applied sciences

MDPI

Article

Kernel Geometric Mean Metric Learning

Zixin Feng, Teligeng Yun, Yu Zhou *, Ruirui Zheng and Jianjun He

School of Information and Communication Engineering, Dalian Minzu University, Dalian 116600, China; zixinfeng126@yeah.net (Z.F.)
* Correspondence: yuzhou829@sina.com

Abstract: Geometric mean metric learning (GMML) algorithm is a novel metric learning approach proposed recently. It has many advantages such as unconstrained convex objective function, closed form solution, faster computational speed, and interpretability over other existing metric learning technologies. However, addressing the nonlinear problem is not effective enough. The kernel method is an effective method to solve nonlinear problems. Therefore, a kernel geometric mean metric learning (KGMML) algorithm is proposed. The basic idea is to transform the input space into a high-dimensional feature space through nonlinear transformation, and use the integral representation of the weighted geometric mean and the Woodbury matrix identity in new feature space to generalize the analytical solution obtained in the GMML algorithm as a form represented by a kernel matrix, and then the KGMML algorithm is obtained through operations. Experimental results on 15 datasets show that the proposed algorithm can effectively improve the accuracy of the GMML algorithm and other metric algorithms.

Keywords: metric learning; kernel methods; weighted geometric mean

Citation: Feng, Z.; Yun, T.; Zhou, Y.; Zheng, R.; He, J. Kernel Geometric Mean Metric Learning. *Appl. Sci.* **2023**, *13*, 12047. https://doi.org/10.3390/app132112047

Academic Editors: Emilio Soria-Olivas and José Salvador Sánchez Garreta

Received: 19 July 2023
Revised: 7 September 2023
Accepted: 9 October 2023
Published: 6 November 2023

1. Introduction

It is well known that the distance measure is one of the most commonly used measures to describe the similarity between samples. At present, various distance metrics have been proposed, such as the Euclidean distance and Mahalanobis distance. However, these distance metric expressions are fixed, i.e., there are non-adjustable parameters, which result in varying degrees of effectiveness in dealing with various problems. Thus, an effective distance metric is proposed by constructing learning from training samples. From the definition of the distance metric, it follows that any binary function $d(x, x')$ defined in the feature space is called a distance function, provided that the four conditions of symmetry, self-similarity, non-negativity, and trigonometric inequality are satisfied simultaneously. Thus, any binary function

$$d_M(x, x') = (x - x')^{\mathrm{T}} M (x - x'), \tag{1}$$

is a distance function determined by any symmetric positive definite (SPD) matrix M, where x, x' are two samples, and usually M is called a metric matrix. The purpose of metric learning is to use training samples to learn a metric matrix M such that the resulting distance function $d_M(x, x')$ can improve the performance of the learning algorithm or satisfy some application requirements. Thus, metric learning has wide applications in many fields, such as pattern recognition [1,2], data mining [3–5], information security [6,7], bioinformatics [8,9], and medical diagnosis [10–12].

Because of the wide application space, metric learning techniques have received a lot of attention, and many excellent algorithms have been proposed. Xing [13] first proposed a metric learning algorithm. The main idea of the algorithm is to learn a metric matrix so that the distance between similar pairs of samples is small and the distance between dissimilar pairs of samples is large. The algorithm can make the distribution of similar

pairs of samples more compact in the new metric space, while the distribution of dissimilar pairs is more discrete. The proposal of this algorithm marked the real development of metric learning, and many subsequent research works were inspired by this algorithm. Davis [14] proposed an information theoretic metric learning algorithm. The basic idea of the algorithm is to assume the existence of a priority metric matrix M_0, while ensuring that the distance between similar pairs of samples is less than a threshold and the distance between dissimilar pairs of samples is greater than a threshold. By minimizing the relative entropy between the multivariate Gaussian distributions corresponding to M and M_0, Wang [15] proposed an information geometry metric learning algorithm. The basic idea of the algorithm is to use the category labeling information of the training samples to construct a kernel matrix that can reflect the desired distance between samples. Construct an actual kernel matrix describing the realistic distance relationship between the samples using the eigenvectors of the samples as well as the metric matrix M. The metric matrix M is then solved by minimizing the distance between these two kernel matrices. Weinberger [16] proposed a metric learning algorithm based on the maximum margin. The basic idea is to define an interval function between similar samples of each sample and other classes of samples. The metric matrix is then solved by minimizing the distance between similar pairs of samples and maximizing the defined interval.

Geometric mean metric learning algorithm was proposed by Pourya [17] in 2016. The essence of most metric learning algorithms is to minimize the distance between similar pairs of samples rather than maximize the distance between similar pairs of samples. Therefore, the sign of the term corresponding to the dissimilar pair of samples in the objective function is usually negative, while the strategy of the geometric mean metric learning model is to use the inverse matrix of the metric matrix M to represent the distance between dissimilar pairs of samples. The advantage of this approach is that the sign of the item corresponding to the dissimilar pair of samples in the objective function becomes positive, which reduces the difficulty of solving the model. Its objective function is as follows:

$$\min_{M>0} \sum_{(x_{1i},x_{2i})\in D^+} d_M(x_{1i},x_{2i}) + \sum_{(\overline{x}_{1j},\overline{x}_{2j})\in D^-} d_{M^{-1}}(\overline{x}_{1j},\overline{x}_{2j}), \tag{2}$$

where the similar pair sample set D^+ and the dissimilar pair sample set D^- can be expressed as

$$\begin{cases} D^+ = \left\{(x_{1i},x_{2i}) \mid x_{1i},x_{2i} \in R^d \text{ are in the same class }, i = 1,2,\ldots,n\right\}, \\ D^- = \left\{(\overline{x}_{1j},\overline{x}_{2j}) \mid \overline{x}_{1j},\overline{x}_{2j} \in R^d \text{ are in different class }, j = 1,2,\ldots,m\right\}. \end{cases} \tag{3}$$

Although the GMML algorithm has some advantages, such as unconstrained convex objective function, closed-form solution and interpretability, and faster calculation speed [18], it is actually a linear learning method and does not work well to address nonlinear problems. Because kernel methods are the key technology for addressing nonlinear problems, kernel algorithms [19–23] have been proposed. In view of this, kernel geometric mean metric learning algorithm is proposed . The closed-form solution of GMML algorithm in the high-dimensional feature space is rephrased. Then, the solution is generalized as a form of the kernel matrix by using the integral representation of the weighted geometric mean and the Woodbury matrix identity, leading to the KGMML algorithm. It retains the advantages of the GMML algorithm, while effectively addressing nonlinear problems.

The structure of this paper is organized as follows: In Section 2, some lemmas of weighted geometric mean and Woodbury matrix identity are formulated. Then, in Section 3, the optimization problem is discussed, then the extension to the weighted geometric mean and solution are discussed. The setup of the experiment, the analysis of parameter sensitivity, and experimental results are given in Section 4. Finally, our conclusions are summarized in Section 5.

2. Preliminaries

Lemma 1 ([24]). *For any $t \in (0,1)$, A is $n \times n$ positive definite, B is $n \times n$ Hermitian, and the following equation holds:*

$$A \sharp_t B = \frac{2\sin(\pi t)}{\pi} A \int_{-1}^{1} (1-s)^{-t}(1+s)^{t-1}\Big((1-s)I + (1+s)B^{-1}A\Big)^{-1} ds,$$

which is the integral representation of the A, B weighted geometric mean, where I is the identity matrix, and $\sharp_t$ represents the weighted geometric mean with weight t

Lemma 2 ([25]). *If A is a $n \times n$ invertible matrix corrected by UCV, where U is the $n \times k$ matrix, C is the $k \times k$ matrix, and V is the $k \times n$ matrix, then the Woodbury matrix identity is*

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\Big(C^{-1} + VA^{-1}U\Big)^{-1}VA^{-1}.$$

Lemma 3 ([24]). *For any $t \in (0,1)$, A is $n \times n$ positive definite, B is $n \times n$ Hermitian, and the following equation holds:*

$$A \sharp_t B = A\Big(B^{-1}A\Big)^{-t}.$$

3. Main Results

3.1. Optimization Problem

The core of the kernel method is to transform the linearly inseparable point set in the low-dimensional space into the high-dimensional space through a nonlinear transformation $\Phi(\cdot)$, making it linearly separable. In this process, it is not necessary to know the specific analytical expression of the nonlinear transformation, but only know the inner product $\langle \Phi(x), \Phi(x') \rangle$ after the transformation of the two samples x, x', and $\langle \Phi(x), \Phi(x') \rangle$ can be represented by a kernel function, i.e., $\langle \Phi(x), \Phi(x') \rangle = k(x, x')$, where $k(x, x')$ is a kernel function. For ease of presentation, we first derive the model using $\Phi(\cdot)$ and then replace it with the kernel function to obtain the final kernel geometric mean metric learning algorithm. Let D_Φ^+ and D_Φ^- represent the similarity sample set D^+ and the non-similar pair sample set D^- mapped to the point set in the high-dimensional space $\mathcal{F}$, that is

$$\begin{cases} D_\Phi^+ = \{(\Phi(x_{1i}), \Phi(x_{2i})) \mid \Phi(x_{1i}), \Phi(x_{2i}) \in \mathcal{F} \text{ are in the same class}, i = 1, 2, \ldots, n\}, \\ D_\Phi^- = \{(\Phi(\overline{x}_{1j}), \Phi(\overline{x}_{2j})) \mid \Phi(\overline{x}_{1j}), \Phi(\overline{x}_{2j}) \in \mathcal{F} \text{ are in different class }, j = 1, 2, \ldots, m\}. \end{cases} \tag{4}$$

In high dimensional space, the objective function (2) can be written in the following form:

$$\begin{aligned} &\min_{M_\Phi > 0} \sum_{(\Phi(x_{1i}), \Phi(x_{2i})) \in D_\Phi^+} d_{M_\Phi}(\Phi(x_{1i}), \Phi(x_{2i})) + \sum_{(\Phi(\overline{x}_{1j}), \Phi(\overline{x}_{2j})) \in D_\Phi^-} d_{M_\Phi{}^{-1}}(\Phi(\overline{x}_{1j}), \Phi(\overline{x}_{2j})) \\ &= \min_{M_\Phi > 0} \sum_{(\Phi(x_{1i}), \Phi(x_{2i})) \in D_\Phi^+} (\Phi(x_{1i}) - \Phi(x_{2i}))^{\mathrm{T}} M_\Phi (\Phi(x_{1i}) - \Phi(x_{2i})) \\ &\quad + \sum_{(\Phi(\overline{x}_{1j}), \Phi(\overline{x}_{2j})) \in D_\Phi^-} \big(\Phi(\overline{x}_{1j}) - \Phi(\overline{x}_{2j})\big)^{\mathrm{T}} M_\Phi^{-1} \big(\Phi(\overline{x}_{1j}) - \Phi(\overline{x}_{2j})\big). \end{aligned} \tag{5}$$

where $d_{M_\Phi}(\Phi(x_{1i}), \Phi(x_{2i})) = (\Phi(x_{1i}) - \Phi(x_{2i}))^{\mathrm{T}} M_\Phi (\Phi(x_{1i}) - \Phi(x_{2i}))$ represents the distance metric in the high dimensional space and M_Φ is the metric matrix.

Rewriting the distance uses traces, Equation (5) can be turned into the following optimization problem:

$$\begin{aligned}
&\min_{M_\Phi>0} \sum_{(\Phi(x_{1i}),\Phi(x_{2i}))\in D_\Phi^+} \operatorname{tr}\Big(M_\Phi(\Phi(x_{1i})-\Phi(x_{2i}))(\Phi(x_{1i})-\Phi(x_{2i}))^{\mathrm{T}}\Big) \\
&\quad + \sum_{\left(\Phi(\bar{x}_{1j}),\Phi(\bar{x}_{2j})\right)\in D_\Phi^-} \operatorname{tr}\Big(M_\Phi^{-1}\left(\Phi(\bar{x}_{1j})-\Phi(\bar{x}_{2j})\right)\left(\Phi(\bar{x}_{1j})-\Phi(\bar{x}_{2j})\right)^{\mathrm{T}}\Big) \\
&= \min_{M_\Phi>0} \operatorname{tr}(M_\Phi S_\Phi) + \operatorname{tr}\Big(M_\Phi^{-1} D_\Phi\Big) \\
&\triangleq \min_{M_\Phi>0} h(M_\Phi),
\end{aligned} \tag{6}$$

where S_Φ , D_Φ are

$$\begin{aligned}
S_\Phi &= \sum_{(\Phi(x_{1i}),\Phi(x_{2i}))\in D_\Phi^+} (\Phi(x_{1i})-\Phi(x_{2i}))(\Phi(x_{1i})-\Phi(x_{2i}))^{\mathrm{T}}, \\
D_\Phi &= \sum_{\left(\Phi(\bar{x}_{1j}),\Phi(\bar{x}_{2j})\right)\in D_\Phi^-} \left(\Phi(\bar{x}_{1j})-\Phi(\bar{x}_{2j})\right)\left(\Phi(\bar{x}_{1j})-\Phi(\bar{x}_{2j})\right)^{\mathrm{T}}.
\end{aligned} \tag{7}$$

Differentiating $h(M_\Phi)$ with respect to M_Φ yields

$$\nabla h(M_\Phi) = S_\Phi - M_\Phi^{-1} D_\Phi M_\Phi^{-1}. \tag{8}$$

Then, $\nabla h(M)$ is set to 0, which implies that

$$M_\Phi S_\Phi M_\Phi = D_\Phi, \tag{9}$$

and it is clear that the above equation is a Riccati equation. Since S_Φ and D_Φ are usually positive definite matrices, Equation (9) has a unique positive solution which is the midpoint of the geodesic joining ${S_\Phi}^{-1}$ to D_Φ [26], that is

$$M_\Phi = {S_\Phi}^{-1} \sharp_{1/2} D_\Phi = {S_\Phi}^{-1/2}\Big({S_\Phi}^{1/2} D_\Phi {S_\Phi}^{1/2}\Big)^{1/2} {S_\Phi}^{-1/2}. \tag{10}$$

3.2. Extension to Weighted Geometric Mean and Solution

To incorporate the weighted geometric mean, it is necessary to take into account the determination of weights for the objective function. When assigning linear weights for ${S_\Phi}^{-1}$ and D_Φ, only the metric matrix M_Φ can be uniformly scaled by a constant factor. Consequently, it is illogical to assign linear weights to the two components in Equation (5). Nevertheless, by employing nonlinear weights derived from the SPD manifold Riemannian geometry, the weights can be transformed into trade-offs between the two terms. Thus, the Riemann distance δ_R is introduced, and the problem of finding the minimum value of $h(M_\Phi)$ is equivalent to solving the minimum value of the following optimization problem:

$$\min_{M_\Phi \succeq 0} \delta_R^2\Big(M_\Phi, {S_\Phi}^{-1}\Big) + \delta_R^2(M_\Phi, D_\Phi), \tag{11}$$

where the Riemannian distance between SPD matrices X and Y is denoted by $\delta_R^2(X,Y) = \left\|\log\left(Y^{-1/2} X Y^{-1/2}\right)\right\|_F$, and $\|.\|_F$ is the Frobenius norm of a matrix. A linear parameter $t \in [0,1]$ is introduced to trade off the relationship of the two terms in the formula:

$$\min_{M_\Phi \succeq 0} (1-t)\delta_R^2\Big(M_\Phi, {S_\Phi}^{-1}\Big) + t\delta_R^2(M_\Phi, D_\Phi) \triangleq \min_{M_\Phi>0} h_t(M_\Phi). \tag{12}$$

Because $h_t(M_\Phi)$ is still geodesically convex [17], Equation (12) has a unique positive solution, that is,

$$M_\Phi = {S_\Phi}^{-1} \sharp_t D_\Phi. \tag{13}$$

Theorem 1. *The solution* $M_\Phi = {S_\Phi}^{-1} \sharp_t D_\Phi$ *of the objective function (5) can be rewritten as*

$$M_\Phi = Q_\Phi \left({K_{PQ}}^{\mathrm{T}} K_{PQ} \right)^{t-1} {Q_\Phi}^{\mathrm{T}},$$

where the kernel matrix $K_{PQ} = {P_\Phi}^{\mathrm{T}} Q_\Phi = K_{P_1Q_1} - K_{P_1Q_2} - K_{P_2Q_1} + K_{P_2Q_2}$,

$$P_\Phi = [\Phi(x_{11})\Phi(x_{12}) \cdots \Phi(x_{1n})] - [\Phi(x_{21})\Phi(x_{22}) \cdots \Phi(x_{2n})] \triangleq P_{\Phi 1} - P_{\Phi 2},$$
$$Q_\Phi = [\Phi(\bar{x}_{11})\Phi(\bar{x}_{12}) \cdots \Phi(\bar{x}_{1m})] - [\Phi(\bar{x}_{21})\Phi(\bar{x}_{22}) \cdots \Phi(\bar{x}_{2m})] \triangleq Q_{\Phi 1} - Q_{\Phi 2}, \textit{and}$$

$$K_{P_1Q_1} = {P_{\Phi 1}}^{\mathrm{T}} Q_{\Phi 1} = \begin{bmatrix} \langle \Phi(x_{11}), \Phi(\bar{x}_{11}) \rangle & \cdots & \langle \Phi(x_{11}), \Phi(\bar{x}_{1m}) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi(x_{1n}), \Phi(\bar{x}_{11}) \rangle & \cdots & \langle \Phi(x_{1n}), \Phi(\bar{x}_{1m}) \rangle \end{bmatrix} = \begin{bmatrix} k(x_{11}, \bar{x}_{11}) & \cdots & k(x_{11}, \bar{x}_{1m}) \\ \vdots & \ddots & \vdots \\ k(x_{1n}, \bar{x}_{11}) & \cdots & k(x_{1n}, \bar{x}_{1m}) \end{bmatrix}$$

$$K_{P_1Q_2} = {P_{\Phi 1}}^{\mathrm{T}} Q_{\Phi 2} = \begin{bmatrix} \langle \Phi(x_{11}), \Phi(\bar{x}_{21}) \rangle & \cdots & \langle \Phi(x_{11}), \Phi(\bar{x}_{2m}) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi(x_{1n}), \Phi(\bar{x}_{21}) \rangle & \cdots & \langle \Phi(x_{1n}), \Phi(\bar{x}_{2m}) \rangle \end{bmatrix} = \begin{bmatrix} k(x_{11}, \bar{x}_{21}) & \cdots & k(x_{11}, \bar{x}_{2m}) \\ \vdots & \ddots & \vdots \\ k(x_{1n}, \bar{x}_{21}) & \cdots & k(x_{1n}, \bar{x}_{2m}) \end{bmatrix}$$

$$K_{P_2Q_1} = {P_{\Phi 2}}^{\mathrm{T}} Q_{\Phi 1} = \begin{bmatrix} \langle \Phi(x_{21}), \Phi(\bar{x}_{11}) \rangle & \cdots & \langle \Phi(x_{21}), \Phi(\bar{x}_{1m}) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi(x_{2n}), \Phi(\bar{x}_{11}) \rangle & \cdots & \langle \Phi(x_{2n}), \Phi(\bar{x}_{1m}) \rangle \end{bmatrix} = \begin{bmatrix} k(x_{21}, \bar{x}_{11}) & \cdots & k(x_{21}, \bar{x}_{1m}) \\ \vdots & \ddots & \vdots \\ k(x_{2n}, \bar{x}_{11}) & \cdots & k(x_{2n}, \bar{x}_{1m}) \end{bmatrix}$$

$$K_{P_2Q_2} = {P_{\Phi 2}}^{\mathrm{T}} Q_{\Phi 2} = \begin{bmatrix} \langle \Phi(x_{21}), \Phi(\bar{x}_{21}) \rangle & \cdots & \langle \Phi(x_{21}), \Phi(\bar{x}_{2m}) \rangle \\ \vdots & \ddots & \vdots \\ \langle \Phi(x_{2n}), \Phi(\bar{x}_{21}) \rangle & \cdots & \langle \Phi(x_{2n}), \Phi(\bar{x}_{2m}) \rangle \end{bmatrix} = \begin{bmatrix} k(x_{21}, \bar{x}_{21}) & \cdots & k(x_{21}, \bar{x}_{2m}) \\ \vdots & \ddots & \vdots \\ k(x_{2n}, \bar{x}_{21}) & \cdots & k(x_{2n}, \bar{x}_{2m}) \end{bmatrix}.$$

Proof. According to Lemma 1,

$$\begin{aligned} M_\Phi &= S_\Phi^{-1} \sharp_t D_\Phi \\ &= \frac{2sin(\pi t)}{\pi} S_\Phi^{-1} \int_{-1}^{1} (1-s)^{-t} (1+s)^{t-1} ((1-s)I + (1+s) D_\Phi^{-1} S_\Phi^{-1})^{-1} \mathrm{d}s \\ &= \frac{2sin(\pi t)}{\pi} \int_{-1}^{1} (1-s)^{-t} (1+s)^{t-1} ((1-s)S_\Phi + (1+s) D_\Phi^{-1})^{-1} \mathrm{d}s \\ &= \frac{2sin(\pi t)}{\pi} \int_{-1}^{1} (1-s)^{-t} (1+s)^{t-1} ((1-s)S_\Phi + (1+s) D_\Phi^{-1})^{-1} \mathrm{d}s. \end{aligned} \tag{14}$$

According to the definitions of P_Φ and Q_Φ, S_Φ and D_Φ can be written as $S_\Phi = P_\Phi P_\Phi^{\mathrm{T}}$, $D_\Phi = Q_\Phi Q_\Phi^{\mathrm{T}}$. Substituting them into Equation (14), it is clear that

$$M_\Phi = \frac{2sin(\pi t)}{\pi} \int_{-1}^{1} (1-s)^{-t} (1+s)^{t-1} ((1-s) P_\Phi P_\Phi^{\mathrm{T}} + (1+s)(Q_\Phi Q_\Phi^{\mathrm{T}})^{-1})^{-1} \mathrm{d}s. \tag{15}$$

For convenience in the following discussion, we introduce the notation

$$G = \left((1-s) P_\Phi P_\Phi^{\mathrm{T}} + (1+s) \left(Q_\Phi Q_\Phi^{\mathrm{T}} \right)^{-1} \right)^{-1}.$$

From Lemma 2, it follows that

$$G = (A + UCV)^{-1} = A^{-1} - A^{-1} U \left(C^{-1} + V A^{-1} U \right)^{-1} V A^{-1}, \tag{16}$$

where $A = (1+s)\left(Q_\Phi Q_\Phi^{\mathrm{T}}\right)^{-1}$, $U = P_\Phi$, $C = (1-s)I$, $V = P_\Phi^{\mathrm{T}}$. Then, taking it into Equation (15), one has

$$\begin{aligned}
M_\Phi &= \frac{2sin(\pi t)}{\pi}\int_{-1}^{1}(1-s)^{-t}(1+s)^{t-1}[\frac{Q_\Phi Q_\Phi^{\mathrm{T}}}{1+s} - \frac{Q_\Phi Q_\Phi^{\mathrm{T}}}{1+s}\cdot P_\Phi\left(\frac{I^{-1}}{1-s} + P_\Phi^{\mathrm{T}}\cdot\frac{Q_\Phi Q_\Phi^{\mathrm{T}}}{1+s}\cdot P_\Phi\right)^{-1} P_\Phi^{\mathrm{T}}\frac{Q_\Phi Q_\Phi^{\mathrm{T}}}{1+s}]\mathrm{d}s \\
&= \frac{2sin(\pi t)}{\pi}\int_{-1}^{1}(1-s)^{-t}(1+s)^{t-1}[\frac{Q_\Phi Q_\Phi^{\mathrm{T}}}{1+s} - \frac{Q_\Phi}{1+s}\cdot {K_{PQ}}^{\mathrm{T}}\left(\frac{I^{-1}}{1-s} + \frac{K_{PQ}{K_{PQ}}^{\mathrm{T}}}{1+s}\right)^{-1}\frac{K_{PQ}{Q_\Phi}^{\mathrm{T}}}{1+s}]\mathrm{d}s \\
&= \frac{2sin(\pi t)}{\pi}\int_{-1}^{1}(1-s)^{-t}(1+s)^{t-1}Q_\Phi[\frac{I}{1+s} - \frac{{K_{PQ}}^{\mathrm{T}}}{1+s}\left(\frac{I^{-1}}{1-s} + \frac{K_{PQ}{K_{PQ}}^{\mathrm{T}}}{1+s}\right)^{-1}\frac{K_{PQ}}{1+s}]Q_\Phi^{\mathrm{T}}\mathrm{d}s \\
&= \frac{2sin(\pi t)}{\pi}\int_{-1}^{1}(1-s)^{-t}(1+s)^{t-1}Q_\Phi((1+x)I + (1-x){K_{PQ}}^{\mathrm{T}}K_{PQ})^{-1}Q_\Phi^{\mathrm{T}}\mathrm{d}s \\
&= Q_\Phi\frac{2sin(\pi t)}{\pi}\int_{-1}^{1}(1-s)^{-t}(1+s)^{t-1}\left((1+s)\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right)^{-1}\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right) + (1-s)\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right)\right)^{-1}\mathrm{d}sQ_\Phi^{\mathrm{T}} \\
&= Q_\Phi\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right)^{-1}\frac{2sin(\pi t)}{\pi}\int_{-1}^{1}(1-s)^{-t}(1+s)^{t-1}\left((1-s)I + (1+s)\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right)^{-1}\right)^{-1}\mathrm{d}sQ_\Phi^{\mathrm{T}} \\
&= Q_\Phi[\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right)^{-1}\sharp_t I]Q_\Phi^{\mathrm{T}}.
\end{aligned} \tag{17}$$

From Lemma 3, it follows that

$$\begin{aligned}
M_\Phi =& Q_\Phi\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right)^{-1}\left(\left(I\cdot{K_{PQ}}^{\mathrm{T}}K_{PQ}\right)^{-1}\right)^{-t}Q_\Phi^{\mathrm{T}} \\
=& Q_\Phi\left({K_{PQ}}^{\mathrm{T}}K_{PQ}\right)^{t-1}{Q_\Phi}^{\mathrm{T}}.
\end{aligned} \tag{18}$$

□

Theorem 2. *The distance* $d_{M_\Phi}(x,x') = (\Phi(x)-\Phi(x'))^{\mathrm{T}}M_\Phi(\Phi(x)-\Phi(x'))$ *of any two sample* x *and* x' *can be rewritten as*

$$d_{M_\Phi}(x,x') = (K_{xQ} - K_{x'Q})\left({K_{PQ}}^{T}K_{PQ}\right)^{t-1}({K_{xQ}}^{\mathrm{T}} - {K_{x'Q}}^{\mathrm{T}}).$$

where $K_{xQ} = \Phi(x)^{\mathrm{T}}Q_\Phi = K_{xQ_1} - K_{xQ_2}$, $K_{x'Q} = \Phi(x')^{\mathrm{T}}Q_\Phi = K_{x'Q_1} - K_{x'Q_2}$, *and*

$$K_{xQ_1} = \Phi(x)^{\mathrm{T}}Q_{\Phi 1} = \begin{bmatrix} \langle\Phi(x_1),\Phi(\bar{x}_{11})\rangle & \cdots & \langle\Phi(x_1),\Phi(\bar{x}_{1n})\rangle \\ \vdots & \ddots & \vdots \\ \langle\Phi(x_n),\Phi(\bar{x}_{11})\rangle & \cdots & \langle\Phi(x_n),\Phi(\bar{x}_{1n})\rangle \end{bmatrix} = \begin{bmatrix} k(x_1,\bar{x}_{11}) & \cdots & k(x_1,\bar{x}_{1n}) \\ \vdots & \ddots & \vdots \\ k(x_n,\bar{x}_{11}) & \cdots & k(x_n,\bar{x}_{1n}) \end{bmatrix}$$

$$K_{xQ_2} = \Phi(x)^{\mathrm{T}}Q_{\Phi 2} = \begin{bmatrix} \langle\Phi(x_1),\Phi(\bar{x}_{21})\rangle & \cdots & \langle\Phi(x_1),\Phi(\bar{x}_{2n})\rangle \\ \vdots & \ddots & \vdots \\ \langle\Phi(x_n),\Phi(\bar{x}_{21})\rangle & \cdots & \langle\Phi(x_n),\Phi(\bar{x}_{2n})\rangle \end{bmatrix} = \begin{bmatrix} k(x_1,\bar{x}_{21}) & \cdots & k(x_1,\bar{x}_{2n}) \\ \vdots & \ddots & \vdots \\ k(x_n,\bar{x}_{21}) & \cdots & k(x_n,\bar{x}_{2n}) \end{bmatrix}$$

$$K_{x'Q_1} = \Phi(x')^{\mathrm{T}}Q_{\Phi 1} = \begin{bmatrix} \langle\Phi(x'_1),\Phi(\bar{x}_{11})\rangle & \cdots & \langle\Phi(x'_1),\Phi(\bar{x}_{1n})\rangle \\ \vdots & \ddots & \vdots \\ \langle\Phi(x'_n),\Phi(\bar{x}_{11})\rangle & \cdots & \langle\Phi(x'_n),\Phi(\bar{x}_{1n})\rangle \end{bmatrix} = \begin{bmatrix} k(x'_1,\bar{x}_{11}) & \cdots & k(x'_1,\bar{x}_{1n}) \\ \vdots & \ddots & \vdots \\ k(x'_n,\bar{x}_{11}) & \cdots & k(x'_n,\bar{x}_{1n}) \end{bmatrix}$$

$$K_{x'Q_2} = \Phi(x')^{\mathrm{T}}Q_{\Phi 2} = \begin{bmatrix} \langle\Phi(x'_1),\Phi(\bar{x}_{21})\rangle & \cdots & \langle\Phi(x'_1),\Phi(\bar{x}_{2n})\rangle \\ \vdots & \ddots & \vdots \\ \langle\Phi(x'_n),\Phi(\bar{x}_{21})\rangle & \cdots & \langle\Phi(x'_n),\Phi(\bar{x}_{2n})\rangle \end{bmatrix} = \begin{bmatrix} k(x'_1,\bar{x}_{21}) & \cdots & k(x'_1,\bar{x}_{2n}) \\ \vdots & \ddots & \vdots \\ k(x'_n,\bar{x}_{21}) & \cdots & k(x'_n,\bar{x}_{2n}) \end{bmatrix}.$$

Proof. According to Theorem 1,

$$\begin{aligned} d_{M_\Phi}(x,x') &= (\Phi(x)-\Phi(x'))^{\mathrm{T}} Q_\Phi \left(K_{PQ}{}^{\mathrm{T}} K_{PQ}\right)^{t-1} Q_\Phi^{\mathrm{T}} (\Phi(x)-\Phi(x')) \\ &= (\Phi(x)-\Phi(x'))^{\mathrm{T}} (Q_{\Phi 1}-Q_{\Phi 2}) \left(K_{PQ}{}^{\mathrm{T}} K_{PQ}\right)^{t-1} (Q_{\Phi 1}-Q_{\Phi 2})^{\mathrm{T}} (\Phi(x)-\Phi(x')) \\ &= (K_{xQ_1}-K_{xQ_2}-K_{x'Q_1}+K_{x'Q_2}) \left(K_{PQ}{}^{\mathrm{T}} K_{PQ}\right)^{t-1} (K_{xQ_1}{}^{\mathrm{T}}-K_{xQ_2}{}^{\mathrm{T}}-K_{x'Q_1}{}^{\mathrm{T}}+K_{x'Q_2}{}^{\mathrm{T}}) \\ &= (K_{xQ}-K_{x'Q}) \left(K_{PQ}{}^{\mathrm{T}} K_{PQ}\right)^{t-1} (K_{xQ}{}^{\mathrm{T}}-K_{x'Q}{}^{\mathrm{T}}) \end{aligned} \tag{19}$$

□

Equation (19) can be taken as the final distance metric of the kernel geometric mean metric learning algorithm. The algorithm is summarized in Algorithm 1.

Algorithm 1 Kernel geometric mean metric learning algorithm.

Input: Similar pair sample set D^+, dissimilar pair sample set D^-, and two samples x, x'.

Parameter: t: $t \in [0,1]$, the weight coefficient in Equation (13), p: kernel parameters .

Output: $d_{M_\Phi}(x,x')$ the distance learned for KGMML,

Step 1. Compute kernel matrices $K_{P_1Q_1}$, $K_{P_1Q_2}$, $K_{P_2Q_1}$, $K_{P_2Q_2}$, and K_{PQ} according to Theorem 1 .

Step 2. Compute kernel matrices K_{xQ} and $K_{x'Q}$ according to Theorem 2.

Step 3. Compute the Schur (spectral) decomposition of $\left(K_{PQ}{}^{\mathrm{T}} K_{PQ}\right)^{t-1} = U\Lambda^{t-1}U = A$, where U denotes the eigenvalue of $K_{PQ}{}^{\mathrm{T}} K_{PQ}$, and Λ denotes the eigenvector of $K_{PQ}{}^{\mathrm{T}} K_{PQ}$.

Step 4. Compute $d_{M_\Phi}(x,x') = (K_{xQ}-K_{x'Q})A(K_{xQ}{}^{\mathrm{T}}-K_{x'Q}{}^{\mathrm{T}})$.

4. Experiment

4.1. Experimental Setup

To verify the effectiveness of Algorithm 1, simulation experiments were conducted on 15 UCI [27] datasets, where the basic information is shown in Table 1. We divided the dataset into large and small datasets according to the size of the sample size (1–11 were small datasets, and 12–15 were large datasets; a sample size greater than 3000 is considered a large dataset, and a sample size of less than 3000 is considered a small dataset).

Table 1. Characteristics of experimental datasets.

	Datasets	Of Features	Of Instances	Of Classes
1	Pima	8	768	2
2	Vehicle	18	846	4
3	German	24	1000	2
4	Segment	18	2310	7
5	Heart-Disease	13	270	2
6	Lymphgraphy	18	148	4
7	Liver-Disorders	6	345	2
8	Hayes-Roth	4	160	3
9	Ionosphere	34	351	2
10	Glasses	9	214	6
11	Balance-Scale	4	625	3
12	Usps	256	9298	10
13	Mnist	784	4000	10
14	DNA	180	3186	2
15	Spambase	57	4601	2

Next, some efficient algorithms are described for distance metric learning, and the proposed method is compared with existing excellent classical algorithms. The specific introduction is given in Table 2. These algorithms are all from the original author's homepage. The kernel function used in the experiment is the Gaussian kernel function. For

the KGMML, the settings of t and p are given in detail in the next subsection. In this paper, 5-fold cross-validation is repeated 10 times to measure the average accuracy. Moreover, as a preprocessing step, each feature of the experimental dataset is scaled to the one with zero mean and unit variance.

Table 2. Brief description of the distance metric learning method used in this paper.

	Name	Description
1	Euclidean	The Euclidean distance metric [28].
2	DMLMJ	Distance metric learning through maximization of the Jeffrey divergence [28].
3	LMNN	Large margin nearest neighbor classification [16].
4	GB-LMNN	Non-linear transformations with gradient boosting [29].
5	GMML	Geometric mean metric learning [17].
6	Low-rank	Low-rank geometric mean metric learning [30].
7	KGMML	The kernelized version of GMML

4.2. Parameter Sensitivity Analysis

To determine the impact of these two parameters on the KGMML algorithm, the experiments are conducted on five datasets selected from 15 datasets. Five-fold cross validation is used to choose the best t-value, and the two-step method is used to test different t-values. The above approach is used to find the best t-value, and its precision can be verified in Figures 1 and 2. Firstly, obtain the optimal t in the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and the result is shown in Figure 1. Secondly, test 5 t-values using intervals with a step size of 0.02. As shown in Figure 2, the variation of accuracy in the test interval with a step size of 0.02 is not significant, so we choose the middle t-value of 0.05. When the p-value is 10, the precision has an inflection point in Figure 3, and the precision is higher at the inflection point. Thus, the p-value is chosen as 10. Vary t in the set $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, and p is fixed.

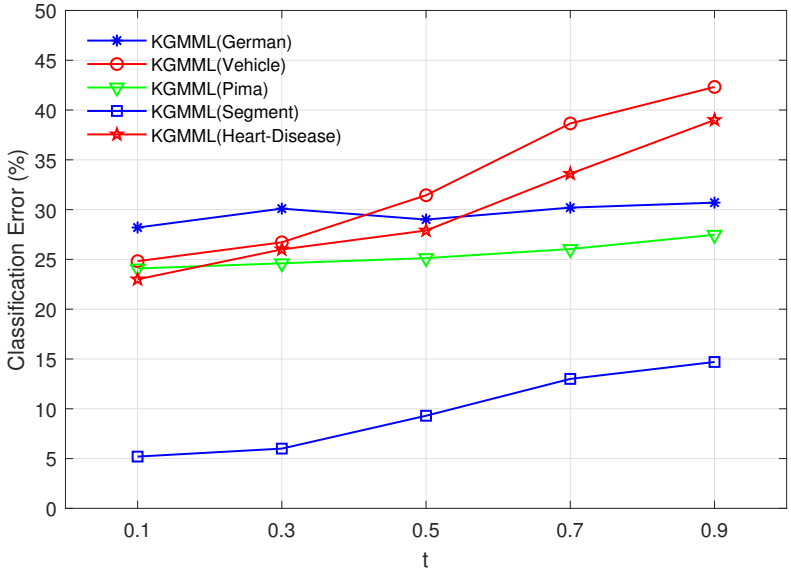

Figure 1. Varying t (p is fixed).

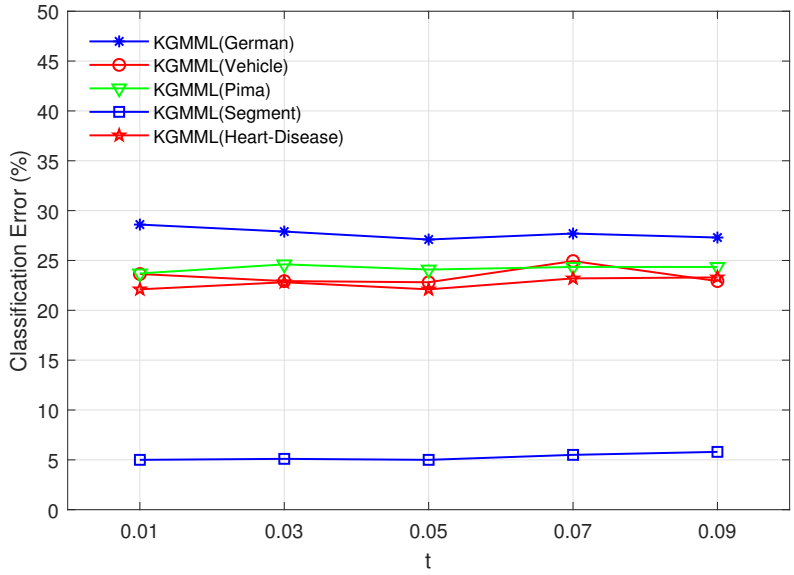

Figure 2. Varying t (p is fixed).

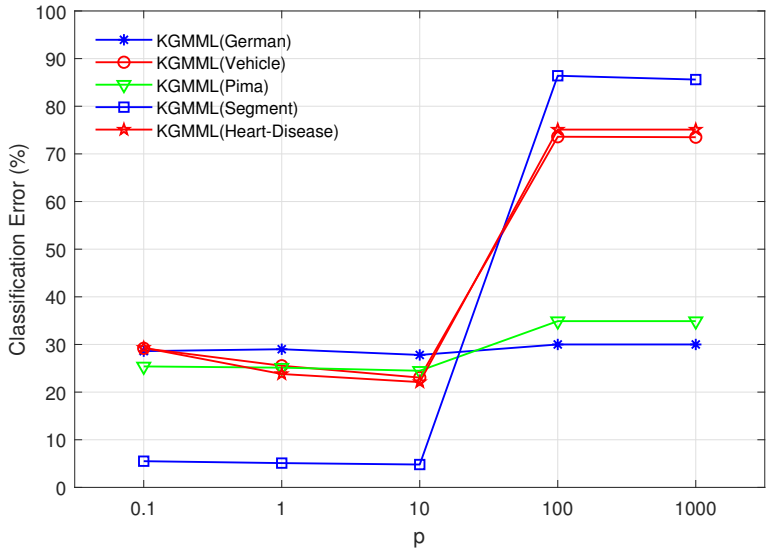

Figure 3. Varying p (t is fixed).

4.3. Experimental Results

The accuracy of each compared algorithm on 15 datasets is shown in Table 3, where the best result on each dataset is shown in boldface. Compared with the other six algorithms, our algorithm achieves the highest accuracy on the eight datasets of pima, vehicle, german, segment, usps, mnist, lymphgraphy and spambase. From the experimental results, we conclude that our proposed KGMML algorithm is more accurate than the GMML algorithm on all 15 datasets. The accuracy of the KGMML algorithm can be improved by roughly 2% to 4% over the GMML algorithm in small datasets, and the improvement is generally smaller for large datasets. In the small dataset, the Hayes-Roth dataset improves accuracy by 18.17%; in the large dataset, the Spambase dataset improves accuracy by 8.13%. The analysis reveals that the accuracy improvement is greater when the number of features is relatively small in both large and small datasets. Table 4 lists the average running time of the GMML algorithm and the KGMML algorithm on the dataset, which shows that the proposed algorithm does not drag down the running efficiency of the original algorithm. All experimental methods are implemented on MATLABR2018b (64-bit), and the simulations are run on a laptop with an Intel Core i5 (2.5 GHz) processor.

Table 3. Classification errors on UCI dataset. The best result is highlighted in boldface.

	Datasets	GMML	DMLMJ	LMNN	GB-LMNN	Euclidean	Low-Rank	KGMML
1	Pima	27.66	30.18	33.82	37.14	27.27	29.58	**25.17**
2	Vehicle	22.09	25.75	46.53	41.24	33.53	41.32	**21.21**
3	German	27.41	24.79	30.50	29.32	31.53	26.96	**24.40**
4	Segment	4.13	3.64	5.19	4.55	6.93	5.59	**3.22**
5	Heart-Disease	20.82	19.73	31.52	**18.51**	33.33	22.52	18.97
6	Lymphography	56.88	73.62	70.76	60.21	75.13	57.57	**53.75**
7	Liver-Disorders	35.00	**30.05**	30.46	34.88	31.88	41.86	30.17
8	Hayes-Roth	37.69	16.84	31.33	31.35	**16.67**	39.55	19.52
9	Ionosphere	15.34	11.27	5.71	4.29	**1.43**	17.26	11.54
10	Glasses	36.96	33.08	30.20	**23.30**	30.23	33.64	32.47
11	Balance-Scale	12.84	**8.62**	12.80	15.20	14.40	12.31	9.19
12	Usps	3.72	2.88	33.11	10.60	10.55	4.08	**2.82**
13	Mnist	9.65	16.44	86.80	82.62	17.12	75.34	**8.32**
14	DNA	23.65	**21.75**	22.32	22.84	27.63	26.24	23.05
15	Spambase	19.33	18.27	38.60	15.80	16.09	11.43	**11.20**

To show the performance advantages of the KGMML algorithm, a score statistic is performed on the KGMML algorithm and other classic algorithms. The scoring process is as follows. Assuming that A is the result of using the KGMML algorithm on a certain dataset, and B is the result of uniting the KGMML algorithm on this dataset (that is, using other algorithms), firstly, comparative analysis of A and B is performed by using a 5% significance level t test. In a statistical sense, if $A > B$, it is considered that the result A obtained by using the KGMML algorithm on this dataset wins the result B using other algorithms. Thus, the statistical result of the score on this dataset is recorded as "1/0/0". If $A < B$, the score statistics result is recorded as "0/0/1". If $A = B$, it means that A and B are the same in the statistical sense. Thus, it is considered that they are tied and recorded as "0/1/0". It is evident that the notation "5/0/0" signifies that the outcomes achieved by the KGMML algorithm outperform those of other algorithms across five datasets. On each dataset and the overall score, the statistical results of the KGMML algorithm are listed in Table 4.

Table 4. Win/tie/loss counts of each algorithm between the results obtained the KGMML algorithm and other classical algorithms.

Datasets	KGMML					
	GMML	DMLMJ	LMNN	GB-LMNN	Euclidean	Low-Rank
Pima	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
Vehicle	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
German	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0
Segment	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0
Usps	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0
Mnist	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
Glasses	1/0/0	0/1/0	0/0/1	0/0/1	0/1/0	1/0/0
DNA	0/1/0	0/0/1	0/0/1	0/1/0	1/0/0	1/0/0
Heart-Disease	1/0/0	0/1/0	1/0/0	0/1/0	1/0/0	1/0/0
Lymphography	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0
Liver-Disorders	1/0/0	0/1/0	0/1/0	1/0/0	0/1/0	1/0/0
Hayes-Roth	1/0/0	0/0/1	1/0/0	1/0/0	0/0/1	1/0/0
Ionosphere	1/0/0	0/1/0	0/0/1	0/0/1	0/0/1	1/0/0
Spambase	1/0/0	1/0/0	1/0/0	1/0/0	1/0/0	0/1/0
Balance	1/0/0	0/1/0	1/0/0	1/0/0	1/0/0	1/0/0
Total	12/3/0	5/8/2	11/1/3	9/2/2	11/2/2	14/1/0

5. Conclusions

In this paper, kernel geometric mean metric learning is proposed for the nonlinear distance metric with the introduction of a kernel function. Traditional metric learning methods aim to learn global linear metrics; for most data, there are nonlinear relationships, so traditional methods are not suitable for nonlinear problems. The experimental results on the UCI dataset show that the algorithm can effectively improve the accuracy of the GMML algorithm. Especially for small datasets, the highest can reach more than 40%, and the proposed algorithm can address nonlinear problems.

In practice, only a small amount of data is tagged, while most of the data remains untagged. Therefore, the question of how to make the best use of this large amount of untagged data prompted us to think. Under the partial marker learning framework, the real class markers of the training samples are usually hidden in a set of candidate markers and are no longer explicit, which makes the construction of learning algorithms more difficult than the traditional classification problems, and the research results are still less available. In future work, we will try to improve the inaccurate similarity pair problem existing in the kernel-based geometric mean metric learning algorithm. Using the partial labeling metric learning algorithm proposed in recent years, a partial labeling algorithm based on kernel geometric mean metric learning will be proposed.

Author Contributions: Conceptualization, J.H.; methodology, J.H.; software, T.Y.; validation, T.Y.; formal analysis, Z.F.; investigation, Y.Z.; resources, Y.Z.; data curation, T.Y.; writing—original draft preparation, Z.F.; writing—review and editing, Z.F.; visualization, Y.Z.; supervision, Y.Z.; project administration, R.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (62102062), the Humanities and Social Science Research Project of Ministry of Education(21YJCZH037), the Natural Science Foundation of Liaoning Province (2020-MS-134, 2020-MZLH-29).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors would like to thank the School of Information and Communication Engineering, Dalian Minzu University for assistance with simulation verifications related to this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, J.; Wang, R.; Mian, A.; Ajay, K.; Sudeep, S. Distance metric learning for pattern recognition. *Pattern Recognit.* **2018**, *75*, 1–3. [CrossRef]
2. Wei, Z.; Cui, Y.; Zhou, X.; Yang, W.; Li, Y.; Yi, X.; Dai, H. A research on metric learning in computer vision and pattern recognition. In Proceedings of the 2018 Tenth International Conference on Advanced Computational Intelligence, Xiamen, China, 29–31 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 254–259.
3. Yan, Y.; Xia, J.; Sun, D.; Hu, Q. Research on combination evaluation of operational stability of energy industry innovation ecosystem based on machine learning and data mining algorithms. *Energy Rep.* **2022**, *8*, 4641–4648. [CrossRef]
4. Wang, F.; Sun, J. Survey on distance metric learning and dimensionality reduction in data mining. *Data Min. Knowl. Discov.* **2015**, *29*, 534–564. [CrossRef]
5. Yan, M.; Zhang, Y.; Wang, H. Tree-Based Metric Learning for Distance Computation in Data Mining. In *Asia-Pacific Web Conference, Proceedings of the 17th Asia-Pacific Web Conference, APWeb 2015, Guangzhou, China, 18–20 September 2015*; Springer International Publishing: Cham, Switzerland, 2015; pp. 377–388.
6. Mojisola, F.O.; Misra, S.; Febisola, C.F.; Abayomi-Alli, O.; Sengul, G. An improved random bit-stuffing technique with a modified RSA algorithm for resisting attacks in information security. *Egypt. Inform. J.* **2022**, *23*, 291–301. [CrossRef]
7. Kraeva, I.; Yakhyaeva, G. Application of the metric learning for security incident playbook recommendation. In Proceedings of the 2021 IEEE 22nd International Conference of Young Professionals in Electron Devices and Materials, Souzga, Russia, 30 June–4 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 475–479.
8. Bennett, J.; Pomaznoy, M.; Singhania, A.; Peters, B. A metric for evaluating biological information in gene sets and its application to identify co-expressed gene clusters in PBMC. *PLoS Comput. Biol.* **2021**, *17*, e1009459. [CrossRef]
9. Makrodimitris, S.; Reinders, M.J.T.; Van Ham, R.C.H.J. Metric learning on expression data for gene function prediction. *Bioinformatics* **2020**, *36*, 1182–1190. [CrossRef] [PubMed]

10. Yuan, T.; Dong, L.; Liu, B.; Huang, J.; Xiao, C. Deep Metric Learning by Exploring Confusing Triplet Embeddings for COVID-19 Medical Images Diagnosis. In Proceedings of the Workshop on Healthcare AI and COVID-19, Baltimore, MA, USA, 22 July 2022; pp. 1–10.
11. Jin, Y.; Lu, H.; Li, Z.; Wang, Y. A cross-modal deep metric learning model for disease diagnosis based on chest X-ray images. *Multimed. Tools Appl.* **2023**, *82*, 33421–33442. [CrossRef] [PubMed]
12. Xing, Y.; Meyer, B.J.; Harandi, M.; Drummond, T.; Ge, Z. Multimorbidity Content-Based Medical Image Retrieval and Disease Recognition Using Multi-label Proxy Metric Learning. *IEEE Access* **2023**, *11*, 50165–50179. [CrossRef]
13. Xing, E.; Ng, A.Y.; Jordan, M.; Russell, S.J. Distance metric learning with application to clustering with side-information. *Adv. Neural Inf. Process. Syst.* **2002**, *15*, 1–8.
14. Davis, J.V.; Kulis, B.; Jain, P.; Dhillon, I.S. Information-theoretic metric learning. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 209–216.
15. Wang, S.; Jin, R. An information geometry approach for distance metric learning. In Proceedings of the Artificial Intelligence and Statistics, Clearwater Beach, FL, USA, 16–18 April 2009; 2009; pp. 591–598.
16. Weinberger, K.Q.; Saul, L.K. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.* **2009**, *10*, 207–244.
17. Zadeh, P.; Hosseini, R.; Sra, S. Geometric mean metric learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016; pp. 2464–2471.
18. Zhou, Y.; Gu, H. Geometric mean metric learning for partial label data. *Neurocomputing* **2018**, *275*, 394–402. [CrossRef]
19. Mika, S.; Ratsch, G.; Weston, J.; Scholkopf, B.; Mullers, K.R. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (cat. no. 98th8468), Madison, WI, USA, 25 August 1999*; IEEE: New York, NY, USA, 1999; pp. 41–48.
20. Li, Z.; Kruger, U.; Xie, L.; Almansoori, A.; Su, H. Adaptive KPCA modeling of nonlinear systems. *IEEE Trans. Signal Process.* **2015**, *63*, 2364–2376. [CrossRef]
21. Lee, J.M.; Qin, S.J.; Lee, I.B. Fault detection of non-linear processes using kernel independent component analysis. *Can. J. Chem. Eng.* **2007**, *85*, 526–536. [CrossRef]
22. Zhang, L.; Zhou, W.D.; Jiao, L.C. Kernel clustering algorithm. *Chin. J. Comput.-Chin. Ed.* **2002**, *25*, 587–590.
23. Choi, H.; Choi, S. Kernel isomap. *Electron. Lett.* **2004**, *40*, 1612–1613. [CrossRef]
24. Fasi, M.; Iannazzo, B. Computing the weighted geometric mean of two large-scale matrices and its inverse times a vector. *SIAM J. Matrix Anal. Appl.* **2018**, *39*, 178–203. [CrossRef]
25. Higham, N. Accuracy and Stability of Numerical Algorithms. *SIAM* **2002**, *258*. Available online: http://en.wikipedia.org/wiki/Woodburymatrixidentity (accessed on 8 October 2023).
26. Bhatia, R. *Positive Definite Matrices*; Princeton University Press: Princeton, NJ, USA, 2009.
27. Asuncion, A.; Newman, D. UCI Machine Learning Repository. 2007. Available online: https://archive.ics.uci.edu/ (accessed on 8 October 2023).
28. Nguyen, B.; Morell, C.; De Baets, B. Supervised distance metric learning through maximization of the Jeffrey divergence. *Pattern Recognit.* **2017**, *64*, 215–225. [CrossRef]
29. Kedem, D.; Tyree, S.; Sha, F.; Lanckriet, G.; Weinberger, K.Q. Non-linear metric learning. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 2582–2590.
30. Bhutani, M.; Jawanpuria, P.; Kasai, H.; Mishra, B. Low-rank geometric mean metric learning. *arXiv* **2018**, arXiv:1806.05454.