Kinship Verification Using Different Distance Metrics In NMD

1st Patnala Surya Vamsi
dept. Information Technology
Vardhaman College Of Engineering
Hyderabad, India
suryavamsipatnala123@gmail.com

2nd Muchinthala Sushmith
dept. Information Technology
Vardhaman College Of Engineering
Hyderabad, India
sushmithmuchinthala15@gmail.com

3rd Pothireddy Bhargava Reddy
dept. Information Technology
Vardhaman College Of Engineering
Hyderabad, India
pothireddyb9@gmail.com

4th Dr. Sagar Gujjunoori
dept. Information Technology
Vardhaman College Of Engineering
Hyderabad, India
sagar.g@vardhaman.org

***Abstract*—The computer vision researchers have become interested in the question of verifying parental and family relationships using the subjects facial features. There have been many proposed feature descriptors for facial images, the Neighbourhood Minimum Distance(NMD) descriptor is one of them. As majority of the images are affected by illumination and expression variations, the NMD desriptor is made to be more resilient to fluctuations in expression. It makes use of euclidean distance to find the most similar pixels of a parent and child image in a specified neighbourhood. Euclidean distance metric being a L2-norm is less robust compared to Manhattan distance metric which is a L1-norm. The main difference between both strategies is that Manhattan reduces the coefficient of the less significant feature to zero, eliminating certain features entirely. Upon testing the NMD descriptor using the Manhattan distance metric results demonstrated that it produced better results compared to that of Euclidean distance metric.**

***Index Terms*—Kinship Verification, Euclidean Distance, Manhattan Distance**

I. Introduction

Based on recent psychological research showing that facial likeness can be utilised for parental verification, kinship verification involves determining if two people are related based on their facial features. [2, 5, 12]. Since it can be used to detect relatives of a suspect, automatically create family albums, look for missing children, locate family members on social media, improve facial recognition performance, etc., kinship verification is a notable problem in computer vision.

In this paper,we compare the results produced when different distance metrics (Euclidean, Manhattan and Minkowski distance metrics) are used in Neighborhood Min Distance (NMD) descriptor to see the changes in the machine learning model's behavior and its impact on the model's accuracy. The main idea is to check if there is any other distance metric other than euclidean distance that helps in increasing the model's accuracy based on the classifier.

NMD works by computing the Euclidean distances between each pixel in the first image of the pair and its neighboring pixels in the second image, with respect to RGB values. The smallest distance in this neighbourhood is then added to the feature vector of the pair. From the above process, we replaced the Euclidean distance with Manhattan distance and then with Minkowski distance metrics to analyse the effect of different distance metrics on the result produced by the NMD descriptor

II. Existing Methodology

The facial images of one person should resemble those of their family members more so than those of other people. However, due to shape variations, comparing the two images pixel by pixel to conduct kinship verification will lead to substandard results.

In NMD descriptor proposed by Abdellah et al, every pixel in the parent's image must have a very similar pixel (in terms of RGB colour) in the child's image., The NMD descriptor searches the child's image for the most similar pixel to the current pixel in the image of the parent in terms of the euclidean distance in the RGB color space. Furthermore, the search is restricted to the neighborhood of the child's pixel that is located in the same position as the parent's.The Algorithm contains four nested loops, the two outer loops are used to run through all the pixels of the parent image that have a neighborhood of the desired size, and the two inner loops are used to look in the child's image for the closest pixel (in terms of the RGB euclidean distance) to the parent's pixel pointed by the current iteration of the two outer loops, the search for the closest pixel is restricted to a local area (a neighborhood) defined by its size N × N. The below formula is used to find the Euclidean distance in the algorithm.

$$d^2([R_1G_1B_1],[R_2G_2B_2]) = (R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2$$

The NMD descriptor produces a feature vector which is computed using Random Forest classifier to classify the pairs of images.

III. PROPOSED FRAMEWORK

Distance metrics are an important part of machine learning algorithms. Different distance metrics are used in different learning algorithms based on the requirement.Distances can be used to find the similarity of data points or to know the data pattern, to make any data based decision.An effective metric improves the efficiency of the machine learning model.So in order to find the effective distance metric, different 2D distances used in machine learning are considered for classifying the kinship problem and an analysis is made based on the results. The distance metrics considered are Manhattan distance, Minkowski distance with different parameters etc.

A. Manhattan Distance

The Manhattan distance is also called as Taxicab distance or City Block Distance. It is a distance measure that is calculated by taking the sum of distances between x and y coordinates.It calculates the distance between two real valued vectors.The Manhattan distance is also known as Manhattan length. In other words, it is the distance between two points measured along axes at right angles.It is perhaps more useful to vectors that describe objects on a uniform grid, like a chessboard or city blocks. The taxicab name for the measure refers to the intuition for what the measure calculates: the shortest path that a taxicab would take between city blocks (coordinates on the grid). Manhattan distance is calculated as the sum of the absolute differences between the two vectors.

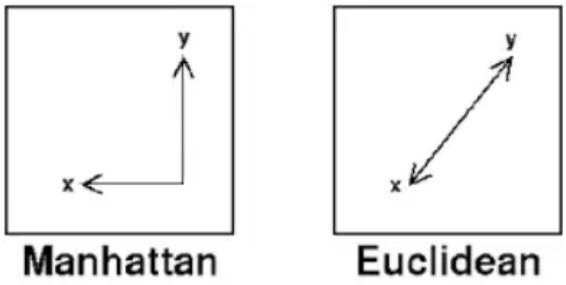
$$Distance(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

From the above formula it is evident that it is a distance metric between two points in a N dimensional vector space. It is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes. In simple terms, it is the sum of absolute difference between the measures in all dimensions of two points.

$\sum_{0}^{n} |x_i - y_i| = |x_1 - y_1| + |x_2 - y_2| + |x_3 - y_3| + ... + |x_n - y_n|$
.

This is the general formula for Manhattan distance where x1,x2,...,xn are the coordinates in geometry plane and y1,y2,y3,,,,,yn are y coordinates in the geometry plane. Manhattan distance works very well for high-dimensional datasets. As it does not take any squares, it does not amplify the differences between any of the features. It also does not ignore any features.

The x,y coordinates in the data set that the Manhattan distance is calculated are the R,G,B components of the parent and child images. By using this Manhattan distance, the similarity between the pixels is calculated. This process is similar to finding the euclidean distance between the pixels but the only difference is using Manhattan distance instead of Euclidean distance. Manhattan Distance is preferred over the Euclidean distance metric as the dimension of the data increases. This occurs due to something known as the 'curse of dimensionality'.



B. Minkowski Distance

Minkowski distance calculates the distance between two real-valued vectors. This distance metric is a generalization of the Euclidean and Manhattan distance metrics.It combines Euclidean and Manhattan distance measures and adds a parameter, called the "order" or "p", that allows different distance measures to be calculated.In machine learning, the distance metric calculated from the Minkowski equation is applied to determine the similarity of size.It is also renowned as the p-norm vector, which represents the order of the norm. It allows the addition of the parameter p that enables the measurement of different distances that are to be calculated.

This is the general formula for Minkowski distance

$$D(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

When the value of P becomes 1, it is called Manhattan distance. When P takes the value of 2, it becomes Euclidean distance. Intermediate values provide a controlled balance between the two measures. It is common to use Minkowski distance when implementing a machine learning algorithm that uses distance measures as it gives control over the type of distance measure used for real-valued vectors via a hyperparameter "p" that can be tuned.

$$d(x, y) = \sqrt[p]{(x_1 - y_1)^p \ + \ (x_2 - y_2)^p \ + \ ... \ + \ (x_n - y_n)^p}$$

The value of P in Minkowski Distance is set based on experimentation. The usual value of P is either 2, 3 or 4 for most problems. The exact value depends on the applications. Having the optimal value of P is critical to achieve the correct results based on the objective of the application.

The P values used in finding the similarity of the parent and child image pixels are 3 and 4. As P=1 and P=2 are already calculated i.e. P=1 for Manhattan distance and P=2 for Euclidean distance, it is obvious that the distances should

be calculated with higher P values. Knowing that P values can also be intermediate values, based on the obtained results P value will be tuned between 3 and 4 if the accuracy increases and varies accordingly with the value of P

When P=3, the Minkowski distance is as follows

$$d(x,y) = \sqrt[3]{(x_1 - y_1)^3 + (x_2 - y_2)^3 + ... + (x_n - y_n)^3}$$

When P=4, the Minkowski distance is as follows

$$d(x,y) = \sqrt[4]{(x_1 - y_1)^4 + (x_2 - y_2)^4 + ... + (x_n - y_n)^4}$$

Where x and y coordinates are the R,G,B components of the parent and child pixels.

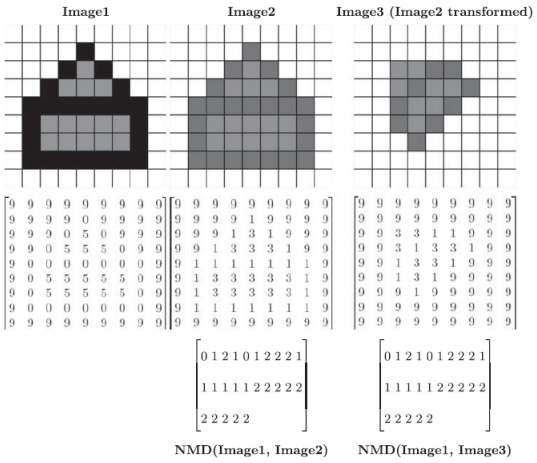
Based on the observations after calculating distances from both P values, intermediate p values are also tested to check for change in accuracy of the machine learning model. Also on the basis of classifier used, the distance metric is also considered to get the optimal results. The classifers used in this machine learning problem is SVM and Random Forest. There might be differences between the accuracy given by these classfiers i.e, one might have increased accuracy while a declination can be seen in the later classfier and vice versa.

While there are also other distance metrics available in machine learning that can be used to solve this kinship problem, they are not very effective for the classifiers considered in this model.

C. Classification

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups. For each kinship image-pair, a single NMD feature-vector is computed, then, the pair's feature vector will be passed as input to a binary classifier that will output the kinship class (positive/negative). Several state-of-the-art classifiers were used to test the NMD descriptor, including KNN, SVM, Random Forest, and AdaBoost, however, the best results were obtained with the Random Forest classifier (aka Random Decision Forest).

It is an Ensemble Learning approach proposed by Ho [9] as a solution to the lack of generalization accuracy (aka overfitting) in Decision Trees. The idea of the Random Forest classifier is to train a bunch of decision trees on copies of the original dataset created through the process of bagging (see Definition 1). At each step of the training, only a randomly selected subset of the variables (columns or features) of the data is used to create the tree nodes. After the training of the forest, the mode classification (most voted decision) of all the trees is used as the output of this classifier. Definition 1 (Bagging) Let X = x1, x2, ..., xN be the training dataset containing N samples associated with labels Y = y1, y2, ..., yN , bagging (from bootstrapping-aggregating) is the process of randomly-selecting (with repetition) N samples from (X, Y) to form a new dataset (Xt, Yt), this process should be repeated T times, where T is the number of decision trees in the forest. It's equation is given by (5).



D. Dataset

In our experiments, we used the KinFaceW datasets(Kinface-I, Kinface-II), these datasets are free to download from the official website. The KinFaceW dataset has two sets i.e., KinFaceW-I and KinFaceW-II, Face images are collected from the internet including some public figures parent's or children's face images. The difference of KinFaceW-I and KinFaceW-II is that face images with a kin relation were acquired from different photos in KinFaceW-I and the same photo in KinFaceW-II in most cases. The KinfaceW datasets, In the first KinFaceW-I dataset, there are 156, 134, 116, and 127 pairs of kinship images for these four relations. For the KinFaceW-II dataset, each relation contains 250 pairs of kinship images. For ease of use, we manually labeled the coordinates of the eyes position of each face image, and aligned and cropped facial region into 64x64 to remove background. The following figures show some cropped face images from KinFaceW-I and KinFaceW-II datasets, respectively.

IV. Experiments and Results

Our experiments usually consists of 2 stages i.e., First stage is where we use manhattan distance metric with NMD

descriptor to extract feature vector. Second stage innvolves providing the feature vector as an input to classifiers. A few other researchers experiments include using Deep-Learning based models and using external dataset instead of local storage which uses similar stages for classificaiton. From the datasets the five-fold cross-validation accuracy is computed to compare the performance of our algorithm with some other state-of-the-art approaches. It is critical for a fair comparison to evaluate using the same experimental protocol as other approaches, This also helps us in using the identical train-test split ratio.

A. Results and Analysis

Here we will analyse all the results obtained from the experiments conducted. In the tables given here the: FS, FD, MS and MD refer to the kinship relations (or subsets): father-son, father-daughter, mother-son and mother-daughter respectively.

To test the effectiveness of the different distance metrics for kinship verification, four different distance measurements were used on the KinFaceW datasets using five fold cross-validation, And we have tested all possible values of the neighborhood size (odd numbers from 3 to 63) using the four classifiers mentioned earlier.

To evaluate the efficiency of all available methods, the selected datasets consits of images of parents and children are available for four different types of relationships: father-daughter (F-D), father-son (F-S), mother-son (M-S), and mother-daughter (M-D).

Tables 1 and 2 shows the best results using all the distance metrics obtained in KinFaceW-I, KinfaceW-II respectively. The best results on all datasets were obtained using the SVM(Support Vector Machine) classifier.

Approach	FS	FD	MS	MD	Mean
Eucledian	67.3	59.7	63.8	72.4	65.8
Minkowski for P=3	57.4	57.1	54.7	56	56.3
Minkowski for P=4	61.5	55.6	57.3	56.3	57.6
Manhattan	**68.9**	**62.3**	**68**	**83.5**	**70.6**

TABLE I
COMPARISION OF DIFFERENT DISTANCE METRICS ON KINFACEW-I DATASET

Approach	FS	FD	MS	MD	Mean
Eucledian	79	74.6	79.2	84.4	79.3
Minkowski for P=3	63.8	63.6	60.8	60.4	62.1
Minkowski for P=4	66.4	65.4	69.6	64.8	66.5
Manhattan	**83.6**	**79.4**	**87.8**	**90.2**	**85.2**

TABLE II
COMPARISION OF DIFFERENT DISTANCE METRICS ON KINFACEW-II DATASET

Tables 3 and 4 shows the best results using all the distance metrics obtained in KinFaceW-I, KinfaceW-II respectively. The best results on all datasets were obtained using the Random Forest classifier.

Approach	FS	FD	MS	MD	Mean
Eucledian	72.1	69	71.1	85.1	74.3
Minkowski for P=3	66	66	66.8	72.3	67.7
Minkowski for P=4	71.8	70	72.3	85.5	74.9
Manhattan	**71.5**	**69**	**73.2**	**84.7**	**74.6**

TABLE III
COMPARISION OF DIFFERENT DISTANCE METRICS ON KINFACEW-I DATASET

Approach	FS	FD	MS	MD	Mean
Eucledian	87.2	82.2	90	91.6	87.7
Minkowski for P=3	73.8	73	73.8	75.8	74.1
Minkowski for P=4	87.2	82.4	89.6	91.2	87.6
Manhattan	**87.2**	**80.8**	**90**	**91.4**	**87.3**

TABLE IV
COMPARISION OF DIFFERENT DISTANCE METRICS ON KINFACEW-II DATASET

1) Experiments and Discussion: Using SVM(Support Vector Machine) classifier We observed that the results obtained using different distance calculation approaches which varies the feature vector greatly. And we observed that in every kin-relation Manhattan distance metric displays similar or higher in most cases. This is the case in both the kinfaceW datasets(KinfaceW-I and KinfaceW-II).

But using Random Forest classifier we have observed that the results obtained using Manhattan distance metric are similar to Eucledian and Minkowski (for p=4) in terms of mean accuracies. So the 3 distance metric gave comparable results in random classifier which needs to be addressed in future experiments to decide which metric is the best.

2) Effect of the classifier: An efficient classifier must be utilised to produce the final result (Kin or No-Kin). We compared two well-known classifiers to demonstrate their efficacy because of this. To compare the effects of our approach we used SVM(Support Vector Machine) and Random Forest. The SVM also achieves notable accuracy while using less processing power. But, Random forest have produced highest accuracy in comparison to SVM. Though it's accuracy varies with the depth value, no.of decision trees specified.And a few approaches(different distance metric) had varying accuracies depending on kin-relationship and dataset.

CONCLUSION

As we have compared different distance measuring metrics for identifying kin relationship between pairs of facial images, the purpose of using the most effective distance metric is to provide more robust feature vector of facial parts. The findings of the experimental study demonstrate the efficiency of the Manhattan Distance metric in terms of accuracy and execution time is better than other metrics. Some issues concerning the minkowski metric where it's accuracy in random forest is parallel to manhattan, needed to be addressed in future research, including the non-robustness of the descriptor to the variations in illumination, and the single direction comparison (parent to child) of images.

References

[1]Alirezazadeh P, Fathi A, Abdali-Mohammadi F (2015) A genetic algorithm-based feature selection for kinship verification. IEEE Signal Process Lett 22(12):2459–2463

[2] Alvergne A, Oda R, Faurie C, Matsumoto-Oda A, Durand V, Raymond M (2009) Cross-cultural perceptions of facial resemblance between kin. J Vis 9(6):23–23 [3] Cao Z, Yin Q, Tang X, Sun J (2010) Face recognition with learning-based descriptor. In: 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, pp 2707–2714

[4] Chen X, An L, Yang S, Wu W (2017) Kinship verification in multi-linear coherent spaces. Multimed Tools Appl 76(3):4105–4122 [5] DeBruine LM, Smith FG, Jones BC, Roberts SC, Petrie M, Spector TD (2009) Kin recognition signals in adult faces. Vision Research 49(1):38–43

[6] Fang R, Tang KD, Snavely N, Chen T (2010) Towards computational models of kinship verification. In: 17th IEEE international conference on image processing (ICIP). IEEE, pp 1577–1580

[7]Fang Y, Chen YYS,Wang H, Shu C (2016) Sparse similarity metric learning for kinship verification. In: Visual communications and image processing (VCIP), 2016. IEEE, pp 1–4

[8] Felzenszwalb PF, Huttenlocher DP (2005) Pictorial structures for object recognition. Int J Comput Vis 61(1):55–79

[9] Ho TK (1995) Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, vol 1. IEEE, pp 278–282

[10] Hu J, Lu J, Yuan J, Tan YP (2014) Large margin multimetric learning for face and kinship verification in the wild. In: Asian conference on computer vision. Springer, pp 252–267

[11]Kaihao Z, Yongzhen H, Chunfeng S, Hong W, Liang W (2015) Kinship verification with deep convolutional neural networks. In: Proceedings of the British machine vision conference (BMVC). BMVA Press, pp 148.1–148.12. https://doi.org/10.5244/C.29.148

[12] Yan H (2017) Kinship verification using neighborhood repulsed correlation metric learning. Image Vis Comput 60:91–97

[13] Yan H, Lu J, DengW, Zhou X (2014) Discriminative multimetric learning for kinship verification. IEEE Trans Inform Forensics Secur 9(7):1169–1178

[14] Yan H, Zhou X, Ge Y (2015) Neighborhood repulsed correlation metric learning for kinship verification. In: Visual communications and image processing (VCIP), 2015. IEEE, pp 1–4

[15]Yang Y, Wu Q (2017) A novel kinship verification method based on deep transfer learning and feature nonlinear mapping. In: International conference on artificial intelligence and engineering applications

[16] Zhao YG, Song Z, Zheng F, Shao L (2018) Learning a multiple kernel similarity metric for kinship verification. Inf Sci 430:247–260

[17] Zhou X, Hu J, Lu J, Shang Y, Guan Y (2011) Kinship verification from facial images under uncontrolled conditions. In: Proceedings of the 19th ACM international conference on multimedia. ACM, pp 953–956

[18] Zhou X, Jin K, Xu M, Guo G (2019) Learning deep compact similarity metric for kinship verification from face images. Information Fusion 48:84–94

[19] Zhou X, Lu J, Hu J, Shang Y (2012) Gabor-based gradient orientation pyramid for kinship verification under uncontrolled environments. In: Proceedings of the 20th ACM international conference on multimedia. ACM, pp 725–728

[20] Zhou X, Shang Y, Yan H, Guo G (2016) Ensemble similarity learning for kinship verification from facial images in the wild. Information Fusion 32:40–48

[21] Zhou X, Yan H, Shang Y (2016) Kinship verification from facial images by scalable similarity fusion. Neurocomputing 197:136–142

[22] Lu J, Zhou X, Tan YP, Shang Y, Zhou J (2014) Neighborhood repulsed metric learning for kinship verification. IEEE Trans Pattern Anal Machine Intell 36(2):331–345

[23]Mahpod S, Keller Y (2018) Kinship verification using multiview hybrid distance learning. Comput Vis Image Underst 167:28–36

[24] Puthenputhussery A, Liu Q, Liu C (2016) Sift flow based genetic fisher vector feature for kinship verification. In: 2016 IEEE international conference on image processing (ICIP). IEEE, pp 2921–2925

[25]Qin X, Tan X, Chen S (2016) Mixed bi-subject kinship verification via multi-view multi-task learning. Neurocomputing 214:350–357