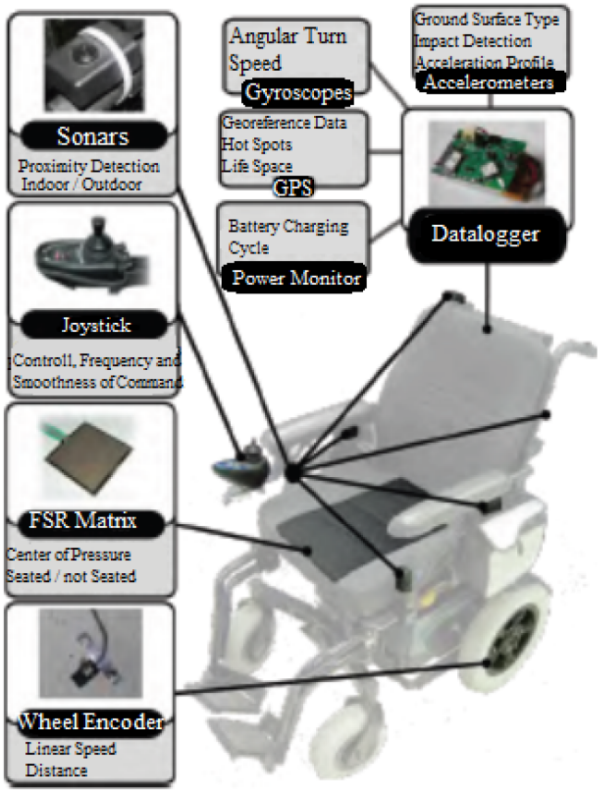
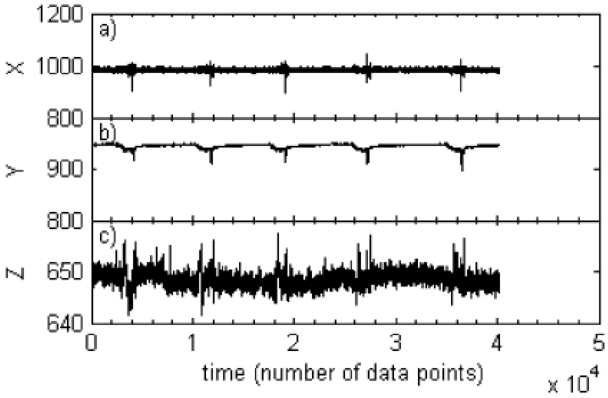


Figure 4–3: Sample 3D accerlerometer signals



data, and use the remainder as training data. This case is denoted ‘Individual-Set’ in results below. Overall classification performance are calculated by taking an average over the accuracy of the personalized classifiers. In the second set of experiments, we build a single classifier over all subjects and evaluate its ability to generalize to new subjects. As such, we use the first participant’s performance as testing data, and use the performance of the other 6 as training data. Since there were no trials conducted by participant 1 on tasks T10, T26 and T27, we will drop all the trials of these three tasks from other participants as well. We use the term ‘Group-Set’ to refer to this classifier in the rest of the thesis.

There are two major differences between the two sets of experiments. First, we obviously have more samples in ‘Group-Set’. Second, samples from the ‘Group-Set’ will have higher variances because they are coming from different participants. We can imagine that over multiple trials of the same task, the variations coming from different people would be greater than the variations coming from the same person. It is also worth emphasizing that in ‘Group-Set’, we are trying to test on an unseen participant.

4.3 Feature Extraction and Dimension Reduction

To convert the recorded time-series data into a discrete set of feature vectors, we split the data stream into regular intervals, called windows, and extract representative properties from each window. A sliding window (i.e. having overlap between windows) of size ranging from one to a few seconds has been shown to produce good results in activity recognition [26, 27, 28, 29]. Previous work [26] also considered a detailed comparison of the classification performance on wheelchair activities using four different properties of time series, namely time-domain features, frequency-domain features, wavelet transform features and time-delay embedded features. Among these, frequency-domain features had the strongest predictive performance. Therefore in this work, we consider only frequency-domain features, using a window size of 2 seconds, with 0.2 seconds sliding overlap.

For each window, we apply a Fast Fourier Transform on each acceleration direction and extract the amplitudes of frequencies ranging from 1 to 50 (we drop frequencies greater than 50 because the signal strength of those are comparatively weak). Altogether in 3 directions, we obtain 150 features, i.e. $\{F_1^x, F_2^x, ..., F_{50}^x, F_1^y, F_2^y, ..., F_{50}^y, F_1^z, F_2^z, ..., F_{50}^z\}$. We then apply Principal Component Analysis to reduce the number of features to a small dimension, d, i.e. $\{F_1, F_2, ..., F_d\}$.

As a minor point, for the purposes of testing our approach for event classification, we eventually divide the recorded data into separate training and testing sets. The best PCA transform is selected using only the training data. We can then apply the same transformation matrix on the testing data.

4.4 Clustering

At this stage, the output of the feature extraction and dimension reduction could be used directly for output classification, as is common in the machine learning literature. However a significant limitation of this approach is that the classification step (especially the training phase) can be computationally expensive because of the large amount of windows. To overcome this, we further reduce the data using clustering methods to find representative samples of the training data.

We apply K-means clustering on all the windows. As a result of this procedure, each window is assigned a cluster ID. The cluster IDs can then be used directly as input features in Topic Modelling (unsupervised branch of the pipeline in Fig 4–1). Alternatively, for the purposes of event classification (supervised branch of the pipeline), we can also compute a cluster composition for each task. Define N_i as the number of windows for task i, and $w_{i,j}$ as the $j-th$ window of task i and $c_{i,j} \in \{1, 2, ...K\}$ as the assigned cluster of $w_{i,j}$ after clustering. Cluster composition of task i, i.e. $\mathbf{CC_i}$, is then defined as a vector, in which each element corresponds to the percentage of which a particular cluster appeared in task i:

$$\mathbf{CC_i} =< CC_i^1, CC_i^2, ..., CC_i^K >, \text{ where } CC_i^k = \sum_{j=1}^{N_i} I\{c_{i,j} = k\}/N_i, \text{ and}$$

$$I\{eq\} = \begin{cases} 1 & \text{if } eq \text{ is true} \\ 0 & \text{if } eq \text{ is false} \end{cases}$$

The cluster composition vector can be used directly as an input to the event classification module. In this case, each sample corresponds to a task, in contrast to the unsupervised case where each sample corresponds to a window with an associated cluster ID.

Similar to PCA, the K-means clustering selects the K centroids using only the training data. Cluster membership of the datapoints in the testing data is assessed using the clusters selected with the training data.

4.5 Parameter Fitting

There are a few parameters to select for the proposed method, in particular the dimensionality of the PCA projection (d) and the number of clusters (K) for

K-means clustering. For the PCA projection, preliminary results show that the 10 most significant components are sufficient to account for over 98% of the variance.

A common way to determine the number of clusters is by analyzing the intra-cluster variances and inter-cluster variances [30]. However, in our case, optimizing cluster quality does not necessarily align with the goal of optimizing classification performance. Instead, we performed a grid search over cluster sizes from 10 to 100, at an interval of 10 using cross-validations within the training set to select the best number of clusters. In general, we found that the performance usually levels off at around 30 to 50 clusters. More clusters sometimes lead to slightly better results, but not significantly. We set the final number to $K = 40$.

4.6 Event Classification

The purpose of the event classification module is to take the cluster composition vector and using supervised learning methods to produce an output corresponding to an activity label.

We considered a variety of methods, including Support Vector Machine and Nearest Neighbour classifiers. Preliminary investigation found that Nearest Neighbour worked faster and achieved better (or equally good) performance. This is consistent with related work on activity recognition from time series data [31]. So all the results reported below use this approach. In short, given a training set D, denote $y_{\mathbf{z}}$ as the label of training sample $\mathbf{z} \in \mathbb{R}^N$, the predicted label $\hat{y}$ on testing sample $\mathbf{x} \in \mathbb{R}^N$ would be:

$$\hat{y}(\mathbf{x}) = y_{\mathbf{z}*} \text{ where } \mathbf{z}* = \underset{\mathbf{z} \in D}{\operatorname{argmin}} \|\mathbf{x} - \mathbf{z}\|_2, \tag{4.1}$$

where $\|\mathbf{x} - \mathbf{z}\|_2$ is the euclidean distance between cluster composition vectors.

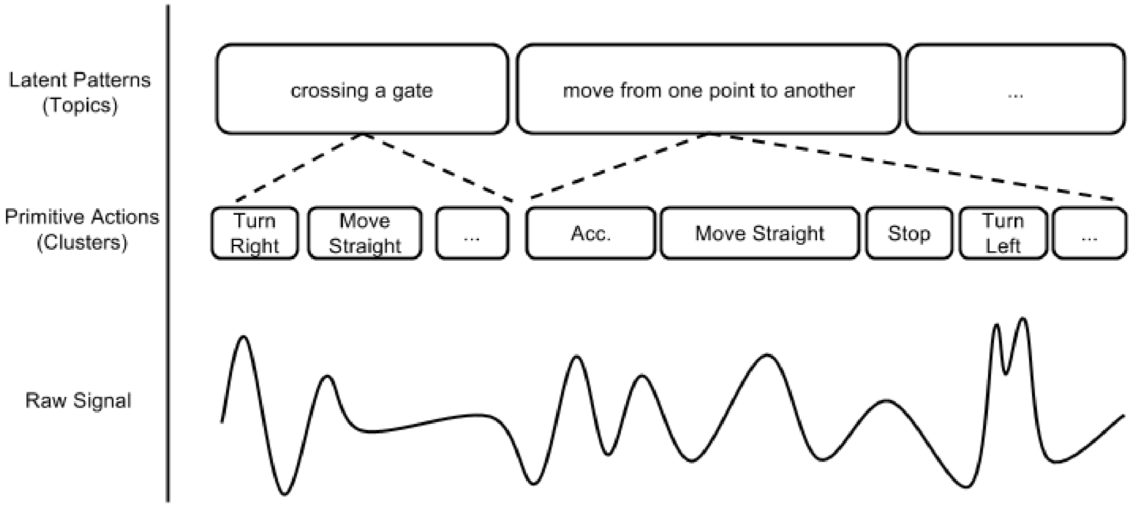
4.7 Topic Modeling

Given enough labelled training samples, event classification can effectively recognize activities with reasonable accuracies. However, in real life conditions, this poses significant limitations in terms of (1) dealing with scarcity of labeled data, (2) handling activities that change over time, and (3) discovering new activities. As such we propose to use methods from topic modeling to characterize activities on the smart wheelchair using only unlabeled sensor data.

The basic hypothesis of this approach is that activity patterns should possess a hierarchical structure, as illustrated in Figure 4–4. The lowest level contains the raw input signal, in our case 3D accelerometer data. On top of this are some primitive action patterns that generate the underlying signals. Primitive actions, as we define them, would be short lasting, roughly 2-3 seconds. These are exactly what we try to capture with the clustering step. Ideally, each cluster would correspond to one type of primitive action. Moving up the hierarchy, these lower level primitive actions are assumed to be generated during the course of some higher level activity patterns, which are unlabeled. The goal of this section is to propose the use of topic modeling methods to infer the high level activity patterns from unabelled data.

To learn the latent semantic, we use the probabilistic topic model, Latent Dirichlet Allocation (LDA). To apply LDA to learn the hidden structure, we first have to

Figure 4–4: Activity pattern hierarchy



define what is a document and what a document constitutes in the context of smart wheelchair activities. Our approach is to pull together a fixed number of consecutive windows and consider them as a single document. If we define the document length as L, then the number of documents we get from task i would be $\lfloor N_i/L \rfloor$. By then putting all the documents from all tasks together, we have a total of M documents, where

$$M = \sum_i \lfloor N_i/L \rfloor.$$

Extending our previous notation, if we define $w'_{i,j}$ as the $j-th$ window of document i, and $c'_{i,j}$ as the assigned cluster of window $w'_{i,j}$, the word vector of document i is defined as

$$\mathbf{d_i} =< c'_{i,1}, c'_{i,2}, ..., c'_{i,L} > .$$

Now that we have the representation of documents, another parameter we need to fix in LDA is the number of topics T in our model. One commonly used metric to evaluate LDA is perplexity [21]. The idea is to set aside some testing data, and infer their likelihood using the trained model. We did cross validation on the training data and found that perplexities stabilized at around 15 topics in most of our experimental settings, and so used 15 topics for the rest of the experiments. In general, there may not be a 'correct' number of topics; different numbers of topics can potentially model different complexities of activity patterns.

One output of LDA that we are interested in is the probability distribution of the T topics for each of the M documents denoted by $\theta \in \mathbb{R}^{M\text{x}T}$, where $\theta_{i,j}$ is the probability that a given word in document i is generated from topic j.

Table 4–1: Dataset Summary

Task Code	Description	Average Duration (secs)	P1	P2	P3	P4	P5	P6	P7	Total
T1	Rolls forward 10m	16.99	5	5	5	5	5	5	5	35
T2	Rolls backward 5m	22.39	5	5	5	5	5	5	5	35
T3	Descends 5deg incline	17.71	5	5	5	5	5	5	5	35
T4	Descends 5deg incline	18.46	5	5	5	5	5	5	5	35
T5	Ascends 5cm level change	10.80	5	0	6	5	8	4	5	33
T6	Gets over 15cm pot-hole	9.88	2	0	5	3	0	3	3	16
T7	Descends 5cm level change	7.55	4	0	5	4	5	4	6	28
T8	Gets through hinged door in push direction	49.78	2	1	1	2	1	1	1	9
T9	Gets through hinged door in pull direction	27.17	2	1	1	2	1	1	1	9
T10	Gets over 2cm threshold	12.25	0	4	5	6	0	5	0	20
T11	Rolls 2m on soft surface	12.27	3	2	3	3	0	2	2	15
T12	Turns 90deg left while moving forward	13.77	5	5	5	5	5	5	5	35
T13	Turns 90deg left while moving backward	22.10	5	5	5	5	5	5	5	35
T14	Turns 90deg right while moving forward	12.67	5	5	5	5	3	5	5	33
T15	Turns 90deg right while moving backward	17.24	5	5	5	5	3	5	5	33
T16	Turns 180deg in place clockwise	8.39	4	5	3	5	0	5	0	22
T17	Turns 180deg in place counterclockwise	8.55	5	5	3	5	0	5	10	33
T18	Maneuvers sideways right	33.55	3	3	3	3	3	3	3	21
T19	Maneuvers sideways left	36.74	2	3	3	3	3	3	4	21
T20	Frontal collision	10.91	5	5	5	5	5	5	5	35
T21	Lateral collision right	13.03	5	5	5	5	5	5	5	35
T22	Lateral collision left	10.60	5	5	5	5	5	5	5	35
T23	Collision on moving object	8.59	5	5	5	5	8	5	5	38
T24	Avoids moving object - left	13.68	5	5	5	5	0	5	5	30
T25	Avoids moving object - right	13.36	5	4	5	5	3	5	5	32
T26	Rolls 2m across 5deg side-slop (right-side down)	10.20	0	5	5	0	0	0	0	10
T27	Rolls 2m across 5deg side-slop (left-side down)	10.70	0	5	5	0	0	0	0	10
T28	Rolls 100m to local gym	66.75	1	0	1	2	0	0	0	4
T29	Gets through swing door	14.68	0	5	6	0	0	0	0	11
Total			103	108	125	113	83	106	105	743

[a] P1 to P7 indicate the participants, numbered from 1 to 7.

[b] The numbers under columns P1 to P7 represent the number of task trials of that participant.

Chapter 5
EXPERIMENTS AND RESULTS

This chapter presents the experimental results using the methodologies outlined in previous sections. We present the classification results on the right hand side of the pipeline 4–1, followed by the pattern discovery results on the left hand side of the pipeline. We end the chapter with some discussions about the results.

5.1 Event Classification

5.1.1 Results

We compare classification accuracy for both the Individual-Set and Group-Set setting in Table 5–1. Note that in addition to the method advocated above, we also present results for the case where we use the reduced FFT output directly as a feature, rather than the output of the clustering step. The overall classification accuracies are slightly less than 50%, with similar accuracy for both feature types in the Individual-Set task, but better accuracy using the cluster output in the Group-Set task. There are two important aspects to observe here. First, cluster composition seems to be more robust to variances, as is the characteristic of the 'Group-Set' data. Second, classification on cluster composition is much cheaper in term of computational cost, and thus useful in real-time operation. Overall, given that classification was performed using only 600 training samples on more than 25 different classes, in which some of them, like 'T28 - Rolls 100m to local gym' are highly complicated, we would argue that the results show significant ability to disentangle complex data from natural sensor data.

Looking at Table 5–2, we can directly observe the confusion matrix, showing which activities are being mis-classified, and as what other activity. We observe that many tasks, like ‘T20, 21 and 22 - frontal and left/ right lateral collisions’, are very similar in nature and the confusion matrix shows that they are mostly confused with one another. Although the overall accuracy is slight below 50%, we can see that most big numbers fall into the diagonal instead of scattering around. Indeed, the accuracy would improve significantly if we were to group similar items together; this is one of the motivations for investigating unsupervised pattern discovery methods.

Table 5–1: Classification Accuracies

Experimental Sets	per-window FFT	per-task Cluster Composition
Individual-Set	49.37%	48.28%
Group-Set	35.92%	46.60%

5.2 Pattern Discovery via Topic Modeling

It is notoriously difficult to evaluate the performance of pattern discovery methods, thus we use a mix of results, including qualitative inspections of topic compositions, and some more quantitative measures in our evaluation.

5.2.1 Topic Composition of Documents

As mentioned above, one output of LDA is the probability distribution of topics for each document. We define the topic composition of document i as

$$TC_i =< \theta_{i,1}, \theta_{i,2}, ..., \theta_{i,T} >$$

which is essentially the same as the probability distribution of topics of document i. Intuitively, we consider the composition as an expected realization of the probability

Table 5–2: Confusion Matrix on Group-Set

	Predicted Tasks																							
Task	T1	T2	T3	T4	T5	T6	T7	T8	T9	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T28
T1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T2	0	0.4	0	0	0	0	0	0	0	0	0	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0
T3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T4	0	0	0.2	0.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T5	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0.4
T6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T8	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T9	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0
T11	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T12	0	0	0	0	0	0	0	0	0	0	0.6	0.2	0	0	0	0.2	0	0	0	0	0	0	0	0	0
T13	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0.8	0	0	0	0	0	0	0	0	0	0	0
T14	0	0	0	0	0	0	0	0	0	0	0.2	0	0.6	0	0	0	0	0	0	0.2	0	0	0	0	0
T15	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0.6	0	0	0	0.2	0	0	0	0	0	0	0
T16	0	0	0	0	0	0	0	0	0	0	0	0	0	0.25	0.25	0.25	0	0	0	0.25	0	0	0	0	0
T17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.8	0.2	0	0	0	0	0	0	0	0	0
T18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
T19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0	0	0	0	0	0	0
T20	0	0	0	0	0	0	0	0	0.2	0	0.2	0	0	0	0	0	0	0	0.2	0.2	0	0.2	0	0	0
T21	0	0	0	0	0	0	0	0	0	0	0.2	0	0.4	0	0	0	0	0	0	0.2	0.2	0	0	0	0
T22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0	0.6	0.2	0	0	0
T23	0	0	0	0	0	0	0	0	0	0	0.4	0	0	0	0	0	0	0	0	0	0	0.2	0.4	0	0
T24	0	0	0	0	0	0	0	0	0	0	0.2	0	0.2	0	0	0	0	0	0	0	0	0.4	0.2	0	0
T25	0	0	0	0	0	0	0	0	0	0	0.2	0	0	0	0	0	0	0	0	0	0	0.4	0.4	0	0
T28	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

distribution. For a given document i of length L, the expected number of words generated from topic t, is simply $\theta_{i,t} \cdot L$. Normalizing it with the total number of words coming from all topics, i.e. $\sum_{t=1}^{T} \theta_{i,t} \cdot L = L$, gives exactly $\theta_{i,t}$.

For demonstration purposes, we show the results for the first participant (similar results are observed for other participants), and three regions are selected to illustrate some observations, as shown in Figures 5–1 to 5–3 (The full graph can be found in Appendix). Each column represents one document, which is composed of multiple vertical bars, which sum to 1. Each of the 15 bars represents the composition of a particular topic. For comparison, we also computed the cluster composition (40 clusters) of each document, and plotted them in Figures 5–4 to 5–6. The three selected regions are i) 'T1: Rolls forward 10m', ii) 'T8-T9: Gets through hinged door' and iii) 'T28: Rolls 100m to local gym'. Tasks in each region possess a certain kind of characteristic: Region i) contains the most clearly defined activities whereas Region ii) contains the most chaotic ones. If you refer to Table 5–2, they correspond to the parts where classification accuracies are 100% and 0% respectively. Region iii), on the other hand, is a good example to demonstrate complex activities. Complex activity is defined as an activity that is composed of numerous inter-related subroutines. As we can imagine, 'Rolls 100m to local gym' potentially involves numerous sub-activities, in which 'Rolls forward' features prominently.

We begin with a few qualitative observations. First, it is readily seen that cluster composition is much more noisy than topic composition, especially for Region ii). Most documents contain more than 4 or 5 major clusters, whereas in the topic composition, documents are mostly dominated by 1 to 2 major topics. Taking a closer look at Region iii) of topic composition, we are able to tell a brief story of what happened during the 'Rolls 100m to local gym' period. The dominant topics in the first 5 documents correspond to the dominant topics in the region of 'T24, T25: Avoids moving objects' (which are not shown in the figure). We then have 2 to 3 not-so-obvious documents, followed by two documents showing backward-moving patterns, which correspond to the dominant topic in the region of 'T2: Rolls Backward 5m' (which is also not shown in the figure). After another 3 to 4 not-so-obvious documents, the activity ends with 2 forward-moving patterns, which correspond to the dominant topic in the region of 'T1: Rolls forward 10m' (same dominated color as the first region).

5.2.2 Task Composition of Topics

Another interesting aspect to consider is what constitutes a topic, in terms of the underlying labels. Ideally, if we have perfectly labelled windows, we could find the composition of primitive actions for each topic. That might not be practical

Figure 5–1: Topic compositions of documents - Moving Forward

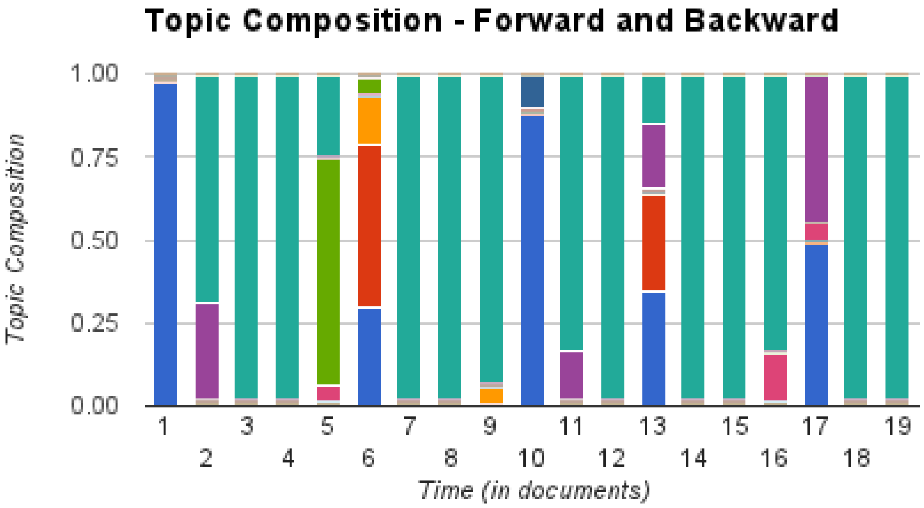


Figure 5–2: Topic compositions of documents - Crossing Gates

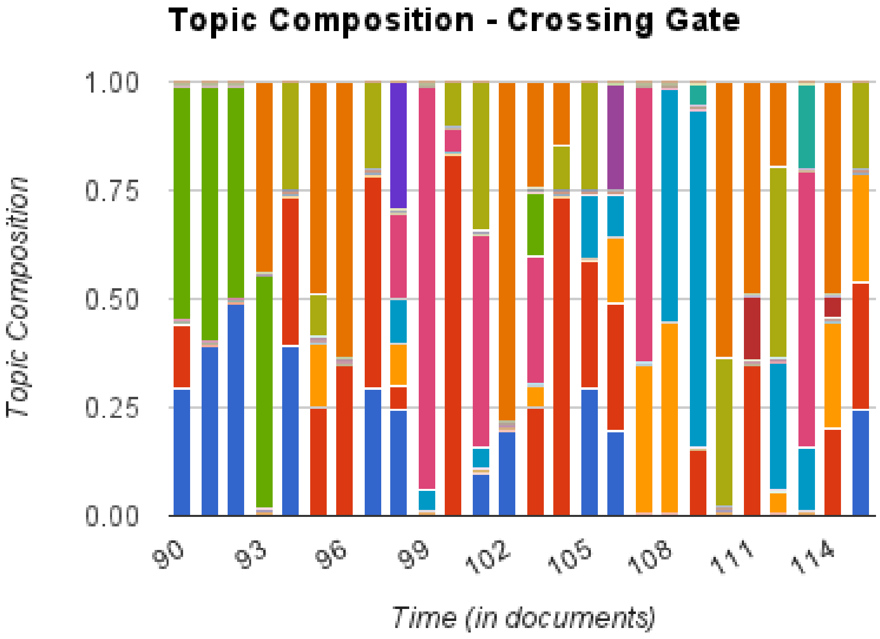


Figure 5–3: Topic compositions of documents - Go to Gym

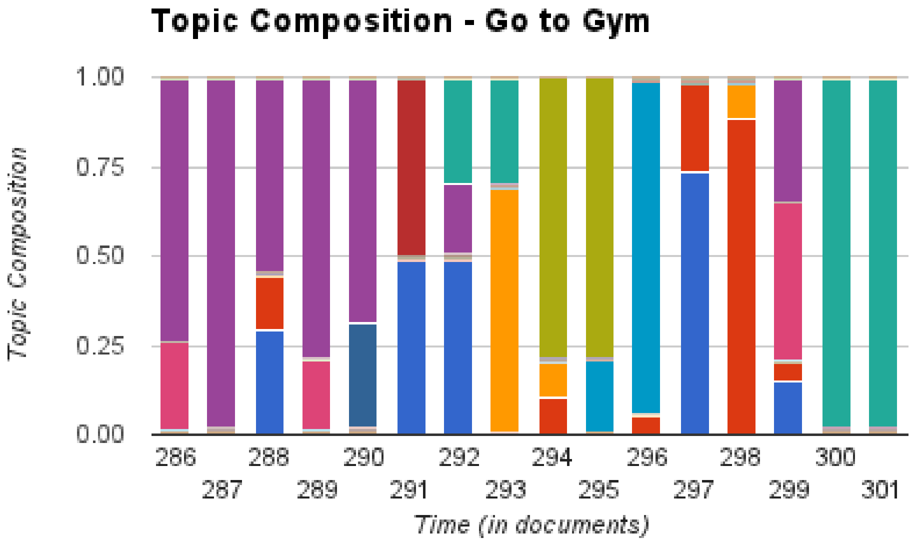
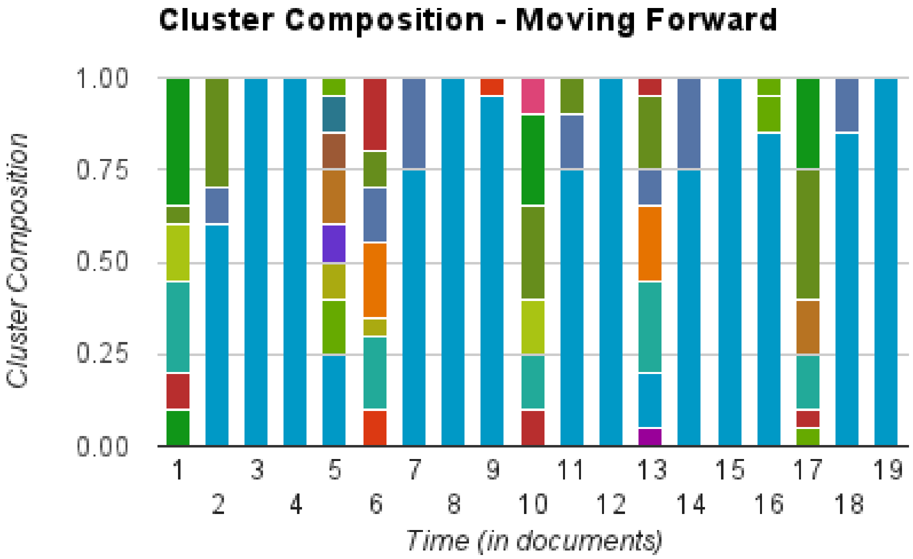
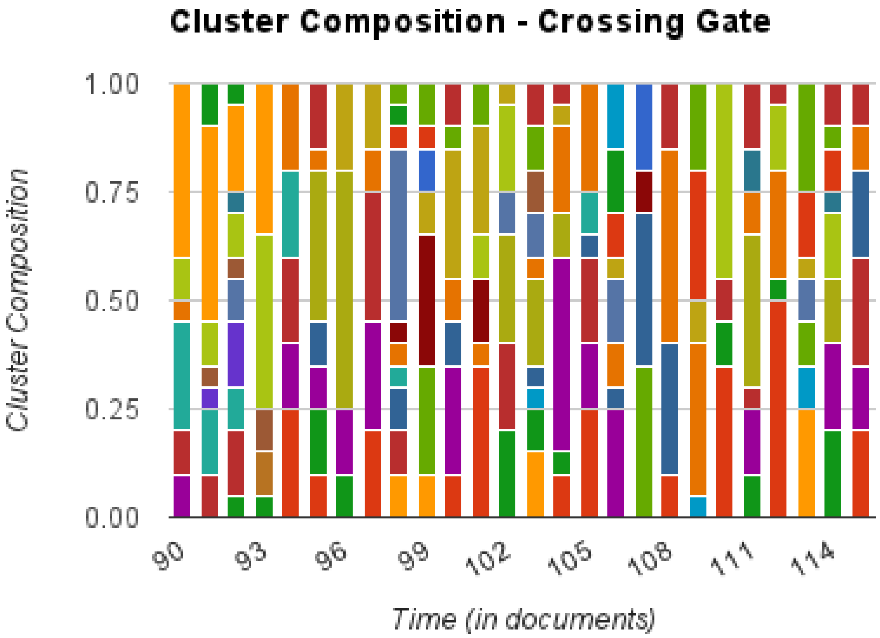


Figure 5–4: Cluster compositions of documents - Moving Forward



though, since giving labels in a per-window basis involves a tremendous amount of work. As a secondary measure, we use per-task labels (which is also the labelled

Figure 5–5: Cluster compositions of documents - Crossing Gates

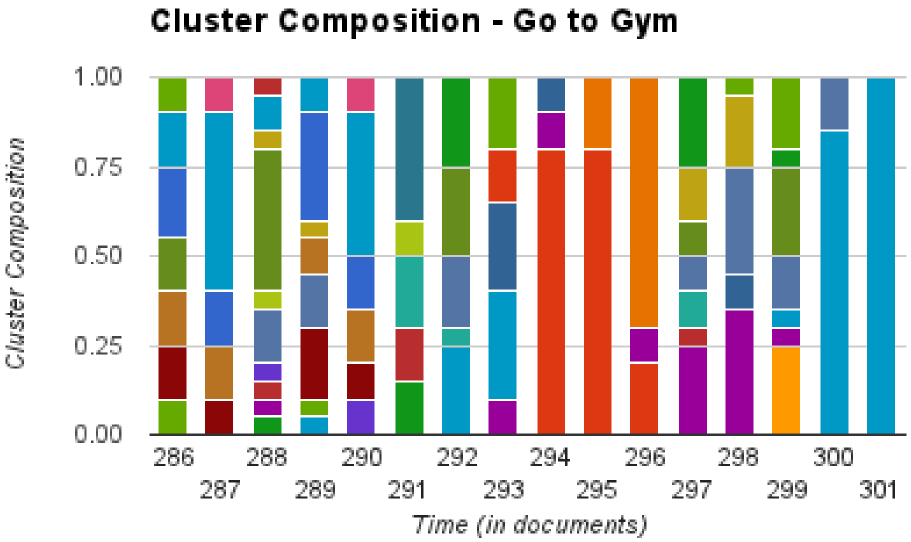


task code in our dataset) to approximate per-window labels. This means that for all windows coming from a particular task, we simply label them with the task code, and use this to calculate the task composition for each topic. The results are shown in Table 5–3 (we only include tasks that account for at least 10% of total.)

5.2.3 Quantitative Measures

In a broader sense, both topics and clusters define a grouping of datapoints, with the goal of putting similar items in the same group and putting different items in different groups. Purity, precision and recall offer quantitative measures to evaluate this kind of grouping quality. They were originally used to analyze clusters, but can be extended easily to analyze topics. In this subsection, we compare these metrics

Figure 5–6: Cluster compositions of documents - Go to Gym



between cluster composition and topic composition. More detailed explanation on these metrics can be found in the Information Retrieval literature (see Chapter 16.3 of [33]); We give a short description here. Continuing with our previous notations, purity, precision and recall of cluster composition are defined as:

$$Purity_C = \frac{1}{\sum_i N_i} \sum_k \max_i (CC_i^k \cdot N_i) \tag{5.1}$$

$$Precision_C = \frac{TP_C}{TP_C + FP_C} \tag{5.2}$$

$$Recall_C = \frac{TP_C}{TP_C + FN_C} \tag{5.3}$$

Table 5–3: Task Compositions of Topics

Topics	Dominant Tasks				
1	T4 (14%)	T5 (13%)	T3 (12%)	T1 (10%)	T21 (10%)
2	T8 (19%)	T24 (10%)	T18 (10%)		
3	T18 (26%)	T19 (24%)			
4	T5 (23%)	T4 (21%)	T22 (14%)	T6 (11%)	T23 (10%)
5	T3 (44%)	T4 (31%)	T11 (13%)		
6	T15 (30%)	T12 (16%)	T13 (10%)		
7	T18 (17%)	T19 (12%)			
8	T12 (13%)	T7 (11%)	T21 (10%)		
9	T24 (14%)	T25 (13%)	T18 (12%)		
10	T4 (30%)	T3 (18%)	T22 (10%)		
11	T25 (31%)	T24 (22%)	T28 (17%)		
12	T1 (71%)	T28 (14%)			
13	T2 (51%)	T13 (20%)			
14	T24 (16%)	T20 (14%)	T3 (13%)	T21 (13%)	T8 (10%)
15	T8 (11%)	T13 (10%)			

$$\begin{aligned} TP_C &= \sum_k \sum_i \binom{CC_i^k \cdot N_i}{2} \\ &= \sum_k \sum_i \frac{(CC_i^k \cdot N_i) \cdot (CC_i^k \cdot N_i - 1)}{2} \end{aligned}$$

$$FP_C = \sum_k \sum_{i_1} \sum_{i_2 > i_1} (CC_{i_1}^k \cdot N_{i_1}) \cdot (CC_{i_2}^k \cdot N_{i_2})$$

$$FN_C = \sum_{k_1} \sum_{k_2 > k_1} \sum_i (CC_i^{k_1} \cdot N_i) \cdot (CC_i^{k_2} \cdot N_i)$$

Purity, precision and recall for topic composition are defined similarly by replacing CC_i^k with TC_i^t, $\sum\limits_k$ with $\sum\limits_t$ and setting $N_i = L$. For example:

$$Purity_T = \frac{1}{\sum\limits_i L} \sum_t \max_i (TC_i^t \cdot L) \tag{5.4}$$

$$= \frac{1}{M} \sum_t \max_i (TC_i^t) \tag{5.5}$$

Intuitively, purity measures the dominance of the most frequent class within the groups, whereas precision and recall measure the correctness of grouping similar items. TP, FP, FN stand for True Positive (number of pairs of windows with the same task labels put in the same group), False Positive (number of pairs of windows with different task labels put in the same group) and False Negative (number of pairs of windows with same task labels put in different groups) respectively.

Table 5–4 shows that topic composition performs much better than cluster composition in terms of these metrics. Column 1 and 3 show the results with the best cross-validation selected parameters (40 clusters and 15 topics respectively). For the purpose of comparing same number of groupings between clusters and topics, we also include results for cluster composition using 15 clusters in Column 2.

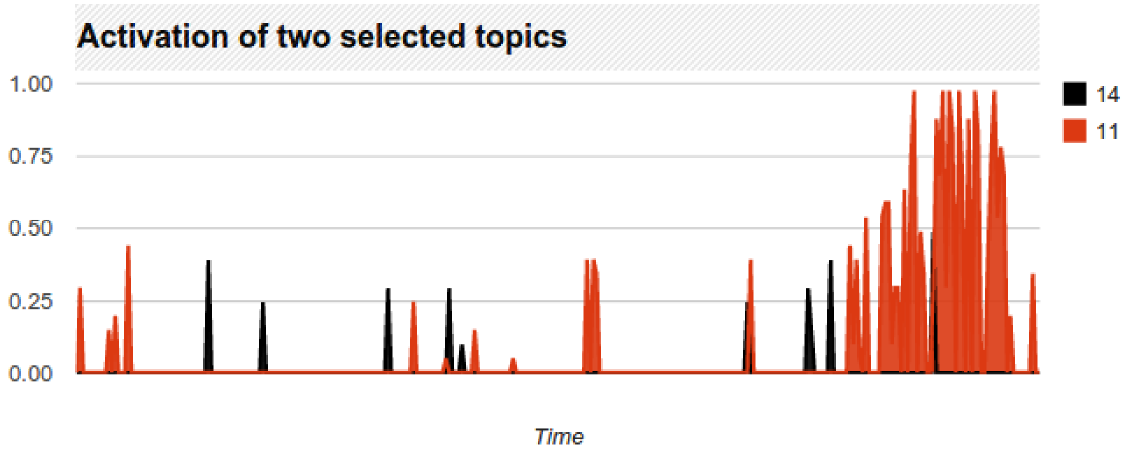
Table 5–4: Purity, Precision and Recall on Individual-Set

	Cluster Composition (40 clusters)	Cluster Composition (15 clusters)	Topic Composition (15 topics)
Purity	35.74%	25.64%	52.25%
Precision	24.01%	13.33%	36.45%
Recall	18.48%	34.40%	65.99%

5.2.4 Hazard Discovery

Finally, we conclude this section by introducing a potential use of inferred topics. We consider a particular topic, and plot its composition across documents, thus we can observe the activation of that topic across time (documents are aligned with time). We selected two topics, 11 and 14, to show the idea. The result is shown in Figure 5–7. Referring to Table 5–3, topics 11 and 14 constitute mostly 'Avoid objects' and 'Collisions'. Suppose, from prior knowledge, we know that these types of tasks are dangerous. By analyzing their activations across time, we could identify some hazardous zones during the use of the smart wheelchair. By correlating this with the smart wheelchair's localization in the environment, it may be possible to identify problematic areas, in addition to difficult activities.

Figure 5–7: Topic activations for two selected topics



5.3 Summary

To summarize the experiments, we have achieved around 50% accuracy in activity classification. We would argue that the classifier is doing reasonably well considering that it is a 25-class classification problem, where a random classifier would give an accuracy of just 4%. Moreover, the misclassified activities are usually not distributed randomly, but mixed up only with similar items, as shown in the confusion matrix.

On the other hand, we demonstrated the usefulness of pattern discovery with topic modeling in terms of story telling and hazard discovery examples. We also quantitatively show that topic composition is a better grouping than cluster composition in terms of purity, precision and recall. Given that unsupervised learning is hard to evaluate, our mixed qualitative and quantitative approach show that topic modeling could be a powerful tool to understand the latent activity patterns.

Chapter 6
CONCLUSION

This thesis presents a machine learning approach to characterize and discover activities during the use of intelligent powered wheelchairs. As per our knowledge, this is one of the first studies applying activity recognition and topic modeling methods to wheelchair data. We have summarized the main contributions and possibilities for future work below.

6.1 Contributions

First, we proposed a new classification method using cluster composition as a feature vector. We compared the results of using cluster composition to ordinary FFT feature and found that cluster composition is more robust to noise. It is also shown to be more straight forward and efficient than per-window FFT features as a way to classify activities.

Second, we explored the possibility of applying topic modeling, more specifically the LDA model, to characterize wheelchair activities. We demonstrated the potential of pattern discovery with a mix of qualitative and quantitative results. As such, it provides a new tool for analysing wheelchair activities in an unsupervised manner.

Third, we constructed an end-to-end pipeline from capturing sensor data to activity recognition. The neat thing about the pipeline is that the supervised classification side and unsupervised topic modeling side share a lot of common components, which eases the implementation work.

Altogether, our work contributes to the development of a full-fledged monitoring system on smart wheelchairs, as well as other assistive and rehabilitation robots because the presented pipeline can be transferred easily to other robotic system.

6.2 Future work

First, we would like to apply the proposed methods on other datasets, especially for the topic modeling part. The dataset we used is task oriented, meaning that well defined tasks are executed for numerous trials. However, to demonstrate the effect of topic modeling, it would be more interesting to see the results on a more continuous, natural and long-term driving sequences. Ideally, the dataset should be labelled with high level patterns as well as low level primitive actions so we can better evaluate the results. A real time video recordings of the ride, so we can show the topic activations alongside a visual, would also be useful to validate the results.

Second, we would like to see more usages of topic modeling other than story telling (Section 5.2.1) and hazard discovery (Section 5.2.4). Our current work is still at the exploratory level, and there are still plenty of room for us to discover what else to do with the latent topics. Nevertheless, having an unsupervised method is very powerful and useful because of the scarcity of labels in real life.

Third, we would like to push the activity recognition pipeline online so we can operate in real time. Right now, we analyse the activities in batch. We foresee that this should not be too difficult given our current pipeline. On the pattern discovery side of the pipeline, we can use online LDA [50] instead of ordinary LDA. On the classification side, nearest neighbour classification can be done easily in real time using a data structure like kd-trees [51].

Appendix A
Topic/Cluster Composition Graphs

Here, we presents the full topic composition graphs and cluster composition graphs across a total of 302 documents mentioned in the Task Composition section in Methodology chapter.

Figure A–1: Topic composition - Full Graph (Part 1)

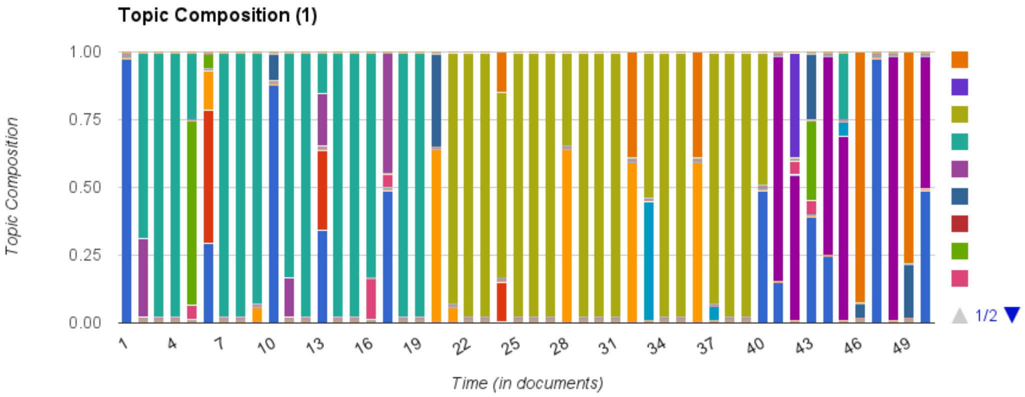


Figure A–2: Topic composition - Full Graph (Part 2)

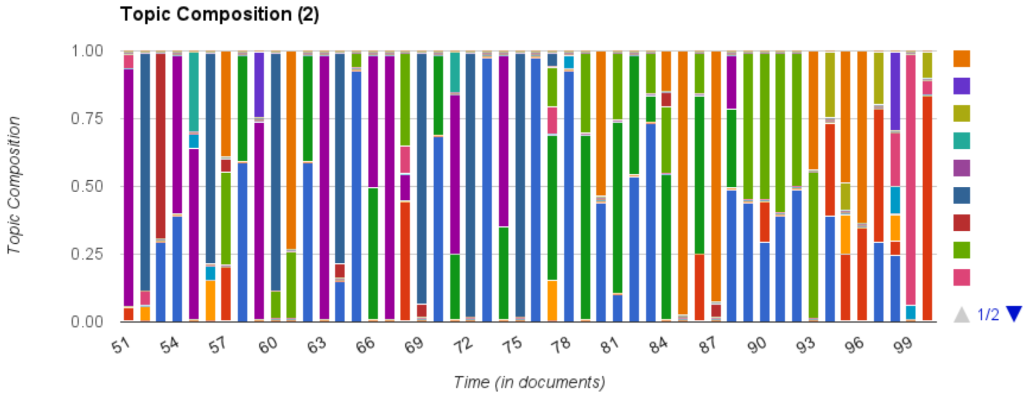


Figure A–3: Topic composition - Full Graph (Part 3)

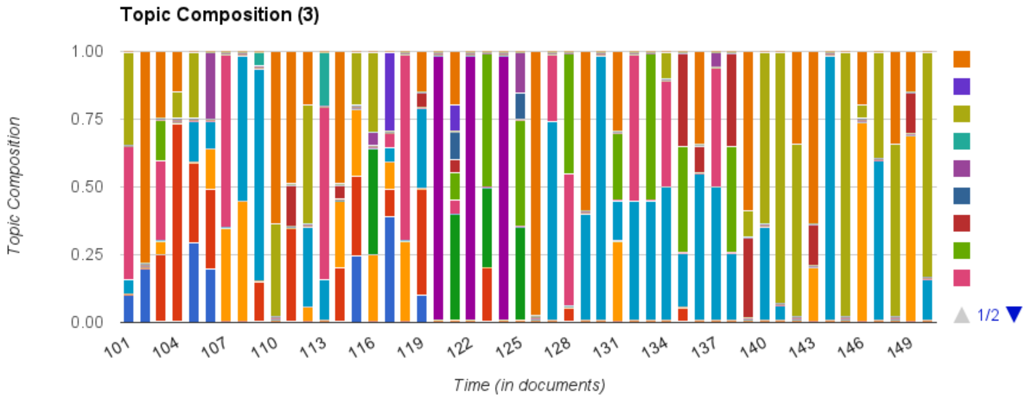


Figure A–4: Topic composition - Full Graph (Part 4)

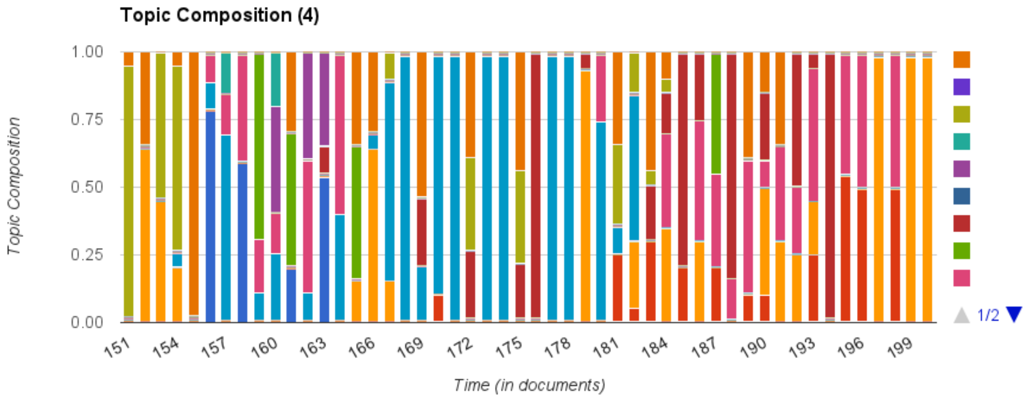


Figure A–5: Topic composition - Full Graph (Part 5)

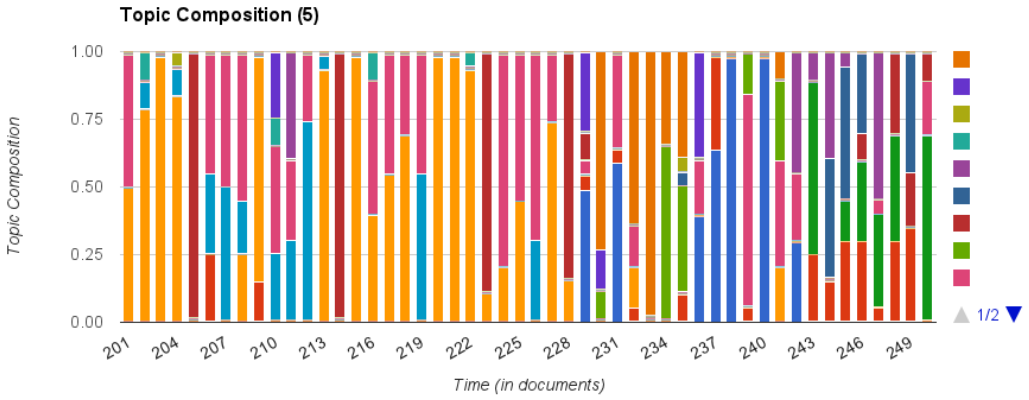


Figure A–6: Topic composition - Full Graph (Part 6)

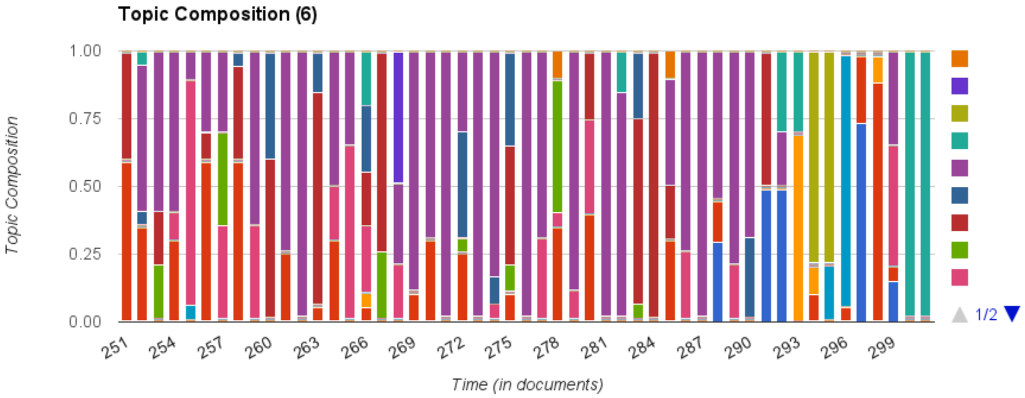


Figure A–7: Cluster composition - Full Graph (Part 1)

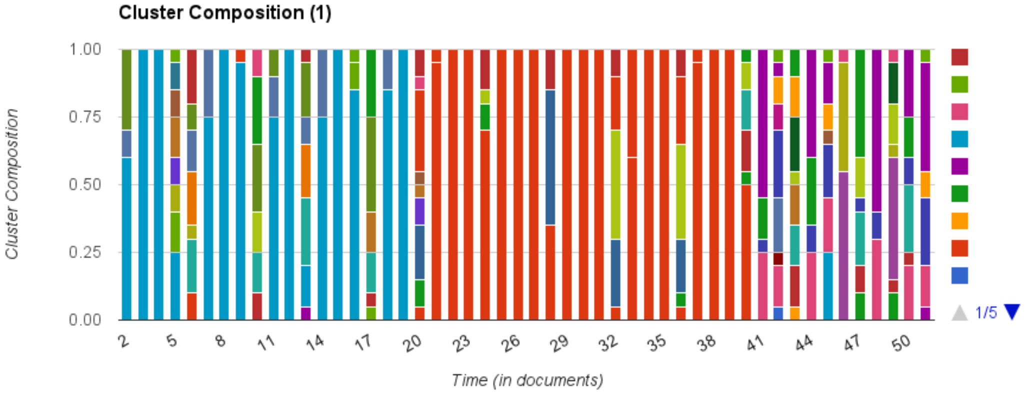


Figure A–8: Cluster composition - Full Graph (Part 2)

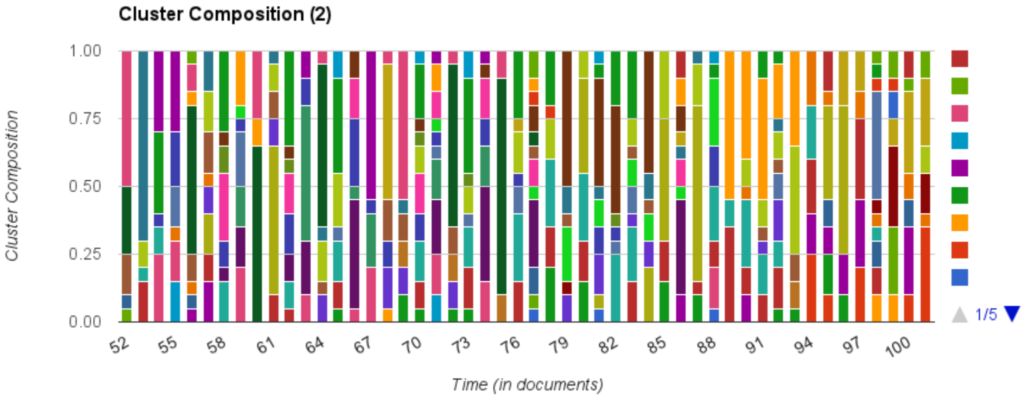


Figure A–9: Cluster composition - Full Graph (Part 3)

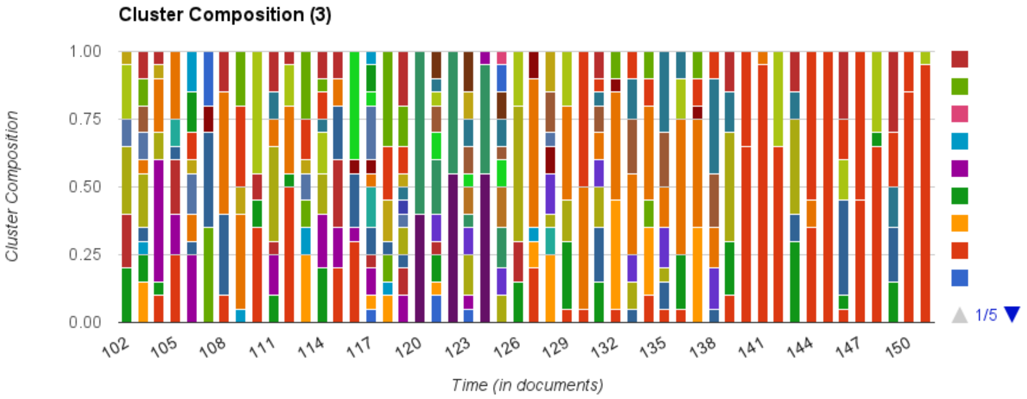


Figure A–10: Cluster composition - Full Graph (Part 4)

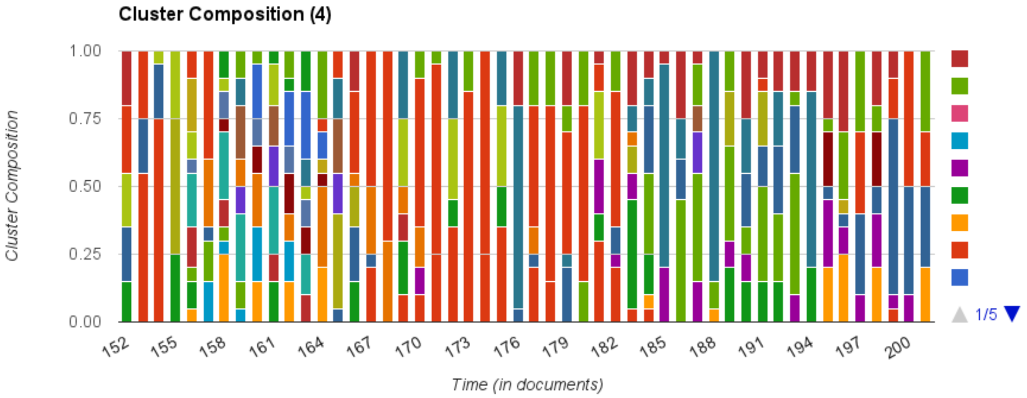


Figure A–11: Cluster composition - Full Graph (Part 5)

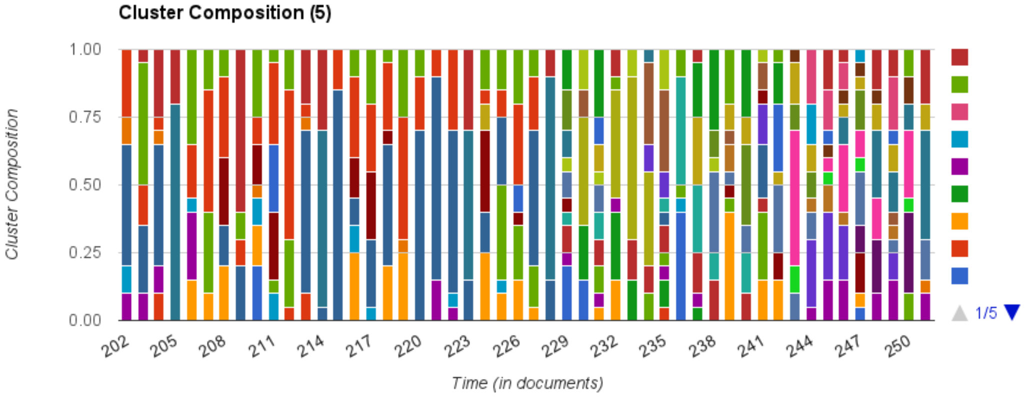
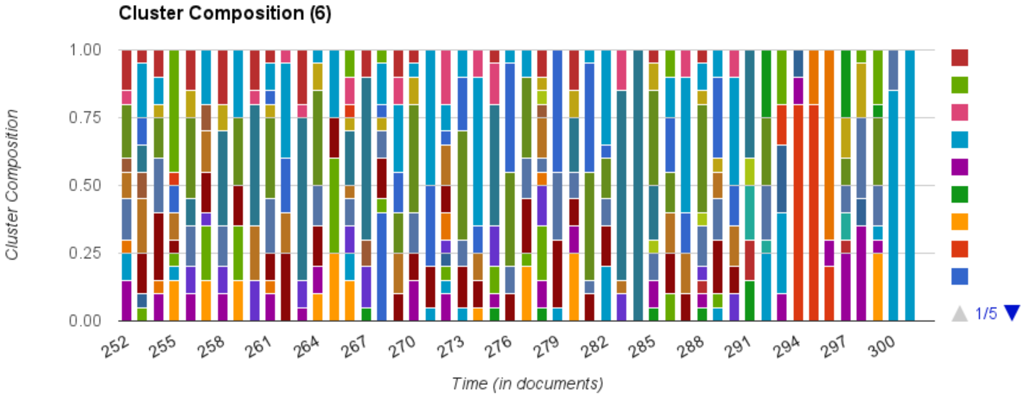


Figure A–12: Cluster composition - Full Graph (Part 6)



References

[1] L. Fehr, W. E. Langbein, and S.B. Skaar. "Adequacy of power wheelchair control i terfaces for persons with severe disabilities: A clinical survey." Development 37.3 (2000): 353-360.

[2] P. Turaga, et al. "Machine recognition of human activities: A survey." Circuits and Systems for Video Technology, IEEE Transactions on 18.11 (2008): 1473-1488.

[3] R. C. Simpson, and S. P. Levine. "Voice control of a powered wheelchair." Neural Systems and Rehabilitation Engineering, IEEE Transactions on 10.2 (2002): 122-125.

[4] S. P. Levine, et al. "The NavChair assistive wheelchair navigation system." Rehabilitation Engineering, IEEE Transactions on 7.4 (1999): 443-451.

[5] H. A. Yanco "Wheelesley: A robotic wheelchair system: Indoor navigation and user interface." Assistive technology and artificial intelligence. Springer Berlin Heidelberg (1998): 256-268.

[6] Y. Kuno, N. Shimada, and Y. Shirai. "Look where you're going [robotic wheelchair]." Robotics & Automation Magazine, IEEE 10.1 (2003): 26-34.

[7] C. Castellini, and R. Koiva. "Intention Gathering from Muscle Residual Activity for the Severely Disabled." IROS 2012, Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs (2012).

[8] J.A. Meyer, and D. Filliat. "Map-based navigation in mobile robots:: II. a review of map-learning and path-planning strategies." Cognitive Systems Research 4.4 (2003): 283-317.

[9] A.K. Moghaddam "Automatic Detection and Classication of Events on Power Wheelchairs Using Embedded Sensors" Master Thesis, McGill University. (2013)

[10] I. Pettersson, G. Ahlström, and K. Törnquist. "The value of an outdoor powered wheelchair with regard to the quality of life of persons with stroke: A follow-up study." Assistive technology 19.3 (2007): 143-153.

[11] R. C. Simpson, E. F. LoPresti, and R. A. Cooper. "How many people would benefit from a smart wheelchair?." Journal of rehabilitation research and development 45.1 (2008): 53-72.

[12] P. Boucher, A. Atrash, S. Kelouwani, W. Honoré, H. Nguyen, J. Villemure, F. Routhier, P. Cohen, L. Demers, R. Forget and J. Pineau. "Design and validation of an intelligent wheelchair towards a clinically-functional outcome." Journal of neuroengineering and rehabilitation 10.1 (2013): 58.

[13] IROS 2012 Workshop on Progress, Challenges and Future Perspectives in Navigation and Manipulation Assistance for Robotic Wheelchairs. http://www.radhar.eu/events/IROS2012-robotic-wheelchairs.

[14] C. Gao, T. Miller, J.R. Spletzer, I. Hoffman, T. Panzarella. "Autonomous docking of a smart wheelchair for the automated transport and retrieval system (ATRS)". J Field Robot (2008): 203-222.

[15] O. Horn O, M. Kreutner. "Smart wheelchair perception using odometry, ultrasound sensors, and camera". Robotica 27.2 (2009): 303-310.

[16] R. C. Simpson "Smart wheelchairs: A literature review." Journal of rehabilitation research & development 42.4 (2005): 423-438.

[17] L. Montesano, et al. "Towards an intelligent wheelchair system for users with cerebral palsy." Neural Systems and Rehabilitation Engineering, IEEE Transactions on 18.2 (2010): 193-202.

[18] S. Gulati, B. Kuipers. "High performance control for graceful motion of an intelligent wheelchair." Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, (2008).

[19] R.L. Kirby, S.A. Ackroyd-Stolarz, M.G. Brown, S.A. Kirkland, D.A. MacLeod. "Wheelchair-related accidents caused by tips and falls among noninstitutionalized users of manually propelled wheelchairs in Nova Scotia." American Journal of Physical Medicine & Rehabilitation 73.5 (1994): 319-330.

[20] N. Cristianini, J. Shawe-Taylor. "An introduction to support vector machines and other kernel-based learning methods" Cambridge university press (2000).

[21] D.M. Blei, A.Y. Ng, and M.I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research Vol.3 (2003): 993-1022

[22] I. Porteous, et al. "Fast collapsed gibbs sampling for latent dirichlet allocation." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM (2008).

[23] F. J. Harris "On the use of windows for harmonic analysis with the discrete Fourier transform." Proceedings of the IEEE 66.1 (1978): 51-83.

[24] J. W. Cooley, P. AW Lewis, and P. D. Welch. "The fast Fourier transform and its applications." Education, IEEE Transactions on 12.1 (1969): 27-34.

[25] P. Boissy, S. Brière, M. Hamel, M. Jog, M. Speechley, A. Karelis, J. Frank, C. Vincent, R. Edwards, C. Duval and the EMAP group. "Wireless inertial measurement unit with GPS (WIMU-GPS)—Wearable monitoring platform for ecological assessment of lifespace and mobility in aging and disease." Annual International Conference of the IEEE. (2011).

[26] J. Pineau, A.K. Moghaddam, H.K. Yuen, P. Archambault, F. Routhier, F. Michaud, P. Boissy. "Automatic Detection and Classification of Unsafe Events during Power Wheelchair Use." Unpublished.

[27] N. Ravi, N. Dandekar, P. Mysore, M.L. Littman. "Activity recognition from accelerometer data." AAAI Conference on Artificial Intelligence. Vol.5. (2005).

[28] L. Bao, S.S. Intille. "Activity recognition from user-annotated acceleration data." Pervasive Computing. Springer Berlin Heidelberg (2004): 1-17.

[29] T. Huynh, M. Fritz, and B. Schiele. "Discovery of activity patterns using topic models." Proceedings of the 10th international conference on Ubiquitous computing. ACM (2008).

[30] S. Ray, and R.H. Turi. "Determination of number of clusters in k-means clustering and application in colour image segmentation." Proceedings of the 4th international conference on advances in pattern recognition and digital techniques. (1999).

[31] J. Frank, et al. "Time series analysis using geometric template matching." Pattern Analysis and Machine Intelligence, IEEE Transactions on 35.3 (2013): 740-754.

[32] Dalhousie University. Wheelchair Skills Test, version 4.1. Available at: http://www.wheelchairskillsprogram.ca/eng/4.1/WST_Manual_Version4.1.51.pdf. Accessed March 5 (2010).

[33] C.D. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval. Vol.1. Cambridge: Cambridge University Press (2008): 356-359.

[34] I. H. Witten, and E. Frank. "Data Mining: Practical machine learning tools and techniques" Morgan Kaufmann (2005).

[35] C. M. Bishop "Pattern recognition and machine learning". Vol.1. New York: springer (2006).

[36] F. Sebastiani. "Machine learning in automated text categorization." ACM computing surveys (CSUR) 34.1 (2002): 1-47.

[37] M. H. Kutner "Applied linear statistical models". Vol.4. Chicago: Irwin, (1996).

[38] A. L. Blum, and P. Langley. "Selection of relevant features and examples in machine learning." Artificial intelligence 97.1 (1997): 245-271.

[39] T. Cover, and P. Hart. "Nearest neighbor pattern classification." Information Theory, IEEE Transactions on 13.1 (1967): 21-27.

[40] P. E. Danielsson "Euclidean distance mapping." Computer Graphics and image processing 14.3 (1980): 227-248.

[41] P. E. Black "Manhattan distance." Dictionary of Algorithms and Data Structures 18 (2006): 2012.

[42] T. Kanungo, et al. "An efficient k-means clustering algorithm: Analysis and implementation." Pattern Analysis and Machine Intelligence, IEEE Transactions on 24.7 (2002): 881-892.

[43] D. Aloise, et al. "NP-hardness of Euclidean sum-of-squares clustering." Machine Learning 75.2 (2009): 245-248.

[44] K. Alsabti, S. Ranka, and V. Singh. "An efficient k-means clustering algorithm." Electrical Engineering and Computer Science (1997): paper 43.

[45] H. Abdi, and L. J. Williams. "Principal component analysis." Wiley Interdisciplinary Reviews: Computational Statistics 2.4 (2010): 433-459.

[46] J. Shlens "A tutorial on principal component analysis." Systems Neurobiology Laboratory, University of California at San Diego 82 (2005).

[47] M. Steyvers, and T. Griffiths. "Probabilistic topic models." Handbook of latent semantic analysis 427.7 (2007): 424-440.

[48] C. Schaffer "Selecting a classification method by cross-validation." Machine Learning 13.1 (1993): 135-143.

[49] P. Bloomfield "Fourier analysis of time series: an introduction" John Wiley & Sons (2004).

[50] M. D. Hoffman, D. M. Blei, and F. R. Bach. "Online Learning for Latent Dirichlet Allocation." NIPS. Vol. 2. No.3. (2010).

[51] A. W. Moore "An introductory tutorial on kd-trees" (Technical report 209). Computer Laboratory, University of Cambridge. Extract from AW Moore's Phd. Diss. thesis: Efficient Memory-based Learning for robot Control, 1991.

[52] J. Connell, and P. Viola. "Cooperative control of a semi-autonomous mobile robot." Proceedings of the IEEE International Conference on Robotics and Automation. Vol.2. (1990).

[53] S. P. Parikh, et al. "Human robot interaction and usability studies for a smart wheelchair." Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. Vol. 4. IEEE, (2003).

[54] N. I. Katevas, et al. "The autonomous mobile robot SENARIO: a sensor aided intelligent navigation system for powered wheelchairs." Robotics & Automation Magazine, IEEE 4.4 (1997): 60-70.

[55] T. L. Lee, et al. "The omni-directional wheelchair for the elderly." Gerontechnology 7.2 (2008): 148.

[56] P. D. Nisbet "Who's intelligent? Wheelchair, driver or both?." Control Applications, 2002. Proceedings of the 2002 International Conference on. Vol. 2. IEEE, (2002).

[57] T. Carlson, and Y. Demiris. "Human-wheelchair collaboration through prediction of intention and adaptive assistance." Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on. IEEE, (2008).

[58] P. Encarnação "Understanding and Improving Power Mobility Use among Older Adults: An Overview of the Canwheel Program of Research." Assistive Technology: From Research to Practice: AAATE 2013 33 (2013): 210.

[59] J. Pineau, and A. Atrash. "SmartWheeler: A Robotic Wheelchair Test-Bed for Investigating New Models of Human-Robot Interaction." AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics. (2007).

[60] V. Madisetti "Digital signal processing fundamentals" CRC press, (2010).