CHAPTER V

CUSTOMIZATION FOR DIMENSIONALITY REDUCTION

In this chapter, we will propose a framework for performing customization for the task of dimensionality reduction. We will also propose an algorithm for dimensionality reduction with additional constraint that the dimensionality reduction is performed with linear function; it is linear dimensionality reduction. The additional constraint in this case reduces the generality for customization to a subgroup of dimensionality reduction. However, the group of algorithms that perform linear dimensionality reduction is still large enough to have considerable impact. With this limitation, we can provide theoretical result up to much more extent than performing it on the much larger set that includes nonlinear dimensionality reduction.

5.1 Framework of Customization for Dimensionality Reduction

The goal of dimensionality reduction is finding a mapping function $\mu : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that will retain the most usefulness of the data, where the definition of usefulness can be different for each algorithm by the objective and constraint on μ. In the most general case where there is no constraint on function μ, this problem can be solved by training another dimensionality reduction $\nu : \mathbb{R}^n \rightarrow \mathbb{R}^m$ based on the customization data and then combining the result when the set of customization data is transformed by μ and ν.

There seem to be no other feasible approaches for customization on dimensionality reduction, due to its nature that is different from classification and regression. The goal for dimensionality reduction is that the resulted data must retain the most useful information, and by any definition of the usefulness, it cannot be measured with just a sole data, $\mathbf{k}$. This is because the usefulness cannot be determined by just its own nonmapped quality such as class for classification or continuous value for regression that can be used to determine the objective, but it must be measured by the information of values of many data

in a dataset. This is completely different from classification and regression that we want the result from the process to be as close as possible to another value associated with the data.

Suppose that $\mathbf{k}$ is transformed by μ and ν to be $\mu(\mathbf{k})$ and $\nu(\mathbf{k})$ respectively, and that the values of $\mu(\mathbf{k})$ and $\nu(\mathbf{k})$ are dependent on each other up to some degree since they are both generated from $\mathbf{k}$ by the procedure. Let us define ρ as a function which $\rho(\mathbf{k}) = [\mu(\mathbf{k})^T \nu(\mathbf{k})^T]^T$. If we perform unsupervised dimensionality reduction on the dataset that is mapped into $\rho(\mathbf{k})$ then it will result in the mapping onto a coordinate of a space that will try to align the value of μ and ν on each other based on the dataset. Thus we can find the most feasible mapping using a dataset X by using all the data used in generating the original dimension reducer and customization data. However, since we might not have the data used in generating the original dimension reducer by the setting, we would have to compromise with just using the customization data to represent the data distribution in the space.

The process of combining two results from dimensionality reduction can be thought of as performing unsupervised dimensionality reduction on the data resulted from both dimensionality reduction. The data resulted from $\mathbf{k}$ which will be used in the unsupervised dimensionality reduction is written as $[\mu(\mathbf{k})^T \nu(\mathbf{k})^T]^T$. Following this procedure, we give the framework for customization for dimensionality reduction as shown in Figure 5.1 and the resulted new dimension reducer is as shown in Figure 5.2.

5.2 Combining Results from Linear Dimensionality Reduction

5.2.1 Algorithm

For the most general formulation of linear dimensionality reduction with n attributes, we can write the linear transformation (Wylie and Barrett, 1982; Nicholson, 2001) as $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ where A is an orthogonal matrix of size n whose rows are sorted in the order of importance and $\mathbf{b}$ is a vector of size n. Given a training procedure for linear dimensionality reduction $f(\mathbf{x})$, one can determine all the $n^2 + n$ parameters of $f(\mathbf{x})$ within A and $\mathbf{b}$ by observing the

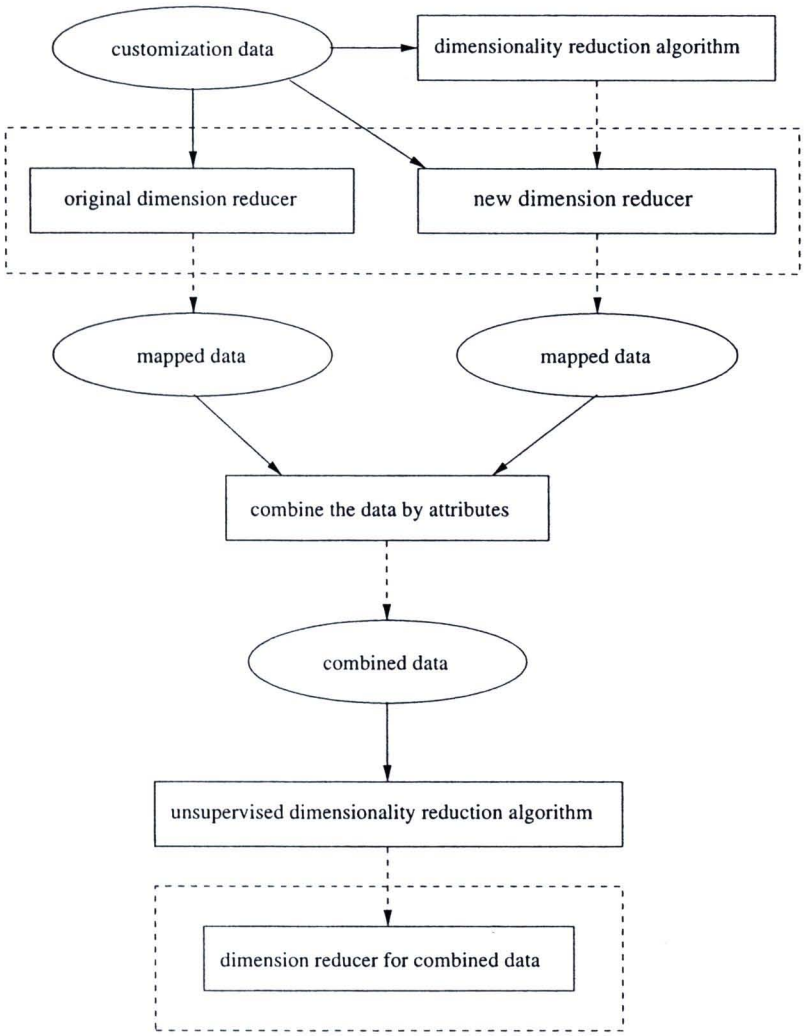

Figure 5.1: Framework for customization for dimensionality reduction.

output given to a set of $(n+1)$ vectors. The simplest of them is a normalized vector in each dimension and the zero vector, which results in the following:

$$\begin{aligned} f(I_n) &= AI_n + \mathbf{b}\mathbf{1_n}^T = A + \mathbf{b}\mathbf{1_n}^T, \\ f(\mathbf{0_n}) &= A\mathbf{0_n} + \mathbf{b} = \mathbf{b}, \\ A &= f(I_n) - \mathbf{b}\mathbf{1_n}^T \\ &= f(I_n) - f(\mathbf{0_n})\mathbf{1_n}^T. \end{aligned}$$

The customization data will be used to train another linear dimension reducer with the same objective as the original given procedure, resulting in another transformation matrix. We will use M^T and N^T to denote the transformation

Figure 5.2: The new dimension reducer as the result of using the framework.

matrix belonging to the original procedure and the transformation matrix created from customization data, respectively.

If we want to perform dimensionality reduction of a dataset with n dimensions to a space with m dimensions, the most idealistic way will be to combine m most important linear spaces resulted from both procedure. The basis for the most important linear space with m dimensions of a linear transformation can be represented with the first m columns of M where each of the columns is a basis vector. For combining two such spaces of the same dimension into a new linear space with the same dimension, we propose considering covariance from all the basis vectors of both spaces, fixing the mean to be at origin. The later process will be to find the m most suitable basis vectors for a new

m-dimensional linear space that gives the greatest variance. This process can be easily described as using the m most important orthonormal basis vectors of both transformation matrices as data to perform PCA, with the mean fixing to be the zero vector. With this interpretation, the procedure will yield the same result as using each normal basis vector along with the unit vector in the opposite direction which will have in the mean of the all the vectors to be zero, and have a covariance matrix with twice the value of covariance matrix of basis vectors with zero mean.

In the case that we want to give a certain weight $\sqrt{\alpha}$ and $\sqrt{1-\alpha}$ to M and N respectively, we can perform scaling on the basis vectors from each matrix with that weight.

Proposition 2. *Performing PCA on the data which are any orthonormal basis vectors that span the same linear space as M and N, with the mean fixed to be zero vector, will always yield the same result.*

Proof. PCA is dependent on calculation of covariance matrix C, which follows the formula $C = DD^T$ when D is the matrix with each column as a data that is translated to have zero mean. From the view of each element in the matrix, this formula can alternatively be written as,

$$C_{ij} = \sum_k D_{ik} D_{jk},$$

when C_{ij} and D_{ij} are the values of the matrix in row i and column j respectively. If we view it from the perspective of each data, C could be written as follows,

$$C = \sum_k \mathbf{D_k} \mathbf{D_k}^T,$$

where $\mathbf{D_k}$ is the k^{th} row of D.

The last equation implies that covariance matrix of a dataset is the summation of covariance matrix for each data, using the same mean. In matrix form, this can be written as:

$$C = [M|N][M|N]^T = MM^T + NN^T.$$

Any set of orthonormal basis vectors that represent the same m-dimensional linear space as a set of orthonormal basis vectors D can be written as DO where O is an orthogonal matrix with m rows. This can be visualized as O represents all possible sets of orthonormal basis vectors in m dimensions, and D is the transformation that aligns the sets of basis vectors to the linear subspace. As shown in the equation, covariance matrix C of a matrix with the data as orthonormal basis of two linear subspaces weighted by $\sqrt{\alpha}$ and $\sqrt{1-\alpha}$ can be calculated as:

$$\begin{aligned}
C &= [\sqrt{\alpha}MO_M|\sqrt{1-\alpha}NO_N][\sqrt{\alpha}MO_M|\sqrt{1-\alpha}NO_N]^T \\
&= (\sqrt{\alpha}MO_M)(\sqrt{\alpha}MO_M)^T + (\sqrt{1-\alpha}NO_N)(\sqrt{1-\alpha}NO_N)^T \\
&= \alpha(MO_M)(MO_M)^T + (1-\alpha)(NO_N)(NO_N)^T \\
&= \alpha MO_M O_M^T M^T + (1-\alpha)NO_N O_N^T N^T \\
&= \alpha MI_m M^T + (1-\alpha)NI_m N^T \\
&= \alpha MM^T + (1-\alpha)NN^T.
\end{aligned}$$

Thus we will obtain the same covariance matrix from eigenvalue decomposition which will yield the same transformation matrix. ■

An important note is that the resulted basis vectors from the linear transformation will not be ordered by the importance in each dimension for the data according to the algorithm for linear dimensionality reduction, but with the probability that the dimension should be selected as the result from combination of the two linear transformations. So if the user want to sort the importance of each axis in the new linear space as in the capability of the linear dimensional reduction algorithm, he must perform the algorithm for linear dimensionality reduction on the customization data after they are transformed by the new linear transformation.

Another note is that this process can also be viewed as combining the unimportant linear space, as they will yield the same result. The importance of this aspect is that since the time taken in calculation of PCA also depends on the number of data, it will be faster to do this if the resulted dimension is more

than half of the original, i.e. $2m > n$.

Proposition 3. *Inverse view of combining linear space with low importance will yield the same result as combining linear space with high importance.*

Proof. Let $M = [M_1|M_2]$ and $N = [N_1|N_2]$ be both orthonormal vectors which fully span the space where M_1 and N_1 span an m-dimensional linear space. We want to proof that the m-dimensional linear space resulted from PCA of $[\sqrt{\alpha}M_1|\sqrt{1-\alpha}N_1]$ with zero mean is the same to the least important m-dimensional linear space from PCA of $[\sqrt{\alpha}M_2|\sqrt{1-\alpha}N_2]$.

Let C_1 and C_2 be covariance matrices of $[\sqrt{\alpha}M_1|\sqrt{1-\alpha}N_1]$ and $[\sqrt{\alpha}M_2|\sqrt{1-\alpha}N_2]$ respectively.

$$\begin{aligned} C_1 &= [\sqrt{\alpha}M_1|\sqrt{1-\alpha}N_1][\sqrt{\alpha}M_1|\sqrt{1-\alpha}N_1]^T \\ &= \alpha M_1 M_1^T + (1-\alpha)N_1 N_1^T \\ C_2 &= \alpha M_2 M_2^T + (1-\alpha)N_2 N_2^T. \end{aligned}$$

Since M is an orthogonal matrix,

$$\begin{aligned} MM^T = I_n &= [M_1|M_2][M_1|M_2]^T = M_1 M_1^T + M_2 M_2^T, \\ M_2 M_2^T &= I_n - M_1 M_1^T. \end{aligned}$$

Let $C_1^T = U\Lambda U^T$ be the result from eigenvalue decomposition of the covariance matrix.

$$\begin{aligned} C_2 &= \alpha M_2 M_2^T + (1-\alpha)N_2 N_2^T \\ &= \alpha(I - M_1 M_1^T) + (1-\alpha)(I - N_1 N_1^T) \\ &= I - \alpha M_1 M_1^T - (1-\alpha)N_1 N_1^T \\ &= I - (\alpha M_1 M_1^T + (1-\alpha)N_1 N_1^T) \\ &= I - C_1 \\ &= I - U\Lambda U^T \\ &= U(I - \Lambda)U^T. \end{aligned}$$

Since C_2 is summation of two positive semidefinite matrices, C_2 is a positive

semidefinite matrix as well. The result is that eigenvalues of C_2 which are the diagonal elements of $(I - \Lambda)$ are nonnegative and are sorted in reverse order to eigenvalues of C_1. Since eigenvectors of both C_1 and C_2 are the same but in reverse order, this proves that the resulted linear space spanned by the m most importance eigenvectors of C_1 is the same to the linear space spanned by m least importance eigenvector of C_2. ■

We will let both spaces be equally important by setting $\alpha = 0.5$. This will yield the same result as letting both weighting terms α and $1 - \alpha$ be 1 which will simplify the calculation. This setting will be used for all the following proofs in this section.

5.2.2 Theoretical Results

Proposition 4. *We can fully describe a linear space spanned by the set of orthonormal basis vectors M with MM^T.*

Proof. First, we will prove that for any orthogonal matrix O with m rows, the space spanned by any MO which is the same space will result in the same value.

$$\begin{aligned} MO(MO)^T &= MOO^TM^T \\ &= MI_mM^T \\ &= MM^T. \end{aligned}$$

For the inverse, since MM^T is both symmetric and positive semidefinite, we can perform eigenvalue decomposition on MM^T. If the space spanned by M is an m-dimensional linear space then rank of M will be m, and as the result the rank of MM^T will be m as well since it is a product of two matrices with rank m. Other than that, all of its eigenvalues will be 1 as well. Therefore, by performing eigenvalue decomposition and eliminating all the elements with 0 in the eigenvalues, we will get the following,

$$\begin{aligned} MM^T &= NI_mN^T, \\ &= NN^T, \end{aligned}$$

where I_m is an identity matrix with m rows and N is another orthonormal basis vectors with the equal number of rows and columns to M. Hence, from the equation,

$$\begin{aligned} MM^T &= NN^T, \\ I_m &= M^T NN^T M, \\ &= M^T N(M^T N)^T, \end{aligned}$$

and thus $M^T N$ is an orthogonal matrix which transforms M into N and that both M and N span the same linear space. ■

From the result of Proposition 4, we will now refer to a linear space spanned by a set of orthonormal basis vectors M with the matrix $M_S = MM^T$.

Proposition 5. *Let* $\mathbf{k}$ *be a unit vector and* X_S *be a linear space,* $X_S\mathbf{k}$ *is the projection of* $\mathbf{k}$ *on the space* X_S.

Proof. Let X be a set of orthonormal basis of X_S that is $X_S = XX^T$. $X^T\mathbf{k}$ is the projection of $\mathbf{k}$ onto each of the orthonormal basis of space X_S. $X(X^T\mathbf{k})$ can be viewed as multiplying each of the orthonormal basis vectors of X_S with the value that $\mathbf{k}$ is projected on them. Hence, $X_S\mathbf{k}$ is the projection of $\mathbf{k}$ on the space X_S. ■

Definition 1. *Let A,B,C and D be the sets of orthonormal basis vectors which span an m-dimensional linear space. We say that* the difference between the space spanned by A and B is equal to the difference between the space spanned by C and D *if and only if there exists an orthogonal matrix O which causes the space spanned by AO and BO be equivalent to the space spanned by C and D respectively.*

Definition 1 arises from the property that the operation of an orthogonal matrix will be equivalent to a sequence of rotation and reflection centered at the origin, which will preserve the distance between any vectors in the space. We will have another constraint for a function which measures difference between two linear spaces.

Definition 2. *Let M and N be two sets of orthonormal basis vectors. A function $f(M,N)$ is* a measure of difference between the spaces spanned by both M and N *if and only if,*

i) $f(M,N) = f(N,M)$,

ii) $f(M,N) = f(OM,ON)$,

iii) $f(M,N) = f(MO_M,NO_N)$,

where O, O_M and O_N are any orthogonal matrices.

In Definition 2, the first condition is from that the function must be reflexive since it is a measure of difference. The second condition is from Definition 1. The third condition is from that the function should yield the same output for any orthonormal basis vectors that span the same space.

Proposition 6. *Let M and N be two sets of orthonormal basis vectors which span an m-dimensional linear space within an n-dimensional space. We can measure the difference between them with the function $f(M,N) = |det(M^TN)|$*

Proof. We will prove that the function satisfies every condition in Definition 2.

For the first condition, by the property of determinant that determinant of a matrix yields the same value to determinant of its transposed matrix, we can conclude that the function is reflexive.

$$f(M,N) = |det(M^TN)| = |det((M^TN)^T)| = |det(N^TM)| = f(N,M).$$

For the second condition, we will prove that if the differences within two sets of linear spaces are equivalent then the function will provide the same value. Suppose that M and N be two sets of orthonormal basis vectors and O be an

orthogonal matrix.

$$\begin{aligned} f(OM, ON) &= |det((OM)^T ON)| \\ &= |det(M^T O^T ON)| \\ &= |det(M^T N)| \\ &= f(M, N). \end{aligned}$$

For the third condition, we will prove that any basis vectors which span the same space will yield the same result. Let O_M and O_N be orthogonal matrices with m rows. All the set of basis vectors which span the same space as M and N can be written as MO_M and NO_N.

$$\begin{aligned} f(MO_M, NO_N) &= |det((MO_M)^T NO_N)| \\ &= |det(O_M^T M^T NO_N)| \\ &= |det(O_M^T) det(M^T N) det(O_N)| \\ &= |det(O_M)||det(M^T N)||det(O_N)| \\ &= |det(M^T N)| \\ &= f(M, N). \end{aligned}$$

This results from the property of multiplicative distribution of determinant and from the fact that the determinant of an orthogonal matrix is ± 1 from its definition that an orthogonal matrix multiplying to its transposed matrix will result in an identity matrix. ■

To get a better understanding of the function in Proposition 6, we first consider the case that $m = 1$. The function $|det(M^T N)|$ will become the absolute value of dot product between two vectors. Its absolute value is the result from that the spanned linear space is in both positive and negative directions of the basis vectors.

Furthermore, we will consider $min(eig(N^T M_S N))$ where $eig(X)$ is the function that returns all of the eigenvalues of X. The minimum eigenvalue will reflect the result of using $\mathbf{k}$, a vector in N_S, which minimizes the function

$\mathbf{k}^T M_S \mathbf{k}$ for all the vectors in N_S. The function can be interpreted as projecting $\mathbf{k}$ onto M_S and projecting it back onto N_S. Hence, the eigenvalue will be $cos^2(\alpha)$ when α is the angle between $\mathbf{k}$ and M_S which also serves as the angular difference between N_S and M_S.

Proposition 7. *Let M and N be two sets of orthonormal basis vectors which span an m-dimensional linear space. The function $f(M,N) = |det(M^T N)|$ will have its maximum value if and only if M and N span the same linear space.*

Proof. We will state a proof that both sets of orthonormal basis vectors will span the same space if and only if the value of the function is 1. The sufficient condition uses the knowledge that has been previously applied.

$$\begin{aligned} f(M, MO_M) &= |det(M^T MO_M)| \\ &= |det(M^T M)||det(O_M)| \\ &= 1. \end{aligned}$$

For the necessary condition, let us consider the following equation:

$$\begin{aligned} f^2(M,N) &= |det(M^T N)|^2 \\ &= (det(M^T N))^2 \\ &= det(M^T N) det(M^T N) \\ &= det(M^T N) det(N^T M) \\ &= det(M^T N N^T M) \\ &= det(M^T N N^T M). \end{aligned}$$

By the property of determinant, its value will be equivalent to the multiplication of all singular values of the matrix. So if there is a pair of M and N that cause the value of $f(M,N) > 1$ then there will be at least an eigenvalue of $M^T N N^T M$ that is higher than 1 since $M^T N N^T M$ is a symmetric positive semidefinite matrix. Suppose that $\mathbf{k}$ is the unit eigenvector with the maximum eigenvalue of

$M^T NN^T M$, it will follow that the eigenvalue of $\mathbf{k}$ will be equal to:

$$\begin{aligned}\mathbf{k}^T M^T NN^T M\mathbf{k} &= (M\mathbf{k})^T NN^T (M\mathbf{k}), \\ &= (M\mathbf{k})^T N_S (M\mathbf{k}).\end{aligned}$$

Since M is an orthonormal basis vector and $\mathbf{k}$ is a unit vector, $M\mathbf{k}$ will be a unit vector that lies in the space spanned by M and $(M\mathbf{k})^T N_S (M\mathbf{k})$ can be viewed as dot product between the unit vector and the projection of that unit vector onto the space N_S. Since projection does not increase the size of vector, it is clear that $(M\mathbf{k})^T N_S (M\mathbf{k}) \leq 1$ and that 1 is the maximum value of $f(M, N)$ ■

From Proposition 7, the main contribution of the function $|det(M, N)|$ is that it can provide an answer if two sets of orthonormal basis vectors span the same space, without requiring any further interpretation. However, difference between two linear spaces cannot be fully described with one real value due to its degree of freedom.

Now, let us consider space $M_S = MM^T$ and $N_S = NN^T$. If the space N_S is exactly the same as M_S, then all orthonormal basis vectors of N_S must be in M_S. Therefore,

$$N^T M_S N = N^T N = I_m,$$

where I_m is an identity matrix with m rows.

The diagonal elements of $N^T M_S N$ will be 1 if and only if each basis vector of N_S in N is in M_S since M_S can be interpreted as projection of the vector onto that space. Moreover, if that is the case, then N_S will span the same space as M_S and the projection of each different orthonormal basis vector in N will still be orthogonal to each other. So we can conclude that $N^T M_S N = I_m$ if and only if $M_S = N_S$.

From Definition 2 which defines the constraint for functions that measure the difference between linear spaces from its orthonormal basis vectors. We can derive the constraints for functions that measure the difference between linear spaces using the matrix representing the spaces directly.

Proposition 8. *Let M_S and N_S be two m-dimensional linear spaces. A function $g(M_S, N_S)$ is a measure of difference between the spaces if and only if,*

i) $g(M_S, N_S) = g(N_S, M_S)$,

ii) $g(M_S, N_S) = g(OM_SO^T, ON_SO^T)$,

where O is any orthogonal matrix with n rows.

Proof. From Definition 2, we can obviously see that $g(AA^T, BB^T) = g(CC^T, DD^T)$ if and only if $f(A, B) = f(C, D)$.

For the first condition, assume the first condition in Definition 2.

$$\begin{aligned} f(M, N) &= f(N, M) \\ g(MM^T, NN^T) &= g(NN^T, MM^T) \\ g(M_S^T, N_S^T) &= g(N_S^T, M_S^T). \end{aligned}$$

For the second condition, assume the second condition in Definition 2.

$$\begin{aligned} f(M, N) &= f(OM, ON) \\ g(MM^T, NN^T) &= g(OM(OM)^T, ON(ON)^T) \\ g(MM^T, NN^T) &= g(OMM^TO^T, ONN^TO^T) \\ g(M_S, N_S) &= g(OM_SO^T, ON_SO^T). \end{aligned}$$

Assuming the third condition in Definition 2.

$$\begin{aligned} f(M, N) &= f(MO_M, NO_N) \\ g(MM^T, NN^T) &= g(MO_M(MO_M)^T, NO_N(NO_N)^T) \\ g(MM^T, NN^T) &= g(MO_MO_M^TM^T, NO_NO_N^TN^T) \\ g(MM^T, NN^T) &= g(MM^T, NN^T) \\ g(M_S, N_S) &= g(M_S, N_S). \end{aligned}$$

Since the third condition from Definition 2 is always true by the definition of the linear space, it needs no further consideration. ■

Proposition 9. *We can measure difference between two linear spaces M_S and N_S with $g(M_S, N_S) = eig(M_S + N_S)$ where $eig(X)$ is the sorted eigenvalues of matrix X.*

Proof. We will prove that the function $g(M_S, N_S) = eig(M_S + N_S)$ satisfies all the conditions in Proposition 8.

For the first condition, since matrix addition is associative, this is really obvious.

$$g(M_S, N_S) = eig(M_S + N_S) = eig(N_S + M_S) = g(N_S, M_S)$$

For the second condition,

$$\begin{aligned} g(OM_SO^T, ON_SO^T) &= eig(OM_SO^T + ON_SO^T), \\ &= eig(O(M_S + N_S)O^T). \end{aligned}$$

From the eigenvalue decomposition, let $M_S + N_S = V\Lambda V^T$.

$$\begin{aligned} g(OM_SO^T, ON_SO^T) &= eig(O(M_S + N_S)O^T) \\ &= eig(OV\Lambda V^T O^T) \\ &= eig((OV)\Lambda(OV)^T) \\ &= eig(\Lambda). \end{aligned}$$

The last line of the equation can be considered as extracting the eigenvalues from the matrix. In the same way,

$$\begin{aligned} g(M_S, N_S) &= eig(M_S + N_S), \\ &= eig(V\Lambda V^T), \\ &= eig(\Lambda), \\ &= g(OM_SO^T, ON_SO^T). \end{aligned}$$

■

Let us further analyse the function in Proposition 9. It is quite obvious that the function is better as a measurement of difference between two linear spaces than the function in Proposition 6, due to its output which has n degree of freedom when n is the number of dimensions. For the case that both spaces are the same:

$$\begin{aligned} eig(M_S + M_S) &= eig(2M_S), \\ &= 2eig(M_S), \end{aligned}$$

which means that it will have m eigenvalues which are 2 and the rest of $n - m$ eigenvalues from calculation will be zero.

Proposition 10. *Let M_S and N_S be two m-dimensional linear spaces. We can fully describe the difference between M_S and N_S with the matrix $D_{MN} = U^T M_S^T U$ when U^T is the eigenvectors of N_S sorted by its eigenvalues.*

Proof. Since the number of dimensions of M_S and N_S are equal, they will have same set of eigenvalues. As a result of eigenvalue decomposition, we have:

$$\begin{aligned} M_S &= V\Lambda V^T, \\ N_S &= U\Lambda U^T, \end{aligned}$$

where V and U are orthogonal matrices. From Proposition 8 the difference between M_S and N_S will be equal to the difference between Λ and $U^T M_S U$. Since Λ is also the eigenvalue part from eigenvalue decomposition of $U^T M_S U$. It follows that the difference between M_S and N_S can be fully described with the matrix $D_{MN} = U^T M_S U$.

Also, a function $h(D)$ can completely measure the difference from the difference matrix D with $h(D) = g(eig(D), D)$. By Proposition 8, we will have the following constraint: $h(D) = h(ODO^T)$ if and only if $Oeig(D)O^T = eig(D)$. ■

From Proposition 10, the first note is that $D_{MN} = U^T M_S U$ and $D_{NM} = V^T N_S V$ are not necessarily the same matrix but they both fully describe the differences which are equivalent to each other.

The constraint in Proposition 10 gives us some guideline in performing pairwise comparation of the difference. However, there is the following problems that given two symmetric positive semidefinite matrices A and B, how to efficiently compute an orthogonal matrix O such that $A = OBO^T$, and how to compute $h(D)$ that will yield the same output if and only if the difference are equivalent.

Theorem 1. *Let M_S and N_S be two m-dimensional linear spaces, the m-dimensional linear space X_S which is spanned by the eigenvectors with m highest eigenvalues of $M_S + N_S$ is the optimal mean of M_S and N_S in the sense that it will minimize the value $max_{\mathbf{k}^T M_S \mathbf{k}=1 \text{ or } \mathbf{k}^T N_S \mathbf{k}=1} \mathbf{k}^T Y_S \mathbf{k}$ when Y_S is any m dimensional linear space, meaning that it will minimize the angular difference between any vectors in M_S or N_S and the new linear space.*

Proof. From Proposition 3, it is enough to prove the result when m is not greater than $n/2$, since performing the process on the none important space will yield the same result. From this, we will be able to use the assumption that M_S and N_S do not intersect each other at any place other than the origin.

Let X_S be an m-dimensional linear space that serves as a mirror plane which transforms the space of M_S into N_S back and forth; i.e. for a unit vector $\mathbf{k}$ in M_S or N_S, there will be another unit vector $\mathbf{j}$ in the other linear space such that the result from both of their projection on X_S will be exactly the same but has the exactly opposite direction of projection. This can be visualized that the linear space X_S will act as a mirror between M_S and N_S. If $\mathbf{k}$ and $\mathbf{j}$ are both unit vectors that yield the value of angular difference between M_S and N_S then it can be seen that the angular difference between X_S and $\mathbf{k}$ will be equivalent to the angular difference between X_S and $\mathbf{j}$ since the projection from both of them on X_S will be at the same place. It becomes obvious that the angular difference between other pair of unit vectors in M_S and N_S will be lower than this value and thus this X_S is an optimal mean space calculated from M_S and N_S in the sense of angular difference.

Let X_S, M_S and N_S be created from their orthonormal bases X, M and N. From the previous definition of X_S, M_S and N_S as in the previous paragraph,

and let $\mathbf{k}$ be a vector in M_S, we can calculate $\mathbf{j}$, a vector in N_S corresponding to $\mathbf{k}$, which will result in the following equation:

$$\begin{aligned}\mathbf{k} &= X_S\mathbf{k} + (\mathbf{k} - X_S\mathbf{k}),\\ \mathbf{j} &= X_S\mathbf{k} - (\mathbf{k} - X_S\mathbf{k}),\\ &= 2X_S\mathbf{k} - \mathbf{k},\\ &= (2X_S - I_n)\mathbf{k}.\end{aligned}$$

This relies on the fact that M_S, N_S and X_S have the same number of dimensions and do not intersect each other. That is why for a vector in M_S, there should be a vector in N_S which has equal angular difference from X_S and will be projected on X_S at the same position but with exactly opposite direction of the projection. That is why we can write the space N_S as following:

$$\begin{aligned}N_S &= NN^T,\\ &= (2X_S - I_n)M((2X_S - I_n)M)^T,\\ &= (2X_S - I_n)MM^T(2X_S - I_n)^T,\\ &= (2X_S - I_n)M_S(2X_S - I_n).\end{aligned}$$

Now let us consider the matrix $M_S + N_S$.

$$\begin{aligned}M_S + N_S &= M_S + (2X_S - I_n)M_S(2X_S - I_n)\\ &= M_S + (4X_SM_SX_S - 2X_SM_S - 2M_SX_S + M_S)\\ &= 4X_SM_SX_S - 2X_SM_S - 2M_SX_S + 2M_S.\end{aligned}$$

Let $\mathbf{k}$ be an eigenvector of $M_S + N_S$. We will observe the result when separating

$\mathbf{k}$ into its elements that are in X_S and that are not.

$$\begin{aligned}
\mathbf{k} &= X_S\mathbf{k} + (I_n - X_S)\mathbf{k} \\
(M_S + N_S)\mathbf{k} &= (4X_SM_SX_S - 2X_SM_S - 2M_SX_S + 2M_S)(X_S\mathbf{k} + (I_n - X_S)\mathbf{k}) \\
&= ((4X_SM_SX_S - 2X_SM_S - 2M_SX_S + 2M_S)X_S\mathbf{k}) \\
&\quad +((4X_SM_SX_S - 2X_SM_S - 2M_SX_S + 2M_S)(I_n - X_S)\mathbf{k}) \\
&= ((4X_SM_S - 2X_SM_S - 2M_S + 2M_S)X_S\mathbf{k}) \\
&\quad +((-2X_SM_S + 2M_S)(I_n - X_S)\mathbf{k}) \\
&= 2X_SM_SX_S\mathbf{k} + 2(I_n - X_S)M_S(I_n - X_S)\mathbf{k}).
\end{aligned}$$

Suppose that $\mathbf{k}$ has elements in both X_S and in the space orthogonal to X_S, which is the $(n-m)$-dimensional linear space that does not intersect with X_S except at the origin, denoted by $X_{S\perp} = I_n - X_S$. If there is only one optimal X_S by the definition then the angular difference between M_S and N_S must be less than $\pi/2$ which causes the angular difference between any vectors in M_S and X_S to be less than $\pi/4$ while also causes the angular difference between any vectors in M_S and $X_{S\perp}$ to be higher than $\pi/4$. For $\mathbf{k}$ to be an eigenvector of $M_S + N_S$, $X_SM_SX_S\mathbf{k}$ must be in the same direction as $X_S\mathbf{k}$ and $X_{S\perp}M_SX_{S\perp}\mathbf{k}$ must be in the same direction as $X_{S\perp}\mathbf{k}$. If we let the angular difference between M_S and $X_S\mathbf{k}$ and between M_S and $X_{S\perp}\mathbf{k}$ be α and β respectively then we will get the following equations:

$$\begin{aligned}
(M_S + N_S)\mathbf{k} &= 2X_SM_SX_S\mathbf{k} + 2(I_n - X_S)M_S(I_n - X_S)\mathbf{k}) \\
&= 2X_SM_SX_S\mathbf{k} + 2X_{S\perp}M_SX_{S\perp}\mathbf{k}) \\
&= 2cos^2(\alpha)X_S\mathbf{k} + 2cos^2(\beta)X_{S\perp}\mathbf{k})
\end{aligned}$$

Since $0 < \alpha < \beta < \pi/2$, it follows that either $X_S\mathbf{k}$ or $X_{S\perp}\mathbf{k}$ must be the zero vector. This means that eigenvectors of $M_S + N_S$ must lie in either X_S or $X_{S\perp}$ and since X_S is m-dimensional then there will be m eigenvectors in X_S and $(n-m)$ eigenvectors in $X_{S\perp}$.

In the same way as M_S, angular difference between any vectors in X_S and N_S must be less than $\pi/4$ and angular difference between any vectors in X_S

and $N_{S\perp}$ must be higher than $\pi/4$. As a result, for any vector $\mathbf{k}$ in X_S:

$$\begin{aligned}\mathbf{k}^T(M_S + N_S)\mathbf{k} &= \mathbf{k}^T M_S \mathbf{k} + \mathbf{k}^T N_S \mathbf{k}, \\ &> 2cos^2(\pi/4), \\ &= 1, \\ &> \mathbf{p}^T(M_S + N_S)\mathbf{p},\end{aligned}$$

where $\mathbf{p}$ is a vector in $X_{S\perp}$. This means that the m eigenvectors with highest eigenvalues will all lie in X_S and are also orthonormal basis vectors. Thus, the linear space resulted from the process is the optimal linear space that minimizes the value of $max_{\mathbf{k}^T M_S \mathbf{k}=1 \text{ or } \mathbf{k}^T N_S \mathbf{k}=1} \mathbf{k}^T Y_S \mathbf{k}$ when Y_S is any m dimensional linear space. ■

In the case that the user want to weight M_S and N_S in a meaningful way, it seems that the most sensible way is to repeat using the algorithm to find the mean of two linear spaces until it reaches the satisfying precision, instead of providing the weight to the linear space directly. The algorithm for the weighted version of both spaces is shown in Algorithm 7.

input : M, N, m, α

output: X

$A \leftarrow M, B \leftarrow N, X \leftarrow A, a \leftarrow 0, b \leftarrow 1, x \leftarrow 0;$

while $|\alpha - x| > \beta$ **do**

  $X \leftarrow PCA(AA^T + BB^T, m);$

  $x \leftarrow (a+b)/2;$

  if $\alpha > x$ **then**

    $a \leftarrow x;$

    $A \leftarrow X;$

  else

    $b \leftarrow x;$

    $B \leftarrow X;$

  end

end

Algorithm 7: Finding the optimal mean between two linear spaces with weight.

In Algorithm 7, $PCA(\mathrm{X}, m)$ is the result of using PCA to choose m dimensions from covariance matrix A_S, α is a real value in the range $[0, 1]$ which is the weight for N_S while the weight M_S will be $1 - \alpha$, β is the constraint for precision in calculation for the result, and m is the dimension for both M and N.

Note that in the case that M_S and N_S are obtained by the linear dimensionality reduction of the same dataset, but with a different algorithm, the process of combining them will act as finding the linear space that is good in both objectives.

Theorem 2. *Using the proposed framework of customization for dimensionality reduction in Figure 5.1 on linear dimensionality reduction will yield the optimal result if we can estimate the distribution of data perfectly.*

Proof. Let M_S and N_S be both spaces to be combined which have M and N as sets of their orthonormal basis vectors. According to the framework, to combine two nonlinear spaces we will have to use the results from mapping customization data on both of them in order to estimate the shape of the nonlinear spaces. However, in the case of linear space, the shape of the space can be fully

described with the matrix and we could estimate the distribution from all the data which are used to generate both spaces by the set of all available data in the spaces. Since the mapping is linear, we can represent the result from mapping the data of all input spaces with identity matrix and its negative. Thus, we get the dataset as $[M|N]^T[I_n| - I_n]$. The mean of this dataset is clearly zero due to that the vectors resulted from $[M|N]^T[I_n]$ will come with their negative vectors $[M|N]^T[-I_n]$. The next step is to perform PCA using this dataset to obtain the following covariance matrix.

$$\begin{aligned}
[M|N]^T[I_n| - I_n]([M|N]^T[I_n| - I_n])^T &= [M|N]^T[I_n| - I_n][I_n| - I_n]^T[M|N] \\
&= [M|N]^T(I_n I_n^T + (-I_n)(-I_n)^T)[M|N] \\
&= [M|N]^T(2I_n I_n^T)[M|N] \\
&= 2[M|N]^T[M|N].
\end{aligned}$$

Since scaling of a matrix will not affect its eigenvectors, we will instead use the matrix $[M|N]^T[M|N]$ which can also be calculated as covariance matrix of the dataset $[M|N]^T$ with its mean fix to zero. By further analysis of the matrix:

$$\begin{aligned}
[M|N]^T[M|N] &= \begin{bmatrix} M^T \\ N^T \end{bmatrix} \begin{bmatrix} M & N \end{bmatrix}, \\
&= \begin{bmatrix} M^TM & M^TN \\ N^TM & N^TN \end{bmatrix}, \\
&= \begin{bmatrix} I_m & M^TN \\ N^TM & I_m \end{bmatrix}.
\end{aligned}$$

Now, let us consider a vector $\mathbf{k}$ which is an eigenvector of the matrix $MM^T + NN^T$ with eigenvalue λ.

$$
\begin{aligned}
\begin{bmatrix} I_m & M^TN \\ N^TM & I_m \end{bmatrix} \begin{bmatrix} M^T \\ N^T \end{bmatrix} \mathbf{k} &= \begin{bmatrix} M^T + M^TNN^T \\ N^TMM^T + N^T \end{bmatrix} \mathbf{k} \\
&= \begin{bmatrix} M^T\mathbf{k} + M^TNN^T\mathbf{k} \\ N^T\mathbf{k} + N^TMM^T\mathbf{k} \end{bmatrix} \\
&= \begin{bmatrix} M^T\mathbf{k} + M^T(\lambda\mathbf{k} - MM^T\mathbf{k}) \\ N^T\mathbf{k} + N^T(\lambda\mathbf{k} - NN^T\mathbf{k}) \end{bmatrix} \\
&= \begin{bmatrix} M^T\mathbf{k} + \lambda M^T\mathbf{k} - M^TMM^T\mathbf{k} \\ N^T\mathbf{k} + \lambda N^T\mathbf{k} - N^TNN^T\mathbf{k} \end{bmatrix} \\
&= \begin{bmatrix} M^T\mathbf{k} + \lambda M^T\mathbf{k} - I_mM^T\mathbf{k} \\ N^T\mathbf{k} + \lambda N^T\mathbf{k} - I_mN^T\mathbf{k} \end{bmatrix} \\
&= \begin{bmatrix} M^T\mathbf{k} + \lambda M^T\mathbf{k} - M^T\mathbf{k} \\ N^T\mathbf{k} + \lambda N^T\mathbf{k} - N^T\mathbf{k} \end{bmatrix} \\
&= \begin{bmatrix} \lambda M^T\mathbf{k} \\ \lambda N^T\mathbf{k} \end{bmatrix} \\
&= \lambda \begin{bmatrix} M^T \\ N^T \end{bmatrix} \mathbf{k}.
\end{aligned}
$$

From the above equation, if a vector $\mathbf{k}$ is an eigenvector of $MM^T + NN^T$ with eigenvalue λ then $[M|N]^T\mathbf{k}$ will be an eigenvector of $[M|N]^T[M|N]$ with eigenvalue λ. Since the matrix $[M|N]^T[M|N]$ is multiplication between $[M|N]$ and its transpose, the rank of $[M|N]^T[M|N]$ will be equal to the rank of $[M|N]$, and thus its maximum value will be the lesser between its number of rows and columns which are n and $2m$ respectively. The result is that in the case that the number of rows of $[M|N]^T[M|N]$ is higher than n the other eigenvectors which are not corresponding to any $[M|N]^T\mathbf{k}$ when $\mathbf{k}$ is an eigenvector of $MM^T + NN^T$ will have zero eigenvalue. One may say that the matrix will have no other eigenvector. So a vector $\mathbf{k}$ will be the eigenvector of $MM^T + NN^T$ with the i-th maximum eigenvalue if and only if $[M|N]^T\mathbf{k}$ is the eigenvector of $[M|N]^T[M|N]$ with the i-th maximum eigenvalue. Hence, the space resulted from performing PCA on $MM^T + NN^T$ will be the same as the space

resulted from performing PCA on $[M|N]^T[M|N]$ after the vector is mapped onto $[M|N]^T$, that is the result from applying the framework for dimensionality reduction on linear dimensionality reduction. Thus, we can say that using our proposed framework for dimensionality reduction on linear dimensionality reduction will yield the optimum result. ■

5.3 Numerical Results

We demonstrate the performance of our proposed algorithm with some datasets from the UCI Machine Learning Repository, whose details are shown in Table 5.1. In the same way to all of our previous experiments, the data were normalized into the range of $[0, 1]$. After that, we separated the data into 4 groups of equal size and performed cross-validation among them. Two of the groups were used as original training data to train the given linear dimensionality reduction process, while the other two of them were used as customization data and test data, resulted in $\frac{4!}{2!1!1!} = 12$ combinations in total. The result was compared with the result from original linear dimensionality reduction and the one trained from customization data.

Table 5.1: Detail of all datasets used in the experiments of combining results from linear dimensionality reduction. The numbers of attributes shown are the numbers of attributes used in dimensionality reduction, omitting some useless attributes like index or specific name.

	No. of attributes	No. of instances
concrete compress	8	1,030
concrete slump	7	103
cpu	6	209
forest	12	517
housing	13	506
parkinsons	19	5,875
servo	4	167

For the linear dimensionality reduction used in the experiments, if one want to use some algorithms with clearly defined objective as the value in optimization then the score can be easily measured from the transformed test data in the same way. However, in our experiments, we used PCA. Since PCA sorts basis vectors for linear space by variance of the data in each axis, the objective

is not well-ordered. So as in many research works, we decide to use the value of determinant and trace of covariance matrix which are the values of products and summation of eigenvalues of the matrix respectively. The perfect result of PCA then will align the axis so that all non-diagonal elements of the covariance matrix will be zero. Since we calculate the mean of the value from cross validation, we will use the natural logarithmic value of determinant instead of using them directly to retain the sense of arithmetic mean. The result is shown in Table 5.2.

Table 5.2: Result from using the proposed frameworks with linear dimensionality reduction. Numbers whose values are minimal in their rows are typeset bold. The second column shows the reduced dimension. All of the equal values are different in higher precision.

dataset	m	logdet(G)	logdet(S)	logdet(N)	trace(G)	trace(S)	trace(N)
concrete	1	3.325	3.182	**3.346**	28.344	25.128	**28.996**
compress	3	7.957	**8.191**	8.190	56.180	57.380	**57.707**
	5	11.452	**12.434**	12.257	78.178	**81.335**	80.777
	7	**12.952**	12.889	12.944	**86.828**	86.567	86.810
concrete	1	1.022	1.030	**1.093**	2.856	2.888	**3.024**
slump	3	1.969	**2.372**	2.349	7.322	7.647	**7.757**
	5	0.909	**1.914**	1.636	9.703	**10.269**	10.035
cpu	1	1.279	1.241	**1.294**	3.782	3.660	**3.843**
	3	0.559	0.555	**0.566**	5.623	5.607	**5.663**
	5	-1.609	-1.679	**-1.575**	6.968	6.965	**6.989**
forest	1	3.007	3.000	**3.012**	20.251	20.114	**20.352**
	3	8.279	8.253	**8.282**	48.808	48.411	**48.855**
	5	11.588	11.519	**11.600**	59.888	59.526	**59.938**
	7	13.757	13.720	**13.772**	66.417	66.256	**66.474**
	9	**13.357**	12.794	13.328	68.430	68.262	**68.442**
housing	1	3.947	3.943	**3.948**	51.814	51.602	**51.872**
	3	8.512	8.475	**8.516**	72.337	71.891	**72.386**
	5	11.754	11.638	**11.769**	82.914	82.307	**83.033**
	7	14.132	14.059	**14.152**	90.018	89.703	**90.115**
	9	**15.172**	14.859	15.070	**94.082**	93.572	93.974
parkinsons	1	5.775	5.774	**5.775**	322.042	321.963	**322.063**
	3	15.148	15.145	**15.148**	545.028	544.694	**545.090**
	5	22.983	22.981	**22.983**	647.335	647.225	**647.361**
	7	28.320	28.315	**28.321**	676.847	676.749	**676.868**
	9	31.271	31.267	**31.272**	686.197	686.153	**686.208**
servo	1	2.065	2.040	**2.085**	7.993	7.829	**8.180**
	3	5.349	5.346	**5.351**	19.025	19.005	**19.032**

It is clearly shown in Table 5.2 that the result from combining both linear

spaces is better than that from the covariance matrix of mapped data, measured by the value of logarithm of determinant and trace which both reflect the eigenvalues of the covariance matrix. Ones might think that combining two models is similar to the idea of customization for classification by patching, but there are some significant differences. The first thing is that combining two linear spaces with this algorithm performs with both of them on equal ground, while the patching approach in customization for classification treats them differently. The second issue is that, customization for classification by patching predicts the result using one of the two models, and with plausible bias, the result is expected to be an improvement. However, combining linear spaces based on the hypothesis that both of these spaces should be the same but is differed due to variation or error in both datasets. Thus, the most likely best linear space is the linear space taken as a mean of them and the improvement in the experimental result is based on different reason which is less obvious than customization for classification by patching.

In order to demonstrate that this algorithm is applicable for dimensionality reduction that will be later used in classification, we conducted other experiments by performing customization on dimensionality reduction before classification. The experiments were conducted on datasets *optdigits* and *pendigits* whose data were normalized into the range of $[0, 1]$ as in all of our previous experiments. However, the dataset *optdigits* used in this experiment did not have dimensionality reduced by PCA since this experiment was about dimensionality reduction as well, and thus dataset *optdigits* had 64 attributes. We performed 10-fold cross validation between data used for customization and test data. We used linear discriminant analysis as dimensionality reduction algorithm and 1-nearest neighbor for classification. The result is shown in Table 5.3.

In Table 5.3, for dataset *optdigits*, we can be quite certain that the proposed algorithm yielded better result, but for *pendigits*, it was only the case that the data was reduced to one dimension that was an improvement from both linear spaces. The main difference between both of these datasets is the numbers of attributes; *optdigits* has 64 attributes, *pendigits* has 16 attributes. Therefore, performing dimensionality reduction on *optdigits* with the same number of dimen-

Table 5.3: Classification accuracy after performing customization upon dimensionality reduction. (G) and (S) are the results from original data and customization data respectively, and (N) is the one created from both models with our algorithm.

dataset	m	LDA(G)	LDA(S)	LDA(N)
optdigits	1	33.28 ± 4.03	**33.61 ± 6.01**	32.23 ± 5.09
	3	77.69 ± 2.98	78.19 ± 3.63	**78.58 ± 3.19**
	5	87.65 ± 3.01	88.70 ± 4.74	**89.15 ± 3.01**
	7	93.32 ± 2.12	92.65 ± 3.26	**93.82 ± 2.43**
	9	95.55 ± 1.70	95.60 ± 2.35	**95.83 ± 1.91**
pendigits	1	39.45 ± 2.11	38.88 ± 4.16	**40.56 ± 3.33**
	3	**81.22 ± 2.26**	78.33 ± 1.74	80.99 ± 2.48
	5	**93.45 ± 1.41**	91.54 ± 1.05	91.57 ± 1.10
	7	**97.54 ± 0.88**	97.51 ± 0.93	97.05 ± 0.77
	9	98.43 ± 1.14	**98.69 ± 0.92**	98.31 ± 0.61

sions yielded the better result due to having larger numbers of choice, while it might be hard to determine a good projection for *pendigits*. Also note that the ratio of the number of data used in calculating LDA(G) to the number of data used in calculating LDA(S) was about twice for both *optdigits* and *pendigits*. The other noteworthy issue is where the value of each attributes of *optdigits* and *pendigits* came from. A value in each attributes of *optdigits* corresponds to the grayscale color in each pixel of an 8x8 image, while the attributes of *pendigits* has various meaning and many different interpretation. By the nature of linear dimensionality reduction, it is better fit to the task that each attribute has similar meaning like *optdigits* since the underlying distribution of the data will more likely be in linear subspace and yield the suitable result upon them. One might argue with the previous statement about linear dimensionality reduction being better on *optdigits* than *pendigits* using the accuracy when both of them are reduced to the same number of dimensions as shown in Table 5.3 that the value in *pendigits* is higher and both of them has the same numbers of classes corresponding to the numbers 0-9. This was likely resulted from that the data obtained directly from pixels of *optdigits* are distributed sparsely in high dimensional linear space while data obtained from feature extraction of *pendigits* lie densely in lower dimensions with more complicated distribution.