

Human Activity Recognition in Smart Home Using Prior Knowledge Based KNN-WSVM Model

M'hamed B. Abidine^(✉) and Belkacem Fergani

Laboratoire d'Ingénierie des Systèmes Intelligents et Communicants
(LISIC Laboratory), Faculty of Electronics and Computer Sciences,
University of Science and Technology Houari Boumediene (USTHB),
32, El Alia, Bab Ezzouar 16111, Algiers, Algeria
abidineb@hotmail.com

Abstract. The ability to recognize human activities from sensed information become more attractive to computer science researchers due to a demand on a high quality and low cost of health care services at anytime and anywhere. In this work, we proposed a new hybrid classification model to perform automatic recognition of activities in a smart home using KNN-WSVM combined the K-Nearest Neighbors (KNN) with Weighted Support Vector Machines (WSVM) learning algorithm, allowing to better discrimination between the classes of activity. We also added the temporal features (TF) in the used method KNN-WSVM. Experiments show our proposed approach outperforms the KNN, WSVM used alone in terms of recognition performance, highlighting the advantages of this method.

Keywords: Smart homes · Activity recognition · KNN
Support vector machines · Weighted SVM

1 Introduction

The growing population of elders in our society calls for a new approach in care giving [1] to ensure the comfort of old people and because the healthcare infrastructures are unlikely to accommodate the drastic growth of elderly population. Smart systems are equipped with sensor networks able to automatically recognize activities about the occupants and assist humans. They must be able to recognize the ongoing activities of the users in order to suggest or take actions in an intelligent manner. Activity recognition can be used to automatically monitor the activities of daily living (ADLs) of old people such as cooking, brushing, dressing, cleaning, bathing and so on substantially increased as well.

Feature extraction and classification are two key steps for activity recognition in a smart home environment [2]. In this paper, we proposed a new hybrid classification method KNN-WSVM using the *K-Nearest Neighbors (KNN)* [3] and *Weighted Support Vector Machines (WSVM)* [4]. The last is used to deal the class imbalance in the training data due to the fact that people do not spend the same amount of time on the

different activities. We also integrated the prior knowledge [5] in this method to improving the classification performances of KNN-WSVM. We introduced a new feature: the temporal feature (TF) of activity when the activity is performed. Then we added them to the existing features obtained by the collected wireless sensors data. The ‘Prepare breakfast’ and ‘Prepare dinner’ activities share the same model as they involve the same set of object interactions. These two activities are distinguished by time of taking place, i.e. ‘Prepare breakfast’ takes place in the morning hours and “Prepare dinner” takes place in the afternoon or evening hours of the day.

Our paper addresses these issues and contributes on the following topics. Firstly, we have presented the related works of human activity classification methods. Then, we explained the proposed activity recognition method using the feature insertion using the temporal attribute to recognize activities of daily living from binary sensor data. The next section presents the experimental setup and discusses the results acquired throughout a series of benchmark dataset [6, 7] constituted of highly imbalanced datasets under different metrics. Finally, conclusions and future work are drawn in the last section.

2 Related Works

The machine learning of the observed sensor patterns is usually done in a supervised manner and requires a large annotated data sets recorded in different settings [6, 7]. For this purpose, annotation of data for classification task can be performed in many different ways, e.g. use of cameras [8], self-reporting approaches [7] and monitoring the diary activity [6]. Several classification algorithms have been employed for ADL recognition tasks [6, 8, 9], e.g. Hidden Markov model (HMM) [6], Conditional Random Fields (CRF) [6], Linear Discriminant Analysis (LDA) [9], Bayes approach [8], Support Vector Machine (SVM) [9] and its Soft-margin multiclass SVM extension [10]. In [11], we developed a new classification method named PCA-LDA-WSVM based on a combination of Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Weighted Support Vector Machines (WSVM). We demonstrated the ability of this method to achieve good improvement over the standard used methods as HMM and CRF. This method resolved two problems result in a degradation of the performance of activity recognition; the non-informative sequence features and the class imbalance problem. In the next section, we present the proposed approach using KNN-WSVM.

3 Proposed Approach

3.1 System Overview

As pointed out in the introduction part of this paper, the core of our proposal strategy relies on the use of a temporal feature (TF) with the new hybrid model combining KNN and WSVM. This fusion is performed as follows: we added two new features into original data (binary state vectors). The first corresponds to the predicted Labels

generated by KNN classifier. The second is the temporal feature (TF). This new distribution data is therefore used for learning and testing the multi-class weighted SVM classifier using a new automated criterion for weighting the data. The choice of WSVM like classifier [12, 13] in this context is motivated by previous work as shown in [11]. A generic diagram is shown in Fig. 1. Particularly, the initial dataset is divided into training and test datasets.

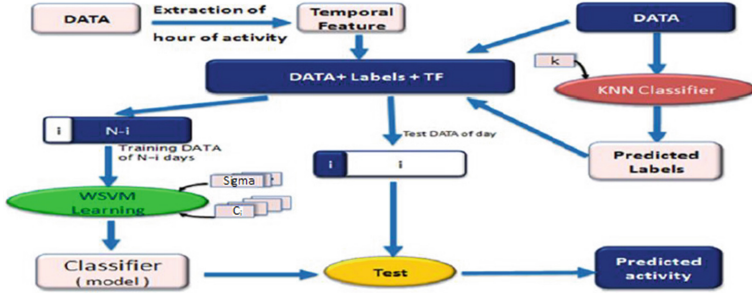


Fig. 1. Multi-class structure of TF-KNN-WSVM

In this work, we improve the classification performances of class activities by introducing the feature insertion stage. We added a new feature to the existent data matrix. This attribute corresponds to the hour of beginning of the activity. We extract this feature directly from the data structure. The sensor activations are collected by the state-change sensors distributed all around the environment. To find out the sensors ID corresponding to the room label of performed activity, we search the different object types he is manipulating in the sensors, see the below figure. ID: is a number representing the sensor ID. Each sensor has its own unique ID (Fig. 2).

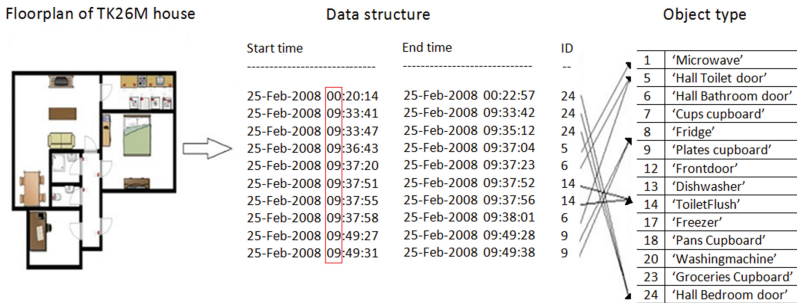


Fig. 2. Temporal feature (TF) for TK26 M dataset. In red, the hour of beginning activity.

3.2 K-Nearest Neighbor (KNN)

The K -nearest neighbor algorithm is amongst the simplest of all machine learning algorithms [3], and therefore easy to implement. The m training instances $x \in R^n$ are

vectors in an n -dimensional feature space, each with a class label. In the KNN method, the result of a new query is classified based on the majority of the KNN category. This rule is usually called ‘voting KNN rule’. The classifiers do not use any model for fitting and are only based on memory to store the feature vectors and class labels of the training instances. They work based on the minimum distance from an unlabelled vector (a test point) to the training instances to determine the K -nearest neighbors. K (positive integer) is a user-defined constant. Usually Euclidean distance is used as the distance metric.

3.3 Support Vector Machines

Support Vector Machines (SVM) is based on statistical learning theory developed by Vapnik [12]. For a two class problem, we assume that we have a training set $\{(x_i, y_i)\}_{i=1}^m$ where the observations are $x_i \in R^n$ and y_i are class labels either 1 or -1 . The primal formulation of the margin in SVM maximizes margin $2/K(w, w)$ between two classes and minimizes the amount of total misclassifications (training errors) ξ_i as follows:

$$\begin{aligned} \min_{w, b, \xi} \quad & 1/2 \cdot K(w, w) + C \sum_{i=1}^m \xi_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, m \end{aligned} \quad (1)$$

where w is normal to the hyperplane, b is the translation factor of the hyperplane to the origin and $\phi(\cdot)$ is a non-linear function which maps the input space into a feature space defined by $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$.

We used the Gaussian kernel as follows: $K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2)$ where σ is the width parameter. The construction of such functions is described by the Mercer conditions [13]. The regularization parameter C is used to control the trade-off between maximization of the margin width and minimizing the number of training error.

Solving the formulation dual of SVM [13] gives a decision function in the original space for classifying a test point $x \in R^n$

$$f(x) = \text{sgn} \left(\sum_{i=1}^{m_{sv}} \alpha_i y_i K(x, x_i) + b \right) \quad (2)$$

with m_{sv} is the number of support vectors $x_i \in R^n$.

In this study, a software package LIBSVM [14] was used to implement the multiclass classifier algorithm. It uses the one-vs-one method [13].

3.4 Weighted Support Vector Machines

For daily recognition activities applications, especially in daily recognition tasks, the misclassification of minority class members due to large class imbalances is undesirable. An extension of the SVM, weighted SVM, was presented to cope with this

problem. Two different penalty constraints were introduced for the minority and majority classes:

$$\begin{aligned} \min_{w, b, \xi} \quad & 1/2 \cdot K(w, w) + C_+ \sum_{y_i=1} \xi_i + C_- \sum_{y_i=-1} \xi_i \\ \text{subject to} \quad & y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, m \end{aligned} \quad (3)$$

C_+ and C_- are cost parameters for positive and negative classes, respectively, to construct a classifier for multiple classes. They are used to control the trade-off between margin and training error.

Some authors [13, 15] have proposed adjusting different cost parameters to solve the imbalanced problem. Veropoulos et al. in [15] proposed to increase the cost of the minority class (i.e., $C_- > C_+$) to obtain a larger margin on the side of the smaller class. Huang et al. [4] raised a Weighted SVM algorithm. The coefficients are typically chosen as:

$$C_+ = C \times w_+ \quad (4)$$

$$C_- = C \times w_- \quad (5)$$

Where C is the common cost parameter of the WSVM. w^+ and w^- are the weights for +1 and -1 class respectively. They put forward the corresponding solutions to deal with this problem in the SVM algorithm like this:

$$\frac{C_+}{C_-} = \frac{m_-}{m_+} \quad (6)$$

To extend Weighted SVM to the *multi-class* scenario in order to deal with N classes (daily activities), we used different **misclassification C_i per class similar to [11]**. By taking $C_- = C_i$ and $C_+ = C$, with m_+ and m_i be the number of samples of majority classes and number of samples in the i^{th} class, the main ratio cost value C_i for each activity can be obtained by:

$$C_i = \text{round}(C \times [m_+ / m_i]) \quad i = 1, \dots, N \quad (7)$$

[] is integer function. This criterion respects this reasoning that is to say that the tradeoff C_- associated with the smallest class is large in order to improve the low classification accuracy caused by imbalanced samples. It allows the user to set individual weights for individual training examples, which are then used in WSVM training.

4 Simulation Results and Assessment

4.1 Datasets

For the experiments, we use an openly datasets gathered from three houses having different layouts and different number of sensors [6, 7]. Each sensor is attached to a wireless sensor network node. The activities performed with a single man occupant at each house are different from each other. Data are collected using binary sensors such as reed switches to determine open-close states of doors and cupboards; pressure mats to identify sitting on a couch or lying in bed; mercury contacts to detect the movements of objects like drawers; passive infrared (PIR) sensors to detect motion in a specific area; float sensors to measure the toilet being flushed. Binary sensor output is represented in a feature space that used by the model to recognize the activities. We choose the ideal time slice length for discretizing the sensor data $\Delta t = 60$ s. Time slices for which no annotation is available are collected in a separate activity labelled ‘Idle’. The data were collected by a Base-Station and labelled using a Wireless Bluetooth headset combined with speech recognition software or a Handwritten diary. Table 1 shows also the number of data per activity in each dataset.

Table 1. List of activities annotated for each dataset and the number of observations of each activity (.).

Dataset	Annotation	Activities
TK26 M	Bluetooth headset	Idle ₍₄₆₂₇₎ ; Leaving ₍₂₂₆₁₇₎ ; Toileting ₍₃₈₀₎ ; Showering ₍₂₆₅₎ ; Sleeping ₍₁₁₆₀₁₎ ; Breakfast ₍₁₀₉₎ ; Dinner ₍₃₄₈₎ ; Drink ₍₅₉₎
TK57 M	Bluetooth headset	Idle ₍₂₇₃₂₎ ; Leaving ₍₁₁₉₉₃₎ ; Eating ₍₃₇₆₎ ; Toileting ₍₂₄₃₎ ; Showering ₍₁₉₁₎ ; Brush teeth ₍₁₀₂₎ ; Shaving ₍₆₇₎ ; Sleeping ₍₇₇₃₈₎ ; Dressing ₍₁₁₂₎ ; Medication ₍₁₆₎ ; Breakfast ₍₇₃₎ ; Lunch ₍₆₂₎ ; Dinner ₍₂₉₁₎ ; Snack ₍₂₄₎ ; Drink ₍₃₄₎ ; Relax ₍₂₄₃₅₎
OrdonezA	Handwritten diary	Idle ₍₁₃₀₇₎ ; Sleeping ₍₇₈₈₆₎ ; Toileting ₍₁₇₃₎ ; Showering ₍₁₂₁₎ ; Breakfast ₍₁₃₂₎ ; Grooming ₍₁₅₄₎ ; Spare_Time/TV ₍₈₆₄₆₎ ; Leaving ₍₁₆₉₂₎ ; Lunch ₍₃₃₁₎ Snack ₍₁₄₎

4.2 Setup and Performances Measures

We splitted the initial dataset into training and testing subsets using the ‘Leave one day out’ approach, retaining one full day of sensor readings for testing and using the remaining sub-samples as training data. Sensors outputs are binary either ‘0’ or ‘1’ and represented in a feature space which is used by the model to recognize the activities. As the activity instances were imbalanced between classes. We evaluate the performance of our models using the F-measure, which is calculated from the Precision and Recall scores. On the other, in order to evaluate the sensitivity of the classifiers, the notions of true positive (TP), false negatives (FN) and false positives (FP), have also been implemented. These measures are calculated as follows:

$$\begin{aligned}
F - Measure &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \text{ with Precision} \\
&= \frac{1}{N} \sum_{i=1}^N \left[\frac{TP_i}{TP_i + FP_i} \right]; \text{ Recall} = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FN_i}
\end{aligned} \quad (8)$$

4.3 Results

We optimized the SVM hyper-parameters (σ , C) for all training sets in the range [0.1–2] and {0.001, 0.01, 0.1, 1, 5}, respectively, to maximize the error rate of Leave-one day-out cross-validation technique. Then, for WSVM classification method, we optimized locally the cost parameter C_i adapted to different classes [11]. In Table 2, the results show that the proposed method TF-KNN-WSVM outperforms SVM, KNN, WSVM, KNN-WSVM and the baseline method PCA-LDA-WSVM [11]. TF-KNN-WSVM model using TF contributes to significantly enhance the performance of KNN-WSVM classifier. One also notices that the WSVM is better than SVM and KNN for recognizing activities. For KNN method, we used also cross-validation technique to select the optimal value of K parameter in the range [1–9].

Table 2. Recall, Precision, F-Measure and Accuracy results for all approaches in (%). Bold values are the results for the proposed approaches for each dataset.

Dataset	Approach	Recall	Precision	F-Measure	Accuracy
TK26M	SVM _($\sigma=1.3$, $C=1$)	61.64	74.43	67.43	95.49
	KNN _(K=9)	67.87	75.84	71.63	94.39
	WSVM _($\sigma=0.8$, $C=0.01$)	75.56	76.53	76.04	92.46
	KNN-WSVM _($\sigma=1.4$, $C=0.01$)	57.43	78.3	76.84	93.86
	TF-KNN-WSVM _($\sigma=1.5$, $C=0.01$)	79.18	79.1	79.14	92.47
TK57M	SVM _($\sigma=1.2$, $C=1$)	35.83	36.33	36.08	80.89
	KNN _(K=7)	35.09	34.22	34.65	79.25
	WSVM _($\sigma=2$, $C=0.01$)	37.86	37.86	39.27	77.08
	KNN-WSVM _($\sigma=1.3$, $C=0.01$)	43.10	41.84	42.46	76.98
	TF-KNN-WSVM _($\sigma=1.3$, $C=0.01$)	47.21	51.93	49.46	76.80
OrdonezA	SVM _($\sigma=1.9$, $C=5$)	60.15	58.31	59.22	90.16
	KNN _(K=9)	55.43	56.67	56.04	84.19
	WSVM _($\sigma=1.4$, $C=0.01$)	66.52	68.27	67.38	84.36
	KNN-WSVM _($\sigma=1$, $C=0.001$)	62.39	69.79	65.89	77.36
	TF-KNN-WSVM _($\sigma=2$, $C=0.1$)	69.94	71.26	70.60	76.48

We report in Fig. 3, the classification results in terms of accuracy measure for each class for TK26M dataset. We show that our proposed combination outperforms the other approaches for ‘Toileting’, ‘Breakfast’ and ‘Dinner’ activities and similar results with other methods for ‘Leaving’, ‘Showering’ and ‘Sleeping’ activities. The majority activities ‘Leaving’ and ‘Sleeping’ are better for all methods while the ‘Idle’ activity is

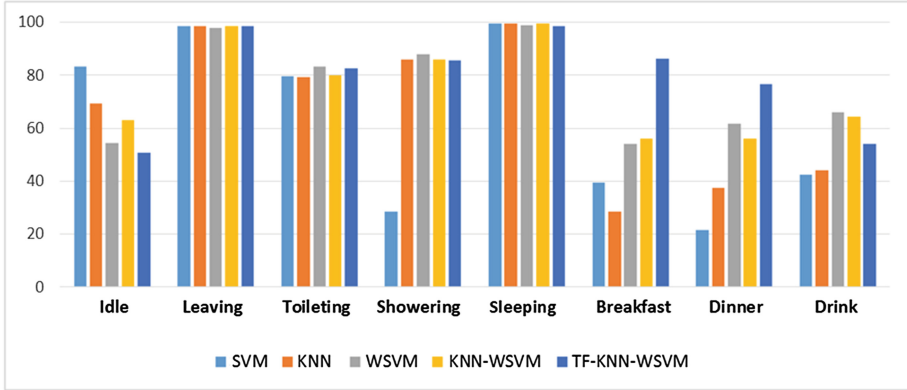


Fig. 3. Accuracy recognition rate for each activity for TK26M dataset.

less accurate for the proposed method compared to other methods. Additionally, the kitchen-related activities as ‘Breakfast’, ‘Dinner’ and ‘Drink’ are in general harder to recognize than other activities.

In order to quantify the extent to which one class is harder to recognize than another one, we analyzed the confusion matrix of TF-KNN-WSVM for TK26M dataset in Table 3. One notices that the activities ‘Leaving’, ‘Toileting’, ‘Showering’, ‘Sleeping’, ‘Breakfast’ and ‘Dinner’ are better recognized comparatively with ‘Idle’ and ‘Drink’. The kitchen activities seem to be more recognized using the proposed method. The high performance obtained in the case of TK26M dataset, which seems to be less vulnerable to class-overlapping than other datasets. This overlapping between the activities is due to the layout of the house. In the TK26M house, there is a separate room for almost every activity. The kitchen activities are food-related tasks, they are worst recognized because most of the instances of these activities were performed in the same location (kitchen) using the same set of sensors. Therefore the location of sensors strongly influences the recognition performance.

Table 3. Confusion matrix (values in %) for TF-KNN-WSVM for the TK26M dataset.

Activity	Id	Le	To	Sh	Sl	Br	Di	Dr
Id	50.94	8.21	3.35	8.41	3.39	11.84	9.96	3.89
Le	1.09	98.42	0.2	0.26	0.0	0.0	0.02	0.0
To	8.95	2.89	82.63	0.79	1.84	2.11	0.26	0.53
Sh	3.4	0.0	4.91	85.66	0.0	6.04	0.0	0.0
Sl	0.94	0.0	0.4	0.0	98.66	0.0	0.0	0.0
Br	10.09	0.0	0.92	0.0	0.92	86.24	0.0	1.83
Di	11.78	0.57	1.15	0.0	0.0	0.0	76.72	9.77
Dr	13.56	3.39	3.39	0.0	0.0	5.08	20.34	54.24

5 Conclusion

Our experiments on real-world datasets from smart home environment showed that TF-KNN-WSVM strategy can significantly increase the recognition performance to classify multiclass sensory data, and can improve the prediction of the minority activities. It significantly outperforms the results of the typical methods KNN, SVM, WSVM. TF-KNN-WSVM is better than KNN-WSVM. The space features needs a prior knowledge about the smart home using the temporal feature, which makes a model very specific for that environment.

The location attribute can also discriminate between the different activity classes that performed in different locations. In the future, it will be interesting to use both spatial and temporal features to improve the activity classification performance. Also, we are going to test the scalability of our approach by considering datasets containing increased classes and various amounts of sensors.

References

1. Brumitt, B., Meyers, B., Krumm, J., Kern, A., Shafer, S.: Easyliving: technologies for intelligent environments. In: *Handheld and Ubiquitous Computing*, pp. 97–119. Springer (2000)
2. Abidine, M.B., Fergani, L., Fergani, B., Fleury, A.: Improving human activity recognition in smart homes. *Int. J. E-Health Med. Commun. (IJEHMC)* **6**(3), 19–37 (2015)
3. Darko, F., Denis, S., Mario, Z.: Human movement detection based on acceleration measurements and k-NN classification. In: *The International Conference on Computer as a Tool, EUROCON 2007*, pp. 589–594. IEEE, September 2007
4. Huang, Y.M., Du, S.X.: Weighted support vector machine for classification with uneven training class sizes. In: *Proceedings of the IEEE International Conference on Machine Learning and Cybernetics*, vol. 7, pp. 4365–4369 (2005)
5. Fleury, A., Noury, N., Vacher, M.: Improving supervised classification of activities of daily living using prior knowledge. *Int. J. E-Health Med. Commun.* **2**(1), 17–34 (2011)
6. Kasteren, T.V., Noulas, A., Englebienne, G., Krose, B.: Accurate activity recognition in a home setting. In: *UbiComp 2008*, pp. 1–9. ACM, New York (2008)
7. Ordonez, F.J., de Toledo, P., Sanchis, A.: Activity recognition using hybrid generative/discriminative models on home environments using binary sensors. *Sensors* **13**, 5460–5477 (2013)
8. Logan, B., Healey, J., Philipose, M., Tapia, E.M., Intille, S.: A long-term evaluation of sensing modalities for activity recognition. In: *Proceedings of the 9th International Conference on Ubiquitous Computing*, pp. 483–500. Springer, Berlin (2007)
9. Abidine, M.B., Fergani, B.: Evaluating C-SVM, CRF and LDA classification for daily activity recognition. In: *Proceedings of IEEE ICMCS, Tangier-Morocco*, pp. 272–277. IEEE (2012)
10. Chen, D.R., Wu, Q., Ying, Y., Zhou, D.X.: Support vector machine soft margin classifiers: error analysis. *J. Mach. Learn. Res.* **5**(Sep), 1143–1175 (2004)
11. Abidine, M.B., Fergani, L., Fergani, B., Oussalah, M.: The joint use of sequence features combination and modified weighted SVM for improving daily activity recognition. In: *Pattern Analysis and Applications*. Springer, London, August 2016
12. Cortes, C., Vapnik, V.: Support vector network. *Mach. Learn.* **20**, 1–25 (1995)

13. Osuna, E., Freund, R., Girosi, F.: Support vector machines: training and applications. Technical report. Massachusetts Institute of Technology, Cambridge, MA, USA (1997)
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2**, 1–27 (2011). <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
15. Veropoulos, K., Campbell, C., Cristianini, N.: Controlling the sensitivity of support vector machines. In: *Proceedings of the International Joint Conference on AI*, pp. 55–60 (1999)