Feature Extraction by Linear Embedding for one-class classification

Jong-Phil Sim, Song-Chun Pang *, Son-Myong Hwang

Faculty of Information Science, Kim Il Sung University, Pyongyang, DPRK

***Song-Chun Pang,** E-mail: bsc197842@star-co.net.kp

Abstract: In this paper, we mainly propose feature extraction algorithm by linear embedding from the outside new data. The formulation of this algorithm aims at minimizing pairwise distances of feature points. To enhance the performance of nonlinear feature learning, we also incorporate the neighborhood reconstruction error to preserve local topology structures. To enhance our algorithm to extract local features from the outside new data, we also add a feature approximation error that correlates features with embedded features by the jointly learnt feature extractor. Thus, the learnt linear extractor can extract the local features from the new data efficiently by direct embedding. To optimize the proposed objective function, we use Eigen-decomposition. Extensive simulation results verify the effectiveness of our algorithm, compared with other related feature learning techniques.

Keywords: Feature extraction, linear embedding, one-class classification

1. Introduction

One-class classification(OCC) describes training data from a single class (called “target class”) as a normalcy model and aims to detect data from any other class (called “outlier class”) as outliers. In the fields of OCC, feature extraction of one-class is always a fundamental problem, since it directly affects the performance of the subsequent pattern recognition and data mining models. Note that real-world data lies in high dimensional input space, which may result in decreased efficiency and suffer from the “curse of dimensionality” [1, 2]. Thus, extracting informative features with lower dimensions from high-dimensional data has been attracting considerable attention during the last decades [3-5]. Generally, feature extraction includes two main tasks, i.e. reducing dimension appropriately and looking for the compact representations [6, 7]. The basic principle is to compute a mapping f that can embed each data x in the high dimensional space R^D into the compact representation y in a reduced feature space R^d:

$$f : R^D \to R^d,\ x \to y,\ d \ll D \tag{1}$$

Related works are Principal Component Analysis (PCA) [2], Multi-Dimensional Scaling (MDS) [8, 9], Locally Linear Embedding (LLE) [1], Laplacian Eigenmaps (LE) [6], Isometric Mapping (Isomap) [10], Locality Preserving Projection (LPP) [11], Neighborhood Preserving Embedding (NPE) [12], *and* Isoprojection [13]. It should be noticed that LPP, NPE and Isoprojection are linear approximations to previous LE, LLE and Isomap, respectively. Compared with LPP, Isoprojection and NPE that can output an underlying projection matrix to reveal the linear relations of samples, LE, LLE and Isomap mainly focus on discovering the nonlinear manifold structures by reducing the number of dimensionality directly without obtaining an explicit mapping.

Isomap is one of the most classical global nonlinear manifold learning methods, and aims at seeking an optimal subspace that best preserves the geodesic distance between the points. Let $X=(x_1,x_2,\cdots,x_N)\in R^{D\times N}$ denote a set of *N* points in the original D-dimensional space, and $Y=(y_1,y_2,\cdots,y_N)\in R^{d\times N}$ a set of the reduced representations in the d-dimensional space .

Then, Isomap can perform manifold feature learning in three steps: 1) Determine the nearest neighbors of each sample by using k-neighborhood [14]; 2) Construct a undirected graph *G*(*V*, *E*), where each node $v_i \in V$ corresponds to a point x_i. $d_G(x_i,x_j)$ is the shortest path distances between x_i and x_j over *G*. Dijkstra's algorithm [15] and Floyd's algorithm [16, 17] can be applied to find the shortest paths. Then, Isomap initializes $d_G(x_i,x_j)$ = $d(x_i,x_j)$ suppose x_i and x_j are connected by an edge, and $d_G(x_i,x_j)$ = $+\infty$ otherwise, where $d(x_i,x_j)$ is Euclidean distance; 3) Obtain the low-dimensional embedding *Y* by solving the following problem:

$$\min_{Y}\sum_{i,j}(d(y_i,y_j)-d_G(x_i,x_j))^2 \tag{2}$$

which can be similarly solved as the classical MDS [8, 18, 19].

Locally Linear Embedding (LLE) aims at preserving neighbor information of feature points. First, LLE finds the nearest neighbor of each point $x_i, i=1,2,\ldots,n$. Second, We obtain the reconstruction weights matrix $W(i,j), i=0,1,...,n$ that each point $x_i, i=1,2,\ldots,n$ is represented by neighbor points ,which can be computed by minimizing the following optimization problem:

$$\arg\max_{W} E_W = \sum_{i=1}^{n}\left\| x_i - \sum_{j=1}^{n} W(i,j)x_{ij} \right\|_2^2 \tag{3}$$

In this term, x_{ij} denotes j-neighbor of x_i.

The weights are 0 if they aren't neighbors, the sum of rows of the weights matrix is 1.

$$\sum_{j=1}^{n} W(i,j) = 1 \tag{4}$$

This meas that the sum of the weights of all neighbors is 1. $y_i \in R^m, i = 0,1,...,n$ computes the low-demensional space at a minimum cost.

$$\arg\max_{Y} E_Y = \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{n} W(i,j) y_j \right\|_2^2 \tag{5}$$

The above optimization has two difficulties to avoid the event obtained degrade result, which are conditions that the outputs should be centralized, namely $\sum_{i}^{n} \mathbf{y}_i = 0$ and have an unit co-variance matrix. Thus, We summarise the proposed feature extraction algorithm by linear embedding: 1) Perform Eigen-decomposition of the matrix $(I-W)^T(I-W)$; 2) Remove eigenvetors corresponding to the minimum eigenvalues; 3) Collect eigenvetors corresponding to the eigenvalues less than the minimuns. We compute the low-demensional space $\mathbf{y}_i, i = 0,1,...,n$ by this eigenvectors.

Related works have advantages that are preserving the geometric constructions and the neighbor information but disadvantages that aren't ensuring linear characteristic obtained representations directly from an outside new data.

Finally, in this paper, we propose a new algorithm that are preserving the geometric constructions of training samples and are ensuring the linear characteristic by incorporating Isomap and LLE and adding a linear feature approximation via.

2. Feature extraction by linear embedding

2.1 Problem formulation

We describe the optimization problem of our new algorithm that is based on the recent Isomap, but improves it by extending the manifold feature learning to linear extension scenario and local feature learning scenario at the same time. Given a set of N training samples in, $X = (x_1, x_2, \cdots, x_N) \in R^{D \times N}$ where D is the original dimensionality of samples. To preserve the geometric constructions of points, it should be used $\min_{Y} \sum_{i,j} (d(y_i, y_j) - d_G(x_i, x_j))^2$. To enable the proposed model to compute low-dimensional local features by using training data, a neighbor preserving regularization $\sum_{i=1}^{N} \left\| y_i - \sum_{j:x_j \in NN(x_i)} W_{ij} y_j \right\|_2^2$ over reduced features is clearly incorporated into the problem of Isomap to build the connection between training data by discovering the pairwise local relationships, where $\sum_i W_{ij} = 1, NN(x_i)$ denotes the neighbor set of x_i, $\|\cdot\|_2$ is Euclidean distance. To enable the proposed our algorithm to learn an explicit projection of feature extractor for handling the outside new data, a feature approximation error $\sum_{i=1}^{N} \|Px_i - y_i\|_2^2$ encoding the mismatch between the embedded features by the extractor and the reduced manifold features is included so that learnt extractor P can embed outside new data efficiently. Thus, the prosposed new algorithm can preserve local neighborhood information and geometric construction of samples. These motivate us to define the following objective function:

$$\min_{Y,P} \sigma(Y,P), \text{s.t.} YY^T = 1, where$$

$$\sigma(Y,P) = \frac{1}{|C|} \sum (d(y_i, y_j) - d_G(x_i, x_j))^2 + \tag{6}$$

$$+\alpha \sum\nolimits_{i=1}^{N} \left\| y_i - \sum\nolimits_{j:x_j \in NN(x_i)} W_{ij} y_j \right\|_2^2 + \beta \sum\nolimits_{i=1}^{N} \|Px_i - y_i\|_2^2$$

where P is linear embedding matrix and α, β are the trade-off parameters. C denotes the numbers of points constraints. Specifically, α mainly trades-off the points discrimination and neighborhood preservation via $\sum\nolimits_{i=1}^{N} \left\| y_i - \sum\nolimits_{j:x_j \in NN(x_i)} W_{ij} y_j \right\|_2^2$ over training data, and β mainly trades-off the neighborhood preserving discriminative manifold feature learning and the linear feature approximation via $\sum\nolimits_{i=1}^{N} \|Px_i - y_i\|_2^2$ for handling outside new data. First, if α is set to a large value, the effects of the neighborhood preservation over the training data would have a greater impact on the objective function value than the feature approximation via. Second, the computation of the feature extractor P only depends on the low-dimensional nonlinear manifold features Y. Thus, suppose α is also set to a small value less than 1, in such cases suppose that β is set to a very large value, the effects of linear approximation would have a greater impact on the objective function value than the effects of other terms, and else the objective function value would be determined by trading-off the involved several terms jointly in the formulation. Note that the reconstruction weights matrix W of the training samples can be computed by minimizing the following LLE-style optimization problem in Equation (3).

$$J = \min_{Y} \sum_{i,j} (d(y_i, y_j) - d_G(x_i, x_j))^2 \tag{7}$$

According to terms J of Isomap defined in Equation (7) and the objective function of our algorighm formulation can be rewritten in matrix form as follows:

$$\min_{Y,P}\sigma(Y,P), s.t. YY^T = 1, \sigma(Y,P) = \frac{1}{|C|}J + \alpha\left\|Y - YW^T\right\|_F^2 + \beta\left\|PX - Y\right\|_F^2 \qquad (8)$$

The above optimization problem can be solved by Eigen-decomposition.

2.2 Eigen-decomposion

In addition to solving the problem of our algorithm, we provide effective scheme by using Eigen-decomposition. So we can rewrite the criterion of our algorithm as

$$\min_{Y,P}\sigma(Y,P) \qquad \begin{gathered}\sigma(Y,P) = \frac{1}{|C|}J + \alpha\left\|Y - YW^T\right\|_F^2 + \beta\left\|PX - Y\right\|_F^2 = \eta_{con}^2 + \\ \eta^2(Y) + \omega(Y) + \varphi(Y,P)\end{gathered} \qquad (9)$$

where each term in the above problem can be expressed as

$$\eta_{con}^2 = \frac{1}{|C|}\sum d_G^2(x_i, x_j) \qquad (10)$$

$$\eta^2(Y) = \frac{1}{|C|}\sum d^2(y_i, y_j)$$

$$(11)\, \omega(Y) = \alpha\left\|Y - YW^T\right\|_F^2 = \alpha tr((Y - YW^T)(Y - YW^T)^T) \qquad (12)$$

$$\varphi(\mathrm{Y, P}) = \beta\left\|\mathrm{P}X - Y\right\|_F^2 = \beta tr((PX - Y)(PX - Y)^T) \qquad (13)$$

where Equation (10) is a constant term. The extractor P in Equation (13) can be obtained by setting the derivative w.r.t. P to zero:

$$\begin{aligned}&\frac{\partial(\gamma(PX - Y)(PX - Y)^T)}{\partial P} / \partial P = \partial(\gamma(PXX^T P^T - PXY^T - YX^T P^T + YY^T)) / \partial P = 0 \\ &\Rightarrow P = YX^T(XX^T)^{-1}\end{aligned} \qquad (14)$$

Equation (11) is transformed as follows:

$$\eta^2(Y) = tr(Y(-V)Y^T) \qquad (15)$$

$$\text{where } V = -\sum_{i,j=1, i\neq j}^{N}\frac{1}{|C|}A^{ij} \qquad (16)$$

Also, we can transform Equation (12) as follows:

$$\omega(Y)=\alpha \| Y-YW^{T} \|_{2}^{2}=\alpha tr((Y-YW^{T})(Y-YW^{T})^{T})= \\ \alpha tr(Y(I-W^{T})(I-W)Y^{T}) \tag{17}$$

where W can be obtained by Equation (3) I is N*N unit matrix.

For convenience, we define $M=\alpha(I-W^{T})(I-W)$.

$$\omega(Y)=tr(YMY^{T}) \tag{18}$$

Next, Equation (13) is transformed by Equation (14).

$$\varphi(Y,P)=\beta \| PX-Y \|_{F}^{2}=\beta tr((PX-Y)(PX-Y)^{T})= \\ \beta tr(Y(I-X^{T}(XX^{T})^{-1}X)Y^{T}) \tag{19}$$

If we define $K=\beta(I-X^{T}(XX^{T})^{-1}X)$, Equation (19) is transformed as follows:

$$\varphi(Y,P)=tr(YKY^{T}) \tag{20}$$

The above Equation (9) is formulated by incorporation Equation (6), (14), (15), (18) and (20) as follows:

$$J=\min_{Y,p}\sigma(Y,P)\ , \quad \eta_{con}^{2}+tr(Y(M+K-V)Y^{T})=\tau(Y,Z) \tag{21}$$

Finally, the optimization problem can be rewritten as

$$\max_{Y} tr(Y(V-M-K)Y^{T}), s.t. YY^{T}=1 \tag{22}$$

Let $A=V-M-K$, we can obtain the d-dimensional embedding as $Y=[\sqrt{\lambda_1}v_1,\sqrt{\lambda_2}v_2,...,\sqrt{\lambda_d}v_d]$, where $v_1,v_2,...,v_d$ are the standard eigenvectors corresponding to first d leading eigenvalues $\lambda_1,\lambda_2,...,\lambda_d$ (in decreasing order) of the matrix A. After the d-dimensional embedding Y is obtained, the linear projection P can be similarly obtained by Equation (14). It is worth noting that the optimization procedures of our algorithm by using Eigen-decomposition are summarized as follows: (1) Compute $V,\ M=\alpha(I-W^{T})(I-W), K=\beta(I-X^{T}(XX^{T})^{-1}X)$. (2) Obtain the first d leading eigenvalues $\lambda_1,\lambda_2,...,\lambda_d$ (in decreasing order) of the matrix A. (3) Obtain Y as $Y=[\sqrt{\lambda_1}v_1,\sqrt{\lambda_2}v_2,...,\sqrt{\lambda_d}v_d]$ and obtain p by Equation (14).

Finally, we can perform feature extraction from outside new data directly by linear projection P.

3. Experimental results

In order to perform one-class classification by the proposed feature extraction algorithm, two real-world datasets, i.e., YALE face database [20] and COIL-20 object database [21] are tested. For the classification experiments, we describe the face and object recognition results.

Figure1.YALE Face database

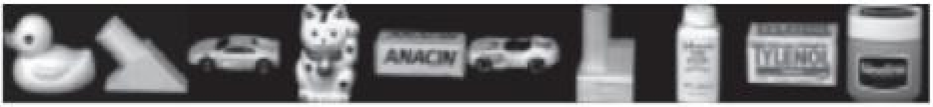

Figure 2. COIL-20 object database

For face recognition, the YALE database are evaluated, while for object recognition, COIL-20 object databases are evaluated.

Table 1. List of used datasets and dataset information.

Data type	Dataset Name	Images	Class
Face image databases	YALE	1650	15
Object image databases	COIL-20	1440	20

In addition to visually evaluating the classification results, we also present the numerical result using F-measure [16,22-24], which are the most commonly used classification evaluation metrics. The F-measure is defined as follos:

$$F_v = \frac{(v^2+1)\Pr ecision \times \operatorname{Re} call}{v^2 \Pr ecision + \operatorname{Re} call}$$

In our experiments, we set parameter $v = 1$. The values of the F-measure range from 0 to 1, that is, the higher the value is, the better the corresponding classification result will be.

3.1 Benchmark database recognition

In this subsection, we evaluate our algorithm and other related methods for representing and reconizing face images over several widely-used benchmark face image databases. Thus, the face recognition performance of our algorithm is mainly compared with those of linear projection based PCA, LE, Isomap, LLE.

Table 2. Face recognition comparisons on the YALE face database

Methods	Settings

	YALE	COIL-20
Isomap	0.1979	0.1068
LE	0.4917	0.2330
PCA	0.6363	0.6033
LLE	0.7037	0.7382
New algorithm	0.7942	0.8428

We summarize the evaluation results in Table 2, that is, our new algorithm can deliver higher mean and better records than other compared algorithms in most cases.

3.2 Investigation of parameter selections

Note that there are two model parameters in our objective function, i.e., α, β, but tuning the two parameters at the same time is not easy. As a widely-used method for the parameter selection, the strategy of grid search [25,26-28] is employed. Specifically, we aim at fixing one parameter and tuning the other one by grid search. We explore the effects of different parameter setting by circulating from candidate set $\{10^{-8},10^{-6},10^{-4},\ldots 10^{-6},10^{-8}\}$ for YALE face database.

Table 3. Investigation results of parameter selection

α β	10^{-8}	10^{-6}	10^{-4}	10^{-2}	10^{0}	10^{2}	10^{4}	10^{6}	10^{8}
10^{-8}	0.042	0.092	0.345	0.259	0.327	0.146	0.346	0.024	0.045
10^{-6}	0.076	0.157	0.767	0.652	0.435	0.327	0.542	0.326	0.087
10^{-4}	0.132	0.252	0.545	0.445	0.837	0.439	0.456	0.437	0.167
10^{-2}	0.147	0.436	0.576	0.452	0.787	0.768	0.767	0.459	0.145
10^{0}	0.107	0.342	0.545	0.659	0.767	0.897	0.865	0.729	0.245
10^{2}	0.096	0.246	0.647	0.329	0.843	0.668	0.678	0.526	0.145
10^{4}	0.143	0.154	0.589	0.445	0.867	0.547	0.675	0.767	0.047

10^6	0.087	0.143	0.767	0.659	0.235	0.329	0.767	0.326	0.142
10^8	0.037	0.042	0.565	0.438	0.347	0.134	0.436	0.246	0.054

We can see the investigation results that the efficiency is the highest when $\alpha=10^2$, $\beta=10^0$ from the Table 3.

4. Conclusion

In this paper, we propose new algorithm to perform efficiently feature extraction of one-class for one-class classification. We incorporate Isomap and LLE to preserve geometric constructions and neighbor information of points and add linear approximation via to represent features directly from outside new data. The experiment results show that the proposed algorithm is more efficient that related works.

In the future research, we will establish new ways so that a more efficient feature extraction algorithm can be inseted.

References

[1] S. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[2] I.T. Jolliffe, in: Principal Component Analysis, 87, Springer, Berlin, 1986, pp. 41–64.

[3] X. Han, L. Clemmensen, Regularized generalized eigen-decomposition with applications to sparse supervised feature extraction and sparse discriminant analysis, Pattern Recognit. 49 (2016) 43–54.

[4] Z. Zhao, L. Jiao, F. Liu, Semisupervised discriminant feature learning for SAR image category via sparse ensemble, IEEE Trans. Geosci. Remote Sensing 54(6) (2016) 1–16.

[5] C L. Lekamalage, Y. Yang, G Huang, Dimension reduction with extreme learning machine, IEEE Trans. Image Process. 25 (8) (2016) 1-1.

[6] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, Neural Inf. Process. Syst. 14 (6) (2002) 585–591.

[7] S.C. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: a general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell. 29 (1) (2007) 40–51.

[8] T.F. Cox, M.A Cox, Multidimensional Scaling, CRC Press, 2000.

[9] I. Borg, P.J.F. Groenen, Modern Multidimensional scaling: Theory and Applications, Springer Science & Business Media, 2005.

[10] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[11] X. He, P. Niyogi, Locality preserving projections, in: Proceedings of Neural Information Processing Systems (NIPS), vol. 16, 2003.

[12] X. He, D. Cai, S. Yan, Neighborhood preserving embedding, in: Proceedings of the IEEE International Conference on Computer Vision, 2005, pp. 1208–1213.

[13] D. Cai, X. He, J. Han, Isometric projection, in: Proceedings of the National Conference on Artificial Intelligence, 22, Menlo Park, CA; Cambridge, MA; London, AAAI Press; MIT Press, 2007, p. 528.

[14] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.

[15] M. Sniedovich, Dynamic Programming: Foundations and Principles, CRC press, 2010.

[16] Q. Wang, M. Chen, X. Li, Quantifying and detecting collective motion by manifold learning, in: *Proceedings of the AAAI Conference on Artificial Intelligence(AAAI*), 2017, pp. 4292–4298.

[17] S. Hougardy, The Floyd–Warshall algorithm on graphs with negative cycles, Inf.Process. Lett. 110 (8) (2010) 279–281.

[18] D.D. Ridder, O. Kouropteva, O. Okun, M. Pietikanien, R.W. Duin, Supervised locally linear embedding, in: Proceedings of the Artificial Neural Networks and Neural Information Processing, 2003.

[19] A. Martinez, A.C. Kak, PCA versus LDA, IEEE Trans. Pattern Anal Mach. Intell.23 (2) (2001) 228–233.

[20] T. Sim, T. Kanade, Combining models and exemplars for face recognition: An illuminating example, in: *Proceedings of the CVPR 2001 Workshop on Models* versus *Exemplars in Computer Vision*, vol. 1, 2001.

[21] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), 1996 Technical report CUCS-005-96.

[22] D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, IEEE Trans. Knowl. Data Eng. 17 (12) (2005) 1624–1637.

[23] X.F. He, D. Cai, P. Niyogi, Laplacian score for feature selection, Adv. Neural Inf. Process. Syst. (2005)

507–514.

[24] X. Li, M. Chen, F. Nie, Q. Wang, A multiview-based parameter free framework for group detection, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2017, pp. 4147–4153.

[25] Z. Zhang, F. Li, M. Zhao, Joint low-rank and sparse principal feature coding for enhanced robust representation and visual classification, IEEE Trans. Image Process. 25 (6) (2016) 2429–2443.

[26] W. Jiang, Z. Zhang, F. Li, Joint label consistent dictionary learning and adaptive label prediction for semisupervised machine fault classification, IEEE Trans.Ind. Inf. 12 (1) (2016) 248–256.

[27] M. Muja, D G. Lowe, Scalable nearest neighbor algorithms for high dimensional data, IEEE Trans. Pattern Anal. Mach. Intell. 36 (11) (2014) 2227–2240.

[28] J. Bergstra, D. Yamins, D. Cox, Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures, in: Proceedings of the International Conf. on Machine Learning (ICML), vol. 28, 2013,pp. 115–123.

.