

# A Survey on Recognition using Psychometric Testbased on Supervised Learning

Shashi Prabha Anan<sup>1</sup>, Dr. R. B. Gautam<sup>2</sup>, Dr. K.K.Maurya<sup>3</sup>, Yashi Gupta<sup>4</sup>

Guest Faculty, Department of Computer Science & Engineering, MCAET, Ambedkarnagar, Akabarpur, India

Assistant Professor, Department of Agricultural Engineering, MCAET, Ambedkarnagar, Akabarpur, India

Professor, Department of Agricultural Engineering, MCAET, Ambedkarnagar, Akabarpur, India

Research scholar, Department of Applied Science, G B Pant university of Agriculture and tech. Pantnagar, India

**ABSTRACT:** Since the beginnings of Computer Science, researchers have been trying to develop ways through which a machine can simulate certain human capabilities. The recent advances in Artificial Intelligence (AI) coupled with the ever increasing speed of decision making computers and access to cheaper technology have made tasks such as computer done Automatic Speech Recognition and Speech Synthesis, Speech recognition, Human face detection, selection, extraction and identification, and even computer aircraft piloting a possible reality, not an unimaginable future. Computer scientists, psychologists and other researchers have been trying to understand and emulate the human mind in various potentially useful applications. In this paper we discuss the computer as the security system to detect people and human features, such as long hair and smiles, from photographs of human faces. For this we are discussing various techniques particularly the kernel methods. Throughout the rest of the paper we discuss in detail how we can solve certain problems regarding human faces using several kernel methods.

**KEYWORDS:** Multilayer Layer perceptron (MLP), pattern recognition, Machine Learning, supervised learning, Automatic Speech Recognition and Speech Synthesis etc.

## I. INTRODUCTION

### 1.1 WHAT IS ARTIFICIAL INTELLIGENCE?

Artificial Intelligence is a branch of Computer Science that tackles different problems in a human- or biological-intelligence inspired way, and which relies on several other branches of Computer Science as well as other fields ranging from Psychology to Biology and Statistics. More concretely, AI solves problems heuristically in such areas as planning, searching, information processing and controlling devices. AI includes genetic algorithms, natural language processing, planning, searching, machine learning and others. In this paper we focus on machine learning. The algorithms in this paper primarily fall under the umbrella of supervised learning.

Machine learning is further subdivided into the following categories based on the type of information available:

- A. **Unsupervised Learning.** In this case the computer finds patterns without any intervention or help from the outside [9]. Applications include clustering of data and principal components analysis (PCA);
- B. **Reinforcement Learning.** In this case for every generated solution, the computer is given a score, with the ultimate goal of maximizing the score. The score reflects how closely the solution achieves the goal of the problem. Basically this trial and error approach involves several runs, during which the algorithm searches for a solution that will earn the highest score [15];
- C. **Supervised Learning.** For this approach the computer learns from examples given by a knowledgeable "teacher." At the end of the learning session, the goal is for the computer to return a correct solution on new, unseen data.

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 1, January 2019

## II. LITERATURE SURVEY

Mathematicians and statisticians have used kernels since the early 1960s. The machine learning community embraced kernel methods only in the last decade. In 1962, Parzan was among the first mathematicians to use an early notion of kernels [13]. Aizerman in 1964 recognized the “kernel trick” for avoiding lengthy computations in high dimensional spaces. Boser in 1992 is generally credited with the creation of Support Vector Machines (SVMs) [2]. He was among the first machine learning researchers to use the idea of a kernel to form a feature space. At the time, neural network research was popular and people were struggling with the idea of crafting feature spaces for neural network learning problems. Several years later, in 1997, Scholkopf found that kernels could be used to increase support vector machine (SVM) applications in non-linear domains [15, 17]. Soon after, the flood gates opened in terms of research with kernels applied to many known algorithms.

Historically, the three primary algorithms for face detection and feature identification have been Principal Components Analysis (PCA), Independent Component Analysis (ICA) and Linear Discriminant Analysis (LDA). These algorithms and techniques implement linear transformations (shearing, rotations and scaling). PCA is also known as eigen-value decomposition and Karhunen-Loeve transformation. PCA works by finding a new basis for a dataset such that the basis vectors are sorted by maximizing the variance in the data [4, 12]. ICA finds a new basis for the data such that the statistical dependencies in the data are minimized [1, 4]. LDA finds a basis such that the basis vectors are most capable of discriminating between the training categories [4, 10, 11]. Moreover, Support Vector Machines (SVMs) are used to find the hyper-plane that maximizes the distance separating the categories of points. Kernel methods can be combined with any of the above approaches in addition to any number of other algorithms to allow for non-linear transformations of the data. Due to the simplicity and performance of kernel methods, the research community has embraced them during the last four years. Thus, there have been hundreds if not thousands of papers published on face recognition and feature detection, mostly during the last decade when computers became powerful enough and algorithms sophisticated enough [3].

## III. SUPERVISED LEARNING

In supervised learning the data is represented as vectors. We usually have a set of vectors  $X = \{ x_i : i = 1..l \}$  representing the input to the problem and a set of vectors  $Y = \{ y_i : i = 1..l \}$  representing the associated desired output, where  $l$  is the number of vectors in the data set. We generally represent the data in pairs such as  $(x_i, y_i)$ . For example, if we want to train a machine to distinguish between pictures of apples and oranges, then  $X$  is the set of pictures of apples and oranges (each picture is reformatted as a vector), and  $Y$  is the set of associated labels for each picture such as  $y_1 = 0, y_2 = 1, y_3 = 1, y_4 = 0, y_5 = 1$ , and so forth, where 0 represents ‘orange’ and 1 represents ‘apple’. Given the training data, the goal of supervised learning is to learn a function  $M$  that maps each example input to the correct output, thus minimizing the sum  $\sum \| M(x_i) - y_i \|^2$ . Oftentimes the function  $M$  has a set of parameters  $\emptyset$  as input so that we have  $M(\emptyset, x_i) \mapsto y_i$ . In reality, most supervised learning algorithms work by a priori choosing a particular kind of function  $M$  and then automatically adjusting the parameters  $\emptyset$  so that the mentioned sum is minimized.

Classification is a particular kind of supervised learning in which  $y_i \in \{c_1, \dots, c_p\}$  where each  $c_j$  is a category and there are  $p$  categories [5]. Binary classification is the case where  $p = 2$ . The “apples and oranges” learning task is a binary classification task. An example of a non-binary classification task is character recognition from handwritten text. In this situation  $p = 62$  (26 lower case letters, 26 upper case letters, and 10 digits).

Moreover, the data is represented as  $n$ -dimensional vectors. Thus, each training and testing point  $x \in \mathbb{R}^n$  lives in an  $n$ -dimensional space.

### 3.1 PERCEPTRON

In the beginnings of supervised learning, perceptrons were the fundamental learning machines [5, 6, 7]. They are binary classification machines that use a weight  $w \in \mathbb{R}^n$  applied to the input to compute their output where  $n \in \mathbb{R}^n$  (see Figure 1.1).

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 1, January 2019

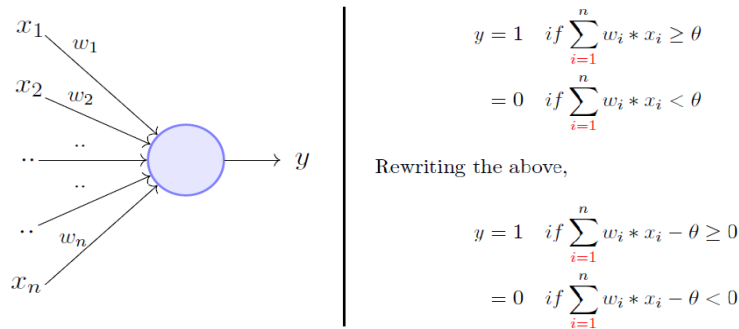


Figure 1.1 Simple Model For Perceptrons

The perceptron model is a more general computational model than McCulloch-Pitts neuron. It takes an input, aggregates it (weighted sum) and returns 1 only if the aggregated sum is more than some threshold else returns 0. Rewriting the threshold as shown above and making it a constant input with a variable weight, we would end up with something like the following in figure 1.2

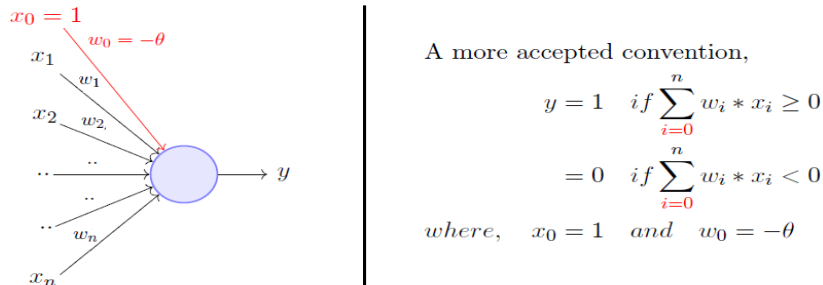


Figure 1.2 Model For Aggregation Sum

### 3.1.1 OR Function Using a Perceptrons

A single perceptron can only be used to implement linearly separable functions. It takes both real and boolean inputs and associates a set of weights to them, along with a bias. We learn the weights, on basis of some rules and we get the function. We defined a few conditions (the weighted sum has to be more than or equal to 0 when the output is 1) based on the OR function output for various sets of inputs, we solved for weights based on those conditions and we got a line that perfectly separates positive inputs from those of negative. Let's use a perceptron to learn an OR functions in figure 1.3.

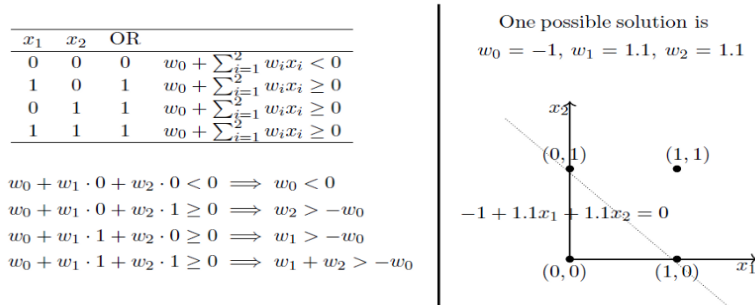


Figure 1.3 OR Function Using a Perceptrons

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 1, January 2019

### 3.1.2 PERCEPTRON LEARNING ALGORITHM

Our goal is to find the  $w$  vector that can perfectly classify positive inputs and negative inputs in our data. I will get straight to the algorithm. Here goes in figure 1.4.

```

while !convergence do
    Pick random  $x \in P \cup N$  ;
    if  $x \in P$  and  $w \cdot x < 0$  then
         $w = w + x$  ;
    end
    if  $x \in N$  and  $w \cdot x \geq 0$  then
         $w = w - x$  ;
    end
end
end

```

Figure 1.4 Perceptrons Learning Algorithm

We initialize  $w$  with some random vector. We then iterate over all the examples in the data,  $(P \cup N)$  both positive and negative examples. Now if an input  $x$  belongs to  $P$ , ideally what should the dot product  $w \cdot x$  be? I'd say greater than or equal to 0 because that's the only thing what our perceptron wants at the end of the day so let's give it that. And if  $x$  belongs to  $N$ , the dot product MUST be less than 0. So if you look at the if conditions in the while loop:

- Case 1: When  $x$  belongs to  $P$  and its dot product  $w \cdot x < 0$
- Case 2: When  $x$  belongs to  $N$  and its dot product  $w \cdot x \geq 0$

Only for these cases, we are updating our randomly initialized  $w$ . Otherwise, we don't touch  $w$  at all because Case 1 and Case 2 are violating the very rule of a perceptron. So we are adding  $x$  to  $w$  (ahem vector addition ahem) in Case 1 and subtracting  $x$  from  $w$  in Case 2.

### 3.2 NEURAL NETWORKS

For some problems the perceptron is not adequate since many learning tasks need to separate non-linearly separable data points. Firstly working with neural networks for supervised and unsupervised learning showed good results while used for such learning applications. MLP's uses feed forward and recurrent networks. Multilayer perceptron (MLP) properties include universal approximation of continuous nonlinear functions and include learning with input-output patterns and also involve advanced network architectures with multiple inputs and outputs [10]. A simple neural network model shown in figure 1.5 a).

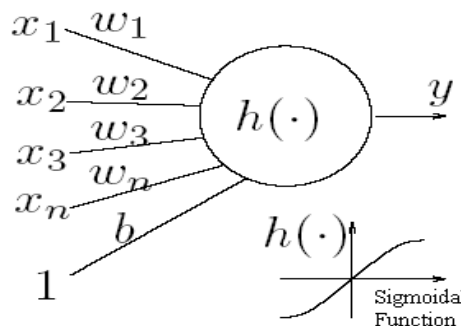


Figure 1.5: a) Simple Neural Network

These are simple visualizations just to have an overview as how neural network looks like in figure 1.5 b).

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 1, January 2019

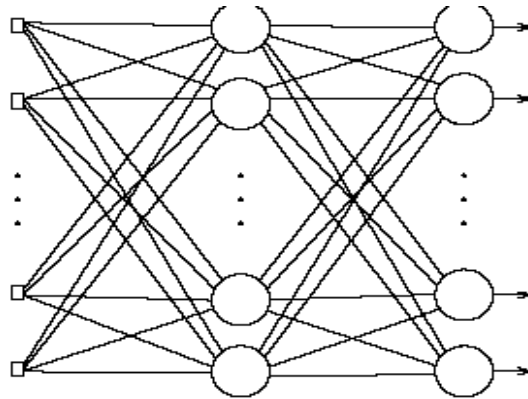


Figure 1.5: b) Multilayer Perceptron.

### 3.3. KERNEL TRICK

**KERNEL:** If data is linear, a separating hyper plane may be used to divide the data. However it is often the case that the data is far from linear and the datasets are inseparable. To allow for this kernels are used to non-linearly map the input data to a high-dimensional space. The new mapping is then linearly separable [1]. A very simple illustration of this is shown below in figure 7 [9] [11] [20].

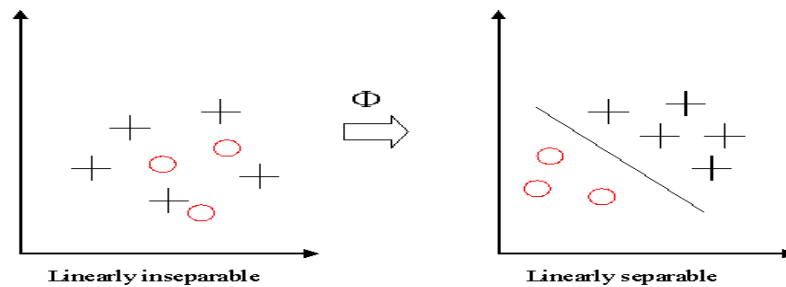


Figure 1.6: Linearly separating the data using the mapping  $\Phi$ .

**FEATURE SPACE:** Transforming the data into feature space makes it possible to define a similarity measure on the basis of the dot product. If the feature space is chosen suitably, pattern recognition can be easy [1]. Feature space representation generates clear understanding of concept shown in figure 1.7.

$$\langle x_1 \cdot x_2 \rangle \leftarrow K(x_1, x_2) = \langle \Phi(x_1) \cdot \Phi(x_2) \rangle$$

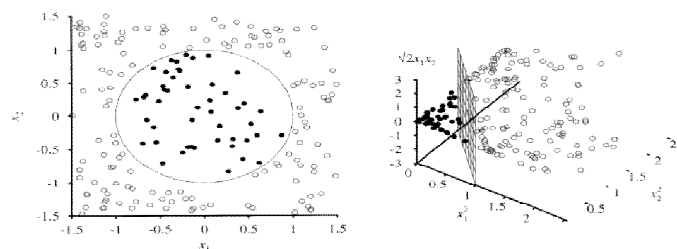


Figure 1.7: Feature Space Representation

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 1, January 2019

- Note the legend is not described as they are sample plotting to make understand the concepts involved.

Now getting back to the kernel trick, we see that when  $w, b$  is obtained the problem is solved for a simple linear scenario in which data is separated by a hyper plane. The Kernel trick allows SVM's to form nonlinear boundaries. Steps involved in kernel trick are given below [12] [24].

- The algorithm is expressed using only the inner products of data sets. This is also called as dual problem.
- Original data are passed through non-linear maps to form new data with respect to new dimensions by adding a pair wise product of some of the original data dimension to each data vector.
- Rather than an inner product on these new, larger vectors, and store in tables and later do a table lookup, we can represent a dot product of the data after doing non-linear mapping on them. This function is the kernel function. More on kernel functions is given below.

## IV. COMPARISONS

The following table summarizes the three supervised learning classifiers:

NAME OF CLASSIFIER	ADVANTAGES	DISADVANTAGES
Perceptrons	<ul style="list-style-type: none"> <li>Precise performance guarantees about future predictions</li> <li>Small number of examples required to train</li> </ul>	<ul style="list-style-type: none"> <li>Only work on linearly separable data</li> </ul>
Neural Networks	<ul style="list-style-type: none"> <li>Work on non-linearly separable data</li> <li>Work well in practice on difficult problems such as speech recognition and handwriting recognition</li> </ul>	<ul style="list-style-type: none"> <li>Lack of exact performance guarantees about future classification performance</li> <li>May require large number of training examples</li> <li>Not suitable for larger problems</li> <li>Hand-crafting features of input space to improve performance not automated and not always optimal</li> </ul>
Kernels	<ul style="list-style-type: none"> <li>Precise performance guarantees</li> <li>Small number of examples required to train</li> <li>May work on non-linearly separable data (also see disadvantages)</li> <li>Allow running several different linear algorithms in very high dimensional feature space</li> </ul>	<ul style="list-style-type: none"> <li>Sometimes the high dimensional feature space cannot linearly separate data</li> </ul>

Table 1: Advantages and Disadvantages of Perceptrons, Neural Networks and Kernel Methods

# International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Visit: [www.ijirset.com](http://www.ijirset.com)

Vol. 8, Issue 1, January 2019

## V. FACE RECOGNITION

The goal of this task is to identify one particular person from a set of photos of various people. We make references to several people from the database. Some pictures are included here for convenience. We use the feature space defined by the Gaussian kernel. In order to better visualize what happens to the data after remapping into feature space we apply principal components analysis (PCA). We use PCA in feature space as a visualization tool to map the data down to a two-dimensional space. We choose the two dimensions that capture most of the variance of the data so that we have a rough idea of the distances between the data points. For the face recognition task the dataset is composed of the first 200 images of the database. To visualize the “position” of these images in feature space, we take the dataset, map each image into the feature space determined by a Gaussian kernel with a width of 4000 and finally project it down onto the two dimensions that capture most of the variance given by PCA.

A kernel principal component analysis (PCA) was previously proposed as a nonlinear extension of a PCA. The basic idea is to first map the input space into a feature space via nonlinear mapping and then compute the principal components in that feature space. This paper adopts the kernel PCA as a mechanism for extracting facial features like smiles and length of hair etc. Through adopting a polynomial kernel, the principal components can be computed within the space spanned by high-order correlations of input pixels making up a facial image, thereby producing a good performance.

## VI. CONCLUSION

The proposed methods deserve special credit due to their high cost benefit, good results, and model simplicity (a simple MLP model) when compared with other algorithms. In conclusion, kernel methods not only are easy to train and accurate but they are also flexible to use for large complicated learning tasks. Because of their advantages, the research community has heavily explored their use during the last decade. For this algorithm, Face Recognition is an easy task as well as classifying people as having long or short hair. Smile Feature Identification is more difficult, but generally the algorithm performs slightly better at this task than Linear Discriminant Classifier.

## REFERENCES

- [1] Bartlett, Marian Stewart, Javier R. Movellan, and Terrence J. Sejnowski. Face Recognition by Independent Component Analysis. IEEE Transactions on Neural Networks. Vol 13. No 6. November 2002.
- [2] Boser, B. E., I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory. Pittsburgh: ACM Press. 1992.
- [3] Delac, K., M. Grgic, and S. Grgic. A Comparative Study of PCA, ICA, and LDA. Proceedings of the 5th EURASIP Conference focuses on Speech and Image Processing, Multimedia Communications and Services, EC-SIP-M 2005.
- [4] Duda, Richard O., Peter E. Hart, and David G. Stork. Pattern Classification. 2nd Ed. New York, NY: John Wiley & Sons, Inc. 2001.
- [5] Hassoun, Mohamad H. Fundamentals of Artificial Neural Networks. Cambridge, MA: The MIT Press. 1995.
- [6] Haykin, Simon. Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ: Prentice Hall. 1999.
- [7] Kirby, Michael. Geometric Data Analysis. New York, NY: John Wiley & Sons, Inc. 2001.
- [8] Kohonen, Teuvo. Self Organizing Maps. Heidelberg, Germany: Springer-Verlag. 1995.
- [9] Lu, Juwei, Kostantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Face Recognition Using LDA-Based Algorithms. IEEE Transactions on Neural Networks. Vol 14. No 1. January 2003.
- [10] David M Skapura, Building Neural Networks, ACM press, 1996.
- [11] Martinez, Aleix M. and Avinash C. Kak. PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 23. No 2. 2001.
- [12] Parzan, E. Extraction and detection problems and reproducing kernel Hilbert spaces. Journal of the Society for Industrial and Applied Mathematics. Series A. on control. 1962.
- [13] Shawe-Taylor, John, and Nello Cristianini. Kernel Methods for Pattern Analysis. Cambridge, UK: Cambridge University Press. 2004.
- [14] Schölkopf, Bernhard. Support Vector Learning. Munich: Oldenbourg Verlag. 1997.
- [15] Sutton, Richard S., and Andrew G. Barto. Reinforcement Learning: An Introduction. Cambridge, MA: The MIT Press. 1998