Time Series Classification by Class-Based Mahalanobis Distances

Zoltán Prekopcsák, Daniel Lemire

Abstract
To classify time series by nearest neighbor, we need to specify or learn a distance. We consider several variations of the Mahalanobis distance and the related Large Margin Nearest Neighbor Classification (LMNN). We find that the conventional Mahalanobis distance is counterproductive. However, both LMNN and the class-based diagonal Mahalanobis distance are competitive.

Key words: Time-series classification, Distance learning, Nearest Neighbor

1. Introduction

Nearest Neighbor (NN) methods classify time series efficiently and accurately [9]. The 1-NN method is especially simple: we merely have to find the nearest labeled instance.

However, we need to specify or learn a distance [27, 29]. We can either learn a single (global) distance function, or learn one distance function per class [8, 19, 20]. That is, to compute the distance between a test element and an instance of class \( j \), we use a distance function specific to class \( j \).

Because the Euclidean distance is popular for NN classification, it is tempting to consider generalized ellipsoid distances [12], that is, distances of the form

\[
D(x, y) = (x - y)^\top M(x - y)
\]

where \( M \) is a positive definite matrix. When the matrix \( M \) is the identity matrix, we recover the (squared) Euclidean distance. We get the (squared) Mahalanobis distance when solving for the matrix \( M \) minimizing the sum of distances \( \sum_{x,y} D(x, y) \) (see §3). We can require \( M \) to be diagonal, thus defining the diagonal Mahalanobis distance [12, 20].

*Corresponding author. Tel.: +36 30 3887691; fax: +36 1 4633107
Email addresses: prekopcsak@tmit.bme.hu (Zoltán Prekopcsák), lemire@gmail.com (Daniel Lemire)

Preprint submitted to Information Processing Letters October 7, 2010
To our knowledge, neither the full nor the diagonal Mahalanobis distances have received much attention as tools to classify time series. Yet, while the result with the global Mahalanobis distance is poor, we show that the diagonal Mahalanobis distance is competitive. Moreover, the class-based diagonal Mahalanobis distance is preferable to the global diagonal Mahalanobis distance. As a further extension of the Mahalanobis distance, we also consider the recently proposed LMNN method [27], and we show that it is also competitive for time series classification.

2. Related Works

Several distance functions are used for time series classification [9]: Dynamic Time Warping (DTW) [23], DISSIM [10], Threshold Queries [2], Edit distances [5, 6], Longest Common Subsequences (LCSS) [26], Swale [18], SpAdE [7], and Cluster, Then Classify (CTC) [14]. Ding et al. [9] had an extensive comparison of these distance functions and concluded that DTW is among the best measures and the accuracy of the Euclidean distance converges to DTW as the size of the training set increases.

In a general Machine Learning setting, Vidal and Paredes [19, 20] compared Euclidean distance with the conventional and class-based Mahalanobis distances. One of our contribution is to validate these generic results on time series: instead of tens of features, we have hundreds or even thousands of values.

More generally, distance metric learning has an extensive literature [4, 11, 21, 25, 28]. We refer the reader to Weinberger and Saul for a review [27]. And there are many extensions and alternatives to NN classification. For example, Jahromi et al. [13] use instance weights to improve classification. Meanwhile, Zhan et al. [30] learn a distance per instance.

3. Mahalanobis distance

For completeness, we derive the Mahalanobis distance [16] as an optimal form of generalized ellipsoid distance. We seek $M$ minimizing

$$\sum_{x,y \in S} (x - y)^T M (x - y) = \sum_{x,y \in S} \left( \sum_{k=1}^{n} \sum_{l=1}^{n} (x_k - y_k) m_{kl} (x_l - y_l) \right)$$

where $S$ is some class of time series. We add a regularization constraint on the determinant $(\det(M) = 1)$.

We have for all $l$ that

$$\det(M) = \sum_{k=1}^{n} (-1)^{k+l} m_{k,l} \det(M_{k,l}) = 1$$

where $M_{k,l}$ is an $(n-1) \times (n-1)$ matrix obtained by deleting $k$-th row and $l$-th column of $M$. Thus, we have

$$\partial \det(M) / \partial m_{k,l} = (-1)^{k+l} \det(M_{k,l})$$

We solve the minimization problem by the Lagrange’s multiplier method with the Lagrangian

$$L(M, \lambda) = \sum_{x,y \in S} \left( \sum_{k=1}^{n} \sum_{l=1}^{n} (x_k - y_k) m_{k,l} (x_l - y_l) \right) - \lambda (\det(M) - 1)$$
Setting the derivatives to zero, we get
\[
\frac{\partial L(M, \lambda)}{\partial m_{k,l}} = \sum_{x,y \in S} (x_k - y_k)(x_l - y_l) - \lambda(-1)^{k+l} \det(M_{k,l}) = 0
\]
and
\[
\det(M_{k,l}) = \frac{\sum_{x,y \in S} (x_k - y_k)(x_l - y_l)}{\lambda(-1)^{k+l}}.
\]
Because \(\det(M) = 1\), the inverse matrix \(M^{-1}\) can be represented as
\[
m_{k,l}^{-1} = \frac{(-1)^{k+l} \det(M_{k,l})}{\det(M)} = (-1)^{k+l} \det(M_{k,l}).
\]
Hence, we have
\[
m_{k,l}^{-1} = \frac{\sum_{x,y \in S} (x_k - y_k)(x_l - y_l)}{\lambda}.
\]
Thus, we have that \(M^{-1} \propto C\) where \(C\) is the covariance matrix. Because we require \(M\) to be positive definite and to satisfy \(\det(M) = 1\), we set \(M = (\det(C))^{-1} C^{-1}\).

The same derivation works when \(M\) is required to be diagonal except that we must set to zero all non-diagonal elements of \(C\). Unfortunately, we cannot extend this analysis to banded matrices since the restriction of a positive definite matrix to a band may fail to be positive definite — except if the band has a width of one (the diagonal matrix). We can also use block-diagonal matrices \([17]\) or introduce other sparsity constraints that keep positive definiteness.

4. LMNN

A conventional distance-learning approach is to find an optimal generalized ellipsoid distance with respect to a specific loss function. The LMNN algorithm proposed by Weinberger and Saul \([27]\) takes a different approach. It seeks to force nearest neighbors to belong to the same class and it separates instances from different classes by a large margin. LMNN can be formulated as a semidefinite programming problem. They also propose a modification which they call multiple metrics LMNN as it learns different distances for each class.

While the class-based Mahalanobis distance can be considered a generative model that trains a model for each class separately, LMNN is a discriminative classifier similar to support vector machines. We have used the source code provided by Weinberger and Saul \([27]\) for the experiments with default parameters.

5. Experiments

The main goal of our experiments is to evaluate Mahalanobis distances and the class-based approach on time series. A secondary goal is to evaluate the LMNN method. The code for the experiments is available online \([1]\) with instructions on how the results can be reproduced.
Table 1: Classification error for the various Mahalanobis distances.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Full</th>
<th>Diagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conventional</td>
<td>class-based</td>
</tr>
<tr>
<td>50 words</td>
<td>0.79</td>
<td>0.86</td>
</tr>
<tr>
<td>Adiac</td>
<td>0.88</td>
<td>0.53</td>
</tr>
<tr>
<td>CBF</td>
<td>0.34</td>
<td>0.51</td>
</tr>
<tr>
<td>ECG</td>
<td>0.32</td>
<td>0.17</td>
</tr>
<tr>
<td>Fish</td>
<td>0.51</td>
<td>0.14</td>
</tr>
<tr>
<td>Face (all)</td>
<td>0.39</td>
<td>0.32</td>
</tr>
<tr>
<td>Face (four)</td>
<td>0.27</td>
<td>0.70</td>
</tr>
<tr>
<td>Gun-Point</td>
<td>0.25</td>
<td>0.12</td>
</tr>
<tr>
<td>Lighting-2</td>
<td>0.39</td>
<td>0.54</td>
</tr>
<tr>
<td>Lighting-7</td>
<td>0.63</td>
<td>0.77</td>
</tr>
<tr>
<td>OSU Leaf</td>
<td>0.74</td>
<td>0.62</td>
</tr>
<tr>
<td>OliveOil</td>
<td>0.23</td>
<td>0.33</td>
</tr>
<tr>
<td>Swedish Leaf</td>
<td>0.75</td>
<td>0.26</td>
</tr>
<tr>
<td>Trace</td>
<td>0.30</td>
<td>0.32</td>
</tr>
<tr>
<td>Two Patterns</td>
<td>0.53</td>
<td>0.49</td>
</tr>
<tr>
<td>Synthetic Control</td>
<td>0.41</td>
<td>0.29</td>
</tr>
<tr>
<td>Yoga</td>
<td>0.45</td>
<td>0.32</td>
</tr>
<tr>
<td># of best errors</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

5.1. Data sets

We use the UCR time series classification benchmark [15] for our experiments as it includes diverse time series data sets from many domains. It has predefined training-test splits for the experiments, so the results can be compared across different papers. We removed the two data that are not z-normalized by default (Beef and Coffee). Indeed, z-normalization improves substantially the classification accuracy—irrespective of the chosen distance. Thus, for fair results, we should z-normalize them, but this may create confusion with previously reported numbers. We also removed the Wafer data set as all distances classify it nearly perfectly. The remaining 17 data sets were used for the comparison of different methods.

5.2. Best Mahalanobis distance for 1-NN accuracy

We compare the various Mahalanobis distances in Table 1. What is immediately apparent is that using the full matrix gives poor classification results. Of course, the diagonal Mahalanobis distances are also considerably faster computationally which makes full-matrix Mahalanobis an especially poor choice. Moreover, class-based distances are preferable. Thus, out of the four variations, we recommend the class-based diagonal Mahalanobis distance.

5.3. Comparing competitive distances

How does the class-based diagonal Mahalanobis distance fares compared to competitive distances? Computationally, it is inexpensive compared to schemes
Table 2: Classification errors for some competitive schemes. We use the class-based diagonal Mahalanobis distance. For the 50 words data set, the LMNN computation fails because it has a class with only one instance.

<table>
<thead>
<tr>
<th>Data set</th>
<th>Euclidean</th>
<th>DTW</th>
<th>Mahalanobis</th>
<th>LMNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>c.-b.</td>
</tr>
<tr>
<td>50 words</td>
<td>0.37</td>
<td>0.31</td>
<td>0.32</td>
<td>—</td>
</tr>
<tr>
<td>Adiac</td>
<td>0.39</td>
<td>0.40</td>
<td>0.36</td>
<td>0.23</td>
</tr>
<tr>
<td>CBF</td>
<td>0.15</td>
<td>0.00</td>
<td>0.05</td>
<td>0.15</td>
</tr>
<tr>
<td>ECG</td>
<td>0.12</td>
<td>0.23</td>
<td>0.08</td>
<td>0.10</td>
</tr>
<tr>
<td>Fish</td>
<td>0.22</td>
<td>0.17</td>
<td>0.18</td>
<td>0.13</td>
</tr>
<tr>
<td>Face (all)</td>
<td>0.29</td>
<td>0.19</td>
<td>0.25</td>
<td>0.16</td>
</tr>
<tr>
<td>Face (four)</td>
<td>0.22</td>
<td>0.17</td>
<td>0.17</td>
<td>0.16</td>
</tr>
<tr>
<td>Gun-Point</td>
<td>0.09</td>
<td>0.09</td>
<td>0.11</td>
<td>0.05</td>
</tr>
<tr>
<td>Lighting-2</td>
<td>0.25</td>
<td>0.13</td>
<td>0.25</td>
<td>0.41</td>
</tr>
<tr>
<td>Lighting-7</td>
<td>0.42</td>
<td>0.27</td>
<td>0.23</td>
<td>0.51</td>
</tr>
<tr>
<td>OSU Leaf</td>
<td>0.48</td>
<td>0.41</td>
<td>0.46</td>
<td>0.57</td>
</tr>
<tr>
<td>OliveOil</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td>Swedish Leaf</td>
<td>0.21</td>
<td>0.21</td>
<td>0.18</td>
<td>0.21</td>
</tr>
<tr>
<td>Trace</td>
<td>0.24</td>
<td>0.00</td>
<td>0.07</td>
<td>0.20</td>
</tr>
<tr>
<td>Two Patterns</td>
<td>0.09</td>
<td>0.00</td>
<td>0.12</td>
<td>0.05</td>
</tr>
<tr>
<td>Synthetic Control</td>
<td>0.12</td>
<td>0.01</td>
<td>0.09</td>
<td>0.03</td>
</tr>
<tr>
<td>Yoga</td>
<td>0.17</td>
<td>0.16</td>
<td>0.17</td>
<td>0.18</td>
</tr>
<tr>
<td># of best errors</td>
<td>1</td>
<td>9</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

such as the DTW or LMNN. Regarding the 1-NN classification accuracy, we give the results in Table 2. As expected [10], no distance is better on all data sets. However, because the diagonal Mahalanobis distance is closely related to the Euclidean distance, we compare their classification accuracy. In two data sets, the Euclidean distance outperformed the class-based Mahalanobis distance and only by small differences (0.09 versus 0.12 and 0.21 versus 0.26). Meanwhile, the class-based Mahalanobis outperformed the Euclidean distance 11 times, and sometimes by large margins (0.07 versus 0.24 and 0.05 versus 0.15).

The LMNN is also competitive: its classification error is sometimes half that of the Euclidean distance. However, the class-based LMNN gets the best result among all methods only three times as opposed to six times for the global LMNN. Moreover, the global LMNN significantly outperforms the class-based LMNN on the Two Patterns data set (0.05 versus 0.24). Overall, the class-based LMNN is not an improvement over the global LMNN.

5.4. Effect of the number of instances per class

Whereas Table 2 shows that the Mahalanobis distance is far superior to the Euclidean distance on some data sets, this result is linked to the number of instances per class. For example, on the Wafer data set (which we removed), there are many instances per class (500), and correspondingly, all distances give a negligible classification error.
Thus, we considered three different synthetic time-series data sets with varying numbers of instances per class: Cylinder-Bell-Funnel (CBF) [24], Control Charts (CC) [22] and Waveform [3]. Test sets have 1 000 instances per class whereas training sets have between 10 to 1 000 instances. We repeated each test ten times, with different training sets. Fig. 1 shows that whereas the class-based diagonal Mahalanobis is superior to the Euclidean distance when there are few instances, this benefit is less significant as the number of instances increases. Indeed, the classification accuracy of the Euclidean distance grows closer to perfection and it becomes more difficult for alternatives to be far superior.

6. Conclusion

The Mahalanobis distance has received little attention for time series classification and we are not surprised given its poor performance as a 1-NN classifier. However, by learning one diagonal Mahalanobis distance per class we get a competitive classifier. Moreover, the diagonal Mahalanobis distance is particularly appealing computationally: we only need to compute the variances of the components. Our results are a validation of earlier work on class-based Mahalanobis distances [17, 19, 20]. Meanwhile, we get good results with the LMNN on time series data, though it is more expensive.

Acknowledgments

This work is supported by NSERC grant 261437.
References


M. Vlachos, D. Gunopoulos, G. Kollios, Discovering similar multidimensional trajectories, in: ICDE ’02, p. 673.

