A New Line Symmetry Distance Based Pattern Classifier

Sriparna Saha    Sanghamitra Bandyopadhyay    Chingtham Tejbanta Singh

Abstract—In this paper, a new line symmetry based classifier (LSC) is proposed to deal with pattern classification problems. In order to measure total amount of line symmetry of a particular point in a class, a new definition of line symmetry based distance is also proposed in this paper. The proposed line symmetry based classifier (LSC) utilizes this new definition of line symmetry distance for classifying an unknown test sample. LSC assigns an unknown test sample pattern to that class with respect to whose major axis it is most symmetric. The mean of all the training patterns belonging to that particular class is taken as the prototype of that class. Thus training constitutes of computing only the class prototypes and the major axes of those classes. Kd-tree based nearest neighbor search is used for reducing complexity of line symmetry distance computation. The performance of LSC is demonstrated in classifying twelve artificial and real-life data sets of varying complexities. Experimental results show that LSC achieves, in general, higher classification accuracy compared to k-NN classifier. Results indicate that the proposed novel line symmetry based classifier is well-suited for classifying data sets having symmetrical classes, irrespective of any convexity, overlap and size. Statistical analysis, ANOVA is also performed to compare the performance of these classifications techniques.

Keywords: Pattern Classification; Kd-tree; Symmetry based distance; Nearest Neighbor Rule; Line Symmetry

I. INTRODUCTION

Pattern classification is a fundamental problem in artificial intelligence and other fields. The problem can be described generally as follows: Given N training samples with known class labels, which can be divided into C classes, \( N_k \) is the size of class \( k, k \in \{1, 2, \ldots, C\} \), how to predict the class label of an unknown sample \( \mathbf{u} \)? Many methods have been suggested to tackle this problem. Nearest neighbor (NN) classifier, first proposed by Fix and Hodges, 1951 [1], is one of the oldest and simplest, yet effective method for performing, in general, non-parametric pattern classification [2]. The NN rule classifies an unseen pattern \( \mathbf{u} \) to the class of its nearest neighbor in the training data. To identify the nearest neighbor of a query pattern, a distance function has to be defined to measure the similarity between two patterns. In the absence of prior knowledge, the Euclidean and Manhattan distance functions have conventionally been used as similarity measures for computational convenience. The empirical evaluation on data in various fields shows that NN is robust and has asymptotic error rate that is at most twice the Bayes error rate [2]. NN classifier suffers from the major limitation of requiring \( N \) distance computations, for computing the nearest neighbor of a point. A simple approach based on rearrangement of the training data set in a certain order, such that the number of distance computations is significantly reduced is proposed in [3]. In the recent years many approaches have been adopted to make the nearest neighbor classification fast [4], [5], [6], [7]. In [8] a technique to choose a proper value of \( k \) in \( k \)-NN classifier has been proposed. Some variants of the nearest neighbor classifier have been proposed in [9] and [10]. There are also other algorithms in the literature, like [11] that use prototypes to classify data.

It may be noted that one of the basic feature of shapes and objects is symmetry. Symmetry is considered a pre-attentive feature which enhances recognition and reconstruction of shapes and objects [12]. Almost every interesting area around us consists of some generalized form of symmetry. As symmetry is so common in the natural world, it can be assumed that some kind of symmetry exists in the classes also. Based on this, a new point symmetry based distance \( d_{PS} \) (PS-distance) is developed in [13]. This overcomes the limitations of an earlier point symmetry distance developed in [14]. For reducing the complexity of computing the PS-distance, use of Kd-tree [15] is also proposed in [13].

From the geometrical symmetry viewpoint, point symmetry and line symmetry are two widely discussed issues. In this paper we have developed a new line symmetry based distance measure, and then proposed a novel classifier with the new line symmetry distance. In the proposed classifier, an unknown test pattern is assigned to the class with respect to whose major axis it is most “symmetric”. The newly developed line symmetry based distance is used to determine the amount of symmetry between an unknown test pattern and the samples of a specific class. The training time of the proposed line symmetry based classifier (LSC) constitutes of computing only the class prototypes and the major axis of those classes. The Kd-tree based nearest neighbor search is also utilized here to reduce the computational complexity of computing the line symmetry distance. The effectiveness of the proposed LSC is demonstrated for classifying twelve artificial and real-life data sets of varying complexities. Results show that the proposed classification technique, line symmetry based classifier (LSC), is capable to classify both convex and non-convex classes of any shape and sizes as long as they do have some line symmetry property. Comparison with the \( k \)-NN rule shows that the newly developed novel classifier is competitive and can be used as an alternative method for pattern classification specially for data sets having classes with symmetrical structure.

The paper is organized as follows. In Section II, the

978-1-4244-1821-3/08/$25.00 ©2008 IEEE
existing point symmetry based distance is described first and then the new definition of line symmetry based distance is presented. Section III deals with the newly proposed line symmetry based classification technique (LSC). Section IV shows the experimental results. Finally Section V concludes the paper.

II. A NEW DEFINITION OF THE LINE SYMMETRY DISTANCE (LSC)

In this section, the existing point symmetry based distance is described to start with. Next, we will define the newly proposed line symmetry based distance. Then, the properties of the newly proposed line symmetry based distance is described in detail. Finally, use of Kd-tree for reducing computational complexity of line symmetry based distance calculation is presented.

A. Existing Point Symmetry Based Distance

In this section, the existing point symmetry distance [13], \(d_{ps}(\vec{x}, \vec{y})\), associated with point \(\vec{x}\) with respect to a center \(\vec{y}\) is described. As shown in [13], \(d_{ps}(\vec{x}, \vec{y})\) is able to overcome some serious limitations of an earlier PS distance [14]. Let a point be \(\vec{x}\). The symmetrical (reflected) point of \(\vec{x}\) with respect to a particular centre \(\vec{y}\) is \(2 \times \vec{x} - \vec{y}\). Let us denote this by \(\vec{y}^*\). Let \(\text{knear}\) unique nearest neighbors of \(\vec{y}^*\) be at Euclidean distances of \(d_i, i = 1, 2, \ldots, \text{knear}\). Then

\[
d_{ps}(\vec{x}, \vec{y}) = d_{sym}(\vec{x}, \vec{y}^*) 	imes d_c(\vec{x}, \vec{y}),
\]

(1)

\[
d_{ps}(\vec{x}, \vec{y}) = \sum_{i=1}^{\text{knear}} d_i 	imes d_c(\vec{x}, \vec{y}),
\]

(2)

where \(d_c(\vec{x}, \vec{y})\) is the Euclidean distance between the point \(\vec{x}\) and \(\vec{y}\). It can be seen from Equation 2 that \(\text{knear}\) cannot be chosen equal to 1, since if \(\vec{y}^*\) exists in the data set then \(d_{ps}(\vec{x}, \vec{y}) = 0\) and hence there will be no impact of the Euclidean distance. On the contrary, large values of \(\text{knear}\) may not be suitable because it may underestimate the amount of symmetry of a point with respect to a particular cluster center. Here \(\text{knear}\) is chosen equal to 2. It may be noted that the proper value of \(\text{knear}\) largely depends on the distribution of the data set. A fixed value of \(\text{knear}\) may have many drawbacks. For instance, for very large clusters (with too many points), 2 neighbors may not be enough as it is very likely that a few neighbors would have a distance close to zero. On the other hand, clusters with too few points are more likely to be scattered, and the distance of the two neighbors may be too large. Thus a proper choice of \(\text{knear}\) is an important issue that needs to be addressed in the future. Note that \(d_{ps}(\vec{x}, \vec{y})\) is a non-metric, is a way of measuring the amount of symmetry between a point and a cluster center, rather than the distance like any Minkowski distance.

The benefits of using several neighbors instead of just one in Equation 2 are as follows.

1) Here since the average distance between \(\vec{y}^*\) and its \(\text{knear}\) unique nearest neighbors have been taken, this term will never be equal to 0, and the effect of \(d_c(\vec{x}, \vec{y})\), the Euclidean distance, will always be considered. Note that if only the nearest neighbor of \(\vec{y}^*\) is considered and this happens to coincide with \(\vec{y}^*\), then this term will be 0, making the distance insensitive to \(d_c(\vec{x}, \vec{y})\). This in turn would indicate that if a point is marginally more symmetrical to a far off cluster than to a very close one, it would be assigned to the farthest cluster. This often leads to undesirable results as demonstrated in [13].

2) Considering the \(\text{knear}\) nearest neighbors in the computation of \(d_{ps}\) makes the PS-distance more robust and noise resistant. From an intuitive point of view, if this term is less, then the likelihood that \(\vec{x}\) is symmetrical with respect to \(\vec{y}\) increases. This is not the case when only the first nearest neighbor is considered which could mislead the method in noisy situations.

B. The Newly Proposed Line Symmetry Based Distance

Given a particular data set, we first find the symmetrical line of each class by using the centroidal moment technique [16]. Let the points in a particular class be denoted by \(X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}\), then the \(\text{p.q}\)th order moment is defined as

\[
m_{pq} = \sum_{\forall (x_i, y_i) \in X} (x_i - \vec{y})^p (y_i - \vec{y})^q,
\]

(3)

By Equation 3, the centroid of the given class is then defined as \((\frac{m_{00}}{m_{01}}, \frac{m_{01}}{m_{00}})\). The central moment is defined as

\[
u_{pq} = \sum_{\forall (x_i, y_i) \in X} (x_i - \vec{y})^p (y_i - \vec{y})^q,
\]

(4)

where \(\vec{x} = \frac{m_{00}}{m_{01}}\) and \(\vec{y} = \frac{m_{01}}{m_{00}}\). According to the calculated centroid and Equation 4, the major axis of each class can be determined by the following two items:

1) The major axis of the class must pass through the centroid.

2) The angle between the major axis and the \(x\) axis is equal to \(0.5 \times \tan^{-1}(\frac{2 \times \nu_{02}}{\nu_{20} - \nu_{02}})\).

Consequently, for one class, its major axis is thus expressed by \((\frac{m_{01}}{m_{00}}, \frac{m_{02}}{m_{00}}), 0.5 \times \tan^{-1}(\frac{2 \times \nu_{02}}{\nu_{20} - \nu_{02}})\).

The obtained major axis is treated as the symmetrical line of the relevant class. This symmetrical line is used to measure the amount of line symmetry of a particular point in that class. In order to measure the amount of line symmetry of a point (\(\vec{y}\)) with respect to a particular line \(\vec{t}\), \(d_{ls}(\vec{x}, \vec{y})\), the following steps are followed.

1) For a particular data point \(\vec{x}\), calculate the projected point \(\vec{p}_i\) on the relevant symmetrical line \(\vec{t}\).

2) Find \(d_{sym}(\vec{x}, \vec{p}_i)\) as:

\[
d_{sym}(\vec{x}, \vec{p}_i) = \sum_{i=1}^{\text{knear}} d_i \times d_{ls}(\vec{x}, \vec{p}_i),
\]

(5)

where \(\text{knear}\) unique nearest neighbors of \(\vec{y}^* = 2 \times \vec{p}_i - \vec{y}\) are at Euclidean distances of \(d_i, i = 1, 2, \ldots, \text{knear}\). ANN library [17] utilizing Kd-tree based nearest neighbor search is used to reduce the complexity of computing these \(d_i\)s (as described in Section II-C). Then the
From Figure 1, it is evident that, for a particular point \( \mathbf{x} \) with respect to that particular symmetrical line of class \( i \), the amount of line symmetry of a particular point \( \mathbf{x} \) with respect to that particular symmetrical line of class \( i \), is calculated as:

\[
d_{ls}(\mathbf{x},i) = d_{sym}(\mathbf{x},\mathbf{\bar{C}}_i) \times d_e(\mathbf{x},\mathbf{\bar{C}})
\]

where \( \mathbf{\bar{C}} \) is the centroid of the particular class \( i \), i.e.,

\[
\mathbf{\bar{C}} = \left( \frac{\min \mathbf{C}_i + \max \mathbf{C}_i}{2} \right).
\]

Next, we will define some properties of the newly proposed line symmetry based distance.

**Definition 1:** The Euclidean distance difference (EDD) property is defined as follows:

Let \( \mathbf{x} \) be a data point, \( \mathbf{\tau}_1 \) and \( \mathbf{\tau}_2 \) be two class centers, and \( \theta \) be a distance measure. Let \( \theta_1 = \theta(\mathbf{x},\mathbf{\tau}_1) \), \( \theta_2 = \theta(\mathbf{x},\mathbf{\tau}_2) \), \( d_{e1} = d_e(\mathbf{x},\mathbf{\tau}_1) \) and \( d_{e2} = d_e(\mathbf{x},\mathbf{\tau}_2) \). Then \( \theta \) is said to satisfy the EDD property if for \( \theta_1 \approx \theta_2 \), point \( \mathbf{x} \) is assigned to \( \mathbf{\tau}_1 \) if \( d_{e1} < d_{e2} \), otherwise it is assigned to \( \mathbf{\tau}_2 \).

**Observation 1:** The proposed line symmetry based distance measure satisfies the Euclidean distance difference property.

**Proof:** Using \( k_{near} = 2 \), let the two nearest neighbors of a data point \( \mathbf{\tau} \) in Figure 1 be at distances of \( d_1 \) and \( d_2 \), respectively. Then

\[
d_{ls}(\mathbf{\tau},1) = d_{sym}(\mathbf{\tau},\mathbf{\bar{C}}_1) \times d_e(\mathbf{\tau},\mathbf{\bar{C}}) = \frac{d_1 + d_2}{2} \times d_{e1}
\]

where \( d_{e1} \) is the Euclidean distance between \( \mathbf{\tau} \) and \( \mathbf{\tau}_1 \). Here \( \mathbf{\tau}_1 \) is the centroid of class 1. Let the two nearest neighbors of \( \mathbf{\tau} \) be at distances of \( d_1 \) and \( d_2 \), respectively. Thus

\[
d_{ls}(\mathbf{\tau},1) = d_{sym}(\mathbf{\tau},\mathbf{\bar{C}}_1) \times d_{e1} = \frac{d_1 + d_2}{2} \times d_{e1},
\]

where \( d_{e1} \) is the Euclidean distance between \( \mathbf{\tau} \) and \( \mathbf{\tau}_1 \). Here \( \mathbf{\tau}_1 \) is the centroid of class 2. Now in order to preserve the EDD property, \( d_{ls}(\mathbf{\tau},1) \) should be less than \( d_{ls}(\mathbf{\tau},2) \) even when \( d_{sym}(\mathbf{\tau},1) \approx d_{sym}(\mathbf{\tau},2) \). Now,

\[
\frac{d_1 + d_2}{2} \times d_{e1} < \frac{d_2 + d_2}{2} \times d_{e2}
\]

\[
\Rightarrow \frac{d_1 + d_2}{2} \times d_{e1} < \frac{d_2 + d_2}{2} \times d_{e2}
\]

\[
\Rightarrow \frac{d_{e1}}{d_{e2}} < \frac{d_2 + d_2}{d_1 + d_2}.
\]

From Figure 1, it is evident that, \( d_{e2} \gg d_{e1} \), so \( \frac{d_{e1}}{d_{e2}} \ll 1 \). Thus even when \( (d_2 + d_2) \approx (d_1 + d_1) \), the inequality in Equation 7 is satisfied. Therefore the proposed distance satisfies EDD property.

**Definition 2:** If two clusters are symmetrical to each other with respect to a third cluster center, then these clusters are called “symmetrical interclusters”.

**Observation 2:** The proposed \( d_{ls} \) measure is able to detect the symmetrical interclusters properly.

**Proof:** In Figure 1, the first and the third clusters are “symmetrical interclusters” with respect to the middle one. As explained in the above example, though \( \mathbf{\tau} \) is symmetric with respect to the symmetrical line of cluster 2, but \( \mathbf{\tau} \) is assigned to cluster 1 as the newly developed \( d_{ls} \) distance satisfies the EDD property. As a result, the three clusters presented in Figure 1 are identified properly. Thus it is proved that the proposed line symmetry based distance is able to detect symmetrical interclusters properly.

It is evident that the symmetrical distance computation is very time consuming because it involves the computation of the nearest neighbors. Computation of \( d_{ls}(\mathbf{\tau},i) \) is of complexity \( O(N) \). Hence for \( N \) points and \( K \) clusters, the complexity of computing the line symmetry based distance between all the points to different clusters is of complexity \( O(N^2 K) \). In order to reduce the computational complexity, an approximate nearest neighbor search using the Kd-tree approach is adopted in this article.

**C. Kd-tree Based Nearest Neighbor Computation**

A K-dimensional tree, or Kd-tree is a space-partitioning data structure for organizing points in a K-dimensional space. A Kd-tree uses only those splitting planes that are perpendicular to one of the coordinate axes. In the nearest neighbor problem a set of data points in d-dimensional space is given. These points are preprocessed into a data structure, so that given any query point \( q \), the nearest or generally \( k \) nearest points of \( p \) to \( q \) can be reported efficiently. ANN (Approximate Nearest Neighbor) is a library written in C++ [17], which supports data structures and algorithms for both exact and approximate nearest neighbor searching in arbitrarily high dimensions. In this article ANN is used to find \( d_{ls} \), where \( i = 1, \ldots, k_{near} \), in Equation 5 efficiently. The ANN library implements a number of different data
structures, based on Kd-trees and box-decomposition trees, and employs a couple of different search strategies. ANN allows the user to specify a maximum approximation error bound, thus allowing the user to control the tradeoff between accuracy and running time. In this work, ANN library utilizing Kd-tree for nearest neighbor search is used. The points in a particular data set are stored in a Kd-tree data structure. Thus, it requires the construction of a Kd-tree consisting of n points in the data set, where n is the size of the data set.

The function performing the k-nearest neighbor search in ANN is given a query point q, a nonnegative integer k, an array of point indices, mni, and an array of distances, dists. Both arrays are assumed to contain at least k elements. This procedure computes the k nearest neighbors of q in the point set, and stores the indices of the nearest neighbors in the array mni. Optionally a real value ε ≥ 0 may be supplied. If so, then i th nearest neighbor is (1 + ε) approximation to the true i th nearest neighbor. That is, the true distance to this point may exceed the true distance to the real i th nearest neighbor of q by a factor of (1 + ε). If ε is omitted then the nearest neighbors will be computed exactly. For the purpose of this article, the exact nearest neighbor is computed; so the ε is set equal to 0 and k = knear, in this article it is k = 2. In order to find line symmetry distance of a particular point τ with respect to the symmetrical line of class i, we have to find the first knear nearest neighbors of τ which is equal to 2 × p_i − τ, where p_i is the projection of τ on the major axis/symmetrical line of class i. Therefore the query point q is set equal to τ. After getting the knear nearest neighbors of τ the symmetrical distance of τ with respect to the symmetrical line of class i is calculated using Equation 6.

III. NEWLY PROPOSED LINE SYMMETRY BASED CLASSIFICATION TECHNIQUE (LSC)

In this section at first the newly proposed line symmetry based classifier is proposed. Finally, a detailed complexity analysis of the proposed classification technique is presented.

A. Line Symmetry Based Classification Technique

The main steps of line symmetry based classification technique are as follows.

- **Training Phase**:
  - Let τ_k be a training sample of class k. Let τ_k be the prototype of class k, which can be calculated by
    \[ τ_k = \frac{\sum_{i=1}^{N_k} \tau_i}{N_k} \]
    where N_k is the total number of points in class k, for k = 1, . . . , C, where C is the total number of classes present in the dataset.
  - Find the symmetrical line for each class. As described in the first paragraph of Section II-B, for each class, we use the moment-based approach to find out the relevant symmetrical line of each class. Here the training points of each class are used to calculate the centroid and the symmetrical line of that particular class.

- **Test Phase**: According to the proposed line symmetry based classifier, an unknown sample τ, is classified as follows:
  - Compute d_k(τ, k) by Equation 6, for k = 1, . . . , C.
  - Then according to the proposed line symmetry based classifier, the unknown sample τ, is classified into class o, where
    \[ o = \arg\min_{k=1}^{C} d_k(τ, k). \] (9)

Here d_k(τ, k) is computed by Equation 6. ANN library [17] is used to reduce the computational burden of computing d_k(τ, k). ANN utilizes Kd-tree based nearest neighbor search, and thus a Kd-tree has to be constructed. Here Kd-tree is consisting of \( N = \sum_{k=1}^{C} N_k \) number of points, i.e., the samples of the training set.

B. Complexity Analysis of the Proposed LSC technique

The training phase of LSC constitutes of 1) computing the centroid and the symmetrical line of each class, and 2) constructing a Kd-tree data structure consisting of total number of training samples. The construction of Kd-tree requires \( O(n \log(n)) \) time and \( O(n) \) space [15], where n is the total number of points in the Kd-tree. For LSC, we have to construct Kd-tree having N number of elements where \( N = \sum_{k=1}^{C} N_k \), N is the total number of training points. So construction time of Kd-tree is \( O(N \times \log(N)) \). Computation of centroid and the symmetrical line for each class is of complexity \( O(N) \). Thus total training time complexity of LSC is \( O(N \times \log(N)) \).

In the test phase, total C line symmetry based distances have to be computed in order to classify an unknown test sample. The computational complexity of d_k is due to calculation of the nearest neighbor distances (d_i, s in Equation 5). For this purpose the Kd-tree based nearest neighbor search is used. If the points are roughly uniformly distributed, then the expected case complexity is \( O(\epsilon^d + \log(N)) \), where \( \epsilon \) is a constant depending on dimension and the point distribution. This is \( O(\log(N)) \) if the dimension d is a constant [18]. Friedman et al. [19] also reported \( O(\log(N)) \) expected time for finding the nearest neighbor. Thus computational complexity of d_k is \( O(\log(N)) \). So, for C classes, total time complexity due to testing becomes \( O(C \times \log(N)). \)

IV. IMPLEMENTATION RESULTS AND COMPARATIVE STUDY

In this section, at first a short description of the data sets used for experiment are provided. Later, experimental results are explained in detail.

A. Data Sets Used

To evaluate the performance of the proposed line symmetry based classifier, six artificial and six real-life data sets are used. A short description of the data sets, used here, in
### Table I
Description of data sets

<table>
<thead>
<tr>
<th>Name</th>
<th>No of points</th>
<th>dimension</th>
<th>No of classes</th>
<th>Points in individual classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2line</td>
<td>400</td>
<td>2</td>
<td>2</td>
<td>200, 200</td>
</tr>
<tr>
<td>2ring</td>
<td>300</td>
<td>2</td>
<td>2</td>
<td>200,200</td>
</tr>
<tr>
<td>3mixed</td>
<td>300</td>
<td>2</td>
<td>3</td>
<td>50, 200, 50</td>
</tr>
<tr>
<td>pat</td>
<td>417</td>
<td>2</td>
<td>2</td>
<td>223, 194</td>
</tr>
<tr>
<td>Two Leaves 1</td>
<td>657</td>
<td>2</td>
<td>2</td>
<td>429, 228</td>
</tr>
<tr>
<td>Two Leaves 2</td>
<td>580</td>
<td>2</td>
<td>2</td>
<td>267, 313</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3</td>
<td>50, 50, 50</td>
</tr>
<tr>
<td>Cancer</td>
<td>683</td>
<td>9</td>
<td>2</td>
<td>444, 239</td>
</tr>
<tr>
<td>Newthyroid</td>
<td>215</td>
<td>5</td>
<td>3</td>
<td>150, 35, 30</td>
</tr>
<tr>
<td>Wine</td>
<td>178</td>
<td>13</td>
<td>3</td>
<td>99, 71, 48</td>
</tr>
<tr>
<td>Vowel</td>
<td>871</td>
<td>3</td>
<td>6</td>
<td>72, 89, 172, 151, 207, 180</td>
</tr>
<tr>
<td>Lung cancer</td>
<td>32</td>
<td>56</td>
<td>3</td>
<td>9, 13, 10</td>
</tr>
</tbody>
</table>

### Table II
Average classification accuracies (Acc) obtained by k-NN classifier, Proposed Point Symmetry Based Classifier (PSC) and Proposed Line Symmetry Based Classifier (LSC) on different data sets

<table>
<thead>
<tr>
<th>Name</th>
<th>where N=10% of data set</th>
<th>where N=20% of the data set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>k-NN</td>
<td>LSC</td>
</tr>
<tr>
<td>2line</td>
<td>85.38 ± 0.25</td>
<td>92.82 ± 1.0</td>
</tr>
<tr>
<td>2ring</td>
<td>89.18 ± 1.5</td>
<td>93.1 ± 1.0</td>
</tr>
<tr>
<td>3mixed</td>
<td>91.48 ± 1.2</td>
<td>96.65 ± 0.08</td>
</tr>
<tr>
<td>pat</td>
<td>75.1 ± 1.2</td>
<td>79.1 ± 1.5</td>
</tr>
<tr>
<td>Two Leaves 1</td>
<td>94.52 ± 1.4</td>
<td>94.6 ± 1.1</td>
</tr>
<tr>
<td>Two Leaves 2</td>
<td>95.64 ± 1.4</td>
<td>98.47 ± 1.0</td>
</tr>
<tr>
<td>Iris</td>
<td>91.7 ± 1.5</td>
<td>94.8 ± 1.2</td>
</tr>
<tr>
<td>Cancer</td>
<td>90.94 ± 1.1</td>
<td>92.78 ± 1.1</td>
</tr>
<tr>
<td>Newthyroid</td>
<td>74.12 ± 0.21</td>
<td>86.2 ± 1.2</td>
</tr>
<tr>
<td>Wine</td>
<td>66.17 ± 0.20</td>
<td>73.1 ± 1.2</td>
</tr>
<tr>
<td>Vowel</td>
<td>73.05 ± 1.3</td>
<td>74.72 ± 1.1</td>
</tr>
<tr>
<td>Lung cancer</td>
<td>67.04 ± 1.1</td>
<td>69.12 ± 1.2</td>
</tr>
</tbody>
</table>

### Table III
Estimated marginal means and pairwise comparison of both the algorithms on classification accuracies obtained by ANOVA testing for all the data sets where training set consisting of 10% of the total data

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Algo. Name (I)</th>
<th>Comp. Algo. (J)</th>
<th>Mean Diff. (I-J)</th>
<th>Significance value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2line</td>
<td>LSC</td>
<td>k-NN</td>
<td>+5.52 ± 1.9</td>
<td>2.4603e−008</td>
</tr>
<tr>
<td>2ring</td>
<td>LSC</td>
<td>k-NN</td>
<td>+4.12 ± 1.1</td>
<td>2.4239e−008</td>
</tr>
<tr>
<td>3mixed</td>
<td>LSC</td>
<td>k-NN</td>
<td>+5.17 ± 1.0</td>
<td>4.2113e−009</td>
</tr>
<tr>
<td>pat</td>
<td>LSC</td>
<td>k-NN</td>
<td>+2.95 ± 0.79</td>
<td>2.4123e−008</td>
</tr>
<tr>
<td>Two Leaves 1</td>
<td>LSC</td>
<td>k-NN</td>
<td>+0.08 ± 1.2</td>
<td>0.95</td>
</tr>
<tr>
<td>Two Leaves 2</td>
<td>LSC</td>
<td>k-NN</td>
<td>+4.83 ± 1.2</td>
<td>9.8183e−009</td>
</tr>
<tr>
<td>Iris</td>
<td>LSC</td>
<td>k-NN</td>
<td>+3.1 ± 1.2</td>
<td>1.1661e−006</td>
</tr>
<tr>
<td>Cancer</td>
<td>LSC</td>
<td>k-NN</td>
<td>+1.84 ± 1.1</td>
<td>2.7671e−008</td>
</tr>
<tr>
<td>Newthyroid</td>
<td>LSC</td>
<td>k-NN</td>
<td>+12.08 ± 1.8</td>
<td>4.1334e−009</td>
</tr>
<tr>
<td>Wine</td>
<td>LSC</td>
<td>k-NN</td>
<td>+6.93 ± 1.6</td>
<td>1.1241e−006</td>
</tr>
<tr>
<td>Vowel</td>
<td>LSC</td>
<td>k-NN</td>
<td>+1.67 ± 1.1</td>
<td>2.2171e−008</td>
</tr>
<tr>
<td>Lung cancer</td>
<td>LSC</td>
<td>k-NN</td>
<td>+2.08 ± 1.1</td>
<td>9.8183e−009</td>
</tr>
</tbody>
</table>
terms of the number of points presents in each dataset, the dimension of the data and the number of classes present in each data set, is provided in Table I. An elaborate description is provided below.

1) 2line: This data set, used in [14], consists of two bands as shown in Figure 2(a).
TABLE IV
ESTIMATED MARGINAL MEANS AND PAIRWISE COMPARISON OF BOTH THE ALGORITHMS ON CLASSIFICATION ACCURACIES OBTAINED BY ANOVA TESTING FOR ALL THE DATA SETS WHERE TRAINING SET CONSISTING OF 20% OF THE TOTAL DATA SET

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Algo. Name (I)</th>
<th>Comp. Algo. (J)</th>
<th>Mean Diff. (I-J)</th>
<th>Significance value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2line</td>
<td>LSC</td>
<td>k-NN</td>
<td>+3.4 ± 0.4</td>
<td>2.4503e−008</td>
</tr>
<tr>
<td>2ring</td>
<td>LSC</td>
<td>k-NN</td>
<td>+2.8 ± 1.0</td>
<td>2.4539e−008</td>
</tr>
<tr>
<td>3mixed</td>
<td>LSC</td>
<td>k-NN</td>
<td>+4.73 ± 0.85</td>
<td>4.2213e−009</td>
</tr>
<tr>
<td>pat</td>
<td>LSC</td>
<td>k-NN</td>
<td>+5.33 ± 1.4</td>
<td>2.4123e−008</td>
</tr>
<tr>
<td>Two leaves1</td>
<td>LSC</td>
<td>k-NN</td>
<td>+0.1 ± 1.0</td>
<td>0.95</td>
</tr>
<tr>
<td>Newthyroid</td>
<td>LSC</td>
<td>k-NN</td>
<td>+4.73 ± 0.82</td>
<td>9.8183e−009</td>
</tr>
<tr>
<td>Iris</td>
<td>LSC</td>
<td>k-NN</td>
<td>+8.95 ± 0.75</td>
<td>4.1334e−009</td>
</tr>
<tr>
<td>Cancer</td>
<td>LSC</td>
<td>k-NN</td>
<td>+1.88 ± 0.9</td>
<td>1.1241e−006</td>
</tr>
<tr>
<td>Vowel</td>
<td>LSC</td>
<td>k-NN</td>
<td>+1.18 ± 1.2</td>
<td>2.217e−008</td>
</tr>
<tr>
<td>Lung-cancer</td>
<td>LSC</td>
<td>k-NN</td>
<td>+1.08 ± 1.1</td>
<td>9.8183e−009</td>
</tr>
</tbody>
</table>

2) **2ring**: This data set, used in [14], distributed on two crossed ellipsoidal shells, shown in Figure 2(b). This is a non-convex symmetrical data set.
3) **3mixedclusters**: This data set, used in [14], (shown in Figure 3(a)) is a combination of ring-shaped non-convex, spherically compact and linear classes.
4) **Pat**: This data set, used in [20], consists of 2 non-linear, non-overlapping, non-symmetric classes. The data set is shown in Figure 3(b).
5) **Two leaves**: Most of the natural scenes, such as leaves of plants, have the line symmetry property. Figure 4(a) shows the two real leaves of *Ficus microcapa* and they overlap a little each other. First the Sobel edge detector [16] is used to obtain the edge pixels in the input data points which is shown in Figure 4(b).
6) **Two leaves**: Figure 5(a) shows the two real leaves of *Ficus lvgi*. The edge map obtained after application of the Sobel edge detector [16] is shown in Figure 5(b).
7) **Iris**: This data set, obtained from [21], represents different categories of irises characterized by four feature values. It has three classes: Setosa, Versicolor and Virginica. It is known that the two classes (Versicolor and Virginica) have a large amount of overlap while the class Setosa is linearly separable from the other two.
8) **Cancer**: Here we use the Wisconsin Breast Cancer data set, obtained from [21]. Each pattern has nine features corresponding to clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. There are two categories in the data: malignant and benign. The two classes are known to be linearly separable.
9) **Newthyroid**: This is thyroid gland data (obtained from [21]). Five laboratory tests are used to predict whether a patient’s thyroid belongs to the class euthyroidism, hypothyroidism or hyperthyroidism.
10) **Wine**: This is the Wine recognition data (obtained from [21]), resulting from a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.
11) **Lung-cancer**: This data set, obtained from [21], describes 3 types of pathological lung cancers.
12) **Vowel**: This data consists of 871 Indian Telegu vowel sounds, described in [22]. These were uttered in a constant-vowel-consonant context by three male speakers in the age group of 30-35 years.

B. Discussion of Results

The proposed line symmetry based classifier (LSC) is tested on the data sets described in Section IV-A. For the purpose of comparison, the $k$-NN nearest neighbor classifier (with $k = \sqrt{\text{trn}}$, where $\text{trn}$ is the total number of points in the training data set), is also executed on the above mentioned data sets. In each case, at first, training set is consisting of randomly chosen 10% data points from the whole data set and the rest of the data set constitutes the test set. For each data set, 5 different training and test sets have been formed randomly and both the classifiers are executed on them. The average classification accuracies and their variances obtained by both the classifiers for these 5 different runs are noted down. Table II shows the average classification accuracies and the variances of both the classifiers for all the data sets used here. Experimental results show that, in general, the proposed line symmetry based classifier (LSC) performs better than the $k$-NN classifier for almost all the data sets used here for experiment. LSC is capable to classify any data sets having line symmetric classes. Next, the training set size is increased to 20% of the whole data set and the average classification accuracies are calculated. Those are also reported in Table II. It can be easily seen from the given result that classification accuracies of all the classifiers have increased after increasing the training set size.
ANOVA [23] statistical analysis is performed on the combined results of the three algorithms. The One-Way ANOVA procedure produces a one-way analysis of variance for a quantitative dependent variable (here it is Classification Accuracy (Acc)) by a single independent variable (here it is the algorithm). Analysis of variance is used to test the hypothesis that several means are equal. ANOVA results for all the data sets where training set consisting of total 10% of the data set are reported in Table III. Similarly Table IV shows the ANOVA results for all the datasets where training set consisting of total 20% of the data set. ANOVA analysis reveals that although the mean accuracy of LSC for TwoLeaves1 is slightly better than that of $k$-NN, the corresponding difference in mean accuracies are not statistically significant (with significance value=0.95). This indicates that for this data set both the classifiers perform similarly. But for all other data sets the performance of LSC is better than that of $k$-NN classifier and this is also statistically significant (in all the cases significance value < 0.05).

Results on the above mentioned data sets show that LSC is able to classify any type of classes, irrespective of their geometrical shape and overlapping nature, as long as they possess the characteristic of line symmetry property. Based on this observation, and the fact that the property of symmetry is widely evident in real-life situations, application of LSC in most classification tasks seems justified and is therefore recommended.

V. DISCUSSION AND CONCLUSIONS

In this paper, a new definition of line symmetry distance is proposed and a classification technique is developed based on this newly defined line symmetry distance. This newly developed line symmetry based classification technique (LSC) classifies an unknown sample to a particular class, with respect to which it has highest degree of symmetry. In order to measure the amount of symmetry of an unknown test pattern with respect to the symmetrical line of each training class, the newly proposed line symmetry based distance is utilized. Kd-tree based nearest neighbor search is used to reduce the computational complexity of computing line symmetry based distance. The proposed LSC is very simple taking $O(N \log(N))$ time for training and $O(C \log(N))$ time for testing, where $N$ is the total number of training samples and $C$ is the total number of classes present in the data set. Experimental results on several artificial and real life data sets of varying complexities show that the proposed LSC is, in general, better than that of $k$-NN classifier. Results also reveal the fact that LSC is capable of classifying data sets having classes of any type of symmetry, irrespective of their size, shape and convexity. Note that LSC seeks for classes which are line symmetric with respect to their centers. Thus LSC will fail if the classes do not have this property.

Future work includes defining some other form of symmetry such as plane symmetry etc. Developing some clustering techniques based on the proposed line symmetry distance is another direction of future work.

REFERENCES