Poincaré Metric in Algorithms for Data Mining Tools

Alenka Trpin[1], Biljana Mileva Boshkoska[1,2](✉), and Pavle Boškoski[1,2]

[1] Faculty of Information Studies in Novo Mesto,
Ljubljanska cesta 31 a, Novo Mesto, Slovenia
biljana.mileva@fis.unm.si
[2] Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia

Abstract. Today we cannot imagine life without computers. The massive use of the information communication technologies has produced large amounts of data that are difficult to interpret and use. With data mining tools and machine learning methods, large data sets can be processed and used for prediction and classification. This paper employees the well known classification algorithm the *k* nearest neighbour and it modified use the Poincaré measurment distance instead of traditional Euclidean distance. The reason is that in different industries (economy, health, military ...) it increasingly uses and stores databases of various images or photographs. When recognizing the similarity between two photographs, it is important that the algorithm recognizes certain patterns. Recognition is based on metrics. For this purposes an algorithm based on Poincaré metric is tested on a data set of photos. A comparison was made on algorithm based on Euclidean metric.

Keywords: Poincaré metric · Weka · K nearest neighbors

1 Introduction

Face recognition is an important role in our lives. Even a baby can recognize her mother's face. Adults recognize friends by their faces, and our perception about how we feel about someone can be seen by our face characteristics (sad, angry, happy ...). Face recognition can be exploited in numerous areas some if which are:

- security (control buildings, airports, ...),
- network security (email authentication on multimedia),
- identity verification (electoral registration, banking, ...),
- criminal justice systems [6].

Face recognition is an automatic way to identify persons based on their faces intrinsic characteristics. It usually involves: detecting the facial area, normalizing the detected faces, extracting facial features from appearance or facial geometry,

S. Kozielski et al. (Eds.): BDAS 2019, CCIS 1018, pp. 195–203, 2019.
https://doi.org/10.1007/978-3-030-19093-4_15

and finally classifying facial images based on the extracted features [9]. Biometric technologies include identification based on physiological characteristics (such as face, fingerprints, finger geometry, hand geometry, palm, iris, ear, voice, ...) and behavioral traits (gait, signature, keystroke dynamic) [7].

In this paper we focused on the well known classification algorithm k nearest neighbor, which we modified such that we changed the Euclidean distance metric with the Poincaré distance metric and tested it on a database of grey images of faces. The remainder of this paper is organized as follows. In Sect. 2 we define the Poincaré metric, and in Sect. 3 we discuss the used kNN classification method and its modification. The preparation of data and experimental results are reported in Sect. 4. Conclusions are given in Sect. 5.

2 Poincaré Metric

Poincaré's metric is useful in hyperbolic geometry. Unlike the Euclidean geometry, where it holds that through a given point we can draw exactly one parallel to a given line, in hyperbolic geometry holds that through a given point there are infinite parallels to a given line [13]. In this research we used Poincaré's disk model, which is in n-dimensional space defined as $B^n = \{x \in \mathbb{R}, \|x\| < 1\}$.

Let P be an arbitrary point in the unit disk and the v vector defined at this point.

Let the γ be a continuous path that maps the interval $[0, 1]$ to the disk D. The definition of Poincaré length of path is:

$$\ell_\rho(\gamma) = \int_0^1 |\dot{\gamma}(t)|_{\gamma(t)}\, dt = \int_0^1 \frac{1}{1 - |\gamma(t)|^2} \cdot \|\dot{\gamma}(t)\|\, dt. \tag{1}$$

Then Poincaré metrics between two different points $P, Q \in D$ is defined as

$$d_\rho(P, Q) = \inf\{\ell_\rho(\gamma) : \gamma \in \mathcal{C}_D(P, Q)\}. \tag{2}$$

The path, which length is defined by the infimum, is called geodesic. This is the shortest distance between two points. If the points P, Q are collinear with the starting point, the shortest path through these two points is a line. In other cases, they are circular arcs in standard Euclidean geometry. The circle on which such a circular arc is located, or a geodesic, can be precisely determined. Using the analytical method we can calculate the radius and center of this circle because we know three points P, Q and $\frac{1}{\overline{P}}$. The third point $\frac{1}{\overline{P}}$ is the point P, which is reflection over the edge of the disk.

A hyperbolic disk with the center a and a radius r in a complex space is denoted by $D_\rho(a, r) = \{z \in D; d_\rho(a, z) < r\}$. That is a set of all points whose distance from the center is less than r, which is exactly an Euclidean disk whose closure is contained in D [8].

If P is the point that lies in the unit disk, then Poincaré distance between starting point and point P is equal to

$$d_\rho(0, P) = \frac{1}{2} \cdot \ln\left(\frac{1 + |P|}{1 - |P|}\right). \tag{3}$$

Poincaré distance between two different points $P, Q \in D$ on a single disc is defined as

$$d_\rho(P,Q) = \frac{1}{2} \ln \left(\frac{1 + \left\| \frac{P-Q}{1-\overline{P}Q} \right\|}{1 - \left\| \frac{P-Q}{1-\overline{P}Q} \right\|} \right). \tag{4}$$

3 Classification Method

In this paper we used a modified k nearest neighbor (kNN) classifier that represents each example as a data point in a d-dimensional space, where d is the number of attributes. KNN involves a two-step process:

- an inductive step for constructing a classification model from data, and
- a deductive step for applying the model to test examples [12].

In the inductive step, the kNN classifier uses distance-based comparisons that intrinsically assign equal weight to each attribute. They therefore can suffer from poor accuracy when given noisy or irrelevant attributes. The method, however, has been modified to incorporate attribute weighting and the pruning of noisy data tuples. The choice of a distance metric can be critical. Usually the Eucledian distance, the Manhattan (city block) distance, or Minkowski metrics are used [3]. In this paper we use instead the Poincare distance as given with Eqs. 3–4.

In kNN, k is a natural number that tells how many adjacent attributes affect the classification of a given instance. Choosing the appropriate k affects the classification of a given instance into a class. If k is too small, the results are too general. If k is small, there are more boundaries and fragments, so the algorithm is more sensitive to noise. If it is too large, local features do not get to the expression, it follows that the classifier classifies the experimental attribute incorrectly. The larger k, leads to more inaccurate and time-consuming models.

4 Data Processing and Results

Face recognition is a specific and hard case of object recognition. The difficulty of this problem stems from the fact that in their most common form (the frontal view) faces appear to be roughly alike and the differences between them are quite subtle. Furthermore, the human face is not a unique, rigid object [6]. A number of conditions are affected by the perception of success:

- expression on the face (cheerful, sad, angry),
- face cover (wearing a scarf, sunglasses ...),
- face view (different profiles),
- camera features (lens, lighting),
- facial features (mustache, beard, glasses),
- the size of the face (the face closer to the camera is bigger),
- face illumination.

We obtained 640 images from the UCI Machine Learning Repository, which include faces in various positions and conditions [4]. The images are given in gray scale and all share the same size and format. Images differ in the facial positions, facial expressions, and the appearances (long or short hair, beard, sunglasses). In the pre-processing step we converted each image into a format that is suitable for processing in WEKA where data are presented as vectors with numerical values. Hence we converted the images into jpg format and then we created an arff file with two attributes: image name and class (Fig. 1).

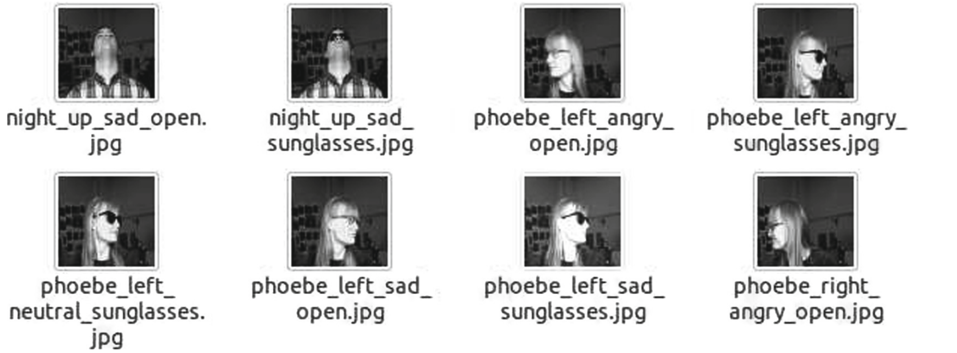
night_up_sad_open.jpg
night_up_sad_sunglasses.jpg
phoebe_left_angry_open.jpg
phoebe_left_angry_sunglasses.jpg
phoebe_left_neutral_sunglasses.jpg
phoebe_left_sad_open.jpg
phoebe_left_sad_sunglasses.jpg
phoebe_right_angry_open.jpg

Fig. 1. Photos in different positions

Next we processed the data with WEKA using different photo filters that were suitable for black and white photographs: Binary patterns pyramid filter, Edge histogram filter, FCTH filter, Gabor filter, JPEG coefficient filter and PHOG filter.

Local binary patterns pyramid (BPP) filter creates 3 × 3 pixel blocks of an image. Each pixel in this block is thresholded by its center pixel value which is a binary number. Next it compares the values of the point and its neighbors: if the value of the point is greater than the value of its neighbor, the algorithm records 0, otherwise it records 1. This creates an eight-digit binary number. Next the histogram of these numbers is created and normalized. The result is a 256-bit vector used for classification [10].

Edge histogram filter (MPEG) is used to compare images at the edges. It describes the spatial distribution of edge parts, non edge parts and four directional edge parts. The image is partitioned in 16 sub-regions. A histogram of five bins (colors) is created for each sub-region. The image is divided into smaller regions, where the edges are categorized. Each bin is normalized and quantized (reduces the range of values). To measure the similarity of two edge histograms, the sum of the absolute differences of individual bins is used [11].

The FCTH filter (fuzzy colour and texture histogram) is limited to 72 bytes per image, so it is suitable for large databases. It uses three channels and forms ten bins. Each part represents a certain color. This filter is not sensitive to noise and deformation [1].

Gabor filter uses Gabor waves to remove attributes from the image. Each wave captures energy at a given frequency and in a certain direction. It provides a local description of the frequency, thereby capturing the local characteristics of the signal. This filter is useful for surface analysis [15].

A JPEG (Joint Photographic Experts Group) coefficient acquires attributes from a sequence of quantized coefficients that a person can not detect when converting to a JPEG format. The filter acquires coefficients with discrete cosine transformation. This is a well-established standard for compression of color and gray-scale images for the purposes of storage and transmission [2].

PHOG filter encodes information on the orientation of the gradient intensity to the image. It is based on a local image set and is effective in spatial distribution of the edge. It divides the image into a net, then compensates for the histogram of the gradient for each network [5].

The result of the processing is an .arff file where the first attribute represents the name of the photo, followed by the numeric attributes, as represented in Fig. 2.

In the processing phase we set up a test and learning sets and used 10 - fold transversal validation [14].

During preprocessing, we removed the attribute image name, selected the kNN classifier, implemented the Poincaré distance as a metric in Java, set up the metric for usage in WEKA. The number of neighbors were selected inductively. To find the best k with k - nearest neighbor, we selected the Linear NN search algorithm.

Table 1 shows the performance of the classifier in percentages in terms of their accuracies, which are rounded up to two decimal places. The variable k is the number of closest neighbors that were searched with the Linear NN search algorithm. In Table 1, the results differ both depending on the filters that were used and on the set k number in the kNN. In Table 1 PM stands for Poincaré metric and EM for Eeuclidean based kNN while the variable k is the number of closest neighbors. The required execution time of the modified kNN with PM was higher than the EM algorithm, since the calculation of the Poincaré distance requires calculation of three different Euclidean distances as given in Eq. 4. However, the complexity of both algorithms is the same.

Table 1. Results

Metrics	PM			EM		
Filters	$k = 1$	$k = 5$	$k = 10$	$k = 1$	$k = 5$	$k = 10$
MPEG	**98,44%**	**97,97%**	**97,97%**	**98,59%**	97,81%	**97,97%**
BPP	**98,44%**	97,81%	97,66%	95,47%	93,59%	91,87%
Gabor	**43,44%**	38,91%	34,68%	**66,41%**	63,91%	61,25%
JPEG coefficient	**95,31%**	94,69%	94,84%	**95,63%**	95,31%	94,69%
PHOG	97,97%	**97,97%**	96,72%	98,44%	**97,97%**	97,81%

```


@relation FIT

....
@attribute 'Spatial Pyramid of Local Binary Pattern 748' numeric
@attribute 'Spatial Pyramid of Local Binary Pattern 749' numeric
@attribute 'Spatial Pyramid of Local Binary Pattern 750' numeric real
@attribute 'Spatial Pyramid of Local Binary Pattern 751' numeric real @attribute 'Spatial Pyramid of Local Binary Pattern 752' numeric
@attribute 'Spatial Pyramid of Local Binary Pattern 753' numeric
@attribute 'Spatial Pyramid of Local Binary Pattern 754' numeric
@attribute 'Spatial Pyramid of Local Binary Pattern 755' numeric
@attribute class {ani, at, boland, bpm, ch, cheyer, chon, danieln, glickman, karyadi, ...}

@data
an2i_left_angry_open.jpg,5,8,5,0,9,1,0,1,15,1,0,0,1,,...
an2i_left_angry_sunglasses.jpg,5,8,6,0,8,1,1,0,15,1,0,...
an2i_left_happy_open.jpg,6,9,6,1,10,1,1,0,15,1,0,0,...
an2i_left_happy_sunglasses.jpg,6,9,6,1,9,1,1,...
an2i_left_neutral_open.jpg,6,9,6,1,10,1,1,1,15,...
an2i_left_neutral_sunglasses.jpg,6,9,6,1,9,1,0,1,...
an2i_left_sad_open.jpg,5,8,6,0,9,1,1,0,15,1,...
```

Fig. 2. Processed data

As shown in Table 1, the accuracy of the algorithms depends on the filter selection. The worst classification in both algorithms was obtained with the Gabor's filter, and the best results were obtained with the Edge histogram filter. The PM algorithm yielded the best result with the Edge histogram and Binary patterns pyramid (BPP) filters, and the EM algorithm with the Edge histogram and PHOG filter. For $k = 1$, the Binary patterns pyramid filter produced better results than the PM algorithm (by almost 3%), and the PHOG filter was more successful EM algorithm (approximately 0,5%). There are no large deviations between the algorithms, except for the Gabor's filter, where by increasing the number of neighbors the error is doubled by almost twice. Using Gabor's filter lead to ineffective classification results for the PM and EM with kNN.

Figures 3 and 4 show the ROC curves when using MPEG and PHOG filters, for both EM and PM algorithms, and for three values of $k = 1, 5, 10$. On x axis

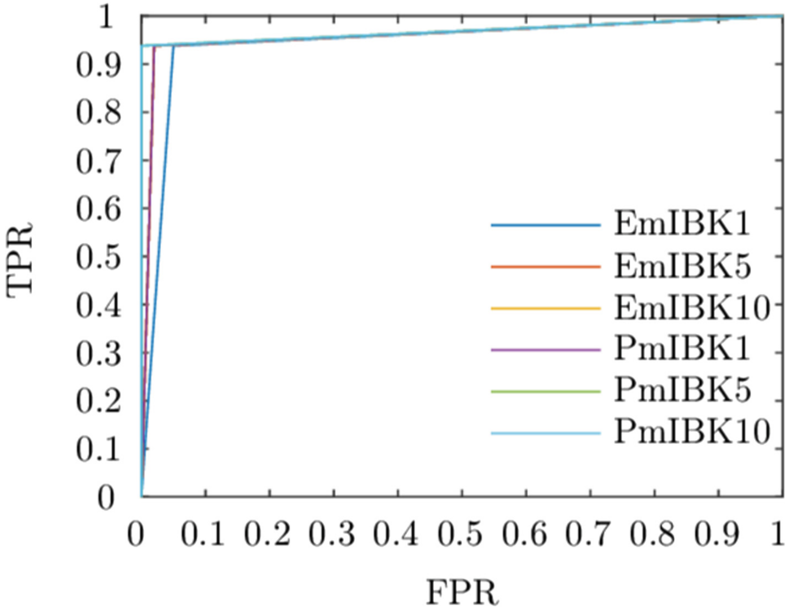

Fig. 3. ROC curve for MPEG filter for $k \in \{1, 2, 3\}$ and for PM and EM

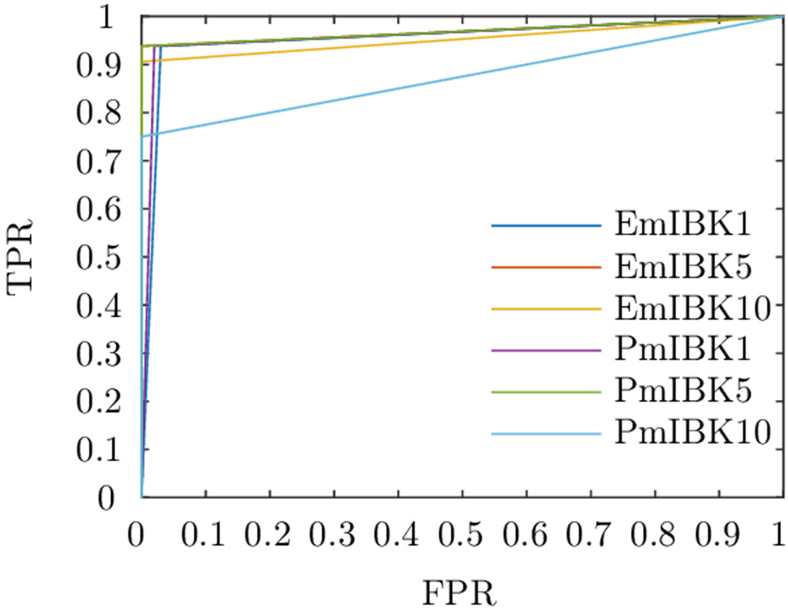

Fig. 4. ROC curve for PHOG filter for $k \in \{1, 2, 3\}$ and for PM and EM

Table 2. Comparison between the two algorithms

Filter	BPP		Edge histogram	
Variables	PM	EM	PM	EM
RMSE	**0,0395**	0,0666	0,0395	**0,0376**
MAE	**0,0047**	0,0076	0,0047	**0,0045**
RRSE	**18,1413**%	30,5454%	18,1413%	**17,2431**%
RAE	**4,9448**%	7,9647%	4,9448%	**4,7859**%

of the ROC curves we show the values of the False Positive Rate (FPR) and on the y axis we show the values of the True Positive Rate (TPR).

Table 2 shows a comparison between the two classification models for $k = 1$ and for two different filters, the BPP and the Edge histogram, for which both models perform successfully. The following statistical variables are compared: the root mean squared error (RMSE), the mean absolute error (MAE), the root relative squared error (RRSE) and the relative absolute error (RAE). With the BPP filter, the PM classification model has all the statistical values smaller than the EM classification model, therefore it follows that the PM classification model is more successful than the EM classification model. Conversely, the results of the Edge histogram filter show that all the statistical values of the EM classification model are smaller than those for PM classification model, hence it is considered as the better one.

5 Discussion and Conclusion

The paper is a first attempt of the authors to use the Poincaré metrics for classification. In particular we modified the well known kNN so that instead of the usual Euclidean metric for calculation of the distances between two points we applied the Poincaré metrics that is used for calculation of distances between points in hyperbolic spaces. We tested the modified kNN on a set of gray scale images that we obtained on the UCI Machine Learning Repository. The result was that the existing classification machine learning algorithm based on the Euclidean metric can be modified so that the Poincaré metric is used.

We have found that the choice of the filter that is used in the pre-processing phase influence the classification efficiency. In particular, the PM algorithm gave the best result with the Edge histogram filters and the Binary patterns pyramid. The EM algorithm provided the best results when used with the Edge histogram and PHOG filters. The comparison of both classification methods showed that the algorithm PM was better with the BPP filter (for $k = 1$ the classification was 98.44%). The same result was obtained with the Edge histogram filter, which was better than the BPP with the increase in the k number. The EM algorithm provided best results when used with the Edge histogram filter (for $k = 1$ the classification was 98.59%).

The future works include testing of the performance of the proposed classifier based on the Poincaré metric on color photos,and usage of other filters in the pre-processing phase. It could also be explored with other classification methods such as SVM, decision trees, or neural networks.

References

1. Chatzichristofis, S.A., Boutalis, Y.S.: FCTH: fuzzy color and texture histogram-a low level feature for accurate image retrieval. In: 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services, pp. 191–196. IEEE (2008)
2. Climer, S., Bhatia, S.K.: Image database indexing using JPEG coefficients. Pattern Recogn. **35**(11), 2479–2488 (2002)
3. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Amsterdam (2011)
4. Hatem, H., Beiji, Z., Majeed, R.: A survey of feature base methods for human face detection. Int. J. Control Autom. **8**(5), 61–78 (2015)
5. Huang, H.M., Liu, H.S., Liu, G.P.: Face recognition using pyramid histogram of oriented gradients and SVM. AISS: Adv. Inf. Sci. Serv. Sci. **4**(18), 1–8 (2012)
6. Jafri, R., Arabnia, H.R.: A survey of face recognition techniques. Jips **5**(2), 41–68 (2009)
7. Jain, A.K., Bolle, R., Pankanti, S.: Biometrics: Personal Identification in Networked Society, vol. 479. Springer, Heidelberg (2006). https://doi.org/10.1007/978-0-387-32659-7
8. Krantz, S.G.: Complex Analysis: The Geometric Viewpoint, No. 23. MAA (2004)
9. Mohamed, A., Yampolskiy, R.: Recognizing artificial faces using wavelet based adapted median binary patterns. In: The Twenty-Sixth International FLAIRS Conference (2013)
10. Pietikäinen, M., Hadid, A., Zhao, G., Ahonen, T.: Local binary patterns for still images. Computer Vision Using Local Binary Patterns, vol. 40, pp. 13–47. Springer, London (2011). https://doi.org/10.1007/978-0-85729-748-8_2
11. Popova, A., Neshov, N.N.: Image similarity search approach based on the best features ranking. Egypt. Comput. Sci. J. **37**(1), 51–65 (2013)
12. Tan, P.N.: Introduction to Data Mining. Pearson Education India (2018)
13. Thorgeirsson, S.: Hyperbolic geometry: history, models, and axioms (2014)
14. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
15. Zhang, D., Wong, A., Indrawan, M., Lu, G.: Content-based image retrieval using Gabor texture features. IEEE Trans. Pami **3656**, 1315 (2000)