Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 37 – 44

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

"Fortune at the bottom of the Classifier Pyramid": A Novel approach to Human Activity Recognition

Vijay Borges[a,*], Wilson Jeberson[a]

[a]*Sam Higginbottom Institute of Agriculture, Technology & Sciences (SHIATS) - Deemed University#, Allahabad, U.P., India 211 007*
#(Accredited by NAAC with 'A' Grade & Awarded 'A' Grade by MHRD, Govt. of India)

Abstract

"Fortune at the Bottom of the Pyramid", penned by the noted economist Prof. C. K. Prahalad[1], talked about the wealth creating potentials of entities at the bottom of the economic hierarchy. Taking the same cue into classifier research in Machine Learning, Naive Bayes classifier is often the *beaten boy* among its contemporaries. In this work, Naive Bayes (NB), is revisited by looking at heuristic ways of improving any skewed data bias, systematic and weighted magnitude errors; to address the problem of human activity recognition and propose the *Improved Naive Bayesian Algorithm*. The novelty of this work is in adjusting the human activity recognition problem as a special case of text classification.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Smart Homes, Sensor Networks, Middleware, Ubiquitous Computing, Machine Learning, Naive Bayes (NB), Activity Recognition, Activity of Daily Living, Text Classification

1. Introduction

Ubiquitous Computing and the established area of Human-Computer Interaction (HCI), are an active cross-disciplinary research area, touching nascent research areas like mobile computing, ambient assistive living, context-aware computing, security and surveillance, intelligent spaces, smart homes, smart hospitals, smart infrastructure, smart educational institutes, etc.[1,2,3,4]. This HCI domain goes beyond the traditional computer-computer interaction onto an integrated human-computer co-existence. Due to the complex human nature, the activities of humans are difficult to recognize; this necessitates that the ubiquitous environments have sufficient reasoning ability for successful human Activity Recognition (AR). AR is a task that does *an automatic recognition of activities* and is a key research area under HCI. Over the past few decades tremendous improvement in sensor technology both in fabrication, underlying wired/wireless communicating ability, processing ability and in economic terms has been instrumental in pushing it to make spaces of its deployment truly ubiquitous (also termed as smart environments). *Smartness*, comes from using the low-level data gathered from the sensors deployed in these environments, processing the same and infering/reasoning

[1] Prof. C. K. Prahalad was Professor of Corporate Strategy at the University of Michigan.
* Corresponding Author. Tel.: +91-9923799058
E-mail address: vijayborges@gmail.com (Vijay Borges).

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.01.047

from the data. This work focuses only on *Activity Recognition System* (ARS), that would use the underlying sensor technology (presentated through a sensor network) to infer/reason the activities of living entities (mostly humans) in an intergrated manner. The goal of this system would be to proactively assist inhabitants with their daily living tasks in these smart environments. ARS (also called *smart environment middleware*[5]), has many applications right from, security surveillance for mitigating threat of risk to life/property, to, activity recognition in smart homes, assistive living for the elderly, nursery care for children, smart hospitals, smart educational institutes etc.[3,2,4].

Naive Bayes classifier has the distinction of being simple to understand, implement and gives fairly good results[6], when applied to the task of activity recognition. But, Naive Bayes classifier is often called as "the punching bag of classifiers"[7] and lowly rated among its contemporary classifiers[8,9]. The novelty of this work is that we are re-visiting the Naive Bayes Classifier and find the reasons for its poor performance[10], and see if some simple heuristics suggested in Section 4, can be used to improve its performance when applied to the task of AR in ARS.

In the paper[6], we presented a novel model, *Activity Recognition using Text Categorization Paradigm (AR-TCP)*, that would be the fundamental building block of the ARS and assist it with efficient reasoning mechanisms. In this work we plan to improve the AR-TCP model by building a better reasoner to improve its performance using suggestions from Section 4. The flow of this paper is built by initially discussing activity recognition (AR) in a sensor based environment and touching on the AR-TCP model. The research contribution is wrapped by discussing the extension to AR-TPC model using the proposed *"Improved Naive Bayesian"* algorithm and highlighting its advantages.

2. Activity Recognition (AR) in the Activity Recognition System (ARS)

Mark Weiser in his seminal paper[11] defined a vision called *"Ubiquitous Computing"*, where environments (enclosed spaces like a room or corridor) saturated with devices (sensors), having computing and communicating capabilities, would gather and process information from many sources to both control physical processes and interact with inhabitants of the environments.

In a ubiquitous/pervasive environment, human Activity Recognition (AR), is of paramount importance to proactively assist users with their daily tasks. In the healthcare scenario AR may be critically required for those humans who may be elderly, young children or persons with special needs. In India, as per the report by the Ministry of Statistics and Planning, Government of India[12], which gives the growth rate of elderly population (above the age group of 60 years) over the decades in certain states of India; states that its steadily growing from *5.6%* in 1961 to *7.1%* in 2001 and is being projected to rise to *12.4%* by 2026. Giving India's changing demographics, increased nuclear families, increased medical and caregiver costs; makes elderly care a matter of grave concern. The elderly population is also highly susceptible to many health concerns like reduction in their cognitive abilities, due to the effects of Alzheimer's, Dementia syndrome etc. and other ailments. It is in these concerning scenarios that the ARS can be an effective solution in providing the necessary aid to the population effected by these medical ailments and be of immense help to their caregivers and families.

Activities of daily living may be performed by inhabitants at varying speeds, in different order, overlapped, interleaved; all these must be captured by the sensors and sent correctly to the ARS. The ARS would be required to automatically detect and infer activities based on the captured data. This inference could be helpful to the caregivers of the inhabitants; be it their family or paramedics. *Activity Recognition using Text Categorization Paradigm (AR-TCP)*[6], is the fundamental building block of the ARS and assists it with efficient inference/reasoning mechanisms. The block diagram of the ARS with the AR-TCP model is shown in Fig. 1. A few abstract steps are given to build clarity in the AR-TCP model:

Step 1. Input to the AR-TCP model comes from the sensor data via interactions of user in a smart environment.

Step 2. Every sensor's activation is due to the user's direct/in-direct interaction with the smart environment (ARS), translating into a *sensor event*. A set of these *sensor events* corresponds to known classes of human *Activity* or *Activity of Daily Living (ADL)*. It is these sets of *sensor events* that a *human observer*, would annotate with proper labels, tagging each of them with an appropriate *Activity Label*, corresponding to actual activity done by the user. These sets of *activity labels*, would then be used in the training phase of the AR-TCP.

Step 3. In the training phase, the extracted features and the annotated activity labels are given as an input to the AR-TCP to build a training model.

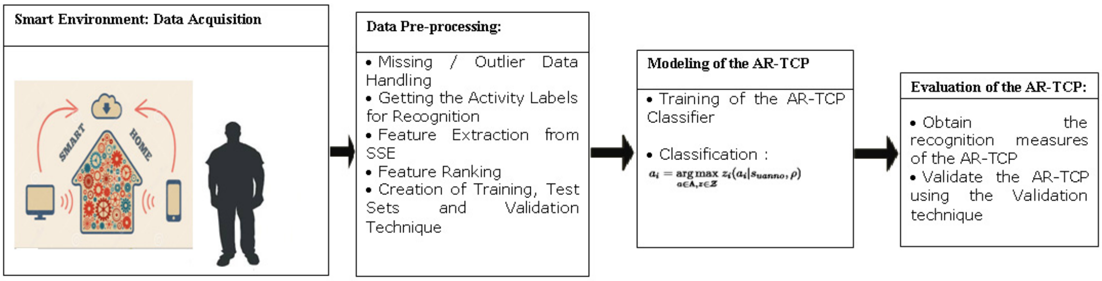

Fig. 1. Activity Recognition using the AR-TCP Model in a Smart Environment (ARS)

Step 4. In the classification phase, for every unknown activity of the user, the features from the sensor data are extracted and along with the training model built previously are given as input to the classifier of the AR-TCP. The AR-TCP classifier would calculate a score to classify the unknown activity.

3. Theoretical extensions to Naive Bayes (NB) in Activity Recognition using Text Categorization Paradigm (AR-TCP)

Our previous paper[6], builds the mathematical foundation of using the text categorization paradigm (TCP) for activity recognition in smart environments. Here we extend it by incorporating heuristics discussed in Section 4.

In[6], it was mathematically theorized that there exists a direct resemblance of the *category* $\mathbb{C}$, of the Text Categorization (a.k.a. Text Classification), to the *activity label* $\mathbb{A}$, of the AR/ADL. The features of the text document, *item* or *word*, resembled the features from the smart environment like, *sensor event*. In TCP, documents are represented by features like *word*. Similarly a collection of *sensor events* got from the sensors in the smart environment; properly segmented, correspond to some user activity and are termed as *"segments of sensor event collections (SSE)"*. In TCP, documents are grouped together under a category label based on the *words* found in the document using some measures or some text classifier model. Analogously, for activity recognition in the smart environment a *activity label* may be assigned to the collection of *sensor events* based on the patterns found in it by applying appropriate measures or models as defined in the AR-TCP.

Let $\mathbb{S} = \{s_1, s_2, ..., s_N\}$, be a set of *"segments of sensor event collections"* ***(SSE)***, and $\mathbb{A} = \{a_1, a_2, ..., a_M\}$, be the set of pre-defined *"activity labels"* ***(AL)***. Each s_i, be a collection of l features of the sensor event vector $\underline{e}_r$, where $r \in 1, 2, ..., l$; to form the feature vector $\underline{E} = [\underline{e}_1, \underline{e}_2, ..., \underline{e}_l]^{\mathbb{T}}$, where $\mathbb{T}$ denotes matrix transpose. AR-TCP model assigns a boolean value to each pair, $< s_i, a_j > \in \mathbb{S} \times \mathbb{A}$. Each pair $< s_i, a_j >$, is assigned a value from the set {**T,F**} using a function $\Omega : \mathbb{S} \times \mathbb{A} \Rightarrow \{T, F\}$ (called labeling).

$$\Omega : \mathbb{S} \times \mathbb{A} \begin{cases} \mathbf{T} & : s_i \in a_j & \text{(1a)} \\ \mathbf{F} & : s_i \notin a_j & \text{(1b)} \end{cases}$$

The function Ω, is the ideal AR-TCP classifier, but we would be more interested in finding the approximate AR-TCP classifier, $\hat{\Omega}$, that would closely match the accuracy of the ideal AR-TCP classifier, Ω. The approximate AR-TCP classifier, $\hat{\Omega}$, would follow the *single-label categorization*. According to[6], the AR-TCP model would be built as follows:

Step 1. A *human observer* would annotate the SSE's got from the user interactions with the smart environment to come up with, $\overline{\mathbb{S}} \subset \mathbb{S}$, having correct *activity label*, a_j or $\overline{a_j}$, for a considered activity from, $\mathbb{A} = \{a_1, a_2, ..., a_M\}$

Step 2. The AR-TCP classifier $\hat{\Omega}$, is built for the activity label a_j, by observing the characteristics (features) of the given annotated SSE, $\overline{\mathbb{S}}$. This step is called as training. Thus an annotated SSE, s_i is a *positive example* of a_j if, $\hat{\Omega}(s_i, a_j) = \mathbf{T}$; else annotated SSE s_i is a *negative example* of a_j if $\hat{\Omega}(s_i, a_j) = \mathbf{F}$. The training phase can be adapted as follows:

i. The AR-TCP classifier $\hat{\Omega}$, is defined with a parameter ρ, depending on the basic machine learning technique adapted. Let the AR-TCP classifier $\hat{\Omega}$, be probabilistic with parameter, $\rho = (\overline{\mathbb{S}}, \mathbb{A})$. The classifier

$\hat{\Omega}$, builds the necessary probabilistic likelihood values using equation 3. Alternatively, the AR-TCP classifier can a non-parametric classifiers such as kNN (k Nearest Neighbours), which requires no training. kNN does classification by matching the activity label of the test SSE to the activity labels of it k-nearest neighbors.

Step 3. On presenting a new un-annotated SSE's s_{uanno}; where $s_{uanno} \in \mathbb{S}$, and, $s_{uanno} \notin \overline{\mathbb{S}}$; to the classifier $\hat{\Omega}$, it would derive characteristics from s_{uanno}, so as to be classified as a_j. This step is called as the testing phase and is done as follows:

i. The trained AR-TCP classifier $\hat{\Omega}$, with parameter $\rho = (\overline{\mathbb{S}}, \mathbb{A})$, when presented with the un-annotated SSE, $s_{uanno} \in \mathbb{S}$, comes with a corresponding score, $\mathcal{Z} = \{z_1, z_2, ..., z_{|\mathbb{A}|}\}$, and

$$z_i(a_i|s_{uanno}, \rho) = Inf(s_{uanno}, \rho);\ \text{where } a_i \in \mathbb{A} \tag{2}$$

where, *Inf(.)*, being the *inference method* with parameter ρ . The inference method used is based on scores derived from probabilistic likelihood values due to the Bayesian approach (refer equation 3).

$$\begin{aligned} log\,(P(a_i|s_{uanno})) &= log(P(a_i)) + \sum_{u=1}^{|r|} w_u log\left(\frac{N_{ui} + \alpha_i}{N_i + \alpha}\right) - log\,(P(s_{uanno})) \\ &= \zeta_i + \sum_{u=1}^{|r|} w_u \chi_{ui} - log\,(P(s_{uanno})) \end{aligned} \tag{3}$$

where, $P(a_i)$ is prior probability of the activity label $a_i \in \mathbb{A}$; w_u is the *term-weight* vector of s_{uanno}; and the ratio $\left(\frac{N_{ui}+\alpha_i}{N_i+\alpha}\right)$ is got from the training set $\overline{\mathbb{S}}$. N_{ui}, is the number of times the *sensor event* $\underline{e}_u$, occurs in the *activity label* a_i; and N_i, is the total number of times the *sensor events* with the *activity label* a_i (i.e. $N_i = |s_i|$), occurs in the training set. $\alpha = \sum_{i=0}^{|\overline{\mathbb{S}}|} \alpha_i$, are called as the *Laplacian Smoothing* constants[13], and are introduced to prevent zero probabilities for infrequent *sensor event* $\underline{e}_r$, from occurring during evidence calculation and may be set heuristically or empirically as suggested by[14]. In general literature, *Laplacian Smoothing* constant are set to 1. We found that for Activity Recognition (AR) problem, setting the *Laplacian Smoothing* constant by default to 1 reduces the accuracy of the classifier. In Section 4, we have discussed the appropriate value to be used when applied to the Activity Recognition domain.

ii. To decide which activity label $a_i \in \mathbb{A}$, to be assigned to the SSE, s_{uanno}; is done by selecting activity label corresponding to the maximum score via:

$$a_i = \underset{a \in \mathbb{A}, z \in \mathcal{Z}}{\arg\max}\ z_i(a_i|s_{uanno}, \rho) \tag{4}$$

Step 4. The classifier $\hat{\Omega}$, would then be compared with the utopian classifier Ω, to determine the accuracy of $\hat{\Omega}$ using selected measures of classifier effectiveness and accuracy[6].

The AR-TCP classifier $\hat{\Omega}$ specified in[6], was based on the probabilistic *Naive Bayesian* (NB) classifier. According to the work by[10] and[9], the poor performance of the NB is due to choices it makes in selecting ζ_i and χ_{ui} (refer equation 3), called as *systematic problems*. One *systematic problem* is when one *activity label* has more training examples in $\overline{\mathbb{S}}$, than the others. This would cause a shrinkage effect on the weights for the *activity label* with lesser training examples. The other *systematic problem* with NB is that of its strong independent assumption; even though there may be certain inherent dependencies.

The suggestions offered by[10 9] to compensate for the *systemic problems* inherent in NB are the following:

i. To offset the effect of unbalanced *activity labels* in the training set examples, is to have a "complement activity label".
ii. By normalizing the classification weights, *activity labels* that are dependent are prevented from dominating.

iii. Have certain *smoothing parameter* α, estimations for the occurrence of the *sensor event* in the SSE's got from the user interactions with the smart environment.

4. Incorporating the Changes in Naive Bayesian Classifier of AR-TCP

For supervised classification, the number of *activity labels* $|\mathbb{A}|$, and the training annotated SSE $\overline{\mathbb{S}}$, are given. The calculations of χ_{ui} and ζ_i, causes the NB some of its many woes clubbed together as *systemic problems* (refer Section 3). We next discuss heuristics to mitigate these problems.

4.1. Skewed Data in the Training Set

Skewed data occurs when there are more training sets $|s_i|$, belonging to an activity label a_i, than the others, i.e. $|s_i| >> |\overline{\mathbb{S}} - s_i|$. This skewed data causes the NB to be inadvertently biased towards a_i. To compensate for this systemic problem, [8], suggests to formulate a *"complimentary estimate"*, $\chi_{u\bar{i}} = log\left(\frac{N_{u\bar{i}}+\alpha_{\bar{i}}}{N_{\bar{i}}+\alpha}\right)$. Substituting $\chi_{u\bar{i}}$ in equation 3, yields:

$$log\left(P(a_i|s_{uanno})\right) = \zeta_i + \sum_{u=1}^{|r|} w_u \left(\chi_{ui} - \chi_{u\bar{i}}\right) - log\left(P(s_{uanno})\right) \tag{5}$$

Note that the *"complimentary estimate"* $\chi_{u\bar{i}}$, is subtracted in equation 5, so as to assign to the activity label a_i all those sensor events that indistinctly match the *"complimentary estimate"*.

4.2. Compensating for the errors in the Weight Magnitude

The independence assumption of the NB is the largest contributor to its miseries of being termed "Naive", and being dumped to the *"bottom of the pyramid"* of the *"classifier economic hierarchy"*. Having skewed training data, would cause NB to be biased towards the activity label a_i, having a larger magnitude in the vector $\underline{\chi}_i$. This is compensated by normalizing the magnitude of the vector $\underline{\chi}_i$, where $i \in \{1, 2, ..., |\mathbb{A}|\}$, as:

$$\chi_{ui} = \frac{log\left(\frac{N_{ui}+\alpha_i}{N_i+\alpha}\right)}{\sum_{v=1}^{|r|} |log\left(\frac{N_{vi}+\alpha_i}{N_i+\alpha}\right)|} \tag{6}$$

4.3. Transformation Functions and Laplacian Smoothing

We have paid attention to the negative effects of having a skewed training data, towards the biased classification done by the NB. There is another subtle problem to be tackled, that of having a long *activity* with a huge amount of *sensor events* from dominating the parameter estimates of the classifier and causing the classifier from wrong reporting. In the text categorization paradigm this effect has been much studied under *"transformation based on length"* and *"transformation based on term frequency"*[7], [15], [16].

In AR, when there is a large SSE, $s_i \in \mathbb{S}$, compared to the other SSE's in the training data and $|s_i| >> |s_j|$; it would try and dominate the parameter estimates of the AR-TCP classifier $\hat{\Omega}$. This would cause $\hat{\Omega}$, to give wrong classification labels. One solution to this problem is to do a length normalization of N_{ui} as:

$$N'_{ui} = \frac{N_{ui}}{\sqrt{\sum_{u=1}^{|r|} (N_{ui})^2}} \tag{7}$$

where r, is the index of sensor event vector $\underline{e}_r$, corresponding to the *activity label* a_i

Similarly, there is a heuristic transform in text categorization paradigm called the "inverse document frequecy": which discounts terms by their document frequency [15, 16]. This same technique can be applied to AR, to discount the sensor event $\underline{e}_r$, by its *activity label* a_i frequency; which translates to:

$$N'_{ui} = N_{ui}\, log\left(\frac{\sum_{k=1}^{|\bar{\mathbb{S}}|} 1}{\sum_{k=1}^{|\bar{\mathbb{S}}|} \varepsilon_{uk}}\right) \tag{8}$$

where ε_{uk} is 1, if the sensor event e_u, occurs for the *activity label* a_k, 0 otherwise. Using this transformation rare sensor events $\underline{e}_u$ are given higher preference than sensor events that occur more commonly.

A simple transformation is suggested by [17, 18], called "term burstiness": which is the rare occurrence of a term in a single document, but its occurrence increases substantially in a large collection of documents. Applied to the domain of AR, wherein a rare sensor event $\underline{e}_r$, in an SSE s_i; may have to be adequately compensated for in a larger collection of that same SSE s_i in the training data. This can be handled by a simple power log transform:

$$N'_{ui} = log\,(N_{ui} + 1) \tag{9}$$

Plug these new value from equations 7, 8, 9 and update equation 6.

Laplacian Smoothing α_{ui}, was discussed very briefly in section 3; here it is elaborated a little: since the choice of α_i, has a bearing on the results of the AR-TCP classifier, especially when using the probabilistic NB approach. In general literature [13], the default value is suggested to be set to $\alpha_i = 1$, so as to avoid the zero probabilities being introduced as a result causing the classifier learning being nullified. In [14], the authors argue that setting α_i to its default value is not the best smoothing method. They advocate its values to be set as $\alpha_i \in (0, 1)$, which is based on the concept of *uniform distribution*. In our work we have set α_i, as:

$$\alpha_i = \frac{1}{|s_i|} \tag{10}$$

We have found that setting α_i as in equation 10, improves the AR-TCP classifier performance.

5. Experimental Results

For implementation of the AR-TCP classifier as per the suggestions of the algorithm 1, and testing its recognition capabilities a publicly available *Cairo* dataset from the WSU's *CASAS* [19], testbed was used. This dataset has two residents and a pet, it has 27 motion sensors (other sensors are ignored), 600 activities that were recorded for a period of 3 months. It has the following 10 self-explaining macro-activity labels $\mathbb{A}$, that were used for annotating the datasets by the *human observer*; *Breakfast, Lunch, Laundry, Dinner, Leave_home (L_Home), Taking_medicine (R_Med), Go_For_Work(C_work), Night_wandering (N_Wand), Bed, Bed_to_toilet (B_Toilet)*. The dataset *Cairo*, was selected to test the ability of the AR-TCP classifier to handle activities that are overlapping, sequential, having multiple residents, etc.; the sensors that were placed in the *CASAS* environment were non-intrusive and non-body worn. These and more such similar demands are placed on the ARS which are discussed in details in [6], and justification for the same is provided.

The experimental setup for the AR-TCP classifier using the *Improved Naive Bayes*, was done as follows:

- The *Cairo* dataset had altogether 600 annotated *macro-activities*. The breakup of the number of individual *macro-activities* were as follows; *B_Toilet=30, Breakfast=47, C_Work=40, Dinner=42, Laundry=10, L_Home=70, Lunch=37, N_Wand=67, R_Med=45, Bed=212*.
- The *Cairo* dataset was divided into two parts called *Training Set* and the other *Testing Set*.
- Since the AR-TCP classifier's recognition capabilities had to be checked, the *k-fold Cross Validation* technique was decided to be adopted.

Algorithm 1 : Improved Naive Bayesian (INB)

Input: The annotated training segments of sensor events (SSE) $\overline{\mathbb{S}} = \{s_1, s_2, ..., s_N\}$, where $s_i \in \mathbb{R}^N$ and $s_i = \{\underline{e}_1, \underline{e}_2, ..., \underline{e}_r\}$.
N_{ui}: is the count of the sensor event $\underline{e}_j$ occurring in s_i.
Let $\mathbb{A} = \{a_1, a_2, ..., a_M\}$, be the activity labels assigned to $\mathbb{S}$.
Let $s_{uanno} \in \mathbb{S}$ and $s_{uanno} \notin \overline{\mathbb{S}}$ and $s_{uanno} = \{\underline{z}_1, \underline{z}_2, ..., \underline{z}_N\}$ where z_k : be the sensor event for the SSE in s_{uanno}.
Output: Put Activity Label a_i for s_{uanno}.
BEGIN:

$N_{ui} = log\,(N_{ui} + 1)$ ▷ Burstiness Transform

$N_{ui} = N_{ui}\; log \left(\frac{\sum_{k=1}^{|\overline{\mathbb{S}}|} 1}{\sum_{k=1}^{|\overline{\mathbb{S}}|} \varepsilon_{uk}} \right)$ ▷ Inverse SSE frequency

$N_{ui} = \frac{N_{ui}}{\sqrt{\sum_{u=1}^{|r|} (N_{ui})^2}}$ ▷ SSE Length Normalization

$\chi_{ui} = \frac{log\left(\frac{N_{ui}+\alpha_i}{N_i+\alpha}\right)}{\sum_{v=1}^{|r|} \left|log\left(\frac{N_{vi}+\alpha_i}{N_i+\alpha}\right)\right|}$ ▷ Weight Normalization

$a_i = \arg\max_{a \in \mathbb{A}, z \in \mathcal{Z}} (z_i\, \chi_{ui})$ ▷ Get activity label for test sample

$a_i \;\rightarrow\; s_{uanno}$ ▷ Set activity label to the test sample

END

- For every validation round of the classifier a different subset was chosen as the *training set*, using the *cross-validation* technique.
- Measures to check the accuracy of the AR-TCP classifier were used.

6. Discussions

We used k-fold cross validation for the AR-TCP classifier with, $k = 7$. The entire dataset was proportionally divided into training and testing set; so that every sample of the entire dataset got a fair chance to be part of the testing set; as required by the cross validation procedure. Table 1, shows the accuracy of the AR-TCP classifier using the simple Naive Bayesian (NB) approach versus its implementation using the Improved Naive Bayesian (INB) algorithm 1.

Table 1. Accuracy of the AR-TCP classifier using the Naive Bayesian and the Improved Naive Bayesian approaches

Validation #	Acc. of NB (%)	Acc. of INB (%)
1	83.95	91.55
2	70.42	70.93
3	64.91	70.58
4	69.23	75.03
5	58.96	74.88
6	61.91	76.20
7	72.73	85.08
Overall Accuracy (%)	**68.44**	**77.75**

The table shows that the difference in AR-TCP Overall Accuracy using the INB over the classical NB approach is 9.31%; which is a significant improvement of 13.60%. There is a marked improved in Activity Detection using the heuristical suggestions incorporated in INB over the classical NB. The NB is a classifier that is simple to implement, understand and has low time complexity compared to its other contemporary classifiers. The NB classifier despite its independence assumption, surprisingly gives fairly decent results in the AR domain. The simple heuristic suggestions incorporated in INB keeps all the advantages of the NB classifier intact, yet gives a marked improvement in activity recognition using the AR-TCP model.

7. Conclusion

It was shown in this effort that the AR-TCP model is easy to use and most of the classifiers from the text classification domain can be applied using it. The proposed algorithm *Improved Naive Bayesian*, comes with a marked improvement over the classical Naive Bayesian and was easily integrated into the AR-TCP. This improvement affirms that the AR-TCP model, *" ... can help the research community by being a bridge between the best practices from the established text categorization domain and use the same in the comparatively more nascent activity recognition domain"*[6] and indeed there is *"Fortune at the bottom of the Pyramid"*[20].

Acknowledgement

The authors would like to thank God for all blessings. Thanks to Computer Society of India for their research grant vide grant number 1-14/2013-09 dated 21/03/2013. Thanks also to Prof. S. K. Pal, CSCR, ISI, Kolkata - India for his support, Prof. C. A. Murthy, MIU, ISI Kolkata for necessary inputs and post-lecture discussions, and to Prof. A. Srivastava, SHIATS- DU for her guidance. Thanks to Mr. Suman Kundu for many fruitful discussions. Big thanks to our families for their co-operation and understanding. Gratitude to the Christian Brothers Community at SMO, Dum-Dum, Kolkata for their hospitality.

References

1. V. Borges, W. Jeberson, Survey of Context Information Fusion for Sensor Networks Based Ubiquitous Systems, Computer Science and Information Technology 2 (3) (2014) 165–178. `doi:10.13189/csit.2014.020306`.
2. MIT, The Futuristic House.
URL `http://architecture.mit.edu/house_n/`
3. Gator, Gator Home.
URL `http://www.icta.ufl.edu/gatortech/`
4. V. Borges, R. Sawant, A. Zarapkar, S. Azgaonkar, Wireless automated monitoring system for an educational institute using Learning Management System (MOODLE) (2011). `doi:10.1109/SoCPaR.2011.6089112`.
5. M.-M. Wang, J.-N. Cao, J. Li, S. K. Dasi, Middleware for wireless sensor networks: A survey, Journal of computer science and technology 23 (3) (2008) 305–326.
6. V. Borges, W. Jeberson, Modeling Activity Recognition Using Text Categorization Paradigm, Elsevier Expert System with Applications (under consideration).
7. D. D. Lewis, Naive (Bayes) at forty: The independence assumption in information retrieval, in: Machine learning: ECML-98, Springer, 1998, pp. 4–15.
8. T. Zhang, F. J. Oles, Text categorization based on regularized linear classification methods, Information retrieval 4 (1) (2001) 5–31.
9. T. Joachims, Text categorization with support vector machines: Learning with many relevant features, Springer, 1998.
10. J. D. Rennie, L. Shih, J. Teevan, D. R. Karger, Others, Tackling the poor assumptions of naive bayes text classifiers, in: ICML, Vol. 3, Washington DC), 2003, pp. 616–623.
11. M. Weiser, J. S. Brown, The coming age of calm technology, in: Beyond calculation, Springer, 1997, pp. 75–85.
12. GOI, Situation Analysis of the Elderly in India.
URL `http://mospi.nic.in/mospi_new/upload/elderly_in_india.pdf`
13. K. Nigam, A. K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine learning 39 (2-3) (2000) 103–134.
14. D. Vilar, H. Ney, A. Juan, E. Vidal, Others, Effect of Feature Smoothing Methods in Text Classification Tasks., in: PRIS, Citeseer, 2004, pp. 108–117.
15. K. S. Jones, A statistical interpretation of term specificity and its application in retrieval, Journal of documentation 28 (1) (1972) 11–21.
16. A. Singhal, G. Salton, M. Mitra, C. Buckley, Document length normalization, Information Processing & Management 32 (5) (1996) 619–633.
17. S. M. Katz, Distribution of content words and phrases in text and language modelling, Natural Language Engineering 2 (1) (1996) 15–59.
18. K. W. Church, W. A. Gale, Poisson mixtures., Natural Language Engineering 1 (2) (1995) 163–190.
19. WSU, CASAS Project.
URL `http://ailab.wsu.edu/casas/`
20. C. K. Prahalad, The Fortune at the Bottom of the Pyramid, Pearson Education India, 2006.