Quantifying Information Loss Resulting from Degradation of IMU Sensor Resolution in PD Symptom Classification

Scientific thesis for the procurance of the degree B.Sc.
from the Department of Electrical and Computer Engineering at the Technical University of Munich.

Supervised by Univ.-Prof. Dr.-Ing. Sandra Hirche
Satoshi Endo,Das, Neha,
Chair of Information-Oriented Control

Submitted by Bachelor of Science Shuyang Tian
Helene-Mayer-Ring 12
80809 Munich
049- 15224376828

Submitted on Munich, 11.03.2021

TECHNISCHE UNIVERSITÄT MÜNCHEN

LEHRSTUHL FÜR INFORMATIONSTECHNISCHE REGELUNG

ORDINARIA: UNIV.-PROF. DR.-ING. SANDRA HIRCHE

October 15, 2021

BACHELOR THESIS
for
Shuyang Tian
Student ID 03709169, Degree EI

Quantifying Information Loss Resulting from Degradation of IMU Sensor Resolution in PD Symptom Classification

Problem description:

Parkinson's Disease (PD) is often characterized by movement abnormalities. Several existing solutions for continually monitoring these symptoms necessitate the use of expensive proprietary hardware for measuring motion signals that can, through data driven methods, indicate the PD symptom and its severity.

There is thus, a demand for more cost-effective solutions that can utilize low-cost sensors for measuring these signals. This work will investigate the viability of using such low cost sensors, which may in effect have lower resolution than their more expensive counterparts.

We will quantify the information loss that results from such degradation of IMU sensor precision on simulated data and validate our findings on different physical IMU Sensors.

Tasks:

- Literature review of methods for quantifying information loss
- Implementation/development of appropriate methods to quantify the information loss resulting from degrading precision of IMU sensor data.
- Inferring minimum specifications required of an IMU sensor for performance beyond a certain threshold

Supervisor:	MSc. Neha Das (neha.das@tum.de), Dr. Satoshi Endo (s.endo@tum.de)
Start:	22.10.2021
Intermediate Report:	03.02.2022
Delivery:	11.03.2022

(S. Hirche)
Univ.-Professor

Abstract

Abnormal movements are often characteristic of Parkinson's disease (PD), and to monitor these movements, IMU sensors can be used to measure the movement signal. Due to the high cost of high-resolution sensors, we will investigate the feasibility of using low-cost sensors, i.e., low resolution. This work will quantify the loss of information to the activity dataset due to the reduced resolution of the IMU sensors. After acquiring the IMU dataset at the original sampling rate, it will be resampled using a different sampling rate to simulate the dataset later collected by the low-resolution IMU sensor. We will then extrapolate the minimum requirements for the IMU sensor within a certain threshold. For this work, we first classify the inertial measurement unit (IMU) data and ratings of Parkinson's symptoms by the k-Nearest Neighbor (KNN) algorithm. Then, by calculating the information loss between different sampling frequencies, we can observe the change in information loss that accompanies the decrease in sampling frequency to derive a threshold value.

Contents

Chapter 1

Introduction

Parkinson's disease (PD) is a chronic neurodegenerative disease affecting the central nervous system, mainly the nervous motor system.[1] Its symptoms usually appear slowly over time, with the most prominent early symptoms being tremors, limb stiffness, reduced motor function, and abnormal gait. There may also be cognitive and behavioral problems; dementia is quite common in patients with severe disease. Major depressive disorder and anxiety disorders also occur in more than a third of cases. Other symptoms accompanying the disease include perception, sleep, and mood problems. The main motor symptoms associated with Parkinson's disease are collectively known as Parkinson's syndrome. Parkinson's disease has four main motor symptoms: tremor, bradykinesia, dyskinesia, and postural instability.
[1, 2]
Tremor is the most apparent and well-known symptom, it occurs most often in the hands or arms, but the trunk or head may also be affected. People with Parkinson's disease do not tremble at the beginning of the disease, but most patients gradually develop this symptom as the disease progresses. In Germany, at least one in every hundred people suffers from primary tremor, i.e., tremor without an identifiable underlying neurological disorder.[3] A tremor in Parkinson's disease is most pronounced when the limb is at rest but disappears during sleep or when the limb is consciously moved. Tremor affects more of the distal extremities, usually appearing initially in one hand or foot at the onset but then spreading to both hands and feet.

Bradykinesia is another feature of Parkinson's disease, where the patient's movements are slowed down and affect the entire process from movement initiation to execution. The patient cannot make continuous movements or perform different movements simultaneously. [2] Bradykinesia, a form of hypokinesia, emphasizes the slow movement of motor execution and is a common symptom in the early stages of Parkinson's disease. The effects of bradykinesia vary according to the type of movement and the patient's physical and mental stateâthe patient's mobility and emotional state influence the degree of impact. In general, patients with Parkinson's disease can improve their symptoms of bradykinesia with treatment.

Dyskinesia is caused by increased muscle tone and sustained muscle contraction, resulting in difficulty moving the limbs. Limb stiffness may also be associated with arthralgia, which is often present in the early stages of the disease. In the early stages of Parkinson's disease, limb stiffness is often asymmetrical. It tends to develop in the neck and shoulders, then spreads to the face and extremities, and eventually to the whole body as the disease progresses, causing a gradual loss of movement.[4] Bradykinesia, dyskinesia, tremor will discuss these three types for our classification of Parkinson's patients' symptoms.

Parkinson's disease may also have other symptoms, as shown in figure 1.1. The patient may have little or no facial expressions in the early stages and may not swing their arms when walking. However, as the disease progresses, there will probably be changes in speech and writing.

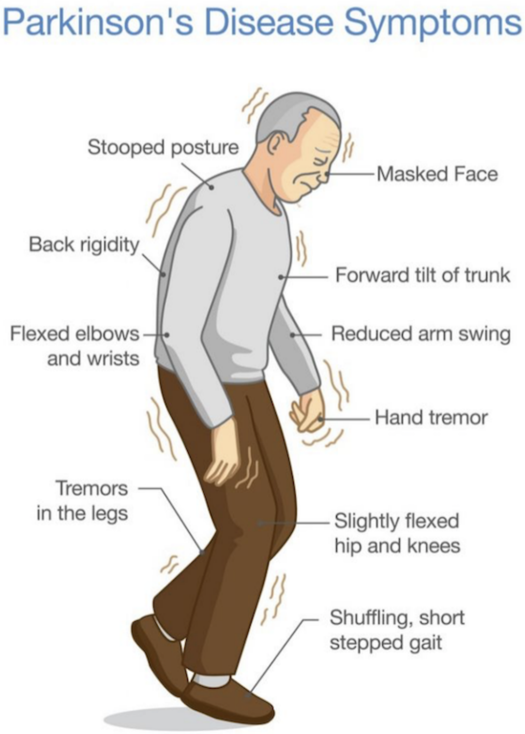


Figure 1.1: Parkinson's Disease Symptoms [5]

Today, IMU sensors are widely used to study Parkinson's motor disease. Chapter 2 clearly describes how IMU sensors work. IMU sensors can collect motion data from the hands or feet of Parkinson's patients, and this motion data can help us analyze Parkinson's disease. However, there are many IMU sensors on the market at different prices, such as aerospace-grade sensors, which are sold at high prices and are used in military aerospace and other fields. In contrast, commercial IMU sensors are relatively inexpensive. Differences in IMU accuracy also determine price differences. Our work quantifies the loss of information when the resolution of IMU sensors is reduced. We resample the dataset collected by the original IMU sensor to create a simulated dataset collected by a low-resolution IMU sensor. After quantifying the information loss, we can infer that the IMU sensor works well at reduced resolution. First, we chose a 5-second Parkinson's time as the test dataset. We want to calculate the information loss by obtaining the predicted classification probability after classification using KNN. It is challenging to handle a super large amount of data. So we made a new step based on the mutual information between IMU data (acceleration data) and symptom score data. We want to calculate the information loss by obtaining the predicted classification probability after classification using KNN. Therefore, we obtained the information loss by simplifying the mutual information formula, as explained in Chapter 3. Chapter 5 specifies the definition of the dataset and variables we use. Chapters 6 and 7 discuss the procedures and methods for calculating the information loss based on KNN classification.

Chapter 2

Technical equipment

2.1 Inertial measurement unit

Inertial measurement unit (IMU) data, an inertial measurement unit (IMU), usually refers to a combined unit consisting of three accelerometers and three gyroscopes mounted on mutually perpendicular measurement axes.[6] Inertial sensors including accelerometers (or accelerometer sensors) and angular velocity sensors (gyros) and their single, dual, and triaxial combinations IMU (Inertial Measurement Unit), AHRS (Attitude Reference System including magnetic sensors). IMU sensors are used in everyday life, smartphones, smartwatches, and commercial drones, among other electronic products. Commercially available off-the-shelf IMU sensors are small, lightweight, wireless devices that IMU sensors can quickly fix to a body part without interfering with motion. Depending on their application, they are typically embedded in Bluetooth/wireless transmitters or SD cards for real-time data streaming or onboard long-term data recording, respectively.

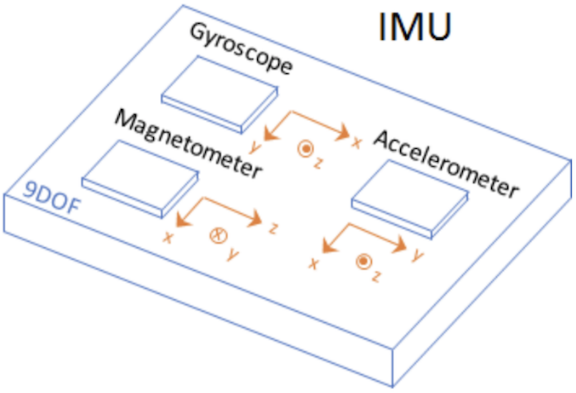


Figure 2.1: IMU Sensors' internal structure includes an accelerometer, Magnetometer, and gyroscope.[7]

2.2 Working Principle

An IMU sensor unit working can be done by using one or more accelerometers to detect linear acceleration and similarly by using gyroscopes to detect rotation rates and magnetometers. [8]

2.3 IMU sensors in our work

Nowadays, wearable IMU sensors are widely used in medical research, and the acceleration data collected by IMU helps to predict and analyze patient data in the future. Wearable IMU sensors will collect daily data from Parkinson's patients in our present work. We will collect acceleration and angular acceleration from the wrists, ankles, and other body parts of people with Parkinson's disease. We use the MJFF Levodopa wearable sensor dataset [9], and the Fox Foundation collects wearable sensor data from patients with Parkinson's disease (PD). Subjects were recruited from two clinical sites and monitored while performing a range of routine activities in the clinic and daily activities at home. In Chapter 5, we describe the data collection information in detail.

Figure 2.2: Shimmer3 offers the best data quality with integrated 9DoF inertial sensing via accel, gyro, mag, and pressure sensor, each with selectable range.
[10]

The Shimmer sensor is used in medical, neuroscience, and clinical research and is a highly accurate IMU sensor. The Shimmer IMU comes with integrated 9 DoF + altimeter inertial sensing via accelerometer, gyroscope, and magnetometer, each with a selectable range.[10] But the shimmer sensor is not cheap. Just one Shimmer sensor needs to spend 598 euros. The truth is that IMU sensors are ubiquitous in our lives, such as cell phones and smartwatches. So it is better value for money to wear a watch and a cell phone than to wear a professional IMU sensor.

Chapter 3

Theoretical background

3.1 Entropy

The entropy of a random variable is a function that describes the "unpredictability" of the random variable. In information theory, entropy is the average amount of information contained in each piece of information received, also known as information entropy. [11, 12]

The information entropy formula given by Shannon [11] , for random variable X, is defined as follows, in bits:

$$H(X) = -\sum_{i=1}^{n} p(x_i)log(p(x_i)) \tag{3.1}$$

where $p(x_i)$ represents the probability of the random variable x.

How much information one has needs to be measured by the amount of information. The amount of information received is related to the specific events that occurred. Therefore, the quantity of information is related to the probability of a random event. The smaller the probability of something happening, the greater the quantity of information generated, for example, an earthquake in one part of the earth; the greater the probability of something happening, the smaller the quantity of information generated, for example, the sun rises in the east every day.

While the information measure is the information brought about by a specific event that has occurred, the entropy is the expectation of the amount of information that possibly generated before the outcome - considering all possible values of that random variable, i.e., the expectation of the amount of information brought about by all possible events.

The joint entropy measures the uncertainty associated with a set of variables. The joint entropy of two random X and Y variables is defined as: [13]

$$H(X,Y) = -\sum_{x}\sum_{y} p(x,y)\log_2 p(x,y) \tag{3.2}$$

Also, the joint entropy satisfies symmetry, i.e. $H(X,Y) = H(Y,X)$.

The conditional entropy quantifies the amount of information needed to describe the outcome of a random variable Y, given that the value of another random variable X is known. The entropy of Y conditioned on X is defined as:

$$\begin{aligned} H(Y|X) &= \sum_{x\in\mathcal{X}} p(x)H(Y|X=x) \\ &= -\sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x) \\ &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y|x) \\ &= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log p(y|x) \\ &= E_{p(x,y)}\left[-\log_2 p(y|x)\right] \end{aligned} \tag{3.3}$$

The Bayesian rule for conditional entropy is expressed as:

$$H(Y|X) = H(X|Y) - H(X) + H(Y) \tag{3.4}$$

3.2 Mutual Information

Suppose there exists a random variable X , and another random variable Y , then their mutual information is defined as : [13]

$$I(X;Y) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = H(X) - H(X \mid Y) \tag{3.5}$$

Similarly, the MI between two continuous random variable X and Y is defined as :

$$I(X;Y) = \iint_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} = H(X) - H(X \mid Y) \tag{3.6}$$

,where $p(x,y)$ is the joint probability density and,

$$p(x) = \int \mathrm{d}y p(x,y) \quad and \quad p(y) = \int \mathrm{d}x p(x,y) \tag{3.7}$$

are the marginals. [14]

In 1948, Shannon defined and analyzed this metric, and later Mutual Information was created by Robert Fano [15]. MI also affects the difference between the joint distribution of random variables (X, Y) and the product of the bounded distributions of X and Y, as we can find from equations (2.3) and (2.4). MI measures the amount of information obtained about one random variable through observing the other random variable. MI can represent a higher-order relationship among variables.
At the same time, mutual information can be equivalently expressed as: [16]

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X,Y) - H(X|Y) - H(Y|X) \end{aligned} \tag{3.8}$$

The relationship between mutual information and information entropy is shown in the following figure:

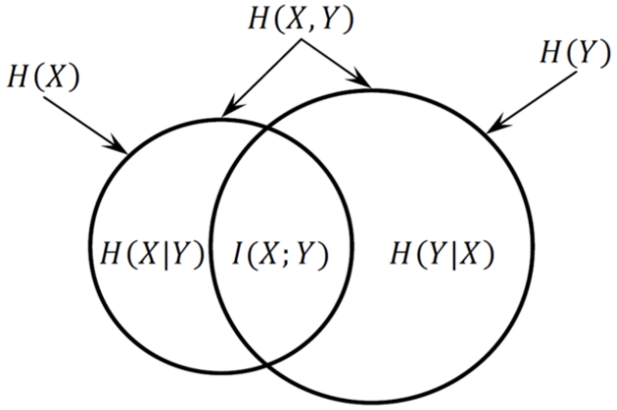


Figure 3.1: The relationship between mutual information and information entropy

The figure shows that the two independent variables X and Y are independent when the mutual information $I(X, Y)$ is equal to zero. Also, the mutual information $I(X, Y)$ must be greater than or equal to $I_{min} = 0$.

3.3 Loss of information

From the theory of mutual information, we know that as X and Y become more correlated, the value of mutual information will become more prominent. A X and Y tend to be independent. The mutual information will become smaller. Therefore, we conjecture that the difference in mutual information will be the amount of information loss concerning information loss. We define information loss as:

$$Loss = I(X;Y) - I(X';Y) \tag{3.9}$$

Under the condition that the mutual information satisfies the symmetry, we take equation (3.8) into (3.9) and get

$$\begin{aligned} Loss &= I(X;Y) - I(X';Y) \\ &= I(Y;X) - I(X';Y) \\ &= H(Y) - H(Y|X) - H(Y) + H(Y|X') \\ &= H(Y|X') - H(Y|X) \end{aligned} \tag{3.10}$$

The simplification gives us equation (3.9) which shows that the information loss we calculate equals the difference in conditional entropy for different X and X' relative to Y. The simplified formula avoids the need to calculate other entropy values, thus making it easier for us to work.

3.4 Monte Carlo method

The Monte Carlo method, also known as the statistical simulation method, is a numerical calculation method guided by statistical probability theory due to the development of science and technology and the invention of electronic computers in the mid-1940s. The Monte Carlo method uses computer-generated samples of a given probability distribution to produce plug-in estimates of specific characteristics of the given distribution. [17, 18]

3.4.1 Monte Carlo approximation

Suppose X is a random variable with a distribution function $F_X(x)$ and suppose that we can generate (through a computer) a sample: [19]

$$\xi_n = [x_1 \dots x_n]$$

of realizations of n random variables $X_1, ..., X_n$ all having distribution function F_X.

Denote by $F_n(x)$ the empirical distribution of the sample ξ_n (i.e.,a probability distribution that assigns probability $\frac{1}{n}$ to each one of the values $x_1, ..., x_n$.) Then, the plug-in estimate is a Monte Carlo approximation :

$$T(F_n)$$

3.4.2 Monte Carlo integration

We can approximate an expectation using Monte Carlo integration, which consists of the average of a random sample of observations drawn from the distribution of X to approximate the expectation of the random variable X: [19]

$$E\left[X\right] = \simeq \frac{1}{n}\sum_{i=1}^{n} x_i$$

This approximation method is called Monte Carlo integration because the expected value being approximated is in fact an integral:

$$E\left[X\right] = \int_{-\infty}^{\infty} x \mathrm{d}F_X(x)$$

where $F_X(x)$ is the distribution function of $\mathcal{X}$. Moreover, if X is a continuous variable, the integral can be written as an ordinary Riemann integral:

$$E\left[X\right] = \int_{-\infty}^{\infty} x f_X(x) \mathrm{d}x$$

where $f_X(x)$ is the probability density function of $\mathcal{X}$.

Chapter 4

Related Work

4.1 Quantifying information loss

From Eric Clarkson's *Quantifying the Loss of Information from Binning List-Mode Data* [20]
This paper proposes quantifying the information loss when list pattern data are effectively binned due to analysis operations. In this paper, the authors show that Fisher information loss is actual and use the Fisher information matrix (FIM) to quantify the loss due to binning. Factors affecting information loss are also derived. The list-based model imaging system applies Poisson statistics. It gives the list's conditional distribution probability (PDF) at a fixed exposure time, i.e., $pr(A|\theta)$, where θ is a p-dimensional parameter vector describing the imaged object. In addition, the authors define the FIM of the list pattern data about *theta* and the binning FIM. The authors discuss the relationship between the two FIMs by replacing the parameter θ with the spatial coordinate function $f(r)$, and finally by weighting the Hilbertspacenorm, after which the difference between the two FIMs is the information loss quantified in the article. The paper demonstrates a loss of Fisher information in any estimation task when the data in list mode is binned. The method is also applicable when the estimation problem is an object reconstruction problem, where a finite-dimensional parameter vector is replaced by a function in an infinite Hilbert space. This method gives us the proposal to calculate the loss by computing the Fisher information matrix. However, replacing the parameters as a function in Hilbert space is not easy. Moreover, the method is proposed under Poisson distribution. However, for our IMU data and Parkinson's disease score data, we do not know which distribution probability the data belongs to, so if we use this method, we need to discuss the distribution of our data first.

From Harikrishnan, K.P. and Misra, R. and Ambika, G's *Quantifying information loss on chaotic attractors through recurrence networks* [21]
This literature presents a measure of entropy and demonstrates its effectiveness in quantifying the loss of structural information of chaotic attractors using measures

of recursive networks. This author proposes a new entropy suitable for analyzing recurrent networks by modifying the fundamental equations of network entropy and calculating the information loss. First understood in terms of information content, the authors determine that entropy can be considered a measure of uncertainty about the information content associated with the network. Information loss is most significant when the network describing a system or process becomes completely homogeneous and vice versa. After defining the probability of information transfer at each node of the network, the authors derive an expression for the information entropy. The method shows that the entropy represents the amount of information required for diffusion on the network and extracts the optimal process by maximizing the entropy. The method discusses the premise that discrete random variables and the probability of each variable are pretty easy to compute. For our IMU data, which are continuous random variables, and Parkinson's ratings, which are discrete random variables, this method requires a step to investigate the relationship between the two random variables.

4.2 Quantifying mutual information and conditional entropy

This work investigates the loss of information between IMU data and Parkinson's disease scores as the sampling frequency decreases. We want to infer the specific information loss by calculating the mutual information or conditional entropy between IMU data and Parkinson's disease scores. The following paper describes the method to calculate the mutual information and conditional entropy. As mentioned in the previous section, the mutual information can be expressed in entropy, so measuring information loss can also be calculated by simplifying the mutual information, as in Equation (3.10).

From Eric Charles Y. Zheng and Yuval Benjamini's *Estimating Mutual Information from Average Classification Error* [22]
Multivariate pattern analysis approaches in neuroimaging are fundamentally concerned with investigating the quantity and type of information processed by various regions of the human brain; typically, estimates of classification accuracy are used to quantify information.
This work combines the features of mutual information $I(X;Y)$, which becomes an excellent method to quantify the information between the random variable X of the neural stimulus and the neuronal signal. Based on supervised learning, a classification up has been established to classify stimulus classes from specific regions of brain activation. However, the accuracy and difficulty of the classification depending on the specific selection of the sample and the number of categories. A popular approach that combines the strengths of the machine learning approach and the advantages of the information-theoretic approach is to obtain a lower bound on the

mutual information by using the confusion matrix. The authors propose a new link classification performance to imply mutual information and build the link in this paper. The authors define a concept of k-class average Bayesian error.
Specifically, the authors establish a relationship between the mutual information $I(X;Y)$ and the average k-level Bayesian error $e_{\text{ABE,k}}$. Briefly, the authors determine a had function π_k (which depends on k):

$$e_{\text{ABE,k}} \approx \pi_k \left(\sqrt{2I(X;Y)} \right) \tag{4.1}$$

The estimator proposed by the authors is:

$$\hat{I}_{HD} = \frac{1}{2}(\pi_k^{-1}(\hat{e}_{gen,\alpha})^2), \tag{4.2}$$

the estimate of the generalization error $\hat{e}_{gen,\alpha}$ is then used to replace $e_{\text{ABE,k}}$.
For the high-dimensional estimator, the authors give the estimator is:

$$\hat{I}_{HD}(M) = \frac{1}{2}(\pi_k^{-1}(\hat{e}_{gen,\alpha}))^2 \tag{4.3}$$

The method's mutual information discriminant estimator can estimate the mutual information of high-dimensional data without assuming full parameters. The method first requires finding a good classifier. Second, the generalization error must be estimated from the test data. If this method is used, we first need to classify the dataset and then estimate the mutual information efficiently by combining downscaling and nonparametric information estimation. For the methods mentioned in the text, all models are identified as "best case," so there is some error. Therefore, one following needs to improve the estimator.

From Alexander Kraskov, Harald Stögbauer, and Peter Grassberger's *Estimating mutual information* [23]
In this paper, the authors propose a mutual information estimator based on k-nearest neighbor (knn) entropy estimation, and in this paper, the authors will always use the natural logarithm, which aims to estimate the mutual information I from the set z_i with uncertain density μ_x, μ_y. The authors introduce the estimation of MI from knn statistics, the estimate for MI is then

$$I^{(1)}(X,Y) = \psi(k) - \langle \psi(n_x+1) + \psi(n_y+1) + \psi(N) \rangle \tag{4.4}$$

The second method, the author replace $n_x(i)$ and $n_y(i)$ by the number of points with $\|x_i - x_j\| \leq \varepsilon_x(i)/2$ and $\|y_i - y_j\| \leq \varepsilon_y(i)/2$. The estimate for MI is then

$$I^{(2)} = \psi(k) - 1/k - \langle \psi(n_x) + \psi(n_y) \rangle + \psi(N) \tag{4.5}$$

Where ψ in formulas (4.8) and (4.9) is the digamma function, digamma function is the logarithmic derivative of the gamma function.

The paper proposes a mutual entropy estimator and yields that a small k reduces the systematic error in terms of statistical and systematic errors, while a large k leads to a minor statistical error. Thus, the choice of a particular estimator depends on the size of the data sample and whether the bias or variance is minimized. This method can minimize the mutual information value to some extent. Still, it cannot estimate the actual value, and the estimator will be more useful in other areas of time series and pattern analysis.

From Weihao Gao, Sreeram Kannan, Sewoong Oh, Pramod Viswanath's *Estimating Mutual Information for Discrete-Continuous Mixtures* [24]
Continuous variables are divided into continuous and discrete variables. When the variables of X and Y do not overlap, the current mutual information $I(X;Y)$ estimator calculates the mutual information according to the 3H principle, i.e., the entropy values of X, Y and the pair (X, Y) are calculated separately before calculating the mutual information. However, calculating the separate entropies is not well defined in the existing hybrid space. This work proposes a new estimator to estimate the mutual information of discrete-continuous random variables. This author gives a mixture of random variables. First, one random variable can be discrete and the other continuous. Second, a random variable with a single scalar can be a mixture of discrete and continuous variables. One of the ideas given by this exciting author is that X and Y can be high-dimensional vectors, and each component can be a discrete, continuous or mixed variable. This author develops new estimators by looking at other estimators under mixed variables.

This author also tested the KSG estimators, namely our estimators (4.4) and (4.5) presented above. This paper proposes estimators for general probability distributions inspired by the KSG estimator. First, the authors give new expressions for mutual information by reviewing the mutual information:

$$I(X;Y) \equiv \int_{\mathcal{X}\times\mathcal{Y}} \log \frac{\mathrm{d}P_{XY}}{\mathrm{d}P_X P_Y} \mathrm{d}P_{XY} \tag{4.6}$$

where $\frac{\mathrm{d}P_{XY}}{\mathrm{d}P_X P_Y}$ is the Radon-Nikodym derivative.

We can note that the mutual information MI is the average of the logarithm of the Radon-Nikodym derivative, so the authors calculated the Radon-Nikodym derivative for each sample i and went to the empirical average. And the authors give a new estimator of the mutual information.

$$\hat{I}(X;Y) \equiv \frac{1}{n}\textstyle\sum_{i=1}^{n} \log \left(\frac{\mathrm{d}P_{XY}}{\mathrm{d}P_X P_Y}\right)_{(x_i,y_i)} \tag{4.7}$$

The basic idea of this estimator is that when X or Y have discrete random variables, we can assert that data point i is one in a discrete component and can use the plug-in estimator for Radon-Nikodym derivatives. When the data points have a joint density, the fixed radius approach proposed by the KSG estimator can estimate the Radon-NIkodym derivative.

The mixed random variable mutual information estimator proposed in this paper effectively addresses the mutual information estimation for variable inconsistencies. For our work on quantifying information loss, typically, acceleration data collected by IMU sensors are continuous random variables, yet the severity of Parkinson's patients' disease is discrete random variables. We must consider the joint probability distribution of random and discrete variables if we rely on mutual information theory to calculate mutual information. The estimator proposed in this article dramatically improves our efficiency in mutual computing information. Based on the KNN algorithm, we also need to consider the choice of k parameters in the KNN algorithm. The choice of parameter k affects the value of the mutual information. On the other hand, we need to explore further extending to high-dimensional data. We should consider the accuracy of this estimator for quantitatively significant and high-dimensional data.

From Alberto Porta, Vlasta Bari's *K-nearest-neighbor conditional entropy approach for the assessment of the short-term complexity of cardiovascular control* [25]

For the complexity analysis of short-term cardiovascular control, the paper proposes a conditional entropy estimation method that does not require the addition of a modification term and does not deal with the lack of reliability of the conditional distribution. The method utilizes the k-nearest-neighbor method to construct the conditional distribution. The method has the advantage of not introducing prior probability information and controlling the loss of the conditional distribution. First, this author is given a stationary time series $y(i)$ of length L, i.e., $y(i)$ is a sample point in the L-dimensional embedding space and is given an expression for the conditional entropy concerning L. The method is estimated according to two strategies, the first one relying on $y|y_L(i-1)$ and the construction of the distribution of y_l, and the second one based on the direct estimation of the conditional probability of y(i) given $y_{L-1}(i-1)$. The article proposes a k-nearest-neighbor conditional entropy estimate, where the authors define all k nearest neighbors of $y_{L-1}(i-1)$ together with theirs to form the conditional distribution $y|y_{L-1}(i-1)$. Then the authors give an expression for the k-nearest-neighbor conditional entropy based on the Shannon entropy calculation. The conditional entropy (CE) estimator is:

$$KNNCE(L) = \frac{1}{N-L+1} \sum_{i=L}^{N} SE(y|y_{L-1}(i-1)) \tag{4.8}$$

,where SE is the Shannon entropy and N is the number of sample points.
This method challenges the problem of estimating conditional entropy. However, this method has a slight peculiarity in requiring many samples. If the number of samples is small, the conditional entropy calculated by this method has a relatively large error. The method quantifies the conditional entropy directly from the data without additional transformation.

Chapter 5

Data set analysis and definition of variables

5.1 Introduction to the data set

We used the MJFF Levadopa dataset. Subjects in this data were furnished with three or eight sensors, with subjects wearing a GeneActiv on the wrist of the most severely affected limb, a Pebble smartwatch on the wrist of the least affected limb, and a Samsung Galaxy Mini smartphone at the waist. These sensors were worn off for the study duration, and subjects recruited from the Boston study site were equipped with five sensors (Shimmer 3), one for each limb and one for the waist. The Shimmer sensors corresponded to the left ankle, right ankle, left wrist, right wrist, and back. [9]

A total of 19 subjects, each subject was collected four days of sensor data. On the first day, they had several motor tasks, including standing, walking in a straight line for the 30s, walking in a straight line for 30s while counting backward, walking up and downstairs, walking through a narrow corridor, finger to nose for 15s (twice per arm), regarding other activities: alternating hand movements for 15s (twice for each arm), drawing, typing on the keyboard for the 30s, opening and pouring bottles (three times), arranging pieces of paper in a folder (twice), assembling nuts and bolts for 30s, folding a towel three times and then sitting. On the second and third day, the test subjects wear the sensors at home and record their usual activities and motor tasks, and finally, on the fourth day, the subjects perform the activities from the first day. For this dataset, we will use Shimmer sensor data and datasets to assess the severity of symptoms of Parkinson's disease. Shimmer sensor data is collected in 3 dimensions, per site (x, y, z), with a total of 5 Shimmer sensors corresponding to 15 different datasets being collected simultaneously. The Shimmer sensor (IMU Sensor) collects IMU data as a variable over four consecutive days. The sensor also records data on the abnormal activity of the test subject's Parkinson's disease over

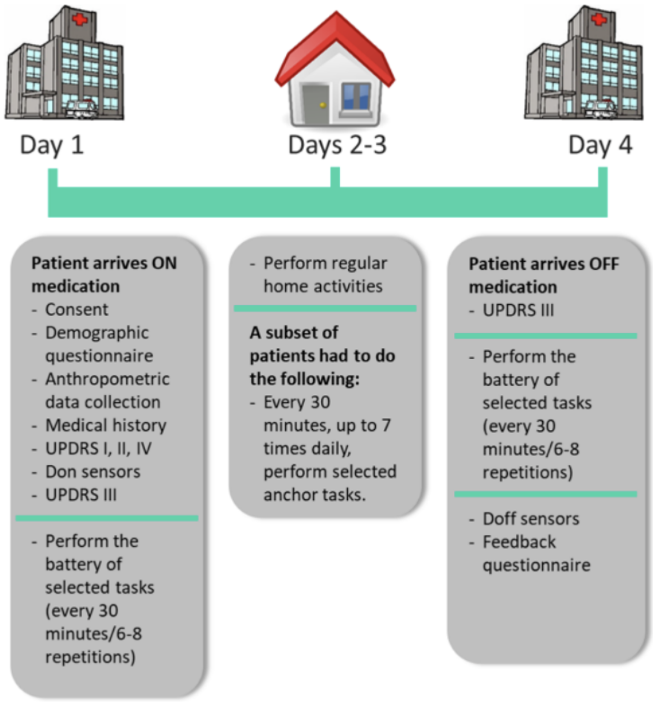

Day 1
Days 2-3
Day 4
Patient arrives ON medication
- Consent
- Demographic questionnaire
- Anthropometric data collection
- Medical history
- UPDRS I, II, IV
- Don sensors
- UPDRS III
- Perform the battery of selected tasks (every 30 minutes/6-8 repetitions)
- Perform regular home activities
A subset of patients had to do the following:
- Every 30 minutes, up to 7 times daily, perform selected anchor tasks.
Patient arrives OFF medication
- UPDRS III
- Perform the battery of selected tasks (every 30 minutes/6-8 repetitions)
- Doff sensors
- Feedback questionnaire

Figure 5.1: The process by which the dataset was collected [9]

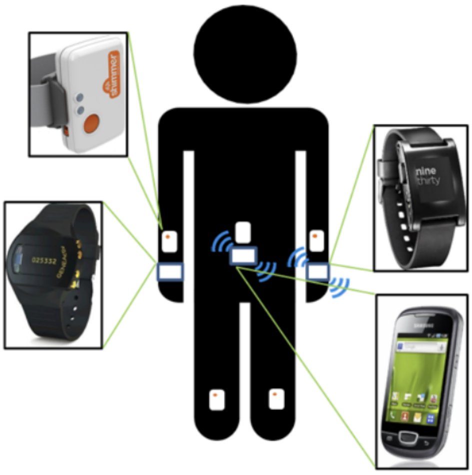
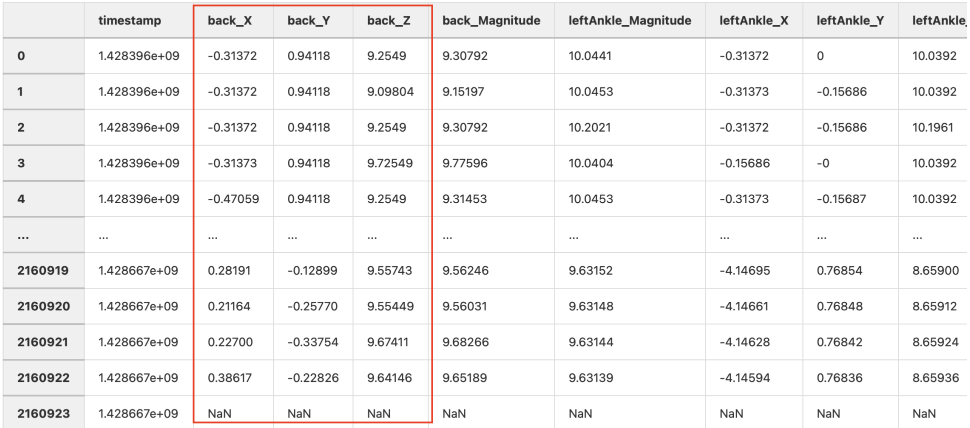


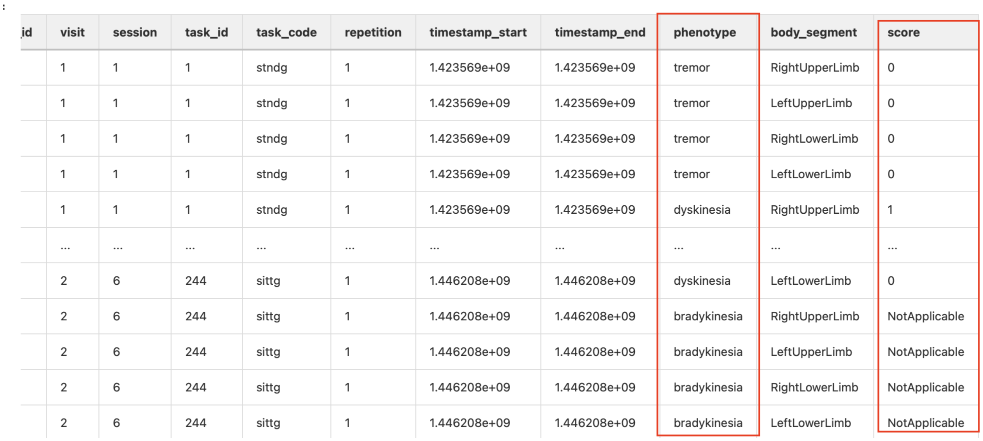
Figure 5.2: About the Sensor Wearing Figure, the Shimmer Sensor is worn on five different parts of the body. [9]

the four days. The accelerations in the X, Y, and Z axes collected by the sensor allow us to determine abnormal activity in Parkinson's patients. For example, when a Parkinson's patient has hand tremors, the sensor worn on the hand can record accelerations that are not consistent with regular activity. The dataset also includes ratings for Parkinson's disease (Bradykinesia, dyskinesia, tremor), each with a 0-4, 0 is no symptom occurrence, and 4 is the most severe occurrence of the symptom. So the IMU data collected for abnormal Parkinson's activity can correspond to the rating for each condition. [9]



	timestamp	back_X	back_Y	back_Z	back_Magnitude	leftAnkle_Magnitude	leftAnkle_X	leftAnkle_Y	leftAnkle_
0	1.428396e+09	-0.31372	0.94118	9.2549	9.30792	10.0441	-0.31372	0	10.0392
1	1.428396e+09	-0.31372	0.94118	9.09804	9.15197	10.0453	-0.31373	-0.15686	10.0392
2	1.428396e+09	-0.31372	0.94118	9.2549	9.30792	10.2021	-0.31372	-0.15686	10.1961
3	1.428396e+09	-0.31373	0.94118	9.72549	9.77596	10.0404	-0.15686	-0	10.0392
4	1.428396e+09	-0.47059	0.94118	9.2549	9.31453	10.0453	-0.31373	-0.15687	10.0392
...	...	...	...	...	...	...	...	...	...
2160919	1.428667e+09	0.28191	-0.12899	9.55743	9.56246	9.63152	-4.14695	0.76854	8.65900
2160920	1.428667e+09	0.21164	-0.25770	9.55449	9.56031	9.63148	-4.14661	0.76848	8.65912
2160921	1.428667e+09	0.22700	-0.33754	9.67411	9.68266	9.63144	-4.14628	0.76842	8.65924
2160922	1.428667e+09	0.38617	-0.22826	9.64146	9.65189	9.63139	-4.14594	0.76836	8.65936
2160923	1.428667e+09	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 5.3: IMU acceleration collected from Shimmer sensors worn by Parkinson's patients [9]



id	visit	session	task_id	task_code	repetition	timestamp_start	timestamp_end	phenotype	body_segment	score
	1	1	1	stndg	1	1.423569e+09	1.423569e+09	tremor	RightUpperLimb	0
	1	1	1	stndg	1	1.423569e+09	1.423569e+09	tremor	LeftUpperLimb	0
	1	1	1	stndg	1	1.423569e+09	1.423569e+09	tremor	RightLowerLimb	0
	1	1	1	stndg	1	1.423569e+09	1.423569e+09	tremor	LeftLowerLimb	0
	1	1	1	stndg	1	1.423569e+09	1.423569e+09	dyskinesia	RightUpperLimb	1
	...	...	...	...	...	...	...	...	...	...
	2	6	244	sittg	1	1.446208e+09	1.446208e+09	dyskinesia	LeftLowerLimb	0
	2	6	244	sittg	1	1.446208e+09	1.446208e+09	bradykinesia	RightUpperLimb	NotApplicable
	2	6	244	sittg	1	1.446208e+09	1.446208e+09	bradykinesia	LeftUpperLimb	NotApplicable
	2	6	244	sittg	1	1.446208e+09	1.446208e+09	bradykinesia	RightLowerLimb	NotApplicable
	2	6	244	sittg	1	1.446208e+09	1.446208e+09	bradykinesia	LeftLowerLimb	NotApplicable

Figure 5.4: The dataset is rated for Parkinson's disease, with each phenotype corresponding to a score. There will also be some non-applicable data. [9]

5.1.1 Visualization Analysis

When visualized, the IMU data helps us understand the data better. Figure 5.5 demonstrates that five different sensors recorded acceleration during dyskinesia, which lasted 32.98 seconds.

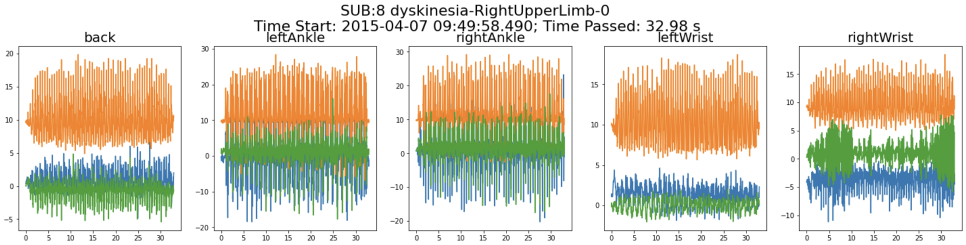


Figure 5.5: Acceleration collected by 5 Shimmer sensors in 32.98 seconds accompanied by dyskinesia [9]

Figure 5.6 shows the percentage of scores for the tremor condition in 34 hours, 52 minutes, and 17 seconds, and we can see that 0 scores account for 72.75% of the scores. The scores of dyskinesia and bradykinesia are also shown. In bradykinesia, the percentage of "none" and "yes" is significant. Therefore, it cannot be used for motor retardation disorders when quantifying information loss. There is also the proportion of simultaneously rated bradykinesia, again with "0" occupying most of the time and rating "4" occupying very little of the time. Most of the time, patients with Parkinson's disease are 'on' state, so the severity of Parkinson's impairment is 0. The ratings '1-4', on the other hand, indicate the severity of Parkinson's impairment from mild to severe.

By visualizing the observed dataset, we first want to associate the IMU data collected by the Shimmer sensor with the Parkinson's disease symptom scores via timestamp mapping, i.e., create a new dataset that includes both IMU data and symptom scores. However, if we use timestamped 1Hz sampling to match the data, there is a significant loss of information. Because the IMU sensor has a sampling rate of 0.02s (50Hz), we map the symptom score scores without changing the sampling rate. From Figure 5.4, we can see the start timestamp and the end timestamp for each Parkinson's disease condition, and this period is typically 30-34 seconds. Theoretically, we have 50 sets of IMU data corresponding to 1Hz, so 30s to 34s corresponds to 1500-1700 points. We have 5 Shimmer sensors corresponding to 3 dimensions of (x, y, z) acceleration data, which means we will get at least 22,500 points under one symptom score. For our 19 patients, each with multiple symptoms, it is challenging to store such a large amount of data. Traversing the patient's data is also tricky due to the high performance of the running memory.

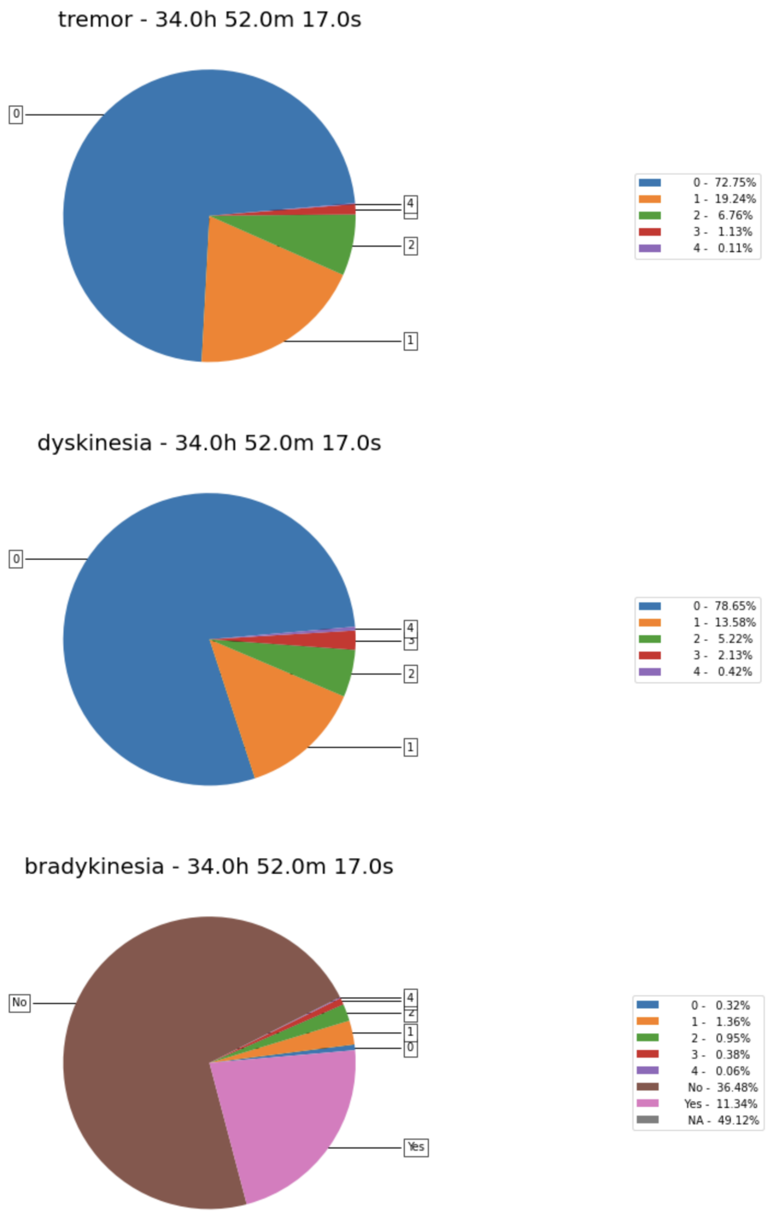


Figure 5.6: Diagram showing the ratings of the three Parkinson's conditions at the same time. [9]

To address the operational difficulties caused by a large amount of data, we should consider not storing all the data, not traversing all the patient data, and creating a new dataset. Therefore, we use a new test set, i.e., for each symptom scoring time, we select 5s of data, i.e., 5 seconds within 34 seconds of Parkinson's disease. Theoretically, we will get at least 3750 points to read the data faster and quantify the information loss.

5.2 Define X and Y in the quantified information

We choose the IMU data as X. Each Shimmer sensor collects acceleration data in 3 dimensions, i.e., (x, y, z). The original data is the IMU data sampled at 50 Hz. We initially obtain the shape of x, y, z as $(N, 250)$, where N is the number of IMU data and 250 is the dimensionality. We use root mean square (RMS) to associate the x, y, and z into a new X. RMS is defined as:

$$M = \sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}} = \sqrt{\frac{x_1^2 + x_2^2 + \cdots x_n^2}{n}} \tag{5.1}$$

RMS is applied to our data:

$$acc_{rms}(x, y, z) = \sqrt{\frac{x^2}{n_x} + \frac{y^2}{n_y} + \frac{z^2}{n_z}} \tag{5.2}$$

We defined the rating data for Parkinson's disease as Y, with each rating representing each category in the classification. The shape of Y is (N,) or (N, 1), where N is the number of points of Y. We need to note that the number N of X and Y should be the same.

5.3 Reduced sampling frequency on IMU data

From the previous subsection, we obtained X with dimension 250. To quantify the information loss after reducing the sampling rate. We will resample the acceleration data distribution of X, i.e., IMU, using 25 Hz, 10 Hz, and 5 Hz to get three new X data. We can see that each frequency corresponds to a time of:

$$\begin{aligned} 50Hz = \quad sampling \quad every \quad 0.02s, \quad 25Hz = \quad sampling \quad every \quad 0.04s \\ 10Hz = \quad sampling \quad every \quad 0.1s, \qquad 5Hz = \quad sampling \quad every \quad 0.2s \end{aligned}$$

The IMU-Shimmer sensor collected our original data set at 0.02s, so we see a data set with an interval of 0.02s per bar. If we resample the data once at an interval, we

are resampling at 25 Hz, and the new data is 0.04s per interval, and so on. Although we resample the data, the data points in our X and Y data sets are the same, and each point's dimension changes.
We can see that the X shape obtained after resampling with reduced sampling frequency is

$$X_{50Hz} : (N, 250) \rightarrow X_{25Hz} : (N, 125) \rightarrow X_{10Hz} : (N, 50) \rightarrow X_{5Hz} : (N, 25)$$

Chapter 6

Quantifying information loss based on K-nearest neighbor (KNN) classification algorithm

6.1 K-nearest neighbor (KNN) algorithm

K-Nearest Neighbor (KNN)[26] is a commonly used supervised learning method that works by a straightforward classification mechanism by measuring the distance between different feature values. [27] KNN is based on the idea that if the majority of the k most similar (i.e., most adjacent) samples in a sample's feature space belong to a particular class, then that sample also belongs to that class. From the Y we defined in the previous section, we can classify each of the 0-4 symptoms of Parkinson's disease into five different categories. Thus, when we use KNN classification, the data from IMU are classified into different categories.
Here is a brief description of the KNN algorithm: [28, 29]

(1) Obtain the distance between the computed and training data. We use the Minkowski distance, which extends the Euclidean distance.The Minkowski distance [30] is defined as:

$$D(X,Y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$$

between two points $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$

(2) Sorting by distance.

(3) Select the K points with the smallest distance.

(4) The categories to which the K points belong are compared, and the test sample points are assigned to the category with the highest percentage of the K points according to the principle of minority rule.

6.2 The choice of k

When classifying the problem point, the k in kNN is the nearest neighbors we select. The choice of k-value is critical because: the larger the k-value, the simpler the model, the larger the bias and the smaller the variance (significant training error, small test error), and the easier the underfitting. If K is equal to N, then it takes all the instances, i.e., it takes the most points under a specific classification in the instances, which has no practical significance for prediction. However, the larger the value of k, the simpler the model, the larger the bias and the smaller the variance (more significant the training error and smaller the testing error), and the easier the underfitting. [31, 32, 33, 34]

We start with k=1 and use the test to estimate the error rate of the classifier, repeat the process, increasing k by one each time, i.e., adding the nearest neighbor, and then select the k with the minimum error rate. It is also worth noting that the value of k is taken as odd as possible to ensure that a more significant number of categories are generated after calculating the results. A similar situation may arise if an even number is taken, which is not conducive to prediction. [35, 32]

6.3 Summary on knn algorithm

The KNN algorithm is a simple and effective classification algorithm and is very easy to implement. However, for an extensive training data set, it is very time-consuming as it needs to calculate the sample distances with test samples and training data set. However, when the data is large, we can present the data in trees, such as kd-tree and ball-tree, and the ball-tree works better for multiple classifications.[36, 37] The KNN classification also has drawbacks, with low prediction accuracy for rare categories when the sample is unbalanced. In the case of sample imbalance, we get significant errors when quantifying information loss. Also, using KD trees, a large amount of memory needs to be built up when using the ball count model. So as the sample data increases, the KNN algorithm requires more time and memory.

6.4 Quantifying information loss after classification based on knn

We recall the formula for information loss (2.9) and the formula for conditional probability (2.7), we can extend the conditional probability, and we obtain:

$$\begin{aligned}
&Loss = H(Y|X') - H(Y|X) \\
&H(Y|X) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log p(y|x) = E_{p(x,y)}\left[-\log_2 p(y|x)\right]
\end{aligned}$$

Where $p(y|x)$ can be given to us by the predicted probability after KNN classification, we can find $p(y_i|x_i)$. Nevertheless, we need to note that we should use the test set of X and Y here to ensure the model's accuracy. After obtaining the probabilities $p(y_i|x_i)$, we will estimate the conditional entropy using the Monte Carlo method.Specifically about the calculation of information loss, as described below: [38]

$$\begin{aligned}
\hat{H}(Y|X) &= E_{p(x,y)}\left[-\log_x p(y|x)\right] \\
&= -\iint p(x,y)log(y|x)dxdy \\
&\approx \frac{1}{N}\sum_{i=1}^{N}(-\log_2 p(y_i|x_i))
\end{aligned} \tag{6.1}$$

Chapter 7

Evaluation

7.1 Experimental Results

In order to avoid the degradation of KNN classification accuracy when the samples are unbalanced, we avoid using data from unbalanced samples in our tests. There is a case where the sample data contains only one category, then the KNN classification is meaningless, and the information loss is 0. After visualizing the data in Section 5, we see that the classification categories under bradykinesia disorders contain 'no,' 'yes,' and 'not applicable. There is a case where the sample data contains only one category, then the KNN classification is meaningless, and the information loss is 0. After visualizing the data in Section 5, we see that the classification categories under bradykinesia disorders contain 'no,' 'yes,' and 'not applicable. We filter the 'Notapplicate' points of bradykinesia and reclassify them, i.e., '0' and 'No' to the new '0' category and '1-4' to a new '1' category. So the classification categories under bradykinesia disorders are '0' and '1'.

A common task in machine learning is learning and predicting algorithms from data. [39] Therefore, we divide the IMU dataset X and the Parkinson's disease score dataset into a test set and a training set. The training set is used to train the model, and the test set is used to do the final evaluation. [40]. When the model is trained, we cannot know how it will perform. We can use the validation set to see how the model performs on new data. The validation set has two purposes, 1. to evaluate the model and tune out the best k for testing KNN classification, and 2. to tune out the hyperparameters so that the model performs best on the validation set. Finally, we use the test set to compute our probability, $p(y|x)$. [41]

We selected four patients with Parkinson's disease, i.e., patients 3, 6, 7, and 9. After getting the dataset, we will divide the dataset into a specific ratio. Our division ratio is 0.25; the test set is 0.25 of the dataset, and the training set is 0.75. We use the training set to build the KNN classification model and the test set to predict our probability $p(y|x)$. After getting the predicted probability $p(y|x)$, we bring in equation (6.1) and calculate the conditional entropy for each frequency. Once we have each conditional entropy, we calculate the difference between the conditional entropies at different sampling frequencies, the information loss.

First, we calculated the information loss on the left wrist of four patients under three conditions with decreasing sampling frequency. where the classification categories of tremor disorders were, '0', '1', '2', '3'. '4'. Dyskinesia disorders were classified in the categories '0', '1', '2', and bradykinesia disorders are classified in categories '0' and '1' as described above.
As in Figure 7.1, the blue, green, and orange lines correspond to the information loss with decreasing sampling frequency for dyskinesia, bradykinesia, and tremor disorders, respectively. We can see that the general trend of information loss is similar. Interestingly, the changes in information loss with decreasing frequency to 25 Hz and 40 Hz are quite close under bradykinesia and tremor disorders. Furthermore, the information loss in dyskinesia disorder has a slightly smaller slope of information loss when the sampling frequency drops to 40 Hz than at 25 Hz. It is noteworthy that the information loss for all three disorders is increased when the sampling frequency decreases from 40 Hz to 45 Hz.

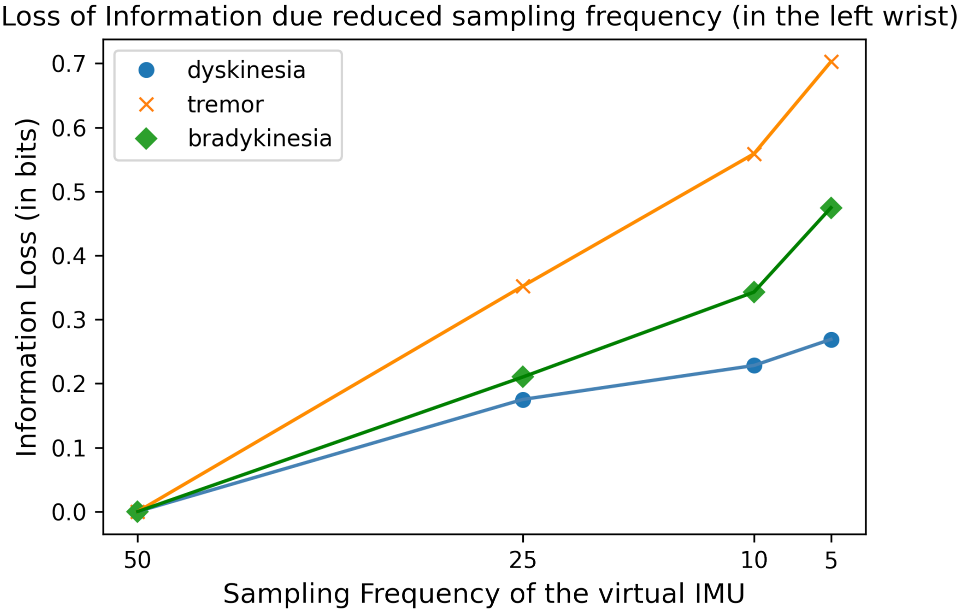


Figure 7.1: Variation in information loss with decreasing sampling frequency in patients 3, 6, 7, 9 left wrists under tremor, dyskinesia, and bradykinesia disorders.

We next proceeded to calculate, for these four patients, the loss of information in the right wrist as the sampling frequency decreased in the three cases. The classification categories for their ratings under the three conditions were the same as for the left wrist.

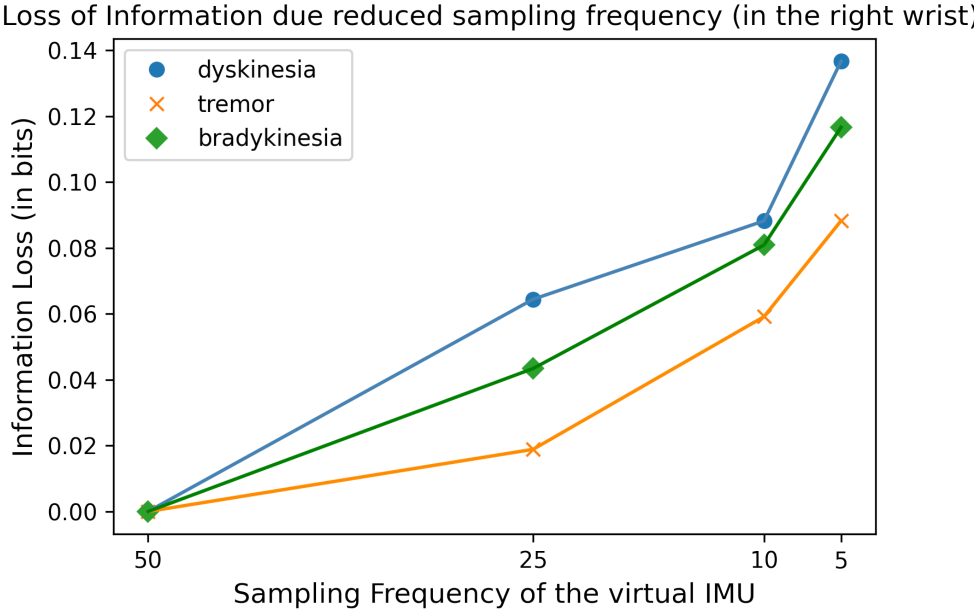


Figure 7.2: Variation in information loss with decreasing sampling frequency in patients 3, 6, 7, 9 right wrists under tremor, dyskinesia, and bradykinesia disorders.

As shown in Figure 7.2, the change in information loss in the patient's right wrist was similar to that in the left wrist in all three conditions, but the information loss in the right wrist was relatively less compared to that in the left wrist.

Finally, we calculated the information loss with decreasing sampling frequency for the left ankle of four patients with tremor and dyskinesia. We did not calculate the information loss for bradykinesia because the data sample was unbalanced. There was no ”4” in the tremor rating classification, which differs from the left and right wrists.

As shown in Figure 7.3, the change in information loss is similar for tremor disorders and dyskinesia disorders. However, it is worth noting that the change in information loss for tremor disorder is relatively close when the sampling frequency drops by 25 Hz and 40 Hz. In contrast, the change in information loss for dyskinesia disorder is more pronounced. It is seen that the information loss change is gradually faster in dyskinesia.

Combining 7 Fig.1 to 7.3, we can see that the information loss changes under all three disorders are similar at different body parts, with relatively more minor information loss at sampling frequencies down to 25 Hz and some variability at sampling frequencies down to 40 Hz and 45 Hz.

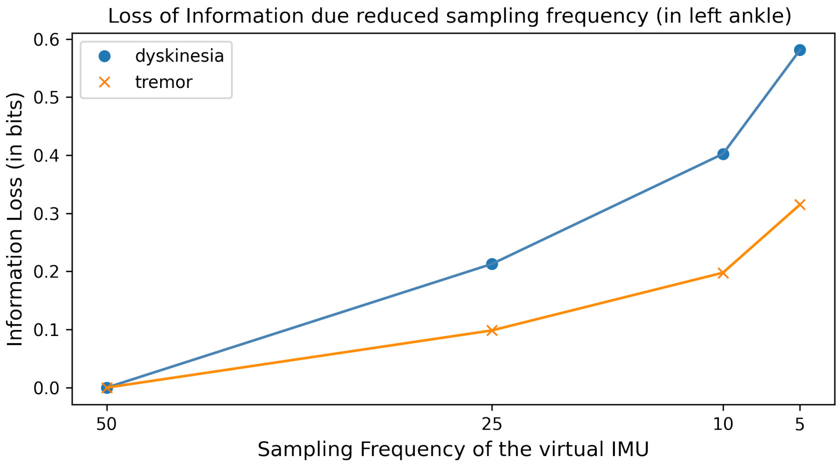
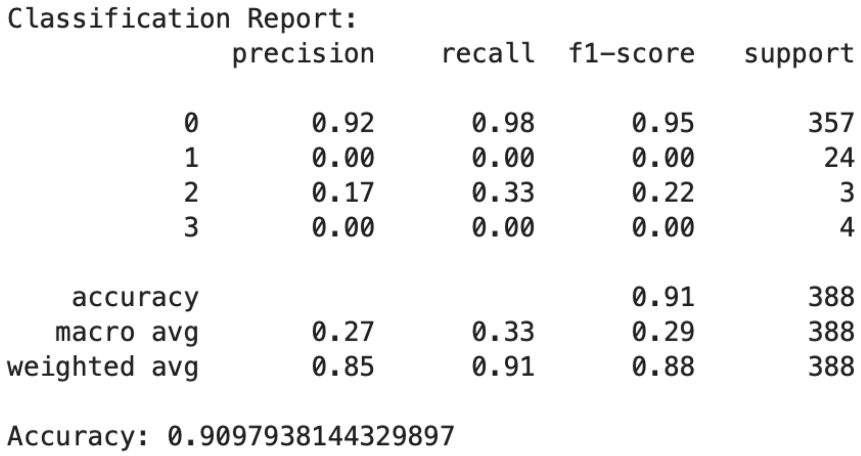


Figure 7.3: Variation in information loss with decreasing sampling frequency in patients 3, 6, 7, 9 left ankle under tremor and dyskinesia disorders.

7.2 Results of unbalanced sample data

In the classification problem with unbalanced data sets, the drawback of KNN is evident, as the minority class is more biased towards the majority class discriminator due to the sample distribution. We can see from the dataset that the severity of Parkinson's disease varies, and typically rating '0' occupies most of the time, while rating '4' does not occupy much time. The sample is unbalanced, increasing our difficulty quantifying the loss of information.

Firstly, we performed KNN classification after classifying the data of patient 9 left wrists under dyskinesia disorder. As shown in Figure 7.4, we have a total of 388 points in our training set, of which class '0': 357 points, class '1': 24 points, class '2' 3 points, and Class '4': 4 points. Such a dataset is unbalanced, and there are far more points in category '0' than in the other categories. kNN classification will incorrectly classify all data points in category 0. Because many categories cannot provide enough training samples, the relatively uniform feature space required by the KNN algorithm cannot be satisfied, making the recognition error larger.
We can calculate the information loss when unbalancing the samples and then compare the line plots of information loss. As shown in Figure 7.5, Information loss for patient9 with left wrist dyskinesia disease decreases as the sampling frequency decreases when unbalanced samples are taken.



```
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.98      0.95       357
           1       0.00      0.00      0.00        24
           2       0.17      0.33      0.22         3
           3       0.00      0.00      0.00         4

    accuracy                           0.91       388
   macro avg       0.27      0.33      0.29       388
weighted avg       0.85      0.91      0.88       388

Accuracy: 0.9097938144329897
```

Figure 7.4: Patient 9 test set is reported under KNN classification, with different categories corresponding to different data points.

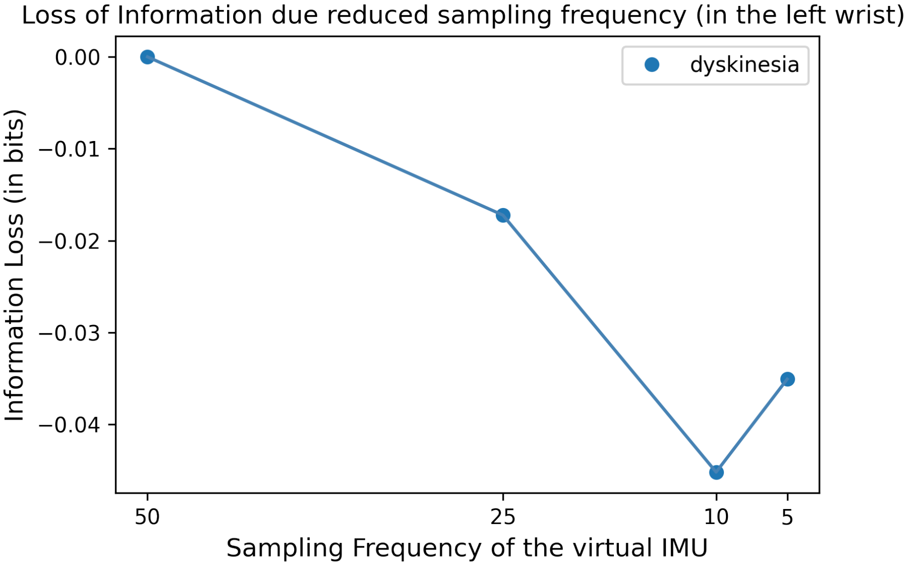


Figure 7.5: Information loss with decreasing sampling frequency of IMU sensors in Parkinson’s patients9 with left wrist dyskinesia

As shown in Figure 7.5, we can see a significant error in the information loss and that the information loss is growing inverse. Instead of increasing the information loss, there is a slight decrease; for example, at 25 Hz, there are -0.017 bits of information loss. Such information loss is negligible, and the reason is that most points are classified as '0' during the KNN classification.

If our samples are unbalanced, the KNN classification model will learn a priori information about the proportion of samples in the training set. There will be many categories with a focus when predicting with practice. (This may result in better accuracy for the majority category and worse for the minority category.) For example, when the sample size of a category is petite, the probabilities calculated after knn classification are the same as those focused on the majority category, which impacts the calculation of our information loss. To overcome the effect of reducing the sample imbalance KNN classification prediction accuracy, we can weight the categories, for example, use smaller weights for categories with large sample data size. [42]
In comparison, we use larger weights for categories with a small sample size. In addition, as the only hyperparameter k of the KNN algorithm, its setting will also significantly impact the algorithm. To reduce the influence of the k value, we can weigh the distance. That is, adding weight to the distance of each point so that the closer points can get a more significant weight. [42]

Similarly, when calculating the information loss, if there is only one type of classification, the information loss is 0 no matter how to reduce the sampling frequency. Hence, KNN classification is meaningless when there is only one classification category.

7.3 The estimated sampling frequency threshold

We analyze the calculated information loss results with the differences between categorical categories, Parkinson's disease differences, and where the patient wears the sensor. We could estimate a threshold, the frequency interval where the information loss is minor when the sampling frequency decreases. Our analysis of the information loss plots for the patients showed a similar trend in information loss. When we reduced the sampling frequency to 25 Hz, there was relatively little information loss. However, the information loss increased when we reduced the frequency to 10 Hz and 5 Hz. Comparing the slopes, we conclude that the slopes are relatively small, decreasing in frequency from the original frequency of 50 Hz to 25 Hz. The slope is relatively large, from the original frequency of 50 Hz to the decreasing frequencies of 10 Hz and 5 Hz. therefore, we should choose a frequency between 25 Hz and 50 Hz to obtain a minor information loss at the decreasing frequency. To get a more accurate sampling frequency, we also need to resample the data further and then calculate the information loss.

Chapter 8

Discussion

Our aim in these works is to quantify the information loss due to the reduced resolution of IMU sensors in PD symptom classification. As mentioned previously, high-precision sensors tend to be expensive. Therefore, we can consider using lower accuracy sensors, i.e., low-cost sensors in daily life if the information loss is slight due to the decreasing resolution of IMU sensors. Parkinson's patients can use low-cost sensors to monitor activities in daily life without using high-precision medical IMU sensors. The previous section quantified the information loss when the sampling frequency decreases and derived thresholds for sampling rates from 25 Hz to 50 Hz. Based on KNN classification, we obtained the classification probability, i.e., $p(y_i|x_i)$. We estimated the joint entropy of the IMU dataset and Parkinson's disease by Monte Carlo estimation. We derived the information loss by calculating the difference between the joint entropy. KNN classification, however, has some advantages and disadvantages.

Advantages of KNN

1. The KNN algorithms are theoretically mature and relatively simple.

2. Compared with the simple Bayesian algorithm, there are no assumptions on the data, high accuracy, and insensitivity to outliers. [43]

3. Since the KNN method relies mainly on the limited number of neighboring samples around, rather than on the discriminative class domain approach to determine the class to which they belong. It is more suitable than other methods for classified sample sets with a large number of classes that cross or overlap.

4. The KNN algorithm is more suitable for automatically classifying class domains with large sample sizes. In contrast, those with small sample sizes are more prone to misclassification using this algorithm.

Disadvantages of KNN

1. Computationally intensive, especially when the number of features is vast.
2. When the samples are unbalanced, KNN classification has low prediction accuracy for rare categories.
3. KD tree, ball tree, and other models need much memory to build.

Combining the advantages and disadvantages of the KNN algorithm, we should first avoid the situation of data sample imbalance in our quantification process. As shown in the above section, when the data sample is imbalanced, the information loss of quantification is challenging to obtain. However, combined with the actual situation, the initial condition of Parkinson's disease patients, often in terms of severity, is not large, so the imbalance of data samples is difficult to avoid in the classification process. On the other hand, the KNN algorithm is very time-consuming in the case of large amounts of data. Our test dataset was calculated only part of the time for patients with Parkinson's disease. If information loss needs to be computed less time, the KNN algorithm requires more extended time and storage space. So when scaling to large amounts of data, we have to consider the time and storage space issues.

In addition to this, we need to consider the advantages and disadvantages of the Monte Carlo Method. [44, 45]

Advantages of the Monte Carlo method

1. The error of the method is independent of the dimensionality of the problem.
2. No discretization is necessary for persistent problems.

Disadvantages of the Monte Carlo method

1. The errors are probability errors.
2. usually requires more computational steps N.

Combining the advantages and disadvantages of the Monte Carlo method, we recall that the shapes of our X and Y are (N; 250, 125, 50, 25) and (N, 1). First of all, our data are high-dimensional, and when using Monte Carlo methods, we can avoid the computational difficulties associated with high-dimensional data. Also, our N-data points are significant in number so that the error can be reduced when using Monte Carlo simulations.

We combine the benefits of KNN's multi-classification and applicability to large samples with Monte Carlo methods, which are not limited to high-dimensional data and do not require discretization of continuous variables. The thresholds derived from our quantification of information loss need to ensure many data points and balance the sample data. For the Parkinson's disease score dataset, if there are categories other than '0-4', the information loss we de-quantified is not applicable. So we need to preprocess the data and clean up the data points if other categories appear in the rating dataset. Alternatively, if fewer categorical categories appear, such as '0' or '1', we can consider using other categorization methods to quantify the information.

Chapter 9

Conclusion

We quantified the information loss accompanying the resolution drop of IMU sensors in Parkinson's disease classification by calculating the difference in conditional entropy between different resolutions after the KNN classification. We resampled the data at sampling frequencies of 25 Hz, 10 Hz, and 5 Hz to simulate low-resolution IMU sensors, respectively, and the results showed that the information loss was less for the 50 Hz drop to 25 Hz and more significant for the 50 Hz drop to 10 and 5 Hz. However, limited by the drawbacks of the KNN classifier, if our dataset is unbalanced, we can increase the weights and then use KNN classification again. We can continue to use different classifiers to quantify information loss in the future. For example, other classifiers can be used only for two sample classification categories. On the other hand, the data of Parkinson's disease patients tend to be large, so KNN classification still has advantages when dealing with large amounts of data. The larger the sample size of the Monte Carlo method, the closer the approximation is to the actual value. Similarly, we can continue to investigate the effect of the choice of k parameters on the information loss in KNN classification. In summary, we conclude that IMU sensors as low as 25 Hz have less information loss and can interpret the results of these works as low-resolution IMU sensors that still work well between 25 and 50 Hz. This could be a promising aspect of using low-cost sensors to study Parkinson's disease.

List of Figures

Bibliography

[1] National Institute of Neurological Disorders and Stroke. Parkinson's disease information page. https://www.ninds.nih.gov/Disorders/All-Disorders/Parkinsons-Disease-Information-Page#disorders-r3, 2016.

[2] Ali Samii, John G Nutt, and Bruce R Ransom. Parkinson's disease. *The Lancet*, 363(9423):1783–1793, 2004.

[3] Tremor. Ursachen symptome. https://www.schoen-klinik.de/tremor, 2020.

[4] Joseph Jankovic. Parkinsonâs disease: clinical features and diagnosis. *Journal of neurology, neurosurgery & psychiatry*, 79(4):368–376, 2008.

[5] Jonathan Smith. Parkinsonâs disease gene therapy clinical trial launched in uk. https://www.labiotech.eu/trends-news/axovant-parkinsons-disease-gene, 2018.

[6] Marco Iosa, Pietro Picerno, Stefano Paolucci, and Giovanni Morone. Wearable inertial sensors for human movement analysis. *Expert review of medical devices*, 13(7):641–659, 2016.

[7] MathWorks. Model imu, gps, and ins/gps. https://de.mathworks.com/help/fusion/gs/model-imu-gps-and-insgps.html.

[8] Desa Hazry, Mohd Sofian, and Mohammad Rosbi. Study of inertial measurement unit sensor. 2009.

[9] MJFF Levodopa Response Study. Mjff levodopa wearable sensors dataset. https://www.synapse.org/#!Synapse:syn20681023/wiki/594678, 2019.

[10] Shimmer Research. Shimmer sensor. https://shimmersensing.com.

[11] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[12] R.K. Pathria and Paul D. Beale. 3 - the canonical ensemble. In R.K. Pathria and Paul D. Beale, editors, *Statistical Mechanics (Third Edition)*, pages 39–90. Academic Press, Boston, third edition edition, 2011.

[13] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[14] Nicholas Carrara and Jesse Ernst. On the estimation of mutual information. *Multidisciplinary Digital Publishing Institute Proceedings*, 33(1):31, 2020.

[15] J. Kreer. A question of terminology. *IRE Transactions on Information Theory*, 3(3):208–208, 1957.

[16] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[17] Herbert L Anderson. Scientific uses of the maniac. *Journal of Statistical Physics*, 43(5):731–748, 1986.

[18] Kurt Binder and Steven M Girvin. The monte carlo method in condensed matter physics. *Physics Today*, 46(6):94, 1993.

[19] Taboga, marco (2021). "the monte carlo method", lectures on probability theory and mathematical statistics. kindle direct publishing. online appendix. https://www.statlect.com/asymptotic-theory/monte-carlo-method.

[20] Eric Clarkson and Meredith Kupinski. Quantifying the loss of information from binning list-mode data. *Journal of the Optical Society of America A*, 37(3):450, Feb 2020.

[21] K.P. Harikrishnan, R. Misra, and G. Ambika. Quantifying information loss on chaotic attractors through recurrence networks. *Physics Letters A*, 383(27):125854, Sep 2019.

[22] Charles Y. Zheng and Yuval Benjamini. Estimating mutual information in high dimensions via classification error, 2016.

[23] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[24] Weihao Gao, Sreeram Kannan, Sewoong Oh, and Pramod Viswanath. Estimating mutual information for discrete-continuous mixtures, 2018.

[25] A Porta, P Castiglioni, Vlasta Bari, T Bassani, A Marchi, A Cividjian, L Quintin, and M Di Rienzo. K-nearest-neighbor conditional entropy approach for the assessment of the short-term complexity of cardiovascular control. *Physiological measurement*, 34(1):17, 2012.

[26] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[27] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[28] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 986–996. Springer, 2003.

[29] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(3):1–19, 2017.

[30] Archana Singh, Avantika Yadav, and Ajay Rana. K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10), 2013.

[31] Laszlo Kozma. k nearest neighbors algorithm (knn). *Helsinki University of Technology*, 32, 2008.

[32] Michael Steinbach and Pang-Ning Tan. knn: k-nearest neighbors. *The top ten algorithms in data mining*, pages 151–162, 2009.

[33] Sabine Landau, Morven Leese, Daniel Stahl, and Brian S Everitt. *Cluster analysis*. John Wiley & Sons, 2011.

[34] Peter Hall, Byeong U Park, and Richard J Samworth. Choice of neighbor order in nearest-neighbor classification. *the Annals of Statistics*, 36(5):2135–2152, 2008.

[35] Zhenyun Deng, Xiaoshu Zhu, Debo Cheng, Ming Zong, and Shichao Zhang. Efficient knn classification algorithm for big data. *Neurocomputing*, 195:143–148, 2016.

[36] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509â517, sep 1975.

[37] Stephen M. Omohundro. Five balltree construction algorithms. Technical report, 1989.

[38] David Luengo, Luca Martino, Mónica Bugallo, Víctor Elvira, and Simo Särkkä. A survey of monte carlo methods for parameter estimation. *EURASIP Journal on Advances in Signal Processing*, 2020(1):1–62, 2020.

[39] Dana Ron. Guest editor's introduction. *Machine Learning*, 30(1):5–6, 1998.

[40] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[41] Yun Xu and Royston Goodacre. On splitting training and validation set: A comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, 2(3):249–262, 2018.

[42] Peter Harrington. *Machine learning in action*. Simon and Schuster, 2012.

[43] Ihsan A. Abu Amra and Ashraf Y. A. Maghari. Students performance prediction using knn and naÃ¯ve bayesian. In *2017 8th International Conference on Information Technology (ICIT)*, pages 909–913, 2017.

[44] David J Earl and Michael W Deem. Monte carlo simulations. In *Molecular modeling of proteins*, pages 25–36. Springer, 2008.

[45] Samik Raychaudhuri. Introduction to monte carlo simulation. In *2008 Winter simulation conference*, pages 91–100. IEEE, 2008.

List of Tables

License