Chapter 4
PCA Based Unsupervised FE

There is no sound that I do not need.
Rio Kazumiya, Sound of the Sky, Season 1, Episode 3

4.1 Introduction: Feature Extraction vs Feature Selection

In this chapter, I mainly discuss about the situation where feature extraction or feature selection is inevitable. When or under what kind of conditions, do we need either or both of two? Here are some examples of such situations.

- **Case 1**: The number of features attributed to individual samples is larger than the number of samples.
- **Case 2**: Features attributed to individual samples are not independent of one another.
- **Case 3**: Some of the features attributed to samples are not related to some properties that we would like to relate features to.

Although these above three cases are not comprehensive, they are good examples by which we can discuss the reason why we need feature extraction and/or feature selection. An example of case 1 is linear equations that can be represented as $A\boldsymbol{x} = \boldsymbol{b}$ where $A \in \mathbb{R}^{N\times M}, \boldsymbol{x} \in \mathbb{R}^{M}, \boldsymbol{b} \in \mathbb{R}^{N}$ and $\boldsymbol{x}$ represents variables, A represents coefficients, and $\boldsymbol{b}$ represents constants. When $N < M$, not only there are no unique solutions, but also there are always solutions, even when A and $\boldsymbol{b}$ are purely random numbers. The fact that there are no unique solutions prevents us from interpreting outcome, because there can be multiple distinct unique solutions. The fact that there are always solutions means that there might be meaningless solutions. In this case, we need feature extraction and/or feature selection such that we can have limited number of features that is smaller than the number of samples. An example of case 2 is multicollinearity. In this case, although apparently, $A\boldsymbol{x} = \boldsymbol{b}$ is uniquely solvable, it is actually not because coefficient matrix A is not regular (in other words, row vectors are not independent of one another). In this case, we need to apply feature extraction or feature selection in order to obtain reduced number of features that

Y-h. Taguchi, *Unsupervised Feature Extraction Applied to Bioinformatics*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-22456-1_4

enables us to get unique solutions. An example of case 3 is that some elements of A are zero. Especially, if A includes column vectors totally filled with zero, variables that correspond to these columns are not related to $\boldsymbol{b}$ at all. When A is given, we can simply discard these variables. Nevertheless, when A is required to be inferred from $\boldsymbol{x}$ and $\boldsymbol{b}$ (e.g., linear regression analysis), it is impossible to exclude there variables in advance. This might result in the incorrect estimation of A. In this case, we need feature selection that enables us to exclude variables not related to $\boldsymbol{b}$ in advance.

From these examples, we can know that the need of feature selection and feature extraction is very ubiquitous. So, the next question is which strategy is better to address these problems. Unfortunately, the answer is highly context dependent and cannot be decided based upon mathematical considerations. For example, let us consider image analysis, e.g., face recognition. In this case, it is rather obvious that not all pixels of digital images but only a limited number of them is useful for the purpose. If small number of features generated from large number of pixels work well, there is no need to go further. On the other hand, suppose that the problem is the inference about bankruptcy, in other words, the prediction of who will bankrupt. In this case, even if a newly generated feature composed of numerous personal information, e.g., income, age, education history, address, and so on, works pretty well, it might not be a final goal. This is because collecting these information might cost or is impossible at all. If another feature composed of more limited number of features works, even if the performance is a little bit less, another one might be employed because of easiness to use. Thus, it is inevitable to specify situation that we want to discuss.

As for the targeted field, I would like to say that the targeted field is bioinformatics as the title of this book says. In bioinformatics analysis, it is very usual that feature selection is more favorable than feature extraction because of the following reasons. In bioinformatics analysis (or in biology although it means the same), measuring individual features often costs. Thus, measuring less number of features can reduce the cost spent to individual observations. This results in the increased number of observations that often leads to better outcome. Even when measuring individual features does not cost, e.g. in the case of high throughput measurements, feature selection is often better than feature extraction, because each feature has its own meaning. For example, if features are genes, the selected limited number of genes are more interpretable than features generated by the combination of large number of genes. Thus, in the following I assume the situation where feature selection is more favorable than feature extraction even if not explicitly denoted.

4.2 Various Feature Selection Procedures

Although there are various ways to classify numerous number of previously proposed feature selection procedures, I would like to employ the one shown in Table 4.1. Feature selection strategies can be classified into two groups in two ways. One way is supervised ones vs unsupervised ones. Not to mention, supervised ones

Table 4.1 Classification of feature selections

	One by one	Collective
Supervised	Statistical tests[a]	Random forest, LASSO
Unsupervised	Highly variable genes, bimodal genes	PCA based unsupervised FE

[a] t test, limma, SAM

are definitely more popular than unsupervised ones. This is because the purpose of feature selection is usually purpose oriented. For example, if the study aims to investigate diseases, it is natural to consider genes expressed differently between patients and healthy controls. If the study aims to predict who will bankrupt, it is reasonable to consider features related to something financial. On the other hand, unsupervised feature selection might sound self-discrepancy, because it is unlikely possible to select features without any clear purposes. In spite of that, unsupervised feature selection is still possible. For example, it is natural to select features with maximum variance, because large variance might reflect the ability of the feature that represents diverse categories hidden in the considered sample. Thus, although it is less popular, unsupervised feature selection is still possible. Another way to classify feature selection strategies is one by one vs collective. The former means that feature selection is performed without the consideration of interaction between features. For example, when conventional statistical tests are applied to a feature of samples composed of two categories, the P-value that rejects the null hypothesis that a feature of members of two samples obeys the same distribution is computed. Then, if P value is small enough, say less than 0.01, the feature is identified as distinct between two categories. This means that each P-value attributed to each feature is not affected by other features at all. On the other hand, the latter considers the interaction between features. For example, when dummy variables are attributed to each of two categories, we can make linear regression using arbitrary number of features to predict dummy variables. In this case, the interaction between features included into regression equation is considered. Then, features used to construct regression equation with good performance are selected.

In order to demonstrate how differently feature selections that belong to four categories listed in Table 4.1 work, I prepare two synthetic data sets. Both are matrices $x_{ij} \in \mathbb{R}^{N\times M}$ where i and j correspond to features' index and samples' index, respectively. In both data sets, the only first $N_1(< N)$ features, $x_{ij}, i \leq N_1$, are distinct between two classes where $j \leq \frac{M}{2}$ and $j > \frac{M}{2}$ belong to the first and second class, respectively. x_{ij} is also drawn from Gaussian or mixed Gaussian distribution where $\mathcal{N}(\mu, \sigma)$ represents Gaussian distribution that has mean of μ and standard deviation σ, respectively.

- Data set 1:

$$x_{ij} \sim \begin{cases} \mathcal{N}(0,\sigma) & j \leq \frac{M}{2},\ i \leq N_1 \\ \mathcal{N}(\mu_0,\sigma) & j > \frac{M}{2},\ i \leq N_1 \\ \frac{1}{2}\mathcal{N}(0,\sigma) + \frac{1}{2}\mathcal{N}(\mu_0,\sigma) & i > N_1. \end{cases} \tag{4.1}$$

- Data set 2:

$$x_{ij} \sim \begin{cases} \mathcal{N}(0, \sigma) & j \leq \frac{M}{2},\ i \leq N_1 \\ \mathcal{N}(\mu_0, \sigma) & j > \frac{M}{2},\ i \leq N_1 \\ \mathcal{N}(\mu_1, \sigma) & i > N_1. \end{cases} \tag{4.2}$$

Thus, the only difference between two synthetic data sets is if the $N - N_1$ features (i.e., $i > N_1$) not distinct between two classes are drawn from bimodal [Eq. (4.1)] or unimodal [Eq. (4.2)] distributions. Specifically, $N = 100$, $M = 20$, $\mu_0 = 4$, $\mu_1 = \frac{\mu_0}{2} = 2$, $N_1 = 10$ and $\sigma = 1$ in the following. Performance is averaged over one hundred independent trials. The number of features distinct between two categories, N_1, is assumed to be known in advance. μ_1 is selected such that the sample mean of ith feature, $\langle x_{ij} \rangle_j$ defined by Eq. (2.56), does not differ between two models.

The statistical tests used belong to either of four categories. t test is employed as a representative of one by one, supervised feature selection. P values computed by t test are attributed to individual features. Top N_1 features with smaller P values are selected. As a representative of collective supervised feature selection, linear regression is employed. The dummy variable $y_j \in [0, 1]^M$ is given such that $y_j = 0, j \leq \frac{M}{2}$ and $y_j = 1, j > \frac{M}{2}$. Then using regression coefficient vector, $a_i \in \mathbb{R}^N$, $X\boldsymbol{a} = \boldsymbol{y}$ is assumed. $\boldsymbol{a}$ is computed with $\boldsymbol{a} = X^\dagger \boldsymbol{y}$ using Moore-Penrose pseudoinverse, $X^\dagger$, because there are no unique solutions due to $N > M$. Top N_1 features with larger absolute a_i are selected. As for representatives of one by one, unsupervised feature selections, two methods are employed. One is highly variable features. Sample variance of each feature,

$$\frac{1}{M} \left(x_{ij} - \frac{1}{M} \sum_{j=1}^{M} x_{ij} \right)^2, \tag{4.3}$$

is computed and top $N_1 = 10$ features associated with larger variance are selected. Another is unimodal test. Unimodal test computes P-values that reject the null hypothesis that x_{ij}s with fixed i are drawn from unimodal distribution; Hartigan's dip test, which rejects the null hypothesis that the distribution is unimodal [1] is used for this purpose. Then top $N_1 = 10$ features associated with smaller P-values are selected. Finally, as a representative of collective unsupervised feature selections, we employ PCA. PCA is applied to x_{ij} such that kth PC score vectors, $\boldsymbol{u}_k \in \mathbb{R}^N$, are attributed to features. In other words, $\boldsymbol{u}_k$ is computed as the eigenvectors of $S_{ii'}$, Eq. (2.50), $S_{ii'}\boldsymbol{u}_k = \lambda_k \boldsymbol{u}_k$ where λ_k is eigenvalue. Then, top $N_1 = 10$ features associated with the larger absolute first PC score, $|u_{1i}|$, are selected (the reason why this procedure works as feature selection will be discussed later).

Table 4.2 shows the number of features that are distinct between two classes and are also selected by individual methods. When tests are applied to data sets 1 and 2, two supervised methods samely achieved well although the collective method achieved a little bit worse than one by one method. The performance achieved by

Table 4.2 Performance of statistical tests applied to two synthetic data set 1 defined by Eq. (4.1) and data set 2 defined by Eq. (4.2)

	Supervised		Unsupervised		
	One by one	Collective	One by one		Collective
Data set	*t* test	Linear regression	Variance	Unimodal test	PCA
1	10.00	9.88	1.20	1.68	8.75
2	10.00	9.79	9.99	5.68	10.00
1 (shuffled)	1.03	0.08	1.34	1.66	8.78
2 (shuffled)	0.94	0.89	10.00	5.76	10.00

Numbers represent mean number of features selected by each method, among N_1 features distinct between two classes, $i < N_1 (= 10)$. Shuffled means that class labels are shuffled

unsupervised method is quite distinct between two data sets. Two unsupervised one by one methods fail when data set 1 is considered while they performed better for data set 2. This is reasonable because all N features obey the identical distribution if class labels are not considered. Thus, unsupervised methods have no ways to distinguish features with and without distinction between two classes. In this sense, it is remarkable that PCA, an unsupervised and collective method, can perform similarly well for both data sets 1 and 2.

One might wonder why unsupervised method must be considered, because supervised methods perform better. This impression changes once the class labels are shuffled. It is reasonable that no supervised methods work well. On the other hand, it is also reasonable that the performance by unsupervised method does not change because of class label shuffling. This suggests that unsupervised feature selections are better choices when class labels are not available or not trustable.

Unsupervised collective feature selection, PCA, is successful for data set 1, for which other unsupervised methods fail, and shuffled data set, for which supervised collective methods fail. It is important why it can happen. In order to see this, we investigate the first PC loading vectors, $\boldsymbol{v}_1 \in \mathbb{R}^M$, which is defined as $\boldsymbol{v}_1 = \frac{1}{\lambda_1} X^T \boldsymbol{u}_1$ (see Eq. (2.21)). Figure 4.1 shows the first PC loading vectors. For all cases, u_{ij}s with $j \leq \frac{M}{2}$ take positive values while u_{ij}s with $j > \frac{M}{2}$ take negative value. Since $\boldsymbol{u}_1 = \lambda_1 X \boldsymbol{v}_1$, u_{1i} reflects the difference between two classes. Thus, selecting is associated with absolutely larger u_{1i} can identify correctly features associated with distinction between two classes for all four cases. This is the reason why PCA can always perform well.

4.3 PCA Applied to More Complicated Patterns

In the previous section, feature selection with two classes was discussed. Nevertheless, it is the simplest case. There are many more complicated feature selections. One direction is to have more classes than two. Another direction is to have more than one classifications simultaneously. Here, let us discuss both together,

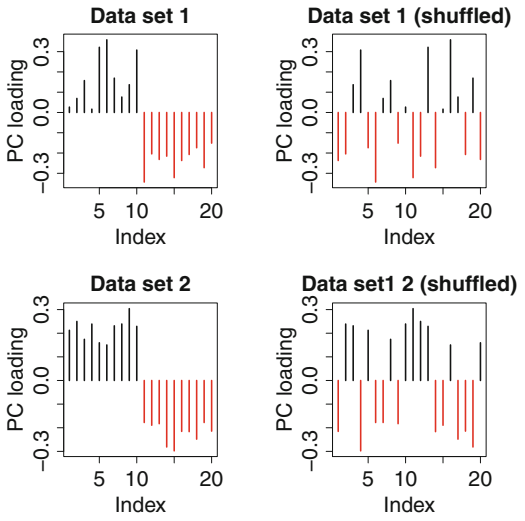

Fig. 4.1 The first PC loading vectors, $\boldsymbol{v}_1 \in \mathbb{R}^M$, for data set 1, shuffled data set 1, data set 2, and shuffled data set 2. Black and red bars correspond to classes 1 and 2, respectively

i.e., feature extraction under the conditions having more than one classification with more than two classes. In order to demonstrate feature selections under this condition, we extend data set 2, Eq. (4.2), as follows.

Data set 3

$$x_{ij} \sim \begin{cases} \mathcal{N}(0,\sigma) & j \le \frac{M}{2}, & i \le N_1 \\ \mathcal{N}(\mu_0,\sigma) & j > \frac{M}{2}, & i \le N_1 \\ \mathcal{N}(0,\sigma) & j \le \frac{M}{4}, & N_1 < i \le N_1 + N_2 \\ \mathcal{N}(\mu_1,\sigma) & \frac{M}{4} < j \le \frac{M}{2}, & N_1 < i \le N_1 + N_2 \\ \mathcal{N}(2\mu_1,\sigma) & \frac{M}{2} < j \le \frac{3M}{4}, & N_1 < i \le N_1 + N_2 \\ \mathcal{N}(3\mu_1,\sigma) & j > \frac{3M}{4}, & N_1 < i \le N_1 + N_2 \\ \mathcal{N}(\mu_2,\sigma) & & i > N_1 + N_2. \end{cases} \tag{4.4}$$

Features $i \le N_1$ are composed of two classes, those $N_1 < i \le N_1 + N_2$ are composed of four classes, and those $i > N_1 + N_2$ are composed of no classes. Thus the feature selection aims to identify which features are composed of how many classes.

Now the problem is more difficult. For example, simply trying to identify which features are composed of two classes does not help us to distinguish between features composed of two classes and those composed of four classes, because four classes can be also considered to be two classes if each two of four classes are considered as one class. Thus in order to perform feature selections under such a complicated condition, we usually need more detailed information about class labeling.

It is not very easy to adapt to this situation. Suppose that we have already known 20 samples classified into the four classes as

$$(A, A, A, A, A, B, B, B, B, B, C, C, C, C, C, D, D, D, D, D) \tag{4.5}$$

or into the two classes as

$$(E, E, E, E, E, E, E, E, E, E, F, F, F, F, F, F, F, F, F, F). \tag{4.6}$$

Even if this is the case, identification of features with four classes is not straightforward. Simple linear regression analysis is not applicable, because we know only that four classes differ from one another. In order to perform linear regression analysis, we need to assign numbers to each of four classes. If we do not know practical relationship between four classes, there are no ways to assign numbers to four classes. Pairwise comparison between four classes might be possible, but might not work well, because we need to integrate pairwise comparisons in order to rank features. Suppose we try all possible six pairwise comparisons in Eq. (4.5), as

$$(A, B), (A, C), (A, D), (B, C), (B, D), (C, D). \tag{4.7}$$

If we consider this is occasionally applied to Eq. (4.6), they correspond to comparisons of

$$(E, E), (E, F), (E, F), (E, F), (E, F), (F, F). \tag{4.8}$$

Thus, in contrast to the expectation, four out of six comparisons will report that they differ. Thus, if difference between two classes, E and F, is greater than that between pairs in four classes, A, B, C, and D, integration of six pairwise comparison might report that Eq. (4.8) more fits to four classes than Eq. (4.7). In the following, we consider occasions where integration of six pairwise comparisons occasionally report that Eq. (4.8) is more likely to be four classes than Eq. (4.7). For the simplicity, we assume that all pairwise comparisons (E,F) in Eq. (4.8) are higher ranked than all pairwise comparisons in Eq. (4.7). The requirement that difference between two classes among four classes should be smaller than that among two classes is not unrealistic. It is very usual that values of features have both upper and lower boundary. In this case, the distinction between two classes when samples are classified into two classes is that between the upper and the lower halves. On the other hand, the distinction between two classes when samples are classified into four classes is that between any pairs of four quantiles. If region is divided into two, the distinction is larger than that when region is divided into four. In this case, the following happens (Table 4.3). Four pairwise comparisons (E,F) in Eq. (4.8) is always higher ranked than corresponding four pairwise comparisons, (A,C), (A,D), (B,C), and (B, D) in Eq. (4.7). On the other hand, two pairwise comparisons (E, E) and (F, F) in Eq. (4.8) are always lower ranked than corresponding two pairwise comparisons, (A, B) and (C, D). There are N_1 features composed of two classes and N_2 features composed of four classes. Thus mean rank of pairs (A, B) and (C, D) are $\frac{N_2}{2}$ because N_2 features composed of four classes are ranked higher

Table 4.3 Mean (expected) rank, mean lowest rank, and mean top ranks of pairwise comparisons

							Integrated rank
Pairs in Eq. (4.7)	(A, B)	(A, C)	(A, D)	(B, C)	(B, D)	(C, D)	
Mean rank	$\frac{N_2}{2}$	$N_1 + \frac{N_2}{2}$	$N_1 + \frac{N_2}{2}$	$N_1 + \frac{N_2}{2}$	$N_1 + \frac{N_2}{2}$	$\frac{N_2}{2}$	$3N_2 + 4N_1$
Pairs in Eq. (4.8)	(E, E)	(E, F)	(E, F)	(E, F)	(E, F)	(F, F)	
Mean rank	$\frac{N+N_2}{2}$	$\frac{N_1}{2}$	$\frac{N_1}{2}$	$\frac{N_1}{2}$	$\frac{N_1}{2}$	$\frac{N+N_2}{2}$	$N + 2N_1 + N_2$
Pairs in Eq. (4.7)	(A, B)	(A, C)	(A, D)	(B, C)	(B, D)	(C, D)	
Mean lowest rank	N_2	$N_1 + N_2$	$N_1 + N_2$	$N_1 + N_2$	$N_1 + N_2$	N_2	$4N_1 + 6N_2$
Pairs in Eq. (4.8)	(E, E)	(E, F)	(E, F)	(E, F)	(E, F)	(F, F)	
Mean top rank	$\frac{N_2+N-N_1}{2}$	1	1	1	1	$\frac{N_2+N-N_1}{2}$	$N - N_1 + N_2 + 4$

Integrated rank is summation of ranks of six pairwise comparisons

than other features. Mean rank of (A, C), (A, D), (B, C), and (B, D) are $\frac{N_2}{2} + N_1$ because N_1 features composed of two classes are always ranked higher than N_2 features composed of four classes. Mean rank of four pairs (E, F) in Eq. (4.8) is $\frac{N_1}{2}$ because N_1 features composed of two classes are higher ranked than other features. Mean rank of two pairs (E, E) and (F, F) are $\frac{N+N_2}{2}$ because N_2 features composed of four classes are higher ranked than others. Next, integrated rank is computed as the summation over six pairwise comparisons. Then, integrated rank of features composed of four classes is

$$2 \times \frac{N_2}{2} + 4 \times \left(N_1 + \frac{N_2}{2} \right) = 4N_1 + 3N_2 \tag{4.9}$$

and integrated rank of features composed of two classes is

$$2 \times \frac{N + N_2}{2} + 4 \times \frac{N_1}{2} = N + 2N_1 + N_2 \tag{4.10}$$

In order that N_2 features composed of four classes are higher ranked than N_1 features composed of two classes based upon integrated rank in average, Eq. (4.9) $<$ Eq. (4.10). Thus

$$\text{Eq. (4.10)} - \text{Eq.(4.9)} > 0 \tag{4.11}$$

$$N + 2N_1 + N_2 - (4N_1 + 3N_2) > 0 \tag{4.12}$$

$$N - 2N_1 - 2N_2 > 0 \tag{4.13}$$

$$N > 2(N_1 + N_2) \tag{4.14}$$

is required. Otherwise, integrated rank based upon six pairwise comparisons, Eq. (4.7), cannot select N_2 features composed of four classes more likely than N_1

features composed of two classes. This means that total number of features distinct between any pairs of classes must not exceed the half of total number of features. This requirement is unlikely fulfilled always.

Equation (4.14) that cannot always be expected to be satisfied is only for average. Even if Eq. (4.14) stands, at most only half of selected features is correctly composed of four classes. If we require that there should not be any false positives, requirement can become more strict (Table 4.3). In order that, we have to require that top ranked features among those composed of two classes must be always ranked lower than the lowest ranked features among those composed of four classes. The rank of bottom ranked feature among those composed of four classes by the two pairwise comparison (A, B) and (C, D) in Eq. (4.7) is N_2 because there are N_2 features that are composed of four classes and are ranked higher than other features. The rank of feature ranked as bottom by the four pairwise comparisons (A, C), (A, D), (B, C), and (B, D) in Eq. (4.7) among those composed of four classes is $N_1 + N_2$ because N_1 features that are composed of two classes and are ranked higher than N_2 features composed of four classes. On the other hand, features ranked as top by two pairwise comparisons (E, E) and (F, F) in Eq. (4.8) among those composed of two classes are ranked uniformly between N_2 and $N - N_1$. This is because N_2 features composed of four classes are higher ranked than N_1 features composed of two classes and there are N_1 features ranked lower than top ranked features among those composed of two classes. Thus, mean top ranked features among those composed of two classes by two pairwise comparisons (E, E) and (F, F) in Eq. (4.8) is $\frac{N-N_1+N_2}{2}$. The rank of feature ranked as top by four pairwise comparisons (E, F) in Eq. (4.8) among those composed of two classes is 1, because N_2 features composed of two classes are higher ranked than other features. Thus integrated bottom rank among N_2 features composed of four classes is

$$2 \times N_2 + 4 \times (N_1 + N_2) = 4N_1 + 6N_2 \tag{4.15}$$

while integrated top rank among N_1 features composed of two classes is

$$2 \times \left(\frac{N - N_1 + N_2}{2} \right) + 4 = N - N_1 + N_2 + 4. \tag{4.16}$$

In order that there are no false positives, i.e., N_2 features composed of four classes is always ranked higher than N_1 features composed of two classes, Eq. (4.16) > Eq. (4.15),

$$\text{Eq. (4.16)} - \text{Eq. (4.15)} > 0 \tag{4.17}$$

$$N - N_1 + N_2 + 4 - (4N_1 + 6N_2) > 0 \tag{4.18}$$

$$N - 5N_1 - 5N_2 + 4 > 0 \tag{4.19}$$

$$N + 4 > 5(N_1 + N_2). \tag{4.20}$$

This means that the number of features composed of two classes and that of four classes must be less than 10% of N if $N_1 = N_2$. This is a less likely fulfilled requirement than Eq. (4.14). Thus integration of six pairwise comparisons unlikely correctly identifies N_2 features composed of four classes when features composed of two classes coexist with them.

Because pairwise comparisons are not expected to work well to identify features composed of multiple classes when more than two kinds of multiple classes coexist, e.g. Eq. (4.4), usually any other alternative strategies are recommended to employ; ones of such alternative strategies are categorical regressions. In categorical regression, class labels are converted to dummy variables, δ_{kj} that takes 1 when jth sample belongs to kth class otherwise 0. Then, categorical regression analysis of x_{ij} is

$$x_{ij} = a_i + \sum_k b_{ik}\delta_{kj} \tag{4.21}$$

where a_i and b_{ik} are the regression coefficients specific to ith feature. Pairwise comparisons that assume four classes could not distinguish features composed of four classes from those composed of two classes well. This problem does not exist in categorical regression analysis anymore. Suppose the simplest cases correspond to two classes, Eq. (4.6), and four classes, Eq. (4.5), as

$$(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2) \tag{4.22}$$

and

$$(1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4) \tag{4.23}$$

respectively. It is obvious that there are no residual errors when Eq. (4.21) assuming four classes (Table 4.4) is applied to Eq. (4.22) if $a_i = \frac{3}{2}$, $b_{i1} = b_{i2} = -\frac{1}{2}$, $b_{i3} = b_{i4} = \frac{1}{2}$. Because there are no residual errors when Eq. (4.21) assuming four classes (Table 4.4) is applied to Eq. (4.23) as well if $a_i = \frac{5}{2}$, $b_{i1} = -\frac{3}{2}$, $b_{i2} = -\frac{1}{2}$, $b_{i3} = \frac{1}{2}$, and $b_{i4} = \frac{3}{2}$, this cannot discriminate four classes from two classes. Nevertheless, Eq. (4.21) assuming two classes (Table 4.4) can discriminate two

Table 4.4 δ_{kj} in categorical regression, Eq. (4.21), assuming either four classes, Eq. (4.5), and two classes, Eq. (4.6), respectively

	Four classes				Two classes	
k	$1 \le j \le 5$,	$6 \le j \le 10$,	$11 \le j \le 15$,	$16 \le j \le 20$,	$1 \le j \le 10$,	$11 \le j \le 20$
1	1	0	0	0	1	0
2	0	1	0	0	0	1
3	0	0	1	0		
4	0	0	0	1		

classes from four classes. If $a_1 = \frac{3}{2}$, $b_{i1} = -\frac{1}{2}$ and $b_{i2} = \frac{1}{2}$, there are no residual errors for Eq. (4.22). On the other hand, there are no solutions with no residual errors when Eq. (4.21) assuming two classes (Table 4.4) is applied to Eq. (4.23). Thus, integration of categorical regression analyses assuming four classes and two classes can identify features composed of two classes and those composed of four classes successfully.

In order to see if categorical regression analysis, Eq. (4.21), can identify features composed of two classes and those composed of four classes simultaneously, we apply categorical regression, Eq. (4.21), to data set 3, Eq. (4.4), as follows. First we apply categorical regression, Eq. (4.21), assuming four classes to data set 3. Because categorical regression assuming four classes are simultaneously coincident with features composed of four classes and those composed of two classes, we select top ranked $N_1 + N_2$ features, which is the total number of features that are composed of either two or four classes, i.e. $i \leq N_1 + N_2$. Then, we apply categorical regression assuming two classes to data set 3. Because categorical regression assuming two classes are coincident with only features composed of two classes, we select top ranked N_1 features, which is the total number of features that are composed of two classes, i.e. $i \leq N_1$. Features selected by categorical regression assuming two classes are considered as features composed of two classes. On the other hand, features selected by categorical regression assuming four classes but not selected by categorical regression assuming two classes are considered as features composed of four classes. Table 4.5 shows the performance of this integrated categorical regression assuming two classes and four classes when $N = 100, M = 20, \mu_0 = 8, \mu_1 = \mu_2 = \frac{\mu_0}{2} = 2, N_1 = 10, N_2 = 10$ and $\sigma = 1$ in data set 3, Eq. (4.4). Performance is averaged over one hundred independent trials. Categorical regression can identify features composed of two classes and four classes completely.

In order to see if PCA based unsupervised FE is applicable, it is applied to the same data set, too. In this case, we selected top 10 features and the second top 10 features (i.e., ranked between 11th and 20th) associated with absolutely larger u_{1i}. Since we do not know which one corresponds to two classes or four classes, after investigating coincidence, we assign top 10 to four classes and the second top 10 to two classes. PCA based unsupervised FE is also successful (Table 4.5). The only disadvantage of PCA based unsupervised FE is that it cannot find the correspondence between selected sets of features and the number of classes in advance.

Table 4.5 Performance of statistical tests applied to synthetic data sets 3 defined by Eq. (4.4)

Categorical regression		PCA based unsupervised FE	
Two classes	Four classes	Two classes	Four classes
10.00	10.00	9.97	9.97

Numbers represent mean number of features distinct between two classes, $i < N_1 (= 10)$, and four classes, $N_1 < i \leq N_1 + N_2$, among N_1 features selected by each method, respectively

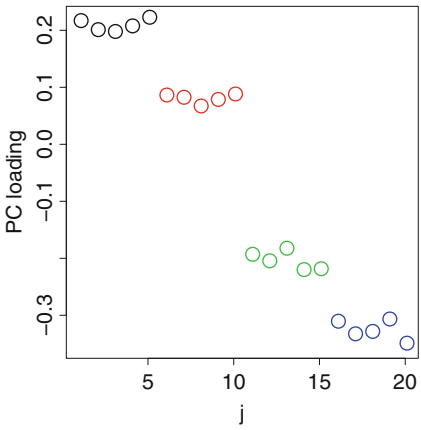

Fig. 4.2 The first PC loading vectors, $\boldsymbol{v}_1 \in \mathbb{R}^M$, for data set 3

In order to see this, we can observe the first PC loading vector, $\boldsymbol{v}_1$ (Fig. 4.2). It is obvious the first PC loading vector is coincident with four classes. This is the reason why the top ranked 10 features are coincident with, not two classes, but four classes. Although we do not repeat the application to shuffled data, it is obvious that categorical regression does not work toward shuffled data because feature selection is performed with class labeling. PCA based unsupervised FE is not affected by shuffling, because PC score vectors, $\boldsymbol{u}_k$s, which is used for feature selection, are not affected by the order of samples, thus are not affected by the class labeling as well. Thus, in this complicated situation, i.e., coexistence of features composed of two classes and four classes, PCA based unsupervised FE is the most favorable method.

4.4 Identification of Non-sinusoidal Periodicity by PCA Based Unsupervised FE

Identification of periodicity, no matter whether it is spatial or temporal, has ever been central issue of data science. In order to identify periodicity, sinusoidal regression is often used. Sinusoidal regression is defined as

$$x_{ij} = a_i + b_i \sin\left(\frac{2\pi}{T} j\right) + c_i \cos\left(\frac{2\pi}{T} j\right) \tag{4.24}$$

where a_i, b_i, c_i are regression coefficients specific to ith feature and T is period. In the following, for the simplicity, $T \in \mathbb{N}$. There are multiple practical problems on regression analysis. At first, we need to know period T in advance in order to apply regression analysis to data set. Of course, it is possible to estimate T from the data set with considering T to be a fitting parameter as well. Nevertheless, there is no known algorithm to find best T values, because any minimization algorithm applied

to residues might fall in local minimum that differs from true T. Second, and more critical problem is that not all periodicity is sinusoidal. Only requirement of x_{ij} to be periodic with the period T is

$$x_{ij} = x_{ij+T} \tag{4.25}$$

which does not restrict functional forms to be sinusoidal at all.

In order to see how well sinusoidal regression, Eq. (4.24), can work, we apply it to the data set 4 with period of T

Data set 4

$$x_{ij} = \begin{cases} f_{(i+j) \bmod T} + a\varepsilon_{ij} & i \leq N_1 \\ a\varepsilon_{ij} & i > N_1 \end{cases} \tag{4.26}$$

where $f_j \in \mathbb{R}^T$ and $\varepsilon_{ij} \in \mathbb{R}^{N \times M}$ are drawn from normal distribution $\mathcal{N}(0, \sigma)$, mod is modulo operation, and $0 < a < 1$ is the coefficient that represents signal noise ratio. Because of the term $(i + j) \bmod T$, $\left\{ x_{ij} \middle| 1 \leq j \leq M \right\}$s have distinct phases from one another. Performance is averaged over 100 independent trials. Table 4.6 shows the performance when $N = 100, M = 50, T = 10, a = 0.1, \sigma = 1, N_1 = 10$. It is as small as 5.72 which is hardly said to be a good performance. This low performance is because of f_j's non-sinusoidal functional form (Fig. 4.3).

Table 4.6 Performance of statistical tests applied to synthetic data sets 4 defined by Eq. (4.26)

Sinusoidal regression	PCA based unsupervised FE
5.72	10

Numbers represent mean number of features with period T, $i \leq N_1 (= 10)$ among N_1 features selected by each method, respectively

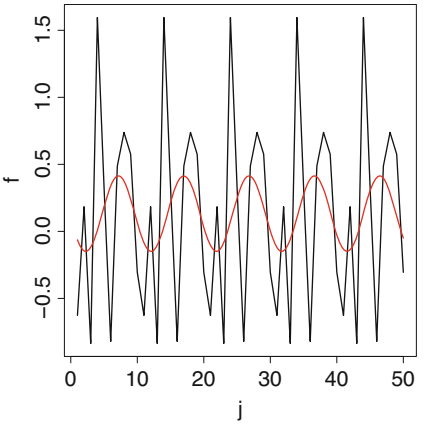

Fig. 4.3 Typical $f_{j \bmod T} \in \mathbb{R}^M (M = 50, T = 10)$ in Eq. (4.26) (black) and its sinusoidal regression, Eq. (4.24) (red)

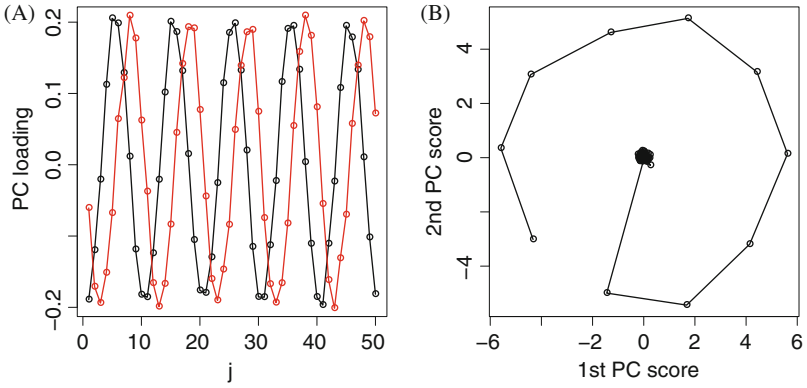

Fig. 4.4 (**a**) A typical first PC loading (black), v_{1j}, and second PC loading (red), v_{2j}. (**b**) Scatterplot of typical first PC score, u_{1i}, and the second PC score, u_{2i}, that correspond to PC loading shown in (**a**)

Next, we apply PCA based unsupervised FE to data set 4, Eq. (4.26), as in Sect. 4.2 excluding one point; instead of ranking features based on the absolute value of the first PC score, $|u_{1i}|$, features are ranked based upon squared sum of the first and second PC scores $u_{1i}^2 + u_{2i}^2$. Table 4.6 shows the performance which is as large as 10, i.e., no errors.

The reason why we need to employ, not only the first PC score, u_{1i}, but also the second PC score, u_{2i}, can be seen in Fig. 4.4. As can be seen in Fig. 4.4a, the first and second PC loading represent periodic function of period $T(= 10)$. And the first 10 pairs of the first and the second PC scores, $u_{ki}, i \leq N_1(= 10), k \leq 2$, form circular trajectory in the plain spanned by the first and the second PC (Fig. 4.4b). This is because of the term $(i + j)$ mod T in Eq. (4.26) that generates phase shift between features $x_{ij}, i \leq N_1(= 10)$. In some cases, the corresponding PC loading, v_{1j} and v_{2j}, represent not the period T, but the period $\frac{T}{2}$ or $\frac{T}{3}$. Nevertheless, in data set 4, Eq. (4.26), only features $i \leq N_1(= 10)$ can be coincident with higher modes, $\frac{T}{2}$ or $\frac{T}{3}$. Thus, these cases also can identify periodic features $i \leq N_1(= 10)$ correctly.

In the above explanation, we use circular trajectory shown in Fig. 4.4b to reasons why we need to employ the first two PC scores for feature selection. Nevertheless, in the practical application, the order of analysis can be reversed. First, we might observe the pairwise scatterplots of PC scores to identify which pairs of features have periodicity because periodic features should draw circular trajectory. Next, we can see individual PC loading as in Fig. 4.4a in order to see period T. This is possible because it is unsupervised method that assumes no specific periodic functional forms in advance. In this sense, PCA based unsupervised FE is superior to the sinusoidal regression to select periodic features.

In order to see if PCA based unsupervised FE can recognize periodicity under the more complicated situation, I modified data set 4, Eq. (4.26), such that cycles with two period, T and T', coexist, i.e.

Data set 5

$$x_{ij} = \begin{cases} f_{(i+j) \bmod T} + a\varepsilon_{ij} & i \leq N_1 \\ g_{(i+j) \bmod T'} + a\varepsilon_{ij} & N_1 < i \leq N_2 \\ a\varepsilon_{ij} & i > N_2 \end{cases} \tag{4.27}$$

where $g_j \in \mathbb{R}^{T'}$ is drawn from normal distribution $\mathcal{N}(0, \sigma)$. Figure 4.5 shows the typical g that is far from sinusoidal profile ($T' = 5, N_2 = 20$, other parameters are the same as those in Eq. (4.26)). Figure 4.6 shows the typical first to fourth PC scores, $\boldsymbol{u}_k, 1 \leq k \leq 4$, and PC loading, $\boldsymbol{v}_k, 1 \leq k \leq 4$. It is obvious that Fig. 4.6a, c corresponds to period $T' = 5$ and Fig. 4.6b, d corresponds to period $T = 10$, respectively. Thus, PCA based unsupervised FE basically has the ability to identify features with two distinct periods even when they coexist. The problem is that the first four PCs do not always correspond to two periods, $T' = 5$ and $T = 10$, but other four PCs, e.g., the second, third, seventh, and eighth PCs, correspond to these two periods, in contrast to data set 4, Eq. (4.26), where the first two PC loading always correspond to period $T = 10$. Thus, in order to make use of PCA to identify features with two distinct periods, we need to identify which PC loading corresponds to two periods, $T = 10$ and $T' = 5$, respectively, by applying sinusoidal regression, Eq. (4.24) with $T = 10$ and $T = T' = 5$. Thus, detailed procedure is as follows:

1. Apply PCA to data set 5, x_{ij} (Eq. (4.27)).
2. Apply sinusoidal regression, Eq. (4.24), with $T = T' = 5$ to PC loading, $\boldsymbol{v}_k$ and select top two, k_1 and k_2.
3. Apply sinusoidal regression, Eq. (4.24), with $T = 10$ to PC loading, $\boldsymbol{v}_k$ and select top two, k'_1 and k'_2.

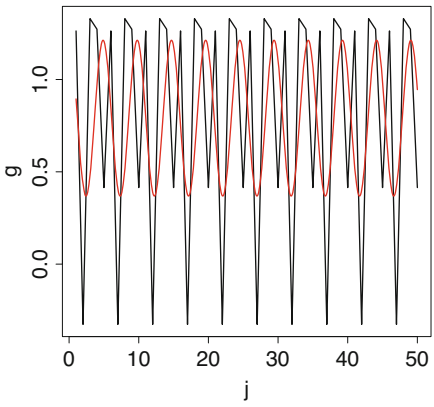

Fig. 4.5 Typical $g_{j \bmod T'} \in \mathbb{R}^M (T' = 5)$ in Eq. (4.27) (black) and its sinusoidal regression, Eq. (4.24) with $T = T' = 5$ (red)

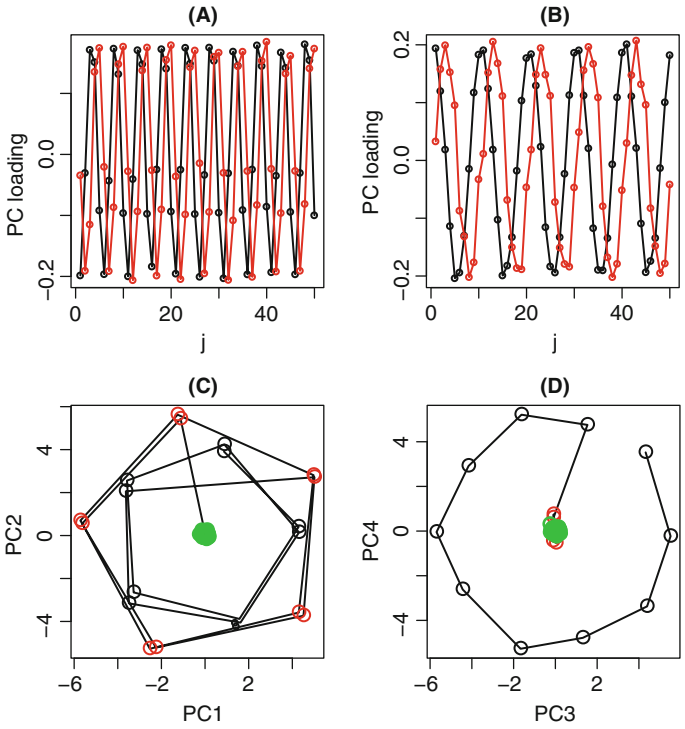

Fig. 4.6 (**a**) Typical first PC loading (black), v_{1j}, and the second PC loading (red), v_{2j}. (**b**) Typical third PC loading (black), v_{3j}, and the fourth PC loading (red), v_{4j}. (**c**) Scatterplot of typical first PC score, u_{1i}, and the second PC score, u_{2i}, that correspond to PC loading shown in (**a**). (**d**) Scatterplot of typical third PC score, u_{3i}, and the fourth PC score, u_{4i}, that correspond to PC loading shown in (**b**). Black open circles: $j \leq N_1 (= 10)$, red open circles: $N_1 < j \leq N_2 (= 20)$, green open circles: $N_2 < j$

4. Select top ranked $N_2 (= 20)$ features using squared sum of two v_{ki}s, $v_{k'_1 i}^2 + v_{k'_2 i}^2$, selected in step 3 (this is because PC score, $\boldsymbol{u}_k$ with period $T' = 5$, identifies features with periods $T' = 5$ and $T = 10$ as can be seen in Fig. 4.6c).
5. Select top ranked $N_1 (= 10)$ features using squared sum of two v_{ki}s, $v_{k_1 i}^2 + v_{k_2 i}^2$, selected in step 2 (this is because PC score, $\boldsymbol{u}_k$ with period $T = 10$, identifies only features with periods $T = 10$ as can be seen in Fig. 4.6d).
6. Identify features selected in step 5 as those with period $T = 10$.
7. Identify features selected in step 4 but not in step 5 as those with period $T = 5$.

Performance is averaged over 100 independent trials (Table 4.7). PCA based unsupervised FE obviously can identify features with two distinct periods almost completely.

In order to see if sinusoidal regressions, Eq. (4.24) with $T = 10$ and $T = T' = 5$, can perform as well as PCA based unsupervised FE, we applied sinusoidal

Table 4.7 Performance of statistical tests applied to synthetic data sets 5 defined by Eq. (4.27)

Sinusoidal regression		PCA based unsupervised FE	
$T = 10$	$T = T' = 5$	$T = 10$	$T = T' = 5$
6.32	6.75	9.73	9.99

Numbers represent mean number of features with period $T = 10$, $i \leq N_1 (= 10)$ among N_1 features selected by each method and that of features with period $T = T' = 5$, $N_1 < i \leq N_2 (= 20)$ among $N_2 - N_1 (= 10)$ features selected by each method, respectively

regression to data set 5, Eq. (4.27), too. Top $10 (= N_1 = N_2 - N_1)$ features were selected with $T = 10$ and $T = T' = 5$, respectively (Table 4.7). Sinusoidal regression is clearly inferior to PCA based unsupervised FE, possibly because of non-sinusoidal nature of f_j (Fig. 4.3) and g_j (Fig. 4.5) in Eq. (4.27).

4.5 Null Hypothesis

In the above examples, the number of features considered, e.g., those composed of multiple classes or those with specific period, is known in advance. Nevertheless, in the real application, it is unrealistic to assume that the number of features that should be selected is known in advance. In this case, usually P-values are attributed to individual features. These P-values represent the possibility that observation can happen accidentally under the null hypothesis that represents something opposite to the nature that selected features should obey.

For example, when we search features composed of two classes, the P-values represent the possibility that absolute difference of means between two classes can become accidentally larger than observed values when all observations are drawn from the same distribution (e.g., normal distribution with the same mean and standard deviation). If P-values are small enough, we can consider these features to be those composed of two classes, because the observed difference can unlikely appear if there are no classes.

There are some issues in this strategy. The first one is how we can select the null hypothesis. P-values are obviously dependent upon the selection of null hypothesis. Thus, it is important to select "correct" null hypothesis to address proper P-values to features. Unfortunately, there is no known established strategy to select the correct null hypothesis. Null hypothesis, which should be rejected, cannot be observable. Even if majority of features do not always follow null hypothesis, it might simply mean that most of the features are associated with properties searched. Therefore, only requirement is to present clearly null hypothesis together with the P-values attributed to features.

Another issue is how small P-values should be. Generally, P-values are considered to be false ratio. In other words, if we select n features associated with P-values smaller than p, there can be at most np features selected wrongly in spite of that they

obey the null hypothesis. Thus, ideally, p should be as small as $\frac{1}{n}$ such that there are no false positives. Nonetheless, it is often unrealistic to require $p < \frac{1}{n}$ especially when n is large and data is noisy. Therefore, practically, p is set to be 0.01 or 0.05, because it is enough if the 99% or 95% of selected feature are correct, for the usual purpose.

The third and the most critical issue is the problem of multiple comparisons. When there are N features to which P-values are attributed, P-values can be accidentally as small as $\frac{1}{N}$. When N is large, e.g., $N \sim 10^4$, it causes a problem. Even if some features have P-values as small as 10^{-4}, we cannot reject null hypothesis. Thus, we cannot select these features as those associated with properties searched, e.g., composed of two classes. In spite of that, it is often unrealistic to require that P-values should be as small as 10^{-4}. Although there are many ways to address this difficulty, we employ Benjamini Hochberg (BH) criterion, because it is known to work practically well, in the applications described in the following chapters.

The basic idea of BH criterion is very simple. If the features obey null hypothesis completely, e.g., apparently two classes features are drawn from the same distribution, e.g., normal distribution, the distribution of P-values should be uniform distribution $\in [0, 1]$, because this is the definition of probability. Thus, if we order P-values in ascending order, the ith largest P-value should be as large as $\frac{i}{N}$. In other words, if the ith largest P-value is smaller than $\frac{i}{N}$, it unlikely occurs under the null hypothesis.

Considering these discussions, BH criterion is as follows:

1. Order P-values attributed to ith feature, P_i, in ascending order.
2. Find the smallest i_0 such that $P_{i_0} > \frac{i_0}{N} p$ where p is threshold P-values.
3. Select features, $i \leq i_0$, such that their attributed P-values are practically supposed to be less than p.

Throughout the remaining part of this book, we employ this criterion to adjust P-values with considering multiple comparisons as many as the number of features, N.

4.6 Feature Selection with Considering P-Values

In order to perform feature selection with considering P-values, we select null hypothesis for the distribution of PC score, u_{ki}, as normal distribution. In order to assign P-values to features, we employ χ^2 distribution as

$$P_i = P_{\chi^2}\left[> \sum_k \left(\frac{u_{ki}}{\sigma_k} \right)^2 \right] \tag{4.28}$$

where $P_{\chi^2}[> x]$ is the cumulative probability that the argument is larger than x. The summation is taken over PCs selected for identification of ith feature that fulfills desired condition. The degrees of freedom of χ^2 distribution is equal to the number of PCs included in the summation. σ_k is the standard deviation of u_{ki}. Then features associated with adjusted P-values less than 0.01 are selected.

Other methods compared with PCA based unsupervised FE in the previous section can also attribute P-values to individual features. Using these P-values, features associated with adjusted P-values less than 0.01 can be selected. This enables us to compare performance between the various methods.

At first, we perform analysis shown in Table 4.2 with replacing identification of features based upon top ranked $N_1 (= 10)$ features with that based upon features associated with adjusted P-values less than 0.01. Unfortunately, not all tests shown in Table 4.2 can derive P-values. Evaluation based upon variance has no ways to attribute to P-values, because no null hypothesis can exist. Regression analysis cannot either, because complete fitting is always possible because the number of features, N, is larger than the number of samples, M. Thus, only remaining three, t test, unimodal test, and PCA based unsupervised FE can be employed. We do not employ shuffling in this case, because the effect of shuffling was presented in Table 4.2.

Evaluations based upon adjusted P-values do not always give us N_1 features selected. Thus, instead of presenting the number of correctly selected features as in Table 4.2, we need to present confusion matrix, which is demonstrated in Table 4.8. Suppose that there are two classes, positive set and negative set (in the case of feature selection, positive corresponds to features with considered properties, e.g., those composed of two classes, and negative corresponds to features without considered properties, e.g., those without any classes). The number of positives predicted as positive is true positive (TP). The number of positives predicted as not positive is false negative (FN). The number of negatives predicted as positive is false positive (FP). The number of negatives predicted as not positive is true negative (TN). If FN = FP = 0, it is complete prediction.

Confusion matrices when three statistical tests are applied to data set 1, Eq. (4.1), and data set 2, Eq. (4.2), are shown in Tables 4.9 and 4.10, respectively. The performance is averaged over 100 independent trials. t test performs almost equally between data sets 1 and 2, although the performance decreases as M decreases or N increases. PCA based unsupervised FE totally fails for data set 1, while it is successful for larger N in data set 2. Unimodal test has never been successful. One

Table 4.8 Confusion matrix

	Real	
Prediction	Positive	Negative
Positive	TP	FP
Negative	FN	TN

TP true positive, *FP* false positive, *FN* false negative, *TN* true negative

Table 4.9 Confusion matrices when statistical tests are applied to synthetic data sets 1 defined by Eq. (4.1) and features associated with adjusted P-values less than 0.01 are selected

	t test		Unimodal test		PCA	
	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$
Data set 1: $N = 100$, $M = 20$						
Selected	10.00	0.10	0.03	0.08	0.00	0.00
Not selected	0.00	89.90	9.97	89.92	10.00	90.00
Data set 1: $N = 100$, $M = 10$						
Selected	5.96	0.18	0.01	0.06	0.00	0.00
Not selected	4.04	89.82	9.99	89.94	10.00	90.00
Data set 1: $N = 1000$, $M = 20$						
Selected	9.98	0.2	0.0	0.2	0.00	0.00
Not selected	0.02	989.8	10	989.8	10.00	990.00
Data set 1: $N = 1000$, $M = 10$						
Selected	1.16	0.2	0.0	0.04	0.00	0.00
Not selected	8.84	989.8	10	989.96	10.00	990.00

$N_1 = 10$

Table 4.10 Confusion matrices when statistical tests are applied to synthetic data sets 2 defined by Eq. (4.2) and features associated with adjusted P-values less than 0.01 are selected

	t test		Unimodal test		PCA	
	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$
Data set 2: $N = 100$, $M = 20$						
Selected	10.00	0.07	0.0	0.07	0.00	0.01
Not selected	0.00	89.93	10.00	89.93	10.00	89.99
Data set 2: $N = 100$, $M = 10$						
Selected	6.08	0.06	0.00	0.00	0.00	0.00
Not selected	3.92	89.94	10.00	90.00	10.00	90.00
Data set 2: $N = 1000$, $M = 20$						
Selected	9.98	0.1	0.00	0.00	9.97	0.07
Not selected	0.02	989.9	10.00	990.0	0.03	989.03
Data set 2: $N = 1000$, $M = 10$						
Selected	1.09	0.01	0.0	0.04	9.4	0.00
Not selected	8.91	989.99	10	989.96	0.6	990.0

$N_1 = 10$

remarkable point is that PCA based unsupervised FE can outperform t test when $N = 1000$ and $M = 10$. This suggests that PCA based unsupervised FE might be the best when $N \gg M$; the situation $N \gg M$ is very usual in the bioinformatics. This is the basic motivation that this textbook is written.

In spite of that PCA based unsupervised FE is an unsupervised method that does not fully make use of available information while t test is a supervised method that fully makes use of available information, the reason why PCA based unsupervised FE can outperform t test when $N \gg M$ is as follows. In t test, P-values increase

as M decreases (i.e., less significant). On the other hand, the correction of P-values considering multiple comparisons is enhanced as N increases. Thus, adjusted P-values become larger (less significant) as N increases. This means, if $N \gg M$, t test hardly computes small enough P-values. On the other hand, in PCA based unsupervised FE where P-values are computed by u_{1i} which is less affected by varying M, P-values are less dependent on M. In Table 4.10, TPs computed by PCA based unsupervised FE do not change much between $M = 10$ and $M = 20$ when $N = 1000$. In addition to this, in this setup, N_1 that represents the number of positives remains unchanged while N increases. This means, the number of negatives increases. Generally, negatives are associated with smaller absolute values of u_{1i} because u_{1i} is associated with v_{1j} that represents distinction between two classes (Fig. 4.1). P-values are computed based upon normalized u_{1i}, Eq. (4.28), thus absolute values u_{1i} attributed to positives become relatively larger as the number of negatives increases. This process has the tendency that increasing the number of negatives reduces P-values attributed to positives (i.e., more significant). Because of that, in Table 4.10, PCA based unsupervised FE is successful only when $N = 1000$.

This is the reason why PCA based unsupervised FE is employed for the feature selection in bioinformatics where $N \gg M$ is quite usual. P-values computed by PCA based unsupervised FE is less affected by M that is typically small in bioinformatics while P-values decrease for larger N that is typically very large in bioinformatics. Thus, PCA based unsupervised FE is very fitted to the problems in bioinformatics.

One might be interested in what will happen if selection based upon adjusted P-values is applied to other examples discussed in the above. The answer is that it is dependent upon various parameters. In the examples analyzed in this section, PCA based unsupervised FE can outperform t test only when $N = 1000$ and $M = 10$. Thus, whether it works well or not when it is applied to real data set is also dependent upon the properties of data sets. The general tendency that PCA based unsupervised FE works well only when $N \gg M$ is universal independent of the data sets considered. Thus, the discussion about in which situation PCA based unsupervised FE that selects features based upon adjusted P-values works well is postponed to the later chapters where PCA based unsupervised FE is applied to real data sets. The readers can see many examples where PCA based unsupervised FE works well or not in these later chapters.

4.7 Stability

Weaker sensitivity of PCA based unsupervised FE on the number of samples, M, naturally results in the stability of feature selection. The stability of feature selection is defined as the robustness of feature selection when samples change. Suppose that samples are drawn from some distributions. If selected features vary every time

samples are drawn from distribution, it is problematic in biology where individual features, e.g., genes, have meanings.

In PCA based unsupervised FE, P-values are less dependent upon the number of samples. In other words, every time we select half of samples among the available samples, P-values attributed to individual features do not change. If P-values attributed to individual features do not change, the selected features do not change, either. This is definitely equivalent to the stability. In the applications of PCA based unsupervised FE to real data sets described in the following chapters, readers will see many examples that PCA based unsupervised FE outperforms other methods from the point of stability. This is yet another reason why PCA based unsupervised FE is a recommended method to be used in bioinformatics.

4.8 Summary

In this chapter, I proposed to make use of PCA as a tool of feature selection. PCA based unsupervised FE can identify features composed of multiple classes better than conventional supervised methods, e.g., t test and categorical regression. When it is applied to identification of non-sinusoidal periodic features, PCA based unsupervised FE can outperform another conventional method, sinusoidal regression. With attributing P-values to features under the null hypothesis that PC scores obey χ^2 distribution, PCA based unsupervised FE correctly identifies features composed of two classes only when $N \gg M$, i.e., the number of features is much larger than the number of samples.

Reference

1. Hartigan, J.A., Hartigan, P.M.: The dip test of unimodality. Ann. Stat. **13**(1), 70–84 (1985). https://doi.org/10.1214/aos/1176346577