Pham Quoc Thang, 2024, 12:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Vector Machine Learning Models for Human Gesture Recognition

Pham Quoc Thang, Hoang Thi Lam
Tay Bac University, Vietnam

Abstract- Human gesture recognition is a complex yet critical task in the field of computer vision, driven by advancements in motion-capture technology and the availability of devices like Microsoft's Kinect sensor. This paper explores the application of vector machine learning models, specifically Support Vector Machines (SVM), Simplified Support Vector Machines (SimpSVM), and Relevance Vector Machines (RVM), to the problem of human gesture recognition. Our experiments on the Microsoft Research Cambridge-12 Kinect dataset demonstrate that these vector machine learning models achieve high accuracy and competitive performance in gesture classification. SVM and SimpSVM, in particular, exhibit superior accuracy compared to RVM, though RVM shows advantages in classification speed due to fewer support vectors. This study confirms that vector machine learning models are effective for human gesture recognition, providing a promising direction for future research and application in interactive systems.

Keywords- Support Vector Machines, Relevance Vector Machines, Human Gesture Recognition.

I. INTRODUCTION

Recognizing objects in images is easy for humans but a complex problem for machines. A recognition system includes main components: an image sensor (or camera to capture images), image preprocessing, object detection, object segmentation, feature extraction, and object classification. The recognition system includes a sample image database that is classified for training. When an object appears, the image will be classified into an appropriate category. Recently, the problem of recognizing human gestures/actions has attracted much attention in the computer vision community due to the significant price reduction in motion-capture devices and the support of software development kits (SDK) by many giant companies, for example, Microsoft's Kinect sensor (Figure 1).

Human gestures/actions are used to express intentions or convey messages naturally, such as using hands to convey sign language for the deaf. Clearly, these gestures are immediately recognized by humans, but for computers, it's difficult due to the complexity of human movements. To address these challenges, gesture recognition models have been proposed: SVM [1], Hidden Markov Models (HMMs)

However, most of these models are often complex, making generalization difficult.

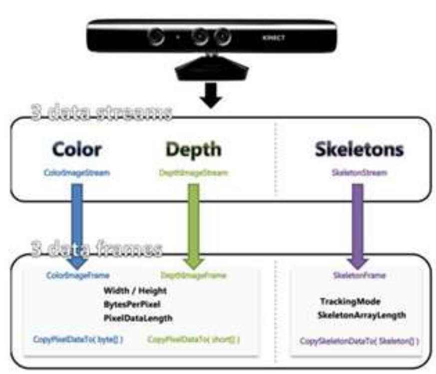

Figure 1: Microsoft's Kinect sensor

One of the unique features of the Kinect sensor is the skeletal data captured. The Kinect sensor provides 20 3D coordinates of joints, through which body movements are recorded and recognized (Figure 2). Like other recognition requirements, the two main challenges to be addressed are feature extraction and recognition.

Figure 2: Kinect sensor captures skeletal data

Vector machine learning models have shown success in pattern recognition tasks such as speech recognition, number recognition, and handwriting recognition. In this paper, we study the effectiveness of vector machine learning models when applied to the problem of human gesture recognition. We experiment to show that vector machine learning models can achieve very good gesture classification performance.

The rest of this paper is structured as follows: Section II presents vector machine learning models such as SVM, SimpSVM, and RVM. Section III presents the classification problem. Section IV describes the problem of human gesture recognition along with experimental results. Finally, there is the conclusion section.

II. VECTOR MACHINE LEARNING MODELS

Basis function model is a basic model that has a simple structure, in the form of a linear combination of basis functions:

$$f(x;w) = \sum_{i=1}^{M} w_i \phi_i(x) + b \qquad (1)$$

Here, it is common to use the radial basis function (RBF)

$$\phi_i(x) = \exp(-\gamma \|x - c_i\|^2), \; i = 1,.., M$$

depends only on the distance from the argument x to a given point ci (called the center) with width γ and M is the number of radial basis functions Φ_i of the model is used to calculate the function f.

1. Support Vector Machine (SVM)

SVM (Support Vector Machines) works in feature space F via a kernel function *K(x, y) = Φ(x).Φ(y)* where Φ: $R^d \rightarrow F$ is a map from the *d*-dimensional input space to a possibly high-dimensional feature space [3]. For a two-class classification problem, thedecision rule takes the form:

$$y = sign\left(\sum_{i=1}^{M} \alpha_i K(x, x_i) + b\right) \qquad (2)$$

where αi are weights of support vectors xi, x is the input vector needed to classify,

$$K(x, y) = \Phi(x).\Phi(y) \,,\; K(x, x_i) = \Phi(x).\Phi(x_i) \qquad (3)$$

is a kernel function calculating the dot product of two vectors Φ(x) and Φ(xi) in the feature space, b is the bias, and M is the number of support vectors. The task of the SVMs training process is to determine all the parameters (xi, αi, b, M); the resulting xi, i = 1,..,M are a subset of the training set and are called support vectors (SVs).

2. Simplification of Support Vector Machine (SimpSVM)

SimpSVM try to approximate the normal vector Ψ of the separating hyperplane

$$\Psi = \sum_{i=1}^{N_s} \alpha_i \Phi(x_i) \qquad (4)$$

expanded in images of input vectors xi ∈ R^d, αi ∈ R, by a reduced set expansion

$$\Psi' = \sum_{i=1}^{N_z} \beta_i \Phi(z_i) \qquad (5)$$

with NZ < NS, zi ∈ R^d, βi ∈ R. To classify a new test point x, calculation (2) is replaced by

$$y = sign\left(\sum_{i=1}^{N_Z} \beta_i K(x, z_i) + b \right) \quad (6)$$

The goal of the reduced set methods is to choose the smallest number NZ < NS, and the corresponding reduced set $\{(z_i , \beta_i)\}$ such that any resulting loss in generation performance remains acceptable.

The solution of SVM also can be analyzed from a mechanical point of view: if each image of support vectors exerts a force $F_i = \alpha_i \hat{\Psi}$ on the decision hyperplane, then the SVM's solution satisfies the conditions of equilibrium: the sum of the forces and the torque all vanish ($\hat{\Psi}$ is the unit vector in the direction Ψ) [4]. In an equilibrium system, if we replace two member forces by an equivalent one, then the equilibrium state of the system will not change. In a SVM's solution, if we replace two images $\Phi(x_i)$ and $\Phi(x_j)$ of two support vectors belonging to the same class xi and xj by a vector M = $m\Phi(x_i) + (1 - m)\Phi(x_j)$, where m = αi/(αi + αj) and weight vector M by $\alpha_m = (\alpha_i+\alpha_j)$, then for any point x in the input space, calculation (2) can be computed through (N_S-1) vectors:

$$y = sign\left(\sum_{k=1, k\neq i, k\neq j}^{N_s} \alpha_k K(x, x_k) + \alpha_m M.\Phi(x_m) + b \right) \quad (7)$$

3. Relevance Vector Machine (RVM)

Given a two-class dataset T = $\{(x_i,y_i), x_i \in R^d, y_i \in \{0,1\}, i = 1,\ldots,n\}$, the RVM [6] uses an assumption that y has a Bernoulli distribution and yi are independent. Likelihood of the training dataset for the parameters w_i is:

$$P(y \mid w) = \prod_{i=1}^{n} \delta(f(x))^{y_i} (1-\delta(f(x)))_i^{1-y_i} \quad (8)$$

in which $\delta(y) = 1/(1+e^{-y})$ is a logistic function whose input is the value of the linear function (1).

To limit the number of components wi ≠ 0, Mike Tipping [6] uses additional prior constraints that each parameter has a normal distribution with a mean of 0 and a hyperparameter αi for the variance:

$$p(w \mid \alpha) = \prod_{i=1}^{n} N(w_i \mid 0, \alpha_i^{-1}) \quad (9)$$

Constraint (9) is set up for two purposes. The first purpose is to make model (1) simpler to avoid over-fitting in training. The second purpose is that model (1) will run faster by using fewer basis vectors.

III. MULTI-CLASS CLASSIFICATION

Both SVM (or SimpSVM) and RVM were originally designed for two-class classification problem. For multi-class sign recognition task, we need to use multiple two-class classifiers. In this work, we use both the one-vs-one majority vote one-vs-all strategies to analyze and compare predictive performance of classifiers. More specifically, the classification result is based on the vote:

$$y = \arg\max_i \sum_{j\neq i} h_{ij}(x), \quad (10)$$

where

$$h_{ij}(x) = \begin{cases} 1 & \text{if } f_{ij}(x) \geq 0 \\ \ldots \end{cases} \quad (11)$$

for the one-vs-all classifier.

In the next sections, we study the feasibility and effectiveness of the vector machine learning models in the human gesture recognition task. We will focus on the predictive performance and recognition speed of the vector machine learning models.

IV. EXPERIMENT

1. Data Sets

The Microsoft Research Cambridge-12 Kinect dataset [7] includes sequences of human movements represented as body part positions and related gestures recognized by the system. The dataset includes 594 sequences and 719,359 skeletal frames collected from 30 participants performing 12 types of gestures. Some examples are shown in Figure 3.

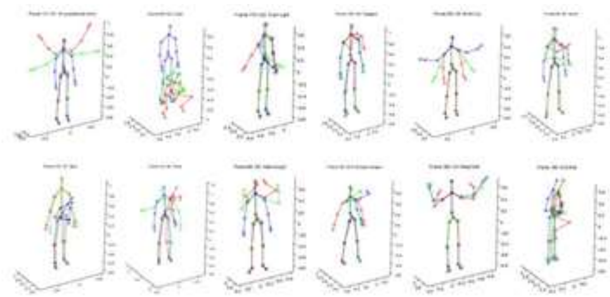

Figure 3: Illustration of examples of 12 gestures in MSRC-12

2. Feature Extraction

The data of the human gesture recognition system transmitted from the Kinect sensor is in numerical form. The paper chooses to use traditional manual feature extraction methods. From the twenty 3D coordinates of the skeletal joints, different feature extraction methods are proposed. A simple method is to use joint angle velocity and joint angular velocity of consecutive frames as described in [8]. For each pair of adjacent frames, these data are considered as components of the feature vector used separately or combined.

Additionally, more complex feature extraction methods are also proposed. Wang, Liu, Wu, and Yuan [9] use data based on the skeleton, such as the relative position of joints, relative disparity, and normalized trajectory of movements. In each skeletal frame, LOP features calculate the local occupancy based on a set of 3D points around a specific joint. Therefore, the dynamics over time of all occupancy values are generally separated into different types of operations. Xia, Chen, and Aggarwal [10] divide postures in 3D space into "bins"; the histogram of 3D joint positions (HOJ3D) is calculated based on action sequences then clustered into k postures. The temporal evolutions of those postures are represented by a discrete Hidden Markov Model (HMM).

In this paper, three types of extracted information are used: relative velocity points (veloPoints), joint angles (Angles), and joint angular velocities (veloAngles). From 20 3D data points, the velocity points are calculated by 60 x, y, z disparities in two consecutive frames. Joint angles (measured in radians) are 35 angles at the arm joints (shoulder, elbow, wrist), leg joints (hip, knee, ankle), symmetrical joints (shoulder center, hip center, spine), and the absolute camera angle (formed by point 0, shoulder center, and hip center, arm and leg joints). Angular velocities are the angle differences between angles in two consecutive frames.

3. Parameter Selection

The parameter γ of the radial basis function plays an important role in the SVM, SimpSVM, and RVM models. The paper conducts a search for the most suitable values of the parameter γ so that the trained SVM, SimpSVM, and RVM models have good prediction accuracy with different feature datasets. The search range for γ is (2-i, 2i), i = -15, .., 10. For all models, the parameter C is set to 1. According to the search results, the most suitable γ values for SVM, SimpSVM, and RVM are 21, 2-8 and 2-5 respectively on the veloPoints, Angles, and Veloangles datasets.

Table-1: Human Gesture Recognition Classification Results

		Velo points	Velo angles	Angles
SVM	Time (s)	230	289	95
	Acc (%)	91.77	83.04	98.75
	F1 (%)	91.83	83.44	98.74
	SVs	2991	4104	2241
SimpSVM	Time (s)	804	52542	1608
	Acc (%)	91.15	82.75	98.74
	F1 (%)	91.2	83.15	98.72
	SVs	1348	2717	452
RVM	Time (s)	4761	6909	3526
	Acc (%)	88.66	78.56	97.59
	F1 (%)	88.78	79.08	97.57
	SVs	779	1280	262

4. Human Gesture Classification

The paper chooses to use and implement the 5-fold cross-validation recognition method: dividing the datasets into 5 equal-sized subsets. In each recognition iteration, 4 subsets are used for training and the remaining subset for recognition. This process is repeated until all subsets are used.

The specific experimental results are shown in Table From these experimental results, the following points can be noted:

Regarding Recognition Accuracy

Illustrated visually through the chart in Figure 4. RVM shows lower classification accuracy (along with the harmonic mean F1 score) than SVM and SimpSVM on all three feature types. The reason might be that RVM significantly eliminates many basis vectors, reducing prediction accuracy.

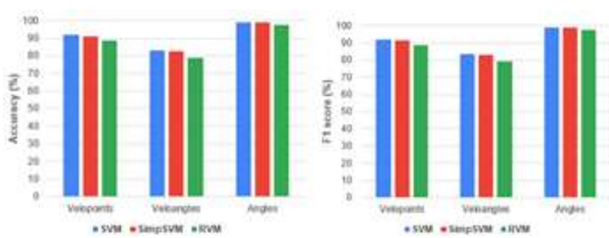

Figure 4: Accuracy and F1 score of SVM, SimpSVM, and RVM models in human gesture classification

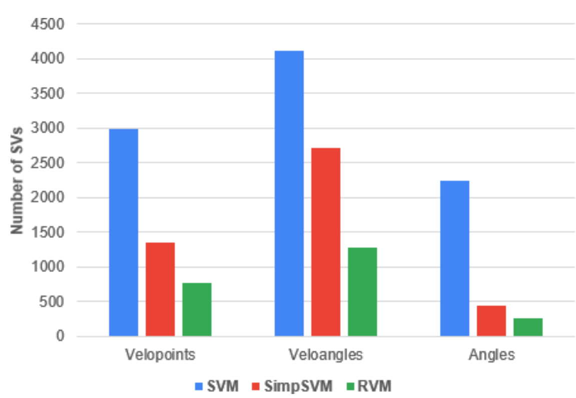

Figure 5: Number of basis vectors of the models learned by SVM, SimpSVM, and RVM in human gesture classification

However, in Figure 5, the smallest number of reduced SVs belongs to RVM (262 SVs) compared to SVM (2241 SVs) and Simp SVM (452 SVs), making RVM faster in the recognition phase. Additionally, the simplified SVM (SimpSVM) significantly reduces the number of SVs of SVM while maintaining high classification accuracy (the error rate increases in significantly).

Regarding recognition phase time: the recognition time is also measured to compare the speed of the models. Table 2 reports the recognition phase time of the SVM, SimpSVM, and RVM models.

The SimpSVM and RVM models have faster classification speeds than the SVM model because they have significantly fewer SVs than the SVM model obtained after the training phase. With the fewest SVs in the model (262 RV), the RVM model shows superior classification speed compared to the other models in the human gesture classification experiment.

Table-2: Recognition Phase Speed of Methods in Human Gesture Recognition

Models	Recognition Phase Time *(ms)*	
	Velopoints	Angles
SVM	34.98	38.73
SimpSVM	12.59	13.93
RVM	6.97	7.72

The paper also compares the experimental results with recent research results by other authors on the same Microsoft Research Cambridge-12 Kinect dataset in Table 3. The SVM and SimpSVM models have higher results than most other methods, except for the results of Li, Zhang, Liao, Jin, and Yang in [11].

However, in [11], the authors only experimented with the classification of 6 gesture classes out of 12 gesture classes, while this paper experiments with the full classification of 12 gesture classes, which is more complex. These results show that the SVM, SimpSVM, and RVM models allow for quite effective human gesture recognition classification.

Table-3: Comparison of Machine Learning Models Used for Human Gesture Classification

Study	Classification Results (Accuracy %)
Li, Zhang, Liao, Jin, Yang [11] (Using 3S Net TTM)	99.01
Wang, Li, Chuankun, Hou [12] (Using CNN)	93.12
Garcia-Hernando, Kim [13] (Using tổ hợp DT)	98.25
Pfitscher, Welfer, Souza Leite Cuadros, Gamarra [14] (Sử dung DCNN)	90.78
Liu, Liu, Chen [15] (Using CNN)	96.62
Truong, Zaharia [16] (Using LMA và HMM)	88.6
Ajili, Mallem, Didier [17] (Using DHMM)	96.33
El-Ghaish, Shoukry, Hussein [18] (Using CovP3DJ)	98.45
Barmpoutis, Stathaki, Camarinopoulos [19](Using DCA và NN)	92.8
Konstantinidis, Dimitropoulos, Daras [20] (Using LSTM and Grassmannian Pyramids)	94.6
Choi, Kim [21](Using DTW)	98.46
Zhu, Yue, Xia, Dong, Qunfei [22] (Using PFR)	96
Peixoto, Pfitscher, Souza Leite Cuadros, Welfer, Gamarra [23](Using CNN)	92.48
SVM	98.75
SimpSVM	98.74
RVM	97.59

V. CONCLUSION

This study demonstrates the feasibility and effectiveness of vector machine learning models, specifically SVM, SimpSVM, and RVM, for human gesture recognition. Through comprehensive experiments using the Microsoft Research Cambridge-12 Kinect dataset, we have shown that these models can achieve high accuracy in classifying human gestures. The results indicate that SVM and SimpSVM offer superior accuracy compared to RVM. However, RVM stands out with a significantly faster recognition phase due to its minimal use of support vectors, making it advantageous in scenarios requiring rapid classification. SimpSVM effectively balances between the complexity and performance by reducing the number of support vectors while maintaining high accuracy.

In conclusion, vector machine learning models, particularly SVM and SimpSVM, are shown to be highly effective for human gesture recognition, offering a promising direction for future research and practical applications in interactive systems and human-computer interaction.

REFERENCES

1. Rahman, M., Afrin, J. (2013), "Hand Gesture Recognition using Multiclass Support Vector Machine", International Journal of Computer Applications, 74, pp. 39–43.
2. Lee, H.-K., Kim, J. (1999), "An HMM-based threshold model approach for gesture recognition", IEEE Transactions on Pattern Analysis and Machine Intelligence, 21 (10), pp. 961–973.
3. Cortes, C., Vapnik, V. (1995), " Support-Vector Networks", Mach. Learn., 20 (3), 273–297.
4. Burges, C. J. C. (1996), "Simplified Support Vector Decision Rules", in: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning, ICML'96, Bari, Italy: Morgan Kaufmann Publishers Inc., 71–77.
5. DucDung Nguyen, TuBao Ho (2005), "An efficient method for simplifying support vector machines," in Proceedings of the 22nd international Conference on Machine Learning, ICML 2005, Bonn, Germany, August 07- 11, 2005, vol. 119, pp. 617624. ACM, New York
6. Tipping, M. E. (1999), "The Relevance Vector Machine", in: Proceedings of the 12th

International Conference on Neural Information Processing Systems, NIPS'99, Denver, CO: MIT Press, 652–658.

7. Fothergill, S., Mentis, H., Kohli, P., Nowozin, S. (2012), "Instructing People for Training Gestural Interactive Systems", in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA: Association for Computing Machinery, 1737–1746.
8. Nguyen, D.-D., Le, H.-S. (2015), "Kinect Gesture Recognition: SVM vs. RVM", in: 2015 Seventh International Conference on Knowledge and Systems Engineering (KSE), pp. 395–400.
9. Wang, J., Liu, Z., Wu, Y., Yuan, J. (2012), "Mining actionlet ensemble for action recognition with depth cameras", in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1290–1297.
10. Xia, L., Chen, C.-C., Aggarwal, J. K. (2012), "View invariant human action recognition using histograms of 3D joints", in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 20–27.
11. Li, C., Zhang, X., Liao, L., Jin, L., Yang, W. (2019), "Skeleton-Based Gesture Recognition Using Several Fully Connected Layers with Path Signature Features and Temporal Transformer Module", Proceedings of the AAAI Conference on Artificial Intelligence, 33, pp. 8585–8593.
12. Wang, P., Li, W., Chuankun, l., Hou, Y. (2016), "Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks", Knowledge-Based Systems, 158.
13. Garcia-Hernando, G., Kim, T. (2017), "Transition Forests: Learning Discriminative Temporal Transitions for Action Recognition and Detection", in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA: IEEE Computer Society, pp. 407–415.
14. Pfitscher, M., Welfer, D., Souza Leite Cuadros, M. A. de, Gamarra, D. F. T. (2020), "Activity Gesture Recognition on Kinect Sensor Using Convolutional Neural Networks and FastDTW for the MSRC-12 Dataset", in: Intelligent Systems Design and Applications, Cham: Springer International Publishing, pp. 230–239.
15. Liu, M., Liu, H., Chen, C. (2017), "Enhanced skeleton visualization for view invariant human action recognition", Pattern Recognition, 68, pp. 346–362.
16. Truong, A., Zaharia, T. (2016), "Dynamic Gesture Recognition with Laban Movement Analysis and Hidden Markov Models", in: Proceedings of the 33rd Computer Graphics International, CGI '16, Heraklion, Greece: Association for Computing Machinery, 21–24.
17. Ajili, I., Mallem, M., Didier, J.-Y. (2018), "An Efficient Motion Recognition System Based on LMA Technique and a Discrete Hidden Markov Model", in: 20th International Conference on Image Analysis and Processing (ICIAP 2018), Paris, France, Oct., pp. 707 –713.
18. El-Ghaish, H. A., Shoukry, A., Hussein, M. E. (2018), "CovP3DJ: Skeletonparts-based-covariance Descriptor for Human Action Recognition.", in: VISIGRAPP (5: VISAPP), pp. 343–350.
19. Barmpoutis, P., Stathaki, T., Camarinopoulos, S. (2019), "SkeletonBased Human Action Recognition through Third- Order Tensor Representation and Spatio-Temporal Analysis", Inventions, 4, p. 9.
20. Konstantinidis, D., Dimitropoulos, K., Daras, P. (2018), "Skeleton-Based Action Recognition Based on Deep Learning and Grassmannian Pyramids", in: 2018 26th European Signal Processing Conference (EUSIPCO), pp. 2045–2049.
21. Choi, H.-r., Kim, T. (2018), "Modified Dynamic Time Warping Based on Direction Similarity for Fast Gesture Recognition", Mathematical Problems in Engineering, pp. 1– 9.
22. Zhu, T., Yue, Z., Xia, Z., Dong, J., Qunfei, Z. (2018), "Progressive Filtering Approach for Early Human Action Recognition", International Journal of Control, Automation and Systems, 16, pp. 2393–2404.
23. Peixoto, J. S., Pfitscher, M., Souza Leite Cuadros, M. A. de, Welfer, D., Gamarra, D. F. T. (2021), "Comparison of Different Processing Methods of Joint Coordinates Features for Gesture Recognition with a CNN in the MSRC-12

Database", in: Intelligent Systems Design and Applications, Cham: Springer International Publishing, pp. 590–599.