Dimension Reduction for Structured Composite Classes in Multi-Object Environments

MARTON SZEMENYEI
Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
2, Magyar tudosok korutja, 1117 Budapest
HUNGARY
szemenyei@iit.bme.hu

FERENC VAJDA
Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
2, Magyar tudosok korutja, 1117 Budapest
HUNGARY
vajda@iit.bme.hu

Abstract: Dimension reduction methods are widely used in computer vision and pattern recognition applications, such as object class recognition. [1] However, certain object description methods produce Structured Composite Classes (SCCs), where a single occurrence or instance of a class is composed of a set of vectors (nodes), each drawn from a different probability distribution. Moreover, for some tasks, such as pose estimation using node locations it is desirable to discriminate not only between different classes but also between the individual nodes of a single object. Classical discriminant analysis (DA) methods are insufficient to perform this task. Therefore, we propose new methods, namely Structured Composite DA (SCDA) and Structured Subclass Composite DA (SSCDA) to solve this particular problem. We also demonstrate the efficiency of our method on simulated and real-world examples.

Key–Words: Dimension reduction, Discriminant analysis, Object recognition, Registration.

1 Introduction

Dimension reduction procedures are found in many important applications in the areas of object classification [1] [2] [3], feature selection [4] or feature extraction [5]. The standard methods such as Principal Component Analysis (PCA) [6], Independent Component Analysis (ICA) [7] and Linear Discriminant Analysis (LDA) [8] are most commonly employed.

In certain applications [9], however, it is more appropriate to describe instances of objects as unordered sets or graphs of feature vectors, that is, we have Structured Composite Classes (SCC). The individual feature vectors of an instance will be referred to as *nodes* of the object or class. While it is possible to apply classic dimension reduction methods to this problem, these algorithms will not produce optimal results when applied to SCCs.

SSCs appear in a wide range of cases in computer vision applications. It is common to describe the visual appearance of an object by using a set of local features, such as SIFT [10]. Some models, such as Bag of Visual Words [11] treat these as sets, while others, such as part-based or constellation models [12] treat them as graphs. The local features approach also appears frequently in 3D shape recognition [13]. Another occurrence of SSCs is in [14] and [9], who describe 3D shapes as graphs of primitive shapes. It is worth noting that graphs of vectors can be reduced to sets of vectors using a graph node embedding technique, such as [15].

Some applications [14] [9] [15] include spotting or localizing subgraphs or subsets corresponding to a certain object class (segmentation by recognition). In these cases it is a common strategy to recognize individual nodes and use that to infer the presence of an object from a give class. These methods do not only require discrimination between classes in general but also between the individual nodes of the class. Being able to tell different nodes apart also gives the algorithm the power to consider their relative importance.

This extra criterion is extremely useful in cases where object registration or pose estimation is also needed. If it is easy to discriminate between nodes, then a class model consisting of multiple nodes can be learned (including the relative positions of the individual nodes). Using node-by-node matching pose estimation can be performed. It is worth noting that in most of these applications only class labels are available, that is, individual nodes are not labeled based on their similarity (or lack of thereof).

In this paper we present dimension reduction methods for Structured Composite Classes, by extending existing discriminant analysis algorithms. Our methods introduce the within instance scatter matrix to ensure the separation between nodes of a given class. Furthermore, we provide a method to use the labeling information of SCCs to avoid the clustering step that is required by some methods.

In the next section we give a brief overview of dimension reduction methods, especially linear discriminant analysis, and present some of the more recent results which are applicable to SCCs. In section 3 we discuss structured composite classes, and present our methods for performing discriminant analysis on them that satisfy both of the aforementioned criteria. Lastly, in section 4 we demonstrate the efficiency of our algorithm by evaluating it on synthetic and real-world datasets and comparing its performance against classic methods.

2 Dimension reduction

In this section we give a brief overview of dimension reduction methods with emphasis on Linear Discriminant Analysis and its variants. The second part of the chapter is devoted to mixture and subclass extensions of LDA.

One of the most well-known dimension reduction methods is PCA [6] which finds the linear combinations of the original dimensions that maximize the variance of the dataset. Thus, it can be understood as an algorithm that computes the optimal compression of the data. Since PCA doesn't use class labels, it is an unsupervised method. However, in classification problems PCA might not select the dimensions that are most useful in telling different classes apart. [16]

Discriminant Analysis (DA) techniques, on the other hand, are a supervised procedures which use class labels to find the directions in the parameter space that are most suited to separate the different classes. The simplest of such methods employs Wilks' lambda [2] statistic, which is computed as follows:

$$\lambda_W = \frac{S_{wg}}{S_{total}} \quad (1)$$

Where S_{wg} is the within group sum of squares and S_{total} is the total sum of squares along a given dimension. The statistic is computed for each dimension independently and the dimensions for which it is close to 0 are kept. An obvious disadvantage of the method is that it evaluates different dimensions independently, therefore it will fail if the data is separable along the linear combinations of the dimensions. In this case it is beneficial to use PCA transformation on the data before computing λ_W (without throwing away dimensions of course).

2.1 Linear Discriminant Analysis

Linear Discriminant Analysis [8] is one of the most widely used dimension reduction methods. Its basic assumption is that the classes are normally distributed with different means but the same covariance matrix. Similarly to PCA, LDA computes optimal orthogonal linear combinations of the original dimensions. However these new base vectors maximize separability of the classes instead of the variance of the entire dataset.

LDA can be formulated as an optimization problem:

$$\max_{w} \frac{\boldsymbol{w}^T \boldsymbol{S_b} \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{S_w} \boldsymbol{w}} \quad (2)$$

$$\boldsymbol{S_b} = \sum_{i=1}^{C} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \quad (3)$$

$$\boldsymbol{S_w} = \sum_{i=1}^{C} \sum_{j=1}^{n_i} (\boldsymbol{\mu}_i - \boldsymbol{x}_{i,j})(\boldsymbol{\mu}_i - \boldsymbol{x}_{i,j})^T \quad (4)$$

Where $\boldsymbol{S_b}$ is the between-class scatter matrix, $\boldsymbol{S_w}$ is the within class scatter matrix, $\boldsymbol{\mu}$ is the mean of the data set $\boldsymbol{\mu}_i$ is the mean of the i_{th} class and C is the number of classes. It is worth noting, that $\boldsymbol{S_w}$ can be replaced with the total scatter matrix $\boldsymbol{S_t}$ because of the following equation:

$$\boldsymbol{S_t} = \boldsymbol{S_b} + \boldsymbol{S_w} \quad (5)$$

Where:

$$\boldsymbol{S_t} = \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^T \quad (6)$$

This optimization criterion leads to a generalized eigenvalue-eigenvector problem, which in the case of an invertible $\boldsymbol{S_w}$ or $\boldsymbol{S_t}$ can be solved by performing eigendecomposition on $\boldsymbol{S_t}^{-1}\boldsymbol{S_b}$ or $\boldsymbol{S_w}^{-1}\boldsymbol{S_b}$ and taking the eigenvector belonging to the largest eigenvalue. More discriminant dimensions may be extracted by taking the eigenvectors corresponding to the second, third, etc. largest eigenvalues.

Over the years numerous variations of LDA have been proposed, such as Penalized Discriminant Analysis (PDA) [17] which is a weighted version of LDA. Weights allow the algorithm to penalize unstable features thus improving the robustness of the method. Another variant is Nonparametric DA (NDA) [18], which uses a nearest-neighbors approach to define the between-class scatter matrix to relax the Normal assumption of LDA.

Another important problem with LDA is that it does not take the local geometry of the dataset into account. [19] To overcome this problem, Locality Sensitive Discriminant Analysis (LSDA) [19] has been proposed, which discovers the local manifold structure of the dataset and uses this information to find the optimal discriminating projection. Methods, like Structured Semi-supervised Discriminant Analysis (SSDA) [16] employ similar strategies, with the addition that they use he local manifold information to make use of unlabeled data as well.

Finally, some researchers used the kernel trick to extend LDA into nonlinear cases, such as the Generalized Discriminant Analysis method [20]. In [21] a Grassmannian graph embedding framework is employed to implement kernel-based DA. However, none of these methods are applicable to the case of SC classes, since they all assume at least partly labeled data.

2.2 Subclass and mixture methods

In this section we discuss a particular variation of LDA methods which make the assumption that the data of each class is generated by the several different normal distributions, or in other words they assume a Mixture of Gaussians (MoG) model. Our reason for discussing these variants separately is that they are relatively easy to extend to the problem of SC classes, since the only important difference is that in SC classes instances from different subclasses might be present *at the same time*.

Some of these mixture methods [22] [23] use the Expectation Maximization (EM) [24] algorithm to estimate the underlying distributions of the classes, then they use classic LDA to find the optimal discriminating projection. A drawback of these methods is that they are not applicable if the number of data points is too low. [25]

A different approach, Subclass Discriminant Analysis (SDA) [25] uses clustering to estimate the means of the underlying normal distributions of the subclasses. With the help of subclass means they define the between subclass scatter matrix as follows:

$$\boldsymbol{S_{bs}} = \sum_{i=1}^{C}\sum_{j=1}^{H}(\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu})(\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu})^T \tag{7}$$

Where C is the number of classes, H_i andH_j are the number of subclasses in the i_{th} and j_{th} classes and $\boldsymbol{\mu}_{i,j}$ is the mean of the j_{th} subclass of the i_{th} class. SDA replaces the between class scatter $\boldsymbol{S_b}$ with the between subclass scatter $\boldsymbol{S_{bs}}$. The procedure also determines the number of clusters using a brute-force iteration from one to a user-defined maximum and selecting the number that maximizes classification accuracy.

There is a modified version of SDA called Mixture Subclass Discriminant Analysis (MSDA) [26], which computes the scatter matrix only between the subclasses of *different classes* in order to prevent the algorithm from preferring directions that can separate subclasses of the same class. They compute the between subclass scatter matrix by:

$$\boldsymbol{S_{bsb}} = \sum_{i=1}^{C-1}\sum_{j=1}^{H_i}\sum_{k=i+1}^{C}\sum_{l=1}^{H_j}(\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu}_{k,l})(\boldsymbol{\mu}_{i,j} - \boldsymbol{\mu}_{k,l})^T \tag{8}$$

It is important to remark that these algorithms assume that all classes have the same number of subclasses which may make them inaccurate in cases when this assumption is clearly false.

3 Structured Composite Discriminant Analysis

In this chapter we present our methods for performing discriminant analysis in cases where classes are composed of a set of vectors instead of a single one. In this problem, labels are only available for object instances, i.e. individual nodes of a single instance are not labeled or ordered. Moreover, instances of the same class might have a different number of nodes present and the distribution of the number of nodes might differ between classes.

It is easy to see that it would be ill-advised to replace the nodes representing an instance of a class with their mean, since the difference between classes might be significant between individual nodes, while the classes have the same mean. This point is illustrated in **Figure 1**. It is also obvious that the distribution of all nodes in a class is not even approximately normal therefore performing LDA on the level of nodes would produce suboptimal results as well. Lastly, classic LDA will not discriminate between the nodes of the same instance, which is an important requirement.

3.1 The within Instance Scatter matrix

The last problem of LDA we presented above can be addressed in a relatively straightforward way: by adding a second discriminant criterion that encourages selecting dimensions that separate nodes within

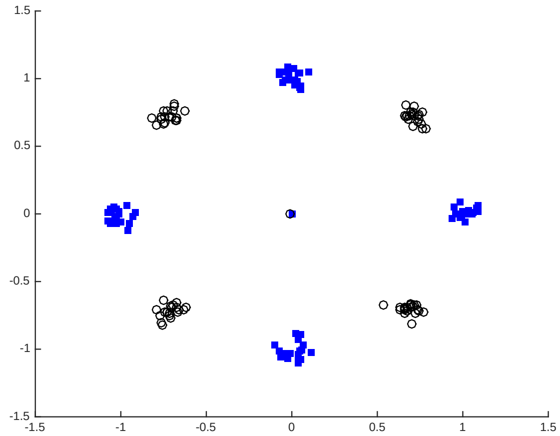

Figure 1: *Typical classes with separable nodes but indistinguishable means.*

instances. We call this addition the within instance scatter matrix which is computed as follows:

$$\boldsymbol{S_{wi}} = \sum_{i=1}^{C}\sum_{j=1}^{N_i}\sum_{k=1}^{n_{i,j}}(\boldsymbol{\mu}_{i,j} - \boldsymbol{x}_{i,j,k})(\boldsymbol{\mu}_{i,j} - \boldsymbol{x}_{i,j,k})^T \quad (9)$$

Where C is the number of classes, N_i is the number of instances in the i_{th} class, $n_{i,j}$ is the number of nodes in the j_{th} instance, and $\boldsymbol{\mu}_{i,j}$ is the mean of nodes in the same class. We can then define the between class node scatter matrix similarly to classic LDA:

$$\boldsymbol{S_{bcn}} = \sum_{i=1}^{C}(\boldsymbol{\mu} - \boldsymbol{\mu}_i)(\boldsymbol{\mu} - \boldsymbol{\mu}_i)^T \quad (10)$$

Where $\boldsymbol{\mu}$ is the mean of all nodes and $\boldsymbol{\mu}_i$ is the mean of all nodes in the i_{th} class. Then the optimization criterion of Structured Composite Discriminant Analysis (SCDA) can be written as:

$$\max_{w} \frac{\boldsymbol{w}^T\boldsymbol{S_{bci}}\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S_t}\boldsymbol{w}} \quad (11)$$

$$\boldsymbol{S_{bci}} = \boldsymbol{S_{bcn}} + \boldsymbol{S_{wi}} \quad (12)$$

It is important to note that in some cases the two scatter matrices might not be in the same order of magnitude, which can lead to one of the criteria being ignored in the favor of the other. In such cases, it is desirable to weigh the two scatter matrices to be added. The relative weight might be determined manually or by iterating through possible values. A special value that ensures the equal importance of the two criteria is:

$$\boldsymbol{S_{bci}} = \boldsymbol{S_{bcn}} + \frac{tr(\boldsymbol{S_{bcn}})}{tr(\boldsymbol{S_{wi}})}\boldsymbol{S_{wi}} \quad (13)$$

3.2 Improved SDA

Despite the addition of the within instance scatter, SCDA still suffers from the inaccuracy of the between class scatter matrix due to the invalid normal assumption. This problem was, however, solved by mixture models and SDA by assuming a mixture of gaussians model and computing the between subclass scatter accordingly. It is important to recognize that SDA is fairly easy to apply to the problem of structured composite classes, especially the variants that discriminate between the subclasses of the same class as well.

In spite of the apparent simplicity of the solution, there are several issues with this approach. First, in certain cases the clustering step of SDA might not put nodes of a given instance into different subclasses. This might occur if there are similar nodes within instances but large differences between different instances. Secondly, SDA might not guess the number of nodes (subclasses) correctly, since it selects the number that optimizes separability of classes, not the separability of nodes within instances. Moreover, since SDA assumes that all classes have the same number of nodes, it will certainly fail to find the optimal separation in cases where classes have different number of nodes.

Lastly, SDA uses a brute force approach to determine the correct number of subclasses, which means performing several clustering and discriminant analysis steps on high dimensional data. This is computationally expensive and should be avoided if possible.

Thankfully, it is possible to address all of these issues by improving the clustering procedure (SDA-IC). It is possible to estimate the number of subclasses in each class separately by setting them to the number of nodes in the largest instance of the given class. We may initialize subclass means by setting them to the nodes of the largest instance. Then, by using a nearest mean approach we can assign the remaining nodes to the different subclasses. This is equivalent with using a single iteration of the k-means clustering algorithm. During the assignment step it is possible to penalize the algorithm for putting two nodes of the same object into the same subclass cluster.

Since we initialized the clusters by considering our requirement of separating the nodes of a single instance, the resulting clusters are much more likely to be close to the optimum. Using this trick we have used the additional information in our dataset to achieve significantly more accurate subclass clusters at reduced computational cost.

3.3 Combining SDA-IC and SCDA

A minor issue with the SDA-IC algorithm is that it compresses the two separability criteria into the same scatter matrix. This prevents us from scaling the relative importance of the two matrices which may be important for the reasons given in chapter 3.1. An additional reason for doing this is that certain applications demand significantly less tolerance for one type of error than the other. This may justify adding artificial bias to the DA algorithm.

This may be achieved by combining the SDA-IC and the SCDA methods by defining the $\boldsymbol{S_{bci}}$ matrix of SSCDA by:

$$\boldsymbol{S_{bci}} = \boldsymbol{S_{bsb}} + \boldsymbol{S_{wi}} \tag{14}$$

Where $\boldsymbol{S_{bsb}}$ is the between subclass scatter given by (8), and the actual subclasses are determined by SDA-IC.

It is important to recognize, however, that values resulting from within instance scatter may remain in the between-subclass scatter matrix, even if the MSDA definition is used. This is because the scatter is computed between a given subclass, and *all* the subclasses from the other class.

Our argument is that with the introduction of the within instance scatter matrix it is sufficient to compute the between subclass scatter only between the *closest* subclasses of different classes (in case of ambiguity all 'close' subclasses will be used). The reason for this is that the within instance scatter matrix will include all the other scatter information that is required to separate subclasses. Therefore this modification will not harm the algorithms ability to discriminate.

It brings two advantages, however. First, that using this new definition we could completely separate the two criteria into two scatter matrices, which can be weighted according to the applications' requirements. Second, during the classification of SC classes their nodes have to be classified first. During the classification of the individual nodes, only similar nodes from the other classes may be responsible for misclassification. Therefore, discriminating between these nodes is far more important than discriminating between dissimilar ones. This way, the between subclass scatter matrix only contains the most important information, while the within instance scatter contains the secondary information.

It is worth noting, that without the within instance scatter matrix the SSCDA algorithm would likely select a projection which ignores the dimensions that separate given nodes from relatively distant ones in other classes, resulting in a deeply flawed solution.

3.4 Extension of Wilks' lambda

In the last part of the current chapter we show that it is possible to use the ideas developed in the previous section to provide an extension of the Wilks' lambda statistic. This is done by creating two different lambdas: one for between-class the other for within instance separability. Since:

$$S_{wc} = S_{bi} + S_{wi} \tag{15}$$

Where S_{wc} is the within class scatter, S_{bi} is the scatter of instance means, and S_{wi} is within instance scatter. We can use a statistic for within instance separability that is computed as follows:

$$\lambda_{wi} = \frac{S_{bi}}{S_{wc}} \tag{16}$$

The lambda statistic used for between class separation can be computed simply using the classic formulation (1). Here, the within group scatter S_{wg} obviously contains within subclass scatter values. In order to select dimensions using these two criteria one might use the sum or the weighted sum of the two lambda values $\lambda_{SSCDA} = \lambda_{bsc} + \mu\lambda_{wi}$, or one may simply combine the dimensions selected by using the two lambda statistics separately.

4 Empirical Results

In this chapter we present the evaluation procedure we used to assess the efficiency of our methods. We have compared the performance of LDA, SDA, SDA-IC, SCDA, SSCDA, and the Wilks lambda method on several different datasets. We used 10-fold cross-validation for computing validation accuracies. We begin by presenting our classification method, followed by presenting the results on different types of datasets.

Our classification method is a semi-supervised one, since we assume that the nodes of instances are not labeled. This means, that we have to construct a class model first that can be used to classify nodes individually. After running the DA algorithm we use a robust k-medoids clustering method to create node clusters. Unlike during the discriminant analysis this clustering step determines the number of clusters automatically by minimizing the within group scatter, while penalizing the number of clusters.

Following that, nodes are labeled according to their distance to the model clusters. Each node is given two different labels: the first is the class label, while the second is the label of the node cluster within the given class. Therefore the class labels of each node can be considered a vote for the presence of a

Results (%)	DN	Dn	ON	On
LDA	87	87	88	91
SDA	96	98	81	98
SDA-IC	99	100	89	98
SCDA	99	98	94	99
SSCDA	99	100	95	99
Wilks	100	100	86	99

Table 1: *Results of the algorithms for the four synthetic datasets. Only results for the metric a_{wi} are displayed, since the datasets were linearly separable.*

Dataset	CN			Δ N		
Results (%)	a_{wi}	a_n	a_c	a_{wi}	a_n	a_c
LDA	78	59	69	78	58	68
SDA	99	58	71	99	62	74
SDA-IC	99	66	82	100	79	92
SCDA	100	65	81	100	73	89
SSCDA	100	71	87	100	74	88
Wilks	100	66	79	100	69	83

Table 2: *Results of the algorithms for the special synthetic datasets.*

certain class. The class label of the object is then decided by the plurality of node votes. The node cluster labels can be used to discriminate between nodes of a single instance.

During the testing we computed three important statistics. The first, a_{wi} is the percentage of nodes that got a different node label within the correct class. This measures the algorithm's ability to discriminate within instance. The second a_n is the percentage of nodes that were assigned to the correct class, while the third a_c is the percentage of object instances that were assigned to the correct class.

4.1 Random Synthetic Classes

Since there isn't a great number of online databases that contain SC classes we first tested the efficiency of our algorithm on synthetic datasets. The first dataset contains five random classes each with five different random nodes that have unique probabilities to appear in a given instance of the class. There are four variations of this dataset: In two cases there is no overlap between dimensions that are useful for separating classes and those that can be used to discriminate between nodes of a particular instance. In the remaining two cases there same dimensions can be used for both kind of separation. Also, in all four datasets we add noise to the nodes, drawn from a normal distribution with zero mean, but in two cases the noise is relatively small, allowing for easy separation, while in other two cases the variance of the noise is comparable to the variance between nodes and classes. Thus the four datasets are: disjunct large noise (DN), disjunct small noise (Dn), overlapping large noise (ON), and overlapping small noise (On). The results of our algorithm for these datasets are shown in **Table 1**.

In addition to the previous four datasets we have created examples for peculiar situations in which the classic methods clearly fail. One such case is when the closest subclasses from different classes are very close compared to other subclasses. (CN Dataset) If other subclasses contain significant noise in the directions that can separate the closest subclasses then all methods with the exception of SSCDA will perform poorly. Another case where LDA and SDA give suboptimal results is when there are classes with different number of nodes (ΔN). We have created datasets to illustrate both cases, each containing two different classes with 500 instances of both classes. The results are shown in **Table 2**.

4.2 3D shape-graphs

In the second stage of testing we use 3D shape recognition as an illustration of our method's efficiency. Schnabel et. al. [27] introduced a method for describing 3D shapes as graphs of primitive shapes. Their method has been used for object recognition [14], grasp planning [28], and augmented reality [9]. Here, we use their algorithm to create graphs from 3D point clouds, where the nodes of the graphs are the primitive shapes and edges represent the geometric transformation between these shapes. For more details, please refer to [9]. In order to construct a structured composite class from this graphical representation, we use a graph node embedding technique from [15].

We created two different databases for testing. The first uses synthetic images created in blender, while the second uses real images of simple objects. In both cases 3D reconstruction was performed using a multi-view technique using 15 successive images of given objects. In both cases we introduced significant variations between objects. The synthetic graph database includes around 2000 graphs in 5 classes, while a real-world database includes roughly 8000 graphs in 5 classes. The results of our algorithm are shown in **Table 3**.

4.3 Feature-based object recognition

The third problem we applied our discriminant analysis algorithms to was visual object classification.

Dataset	Syn.			Real		
Results (%)	a_{wi}	a_n	a_c	a_{wi}	a_n	a_c
LDA	53	78	83	51	57	56
SDA	63	77	84	67	65	65
SDA-IC	71	74	86	71	63	64
SCDA	83	73	87	65	74	74
SSCDA	81	76	86	65	75	75
Wilks	67	79	85	75	69	69

Table 3: *The algorithms' performance on 3D shape graph datasets.*

Dataset	VML			UIUC		
Results (%)	a_{wi}	a_n	a_c	a_{wi}	a_n	a_c
LDA	29	44	56	12	46	59
SDA	51	74	81	54	74	84
SDA-IC	50	72	81	69	82	89
SCDA	66	75	83	83	81	89
SSCDA	66	75	84	82	80	87
Wilks	57	73	82	70	77	87

Table 4: *The methods' accuracy on image datasets with few (< 5) classes.*

Dataset	CVA			VGG		
Results (%)	a_{wi}	a_n	a_c	a_{wi}	a_n	a_c
LDA	54	41	46	42	29	39
SDA	49	36	44	45	31	38
SDA-IC	53	35	47	50	36	47
SCDA	66	47	55	61	43	56
SSCDA	66	43	55	62	45	57
Wilks	56	45	51	53	49	55

Table 5: *The methods' accuracy on image datasets with many (> 5) classes.*

Here, we constructed structured composite classes by detecting the 15 strongest SURF features on each image and storing their descriptor vectors as nodes of an instance of the given class. We performed this on several object recognition databases, including datasets with few classes, such as VML Action Class Database [29], the UIUC Image Database for Car Detection [30], and sets with numerous classes, such as the Caltech Computational Vision Archive (CVA) [31] and the VGG Flower Database [32]. The results of our algorithms are shown in **Table 4** and **Table 5**.

The results presented in this chapter show that our methods are successful in achieving out two criteria on both synthetic and real-world datasets. It is also demonstrated that SCDA and SSCDA usually outperform the classic methods on structured composite classes.

5 Conclusion

In this paper we presented methods for dimension reduction of structured composite (SC) classes by discriminant analysis. An extension of LDA called Structured Composite DA was introduced, followed by an extension of SDA called Subclass Structured Composite DA. We have also discussed improving the clustering method of SDA and extending the Wilks lambda statistic for SC classes.

We have evaluated the efficiency of the methods using both synthetic and real-world datasets and compared them to the classic versions. We have found that our additions have significantly increased the efficiency of the original algorithms by taking the structure of the data into account.

References:

[1] S. Yasuoka, Y. Kang, K. Morooka and H. Nagahashi, Texture Classification Using Hierarchical Discriminant Analysis, *IEEE Conf. on Systems, Man and Cybernetics* 2004, pp. 6395–6400.

[2] B. Surendiran, A. Vadivel, Feature Selection using Stepwise ANOVA Discriminant Analysis for Mammogram Mass Classification, *ACEEE Int. J. on Signal and Image Processing* 2(1), 2011, pp. 17-19

[3] X. Liu, Z. Wang, J. Liu and Z. Feng, Face Recognition with Locality Sensitive Discriminant Analysis Based on Matrix Representation, *IJCNN* 2008, pp. 4052–4058.

[4] F. Song, Z. Guo, D. Mei, Feature selection using principal component analysis, *ICSEM*, 2010, pp. 27–30.

[5] T.-S Lin, Statistical feature extraction and selection for IC test pattern analysis, *ISCAS* , 1992, pp. 391–394.

[6] I.-T. Jolliffe, *Principal Compoent Analysis* Springer Series in Statistics, 2002.

[7] A. Hyvrinen, J. Karhunen, E. Oja, *Independent Compoent Analysis* Wiley, 2001.

[8] G.-J. McLachlan, *Discriminant Analysis and Stiatistical Pattern Recognition* Wiley, 2004.

[9] M. Szemenyei, F. Vajda, Learning 3D Object Recongition Using Graphs Based on Primitive Shapes, *WAIT*, 2015, pp. 187–195.

[10] D.G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision* 60(2), 2004, pp. 91–110.

[11] L. Fei-Fei, P. Perona, A Bayesian Hierarchical Model for Learning Natural Scene Categories, *CVPR05*, 2005, pp. 524-531.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(9), 2010, pp. 1627–1645.

[13] A. M. Bronstein, M. M. Bronstein, and M. Ovsjanikov, Feature-based Methods in 3D Shape Analysis, in: N. Pears, Y. Liu and P. Bunting (eds): *3D Imaging Analysis and Applications*, 2012, Springer, pp. 185–216.

[14] R. Schnabel, R. Wessel, R. Wahl, and R. Klein, Shape Recognition in 3D Point-Clouds, *WSCG*, 2008, pp. 1–8.

[15] M. F. Demirci, Y. Osmanlioglu, A. Shokoufandeh and S. Dickinson, Efficient many-to-many feature matching under the l1 norm, *Computer Vision and Image Understanding* 115(7), 2011, pp. 976-983.

[16] M. Yang, X.-M. Yuan, Structured Semi-Supervised Discriminant Analysis, *ICWAPR*, 2009, pp. 148–153.

[17] T. Hastie, A. Buja, and R. Tibshirani, Penalized Discriminant Analysis, *Annals of Statistics* 23, 1995, pp. 73–102.

[18] K. Fukunaga, *Introduction to Statistical Pattern Recognition* , Academic Press, 1990.

[19] D. Chai, X. He, K. Zhou, J. Han, H. Bao, Locality sensitive discriminant analysis, *IJCAI*, 2007, pp. 708–713.

[20] G. Baudat and F. Anouar, Generalized Discriminant Analysis, *Neural Computation* 12, 2000, pp. 2385–2404.

[21] M. T. Harandi, C. Sanderson, S. Shirazi, B. C. Lovell, Graph embedding discriminant analysis on Grassmannian manifolds for improved image set matching, *CVPR*, 2011, pp. 2705–2712.

[22] T. Hastie and R. Tibshirani, Discriminant Analysis by Gaussian Mixtures, *Journal of the Royal StatisticalL Society* 58(1), 1996, pp. 155–176.

[23] M.- H. Yang and N. Ahuja, Face Detection Using Multimodal Density Models, *International Series in Video Computing* 1, 2001, pp. 97–122.

[24] A. P. Dempster , N. M. Laird , D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal StatisticalL Society* 39(1), 1977, pp. 1–38.

[25] M. Zhu and A. M. Martinez, Subclass discriminant analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* 28(8), 2006, pp. 1274–1286.

[26] N. Gkalelis, V. Mezaris, I. Kompatsiaris, Mixture Subclass Discriminant Analysis, *IEEE Signal Processing Letters* 18(5), 2011, pp. 319–322.

[27] R. Schnabel, R. Wahl, and R. Klein, Efficient RANSAC for Point-Cloud Shape Detection, *Computer Graphics Forum* 26(2), 2007, pp. 214–226.

[28] J. Stckler, R. Steffens, D. Holz and S. Behnke, Efficient 3D Object Perception and Grasp Planning for Mobile Manipulation in Domestic Environments, *Robotics and Autonomous Systems* 1, 2012, pp. 123–130.

[29] Y. Wang, H. Jiang, M. S. Drew, Z-N. Li and G. Mori, Unsupervised Discovery of Action Classes, *IEEE Conference on Computer Vision and Pattern Recognition* 9, 2006

[30] S. Agarwal, A. Awan, and D. Roth, Learning to detect objects in images via a sparse, part-based representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(11), 2004, pp. 1475–1490.

[31] L. Fei-Fei, R. Fergus and P. Perona, Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, *IEEE. CVPR*, 2004

[32] M-E. Nilsback, and A. Zisserman, Automated flower classification over a large number of classes, *Proc. of the Indian Conf. on Computer Vision, Graphics and Image Processing*, 2008