Gesture recognition using Static Bayesian Tree algorithm

[1]Peter ADAMONDY (2nd year of master degree), [2]Peter PAPCUN (1st year)
Supervisor: [3]Ján Jadlovský
[1,2,3]Dept. of Cybernetics and Artificial Intelligence, FEI TU of Košice, Slovak Republic

[1]peter.adamondy@student.tuke.sk, [2]peter.papcun@tuke.sk, [3]jan.jadlovsky@tuke.sk

***Abstract* – We propose method to recognize movement gestures based on Static Bayesian Classification Tree algorithm. This algorithm is mergence of two classification methods. It inherits naive Bayesian classifier and implements it into predefined decision tree structure. Every node of this tree is independent Bayesian classifier trained only to distinguish between lower level nodes. This algorithm is very fast to compute and thanks to Bayesian probabilistic model it is also very robust.**

***Keywords* — Bayesian classification, decision tree, gesture recognition.**

I. Introduction

Human-machine interaction designs, as described by Guy A. [1], are in the present times shifting from the machine centered design (MCD) to the human centered design (HCD). The main idea of HCD is to take control of a machine as close as possible to human nature. Control of computer based systems relies strongly on the cognitive skills of their users. They need to learn how to control software and hardware to finish given task. HCD allows them to decrease learning time to minimum. Correct recognition of nature human behavior and response in adequate manner is great part of HCD. Here we will focus on dynamic human gesture recognition.

II. Overview

Our aim was to create simple, fast and robust algorithm that can be used not only for gesture recognition, but for various classification problems as well. For purposes of this paper we will focus on gesture recognition only.

First we need to specify what we mean by "gesture". Gesture is deterministic movement of human body part or parts. For example, if you want to say "*hello*" to someone, you can express that through waving gesture of your hand. Human brain recognizes that gesture and can respond to it accordingly i.e. wave back. In computer systems we use one or multiple cameras to catch visual objects and points of interest (POI). In this case POI is human hand. For standard cameras and computers is extracting of hand position (skeletal point) time demanding. We decided to use Kinect sensor that already has this ability. Computer will be employed only by gesture recognition task.

Static Naive Bayesian Tree algorithm (SNBT) or shorter Static Bayesian Tree (SBT) consists of three parts and its implementation can be divided to two stages. First part of algorithm is data extraction in form of vectors. After extracting of x, y and z coordinates, those vectors are passed to preprocessing stage of algorithm. We need to extract several features that will determine characteristics of specified gesture. During preprocessing stage we will merge all dimensions into one vector. From merged vector several features will be extracted:

- Mean crossing rate (MCR)
- Short time average energy (STAE)
- Standard deviation
- Mean

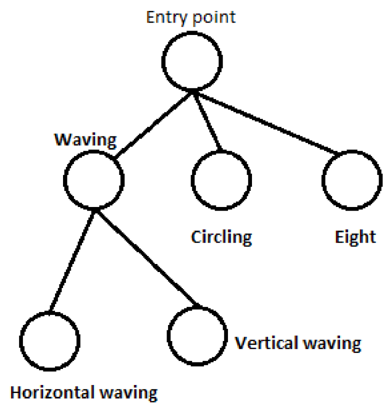

Fig. 1: Decision tree structure with Bayesian classifier nodes.

Those features are enough to correctly classify specific gestures. For our example four different gestures have been used:

- Horizontal waving
- Vertical waving
- "Eight"
- Circling

Those classes are structured into decision tree as seen on Fig. 1. All classes with their features are passed into learning stage of algorithm where mean and variance of features for each node is computed, so that they can be used later in classification stage.

Classification part of algorithm is of course separated and it will use precomputed parameters from the learning stage. Classified sample sets must be also preprocessed. Here we are extracting exactly same features as in preprocessing stage of learning part of algorithm.

III. Preprocessing and feature extraction

As mentioned above getting x, y and z coordinate vectors of POI must be preprocessed and few features must be extracted from them. Firstly vectors are merged into one by following formula:

$$f_i = \sqrt{(x_i^2 + y_i^2 + z_i^2)} \quad (1),$$

where f_i is merged value vector, and x_i, y_i, z_i are coordinate vectors. Significant amount of noise is always present when collecting data from sensors. Kinect is no exception. To cut off high frequency noise from merged vector, Butterworth low pass filter is applied [3].

After this procedure we can now extract features mentioned above. Mean crossing rate (MCR) is calculated as rate of crossing over mean value of given vector. Frequency is in direct proportion to MCR. Second feature is short time average energy (STAE). STAE is calculated by formula (2).

$$s_i = \sum \left| (\mathfrak{F}(f_i^2)) \right| \quad (2),$$

where, s_i is STAE value, f_i^2 is square values of merged vector and $\mathfrak{F}$ is Fourier transform of given vector. Mean μ_c as third value if self explanatory, as it is mean of all values in given vector. Last feature is standard deviation calculated by formula (3).

$$s_{dev} = \sqrt{\frac{1}{N} - 1 \sum_{i=1}^{N} (x_i - \overline{x})} \quad (3),$$

where s_{dev} is standard deviation, N is count of samples in given vector f_i, x_i is i-th sample and $\overline{x}$ is mean value.

IV. Background and calculations

Let $F = (e_1, ..., e_n)$ be a vector of features, where each feature takes values form its domain D_i as described in [2]. The set of all feature vectors is denoted $\Omega = D_1 \times \ldots \times D_n$. Let C be an unobserved random variable denoting *class* of an example, where C can take one of m values $c \in \{0, \ldots, m-1\}$.

A function g: $\Omega \rightarrow \{0, \ldots, m-1\}$, where $g(x) = C$, denotes a *concept* to be learned. Deterministic $g(x)$ corresponds to a concept without noise, which always assigns the same class to given example (e.g., disjunctive and conjunctive concept are deterministic).

A *classifier* is defined by a (deterministic) function $h: \Omega \rightarrow \{0, \ldots, m-1\}$ (a *hypothesis*) that assigns a class to any given example. Common approach is to associate each class i with a discriminant function $f_i(x)$, $i = 0, \ldots, m-1$, and let the classifier select the class with maximum discriminant function on a given example $h(x) = \arg\max_i f_i(x)$. The Bayes classifier uses as discriminant functions the class posterior probabilities given feature vector F. Applying Bayes rule gives

$$P(C=i | X=F) = \frac{P(X=F | C=i) P(C=i)}{P(X=F)} \quad (4),$$

where $P(X=F)$ is identical for all classes, and therefore can be ignored. This yields Bayes discriminant functions

$$f_i^x(F) = P(X=F | C=i) P(C=i) \quad (5),$$

where $P(X=F|C=i)$ is called as class-conditional probability distribution (CPD). Thus, the Bayes classifier finds the maximum *a posteriori* probability (MAP) hypothesis given example F.

Direct estimation of $P(X=F|C=i)$ from a given set of training examples is hard when the feature space is high-dimensional. Therefore using simplifying assumption that features are independent given the class. Another simplifying assumption is made as we assume that *a priori* probability of certain class is same for all classes. In other words we assume *equiprobability* of classes. We have no reason for this assumption so it might be bad idea. In order to decrease error risk distributed through learning we use only binary tree structure.

We are dealing here with continuous data sets. Typical assumption for this type of data is that values associated with each class are distributed according to Gaussian distribution. We firstly segment data by class, and then compute mean and variance of feature e_i in each class. Since we use tree structure, every analyzed vector e_i contains data from nodes or leaves of subtree that lies under currently calculated node. Probability of some value given a class $P(e_i=v|c)$ can be then computed by formula (6).

$$P(e_i = v | c) = \frac{1}{\sqrt{2\pi o_c^2}} e^{\frac{-(v-\mu_c)^2}{2 o_c^2}} \quad (6),$$

where $P(e_i = v|c)$ is probability of some value given class, μ_c is mean of the values in e_i associated with class c and o_c^2 is variance of the values in e_i in given class c.

Class with highest probability is considered as given class and therefor is assumed as currently performed gesture.

V. Gestures

As mentioned in section II. of this paper we are using four gestures. In this section we will focus on explaining each of them.

Horizontal waving movement is shown on Fig. 2. It is cyclic movement of POI between P1 and P2.

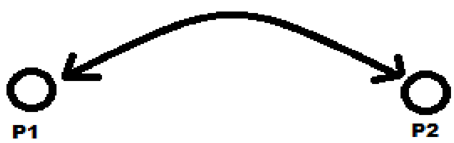

Fig. 2: Horizontal waving of POI

Vertical waving movement is shown on Fig.3. It is similar to the horizontal waving.

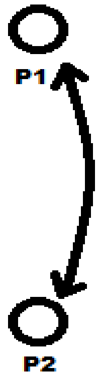

Fig. 3: Vertical wave of POI

Previous gesture types were pretty self explanatory. But following gestures need a little explanation. On Fig. 4 is circling movement of POI. We does not care about starting point of circling since we have implemented value vector shifting. Each collected frame of data are overlapping previous frame with specified amount of samples. With feature extraction from merged frames and strongly specifiable STAE parameter that is extracted from Fourier transform of data frames, we no longer need to care where will gesture start or end. This fact gives us high robustness for classification.

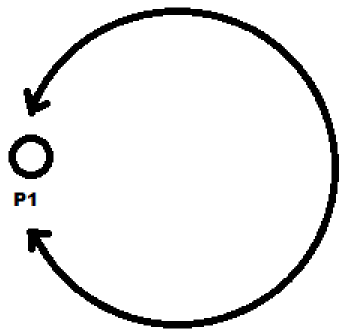

Fig. 4: Circling movement of POI.

The last gesture is so called “eight” which is similar to circular movement. Its scheme is drawn in Fig. 5.

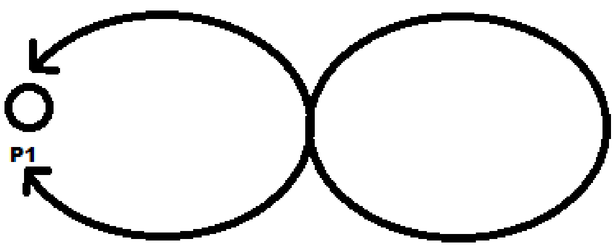

Fig. 5: "Eight" circling movement of POI.

We use only four types of gestures, but any number of gestures can be added as required. Bayesian classifier has several properties that make it very useful. In particular, the decoupling of the class condition feature distribution means that each distribution can be independently estimated as a one dimensional distribution. Combining with static tree structure it will eliminate problem with exponentially decreasing accuracy with increasing count of classes. Also more feature vectors can be added, that can increase accuracy of algorithm.

VI. Classification

To determine effectiveness of this classification algorithm we will use confusion table. In each cell of this table will be amount of data frames classified as probable class. We used 1000 samples for each class. If sample is classified properly it is added to cell where column and row has same name. Otherwise classification failed.

	horizontal wave	vertical wave	circling	„eight“	accuracy
horizontal wave	940	60	0	0	94,0000%
vertical wave	79	893	28	0	89,3000%
circling	0	42	872	86	87,2000%
„eight“	0	2	72	928	92,8000%
				overall accuracy	90,8250%

Tab. 1: Confusion matrix of SNBT algorithm.

VII. Conclusion and future work

Static Naive Bayesian Tree algorithm is easy fast and robust. As we shown in this paper it is also accurate with overall accuracy around 90.825% of properly classified samples. We found that significant impact on accuracy has size of data frame and size of overlapping vector. SNBT can be used for classification problems where classes are structured into static tree. Further research time should be focused on implementing Bayesian classifiers into dynamic tree structure which will be created during learning stage of algorithm.

In future, gesture recognition will be used for starting specific procedure. For example we are using kinect to control Mitsubishi robot. Robot is copying movement of human hand. We can use other hand to perform specific gestures, that will for example start recording of the robot movement or start some device connected on the robot's universal device slot (i.e. welding machine).

Acknowledgment

This work is the result of the project implementation: Development of the Center of Information and Communication Technologies for Knowledge Systems (ITMS project code: 26220120030) supported by the Research & Development Operational Program funded by the ERDF.

This work has been supported by the Scientific Grant Agency of Slovak Republic under project Vega No.1/0286/11 Dynamic Hybrid Architectures of the Multiagent Network Control Systems.

References

[1] Boy, G. A., “*Percieved Complexity and Cognitive stability in Human-Centered design*”, Proceedings of the HCI International 2007 Conference, Bejing,China.,(2007).

[2] Rish, I., *“Empirical study of the naive Bayes classifier”*. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, (2001).

[3] Butterworth S., *“On the Theory of Filter Amplifiers”*,In Wireless Engineer, vol.7,pp. 536-541, (1930)