

UNIVERSITY OF CALIFORNIA

Los Angeles

**On Classification with Unreliable Labels for
Environmental and Medical Applications**

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Mohamed Nabil Hassan Hajjchade

2012

© Copyright by
Mohamed Nabil Hassan Hajjchade
2012

The dissertation of Mohamed Nabil Hassan Hajjchade is approved.

Hongquan Xu

Richard Wesel

Lieven Vandenberghe

Gregory J. Pottie, Committee Chair

University of California, Los Angeles

2012

To my family ...

TABLE OF CONTENTS

1	Introduction	1
1.1	Introduction	1
1.1.1	Environmental Monitoring Systems	2
1.1.2	Medical Systems	3
1.2	Contributions and Organization	3
2	Multi-Class SVM for Forestry Classification	6
2.1	Introduction	6
2.2	Support Vector Machines	7
2.2.1	The two-class SVM	8
2.2.2	The multi-class SVM	11
2.3	Haralick Features for texture extraction	12
2.4	Experiments and Results	13
2.5	Conclusion and Future Work	16
3	Monitoring Workspace Activities Using Accelerometers	17
3.1	Introduction	17
3.2	System Architecture	18
3.2.1	System Components	18
3.2.2	Training Data Collection	19
3.2.3	Testing Data Collection	19
3.2.4	Data Analysis	19

3.2.5	Statistical Classification	20
3.2.6	Naive Bayes Classifier	20
3.3	Experiments and Results	20
3.3.1	Training Set Duration	22
3.4	Conclusion	22
4	Feature Selection Based on Mutual Information for Human Activity Recognition	25
4.1	Introduction	25
4.2	Methodology	27
4.2.1	Training Data Collection	27
4.2.2	Features Computation	28
4.2.3	Classification	28
4.2.4	Feature Selection Algorithm	30
4.3	Results	32
4.4	Conclusion	33
5	Stumble Detection using Accclerometers	35
5.1	Introduction	35
5.2	Related Work	37
5.2.1	Fall Detection	37
5.2.2	Gait Analysis	38
5.2.3	Stumble Detection	39
5.3	Stumble Detection Methodology	41

5.3.1	Background and Notation	41
5.3.2	Problem Formulation	42
5.3.3	Feature Extraction	43
5.3.4	Stumbles as Anomalies	45
5.3.5	Estimating $P(f_{l,t_i:t_i+T} S_{t_i:t_i+T} = 0)$	46
5.3.6	Setting the threshold τ_l	47
5.3.7	Choosing T	47
5.4	Experiments	48
5.4.1	Data Collection	48
5.4.2	Limitations of the experiments	50
5.5	Results	51
5.6	Conclusion	54
6	Active Learning for Model Selection	56
6.1	Introduction	56
6.2	The Regression Framework	57
6.3	Sampling Designs for Model Selection	58
6.4	The fundamental model selection problem	59
6.4.1	Related Work	59
6.4.2	Maximum Likelihood Framework	61
6.4.3	Akaike's and Bayes' Information Criteria	63
6.4.4	Evaluation and Discussion	65
6.5	Path Planning	67

6.5.1	Related Work	70
6.5.2	Our Path Planning Algorithm	71
6.6	Conclusion	72
7	Conclusion	74
7.1	Conclusions	74
7.2	Future Work	76
	References	78

LIST OF FIGURES

2.1	Two-class SVM.	8
2.2	Three-class SVM.	11
2.3	The test CIR image. ©IFN.	13
2.4	The classification result. Blue for <i>coniferous</i> , green for <i>leafy</i> and white for <i>other</i>	15
2.5	The current IFN map. ©IFN.	15
3.1	Walking, sitting, and standing classification statistics.	21
3.2	Walking, sitting, and standing classification statistics.	21
3.3	Walking, sitting, and standing classification statistics.	22
3.4	Walking, sitting, and standing classification statistics.	23
3.5	Walking, sitting, and standing classification statistics.	23
3.6	Walking, sitting, and standing classification statistics.	24
4.1	Location of the 14 accelerometers on the body.	27
4.2	Decision tree used.	30
5.1	A 13 seconds window of walking containing a stumble at 6.5, seen from different locations on the body. The x-axis is in seconds, and the y-axis is in g	44

5.2	A 13 seconds acceleration time series. The upper figure shows the raw acceleration, the red axis is at $-1g$ picking up the gravity acceleration. The lower figure show the acceleration after removing gravity by subtracting the decaying average. The x-axis is in seconds, and the y-axis is in g	45
5.3	Obstacle used	49
5.4	Sensor positions	49
5.5	ROC curve. The chest achieves a detection of 99% with a false alarm of 0.5%.	52
5.6	Precision Recall Curve. The chest achieves a precision of 99% precision with a recall 94% (The end point of the chest curve).	53
6.1	Error Rate versus DPNR. The solid lines correspond to the sequential design and the dashed lines correspond to the random design	67
6.2	Error Rate behavior over iterations. The solid line corresponds to a DPNR=4.29dB and the dashed line corresponds to a DPNR=-0.56dB.	68
6.3	Error Rate versus DPNR. The solid lines correspond to the sequential design and the dashed lines correspond to the random design	68
6.4	Error Rate behavior over iterations. The solid line corresponds to a DPNR=4.54dB and the dashed line corresponds to a DPNR=-0.31dB.	69
6.5	Error Rate versus distance traveled. DPNR=1.8dB	72

LIST OF TABLES

4.1	Summary of previous research	26
4.2	The 14 activities that were classified	28
4.3	Features used. (m refers to the magnitude of the 3D acceleration vector.)	29
4.4	Average accuracy for each of the activities. The last	32
4.5	Average accuracy for each of the test subjects. We compare our algo- rithm to an algorithm that selects 14 random features at every internal node. The last column shows the number of sensors used by our algo- rithm.	33
5.1	Features used.	46
5.2	Number of stumbles for each subject.	50

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Professor Greg Pottie for being the nicest advisor ever. Greg provided me with full support for all the years I spent at UCLA, and allowed me to go on 3 internships to France and Australia. Greg introduced me to the data modeling and analysis research, I will always be grateful for that!

I would like to acknowledge the teachings of Professor Lieven Vandenbergh, I'm lucky to have studied optimization with him. I would like to thank him for always being available to meet and discuss optimization problems.

I would like to thank Professor Rick Wesel, and Professor Hongquan Xu for serving on my committee, especially for being patient and accommodating in scheduling the defense.

I would like to thank Professor Bill Kaiser for all the help and inspiration. The discussions I had with you over the years, helped me in refining and developing the ideas of chapter 3, and 4. You are really a great visionary, and I am very lucky to have worked with you.

I would like to also thank Professor Josiane Zerubia for hosting me at the ARIANA group in INRIA, and Professor Fabio Ramos for hosting me at the University of Sydney, and suggesting the problem of Chapter 5. I would like to thank Professor Mark Hansen for all the discussions, and for inspiring the work of Chapter 6.

To my great friend and colleague Laura Balzano, your friendship is inspiring. I thank you for all the discussions we had over the years, for always sharing your insight about research, and all your help during the dissertation writing. You are truly a great person, I am very lucky to have met you!

I feel very lucky to have great colleagues at UCLA, I would like to thank Jay

Chien, James Xu, Celia Xu, Maxim Batalin, Kevin Ni, Supriyo Chakraborty, Cathy Kong, Rahul Balani, Zainul Charbiwala for all the discussions over the years. I would like to also acknowledge the CENS summer internship program, I supervised a total of 10 students over 2 summers, and the interns helped with the data collection, and the implementation of the algorithms. So Thank you to Natali Ruchanski, Claire Lochner, Elizabeth Do, Tremaine Rawls, Benjamin Fish, Ammar Khan, Pinar Ozirik, James Gomes, Travis Rodriguez, And Ascher Friedman. I would like to also thank all the people at CENS, especially Wes Uehara for all the help during the summer internships.

A special acknowledgement goes to all my friends in Lebanon and USA. So Rawia Mecherkany, Spiro Sacre, Mazen Hajj, Tarek Jazzar, Wissam Abdelsamad, Mohamad Karaki, Antonio Challita, Michel Geadah, Oula Hajjar, Raffi Sakabedoyan, Jason Wilson, Sleiman Itani, Rani Husseiki, Ibrahim Khalil, Ramzi Sacre, Mansour Rachid, Rafael Laufer, Shaun Ahmadian, and Roja Bandari, thank you for friendship. Without you, the PhD years (and life in general) would have been much harder. A special thank you to Shaun and Antonio for helping in organizing my thoughts, and always motivating me. I hope I didn't forget anyone, and If I did, blame the mind not the heart.

Last but not least, my family! You are simply the best. Thank you for all the laughter, and all the love. To my parents Hassan and Alia, thank you for always believing in me. To my sister Manar, thank you for all the love. To my brother Samer, thank you for all the inspiration. To my sister Hiba, thank you for all the support. To my brother and best friend Bassel, thank you for all the inspiration and always making things look easy. Your music was a great companion during the solitary days of dissertation writing. To Mohamad, Lynn, Asia, and Fline: I love you!

VITA

1982	Born, Chhim, Lebanon.
1985–2000	Lebanese Bacculaureate, College Universel, Chhim, Lebanon.
2000–2004	B.E. (Computer and Communications Engineering) and minor (Physics), American University of Beirut, Lebanon
2003–2003	Research Intern, German Aerospace Center (DLR), Germany
2004–2007	M.S. (Electrical Engineering), UCLA, Los Angeles, California.
Aug. 2008	PhD qualifier exam (Electrical Engineering), UCLA, Los Angeles, California.
2008–2008	Research Intern, ARIANA project, Institut National de Recherche en Informatique et Automatique (INRIA), France
2009–2009	Research Intern, ARIANA project, Institut National de Recherche en Informatique et Automatique (INRIA), France
2011–2011	Research Intern, Australian Center of Field Robotics (ACFR), University of Sydney, Australia

PUBLICATIONS

1. B. Fish, A. Khan, N. Hajj Chehade, C. Chien and G. Pottie "Feature Selection

- Based on Mutual Information for Human Activity Classification,” *In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.
2. R. Balani, N. Hajj Chehade, S. Chakraborty, M. Srivastava ”Distributed Coordination for Fast Iterative Optimization in Wireless Sensor/Actuator Networks,” *In Proceedings of the IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2011.
 3. N. Ruchansky, C. Lochner, E. Do, T. Rawls, N. Hajj Chehade, C. Chien, G. Pottie, W. Kaiser ”Monitoring Workspace Activities using Accelerometers,” *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.
 4. N. Hajj Chehade, J. G. Boureau, C. Vidal, J. Zerubia ”Multi-Class SVM for Forestry Classification,” *In Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2009.
 5. K. Ni, N. Ramanathan, N. Hajj Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, M. Hansen, M. Srivastava, D. Estrin, G. Pottie ”Sensor Networks Data Fault Types,” *In the Transactions on Sensor Networks (TOSN)*, 2009.

ABSTRACT OF THE DISSERTATION

**On Classification with Unreliable Labels for
Environmental and Medical Applications**

by

Mohamed Nabil Hassan Hajjchade

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2012

Professor Gregory J. Pottie, Chair

Information technology is undergoing yet another revolution, dubbed the data revolution. The recent advancements in sensing technology, storage systems, computational systems, and mathematical tools are enabling the realization of systems that can observe the world at a low cost, store large amounts of data, and run complex algorithms efficiently to process these large datasets. A key component in the design of such systems is the validation process where we need to evaluate the system on datasets representative of real life. In this dissertation, we consider environmental and medical classification problems where the validation process is challenging due to the difficulty of collecting class labels and ground truth.

We divide our work into three parts. In the first part, we present a system for tree type classification using satellite or aerial images. The system is used to update the current forest maps of the National French Forest Inventory (IFN).

In the second part, we present three motion recognition systems using wearable accelerometers designed for healthcare and medical applications. The first system is designed to monitor the workplace activities and study the seated posture habits of the user. The second one is designed to recognize the activity of the user from a set

of 14 common daily activities. The third system is designed for stumble detection in analyzing the gait of the user, and studying the effect of frequent stumbles on the risk of falling. We also present two large datasets collected for the validation of the systems.

In the third part, we present a novel algorithm to collect data that optimizes the model selection in the maximum likelihood framework, for linear regression models used in spatial process estimation.

CHAPTER 1

Introduction

1.1 Introduction

With the beginning of the 21st century, information technology is witnessing yet another revolution: the data revolution. This revolution is due to the recent advancements in low cost sensing technology, storage systems, computational and processing systems, and mathematical tools. We are now able to observe the world at a low cost, store large amounts of data, and run complex algorithms efficiently to process these data. This revolution is affecting many different fields: astronomy, medicine, environmental science, entertainment, telecommunications, social science, economics, etc. We are now able to observe large-scale phenomena, and analyze them to improve our quality of life on our planet.

A general data-based system has many components:

- Phenomenon component: the phenomenon of interest that the system is sensing and analyzing. This component include the prior information we have about the signal or data model.
- Sensor component: the data acquisition system which includes the sensor model, as well as the cost, and the power required for the sensing technology.
- Storage component: the system could be analyzing a large-scale spatial and temporal phenomenon, and needs a large storage system for the data acquired.

- Processing and modeling component: the set of mathematical methods used to model and process the data acquired, in order to make inferences about the phenomenon of interest.
- Validation component: the set of experiments needed to validate the system. This includes producing the ground truth about the data acquired through these experiments. Producing the ground truth information could be very challenging depending on the system and desired application.

In this dissertation we focus on classification systems for environmental and medical applications. In these two fields, the validation component of the system is very challenging.

1.1.1 Environmental Monitoring Systems

Studying the environment is now possible by observing processes over a large scale both in space and time. It is now possible to embed the physical world with wireless sensor networks, and use them to monitor many aspects of the environment. In chapter 2, we study the problem of creating forest maps using satellite and aerial images. Forest maps are currently constructed manually, and the problem is clearly in the efficiency and scalability of the manual forest mapping. We designed an automated mapping method using support vector machines and texture analysis of the forest images. The challenge in this system is in the validation component, since it is very hard to create the ground truth and measure the performance of the system. In our solution, we seek the help of an expert on forest maps to create labels for a training set, and to evaluate our results.

1.1.2 Medical Systems

The medical and healthcare field is undergoing a technology revolution, and a new field called mobile health (or wireless health) is emerging. Doctors will soon be able to monitor their patients wirelessly and continuously from their communities. This will have a huge impact and reduce the costs of healthcare tremendously. In the work of this dissertation, we consider medical and healthcare applications that are based on the body motion, e.g. motion rehabilitation after an injury or stroke, fall prevention, athletic training, etc. Specifically we build systems for continuous activity monitoring using low-cost wearable sensors [htt]. The challenge for these systems is again the validation component, where we need to evaluate our systems on users in their daily life and community. This is very challenging, and the challenge is due to the difficulty of producing labels and ground truth for the experiments.

1.2 Contributions and Organization

We make several contributions and we list them below:

- We provide an algorithm for tree type classification using satellite or aerial color-infra-red imaging, to update the French forest maps. We validate our method on a real image provided by the French forest inventory (IFN).
- We provide a system to monitor the workplace activities, and the seated posture habits of the user.
- We provide a motion recognition system for a set of 14 common daily life activities.
- We provide a system for stumble detection in analyzing the gait of the user, and studying the effect of frequent stumbles on the risk of falling.

- We provide a large data set of 14 activities from 8 subjects instrumented with 14 accelerometers on different locations of the body.
- We provide a large data set of realistic stumbles during walking, for 9 subjects. The data set contains a total 100 stumbles and 45 minutes of walking with reliable labels.
- We provide a novel algorithm to collect data that optimizes the model selection in the maximum likelihood framework, for linear regression models used in spatial process estimation. We also provide a path planning version of the algorithm.

This dissertation is structured as follows. In chapter 2, we present a method for classifying the vegetation types in an aerial Color Infra-Red (CIR) image. Different vegetation types do not only differ in color, but also in texture. We study the use of four Haralick features (energy, contrast, entropy, homogeneity) for texture analysis, and then perform the classification using the One-Against-All (OAA) multi-class Support Vector Machine (SVM), which is a popular supervised learning technique for classification. The choice of features (along with their corresponding parameters), the choice of the training set, and the choice of the SVM kernel highly affect the performance of the classification. The study was done on several CIR aerial images provided by the French National Forest Inventory (IFN). We will show one example on a national forest near Sedan (in France), and compare our result with the IFN map.

In chapter 3, we describe a physical activity classification system using a body sensor network (BSN) consisting of low cost tri-axial accelerometers. We focus on workspace activities (different motions and sitting postures). We use a Naive Bayes classifier and show that we can train the system simply and systematically. For each task, we find a set of features that separate the corresponding activities.

In chapter 4, we consider a classification problem of 14 physical activities using a body sensor network (BSN) consisting of 14 tri-axial accelerometers. We use a tree-based classifier, and develop a feature selection algorithm based on mutual information to find the relevant features at every internal node of the tree. We evaluate our algorithm on 31 features per accelerometer (total of 434), and we present the results on 8 subjects with a 96% average accuracy.

In chapter 5, we develop a system for the detection of stumbles with an accelerometer system. Our system consists of low cost triaxial accelerometers that may be worn by patients and are convenient for a wide range of subjects. We use machine learning and data mining techniques to detect and count the stumbles in the acceleration data. We also validate our system with a large data set collected from 9 subjects. The data set contains a total of 100 stumbles and 45 minutes of walking.

In chapter 6, we consider the problem of estimating a spatial environmental process using a sensor network. Finding strategies for sensor placement and data collection to optimize the estimation is a fundamental task for these applications. We present optimization formulations for obtaining good strategies. The formulations are based on minimizing the probability of error in selecting the correct model. The locations for this formulation are found in a sequential algorithm that provides a strategy for sensor placement and data collection. We focus on linear fields, and we provide simulations that describe these solutions and show their benefits.

Finally, in chapter 7 we present our conclusions and plans for future research.

CHAPTER 2

Multi-Class SVM for Forestry Classification

This chapter presents collaborative work with Jean-Guy Bourean and Claude Vidal of the Inventaire Forestier National (IFN) in France, and Professor Josiane Zerubia in the ARIANA group of the Institut National de Recherche en Informatique et Automatique (INRIA), in France. This work was first published as a paper in the IEEE International Conference on Image Processing (ICIP), 2009 [HBV09].

2.1 Introduction

Forest management has a very broad social, environmental and economical impact in our societies. It is largely needed for balancing the society needs for forest products, and the preservation of the society, the environment, the vegetation, and the animal health. Forest maps (type, density, age, etc.) are fundamental components for this task.

With the recent advances in space and computer technologies, the combination of remote sensing and image processing became a very powerful tool for developing these maps. Acquiring global and local forest images became easier, and processing these images for information extraction became faster.

SVM has been widely applied in the classification of remote sensing images. In [ZDZ07], SVM is used to detect burnt areas in satellite images, in a two-class classification setting. In [FCB06], multi-class SVM is used for the classification of hyperspec-

tral remote sensing data. The classification was done using the image spectral bands, without any texture analysis. In [KMS07], tree species classification was done in high resolution CIR aerial images, where the tree crown shape feature was very useful. The images that we used, were of a lower resolution making the shape feature uninformative, and the focus on texture features. In [MB04], multi-class SVM is shown to be a valid alternative to other common classification methods for the classification of hyperspectral remote sensing images.

In this chapter, we present a method to classify tree types in an aerial Color Infra-Red (CIR) image. The method consists of extracting relevant image features that enhance the distinguishability of the tree types, and then performing the classification in the feature space. The Haralick features were chosen for the feature extraction, and the Support Vector Machine (SVM) for the classification. The image that we used was provided by the French National Forest Inventory (IFN). It contains two types of trees, which makes the problem a three-class classification one. This research was mainly motivated by the need of IFN to improve their forestry maps. Fig.2.5, taken from the IFN registry available on [ifn], is the IFN map for the test image (Fig.2.3), where it is clear that the map does not show enough details.

2.2 Support Vector Machines

The support vector machine (SVM) is one of the most popular methods for classification. It has been used in many domains and applied to different classification problems [ZDZ07], [WMB07], [Bur98]. SVM falls in the category of supervised learning techniques, and consists of finding the optimal separating hyperplanes between the classification classes, based on training data. The training data are representational data from each class. The task is, then, to find the optimal hyperplanes separating the training data, and classifying a new datum depending on where it falls with respect to the

hyperplanes. The popularity of SVM is due to its small generalization error, and its low computational complexity as it is formulated as a convex program. In this section we will briefly review the theory behind SVM.

2.2.1 The two-class SVM

The basic version of SVM deals with two linearly separable classes. Given two linearly separable classes of data, there is an infinite number of linear hyperplanes to separate them. SVM finds the optimal one, which is the hyperplane that results in the maximal separation margin. The attractive feature of SVM is that it can be rigorously formulated as a convex problem [Vap95].

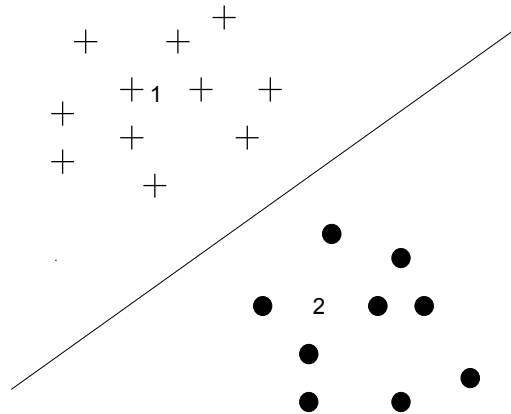


Figure 2.1: Two-class SVM.

Suppose the training data are x_i in \mathbf{R}^d with corresponding labels y_i in $\{-1, +1\}$, where $i = 1, \dots, N$. A linear classifier consists of finding a linear hyperplane separating the positive labeled data ($y_i = +1$), from the negative labeled data ($y_i = -1$). Mathematically, we seek to find a linear function $f(x) = w \cdot x + b$, where w and b are

the function parameters, that classifies the points, i.e. [BV04]

$$y_i \cdot f(x_i) \geq +1 \quad (2.1)$$

The inequality (2.1) defines the feasibility condition for a linear hyperplane to separate the two classes of the training data. Any w and b that satisfy (2.1) define a feasible separating hyperplane. The resulting separation margin between the two classes can be found to be $2/\|w\|$, where $\|w\|$ denotes the vector norm of w [Bur98].

SVM consists of finding the separating hyperplane, (2.1), that maximizes the separation margin, $2/\|w\|$, between the two classes. This can be formulated as:

$$\begin{aligned} \min_{w,b} \quad & \|w\|^2/2 \\ \text{subject to} \quad & y_i(w \cdot x_i + b) \geq +1 \end{aligned} \quad (2.2)$$

which is a convex problem. The Lagrange dual of this problem can be found to be [Bur98]:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=0}^N \alpha_i - 1/2 \sum_{i=0}^N \sum_{j=0}^N \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{subject to} \quad & \sum_{i=0}^N \alpha_i y_i = 0, \alpha_i \geq 0 \end{aligned} \quad (2.3)$$

where α_i , the Lagrangian multipliers, are the optimization variables. The dual problem (2.3) is also convex, and is easier to solve than (2.2), since the constraints are linear. The parameters w and b are found from the solution of (2.3) [Bur98]; $w = \sum_{i=0}^N \alpha_i y_i x_i$, and $b = y_i - w \cdot x_i$ by choosing any i such that $\alpha_i \neq 0$. The classification function is thus:

$$f(x) = \sum_{i=0}^N \alpha_i y_i x_i \cdot x_j + b \quad (2.4)$$

When the data are not linearly separable, i.e. the set of inequalities (2.1) is unfeasible for any w and b , we can relax the feasibility constraints by introducing nonnegative slack variables ξ_i , to allow some misclassification: $y_i \cdot f(x_i) \geq +1 - \xi_i$. Moreover, we

can consider the tradeoff between the number of misclassified points and the width of the separation margin, by introducing a regularization term to the optimization problem:

$$\min_{w,b} \quad \|w\|^2/2 + C \sum_{i=0}^N \xi_i \quad (2.5)$$

$$\text{subject to} \quad y_i(w \cdot x_i + b) \geq +1 - \xi_i, \xi_i \geq 0$$

The dual problem becomes

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=0}^N \alpha_i - 1/2 \sum_{i=0}^N \sum_{j=0}^N \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{subject to} \quad & \sum_{i=0}^N \alpha_i y_i = 0, 0 \leq \alpha_i \leq C \end{aligned} \quad (2.6)$$

We can as well seek a nonlinear function f , that separates the two classes, i.e. satisfying (2.1). Boser et al. [BGV92] showed that we can use Aizerman's kernel trick to generalize the above methods to nonlinear functions (or nonlinear separating hyperplanes). The idea is to map the original non-linearly separable data into a higher-dimensional Euclidean space H , where the linear classifier can be used. We will call the mapping $\phi : \mathbf{R}^d \mapsto H$. This makes a linear classification in the new space H equivalent to a non-linear classification in the original space \mathbf{R}^d . This generalization is simple since the classification function depends only on the dot product between the data ($x_i \cdot x_j$), and it consists of simply replacing every dot product $x_i \cdot x_j$ by a dot product in H , $\phi(x_i) \cdot \phi(x_j)$.

We would like to avoid the expensive computation of $\phi(x_i) \cdot \phi(x_j)$ due to the high-dimensionality of H . We can accomplish this by using Mercer's theorem, which finds a kernel function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ [Vap95]. Now we can only use K without the need to know ϕ . The non-linear classification function becomes:

$$f(x) = \sum_{i=0}^N \alpha_i y_i K(x_i, x_j) + b \quad (2.7)$$

2.2.2 The multi-class SVM

The most popular approach to build a multi-class SVM, is to combine multiple binary SVMs. In this work we use the One-Against-All (OAA) scheme, which consists of building one SVM for each class [HL02]. Given M classes of data, we construct M binary SVMs where the i th SVM separates class i (labeled positive) from the rest $M - 1$ classes (labeled negative), where $i = 1, \dots, M$. The resulting M classification functions, $f_i(x)$, are then combined by a *max* rule to form one global classification function. Given the labeling assumed, a data point x is assigned to the class corresponding to the largest classification function:

$$Class(x) = \arg \max_i f_i(x) \quad (2.8)$$

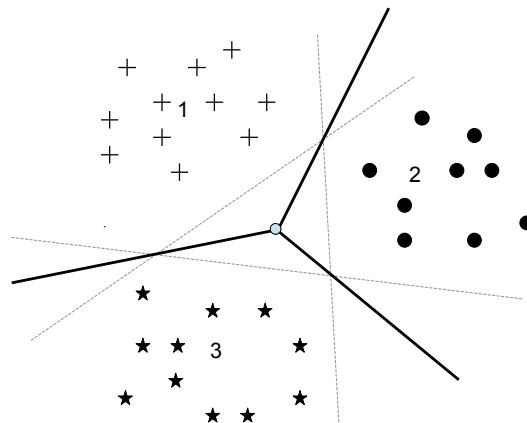


Figure 2.2: Three-class SVM.

2.3 Haralick Features for texture extraction

Different types of trees do not only differ in color but also in texture. For texture extraction, we investigated the use of the popular Haralick features [Har79], which are based on the gray level co-occurrence matrix (GLCM). The GLCM represents the distribution of repeated values at a given offset $(\Delta x, \Delta y)$ ¹, in an image (or a window of an image) [Har79].

Mathematically, for an $A \times B$ window, I , of an image with G gray levels, the GLCM C is a $G \times G$ matrix defined by:

$$C(i, j) = \sum_{i=0}^A \sum_{j=0}^B \begin{cases} 1, & I(i, j) = i, I(i + \Delta x, j + \Delta y) = j \\ 0, & \text{otherwise} \end{cases} \quad (2.9)$$

To get a normalized distribution p , we normalize C

$$p(i, j) = \frac{C}{\sum_{i=0}^G \sum_{j=0}^G C(i, j)} \quad (2.10)$$

Haralick defined 13 features in [Har79]. In this work, we chose to study four of them (energy, contrast, entropy, homogeneity):

$$\text{Energy} = \sum_{i=0}^G \sum_{j=0}^G p^2(i, j) \quad (2.11)$$

$$\text{Contrast} = \sum_{i=0}^G \sum_{j=0}^G (i - j)^2 p(i, j) \quad (2.12)$$

$$\text{Entropy} = - \sum_{i=0}^G \sum_{j=0}^G p(i, j) \log p(i, j) \quad (2.13)$$

$$\text{Homogeneity} = \sum_{i=0}^G \sum_{j=0}^G \frac{p(i, j)}{1 + |i - j|} \quad (2.14)$$

In addition to the Haralick features, we studied the use of the mean, the standard deviation and the normalized difference vegetation index ($NDVI$). The $NDVI =$

¹ Δx is in the horizontal direction, and Δy in the vertical direction

$\frac{(NIR-red)}{(NIR+red)}$, is computed from the spectral values of each pixel, and reflects the vegetation density for the pixel [ZDZ07]. Finally, we chose G to be 32.

2.4 Experiments and Results

In this section, we discuss the choices we made for the different parameters of the system, and we show the results obtained on one color infrared (CIR) aerial image, provided by the French National Forest Inventory (IFN) (Figure 2.3). The image has three spectral bands (NIR, red and green), and contains two types of trees (coniferous, and leafy). Our goal is to classify each pixel, to check if it belongs to the first type of trees, the second type, or neither. Thus we have a three-class classification problem, where we call the classes *coniferous*, *leafy*, and *other*. The *other* class contains the non-vegetation, the very young regeneration, and the shadows.

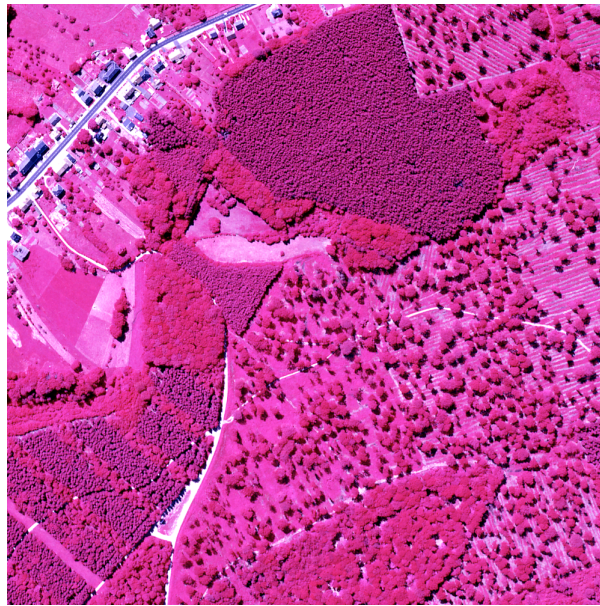


Figure 2.3: The test CIR image. ©IFN.

As mentioned above, we have a total of 7 features, which we compute for each pixel

of the image. All of them, except the *NDVI*, are computed over a window centered at the pixel. We tested different window sizes: 5x5, 7x7, 11x11, 15x15, 17x17, 21x21 and 25x25, and we chose the 17x17 as it showed better distinguishability of textures than the other sizes. A lot of work could be done for more rigorous window size selection, but this was out of the scope of the current work. We have plans to study it more carefully in the future.

After the study of the 7 features, we decided to only use the *NDVI*, the mean, the standard deviation, and the entropy. The other features did not seem to discriminate well between the different classes. In addition to that, we used the spectral bands (*NIR*, red, and green). In summary, vectors of the form (*NDVI*, *NIR*, red, green, mean, standard deviation, entropy) were the inputs to the multi-class SVM.

For the SVM, we used the Gaussian kernel $K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$. The choice of the SVM parameters (σ , and C) was done by running a series of simulations over a range of values, on the training set. We chose the ones that resulted in the best classification of the data set. We did not have a ground truth for evaluating the classification, and the evaluation was done visually. The parameters will work well for the same type of images with the same resolution. We tested the same parameters on different images of the same type of trees, but with different resolutions and the results were not so good. The parameters needed to be changed to get a better classification on the other image.

The classification result is shown in Figure 2.4, where blue represents *coniferous*, green represents *leafy*, and white represents *other*. This result was obtained for $\sigma = 0.1$, and $C = 1$. This result shows an improvement over the IFN map shown in Fig. 2.5. The closed forests were very well classified; the three classes were clearly distinguished even in the overlapped regions. In the open forests, the classification problems were due on the one hand to the shadow, and on the other to the fact that the

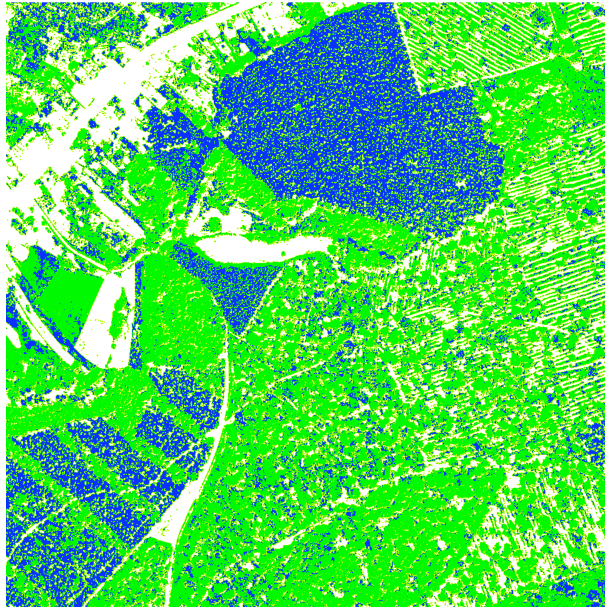


Figure 2.4: The classification result. Blue for *coniferous*, green for *leafy* and white for *other*.



Figure 2.5: The current IFN map. ©IFN.

young regeneration has a similar reflectance and texture as the trees. However, in the open forest our result shows a noticeable difference between the regions where the regeneration was cleared (look at the thin white lines in Fig. 2.3), and the ones where the clearing was not done. This differentiation is very important for forest management. We are working on removing the artifacts and making the classification smoother.

2.5 Conclusion and Future Work

In this chapter, we presented a method for tree type classification based on texture features and support vector machines. We used the Haralick features for texture extraction, and the one-against-all multi-class support vector machine for the classification. The result was encouraging for more investigation of the methods used, as it shows a big improvement over the IFN map. The goal of the chapter was to show that the technique we used, works well for the type of images and the desired objective of IFN. A parameter learning phase is sought to be done in the future. Other directions are automatic feature selection, and automatic training set selection.

CHAPTER 3

Monitoring Workspace Activities Using Accelerometers

This chapter presents collaborative work with PhD student Chieh Chien in the electrical engineering department at UCLA, Professors Gregory Pottie and William Kaiser in the electrical engineering department at UCLA, and four summer interns in the Center of Embedded Networked Sensing (CENS) at UCLA Natali Ruchansky, Claire Lochner, Elizabeth Do and Tremaine Rawls. This work was first published at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2011 [RLD11].

3.1 Introduction

Physical activity monitoring (PAM) systems comprised of on-the-body accelerometers are effective tools for monitoring physical activity with medical, athletic training, and general health applications. For this work we used a PAM system for monitoring people at their workplace. Accelerometer systems have already proven their effectiveness for the physical activity classification necessary for health monitoring, successfully classifying basic physical activities including walking, jogging, and going upstairs and downstairs [LCB05a]. In this chapter we demonstrate that our PAM system is capable of utilizing a personalized training set easily acquired by the user in a clinical setting. We also investigated the effect of training set duration on overall classification accuracy. Previous research has used pressure sensors embedded in a worker's chair for

seated posture classification [TSP01] [MP03], and video surveillance has been used to classify standing, sitting and lying down postures [CGP05]. However, there has been little research concentrated on workspace activity and seated posture classification with accelerometers. These sensors are less costly for mass production than chairs equipped with embedded sensors and less invasive than video surveillance. Workplace activities that are of interest for classification in our system include walking, standing, sitting, sitting posture, and seated movements such as shaking one's leg or twisting in a chair. Studies have shown that too much sedentary behavior, such as sitting at a computer during the work day, is detrimental to health so much so that it leads to increased risks of cardiovascular disease, despite regular participation in moderate to intense exercise [BGG94]. It has also been shown that bad workplace posture can result in widespread physical pain [Rei93]. Fidgeting and restlessness, such as shaking ones leg and twisting in a chair, are both symptoms of anxiety, and thus the monitoring of such activities could potentially provide insight into the stress level of workers [BHG91] [BGG94]. With the tri-axial accelerometers used in this work, employers and employees will be able to monitor exactly how much time they are spending in sedentary positions, whether or not they have proper posture when sitting, and to what extent they are exhibiting anxious physical behaviors. This data will be able to be used as a guideline for altering their behaviors for the preservation and improvement of their health.

3.2 System Architecture

3.2.1 System Components

Our system consists of tri-axial wireless and wearable Gulf Coast Data Concept X6-2mini accelerometer sensors. The sensors continually collect data in the X, Y, and Z

directions once removed from a USB port, at a selected rate of 160Hz.

3.2.2 Training Data Collection

Sensors are placed on the user in specified locations (knees, chest). When collecting a training set, an easily recognizable signature (such as jumping or leaning back and forth five times) is performed before each activity. Each activity is performed for the same amount of time and its occurrence and duration is recorded, facilitating the ease of labeling the data. The orientation and location of each sensor is also recorded in order to remain consistent between data collection sessions. The workspace activities performed are walking, standing, sitting, sitting with the back reclined 85-95 degrees, sitting with the back reclined 115-125 degrees, sitting while slouching, shaking one leg while sitting, crossing one leg while sitting, swiveling in a chair, and sliding in a chair away from and towards a desk.

3.2.3 Testing Data Collection

Sensors are placed in the same locations and with the same orientations as they are in training data collection. The same activities are performed but in a natural, unplanned manner. An observer (or the user) records the activities that the user performs so that the data can be labeled appropriately, creating ground truth for classification. In a deployed system, this recording would not be necessary as we would rely on the classifications, presuming they yield sufficiently accurate statistics.

3.2.4 Data Analysis

For the classification, we used a Naive Bayes classifier [Lew98] over a feature space. The features were calculated over a window of 4 seconds. The features were: mean,

maximum value, and frequency energy. This was done on several levels first by classifying sitting, walking, and standing, and then within sitting, classifying the postures and movements.

3.2.5 Statistical Classification

Statistical classification refers to the problem of identifying the category (or class), an object of interest (or data point) belongs to. The object could be a document, an image, a pixel, a time window of acceleration data, etc. This is done by building a rule, called a classifier, that maps data points to categories or classes. A classifier takes the data point as an input, and outputs a *label*.

3.2.6 Naive Bayes Classifier

The naive Bayes classifier is one of the most simple and most used classifiers. It is a supervised probabilistic classifier, that models each class with a Gaussian probability distribution. Given a set $S = \{C_1, C_2, \dots, C_M\}$ of M classes, the naive Bayes classifier determines the probabilities that a new observation D_i belongs to each of class C_i in the set S .

3.3 Experiments and Results

The activities classification performed were structured into three levels. The first classification level comprised of walking, sitting, and standing, the second of seated posture, and the third of seated movements: twisting in the chair, shaking the leg, sliding to and from the desk, and crossing legs. From these data the frequency, max, min, mean, and standard deviation values were extracted on the three levels of classification. Splitting the classification into several levels has advantages for (1) producing a model that

		Classification			
		Unknown	Walk	Stand	Sit
True Class	Unknown	5.2632	40.3509	15.7895	38.5965
	Walk	0	100	0	0
	Stand	0	1.0929	98.9071	0
	Sit	0	0.4651	0	99.5349

Figure 3.1: Walking, sitting, and standing classification statistics.

		Classification			
		Unknown	Proper Reclined	Proper Upright	Improper
True Class	Unknown	0.5464	25.1366	59.5628	14.7541
	Proper Reclined	0	98.4962	1.5038	0
	Proper Upright	0	0	100	0
	Improper	0	0	0	100

Figure 3.2: Walking, sitting, and standing classification statistics.

is physically understandable, and thus leading to information that is more useful in advising subjects on how to change their behavior (e.g., degree of slouch, take more breaks, etc.), and (2) reducing training time, since each decision is low-dimensional. We chose the Naive Bayes classifier [Lew98] because it is simple and it worked well. On the first level (walking, sitting, and standing) an accuracy of 99.5% was achieved as seen in Figure 6.1. For the posture, where the angle of the back was measured, the accuracy was 99.6%, as seen in Figure 6.3. On this level, two types of proper posture [Lab] were classified; reclined at 120 degrees, and upright at 105 degrees and many types of slouching were classified as improper. For the various movements while sitting, an accuracy of 96.5% was obtained, as seen in shown in Figure 3.3.

		Classification				
		Unknown	Twist	Shake	Cross	Slide
True Class	Unknown	0.6342	1.0571	20.9302	67.6533	9.7252
	Twist	0	100	0	0	0
	Shake	0	0	97.3684	0	2.6316
	Cross	0	0	0	92.6829	7.3171
	Slide	0	2.7778	0	0	97.2222

Figure 3.3: Walking, sitting, and standing classification statistics.

3.3.1 Training Set Duration

An integral part of our PAM system is the training data set, the set of data that the classifier bases its classifications on. In order to determine the optimum amount of time for which each activity should be performed in the training data set, a single set of data was broken up into subsets of time intervals. One time interval was dedicated as the testing data set, while the other time interval's length was varied and dedicated as the training data sets. The same testing data set was tested against each of these training data sets, and the overall accuracy for each training data set length was recorded. For walking, standing, and sitting classification, it was found that 2 minutes was an adequate training interval, as seen in Figure 3.4. For seated posture classification it was found to be 2 minutes, as seen in Figure 3.5. For seated motions it was found to be 15 seconds, as seen in Figure 3.6.

3.4 Conclusion

In this chapter we present a system that can accurately classify daily life activities in the work place. The systematic and simple method of training that has been developed is key. The procedures developed and results obtained allow, with research on basic activities as a foundation, for the monitoring of work place physical activity. Subjects

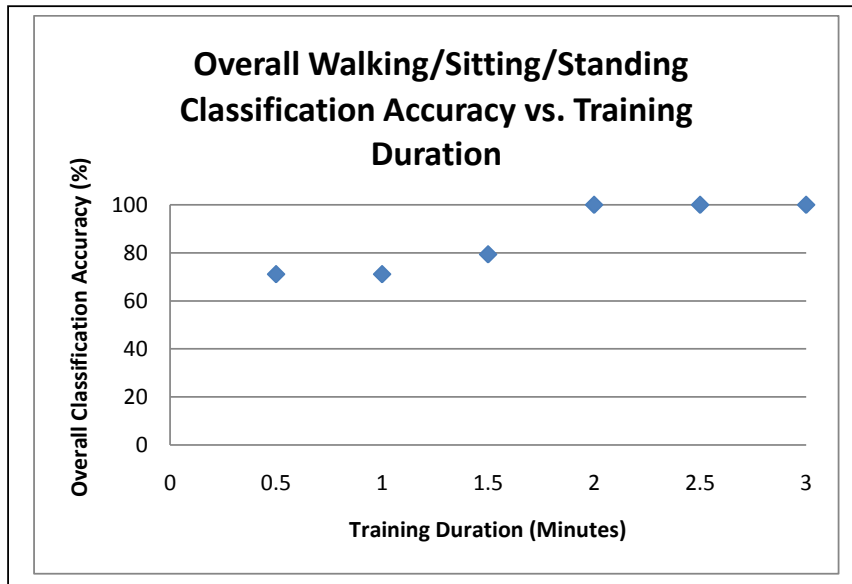


Figure 3.4: Walking, sitting, and standing classification statistics.

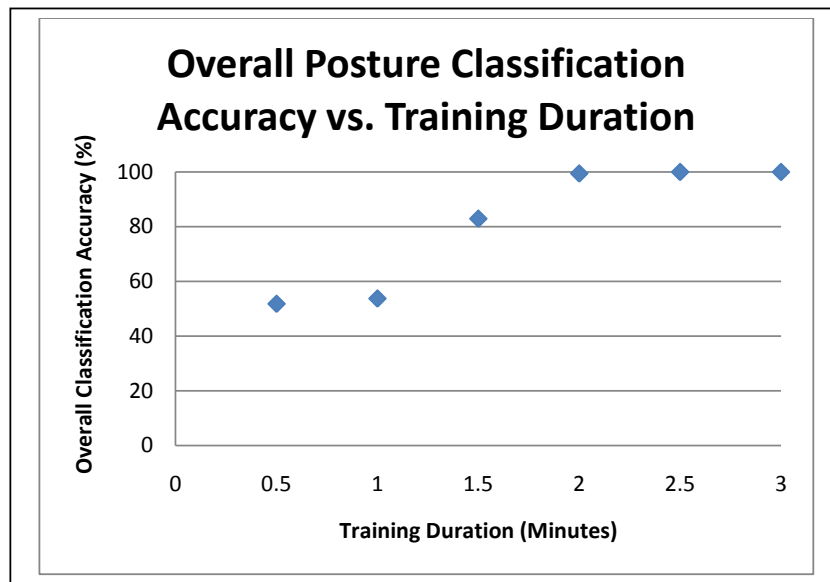


Figure 3.5: Walking, sitting, and standing classification statistics.

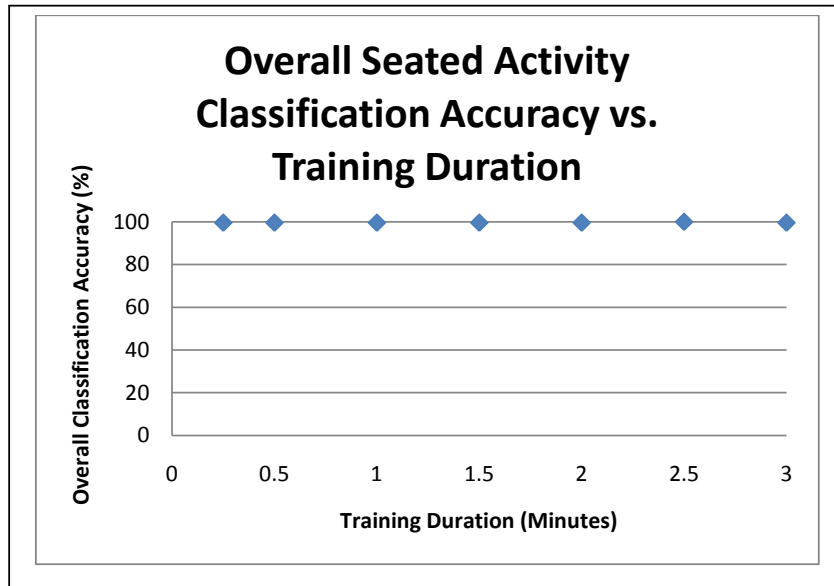


Figure 3.6: Walking, sitting, and standing classification statistics.

who have expressed pain or discomfort in their body would be enabled with such a system to track their daily movement and posture without any disruption to their daily life. Further work in two main areas is desired, expanding on posture and feature selection. We have only studied the posture of the back; whether the subject is sitting at a proper angle or slouching. However, there are many aspects to posture and proper physical motion in the workplace, such as how long subjects stare at a computer monitor, the angle of the knees, whether the subject's feet are flat on the floor, the level of the arm rest, etc. With more data collected in these areas, the applications of the system would be even greater. Another interest is in feature selection. There are many possible features to compute, and many possible sensor positions. Further analysis on the best features and combination of features is needed. There are also more feature selection algorithms that could be investigated, such as giving features weights that reflect their ability to differentiate between activities. In the future, we would like to test this system on various subjects for longer periods of time.

CHAPTER 4

Feature Selection Based on Mutual Information for Human Activity Recognition

This chapter presents collaborative work with PhD student Chieh Chien in the electrical engineering department at UCLA, Professor Gregory Pottie in the electrical engineering department at UCLA, and two summer interns in the Center of Embedded Networked Sensing (CENS) at UCLA Benjamin Fish and Ammar Khan. This work will be published at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2012 [RLD11].

4.1 Introduction

Human activity recognition is central to many fields such as neurological rehabilitation [WWD07], context-aware computing, and athletic training [IMC06]. For example, in neurological rehabilitation, doctors are interested in monitoring their stroke patients' activities at home and in the community. Traditional methods of motion monitoring are based on tedious manual techniques such as self-monitoring or constant monitoring by an observer. These techniques are prone to error due to forgetfulness and other kinds of misreporting. Recent advances in low-power and compact sensor technology made the automation of activity monitoring, using a body sensor network, feasible and low cost. In [LCB05b], multi-modal sensor systems were used to classify basic phys-

Ref.	No. Activities	No. Sensors	No. Subjects	Accuracy
[BI04]	20	5	20	84%
[ZCL11]	5	1	10	85%
[ALK10]	5 groups	6	11	N/A
[ARB99]	5	2	5	89 %
[LCB06]	8	7	12	90%
[KSS]	8	12	1	65% - 95%
[JWC08]	8	1	7	95%
[LCB05b]	10	7	2	95%
Ours	14	14	8	96 %

Table 4.1: Summary of previous research

ical activities, including walking, jogging, and going up and down stairs. In [BI04] and [RLD11], sensor systems using only accelerometers were used for activity classification; [BI04] used biaxial accelerometers to monitor both ambulatory and sedentary motions, while [RLD11] used tri-axial accelerometers to monitor workspace activities. Smart phone-based accelerometers were also used for activity recognition as in [ZCL11]. A representative sampling of previous research is presented in Table (4.1).

In our work, we aim at capturing the motions of all the parts of the body for a thorough study of the activity recognition problem. We over-instrument the subjects with 14 tri-axial accelerometers placed on various parts of the body, and we consider the classification of 14 common daily activities. We take a supervised learning approach, using a binary decision-tree with a naïve Bayes classifier at every internal node and a large feature set of 31 features per accelerometer (total of 434 features). This is a high-dimensional problem where brute force is not possible, and a feature selection algorithm is needed to find the best features for every naïve Bayes classifier (present at every internal node). Feature selection is a problem that has been studied many times before in other contexts. Different types include margin-based algorithms such

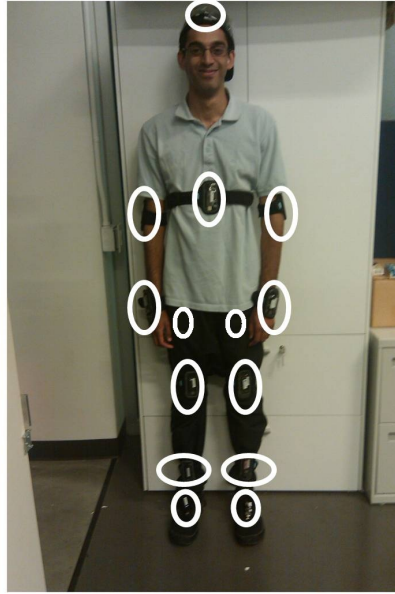


Figure 4.1: Location of the 14 accelerometers on the body.

as RELIEF [KR92] and mutual information-based algorithms such as MIFS [Bat94]. We use a mutual information-based algorithm because it is computationally capable of handling the large amount of data captured by 14 accelerometers. Our contribution is that we describe an activity classification system that can handle a large set of activities (14 activities) representative of the common daily activities, and achieves a very high recognition accuracy. The system was tested on 8 different subjects.

4.2 Methodology

4.2.1 Training Data Collection

Accelerometers are placed on an individual at fourteen locations, as shown in figure 4.1. The accelerometers we used were tri-axial wireless Gulf Coast Data Concept X6-2mini accelerometers ($\pm 6g$) [htt], which continually collected data at a rate of 160Hz. Fourteen different activities are performed, as described in table 4.2. To collect labels

for ground truth, we used an Android phone application. The application has a list of the activities to choose from and a start/stop button to record the time the subject started the activity, and the time he/she stopped. Eight different data sets were collected from eight different healthy individuals for a length of five minutes per activity.

Active	Stationary
Slow walk	Stand
Fast walk	Sit (upright)
Walk (up-slope)	Sit(hunch)
Walk (down-slope)	Sit (slouch)
Walk (up stairs)	Lie down (on back)
Walk (down stairs)	Lie down (on stomach)
Run	Lie down (on side)

Table 4.2: The 14 activities that were classified

4.2.2 Features Computation

Features were computed on 4 second windows of acceleration data with 3 second overlapping between consecutive windows. We compute 31 different features for each sensor, shown in table 4.3. Since we used 14 different sensors, this meant a total of 434 features from which to choose.

4.2.3 Classification

We used the binary decision tree shown in figure 4.2, with a naïve Bayes classifier at each node. The naïve Bayes classifier is a probabilistic method given by the function

Features
Standard deviation of x,y,z axes and m
Mean of x,y,z axes and m
Absolute mean of x,y,z axes and m
Energy ratio of x,y,z axes and m
Ratio of DC to sidelobe of x,y,z axes
First sidelobe location of x,y,z axes
Max value of x,y,z axes and m
Short time energy in x,y,z axes and m
Correlation between x and y axes

Table 4.3: Features used. (m refers to the magnitude of the 3D acceleration vector.)

(4.1), where \mathcal{C} is the set of classes and \mathcal{F} is the set of features.

$$\max_{C \in \mathcal{C}} \{p(C) \prod_{f \in \mathcal{F}} p(f|C)\} \quad (4.1)$$

This classification was performed offline. A tree was used so that the classifier would not have to distinguish between all 14 of the classes using the same set of features. Instead, classifiers are used to partition the data into smaller and smaller categories of classes until the categories consist of a single class, at which point the data is fully classified. In the probability calculations (given by Bayes' rule), the features were assumed to be independent with a Gaussian distribution, as required by the naïve Bayes classifier. For every subject, the naïve Bayes classifiers (at the internal nodes of the tree) were trained on his/her training data; this is often called user-dependent procedure. The feature selection was also personalized to every subject.

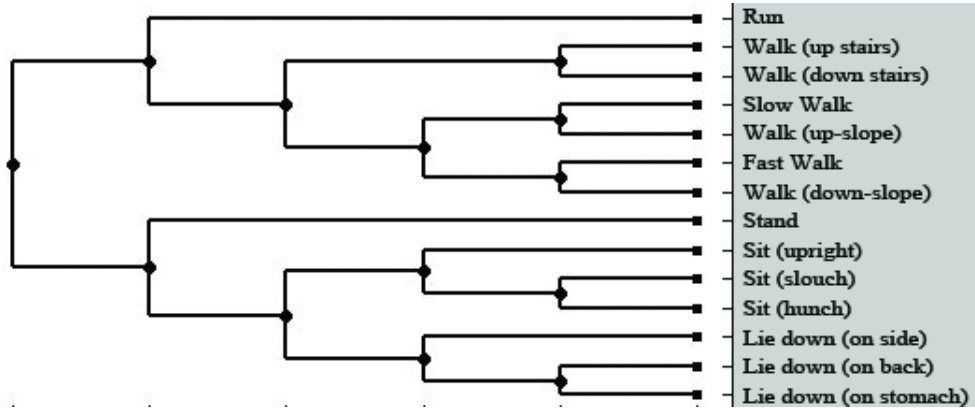


Figure 4.2: Decision tree used.

4.2.4 Feature Selection Algorithm

The high-dimensionality of the problem requires a good feature selection algorithm to find the best features for the naïve Bayes classifier at every internal node. In order to minimize computational complexity while maximizing accuracy, this algorithm employs a filter solution first, and then a wrapper. The algorithm works as follows:

1. We determine the Gaussianity of each feature by calculating the negentropy of each feature given each class using the approximation given in equation (4.2), where J is the negentropy, x is a random variable, E is the expected value, and $kurt$ is the kurtosis, the fourth central moment of the distribution [RE01] [CT06].

$$J(x) \approx \frac{1}{12}E\{x^3\}^2 + \frac{1}{48}kurt(x)^2 \quad (4.2)$$

We remove all features with negentropy values that are higher than an a priori threshold.

2. Using the mRMR algorithm, we ranked the features. [PLD05] The term this

algorithm wishes to maximize is given by formula (4.3).

$$I(C; f_i) - \frac{1}{|S|} \sum_{f_s \in S} I(f_s; f_i) \quad (4.3)$$

I is the mutual information, C is the class variable, f_i is the feature under consideration, and S is the set of features already selected and ranked. Calculating mutual information requires calculating the entropy of a feature or set of features, a computationally expensive process because each feature is a mixture of Gaussians. Hence a Taylor series approximation of the entropy was employed [HBD08].

3. By now, there are a few parameters that can be changed: the threshold for the negentropy values and the degree of the Taylor series approximation. In addition, there are really two different possible algorithms, using only the first term of (4.3) (Max-Relevance), or both (Max-Relevance and Min-Redundancy) [PLD05]. Instead of choosing one algorithm, or just one set of parameters, a range of parameters is used over both algorithms, and the sets of features returned by these algorithms are captured. Because we wish to minimize the number of features, we use the first k features in each ranking, where k ranges from 1 to the full set.
4. This gives us a list of feature sets. We pick the feature set that minimizes the training error¹.
5. The above steps are repeated for each node in the tree. Then for each node, the highest ranking set of features are chosen, and the total number of sensors used so far is updated.

¹This corresponds to choosing the feature set that gives the highest discrimination between the two branches of the tree at the corresponding node. Training error is the percentage of misclassified training data

Activity	Percent Correct
Run	100%
Walk (up stairs)	97.67 %
Walk (down stairs)	94.54 %
Slow Walk	92.77 %
Walk (up-slope)	95.95 %
Fast Walk	96.81 %
Walk (down-slope)	95.32 %
Stand	99.41 %
Sit (upright)	89.90 %
Sit (slouch)	94.62 %
Sit (hunch)	99.24 %
Lie down (on side)	100 %
Lie down (on back)	94.83 %
Lie down (on stomach)	99.66 %

Table 4.4: Average accuracy for each of the activities. The last

4.3 Results

We collected sets of data from eight different individuals where the participant did five minutes of each of the 14 activities while wearing all 14 accelerometers. For every subject, we build a personalized classifier; we trained on half the data, 2.5 minutes per activity, and tested on the other half, a time suggested by [RLD11]. We got an average overall accuracy of 96.5%, as seen in table (4.5). These results show that a high activity recognition rate is achievable for a large set of activities. Table (4.5) shows that a large number of sensors was used for every subject. This is due to the fact that the feature selection algorithm does not take into consideration from which sensor the features were selected. It would be interesting to change the feature selection algorithm to a

Subject	Random Features	Our algorithm	No. of Sensors
1	51 %	93 %	10
2	64 %	98 %	12
3	71 %	96 %	9
4	86 %	97 %	10
5	70 %	98 %	12
6	83 %	99 %	10
7	83 %	96 %	13
8	74 %	95 %	11
Average	72.75 %	96.5 %	10.875

Table 4.5: Average accuracy for each of the test subjects. We compare our algorithm to an algorithm that selects 14 random features at every internal node. The last column shows the number of sensors used by our algorithm.

sensor selection algorithm, while maintaining a relatively high accuracy. This could be done by adding a term to favor features from the same sensors. It is also worth noting that for different subjects, different features were selected. This is due to the variation in the acceleration data belonging to different subjects (e.g. different subjects walk differently, sit differently, and lie differently.).

4.4 Conclusion

This chapter presents a combination of a tree-based classification and a feature selection algorithm for human activity recognition, and shows that a high activity recognition rate is achievable for a large set of common daily activities. More than just a specific algorithm, this paper presents a framework that maximizes the accuracy that can be garnered from the results of specific algorithms, like the mRMR algorithm that we used. This work shows that different sensors (at different locations on the body) are

the best for discriminating between subset of the activities. The algorithm presented could be changed to minimize for the number of sensor used. This is a step forward towards understanding human activities and towards finding the best placements of sensors on the body for the recognition of a large set of activities.

CHAPTER 5

Stumble Detection using Accclerometers

This chapter presents collaborative work with Professor Fabio Ramos in the Australian center of field robotics at the University of Sydney, Professor Gregory Pottie in the electrical engineering department at UCLA, and two summer interns in the Center of Embedded Networked Sensing (CENS) at UCLA, Pinar Osirik and James Gomez.

5.1 Introduction

Falls pose a major health problem in the elderly population, and they account for a significant portion of their injury and death. Injuries resulting from falls can be not only physically, but mentally detrimental [NFR07], resulting in the reduction or loss of one's independence [GLS96]. A recent study by the center for disease control and prevention (CDC) shows that one in every three adults (age 65 and older) falls each year. For example, in 2008 more than 19,700 adults (65 or older) died from unintentional fall injuries. In 2009, 2.2 million injuries were treated in emergency departments, and more than 581,000 were hospitalized. The injuries included: hip fractures, spine fractures, leg fractures, and head traumas. In the same year more than \$19 billion were spent on fall related injuries ¹.

Due to these reasons, understanding falls and their precursors became a major topic in medicine and public health. In fact, in 2009 the institute of medicine (IOM) listed

¹<http://www.cdc.gov/HomeandRecreationalSafety/Falls/adultfalls.html>

research on falls and their prevention among older adults, in their first quartile of priorities [REW09]. The current methods of fall prevention and predictions consist of exercise and balance training. But as engineers, we can imagine a system of a body sensor network, monitoring individuals, assessing the risk of falls, and predicting the likelihood of a future fall. This system would alert the user in case of a high risk of falling, and suggest what precaution they should take (e.g. resting, minimal activities, checking with a doctor, etc.). Statistically, the two leading causes of unintentional falls are loss of balance and stumbling [NDA10].

Much of the effort in the research community focused on monitoring gait and loss of balance to assess the risk of falls [NDA10], but the relationship between stumbles and falls is still not well understood. In other words, the following question is not yet answered: *Are frequent stumbles an indicator of high risk of falling?*

In the study of stumbles, it was established that there are two general strategies of recovery from stumbling: elevating and lowering. The elevating strategy occurs as a response to a perturbation during the early swing phase of gait, whereas the lowering strategy occurs during the late swing phase of gait [SWM00] [CKH04]. It was established that one's ability to perform such recovery methods after a stumble could be determined by one's ability to perform quick steps [CKH04]. Weakening of the nervous system in the elderly may seriously restrict their ability to perform these quick steps during recovery, which may lead to a fall.

To date, the most common form of stumble reporting continues to be self-report. This method is obviously flawed and not reliable since it relies entirely on the ability to remember and report stumble events. In this work, we design a system that will help in answering the question above, and in studying the relationship between stumbles and falls. We present a methodology for simulating stumbles, collecting activity data from body-mounted accelerometers, and a method for detecting stumbles.

5.2 Related Work

Most of the related work we found in the engineering literature focuses on fall detection and fast reporting to doctors. Other related work focused on gait analysis and monitoring, such as balance, gait symmetry, and gait speed. Very few papers considered the problem of stumble detection, and no work was found on relating stumbles and falls. Related work in the medical literature focused on understanding the biomechanics and muscle behavior of the stumbles. We divide our related work section into three parts, *fall detection*, *gait analysis*, and *stumble detection*.

5.2.1 Fall Detection

Fall detection is related to our work because our main purpose of stumble detection is to relate the frequency of stumbles to the risk of falling. Much of the work in fall detection has included the use of deployable body-mounted sensors, such as accelerometers, and data thresholding techniques [NFR07] [BOL07] [WSG10]. In fact, it has been shown that a single tri-axial accelerometer mounted on the trunk provides sufficient data for reliably distinguishing falls and near falls from normal gait patterns when appropriate thresholding is applied [WSG10]. One study was even able to achieve 100% detection of all simulated fall events by applying such methods [BOL07]. Other work has included the use of machine learning, which involves the training of neural networks using distinct data [NFR07]. Once trained, these neural networks can be used for the automatic classification of trained activities. Some more elaborate methods have included the use of image processing techniques of video data and even the use of pressure-sensitive floor tile arrays to detect fall events [NFR07]. Narayanan et al. were able to incorporate body-mounted sensors with onboard, real-time fall detection capabilities into a distributed detection and reporting system [NLB07]. When a fall is

detected, the wearable sensor signals a separate unit to dial an emergency number and establish an audio channel between the user of the system and emergency respondents. In addition to immediate responses, long-term logging was achieved via daily uploading of activity acceleration data to a remote server. This allows for a more detailed analysis and evaluation of user activity patterns. Interested parties such as physicians can view the status of users and prescribe appropriate treatment or preventative care for at risk users. Our work differs in that it would be used to prevent falls, and we focus on detecting stumbles which could be precursors to falls.

5.2.2 Gait Analysis

There are two types of gait analysis. The first type consists of analyzing the periodic parameters of gait, such as balance, symmetry, speed, etc. These parameters are characteristics of every walk stride. The second type consists of detecting anomalies and rare events such as stumbling, tripping, slipping, etc. These parameters are characteristics of only a few walk strides. We differentiate these two types because the methods of monitoring and inferring the parameters of these two types of analysis are different.

In [NDA10], the authors build a system for human instability and balance monitoring based on a smart shoes with embedded pressure sensors, accelerometers, and gyroscopes. The system monitors walking behavior and assesses the fall risk based on the instability or balance of the walk. The instability analysis is based on the step length, the step time, the cadence, the stride length, the gait speed, and other features. All of these features can be inferred from a periodic analysis of every stride or step. They also use a risk of fall parameter computed from the features above. In [XBK11], the authors build a system to monitor the gait speed and argue that it is an indicator of the rehabilitation progress for patients who suffer from problems in their gait. The system uses accelerometers worn on the ankles, and was evaluated on 6

subjects. There are many papers that study different aspects of gait analysis based on the periodic parameters of the gait. It is worth to mention that the problem of modeling and monitoring the periodic parameters of gait is easier than rare events, due to the simple fact that collecting data sets and reliably labeling them is easier in the first case. In monitoring and detecting rare events, the validation of the system requires a realistic data set containing the reliably labeled rare event. This is a great challenge, and our work contributes to this part of the big problem of gait analysis and fall prediction/prevention. We build a stumble detection system and validate it on a big and representative data set containing 100 stumbles, with reliable ground truth.

5.2.3 Stumble Detection

There has been a good amount of research in the medical and human biomechanics field, to understand stumbles, their causes, and their recovery strategies. The objective was not about detecting stumbles, but rather about understanding the muscle and body behavior when a stumble happens. It was established that a fall occurs when the person is not quick enough to recover from a stumble and prevent the fall. The human response to stumbling has been well characterized [SWD96] [CKH04]. Two recovery strategies were identified: elevating, and lowering. The elevating strategy occurs as a response to a perturbation during the early swing phase of gait, whereas the lowering strategy occurs during the late swing phase of gait. The related aspect to our research is in the simulation of stumbles. We found out that there are two major categories in simulating stumbles: treadmill-based, and terrain-based. In [SWD96], an experiment was designed based on a treadmill; subjects were asked to walk on a treadmill, and obstacles were dropped during early swing. It was found that the knee is bent more extensively in order to lift the foot over the obstacle during stumbling. In [CKH04], a treadmill-based experiment was also performed and the authors found that not only

the lower limb motions, but also the control of the trunk is necessary to understand the recovery process. A mechanical model for recovery is also developed and used to analyze a series of experiments. It is shown that the position of the front foot is crucial in keeping the trunk upright. This result bolsters the hypothesis that the response speed, as in step speed, is important for recovery. If the step is too low, the hip can't be moved extensively enough to compensate the forward trunk movement. Hence, smaller steps become necessary, suggesting that the capacity to perform quick steps determines the strategy chosen for recovery.

Surprisingly, there has been a very little amount of research on systems for stumble detection. We only found two papers that considered a version of the stumble detection problem. Both papers were motivated by the design of an intelligent and active lower limb prosthesis. The objective is to build an intelligent prosthesis that can detect a stumble when it occurs and have an active recovery response to prevent the subject from falling. In [LVS10], the stumble detection and active recovery are based on accelerometers mounted on the prosthetic limb. To simulate stumbles, a walkway is constructed with hidden obstacles that appear as subjects walk through it. They collected their data from 10 healthy subjects who were instrumented with three accelerometers on the left leg (foot, shank, and thigh). They only collected 19 stumbles and 34 normal walk strides from 10 subjects. They report 100% detection accuracy, and they don't report any false alarm rate or precision-recall results. This data set is unfortunately too small and not representative. In [ZDN11], the detection is based on accelerometers and EMG sensors that measure the muscle activity of the hip. They collected two sets of data, the first set is from 7 subjects with transfemoral amputations. Five of these subjects were asked to walk on a treadmill where sudden accelerations or decelerations of the treadmill were used to simulate stumbles; for each subject ten trials with sudden treadmill acceleration and then trials with sudden treadmill deceleration were tested. The speed, accelerations and decelerations were fixed and were the same for

all 5 subjects. The other two were asked to walk on an obstacle course. A total of 15 trials were tested for each subject, and each trial was for 5 minutes. The total number of stumbles is not reported, but they report 100% stumble detection, and between 0% and 0.0009% false alarm rate (depending on the subject). They also report that the EMG sensors were necessary to achieve this very small false alarm rate. We think that a treadmill-based data set is far from being realistic. The main difference in our work is that we collect a bigger and more realistic data set, and that we achieve very good results with a much simpler system of only one accelerometer. Our data set is collected on an outdoor terrain, and contains a total of 100 stumbles, and over one hour of normal walk for 9 subjects.

5.3 Stumble Detection Methodology

5.3.1 Background and Notation

Before we formulate the the problem, we introduce and define the mathematical terms we are going to use. We borrow some of the definitions from [MKZ09], and extend them to our problem.

Definition 1: A 1-D Time Series of length m is a sequence $D = \{d_{t_1}, d_{t_1+1}, \dots, d_{t_m}\}$ of an ordered set of m real valued numbers, with time indices $t_i, i = 1, \dots, m$.

Definition 2: A 3-D Time Series is a sequence $D = \{d_{t_1}, d_{t_1+1}, \dots, d_{t_m}\}$ of an ordered set of m real valued column vectors d_{t_i} , with time indices $t_i, i = 1, \dots, m$.

Definition 3: A 3-D subsequence of length T of a 3-D time series D , is a 3-D time series $D_{t_i:t_i+T} = \{d_{t_i}, d_{t_i+1}, \dots, d_{t_i+T}\}$.

We can represent the 1-D time series D by a $(2 \times m)$ matrix where the first row is the time indices t_i , and the second row has the corresponding values d_{t_i} . We represent

the 3-D Time Series D by a $(4 \times m)$ matrix where the first row is the time indices t_i , and the second to fourth rows correspond to the column vectors d_{t_i} . We see now that the matrix corresponding to a 3-D subsequence of a time series D , is a sub-matrix of the matrix corresponding to D .

5.3.2 Problem Formulation

We consider the problem of human stumble detection using one acceleration sensor worn on the body, and we would like to test the effectiveness of different body locations. The accelerometer measures the [X, Y, Z]-components of the acceleration at the body location where it is mounted. The sensor could be worn on L different parts of the body. We would like to instrument the user for a long period of time (e.g. a day, a week, a month), and detect how many stumbles they had during that time. We model the acceleration data of each sensor as a 3-D time series of length m (m corresponds to the time period of interest, e.g. day, week, month, etc.), $D_l = \{d_{l,t_1}, d_{l,t_2}, \dots, d_{l,t_m}\}$, where D_l is the acceleration data indexed by the body location $l \in \{1, \dots, L\}$ between the times t_1 and t_m , and d_{l,t_i} is the three dimensional vector of acceleration at location l and time t_i . To detect and count the number of stumble in a 3-D time series D_l , we detect the occurrence of a stumble in every 3-D subsequences of $D_{l,t_i:t_i+T}$. We associate a variable $S_{t_i:t_i+T} \in \{0, 1\}$ with every subsequence $D_{l,t_i:t_i+T}$ to represent the stumble, where $S_{t_i:t_i+T} = 1$ if a stumble occurs, and $S_{t_i:t_i+T} = 0$ otherwise. We assume that a single stumble could happen between t_i and $t_i + T$. We use a probabilistic approach, and define the following probabilities:

$$P(D_{l,t_i:t_i+T} | S_{t_i:t_i+T} = 0) = \text{Probability that } D_{l,t_i:t_i+T} \text{ is a normal walk pattern,}$$

$$P(D_{l,t_i:t_i+T} | S_{t_i:t_i+T} = 1) = \text{Probability that } D_{l,t_i:t_i+T} \text{ is a stumble walk pattern.}$$

The stumble detection problem now becomes a likelihood test comparing

$P(D_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)$, and $P(D_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 1)$:

$$\frac{P(D_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 1)}{P(D_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)} \geq \tau, \quad (5.1)$$

i.e. a stumble is detected in $D_{l,t_i:t_i+T}$ when the likelihood ratio exceeds a threshold τ .

Again in this problem, we move to the feature space and we compute features from the raw acceleration $D_{l,t_i:t_i+T}$, and compute the likelihood test in the feature space. We compute a feature vector $f_{l,t_i:t_i+T}$, and get the following likelihood test:

$$\frac{P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 1)}{P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)} \geq \tau. \quad (5.2)$$

5.3.3 Feature Extraction

The main objective of feature extraction is to reduce the dimensionality of the data. In this work the sensors are measuring the accelerations at 160hz, which creates $(3 \times 160 \times T)$ samples in every 3-D subsequence. Feature extraction summarizes the information in every 3-D subsequence, and results in a compact representation of the 3-D subsequence $D_{l,t_i:t_i+T}$. Figure 5.1 shows a 13 seconds window of walking containing a stumble around the 6.5 seconds mark.

The first challenge is that the accelerometer measurements depend on the orientation of the accelerometers. Moreover, the accelerometers pick the gravity acceleration on the vertical axis which changes depending on the orientation of the sensor. Therefore, we first remove the gravity component by subtracting the decaying average from each dimension of the raw accelerometer data [FOC11], and then compute orientation invariant features. The decaying average is given by

$$\overline{D}_{l,t_i} = \frac{D_{l,t_i}}{L} + \frac{L-1}{L} \overline{D}_{l,t_i-1}, \quad (5.3)$$

where L is the decay factor. The important characteristic of the decaying average is that it adapts quickly to the orientation of the accelerometer, and removes the gravity

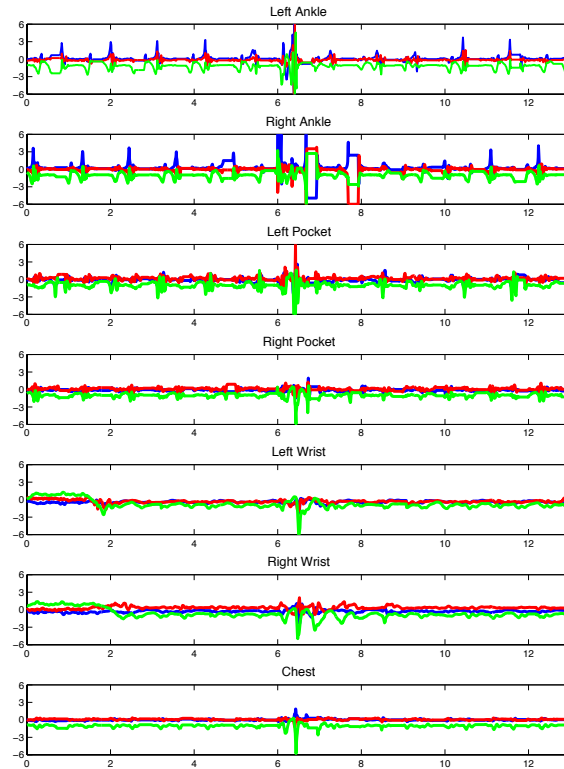


Figure 5.1: A 13 seconds window of walking containing a stumble at 6.5, seen from different locations on the body. The x-axis is in seconds, and the y-axis is in g .

component from all of the axes. Figure 5.2 shows a 13 seconds 3-D time series, the upper figure shows the raw data where gravity is picked up by red axis. The lower figure shows the acceleration data after subtracting out the decaying average, the gravity component is gone. Note that the decaying average was subtracted from all three axes. After removing gravity, we extracted orientation-invariant features. In fact, we extracted features from the magnitude (the euclidean norm) of the $[X, Y, Z]$ acceleration vector, $m = \sqrt{X^2 + Y^2 + Z^2}$, which is orientation-invariant. For the detection of stumbles in a 3-D time series D_t , we compute the features for the 3-D subsequences,

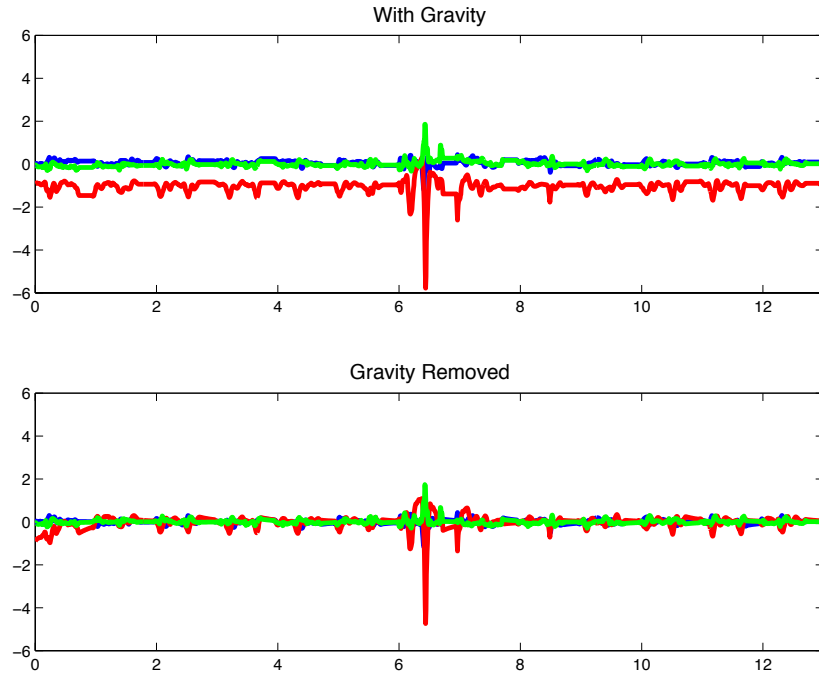


Figure 5.2: A 13 seconds acceleration time series. The upper figure shows the raw acceleration, the red axis is at $-1g$ picking up the gravity acceleration. The lower figure show the acceleration after removing gravity by subtracting the decaying average. The x-axis is in seconds, and the y-axis is in g .

$D_{l,t_i:t_i+T}$. The features we used in this work are listed in Table 5.1.

5.3.4 Stumbles as Anomalies

We now describe our approach to detect stumbles from the acceleration data, using anomaly detection techniques. Anomaly detection refers to the problem of finding patterns in the data that do not follow a normal or expected behavior. These abnormal patterns are called anomalies or outliers. In our problem, a normal walk pattern represents the normal behavior in the data, and a stumble pattern corresponds to the

Features
Maximum value of m
Non Linear Energy Operator of m
FFT of m

Table 5.1: Features used.

anomaly. The key idea to our approach is that since the stumbles are rare events, and it is hard to collect training data to model them. Therefore we use a semi-supervised anomaly detection approach where we only train a model for the normal walk patterns, and we detect stumble patterns as deviations from this model [CBK09]. Mathematically, we need to estimate the density $P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)$ from training data, and detect stumbles if $P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)$ is low when $S_{t_i:t_i+T} = 1$. The stumble detection will be based on $P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)$ as follows:

$$P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0) \leq \tau_l, \quad (5.4)$$

for some sensor specific threshold τ_l .

5.3.5 Estimating $P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)$

Density estimation is one of the fundamental problems in statistics and machine learning. There are two general ways of doing it: parametric and non-parametric density estimation. Parametric density estimation assumes that the density belongs to some parametric family of distributions $\phi(\theta)$, and the goal is then to estimate the parameters θ from the training data set. Non-parametric density estimation drops the parametric assumption, and estimates the density directly from the training data (histograms are the most common non-parametric density estimation). In this work, we use a parametric approach and we estimate $P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0)$ by a Gaussian distribution:

$$P(f_{l,t_i:t_i+T}|S_{t_i:t_i+T} = 0) = (2\pi)^{-\frac{k}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(f-\mu)'\Sigma^{-1}(f-\mu)} \quad (5.5)$$

where the parameters are the mean μ and the covariance matrix Σ . k is the dimension of the feature vector f , i.e. the number of features used. We estimate μ and Σ from training data.

5.3.6 Setting the threshold τ_l

After estimating $P(f_{l,t_i:t_i+T} | S_{t_i:t_i+T} = 0)$, we need to set the threshold τ_l . Again, τ_l is the threshold that corresponds to each location $l \in \{1, \dots, L\}$. We choose τ_l in order to control the false alarm rate of the stumble detector. Since the stumbles are rare events, there is a high probability that a lot of the detected stumbles are actually normal walk patterns. Thus, we would like to control the false alarm rate, and we choose to set τ_l in order to achieve a desired false alarm rate $FA R_l$. Note that $FA R_l$ also depends on the sensor location on the body.

5.3.7 Choosing T

T is the time length used in parsing the 3-D time series D_l , or in other words T is the time length of the 3-D subsequences $D_{l,t_i:t_i+T}$. The choice of T depends on the application of the stumble detection. For example, in [LVS10] and [ZDN11] a stumble detection system was built for people with prosthetic legs, and the goal was to detect stumbles in real time and right when they happen to actively move the leg accordingly, and prevent the fall from happening. In their application, T needs to be in the order of milliseconds, since the leg had to be moved quickly to help with the recovery from the stumble and prevent the fall. T was chosen to be 100ms in [LVS10], and as low as 10ms in [ZDN11]. In other applications, doctors could be interested in monitoring their patients not in real time, and statistics of stumbles calculated at the end of the day, or the week. In these applications, the stumble detection is done in post processing of the data, and a larger T could be used (now in the order of seconds). In our work, we

chose $T = 4seconds$.

Another parameter that is associated with T is the step size used in parsing D_l , to get the corresponding subsequences $D_{l,t_i:t_i+T}$. We choose non-overlapping subsequences, and choose the step size to be 4 seconds.

5.4 Experiments

5.4.1 Data Collection

Our system consists of Gulf Coast Data Concept X6-2 mini tri-axial accelerometer sensors [htt]. A total of 7 accelerometers were placed on various parts of the subjects' bodies; they were placed on the left and right wrist, the middle of the chest, left and right pocket, and left and right ankle (figure 5.4). The sensors were set to collect data at a rate of 160 Hz with a 16-bit data resolution and $\pm 6g$ acceleration range.

In order to simulate stumbles, we used a simple platform consisting of a 1 x 8 x 30 inch wooden base plank attached to two stacked 2 x 4 x 30 inch wooden studs (Figure 5.3), which created a low vertical barrier for test subjects during gait. Test subjects were blindfolded and listened to music through headphones so that they do not see or hear the person setting the obstacle in front of them, and to provide distraction. The volunteers were then told to walk in a straight line. The obstacle was placed in front of them at random times but no less than ten seconds after the start of a walk interval, whereupon the subject would come into the vicinity of the obstacle and either stumble and recover or walk over the obstacle and continue walking. After the stumble, subjects would continue walking for one minute or more. The process was repeated about twenty times per test subject. Stumbling was followed by a long walking interval (about 5 minutes) in order to more sufficiently train the classifier for walks. Two individuals walked with the test subject to secure their safety, and prevent them from

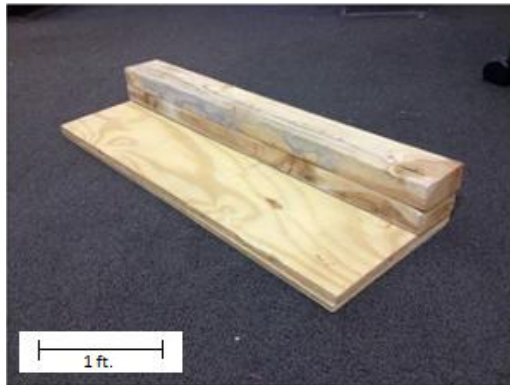


Figure 5.3: Obstacle used

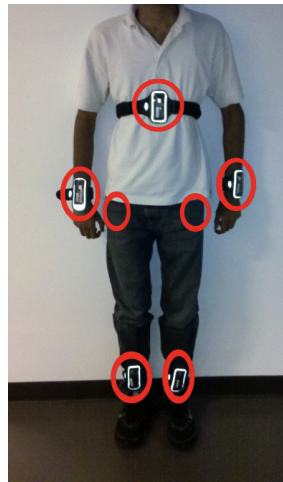


Figure 5.4: Sensor positions

falling. The experiments took place on an open grassy court, and a video was recorded for ground truth. The data was collected from 9 test subjects (7 Males and 2 Females). We got a total of 100 stumbles, and table 5.2 shows the number of stumbles for each subject.

Subject	Number of Stumbles
1	7
2	10
3	12
4	11
5	11
6	12
7	14
8	10
9	13
Total	100

Table 5.2: Number of stumbles for each subject.

5.4.2 Limitations of the experiments

A main obstacle in designing activity monitoring systems is the validation process of the system. The system needs to be validated on people in realistic situations, as close as possible to every day life. A data set should be collected and labeled for the real stumbles, and this procedure should be done for many people. This is very challenging and we thought that a reasonable first step towards this goal, is to collect data in a controlled environment and simulate stumbles which are easy to be labeled. Our experiments and data sets suffer from the following limitation:

1. One type of stumbles: We only simulate one type of stumble; stumbling on a fixed obstacle. Other stumbling types we didn't consider: stumbling on a cord, stumbling on a moving object, and slipping.
2. Controlled environment: The data sets are collected in a controlled environment.

We realize that the experiments we have done have certain limitations, but nonetheless we think that they are more representative than other experiments used in the literature. The two research papers we found in the stumble detection literature, [LVS10] and [ZDN11], use also a controlled environment, but use a treadmill where they can control the speed to produce stumbles. The subjects were asked to walk on the treadmill, and stumbles were created with unexpected sudden changes in the treadmill’s speed.

5.5 Results

In this work we build a stumbling detection system based on *one* accelerometer worn on the body, and we test 7 different locations. As described in Section 5.3, we use a semi-supervised anomaly detection in the feature space, by estimating $P(f_{l,t_i:t_i+T} | S_{t_i:t_i+T} = 0)$ from the data. In this section, we present the stumbling detection results for our system, on the data described in Section . We used half of the data for training the stumbling detector, and half of it to test it.

In rare event detection, it is not enough to look at accuracy and probability of misdetection since the events we are trying to detect are rare, and therefore the accuracy and probability of misdetection would be dominated by the non-rare events. These issues arise in classification problems when one class has far fewer points than the other classes. To resolve these issues we use the following quantities:

$$FAR = \frac{\text{Number of normal subsequences detected as stumbles}}{\text{Total number of stumbles}} \quad (5.6)$$

$$Precision = \frac{\text{Number of stumble subsequences detected as stumbles}}{\text{Number of all subsequences detected as stumbles}} \quad (5.7)$$

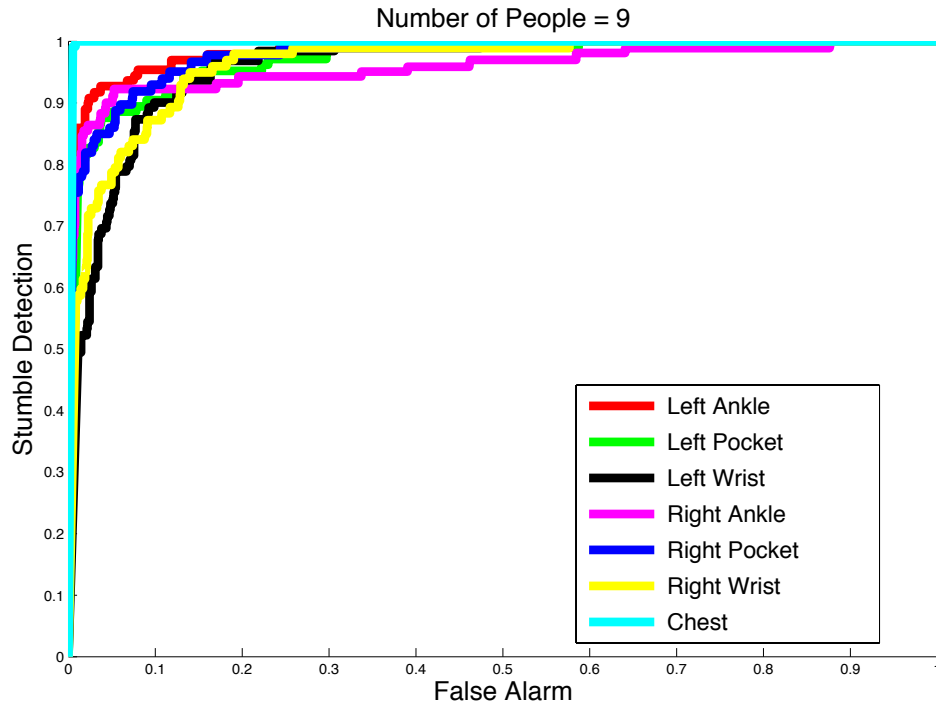


Figure 5.5: ROC curve. The chest achieves a detection of 99% with a false alarm of 0.5%.

$$Recall = \frac{\text{Number of stumble subsequences detected as stumbles}}{\text{Number of subsequences detected as stumbles}} \quad (5.8)$$

In Figure 5.5, we present the performance of our system as a ROC curve, which is probability of detection versus false alarm rate. We evaluate each of the 7 locations, and show each as a curve. We see that the chest performs better than all the other locations; its ROC curve is higher and to the left of all other locations. With one chest sensor we could achieve a stumble detection of 99%, with a false alarm rate of 0.5%. In figure 5.6, we show the precision-recall plots for our stumble detection system. We also see that the chest outperforms the other locations for the accelerometer.

There are two main reasons why the chest performs better than all the other loca-

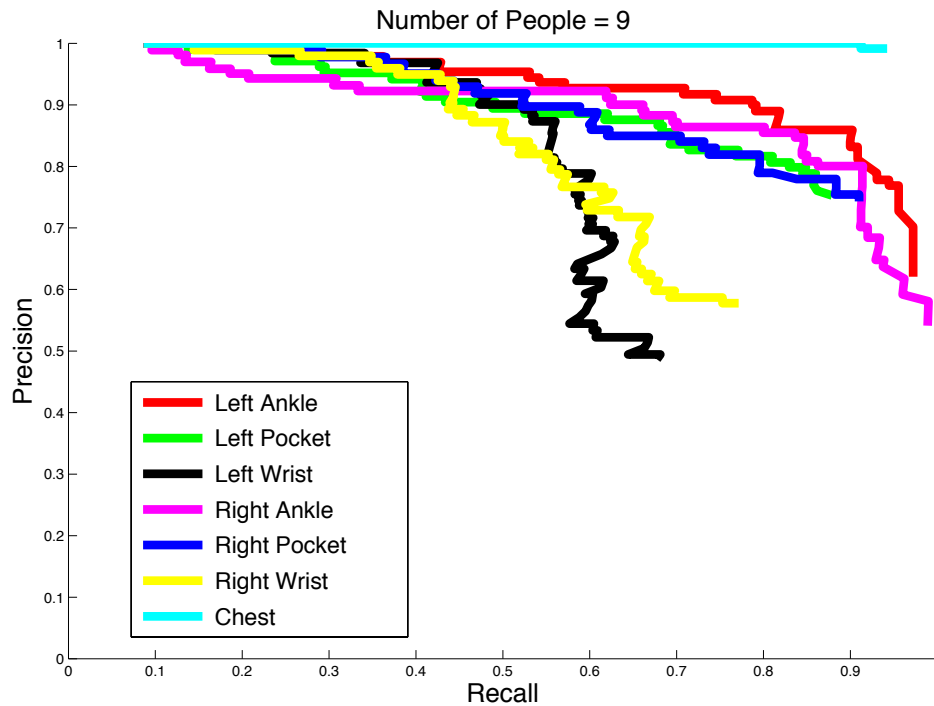


Figure 5.6: Precision Recall Curve. The chest achieves a precision of 99% precision with a recall 94% (The end point of the chest curve).

tions. The first one is that the chest is the highest location from the ground, thus the shocks produced by walking get absorbed by the body and diluted by the time they reach the chest. We can see this phenomenon in Figure 5.1, the walk peaks are smaller in the acceleration measured at the chest and the stumble pattern stands out more. The second reason is that the chest sensor is placed in the middle of the body, whereas all the other locations is either on the left or on the right of the body. We noticed that if a stumble happens to the left leg, the right side of the body might not pick up the stumble acceleration pattern, and vice versa. The only limitation of placing the accelerometer on the chest is that it could be hard to mount it on the chest, it is more feasible to place it in one of the pockets or clip it to the belt at the waist. With the method used, the pockets were not good locations to detect the stumbles, but we plan to extend this work

and use more powerful methods to be able to detect the stumbles more reliably at the pockets and the waist.

5.6 Conclusion

In this chapter we studied the problem of detecting stumbles using *one* accelerometer worn on the body. We evaluated seven locations and found that the chest is the best location to detect stumbles. Our approach learns a statistical model to characterize normal walking, and detects stumbles as anomalies or deviations from this model. Our approach guarantees a certain false alarm rate, and we show that stumbles could be detected with a 99% accuracy and 0.5% false alarm rate with one sensor worn on the chest. Our system is personalized where we build a detector for each user tailored to their normal and stumble walking patterns. We also provided a big data set for 9 people, containing a total of 100 stumbles and more than one hour of normal walking. This is a much bigger data set than all other works in the literature. In fact [LVS10] evaluated their system on 19 stumbles and 34 normal walking strides collected from 10 people on a treadmill, and [ZDN11] collected their data for 7 people on a treadmill, and for only 2 people on a realistic terrains. [ZDN11] did not report the total number of stumbles they collected. Moreover, they used EMG and acceleration sensors in their system. We strongly believe that accelerometers are enough for the detection of stumbles. A main limitation of treadmill data, is that the acceleration is controlled and the speed of the walking is fixed. Our dataset is more realistic and takes the research on stumble detection a step further towards the goal of realistic stumble detection systems, and towards understanding the relationship between stumbles and falls.

For future work, we plan to take this study in three directions. The first direction is to collect more realistic data, containing more types of stumbles. The second direction is to collect data for elderly patients who have a high risk of falling. We plan to mon-

itor them over a long period of time and use our current detector based on controlled experiments, to help with the labeling of the real life data set. We believe that running our current detector on the real life data set, would detect potential stumbles that could be combined with the information from the subject to reliably label the real stumbles. The third direction is more technical, and consists of developing more powerful detection techniques that could be used to detect stumbles from more plausible placements of the accelerometers, such as pockets.

CHAPTER 6

Active Learning for Model Selection

This chapter presents collaborative work with Professor Gregory Pottie, and was triggered by some discussions with Professor Mark Hansen in the statistics department at UCLA.

6.1 Introduction

Sensor networks have proved to be useful for learning about natural phenomena (e.g. temperature, humidity, light, etc.) over spatial and temporal fields [BRY04] [TPS05]. Statistical models are widely used to make sense of the collected measurements and to answer questions regarding these phenomena such as the estimation of the phenomena, the prediction of the phenomena changes over time or space, the effects of some phenomena on other phenomena, or any sort of inference [BRY04] [KGG06].

One can randomly place sensors, collect measurements and make inferences about the phenomenon in question. This work instead focuses on the problem of directing the placement of sensors and the collection of measurements, depending on the application and the desired final inferences. The use or choice of certain statistical models, combined with the desired inferences, leads us to formulate methods which direct sensor placement for high quality inference results.

In this work we focus on the model selection problem. We assume that we have a set of plausible models and we want to select the correct one, i.e. the set of models

contains the correct model. We intend to develop methods (or algorithms) that directs the collection of measurements to optimize the selection of the model structure, from the set of plausible model structures.

Our philosophy is to minimize the probability of error in the selection of the correct model structure. We show in Section 6.3, that in the fundamental selection problem the probability of error depends on the data considered. And we can formulate optimization problems to select the data that minimize the probability of error.

We will focus on the linear regression framework, but we will argue in Section 6.4.1 that the developed methods work for other types of models (e.g. Gaussian processes, splines, kernel-based models).

The remainder of the chapter is divided into two sections, the first one is about the work and results done so far while the second one is about the suggested problems for future work. Furthermore, each section contains some literature review done on the problem considered.

We will start with a small introduction about regression models since they are used throughout the chapter.

6.2 The Regression Framework

In this work, we shall be concerned with the linear regression framework where the measurements are given by:

$$\mathbf{y}_i = h(x_i, a) + \mathbf{e}_i, \quad i = 1, \dots, m \quad (6.1)$$

where x_i is a vector corresponding to the i th measurement location, \mathbf{y}_i is the measurement at x_i , \mathbf{e}_i is the measurement noise at x_i , a is the vector of unknown parameters and m is the number of measurements. The range over which we collect measurement

is called Z . The linearity of the model $h(x_i, a)$ is with respect to a i.e. $h(x_i, a) = x_i^T a$.

If we collect y_i in an m -dimensional vector \mathbf{y} we get:

$$\mathbf{y} = X^T a + \mathbf{e}, \quad (6.2)$$

where X is the matrix whose columns are the x_i vectors. X is called the *design matrix*.

We assume that $\mathbf{e} \sim \mathcal{N}(0, C_e)$ where C_e is known.

Given a set of measurements, the parameter a can be found by solving the following least-squares problem [HTF01]

$$\min_a \|(y - X^T a)^T C_e^{-1} (y - X^T a)\|_2^2 \quad (6.3)$$

The solution is given by:

$$\hat{\mathbf{a}} = (XC_e^{-1}X^T)^{-1}XC_e^{-1}\mathbf{y} \quad (6.4)$$

6.3 Sampling Designs for Model Selection

In this section, we look at the situation when the field is one of two competing regression models and we want to design a strategy to determine which of the models is true. A common method is to randomly place sensors, collect measurements, estimate the models and use a model selection criterion to pick the model given the measurements collected. In this section we show how to find placements and collect measurements to optimize the model selection. This falls into the fundamental experimental procedure where one has a prior set of plausible models for the field and resorts to guided measurement collection to pick the model, from the set, that best describes the field. It turns out that the measurement collection should be done in iterations with the estimation of the parameters of the regression models. Assume that the two competing linear

regression models are:

$$\begin{aligned} h_1(x, a) &= X_1^T a \\ h_2(x, b) &= X_2^T b. \end{aligned} \tag{6.5}$$

where X_1 and X_2 are the corresponding design matrices.

If m measurements are collected at locations x_1, \dots, x_m then

$$\mathbf{y} = h(x) + \mathbf{e}, \tag{6.6}$$

where the phenomenon $h(x)$ is the true model and $\mathbf{e} \sim \mathcal{N}(0, C_{\mathbf{e}})$ is the measurement noise. We shall assume, without loss of generality, that $C_{\mathbf{e}}^{-1} = \sigma^2 I$.

The parameters a and b are estimated, as in (6.4), by

$$\hat{a} = (X_1 C_{\mathbf{e}}^{-1} X_1^T)^{-1} C_{\mathbf{e}}^{-1} X_1 y \tag{6.7}$$

$$\hat{b} = (X_2 C_{\mathbf{e}}^{-1} X_2^T)^{-1} C_{\mathbf{e}}^{-1} X_2 y. \tag{6.8}$$

6.4 The fundamental model selection problem

This section is organized as follows. In section 6.4.1, we review the related work. In Section 6.4.2, we formulate the problem mathematically for the maximum likelihood criterion (MLC), Akaike' information criterion (AIC) and Bayes' information criterion (BIC) frameworks and we present the algorithms that define a strategy for the measurement collection. Section 6.4.4 shows the performance of the algorithms along with discussions on the results obtained.

6.4.1 Related Work

The problem considered here is how to select training data in the modeling process. This type of problem has been called experiment design in the statistics literature [AF75] [Puk93], active learning in the machine learning literature [CWN05] [CGJ96]

[Mac92], and sensor placement or adaptive sampling in the sensor networks literature [KGG06]. The work in [KGG06] is based on the Gaussian process framework to model the spatial field. In this framework a Gaussian random variable is associated with the phenomenon at each location and for any set of locations a Gaussian random vector is used to model the phenomenon at these locations. Given measurements at certain locations, the field value at these locations can be estimated using the ML criterion with an error covariance matrix which depends on the covariance matrix of the Gaussian process. In [KGG06], a pilot deployment is used to collect data and estimate the covariance matrix of the Gaussian process using a kernel function. The work in this chapter could be used in the estimation of the covariance matrix structure. Instead of assuming one certain structure, one can use the work in this chapter to optimize the selection of a structure for the matrix from a set of plausible structures.

The work in [RHK05] considers the sensor placement for maximal coverage in the design space without any knowledge about the model of the field. Mobile sensors are used to travel and collect measurements at these placements. The travel time is taken into consideration in finding the placements. Here we assume that the model is one of many defined models.

The work in active learning [CWN05] and [CGJ96] considers the problem for the regression and neural networks respectively. They assume one class of models and perform active training data selection to learn the model.

The difference in this work is the consideration of the uncertainty in the structure of the statistical model used. Instead of assuming one structure (one class) for the model, we devise algorithms for optimizing the selection of the structure of the model. We do experiment design, active learning or sensor placement for the model selection problem.

The work of [AF75] is the closest to the work presented here. It considers the ML

selection criterion where the solution (measurement locations) is called T -optimum design. Here we extend the work of [AF75] to consider the AIC [Aka74] and BIC[HTF01] criteria as well. Moreover we present error rates for the algorithms and the improvement over random designs.

6.4.2 Maximum Likelihood Framework

The maximum likelihood framework is widely used for model selection and it is based on the likelihood test. The performance measure used in this framework is the probability of error in the selection. The sensor placement would be designed to minimize the probability of error. We do the following in this order: we derive the likelihood test then we formulate the sensor placement problem for optimizing the likelihood test and finally we present the algorithm that solves the optimization problem.

The likelihood test is to compare the likelihoods of $h_1(x, \hat{a})$ and $h_2(x, \hat{b})$, $P(h_1|y)$ and $P(h_2|y)$ respectively, and pick the one with the largest value. The log-likelihood test is

$$\max_i \log(f(y|h_i)) \quad (6.9)$$

where $f(y|h_i) \sim \mathcal{N}(h_i, \sigma^2 I)$. This is a binary hypothesis testing problem and the probability of error, P^e , in selecting the model is [Poo98]

$$P^e = Q\left(\frac{\sqrt{\|h_1(x, \hat{a}) - h_2(x, \hat{b})\|_2^2}}{2\sigma}\right) \quad (6.10)$$

where $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{s^2}{2}\right)$

To optimize the selection we choose to minimize the probability of error (6.10). Since the function Q is strictly decreasing we find the sensor placements by maximizing the argument of $Q(\cdot)$ in (6.10). We discretize the range and introduce a weight

vector w to get the following problem

$$\begin{aligned} \max_w \quad & \|diag(w) (h_1(x, \hat{a}) - h_2(x, \hat{b}))\|_2^2 \\ \text{where } \hat{a} = \arg \min_a \quad & \|diag(w) (y - h_1(x, a))\|_2^2 \\ \hat{b} = \arg \min_b \quad & \|diag(w) (y - h_2(x, b))\|_2^2 \end{aligned} \quad (6.11)$$

The weight vector w reflects how many times each location was selected. A non-zero w_i means the location x_i was selected. This problem (6.11) is similar to the problem derived in [AF75] where the solution \hat{w} was called the T -optimum design. The derivation of (6.11) here is different than the one in [AF75] and follows from the minimization of the probability of error in the likelihood test.

We see from (6.11) that we need to know \hat{a} and \hat{b} to find the weights \hat{w} and we need to know \hat{w} to estimate \hat{a} and \hat{b} , i.e. on the one hand we need to know the parameters of the models to find the optimal sensor placement and on the other hand we need to collect measurements at certain locations to estimate the parameters of the models. The solution \hat{w} of (6.11) was also given in [AF75] by the following sequential algorithm, *Algorithm (1)*:

1. Given a design $w(j)$ and a number of measurements N find:

$$\begin{aligned} \hat{a}_j &= \arg \min_a \|(\mathbf{y}_j - h_1(x, a))\|_2^2 \\ \hat{b}_j &= \arg \min_b \|(\mathbf{y}_j - h_2(x, b))\|_2^2 \end{aligned}$$

2. Add to the design the location z_{j+1} such that:

$$z_{j+1} = \arg \max_{z \in Z} \left(h_1(z, \hat{a}_j) - h_2(z, \hat{b}_j) \right)^2$$

3. Make a measurement s at z_{j+1} and update

$$w(j+1) = \left(1 - \frac{1}{N+1}\right)w(j) + \frac{1}{N+1}\delta(z_{j+1})$$

4. Go back to 1

where $\delta(z_{j+1})$ is a vector with all entries equal to *zero* except the entry corresponding to z_{j+1} is equal to 1. In each iteration the algorithm estimates the parameters, \hat{a} and \hat{b} , from the available measurements, picks the point that maximizes the difference between the models, makes a measurement there and updates the weights associated with locations by adding $\frac{1}{N+1}$ weight to the point selected and rescaling so that the weights add up to 1. Details on the convergence of the algorithm to the solution of (6.11) are available in [AF75]. Finally, given the measurements collected, a likelihood test is used to select the model.

The initial locations are found by the D -optimum design [Puk93]. The D -optimum design consists of the locations that minimize the variance of the error in the estimation of the models parameters. This initial design is also used in [AF75].

We would like to note that in the case where the two models are nested, for example

$$\begin{aligned} h_1(x_i, v_i) &= a_0 + a_1 x_i, & i = 1, \dots, m \\ h_2(x_i, v_i) &= b_0 + b_1 x_i + b_2 x_i^2, & i = 1, \dots, m, \end{aligned} \tag{6.12}$$

the space of the second model includes the space of the first model. In this case one would be interested in the structure of the model and would add the constraint $b_2 \geq 1$ in the estimation of the parameters (step 1 of the algorithm) [AF75].

6.4.3 Akaike's and Bayes' Information Criteria

In this section we consider the Akaike's Information Criterion AIC [Aka74] and Bayes' Information Criterion BIC [HTF01] frameworks for the model selection. In these frameworks the information criterion of a model h , $IC(h)$ is given by:

$$IC(h) = (-2)\log(ML(h)) + k_h, \tag{6.13}$$

where $ML(h)$ is the maximum likelihood of h , and k_h is a term penalizing for the degree of the model. For AIC , $k_h = 2d$ where d is the number of free parameters in

the model h . For BIC , $k_h = (\log m)d$ where m is the number of measurements. The model having the smallest IC is selected. The difference from MLC is the addition of the k_h term reflecting the model complexity, so a low complexity model is favored over a high complexity one. Note that if the two models have the same complexity, both AIC and BIC reduce to maximum likelihood.

As in the previous section, we are interested in a measurement collection strategy to minimize the probability of error in the model selection when there are two competing models. The probability of error analysis is similar to the analysis in the maximum likelihood framework. Given some measurements, the probability of error in using AIC or BIC can be shown to be:

$$\begin{aligned}
P^e &= 0.5Q\left(\frac{(k_2-k_1)\sigma^2+\|h_1(x,\hat{a})-h_2(x,\hat{b})\|_2^2}{2\sigma\|h_1(x,\hat{a})-h_2(x,\hat{b})\|_2}\right) \\
&+ 0.5Q\left(\frac{(k_1-k_2)\sigma^2+\|h_1(x,\hat{a})-h_2(x,\hat{b})\|_2^2}{2\sigma\|h_1(x,\hat{a})-h_2(x,\hat{b})\|_2}\right)
\end{aligned} \tag{6.14}$$

We see that the probability of error P^e is a function of the selected locations. To optimize the model selection in these frameworks, the algorithm developed is similar to *algorithm* (1) with the modification of step 2 to

$$z_{j+1} = \arg \min_{z \in Z} P^e. \tag{6.15}$$

Examination of (6.14) shows that (6.14) is decreasing in the distance between the models. Thus we can in performing the minimization (6.15) simplify it to the maximization of the distance between the models. The AIC and BIC will thus select the same locations and samples as ML , but will produce different model selections due to the selection criteria. Different model selection criteria can result in different samples being selected.

These conclusions are the consequence of the use of the probability of error expression in (6.14) which averages the missed detection and the false alarm probabilities.

However, we found that this choice led to better performance than for example minimizing only the missed detection probability.

6.4.4 Evaluation and Discussion

In this section we present an example that shows the performance of the algorithm, presented above, in terms of the improvement in the correct model selection over using a collection of measurements at randomly selected locations. We show plots of the error rate as well as plots of the behavior of the algorithms versus the number of iterations. The example used in the simulations is described in the following. The two competing models are:

$$\begin{aligned} h_1(x_i; a) &= a_0 + a_1 x_i, & i = 1, \dots, m \\ h_2(x_i; b) &= b_0 + b_1 e^{x_i} + b_2 e^{-x_i}, & i = 1, \dots, m \end{aligned} \quad (6.16)$$

We define a Difference Power to Noise Ratio (DPNR):

$$DPNR = \frac{\|h(x) - h_2(x, \hat{b})\|}{\sigma} \quad (6.17)$$

where $h_2(x, \hat{b})$ is obtained by fitting h_2 to the truth over the whole range considered. This measure reflects the minimum distance between the truth and the space of the second model, normalized by the noise standard deviation. We show it in the plots since SNR (which is the truth power to noise ratio) does not reflect the closeness of the models, i.e. a high SNR could correspond to a low DPNR depending on the competing models.

The algorithm was tested in two scenarios. The first one is when the truth was $h(x) = 4.5 - 0.6e^x - 1.3e^{-x}$ over the range $Z = [-1.5; 1.5]$. The number of realizations was 100,000. The error rate was computed as the ratio of the number of wrong selections to the number of realizations. The performances of the ML, AIC and BIC

are shown in Fig.6.1. We can see that there is a significant improvement, around $3dB$, over the random collection of measurements. As expected the ML selection performs better than both AIC and BIC. This is due to the properties of ML, AIC and BIC: both AIC and BIC would favor the lower complexity class while the true model lies in the higher complexity one. Moreover the reason that the AIC selection is better than the BIC one is that, over a finite number of measurements, BIC has a heavier penalty on complexity than AIC since it includes a term related to the number of measurements [HTF01]. Fig.6.2 shows that the error rate decreases with increasing numbers of iterations (samples). The group of dashed lines shows the error rate when $DPNR = -0.56dB$ and the group of solid lines shows the error rate when $DPNR = 4.29dB$. The sequential method has a much steeper slope for the three error rates than the ones for random methods for all model selection criteria. The higher the DPNR, the greater the difference between the sequential method and the random method. When the DPNR is very low, the error rate decreases slowly. For the random design, we only showed the behavior of BIC since all of them behave similarly.

In the second scenario, the truth was $h(x) = 2 + 5x$ over the range $Z = [-1.5; 1.5]$. Fig.6.3 shows the performances of the ML, AIC and BIC. Again we gain over $3dB$ over the random collection of measurements. In this scenario BIC and AIC perform better than ML since the truth lies in the lower complexity class. The performances of AIC and BIC are very similar. Fig.6.4 shows that the error rate decreases with increasing numbers of iterations. The group of dashed lines shows the error rate when $DPNR = -0.31dB$ and the group of solid lines shows the error rate when $DPNR = 4.54dB$. As in the first example, The higher the DPNR, the greater the difference between the sequential method and the random method. When the DPNR is very low, the error rate decreases slowly and ML's performance is close to the random design performance in the beginning but gets better with increasing the number of iterations. For the random design, we only showed the behavior of BIC.

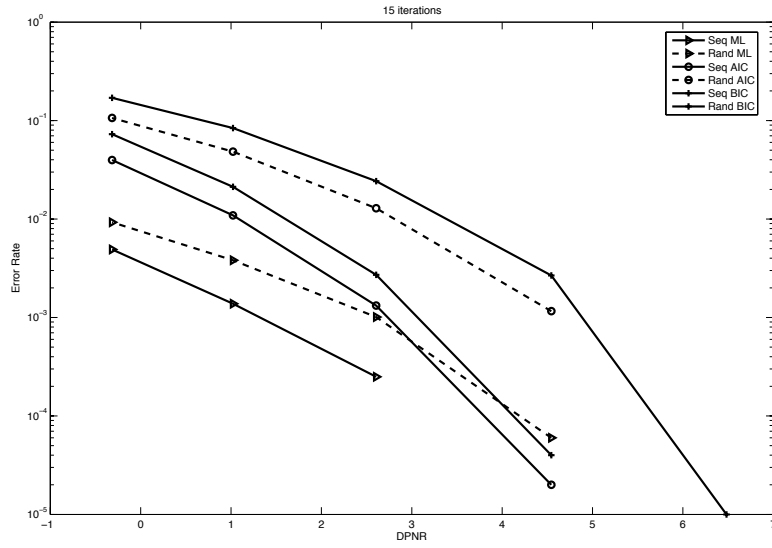


Figure 6.1: Error Rate versus DPNR. The solid lines correspond to the sequential design and the dashed lines correspond to the random design

We also checked the locations selected by the sequential algorithm; around 7 locations over the whole range. In the random method, the selected points are distributed uniformly everywhere within the range. This agrees with the intuition behind the sequential algorithm; it is picking the most informative points in terms of the discrimination between the models. Collecting measurements at these points is highly beneficial for the selection.

6.5 Path Planning

In the prior section, we considered the problem of finding the sampling points to optimize the model selection. This can be viewed as probing sensors, or placing static ones. The cost of deployment and sampling was ignored and the purpose was to find the most informative points (for selecting the correct model). The study of this prob-

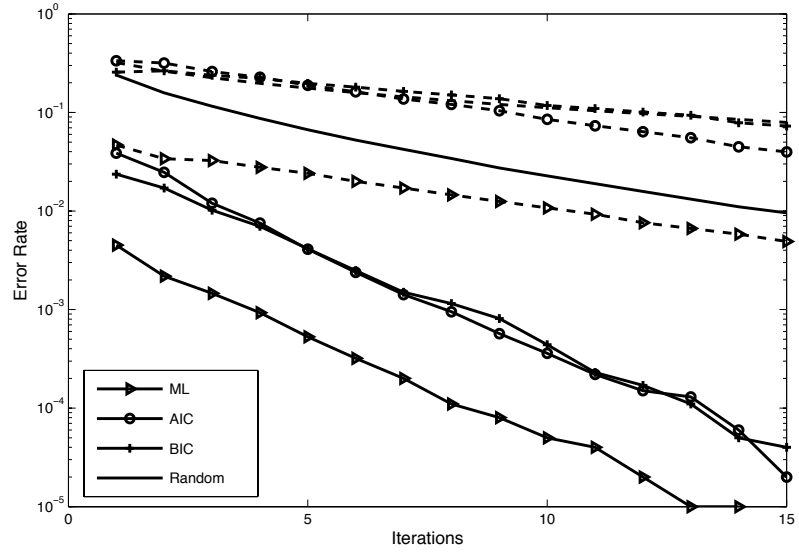


Figure 6.2: Error Rate behavior over iterations. The solid line corresponds to a DPNR=4.29dB and the dashed line corresponds to a DPNR=-0.56dB.

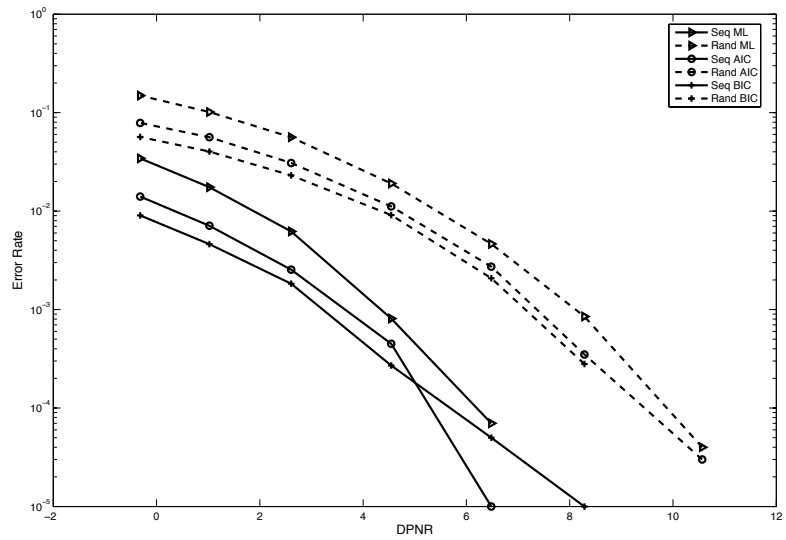


Figure 6.3: Error Rate versus DPNR. The solid lines correspond to the sequential design and the dashed lines correspond to the random design

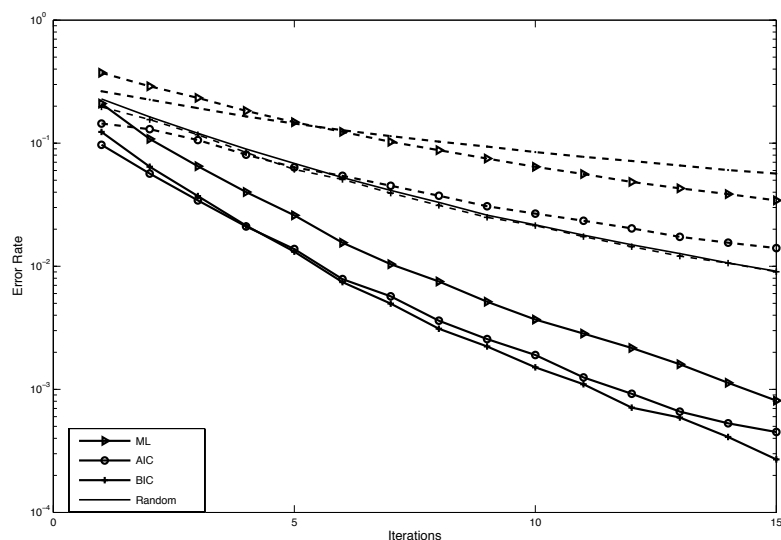


Figure 6.4: Error Rate behavior over iterations. The solid line corresponds to a DPNR=4.54dB and the dashed line corresponds to a DPNR=-0.31dB.

lem is very interesting for the theoretical limits of identifying the correct model. But practically, in many applications, the cost of deployment and sampling is high. In this section, we intend to take these costs into consideration through one example: the use of mobile sensors to sample the field. Now, the cost of mobility is included in the optimization of selecting the sampling points, and the problem becomes a path planning problem for the mobile sensors to optimize the selection of the correct model. Since the mobile sensors have finite energy to spend, we want to optimize the model selection with a constraint on the energy spent. It is easy to see that if we did not have an energy constraint, the mobile sensors should sample at the points selected by the algorithm above. But these points could be far from each other, and the cost of visiting them would be high and above the energy budget allowed. In this case, the cost of sampling is directly related to the distance traveled by the mobile sensors.

Moreover, in some applications, the cost of deployment is very high and mobility

is needed to achieve the required task. For example, in aquatic applications [ZS07], such as estimating some environmental process for a lake or a river, deploying (static) sensors is costly. So using the algorithm presented above requires us to deploy static sensors iteratively, to move existing ones, or to densely deploy sensors and probe them according to the algorithm. All these three options seem difficult and costly, while using mobile sensors makes the whole sampling design much easier and cheaper. That is, instead of iteratively deploying static sensors, moving existing ones, or densely deploying sensors and probing them, we use mobile sensors that can move and sample at the desired locations. Now the sampling problem becomes a path planning problem for the mobile sensors. For now, we looked at the problem of a single mobile sensor.

6.5.1 Related Work

The path planning problem has been studied extensively and many algorithms already exist. Many papers have been written in the context of sensor networks. To our knowledge all the work that has been done, was directed to the field estimation using different statistical models for the field. Adaptive path planning algorithms were developed to refine the estimation. The work of Zhang et al. [ZS07], considered linear local regression and minimized the Integrated Minimum Squared Error (IMSE) of the estimate. In [SBS07], the authors used Gaussian processes and maximized the mutual information between the path and the field. In [UK99], the authors used a distributed dynamic system and minimized the Frobenius condition number for the Hessian of the least squares in estimating the parameters of the dynamic system. In [Raf86], a distributed dynamic system was used as well, but the optimization was to minimize the determinant of the Hessian matrix. In all these papers, a single model was assumed and the paths of the mobile sensors were found to optimize the estimation of the model assumed. In our work, as said above, we consider the model selection problem. In other words, we are

adding one layer of uncertainty in incorporating uncertainties in models.

6.5.2 Our Path Planning Algorithm

The path planning algorithm we developed, is an extension of *Algorithm – 1*, where the only change is in the selection of the next point in *step3*. We added a distance penalization term, and we found the point that maximizes the discrimination plus a penalty term corresponding to the distance traveled:

$$z_{j+1} = \arg \max_{z \in Z} \left(h_1(z, \hat{a}_j) - h_2(z, \hat{b}_j) \right)^2 - c(z - z_j)^2 \quad (6.18)$$

The parameter c controls how much we are penalizing for the distance traveled. We ran three simulations for three values of c and the results are showed in Fig.6.5. Fig.6.5 shows three graphs corresponding to three values of c .

The algorithm presented is a greedy algorithm that optimizes the selection at each iteration and it is not clear yet what is happening globally. Nevertheless, it was just a preliminary algorithm for the bigger goal of formulating and solving the following problem:

$$\max \quad \textit{Discrimination} \quad (6.19)$$

$$\textit{s.t.} \quad \textit{Energy} < \textit{Budget} \quad (6.20)$$

In this work, we use a combination of static and mobile sensors. As we saw in the section above, the algorithm used needed an initialization phase where initial estimates of the models are computed. The static sensors in this section, are used to compute the initial estimates of the models, and the mobile one is used to optimize the selection. The initialization phase is done without any collection of measurements, so it is done

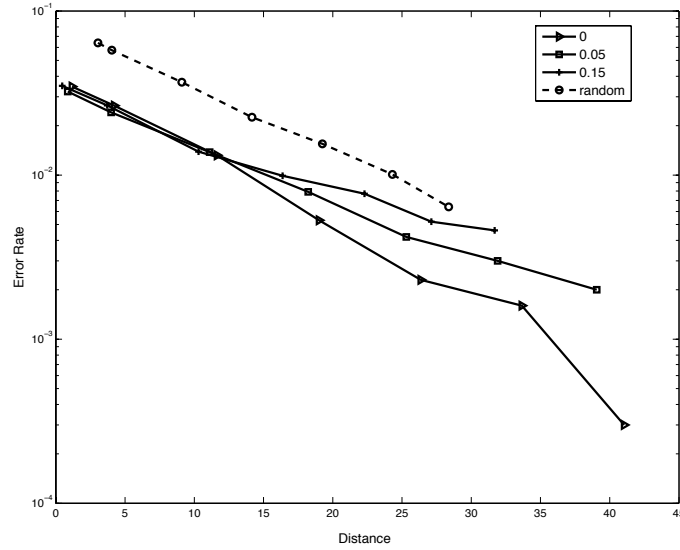


Figure 6.5: Error Rate versus distance traveled. DPNR=1.8dB

prior to any deployment and that is why we are using static sensors. The mobile sensors are used for the rest of the algorithm.

6.6 Conclusion

In this chapter, we presented two novel active learning algorithms for model selection in spatial modeling problems. In section 6.4.2, we present the first algorithm where we consider the discrimination between two linear regression models. The algorithm collects measurements sequentially to optimize the discrimination between the two models. This algorithm could be seen as an extension of the maximum likelihood framework where we don't only find the maximum likelihood model given the data, but we also find the data points that result in the best likelihood test by minimizing the probability of error in the model selection. In section 6.5, we present the second algorithm which is the path planning version of the first one. We add a term to account

for the cost of acquiring a measurement by mobile sensor, and we find the data points that result in the best likelihood test given a cost constraint. For example, the optimal data points could be far from each other such that a mobile sensor would need to travel long distances to collect the optimal measurements. Instead our path planning algorithm finds the points that result in a sub-optimal discrimination to satisfy the cost constraint. This work could be extended and applied to any sort of measurement cost, especially when there is model that relates the cost to the measurements.

Finally, this work finds its applications in the situations where the measurement acquisition is complicated or expensive, and where we need to find the minimum number of measurements to achieve a certain fidelity in modeling a spatial process. An example is under-water field modeling using a sensor network where the cost and effort in obtaining measurements are high, and we would like to find the minimum number of measurements required to find the correct model of the field.

CHAPTER 7

Conclusion

7.1 Conclusions

Present advances in technology are enabling us to monitor our life and the world around us. We now have tools to observe our environment on a large scale, monitor our health, and study our social interactions.

The work in this dissertation is motivated by these recent advances, and considers problems in environmental monitoring and activity recognition. In chapter 2, we presented a method to automate the creation of forest maps. We used a classification method for the vegetation type, based on texture analysis of color and infra red (CIR) aerial images. The problem was motivated by the need of the French national forest inventory (IFN) to update their forest maps. Our result show a great improvement over the current IFN maps, as shown in Figure 2.4 and Figure 2.5. In this problem we used the help of Mr. Jean-Guy Boureau from IFN, an expert on forests and vegetation types, to validate our results.

In chapter 3, we developed a system to recognize and classify the work place activities for people who work at a desk using two wearable accelerometers. This system classifies the activity into three categories common at the work place; walking, standing and sitting. It then monitors the sitting habits, and tracks the seated posture and movements that a worker might do at their desk. The system uses a multi-level supervised classifier using a naïve Bayes classifier at each level. We evaluate our system on

two subjects performing their daily activities at their desks.

In chapter 4, we extended the work of chapter 3 and built a system to recognize 14 common daily activities (shown in Table 4.2) using 14 sensors worn on the body (Figure 4.1). We used a tree-based classifier with a naïve Bayes classifier on each node, and we developed a feature selection algorithm to select the best features to use at each node. This time we collected a large data set from 8 people performing the 14 activities, and we labeled the data using an android application on a smartphone. We used a 2-fold cross validation, and got an average classification accuracy of 96%.

In chapter 5, we developed a system for the detection of stumbles using one accelerometer, that could be used by doctors for fall risk assessment. We used anomaly detection methods, and evaluated the system on large data set containing 100 stumbles and more than 45 minutes of walking collected from subjects. Our result shows that the chest is the best location for the accelerometer, and achieves a 99% detection accuracy with a 0.5% false alarm rate.

Finally in chapter 6, we presented a more theoretical result, where we developed active learning algorithms to optimize the model selection in the maximum likelihood framework.

Our final conclusion is that the design of large data-based classification systems is creating a new challenge; the validation on large data sets representative of the desired application. The validation process includes designing and conducting data collection campaigns, as well as labeling the data to create ground truth. We believe that the realization of these systems depends on the development of scalable and cost-sensitive methods for the validation process.

7.2 Future Work

In this section, we discuss the ideas we have for extending the work we presented in the previous chapters.

The first step in extending the work of chapters 3, 4, 5, is to conduct more experiments in uncontrolled environments and validate our systems on larger data sets. As mentioned throughout the dissertation, collecting testing data sets is challenging since it is hard to have a scalable procedure for labeling the datasets. We plan to investigate the use of semi-supervised learning where we use the classifiers obtained from the controlled experiments to create and label the larger scale and uncontrolled experiments. This requires an understanding of how the uncontrolled experiments are different from the control ones. This idea can be thought of as a multi-level approach where the control experiments are used to create an initial model and an initial classifier that can be used on part of the data from the uncontrolled experiments for the purpose of labeling it. This labeled part will then be used to update the models and classifiers and adapt them to the uncontrolled environment.

For the work of chapter 4, we plan to investigate the sensor selection problem, that is finding the best sensor placements on the body to achieve a desired classification accuracy. We believe that the feature selection algorithm we have can be extended to a sensor selection algorithm. Since our feature selection algorithm adds more features when we go from one level of the decision tree to another, we can add weights to the sensors to favor the features from an already selected sensor over the features from a non-selected sensor. This problem becomes about minimizing the number of sensors required for the activity classification.

For the work of chapter 5, a direct extension is to conduct more experiments and simulate more types of stumbles. We plan to collect data from patients who are suffer-

ing from gait problems, since they are the ones who have a high risk of falling. We also plan to use more powerful algorithms for the stumble detection, especially to make the pockets better placements in terms of the detection accuracy, since they are the easiest placements for accelerometers.

The work of chapter 6 can be extended by rigorously formulating the problem presented in equation 6.20. Another extension is to investigate the use of the algorithms of chapter 6 with the activity classification systems, in finding the features that optimize the discrimination between the activity classes.

REFERENCES

- [AF75] A.C. Atkinson and V.V. Fedorov. “The design of experiments for discriminating between two rival models.” *Biometrika*, **62**(1):57 – 70, 1975.
- [Aka74] H. Akaike. “A new look at the statistical model identification.” *Automatic Control, IEEE Transactions on*, **19**(6):716–723, Dec 1974.
- [ALK10] L. Atallah, B. Lo, R. King, and G. Yang. “Sensor Placement for Activity Detection Using Wearable Accelerometers.” *International Conference on Body Sensor Networks (BSN)*, 2010.
- [ARB99] K. Aminian, Ph. Robert, E. Buchser, B. Rutschmann, D. Hayoz, and M. Depairon. “Physical activity monitoring based on accelerometry: validation and comparison with video observation.” *Medical and Biological Engineering and Computing*, 1999.
- [Bat94] R. Battiti. “Using mutual information for selecting features in supervised neural net learning.” *IEEE Transactions on Neural Networks*, 1994.
- [BGG94] G.S. Bruss, A.M. Gruenberg, R.D. Goldstein, and J.P. Barber. “Hamilton Anxiety Rating Scale Interview guide: joint interview and test-retest methods for interrater reliability.” *Psychiatry research*, **53**(2):191202, 1994.
- [BGV92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. “A training algorithm for optimal margin classifiers.” *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, 1992.
- [BHG91] D. Blazer, D. Hughes, L.K. George, M. Swartz, and R. Boyer. “8 Generalized Anxiety Disorder.” *Psychiatric disorders in America: The epidemiologic catchment area study*, 1991.
- [BI04] L. Bao and S. Intille. “Activity Recognition from User-Annotated Acceleration Data.” *Pervasive Computing*, 2004.
- [BOL07] A.K. Bourke, J.V. O’Brien, and G.M. Lyons. “Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm.” *Gait and Posture*, **26**(2), 2007.
- [BRY04] M. Batalin, M. Rahimi, Y. Yu, D. Liu, A. Kansal, G. Sukhatme, W. Kaiser, M. Hansen, G. Pottie, M. Srivastava, and D. Estrin. “Call and response: Experiments in sampling the environment.” In *Proceedings of ACM SenSys*, 2004.

- [Bur98] C. J. C. Burges. “A tutorial on support vector machines for pattern recognition.” *Data Mining and Knowledge Discovery*, **2**:121–167, 1998.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [CBK09] V. Chandola, A. Banerjee, and V. Kumar. “Anomaly detection: A survey.” *ACM Computing Surveys*, **41**, July 2009.
- [CGJ96] D.A. Cohn, Z. Ghahramani, and M.I. Jordan. “Active learning with statistical models.” *Journal of Artificial Intelligence Research*, **4**:129–145, 1996.
- [CGP05] R. Cucchiara, C. Grana, A. Prati, and R. Vezzani. “Probabilistic posture classification for human-behavior analysis.” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **35**(1):4254, 2005.
- [CKH04] A. F. Cordero, H. J. F. M. Koopman, and F. C. T. van der Helm. “Mechanical model of the recovery from stumbling.” *Biological Cybernetics*, **91**:212–220, 2004.
- [CT06] T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2006.
- [CWN05] R. Castro, R. Willett, and R. Nowak. “Faster rates in regression via active learning.” In *Proceedings of NIPS*, 2005.
- [FCB06] M. Fauvel, J. Chanussot, and J.A. Benediktsson. “Evaluation of kernels for multiclass classification of hyperspectral remote sensing data.” *International Conference on Acoustics, Speech and Signal Processing*, May 2006.
- [FOC11] M. Faulkner, M. Olson, R. Chandy, J. Krause, K.M. Chandy, and A. Krause. “The next big one: Detecting earthquakes and other rare events from community-based sensors.” In *10th International Conference on Information Processing in Sensor Networks (IPSN)*, april 2011.
- [GLS96] R. Jan Gurley, Nancy Lum, Merle Sande, Bernard Lo, and Mitchell H. Katz. “Persons Found in Their Homes Helpless or Dead.” *New England Journal of Medicine*, 1996.
- [Har79] R. M. Haralick. “Statistical and structural approaches to texture.” *Proceedings of the IEEE*, **67**(5):786–804, 1979.

- [HBD08] M.F. Huber, T. Bailey, H. Durrant-Whyte, and U.D. Hanebeck. “On entropy approximation for Gaussian mixture random vectors.” In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2008.
- [HBV09] N. Hajj Chehade, J.-G. Boureau, C. Vidal, and J. Zerubia. “Multi-class SVM for forestry classification.” In *the 16th IEEE International Conference on Image Processing (ICIP)*, pp. 1673 –1676, November 2009.
- [HL02] C. W. Hsu and C. J. Lin. “A comparison of methods for multiclass support vector machines.” *IEEE Transactions on Neural Networks*, **13**(2):415–425, 2002.
- [HTF01] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.
- [htt] Gulf Coast Data Concepts: <http://www.gcdataconcepts.com>.
- [ifn] “Inventaire Forestier National.” www.ifn.fr.
- [IMC06] F. M. Impellizzeri, S. M. Marcora, C. Castagna, T. Reilly, A. Sassi, F. M. Iaia, and E. Rampinini. “Physiological and performance effects of generic versus specific aerobic training in soccer players.” *International Journal of Sports Medicine*, 2006.
- [JWC08] J.Y. Jhun-Ying Yang, J.S. Wang, and Y.P. Chen. “Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers.” *Pattern Recognition Letters*, 2008.
- [KGG06] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. “Near-optimal Sensor Placements: Maximizing Information while Minimizing Communication Cost.” In *Proceedings of Information Processing in Sensor Networks (IPSN)*, April 2006.
- [KMS07] M. S. Kulikova, M. Mani, A. Srivastava, X. Descombes, and J. Zerubia. “Tree Species Classification Using Radiometry, Texture and Shape Based Features.” In *Proc. European Signal Processing Conference (EUSIPCO)*, 2007.
- [KR92] Kenji Kira and Larry A. Rendell. “A practical approach to feature selection.” *Proceedings of the ninth international workshop on Machine learning*, 1992.
- [KSS] N. Kern, B. Schiele, and A. Schmidt. “Multi-sensor Activity Context Detection for Wearable Computing.” In *Ambient Intelligence*.

- [Lab] Occupational Safety & Health Administration U.S. Department of Labor.
- [LCB05a] J. Lester, T. Choudhury, and G. Borriello. “Sensing and modeling activities to support physical fitness.” In *Proceedings of UbiComp*, 2005.
- [LCB05b] J. Lester, T. Choudhury, G. Borriello, S. Consolvo, J. Landay, K. Everitt, and I. Smith. “Sensing and Modeling Activities to Support Physical Fitness.” *Proceedings of UbiComp*, 2005.
- [LCB06] J. Lester, T. Choudhury, and G. Borriello. “A Practical Approach to Recognizing Physical Activities.” *Pervasive Computing*, 2006.
- [Lew98] D. Lewis. “Naive (Bayes) at forty: The independence assumption in information retrieval.” In *Machine Learning: ECML-98*, volume 1398 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 1998.
- [LVS10] B.E. Lawson, H.A. Varol, F. Sup, and M. Goldfarb. “Stumble detection and classification for an intelligent transfemoral prosthesis.” In *Proceedings of the 32nd Annual International Conference of the IEEE EMBS*, 2010.
- [Mac92] D. MacKay. “Information-Based Objective Functions for Active Data Selection.” *Neural Computation*, 4(4):590–604, 1992.
- [MB04] F. Melgan and L. Bruzzone. “Classification of hyperspectral remote sensing images with support vector machine.” *IEEE Trans. Geoscience and Remote Sensing*, 42(8):1778-1790, 2004.
- [MKZ09] A. Mueen, E. Keogh, Q. Zhu, and S. Cash. “Exact Discovery of Time Series Motifs.” In *SDM*, 2009.
- [MP03] Selene Mota and Rosalind W. Picard. “Automated Posture Analysis for Detecting Learner’s Interest Level.” In *Computer Vision and Pattern Recognition Workshop*, volume 5, p. 49, June 2003.
- [NDA10] H. Noshadi, F. Dabiri, S. Ahmadian, N. Amini, H. Hagopian, and M. Sarrafzadeh. “HERMES: Mobile System for Instability Analysis and Balance Assessment.” *ACM Transactions on Embedded Computing Systems (TECS)*, 2010.
- [NFR07] N. Noury, A. Fleury, P. Rumeau, A.K. Bourke, G.O. Laighin, V. Rialle, and J.E. Lundy. “Fall detection - Principles and Methods.” In *Proceedings of the 29th Annual International conference of IEEE Engineering in Medicine and Biology Society*, 2007.

- [NLB07] M.R. Narayanan, S.R. Lord, M.M. Budge, B.G. Celler, and N.H. Lovell. “Falls Management: Detection and Prevention using a Waist-mounted Tri-axial Accelerometer.” In *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, 2007.
- [PLD05] H. Peng, F. Long, and C. Ding. “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005.
- [Poo98] H. V. Poor. *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1998.
- [Puk93] F. Pukelsheim. *Optimal Design of Experiments*. Wiley, 1993.
- [Raf86] E. Rafajlowicz. “Optimum choice of moving sensor trajectories for distributed parameter system identification.” *Journal of Control*, **43**, 1986.
- [RE01] S. Roberts and R. Everson. *Independent component analysis: principles and practice*. Cambridge University Press, 2001.
- [Rei93] P.A. Reilly. “Fibromyalgia in the workplace: a management problem.” *Annals of the rheumatic diseases*, **52**(4), 1993.
- [REW09] R. Ratner, J. Eden, D. Wolman, S. Greenfield, and H. Sox. *Institute of Medicine: initial national priorities for comparative effectiveness research*. National Academies Press, 2009.
- [RHK05] M. Rahimi, M. Hansen, W. Kaiser, G. Sukhatme, and D. Estrin. “Adaptive sampling for environmental field estimation using robotic sensors.” In *Proceedings of The IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.
- [RLD11] N. Ruchansky, C. Lochner, E. Do, T. Rawls, N. Hajj Chedahde, J. Chien, G. Pottie, and W Kaiser. “Monitoring Workspace Activities Using Accelerometers.” *Proceedings of ICASSP*, 2011.
- [SBS07] A. Singh, M. Batalin, M. Stealey, V. Chen, M. Hansen, T. Harmon, G. Sukhatme, and W. Kaiser. “Mobile Robot Sensing for Environmental Applications.” In *Proceedings of The 6th International Conference on Field and Service Robotics*, July 2007.
- [SWD96] A.M. Schillings, Van Wezel, and J. Duysens. “Mechanically Induced Stumbling During Human Treadmill Walking.” *Journal of Neuroscience Methods*, **67**:11–17, 1996.

- [SWM00] A. M. Schillings, B. M. H. Van Wezel, Th. Mulder, and J. Duysens. “Muscular Responses and Movement Strategies during Stumbling over Obstacles.” *Journal of Neurophysiology*, **83**:2093–2102, 2000.
- [TPS05] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. “A macroscope in the redwoods.” In *Proceedings of ACM SenSys*, pp. 51–63, 2005.
- [TSP01] H.Z. Tan, L.A. Slivovsky, and A. Pentland. “A sensing chair using pressure distribution sensors.” *IEEE/ASME Transactions On Mechatronics*, **6**(3):261268, 2001.
- [UK99] D. Ucinski and J. Korbicz. “Path Planning for Moving Sensors in Parameter Estimation of Distributed Systems.” In *Proceedings of the First Workshop on Robot Motion and Control*, 1999.
- [Vap95] V. N. Vapnik. *The Nature of Statistical Learning*. New York: Springer-Verlag, 1995.
- [WMB07] B. Waske, G. Menz, and J.A. Benediktsson. “Fusion of support vector machines for classifying SAR and multispectral imagery from agricultural areas.” *International Geoscience and Remote Sensing Symposium*, pp. 4842–4845, July 2007.
- [WSG10] A. Weiss, I. Shimkin, N. Giladi, and J. Hausdorff. “Automated detection of near falls: algorithm development and preliminary results.” *BMC Research Notes*, **3**(1):62, 2010.
- [WWD07] D.K. White, R.C. Wagenaar, M.E. Del Olmo, and T.D. Ellis. “Test-retest reliability of 24 hours of activity monitoring in individuals with Parkinsons disease in home and community.” *Neurorehabil Neural Repair*, 2007.
- [XBK11] X. Xu, M. A. Batalin, W. J. Kaiser, and B. Dobkin. “Robust Hierarchical System for Classification of Complex Human Mobility Characteristics in the Presence of Neurological Disorders.” In *International Conference on Body Sensor Networks (BSN)*, pp. 65 –70, may 2011.
- [ZCL11] Zhongtang Zhao, Yiqiang Chen, Junfa Liu, Zhiqi Shen, and Mingjie Liu. “Cross-People Mobile-Phone Based Activity Recognition.” *International Joint Conference on Artificial Intelligence*, 2011.
- [ZDN11] F Zhang, S.E. D’Andrea, M.J. Nunnery, S.M. Kay, and H. Huang. “Towards Design of a Stumble Detection System for Artificial Legs.” *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, **19**(5):567–577, oct. 2011.

- [ZDZ07] O. Zammit, X. Descombes, and J. Zerubia. “Support Vector Machines for burnt area discrimination.” Research Report 6343, INRIA, November 2007.
- [ZS07] B. Zhang and G.S. Sukhatme. “Adaptive sampling for estimating a scalar field using a robotic boat and a sensor network.” In *Proceedings of IEEE Int. Conf. Robot Autom.*, 2007.