dtFall – Decision-Theoretic Framework to Report Unseen Falls

Shehroz Khan*
University of Waterloo
200 University Ave W,
Waterloo, ON N2L 3G1, Canada
s255khan@uwaterloo.ca

Jesse Hoey
University of Waterloo
200 University Ave W,
Waterloo, ON N2L 3G1, Canada
jhoey@uwaterloo.ca

ABSTRACT

Automated systems to report falls have long been sought. However, it is very difficult to train classifiers for falls as these are rare events that are difficult to gather training data for. Further, the costs associated with false alarms and missed alarms are not very well known or understood. In this paper, we present a decision-theoretic framework to fall detection (*dtFall*) that aims to tackle the core problem of when to report a fall, given an arbitrary amount (possibly zero) of training data for falls, and given little or no information about the costs associated with falls. We derive equations for the expected regret (for not using a decision-theoretic approach), and present a novel method to parameterize unseen falls, such that we can accommodate training situations with no fall data. We identify problems with theoretical thresholding to identify falls using decision-theoretic modeling when training data for fall data is absent, and present a modified empirical thresholding technique to handle imperfect models for falls and non-falls. We present results on two activity recognition datasets and show that knowing the difference in the cost between a reported fall and a false alarm is useful, as the cost of false alarm gets bigger this becomes more significant. The results also show that the difference in the cost of between a reported and non-reported fall is not that useful.

CCS Concepts

•**Computing methodologies** → **Machine learning; Cost-sensitive learning; Anomaly detection;** *Mixture modeling;* •**Networks** → *Sensor networks;*

Keywords

Fall Detection; Decision Theory; Mixture Modelling; One-Class Classification

1. INTRODUCTION

*Corresponding Author.

Pervasive Health '16 May 16–19, 2016, Cancun, Mexico
© 2016 ACM. ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

Activity recognition aims to identify both normal and abnormal activities of an individual with an aim to provide some sort of assistance [1]. One of the most common abnormal activity is incurring a *fall*, which is also the most common cause of both fatal and nonfatal injuries among older adults [3]. However, falls occur rarely, infrequently and unexpectedly in comparison to normal activities. The Centers for Disease Control and Prevention, USA [3], suggests that on an average, nursing home residents incur 2.6 falls per person per year. If an activity is monitored every second, then we get around 31.5 million non-fall (normal) activities per year. This high skew in the training data makes it difficult to develop generalizable classifiers to identify falls. The approaches that exclusively collect fall data still suffer from their limited quantity, artificially induced falls and ethics clearances.

In this paper, we argue that the traditional classification approaches to tackle fall detection that includes thresholding techniques, supervised machine learning methods and one-class classification (OCC) / outlier detection methods are not well-posed for this problem. The primary reason being unavailability or lack of sufficient training data for falls. Unlike the conventional classification methods, incorrectly identifying a normal activity as a fall or vice-versa should not be treated with equal cost, moreover this cost is mostly unknown and hard to compute. The question that the traditional methods for fall detection seek to answer is "Is an action a fall?"; however, the research question we address in this paper is "Is it right to report an action as a fall?". To answer this question, this paper presents a **d**ecision **t**heoretic framework for **fall** detection, *dtFall*, based on expected utility theory (EUT) that introduces a global utility function to encode prior knowledge about falls and normal activities and utilities of reporting/not-reporting a fall/non-fall activity. Firstly, we compare EUT method with the traditional maximum likelihood (ML) classifier and theoretically show that (a) the expected value to report/not-report a fall using EUT will always be larger or the same as ML, and (b) ML method is a special case of EUT. We also present a novel method to identify falls, when their training data is unavailable, by parameterizing falls and integrating it out using a prior distribution that enables to estimate the expected likelihood of unseen falls. The probabilistic models learned for fall detection may not represent the true distributions of falls/non-falls and may not be expressive enough due to limited or unclean training data, underlying assumptions and parameters of the algorithm. Due to these issues, the theoretical threshold may not provide the desired results. To tackle this issue, we modify an empirical thresholding algorithms that can deduce a probabilistic threshold from the training data for *dtFall*. We perform an experimental evaluation for both the *EUT* and *ML* approaches on two activity recognition datasets and show how the results may generalize to the difficult problem of fall detection in the real world using the proposed decision-theoretic formulation.

2. RELATED WORK

Most of the research on fall detection is focused on modelling activities of daily living with wearable devices, ambient devices and vision based sensors, and employing either thresholding techniques, supervised classification or one-class classification (OCC) [10] / outlier detection methods [7]. Thresholding methods for fall detection are simple; however, fixed thresholds lack generalization and adapting to new data can be detrimental to these classifiers. Both supervised and thresholding methods assume availability of sufficient training data for normal activities and falls. As discussed in the previous section, falls occur rarely; therefore, presuming sufficient amount of fall data is an unrealistic assumption. OCC methods that build one-class classifiers using only fall data (as the positive class) assume that sufficient fall data is present which is difficult to glean. On the other hand, the performance of OCC methods that use non-fall data to build their models is severely marred due to their dependence on model parameters that can result in lack of generalization to adapt to new types of activities.

An important aspect of fall detection, which is absent in most of the studies is the incorporation of cost of classification, in the best case the cost of errors are considered equal. An effort in identifying rare activity, such as falls, indicates their importance, relevance and criticality. Identifying such important activities correctly and miss-classifying them should not carry equal cost; however, such cost can be hard to compute and should not be data-dependent. Maloof [11] present a technique to deal with skewed datasets and unequal but unknown costs of error by performing ROC analysis to find the optimal operating threshold. However, the selection of appropriate decision threshold is not automatic and it is unclear if this technique will work in the case of OCC; when the data for the class of interest is absent. Huang et al. [6] perform cost-sensitive analysis for fall detection using Bayesian minimum risk and the Neyman-Pearson method. They vary the ratio of cost of miss alarm to false alarm to find an optimal region of operation using the ROC curve. On the contrary, this ratio is generally fixed and must not depend on the dataset. The technique presented in the paper to estimate cost ratio can overfit the dataset without providing any intuitive interpretation about it.

There is very sparse literature on decision-theoretic methods for identification of outliers or rare events/activities. Decision-theoretic approaches have been applied in some tasks including detecting anomalies in internet path [4], intrusion detection [12], fault detection in wireless sensor networks [16]. Fida et al. [4] propose an algorithm to decide the class of bandwidth in internet path based on the likelihood ratio. The normal internet responses are modelled using a Gaussian distribution, whereas the anomalous activities are modelled with a different mean than normal. Based on user defined true positive and false positive rates they define thresholds to detect both hypotheses. Nandi et al. [16] define an overall risk function for fault detection on sensor networks and seek a Bayes test which minimizes the overall risk function in the critical region. Decisions are then made purely on the optimal Bayes test. Our decision-theoretic framework for fall detection differs from these methods in the following ways:

1. We do not restrict the form of the likelihood function, hence it will be more flexible to be applied on different data sets.
2. Unlike likelihood ratio, which can be biased when the data set is relatively small, a global utility function is introduced to encode prior knowledge about fall detection.
3. When fall data is unavailable during training, we take a Bayesian approach to average over the parameter space of all possible likelihood functions.

3. DECISION-THEORETIC FRAMEWORK

The *EUT* states that when people make a choice between risky outcomes, they will choose the 'rational' option that maximizes their expected utility [14]. The expected utility (or value) V of an option O is defined as:

$$V(O) = \sum_i p_i u(x_i) \tag{1}$$

where p_i is the probability of outcome x_i and $u(.)$ is a utility function that defines the subjective utility of x_i.

3.1 Formulation for Fall Detection

A fall detection system's job is to report a fall and remain passive otherwise. Let us use R to denote a binary decision variable where $R = r$ means the report is made (that an action is a fall) and $R = \bar{r}$ means there is no report made. Similarly, let F denote a binary random variable where $F = f$ means there is a fall and $F = \bar{f}$ means there is no fall. There are; therefore, four different situations one needs to consider, as shown in the following utility table (see Table 1), where $U(F, R)$ gives the utility for the outcome F after the decision R.

R	F	U(F,R)	Remark
$\bar{r}$	f	0	*Miss Alarm*
r	$\bar{f}$	q	*False Alarm*
r	f	p	*True Positive*
$\bar{r}$	$\bar{f}$	1	*True Negative*

Table 1: Utility Table.

We have deliberately set the $U(F = \bar{f}, R = \bar{r}) = 1$ for the best possible situation (there is no fall and no report), and $U(F = f, R = \bar{r}) = 0$ for the worst (there is a fall and it is not reported). The other two utilities will be somewhere in between, to be determined through some preference elicitation mechanism or expert knowledge. Using Table 1 and Equation 1 we compute the expected value of generating a report ($R = r$) given an observation o as:

$$V(R = r|o) = Pr(\bar{f}|o)U(\bar{f}, r) + Pr(f|o)U(f, r)$$

Applying Bayes' theorem, we get

$$V(R=r|o) = \frac{1}{Pr(o)}[Pr(o|\bar{f})Pr(\bar{f})q + Pr(o|f)Pr(f)p] \quad (2)$$

and the expected value of not generating a report ($R = \bar{r}$) given an observation o as:

$$V(R=\bar{r}|o) = \frac{1}{Pr(o)}Pr(o|\bar{f})Pr(\bar{f}) \quad (3)$$

Let $\mathbb{D}(o)$ be a decision function which maps an observation o to a binary $[0,1]$ representing the decision to [report,not report], respectively. For example, a simple threshold function on the posterior over falls would be:

$$\mathbb{D}(o) = \begin{cases} 1, & Pr(f|o) \geq \tau \\ 0 & otherwise \end{cases} \quad (4)$$

Then, the expected value for applying this decision function for observation o is

$$Q(o) = \frac{1}{Pr(o)}([Pr(o|\bar{f})Pr(\bar{f})q + Pr(o|f)Pr(f)p]\mathbb{D}(o) + Pr(o|\bar{f})Pr(\bar{f})[1 - \mathbb{D}(o)]) \quad (5)$$

3.2 Maximum Likelihood Decision Function

The traditional supervised method for reporting falls is based on the normalized posterior probability of falls and non-falls and not on the expected value of generating a report/not-report. We call it the Maximum Likelihood (ML) approach. In this case, $\tau = 0.5$ in Equation 4, and the decision surface is a horizontal plane (independent of p and q in Figure 1).

3.3 Expected Utility Decision Function

The rational decision is to maximize over the expected values, and so the decision surface can be deduced by equating $V(R=r|o) = V(R=\bar{r}|o)$, and setting $Pr(f|o) = 1 - Pr(\bar{f}|o)$, we get

$$Pr(f|o) = \frac{1}{1 + \frac{p}{1-q}}$$

Thus, in the decision function in Equation 4, $\tau = \frac{1}{1+\frac{p}{1-q}}$, and we call this the EUT approach. The decision surface is now curved in Figure 1 and we can see regions where the decisions will be different than for the ML case. Specifically, when $p < (1-q)$, the EUT approach will report less falls, whereas the opposite is true when $p > (1-q)$. When $p = (1-q)$, the two decision functions are the same; therefore, ML becomes a limiting case of EUT. We term this decision-theoretic framework for fall detection as *dtFall*.

3.4 Regret

We define the expected regret for taking a decision based on the ML decision function (rather than the EUT one) as the difference between the two value functions given by Equation 5. Denoting $Q_S(o)$ the value when using $\mathbb{D}(o)$ (ML) and $Q_L(o)$ the value when using $\mathbb{D}(o)$ (EUT), we have the regret for a particular value of p and q defined as

$$regret(p,q,o) = Q_L(o) - Q_S(o) \quad (6)$$

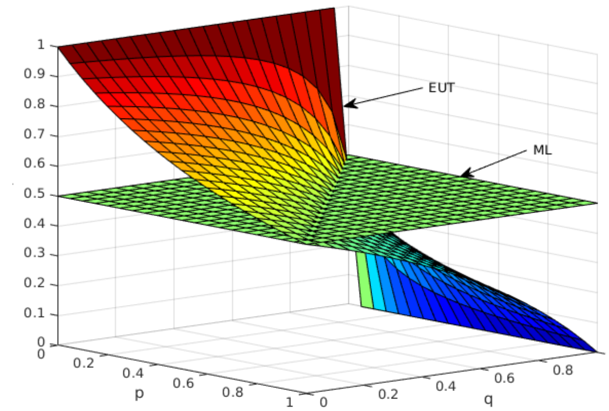


Figure 1: Decision surface for EUT and ML classifier.

THEOREM 3.1. *The regret, $regret(p,q,o)$, is always greater than or equal to zero.*

PROOF. *Lets us denote $x = Pr(f|o)$ and $\tau = \frac{1}{1+\frac{p}{1-q}}$. In Figure 1, there are four regions:*

1. $x > 0.5, x > \tau$: *regret* $= [(1-x)q + xp] * (1-1) + (1-x) * (0-0) = 0$
2. $x > 0.5, x <= \tau$: *regret* $= [(1-x)q + xp] * (-1) + (1-x) * 1 > 0$, *since* $x <= \tau$
3. $x <= 0.5, x <= \tau$: *regret* $= [(1-x)q + xp] * (0-0) + (1-x) * (1-1) = 0$
4. $x <= 0.5, x > \tau$: *regret* $= [(1-x)q + xp] * 1 + (1-x)(-1) > 0$, *since* $x > \tau$

□

The positive regret means that it will be always worthwhile to figure out what p and q are and take the corresponding rational decision. However, this will only be true if we have a correct observation model $Pr(o|f)$ and $Pr(f)$. There are two reasons we may not, and the EUT approach may fail. The first is if the model is incorrectly estimated from the data, due to insufficient training data for one of the classes (e.g. falls) and this may cause the estimation to be biased towards the other classes. The second is if the model is insufficiently expressive to separate falls from non-fall data. We consider both of these issues in Section 4 and present solutions to tackle them.

3.5 Decision-making without training data for falls

The previous section builds the concept of *dtFall* when sufficient data is available for both falls and non-fall activities. As discussed previously, falls occur rarely and infrequently; therefore, in a realistic setting we may have no training data for falls. That is, we are considering the OCC case where we have lots of training examples for $F = \bar{f}$, but none for $F = f$ (see Table 1). Thus, we have some estimate of $Pr(o|\bar{f})$ but we don't for $Pr(o|f)$. Taking a Bayesian approach, we characterize our uncertainty about this function with a set of parameters θ_f describing a model, such that

$Pr(o|f) = Pr(o|f, \theta_f)$. We then propose a prior distribution over the model parameters, $Pr(\theta_f)$, and compute the expected value by summing over all the values that θ_f can take. Therefore, the expected value to report can be written as:

$$V_E(R = r|o, \theta_f) = \frac{1}{Pr(o)}[Pr(o|\bar{f})Pr(\bar{f})q+ \\ \mathbb{E}_{\theta_f}\left[Pr(o|f, \theta_f)\right]Pr(f)p] \tag{7}$$

$V_E(R = \bar{r}|o, \theta_f)$ remains unaffected by θ_f because it only uses information from non-fall data (although we could also parameterize the non-falls model and integrate out as well). In Section 5.1, we present a specific case of using Gaussian Mixture Model (GMM) to calculate the expected likelihood for unseen falls using only the data from normal activities.

4. THRESHOLD OPTIMIZATION

The EUT method works well when the true distribution for falls and normal activities is known. The decision-theoretic probability threshold, $\tau = \frac{1}{1+\frac{p}{1-q}}$, to take an action with maximum utility is also derived under the same assumptions. However, in a real-world scenario these assumptions may not hold good due to limited training data, limitations of learning algorithm and underlying assumptions regarding the model and its parameters, spurious sensor data and labelling errors. Therefore, instead of a true model, we may learn an impoverished model for the training data that may not be expressive enough and may not provide accurate estimation of true probabilities of the models for falls and non-falls. We consider a case when falls are not available during training; therefore, the parameterized expected likelihood of the unseen falls ($\mathrm{E}_{Pr(\theta_f)}\left[Pr(o|f, \theta_f)\right]$, see Equation 7) may not represent the actual likelihood of falls. The posterior probability estimates, thus obtained may be biased and over-estimated in comparison to the true model for falls. We now define regret of using EUT instead of ML for this case and discuss situations when theoretical threshold may not be the right choice.

4.1 Regret

The scenario we consider here is that there is no training data for falls during training; however, some falls may be available during testing along with sufficient data for normal activities. The datasets collected in laboratory settings can contain many instances of real or simulated falls along with other normal activities. Our experimental method consists of splitting these datasets into training and test sets (see Section 6), and then building classifiers on the training sets, computing decisions (based on ML and EUT decision functions) on the test sets, and then using the results to estimate the expected regret incurred in a real situation where falls occur infrequently. The normal activities and fall data available for testing the models may contain many more falls (and less non-falls) than one would expect in a real situation. Therefore, for cost sensitive classification during testing phase, it needs to be re-scaled with the actual fraction of falls and non-falls expected in real data.

Let us now define,

(i) $\Delta U(f, r)$ – the difference in the number of reported falls (true positives) in the experimental test set between EUT and ML.

(ii) $\Delta U(\bar{f}, r)$ – the difference in the number of reported non-falls (false alarms) in the experimental test set between EUT and ML.

(iii) $\Delta U(\bar{f}, \bar{r})$ – the difference in the number of not-reported non-falls (true negatives) in the experimental test set between EUT and ML.

It is to be noted that the absolute value of the difference of false alarms between EUT and ML is the same as the absolute value of the difference of true negatives between them.

Now, we define $regretUtility_{p,q}$ as the expected regret of using the EUT instead of the ML decision function in a real situation with α falls and β non-falls s.t. $\beta \gg \alpha$. We compute this using the expected regret on the experimental dataset, so that

$$\begin{aligned} regretUtility_{p,q} &= \text{Regret for falls}_{pq} + \text{Regret for non-falls}_{pq} \\ &= \frac{\Delta U(f,r)p\alpha}{N_f} + \frac{\left(\Delta U(\bar{f},r)q + \Delta U(\bar{f},\bar{r})\right)\beta}{N_{\bar{f}}} \end{aligned} \tag{8}$$

where N_f and $N_{\bar{f}}$ are the number of falls and non-falls in the experimental test set. We use the average expected regret (across all subjects) as a metric to evaluate the performance of the EUT and ML methods. We assume that in the test set, both $N_f > 0$ and $N_{\bar{f}} > 0$, otherwise $regretUtility_{p,q}$ will be undefined.

4.2 Negative Regret

The decision regions for theoretical thresholds for ML and EUT in Figure 1 are based on the assumption that the true models for falls and non-falls are learned from sufficient data. As discussed earlier, in a real world scenario there may be only limited training data available for non-falls and no training data for falls. Therefore, the models learnt can be impoverished, less expressive and the probability estimates may be biased. In such cases, a situation can arise when the regret may not remain positive. This can happen when EUT wrongly reports a normal activity as a fall instead of not-reporting it, whereas ML does not report it. Therefore, EUT will have less not-reported non-falls than ML, which means that in the expression for regret in Equation 8, $\Delta U(\bar{f}, \bar{r}) < 0$, when multiplied by β (and $q \neq 1$) it will result in negative regret.

4.3 Empirical Threshold

Due to the problems associated with the theoretical threshold, empirical adjustment is needed for it. Sheng and Ling [18] present a threshold adjusting method, *Thresholding*, for selecting an empirical threshold from the training instances according to the misclassification cost. This method can convert any cost-insensitive algorithm to cost-sensitive one by doing an internal cross-validation step that looks for a threshold in the probability of an observation given each class and optimizes that by using an exhaustive search over all possible thresholds. The *Thresholding* method is least sensitive to the high difference in misclassification costs and does not require accurate estimation of probabilities, rather an accurate ranking is sufficient. Since falls occur rarely; in a supervised case during training and testing we expect very few falls but sufficient non-fall activities. Therefore, we need to re-scale both falls and non-falls by factors α and β to make the classifier cost-sensitive. Considering this severely skewed scenario, we present a **m**odification to the **Th**resholding algorithm (*mTh*) that simulates a real scenario by choosing an empirical threshold from the training data. The *mTh* algorithm re-scales the training data by factors α and β in the internal cross-validation step to compute the utility corresponding to every probability threshold in the training set followed by maximization as:

$$Utility_i = \frac{TPp\alpha}{N_f} + \frac{(FPq + TN))\,\beta}{N_{\bar{f}}} \tag{9}$$

where TP, FP and TN are the number of true positives, false positive and true negatives in the training data (falls is the positive class), $i = 1, \ldots, \mathbb{N}$ where $\mathbb{N}$ is the number of training instances. The probability threshold with the maximum value of $Utility_i$ is used as a threshold to identify falls. The *mTh* algorithm can be adapted to *ML* by maximizing over geometric mean as a performance metric [9] because it does not use expected utility.

4.3.1 OCC Case

The *mTh* algorithm discussed above cannot be directly applied in the OCC case due to the absence of fall samples in the validation set of the internal cross validation step for optimizing the probability threshold. Khan et al. [9] present a method to reject outlier data from the normal activities using inter-quartile range (IQR) and use them as proxy for falls to estimate the parameters of the unseen falls. We adapt their technique for the GMM by calculating the log-likelihood of the non-fall training instances and setting a user-defined threshold to reject a small percentage of data points from the non-fall class. Given the likelihoods of the training non-fall data for a model, the lower quartile (Q_1), the upper quartile (Q_3) and the inter-quartile range ($IQR = Q_3 - Q_1$), a point P is qualified as an outlier of the non-fall class, if

$$P > Q_3 + \omega \times IQR \quad || \quad P < Q_1 - \omega \times IQR \tag{10}$$

where ω represents the percentage of data points that are within the non-extreme limits. These rejected outliers may not be actual falls; however, they are seen as deviations from the non-fall activities, serve as representative for falls and can be plugged into the *mTh* algorithm. In this paper, ω is set to 1.732 which corresponds to 0.01% of the normal data to be rejected as outliers.

5. MODELLING UNSEEN FALLS

The previous section discusses the applicability of *dtFall* for the case when the data for falls is not present in the training set. Note that *dtFall* posits no restriction on the form of $Pr(o|f, \theta_f)$ and may expect to observe a few falls along with normal activities during the testing phase. In the following section, we will show a method to compute expected likelihood for unseen falls using only the training data for non-fall activities.

5.1 Mixture of Gaussians

We now examine a particular case of a GMM for modelling non-falls, and a particular prior distribution over model parameters to compute $\mathbb{E}_{\theta_f}[Pr(o|f, \theta_f)]$ (see Equation 7) for unseen falls. We propose to model unseen falls by using the X-factor approach [17, 8], which differs from the model for non-fall data only in the variance and the mean remains the same as non-falls. Assuming the observations $\mathbf{O} \in \mathbb{R}^n$, let all the non-fall activities be modelled by a GMM with K Gaussian mixtures [2]:

$$Pr(\mathbf{O}|\bar{f}) = \sum_{k=1}^{K} w_k \frac{1}{\sqrt{(2\pi)^n |\Sigma_k|}} e^{-\frac{(\mathbf{O}-\mu_{\mathbf{k}})^T \Sigma_k^{-1} (\mathbf{O}-\mu_{\mathbf{k}})}{2}}$$

For Gaussian X-factor model, falls activity can be modelled as followed:

$$Pr(\mathbf{O}|f) = \sum_{k=1}^{K} w_k \frac{1}{\sqrt{(2\pi)^n |\theta_{fk}\Sigma_k|}} e^{-\frac{(\mathbf{O}-\mu_{\mathbf{k}})^T \Sigma_k^{-1} (\mathbf{O}-\mu_{\mathbf{k}})}{2\theta_{fk}}}$$

where $\theta_{\mathbf{f}} = (\theta_{f1}, \ldots, \theta_{fK})^T$ is the model parameter, each $\theta_{fk} \in [1, \infty], k = 1, \ldots, K$, and w_k satisfies $w_i \geq 0$ and $\sum_{k=1}^{K} w_k = 1$.

Let us consider the case of one Gaussian mixture component to represent normal activities.

$$\mathbb{E}_{\theta_f}[Pr(\mathbf{O}|f, \theta_f)] = \int_{\theta_{f\min}}^{\theta_{f\max}} \frac{Pr(\theta_f)}{\sqrt{(2\pi)^n |\theta_f \Sigma|}} e^{-\frac{(\mathbf{O}-\mu)^T \Sigma^{-1} (\mathbf{O}-\mu)}{2\theta_f}} \, d\theta_f \tag{11}$$

Assume $Pr(\theta_f)$ to be uniform distribution in $[\theta_{f\min}, \theta_{f\max}]$, where $\theta_{f\min} \geq 1$ and $\theta_{f\max} \to \infty$. The integral in Equation 11 will not give a closed form solution and needs to be evaluated numerically. We use MATLAB's *integral* function [13] that uses global adaptive quadrature method for computing an approximation of the integrand. Equation 11 can be extended to be used for a mixture of Gaussians as

$$\mathbb{E}_{\theta_f}[Pr(\mathbf{O}|f, \theta_f)] = \sum_{k=1}^{K} w_k \mathbb{E}_{\theta_{fk}}[Pr(\mathbf{O}|f, \theta_{fk})]$$

6. EXPERIMENTAL ANALYSIS

6.1 Datasets

We perform experiments on the following two activity recognition datasets.

- German Aerospace Center (DLR) [15]: This dataset is collected using XSens MTx sensor embedded with accelerometer, and gyroscope with sample frequency set to 100 Hz. The dataset contains samples from 19 people of both genders of different age groups. The data is recorded in indoor and outdoor environments under semi-natural conditions. The sensor is placed on the belt either on the right or the left side of the body or in the right pocket in different orientations. In total the dataset contains labelled data of over 4 hours and 30 minutes of the following 7 activities: Standing, Sitting, Lying, Walking (up/downstairs, horizontal), Running/Jogging, Jumping and Falling. One of the subjects did not perform fall activity; therefore, their data is omitted from the analysis.

- MobiFall (MF) [19]: This dataset is collected using a Samsung Galaxy S3 mobile device with inertial module integrated with 3D accelerometer and gyroscope and placed in a trouser pocket freely chosen by the subjects in random orientations. The mean sampling of 87Hz is reported for the accelerometer and 200Hz for the gyroscope. The dataset is collected from 11 subjects performing various normal and fall activities and 2 subjects only performing falls activity; therefore, they are removed from the analysis. Eight normal activities are recorded in this dataset: step-in car, step-out car, jogging, jumping, sitting, standing, stairs (up and down grouped together) and walking. Four different types of falls are recorded – forward lying, front knees lying, sideward lying and back sitting chair, that are joined together to make a separate class for falls.

For the MF dataset, the gyroscope sensor has a different sampling frequency than the accelerometer and their timestamps are also not synchronized; therefore, the gyroscope readings are interpolated to synchronize them with the accelerometer readings. Both the datasets have 3 readings each for the accelerometer (a) and gyroscope (ω) in the x, y and z directions. Sensor noise is removed by using a 1^{st} order Butterworth low-pass filter with a cutoff frequency of 20Hz. The dataset is segmented with 50% overlapping windows, where each window size is 1.28 seconds for DLR dataset and 3 seconds for MF dataset and 31 features are computed for each window as shown in Table 2. All the normal activities are joined together to represent a normal class. During the training phase only the data from normal activities are used and during the testing, data from both normal activities and falls are used. GMM can give underflow error due to large number of features; therefore, we employ Relief-F feature selection algorithm [20] and choose the first 15 top ranked features from the total list of features.

#features	**Type of feature**
20	Mean, maximum, minimum, standard deviation of $a_x, a_y, a_z, a_{norm}, \omega_{norm}$
2	Difference between the 75^{th} and the 25^{th} percentiles of a_{norm} and ω_{norm}
1	Normalized Signal Magnitude Area (SMA)
1	Normalized Average Power Spectral Density of a_{norm}
1	Spectral Entropy of a_{norm}
1	DC component after FFT of a_{norm}
1	Sum of the squared discrete FFT component magnitudes of a_{norm}
1	Normalized information entropy of the discrete FFT component magnitudes of a_{norm}
3	Correlation between each of the three acceleration readings a_x, a_y and a_z

Table 2: Number of computed features (for details see [9])

To estimate the performance of the proposed classifiers, we perform leave-one-subject-out cross validation [5], where normal activities from $(N-1)$ subjects are used to train the classifiers and the N^{th} subject's normal activities and fall events are used for testing. This process is repeated N times and the average performance metric is reported. This evaluation is person independent and demonstrates generalization capabilities as the test subject is not included in training the classifiers.

6.2 Results

During training, only one GMM is trained to model normal activities and its parameters are estimated by the Expectation Maximization (EM) algorithm. The number of mixtures in the GMM for modelling normal activities is set equal to the number of non-fall activities present in the data. The EM algorithm is initialized by K-means clustering and maximum number of iterations are set to 100. Diagonal covariance matrix is used and shared by all Gaussian components and a non-negative regularization number (= 0.0001) is added to the diagonal of covariance matrices to make them positive-definite. The values of utilities p and $q \in [0, 1]$ with a step size of 0.1. The pseudo counts for falls and non-fall activities per year is $\alpha = 2.6$, $\beta = 3.15569 \times 10^7$ [3]. Since DLR dataset is sampled at 1.28 seconds and MF dataset at 3 seconds; therefore, the value of α and β are scaled accordingly. A 2-fold internal cross validation is performed to find the optimal threshold from the training data (see Section 4.3). The values of $\theta_{f\min}$ and $\theta_{f\max}$ are set to be 1 and 100.

Figures 2a and 3a show the contour maps of the regret for using ML instead of EUT for the DLR and MF datasets when fall data is not present during training on different utilities p and q and averaged over actual number of activitie. We observe that the regret is positive in all the cases for both the datasets. However, we notice that in the empirical setting,the regret depends more on utility q and less dependent on utility p.

We observe that, for both the datasets, the empirical threshold is most of the time bigger than the ML threshold of 0.5. The reason for the large value of the empirical threshold is due to the maximization of the utility function (see Equation 8 and Section 4.3). In this step, higher utility is obtained for a given probability threshold, if more non-falls (re-scaled by β) are classified correctly in comparison to falls (re-scaled by α). In our setting $\beta \gg \alpha$; therefore, the probability threshold is chosen in the inner cross-validation step of mTh s.t. it is a large value that leverages more non-falls to be correctly identified at the cost of missing some falls – but their effect on the overall utility is minimal because they occur rarely in the test set. The value of the empirical probability threshold does not change much for different values of p and q (except for the boundary condition $q = 1$) because for each p and q the training set is the same; therefore, the model for falls and non-falls is the same. Hence, the pool of probability thresholds to look for to maximize the utility is the same. The magnitude of p and q is much smaller than β; therefore, most of the time the same probability threshold is chosen by the mTh algorithm for different values of p and q. Different values of the empirical threshold could be selected by mTh if the number of falls and non-falls are of the same order with similar utilities. In our problem setting, falls are rare, utilities are unequal and test data is severely skewed that leads to similar choice of the empirical thresholds. The pool of probability thresholds is limited by the number of training samples; therefore, the mTh algorithm can sometimes give sub-optimal choice of probability threshold leading to a negative threshold; however, on average the regret is positive. Figures 2b and 3b show the variation of the regret for the two datasets for all the values of utilities p and q. The dark circles ($\bullet$) show the average value of regret for a given utility p and q across all the subjects for each of the dataset, the lines protruding the dark circles are the standard deviation across the subjects and the blue mesh-grid is the zero regret for reference. The regret is higher at lower values of utilty q and so is their variation, which shows that the cost of false alarms is very important factor in designing a decision-theoretic fall detection system. At lower utility of false alarms, ML performs worse in comparison to EUT as shown in Figures 2c and 3c for the two datasets at utility $p = 0.5$.

These results suggest that if the model of falls and non-falls do not represent their respective true distributions, we can use empirical thresholding technique (mTh) to almost achieve the theoretical guarantee that EUT will always give better utility than ML. The results show that utility p does not matter so much, we don't need to precisely know the value of reporting a fall; however, utility q matters much more so knowing the utility of a false alarm is very important. As q gets smaller (false alarms have greater cost), using EUT will be more and more useful than ML.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a decision-theoretic framework for designing an automated fall detection system (*dtFall*) when the utilities of various actions are not known and fall data is not available. We presented a novel method to parameterize unseen falls that only uses the information from the model of training data available from the normal activities. We showed experimentally that using *dtFall* framework for fall detection, (a) knowing the difference in cost between a reported fall and false alarm is useful, (b) knowing this cost difference is more helpful as the cost of a false alarm gets bigger, and (c) knowing the difference in cost of between a reported and non-reported fall is not that useful. In future, we are interested to develop a mathematical model to estimate a cost model using the proposed decision-theoretic framework.

8. REFERENCES

[1] G. Acampora, D.J. Cook, P. Rashidi, and A.V. Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12):2470–2494, Dec 2013.

[2] Felicity R Allen, Eliathamby Ambikairajah, Nigel H Lovell, and Branko G Celler. An adapted gaussian mixture model approach to accelerometry-based movement classification using time-domain features. In *EMBS'06.*, pages 3600–3603. IEEE, 2006.

[3] CDC. Falls in nursing homes. http://www.cdc.gov/HomeandRecreationalSafety/Falls/nursing.html, 2014. Accessed on 5^{th} June 2015.

[4] S. Fida, U. Kalim, N. Latif, and S.A. Khayam. Decision theoretic approach to detect anomalies beyond enterprise boundaries. In *4th Symposium on Configuration Analytics and Automation (SAFECONFIG)*, pages 1–9, 2011.

[5] Zhenyu He and Lianwen Jin. Activity recognition from acceleration data based on discrete consine transform and svm. In *SMC*, pages 5041–5044. IEEE, 2009.

[6] Shuai Huang, Yujiu Yang, and Wenhuang Liu. An enhanced fall detection approach based on cost sensitivity analysis. In *Software and Network Engineering (SSNE), 2011 First ACIS International Symposium on*, pages 81–85. IEEE, 2011.

[7] Raul Igual, Carlos Medrano, and Inmaculada Plaza. Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine*, 12(1):1–24, 2013.

[8] Shehroz S. Khan, Michelle E. Karg, Jesse Hoey, and Dana Kulić. Towards the detection of unusual temporal events during activities using hmms. In *SAGAWARE - Proceedings of the ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 1075–1084. ACM, 2012.

[9] Shehroz S. Khan, Michelle E. Karg, Dana Kulić, and Jesse Hoey. X-Factor HMMs for detecting falls in the absence of fall-specific training data. In *Ambient Assisted Living and Daily Activities*, volume 8868 of *Lecture Notes in Computer Science*, pages 1–9. Springer International Publishing, 2014.

[10] Shehroz S. Khan and Michael G. Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29:345–374, 6 2014.

[11] Marcus A Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In *ICML-2003 workshop on learning from imbalanced data sets II*, volume 2, pages 2–1, 2003.

[12] Marcus A Maloof. *Machine learning and data mining for computer security.* Springer New York, 2006.

[13] MATLAB$^{\text{TM}}$. Numerical integration. http://www.mathworks.com/help/matlab/ref/integral.html. Accessed on 19^{th} January 2016.

[14] Oskar Morgenstern and John Von Neumann. *Theory of games and economic behavior*. Princeton University Press, 1953.

[15] Maria Josefa Vera Nadales. Recognition of human motion related activities from sensors. Master's thesis, University of Malaga and German Aerospace Cener, 2010.

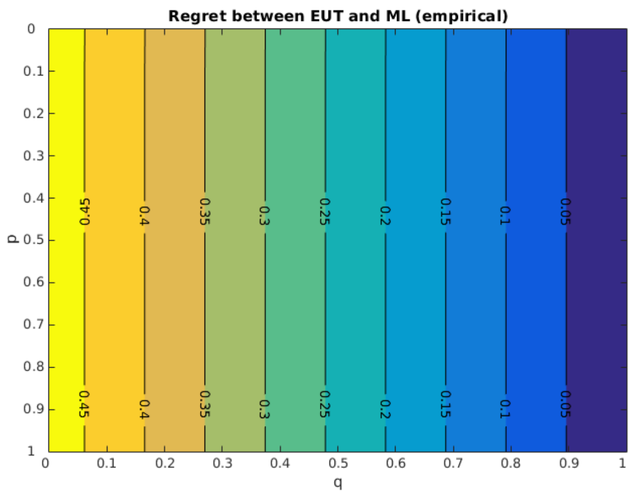
[16] Mrinal Nandi, Amiya Nayak, Bimal Roy, and Santanu Sarkar. Hypothesis testing and decision theoretic approach for fault detection in wireless sensor networks. *International Journal of Parallel, Emergent and Distributed Systems*, pages 1–24, 2014.

[17] J. A. Quinn, C. K.I. Williams, and N. McIntosh. Factorial switching linear dynamical systems applied to physiological condition monitoring. *IEEE Transactions on PAMI*, 31(9):1537–1551, 2009.

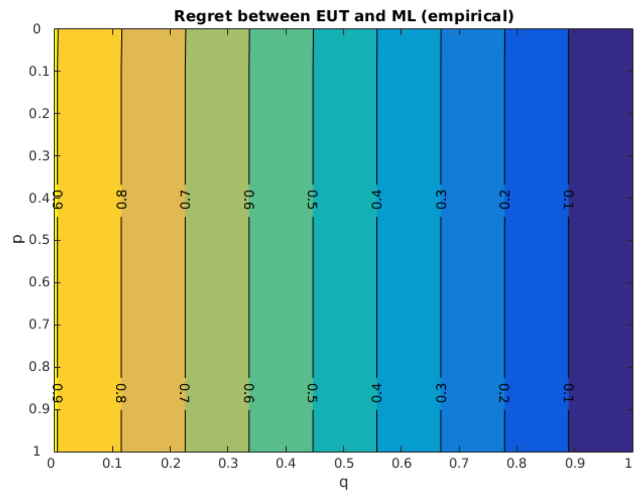
[18] Victor S Sheng and Charles X Ling. Thresholding for making classifiers cost-sensitive. In *Proceedings of the 21st national conference on Artificial intelligence-Volume 1*, pages 476–481. AAAI Press, 2006.

[19] G. Vavoulas, M. Pediaditis, E.G. Spanakis, and M. Tsiknakis. The mobifall dataset: An initial evaluation of fall detection algorithms using smartphones. In *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on*, pages 1–4, Nov 2013.

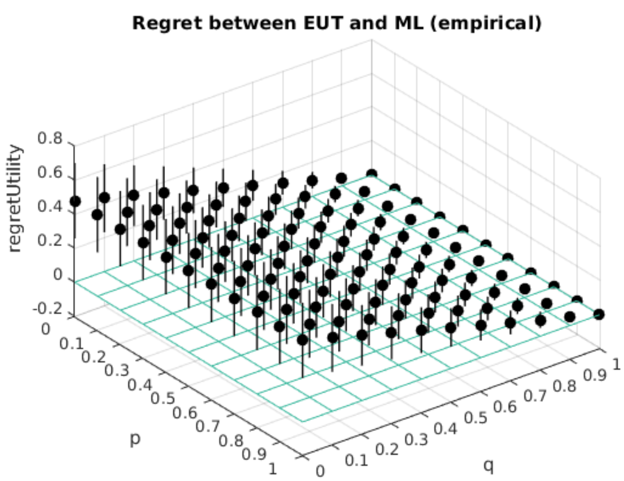
[20] Mi Zhang and Alexander A. Sawchuk. A feature selection-based framework for human activity recognition using wearable multimodal sensors. In *Proc. of the 6th International Conference on Body Area Networks*, pages 92–98, Belgium, 2011.



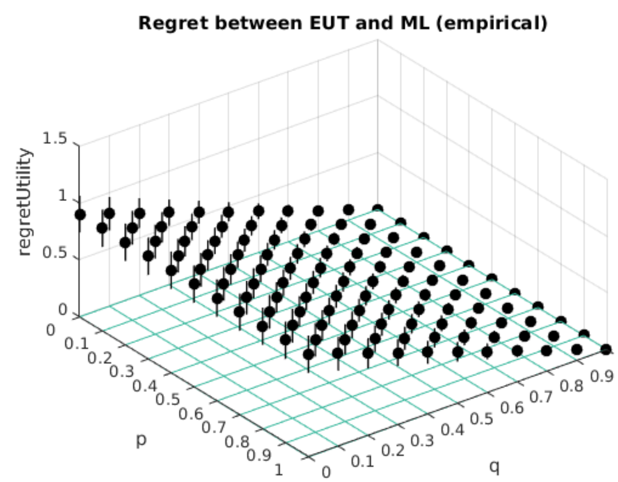
(a) Heatmap of Regret



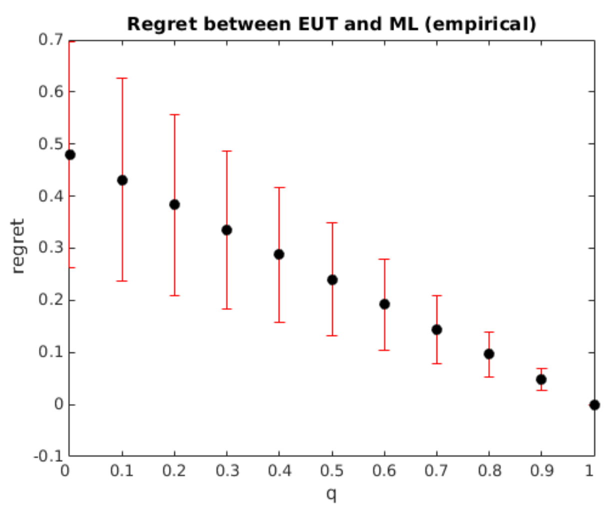
(a) Heatmap of Regret



(b) Regret with Standard Deviation

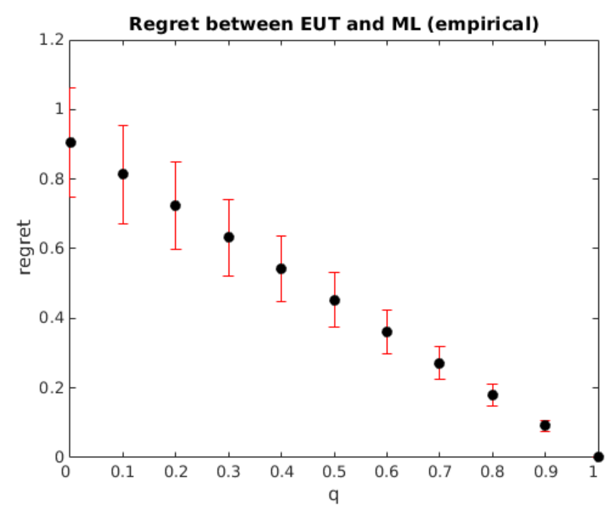


(b) Regret with Standard Deviation



(c) Variation of Regret w.r.t. q (for $p = 0.4$)

Figure 2: Comparison of Regret between EUT and ML for DLR dataset



(c) Variation of Regret w.r.t. q (for $p = 0.4$)

Figure 3: Comparison of Regret between EUT and ML for MF dataset