Chapter 5
TD Based Unsupervised FE

Although our world might have no reason to exist, it sounds fantastic, because we can make the reason for ourselves.
Filicia Heideman, Sound of the Sky, Season 1, Spisode 7

5.1 TD as a Feature Selection Tool

In this chapter, I would like to make use of TD as a feature selection tool. Suppose that $x_{ijk} \in \mathbb{R}^{N\times M\times K}$ represents the value of the ith feature of the samples having jth and kth properties as

Data set 6:

$$x_{ijk} \sim \begin{cases} \mathcal{N}(\mu,\sigma), & i \le N_1,\ j \le \frac{M}{2},\ k \le \frac{K}{2} \\ \mathcal{N}(0,\sigma), & \text{otherwise} \end{cases} \tag{5.1}$$

In this example, j and k are supposed to be classified into two classes, $j \le \frac{M}{2}$, $K \le \frac{M}{2}$ and $j > \frac{M}{2}$ or $j > \frac{K}{2}$ for $i \le N_1$. Then, x_{ijk} is drawn from normal distribution, $\mathcal{N}(\mu,\sigma)$, with positive mean, $\mu > 0$, only when $j \le \frac{M}{2}, k \le \frac{K}{2}$, otherwise $\mu = 0$. The purpose of feature selection is to find N_1 features associated with two classes shown in Eq. (5.1).

Tucker decomposition, Eq. (3.2), with HOSVD algorithm, Fig. 3.8, is applied to data set 6, Eq. (5.1), with $N = 1000$, $M = K = 6$, $N_1 = 10$, $\mu = 2$, $\sigma = 1$, as

$$x_{ijk} = \sum_{\ell_1=1}^{N}\sum_{\ell_2=1}^{M}\sum_{\ell_3=1}^{K} G(\ell_1,\ell_2,\ell_3)u^{(i)}_{\ell_1 i}u^{(j)}_{\ell_2 j}u^{(k)}_{\ell_3 k} \tag{5.2}$$

where $\boldsymbol{u}^{(i)}_{\ell_1} \in \mathbb{R}^N$, $\boldsymbol{v}^{(i)}_{\ell_2} \in \mathbb{R}^M$, $\boldsymbol{u}^{(k)}_{\ell_3} \in \mathbb{R}^K$, $G(\ell_1,\ell_2,\ell_3) \in \mathbb{R}^{N\times M\times K}$. Figure 5.1a, b shows a typical realization of $\boldsymbol{u}^{(j)}_1$ and $\boldsymbol{u}^{(k)}_1$, respectively. It is obvious that these two correctly reflect the distinction between $j > \frac{M}{2}, k > \frac{K}{2}$ and $j \le \frac{M}{2}, k \le \frac{K}{2}$. Next,

Y-h. Taguchi, *Unsupervised Feature Extraction Applied to Bioinformatics*,
Unsupervised and Semi-Supervised Learning,
https://doi.org/10.1007/978-3-030-22456-1_5

we would like to identify which $\boldsymbol{u}_{\ell_1}^{(i)}$ can be used for feature selection. In contrast to PCA based unsupervised FE, it is not clear which $\boldsymbol{u}_{\ell_1}^{(i)}$ should be used, because there is no one-to-one correspondence among $\boldsymbol{u}_{\ell_1}^{(i)}, \boldsymbol{u}_{\ell_2}^{(j)}, \boldsymbol{u}_{\ell_3}^{(k)}$; instead of that, their relationship is represented through the core tensor, G.

In order to see this relationship, we order $G(\ell_1, 1, 1)$ with descending order of absolute values; Table 5.1 shows the core tensors, $G(\ell_1, 1, 1)$, sorted in this order. Table 5.1 suggests that $\boldsymbol{u}_1^{(i)}$ is most likely associated with $\boldsymbol{u}_1^{(j)}$ and $\boldsymbol{u}_1^{(k)}$, because $G(1, 1, 1)$ has the largest absolute value among $G(\ell_1, 1, 1)$. Actually, $\boldsymbol{u}_1^{(i)}$ shown in Fig. 5.1c obviously has larger absolute values for $i \leq N_1$ than others. Thus, the strategy proposed here, i.e., first find singular value vectors attributed to samples and associated with desired class dependence, then identify singular value vectors, attributed to features, that share G having larger absolute values with them, can identify features with not known in advance j, k dependence in fully unsupervised manner. The reason why it works so well is obvious. If we see $\boldsymbol{u}_{\ell_2}^{(j)} \times^0 \boldsymbol{u}_{\ell_3}^{(k)}$ that is shown in Fig. 5.1d, it is fully associated with the j, k dependence defined in Eq. (5.1) that means only $j, k < \frac{M}{2}$ are drawn from normal distribution with positive mean while others are drawn from those with zero mean.

Next issue might be if TD based unsupervised FE can outperform conventional methods. As a representative of conventional methods, we employ again categorical regression analysis, Eq. (4.21), that is modified to be adapted to co-existence of two kinds of classes,

Table 5.1 $G(\ell_1, 1, 1)$s that correspond to Fig. 5.1

ℓ_1	1	4	2	6
$G(\ell_1, 1, 1)$	−35.484412	2.137686	1.748955	−1.705922

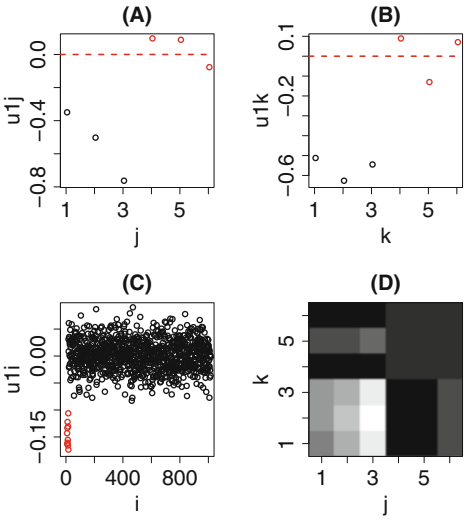


Fig. 5.1 A typical realization of $\boldsymbol{u}_1^{(i)}, \boldsymbol{u}_1^{(j)}, \boldsymbol{u}_1^{(k)}$ when Tucker decomposition, Eq. (3.2), with HOSVD algorithm, Fig. 3.8 is applied to data set 6, Eq. (5.1) with $N = 1000$, $M = K = 6$, $N_1 = 10$, $\mu = 2, \sigma = 1$. (**a**) $\boldsymbol{u}_1^{(j)}$, (**b**) $\boldsymbol{u}_1^{(k)}$, black and red circles correspond to $j \leq \frac{M}{2}, k \leq \frac{K}{2}$ and $j > \frac{M}{2}, k > \frac{K}{2}$, respectively. Red broken lines show baseline. (**c**) $\boldsymbol{u}_1^{(i)}$. Red open circle corresponds to $i \leq N_1$, i.e., features associated with j, k dependence. (**d**) $\boldsymbol{u}_1^{(j)} \times^0 \boldsymbol{u}_1^{(k)}$. Brighter squares indicate larger values

$$x_{ijk} = a_i + \sum_{s=1}^{2} b_{is}\delta_{sj} + \sum_{s=1}^{2} c_{is}\delta_{sk} \tag{5.3}$$

where a_i, b_{is}, c_{is} are the regression coefficients. δ_{sj} and δ_{sk} are the function that takes 1 only when sample j or k belongs to the sth class otherwise 0.

In order to perform feature selection, P-values need to be addressed to features. For categorical regression analysis, P-values computed by categorical regression analysis is used as it is. For TD based unsupervised FE,

$$P_i = P_{\chi^2}\left[> \left(\frac{u_{1i}^{(i)}}{\sigma_1} \right)^2 \right] \tag{5.4}$$

is used to attribute P-values to features where σ_1 is the standard deviation of $u_{1i}^{(i)}$. Both P-values, i.e., computed with TD based unsupervised FE and categorical regression analysis, are corrected by BH criterion and features associated with adjusted P-values less than 0.01 are selected. Table 5.2 shows the performances achieved by TD based unsupervised FE and categorical regression, Eq. (5.3). Performance is averaged over 100 independent examples. In contrast to TD based unsupervised FE that can identify more than 60% of features associated with searched j, k dependence, categorical regression, Eq. (5.3), could identify almost no features. The cause of this drastic low performance is obvious. Equation (5.3) assumes four classes, because j and k are composed of two classes, respectively. Thus, two classes times two classes are equal to four classes. Nevertheless, Eq. (5.1) obviously admits two classes, i.e., $j \leq \frac{M}{2}, k \leq \frac{K}{2}$ versus others. This not proper assumption in the model (categorical regression analysis) results in poor performance. In actuality, if we employ categorical regression as

$$x_{ijk} = a_i + \sum_{s=1}^{2} b_{is}\delta_{sjk} \tag{5.5}$$

Table 5.2 Confusion matrices when statistical tests are applied to synthetic data sets 6 defined by Eq. (5.1) and features associated with adjusted P-values less than 0.01 are selected

	TD based unsupervised FE		Categorical test(four classes)		Categorical test(two classes)	
Data set 6	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$
Selected	6.34	0.00	0.63	0.00	7.35	0.00
Not selected	3.66	990	9.37	990	2.65	990

	TD based unsupervised FE		Categorical test(nine classes)		Categorical test(two classes)	
Data set 7	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$	$i \leq N_1$	$N_1 < i$
Selected	8.73	0.00	4.58	0.00	10.0	0.00
Not selected	1.27	990	5.42	990	0.00	990

$N_1 = 10$. "categorical test(two classes)" corresponds to Eq. (5.3), "categorical test(four classes)" corresponds to Eq. (5.5), and "categorical test(nine classes)" corresponds to Eq. (5.7)

where δ_{sjk} is a function that takes 1 only when

$s = 1$: $j \leq \frac{M}{2}$ and $k \leq \frac{K}{2}$
$s = 2$: $j > \frac{M}{2}$ or $k > \frac{K}{2}$

otherwise 0 and a_i, b_{sjk} are the regression coefficients, categorical regression can outperform TD based unsupervised FE as expected (Table 5.2). The only problem is that it is usually impossible to assume two classes in spite of that there are four classes based upon the apparent category. In this case, unsupervised method can outperform supervised method.

In order to confirm these tendencies, we prepare additional synthetic data.

Data set 7:

$$x_{ijk} \sim \begin{cases} \mathcal{N}(\mu, \sigma), \ i \leq N_1, \ \frac{M}{3} < j \leq \frac{2M}{3}, \ \frac{K}{3} < k \leq \frac{2K}{3} \\ \mathcal{N}(0, \sigma), \qquad \text{otherwise} \end{cases}. \tag{5.6}$$

Equation (5.3) is modified as

$$x_{ijk} = a_i + \sum_{s=1}^{3} b_{is}\delta_{sj} + \sum_{s=1}^{3} c_{is}\delta_{sk} \tag{5.7}$$

with three classes, $1 \leq j \leq \frac{M}{3}$ or $1 \leq k \leq \frac{K}{3}$ for $s = 1$, $\frac{M}{3} < j \leq \frac{2M}{3}$ or $\frac{K}{3} < k \leq \frac{2K}{3}$ for $s = 2$, and $\frac{2M}{3} < j \leq M$ or $\frac{2K}{3} < k \leq K$ for $s = 3$. On the other hand, Eq. (5.5) remains unchanged although δ_{sjk} takes 1 only when

$s = 1$: $\frac{M}{3} < j \leq \frac{2M}{3}$ and $\frac{K}{3} < k \leq \frac{2K}{3}$
$s = 2$: $j \leq \frac{M}{3}$ or $j > \frac{2M}{3}$ or $k \leq \frac{K}{3}$ or $k > \frac{2K}{3}$

otherwise 0. $M = K = 12$ and other parameters remain unchanged. As expected (Table 5.2), the performances of categorical regressions applied to set 7 are improved from those applied to data set 6, because the number of samples, MK, increases while the number of features, N, remains unchanged. In spite of these improved performances of categorical regression analyses, TD based unsupervised FE still outperforms three classes × three classes = nine classes categorical regression analysis, Eq. (5.7) (see Table 5.2). Thus, as far as apparent categories that do not correctly reflect true category are considered, TD based unsupervised FE can outperform supervised method. It is very usual in genomic data analysis that it is unclear if apparent categories are coincident with true, but unknown, classes. This is possibly the reason why TD based unsupervised FE often outperforms supervised methods in the applications to bioinformatics that will be introduced in the later part of this book.

It should be also emphasized that TD based unsupervised FE can outperform supervised methods only when $N \gg MK$, i.e., the number of features is much larger than the number of samples. Although we do not demonstrate this using more synthetic data sets, one should remember this point when one would like to employ TD based unsupervised FE.

5.2 Comparisons with Other TDs

Here I employed only Tucker decomposition, Eq. (3.2), with HOSVD algorithm, Fig. 3.8, for feature selection. Since I have already argued the superiority of Tucker decomposition toward other two TDs, CP decomposition and tensor train decomposition, it might not be necessary to demonstrate superiority of Tucker decomposition to other two TDs. Nevertheless, it is not meaningless to see what we can get when the other two TDs are applied to data set 6.

First, tensor train decomposition, Eq. (3.3), with $R_1 = R_2 = M = K = 6$ is applied to data set 6, whose results obtained by Tucker decomposition are shown in Fig. 5.1 (Fig. 5.2). Figure 5.2 looks very similar to Fig. 5.1. In spite of that, tensor train decomposition is still inferior to Tucker decomposition. First of all, we have no idea how we should choose R_is that decide the rank of tensor train decomposition. In the present case, we can try to find R_is that result in the same result as that in Fig. 5.1. If not, we can have no ways to decide R_is. Second, we do not know how to relate $G^{(j)}(j, 1, 1)$, $G^{(k)}(k, 1)$, and $G^{(i)}(i, 1)$ with one another, because there is no core tensor that plays the role to connect singular vectors in Tucker decomposition (Table 5.1) where we know what I should search. If not as in the present case, i.e., tensor train decomposition, we have no idea which core tensors given by tensor train decomposition are selected for the feature selection.

Next, we apply CP decomposition, Eq. (3.1), with $L = 1$ to data set 6, whose results obtained by Tucker decomposition are shown in Fig. 5.1. Figure 5.3

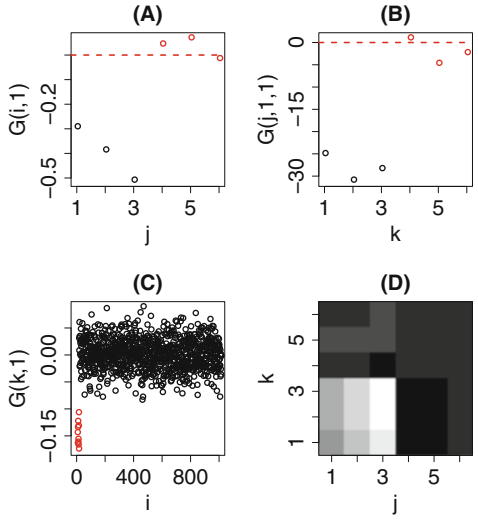


Fig. 5.2 $G^{(j)}(j, 1, 1)$, $G^{(k)}(k, 1)$, $G^{(i)}(i, 1)$ when tensor train decomposition, Eq. (3.3), with $R_1 = R_2 = M = K = 6$ is applied to data set 6, Eq. (5.1) whose results obtained by Tucker decomposition are shown in Fig. 5.1. (**a**) $G^{(j)}(j, 1, 1)$, (**b**) $G^{(k)}(k, 1)$, black and red circles correspond to $j \le \frac{M}{2}, k \le \frac{K}{2}$ and $j > \frac{M}{2}, k > \frac{K}{2}$, respectively. Red broken lines show baseline. (**c**) $G^{(i)}(i, 1)$. Red open circle corresponds to $i \le N_1$, i.e., features associated with j, k dependence. (**d**) $G^{(j)}(j, 1, 1) \cdot G^{(k)}(k, 1)$. Brighter squares indicate larger values

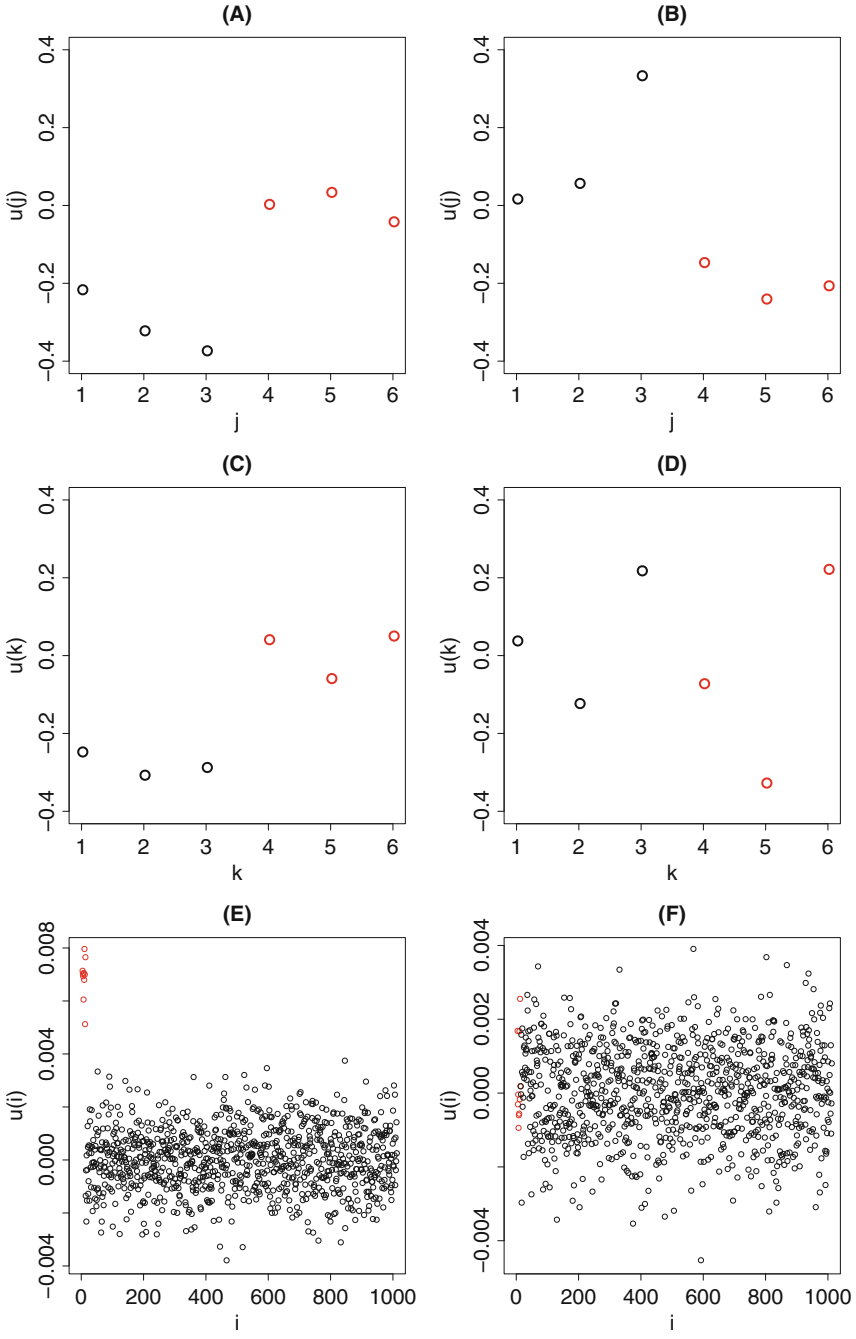


Fig. 5.3 Two typical convergent realizations starting from different initial values of CP decomposition, Eq. (3.1), with $L = 1$ applied to data set 6, Eq. (5.1), whose results obtained by Tucker decomposition is shown in Fig. 5.1. (**a**) and (**b**) $\boldsymbol{u}_1^{(j)}$, black and red circles correspond to $j \leq \frac{M}{2}$ and $j > \frac{M}{2}$, respectively. (**c**) and (**d**) $\boldsymbol{u}_1^{(k)}$, black and red circles correspond to $k \leq \frac{K}{2}$ and $k > \frac{K}{2}$, respectively. (**e**) and (**f**) $\boldsymbol{u}_1^{(i)}$. Red open circle corresponds to $i \leq N_1$, i.e., features associated with j, k dependence

represents the two independent results starting from different initial values (one should remember that CP decomposition need to be given by initial values from where computation starts). At first, they clearly differ from each other. Second, the second realizations, (b), (d), and (f), do not correspond to the distinction between two classes and fail to identify features with not known in advance j, k dependence, $i \leq N_1$. Thus, CP decomposition is inferior to Tucker decomposition because of initial condition dependence as discussed earlier.

These comparisons suggest that Tucker decomposition is superior to tensor train decomposition and CP decomposition as a tool of feature selection.

5.3 Generation of a Tensor From Matrices

In the previous section, we showed that TD based unsupervised FE can outperform conventional supervised feature selection, categorical regression analysis, when the number of features is much larger than the number of samples and true classification is a complex function of apparent labeling. Although TD based unsupervised FE is shown to be effective, it is unfortunately not so frequent that there are data sets formatted as tensor, because getting tensor requires more observation than matrices. In order to get $N \times M$ matrix that represents M samples with N features, required number of observations is as many as the number of samples, i.e., M. On the other hand, in order to get $N \times M \times K$ tensors that correspond to N features observed under the combination of M times and K times measurements, the required number of observation is as many as $K \times M$. If we need to have tensors with more modes, the number of observation will increase, too. Thus, even if TD based unsupervised FE is an effective method, we usually cannot have data set formatted as tensors, to which TD based unsupervised FE is applicable.

In order to have more opportunities to which we can apply TD based unsupervised FE, we can propose to generate tensors from matrices [1], which are obtained more easily than tensors. Suppose that we have two matrices, $x_{ij} \in \mathbb{R}^{N \times M}$ and $x_{ik} \in \mathbb{R}^{N \times K}$, which represent i features under the jth experimental conditions and the kth experimental conditions, respectively. A typical observation is that N health conditions, blood pressure, body mass, body temperature, height, weight, etc. are observed M individuals in Japan and K individuals in the USA. Then we can get tensor $x_{ijk} \in \mathbb{R}^{N \times M \times K}$ by simply multiplying x_{ij} and x_{ik},

$$x_{ijk} = x_{ij} x_{ik} \tag{5.8}$$

TD can be applied to x_{ijk} as usual. It does not have to be restricted to the product of two matrices. We can generate $m + 1$ mode tensor by multiplying m matrices, $x_{ij_1}, x_{ij_2}, \ldots, x_{ij_m}$ as

$$x_{ij_1 j_2 \cdots j_m} = \prod_{s=1}^{m} x_{ij_s} \tag{5.9}$$

On the other hand, we can consider the alternative cases where not features but samples are common between two matrices. Suppose that for K individuals two distinct N and M observations are performed and are recorded as matrices form, $x_{ik} \in \mathbb{R}^{N\times K}$ and $x_{jk} \in \mathbb{R}^{M\times K}$. A typical example is that there are N goods in kth shop and x_{ik} represents a price of ith good in kth shop. On the other hand, x_{jk} represents the number of customers at jth time point at kth shop. We can generate tensor $x_{ijk} \in \mathbb{R}^{N\times M\times K}$ as

$$x_{ijk} = x_{ik}x_{jk} \tag{5.10}$$

Again we can employ more matrices as

$$x_{i_1 i_2 \cdots i_m j} = \prod_{s=1}^{m} x_{i_s j} \tag{5.11}$$

From the mathematical point of view, although there are no needs to distinguish between equations Eqs. (5.11) and (5.9), they should be considered separately from the data science point of view. Then hereafter we denote Eq. (5.11), i.e., the cases sharing samples, as case I while Eq. (5.9), i.e., the cases sharing features, as case II, respectively.

5.4 Reduction of Number of Dimensions of Tensors

It is possible to produce tensors from matrices. However, it increases the number of features. When two matrices, $x_{ij} \in \mathbb{R}^{N\times M}$ and $x_{ik} \in \mathbb{R}^{N\times K}$ are multiplied in order to generate a tensor $x_{ijk} \in \mathbb{R}^{N\times M\times K}$ (case II), the number of features increases from $N \times (M + K)$ to $N \times M \times K$. Thus, we need some way to reduce the number of dimensions of generated tensors. Here we propose taking summation of shared features, i.e.,

$$\tilde{x}_{i_1 i_2 \cdots i_m} = \sum_j x_{i_1 i_2 \cdots i_m j} \tag{5.12}$$

$$\tilde{x}_{j_1 j_2 \cdots j_m} = \sum_i x_{i j_1 j_2 \cdots j_m} \tag{5.13}$$

Then the number of dimensions increases from $N \times (M + K)$ not to $N \times M \times K$ but to $M \times K$ for case II while from $(N + M) \times K$ not to $N \times M \times K$ but to $N \times M$ for case I.

One might wonder how we can compute singular value matrices that correspond to indices of which are taken summation when TD is applied to $\tilde{x}_{i_1 i_2 \cdots i_m}$ or $\tilde{x}_{j_1 j_2 \cdots j_m}$. These missing singular value matrices are recovered by the following computations,

Table 5.3 Distinction between cases and types

	Type I		Type II	
Case I	$x_{i_1 i_2 \cdots i_m j} = \prod_{s=1}^{m} x_{i_s j}$	Eq. (5.11)	$\tilde{x}_{i_1 i_2 \cdots i_m} = \sum_j x_{i_1 i_2 \cdots i_m j}$	Eq. (5.12)
Case II	$x_{i j_1 j_2 \cdots j_m} = \prod_{s=1}^{m} x_{i j_s}$	Eq. (5.9)	$\tilde{x}_{j_1 j_2 \cdots j_m} = \sum_i x_{i j_1 j_2 \cdots j_m}$	Eq. (5.13)

$$\boldsymbol{u}_\ell^{(i;j_s)} = X^{(ij_s)} \times_{j_s} \boldsymbol{u}_\ell^{(j_s)} \tag{5.14}$$

$$\boldsymbol{u}_\ell^{(j;i_s)} = X^{(ji_s)} \times_{i_s} \boldsymbol{u}_\ell^{(i_s)} \tag{5.15}$$

where $X^{(ij_s)} \in \mathbb{R}^{N \times M_s}$ and $X^{(ji_s)} \in \mathbb{R}^{M \times N_s}$, respectively. Thus, we have m singular value matrices that correspond to i_s or j_s, instead of one singular value matrix. This might look problematic. Nevertheless, practically, if m singular value matrices obtained are mutually highly correlated, it is not practically problematic. Thus, case to case, we might employ this approximate strategy. In order to distinguish these tensors from the previous one, we call those generated after the partial summation of index, Eqs. (5.12) and (5.13) as type II while those without partial summation, Eqs. (5.9) and (5.11), as type I. Table 5.3 summarizes the distinction between cases and types.

5.5 Identification of Correlated Features Using Type I Tensor

The purpose of introduction of tensors summarized in Table 5.3 is simply because we would like to make use of TD based unsupervised FE when no tensors are available. Nevertheless, we can make use of tensors listed in Table 5.3 for the additional alternative purpose as bi-product: identification of mutually correlated features. Suppose we have two sets of observations to K samples formatted as matrices, $x_{ik} \in \mathbb{R}^{N \times K}$ and $x_{jk} \in \mathbb{R}^{M \times K}$. The question is to search pairs of features between two sets.

The standard strategy is to compute pairwise correlation between x_{ik} and x_{jk},

$$r_{ij} = \frac{\frac{1}{K}\sum_k \left(x_{ik} - \frac{1}{K}\sum_{k'} x_{ik'}\right)\left(x_{jk} - \frac{1}{K}\sum_{k'} x_{jk'}\right)}{\sqrt{\frac{1}{K}\sum_k \left(x_{ik} - \frac{1}{K}\sum_{k'} x_{ik'}\right)^2 \frac{1}{K}\sum_k \left(x_{jk} - \frac{1}{K}\sum_{k'} x_{jk'}\right)^2}} \tag{5.16}$$

and to identify pairs of i and j associated with significant correlation. In the following, we will show some synthetic data set where pairwise computation of correlation does not work well while TD applied to a tensor generated from the product of two matrices, $x_{ijk} = x_{ik} x_{jk}$, can identify correlated pairs successfully.

In order for this purpose, we prepare data set 8 as follows.
Data set 8:

$$x_{ik} \sim \begin{cases} k + \mathcal{N}(\mu, \sigma) & i \leq N_1 \\ \mathcal{N}(\mu, \sigma) & \text{otherwise} \end{cases} \tag{5.17}$$

$$x_{jk} \sim \begin{cases} k + \mathcal{N}(\mu, \sigma) & j \leq M_1 \\ \mathcal{N}(\mu, \sigma) & \text{otherwise} \end{cases} \tag{5.18}$$

This means, only features $i \leq N_1$ and $j \leq M_1$ share the k dependence while no other pairs are correlated. In this setup, the number of positive (correlated) pairs is $N_1 \times M_1$ among total number of pairs, $N \times M$.

In order to see if pairwise correlation analysis can identify correlated pairs, we compute Pearson's correlation coefficients between all $N \times M$ pairs, x_{ik} and x_{jk}. Then computed correlation coefficient, r_{ij}, is converted to t_{ij} as

$$t_{ij} = \frac{r_{ij}(K-2)}{\sqrt{1-r^2}} \tag{5.19}$$

that is known to obey t distribution with the degrees of freedom of $K-2$. Then P-values are computed using t distribution and are attributed to all of $N \times M$ pairs. These P-values are corrected by BH criterion and pairs associated with adjusted P-values less than 0.05 are considered to be correlated. Table 5.4 shows the confusion matrix averaged over 100 independent trials when $N = M = 100$, $N_1 = M_1 = 10$, $K = 6$, $\mu = \sigma = 1$. In this setup, the number of positive pairs is $N_1 \times M_1 = 100$. It is obvious that there are more false positives (38.49) than true positives (15.47). Thus, it unlikely works well. Next, we apply TD based unsupervised FE to data set 8 with generating case I type I tensor (Table 5.4) as Eq. (5.10). We apply HOSVD algorithm, Fig. 3.8, to data set 8. Figure 5.4a and b shows typical $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ obtained when HOSVD is applied to data set 8, respectively. These two have obviously larger absolute values for $i \leq N_1$ and $j \leq M_1$ than $i > N_1$ and $j > M_1$, respectively. This suggests that $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ can successfully identify features with correlations ($i \leq N_1$ or $j \leq M_1$) from those without correlations ($i > N_1$ or $j > M_1$). How it comes to be possible can be understood by observing $\boldsymbol{u}_1^{(k)}$ (Fig. 5.5). $\boldsymbol{u}_1^{(k)}$ clearly reflects the dependence upon k shown in Eqs. (5.17)

Table 5.4 Confusion matrices when statistical tests are applied to synthetic data sets 8 defined by Eqs. (5.17) and (5.18) and features associated with adjusted P-values less than 0.05 are selected for pairwise correlation and 0.1 for TD based unsupervised FE

	Pairwise correlation		TD based unsupervised FE			
Data set 8	$i \leq N_1$ and $j \leq M_1$	Otherwise	$i \leq N_1$	$N_1 < i$	$j \leq M_1$	$M_1 < j$
Selected	15.47	38.49	6.20	0.00	6.14	0.00
Not selected	84.53	9861.51	3.80	90.00	3.86	90.00

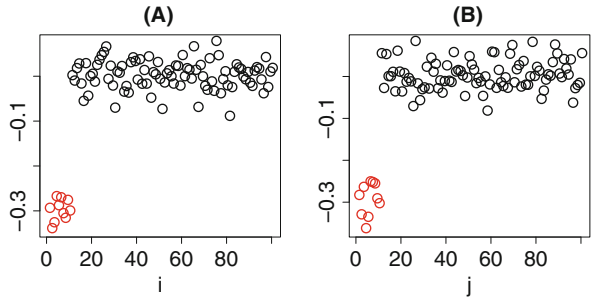
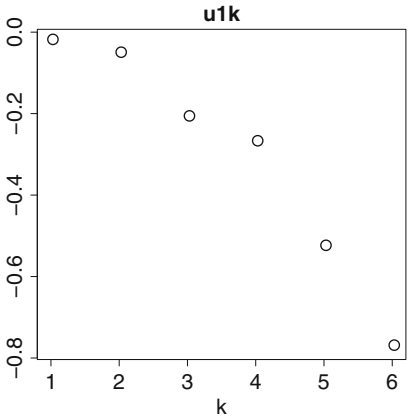


Fig. 5.4 A typical realization of $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ when Tucker decomposition, Eq. (3.2), with HOSVD algorithm, Fig. 3.8 is applied to data set 8, Eqs. (5.17) and (5.18) with $N = M = 100, N_1 = M_1 = 10, K = 6, \mu = \sigma = 1$. (**a**) $\boldsymbol{u}_1^{(i)}$, red and black open circles correspond to $i \leq N_1$ and $i > N_1$, respectively. (**b**) $\boldsymbol{u}_1^{(j)}$, red and black open circles correspond to $j \leq M_1$ and $j > M_1$, respectively

Fig. 5.5 $\boldsymbol{u}_1^{(k)}$ that corresponds to $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ shown in Fig. 5.4



and (5.18). Since $G(1, 1, 1)$ is the largest among $G(\ell_1, \ell_2, 1)$, $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ naturally assign larger absolute values to $u_{1i}^{(i)}$ and $u_{1j}^{(j)}$ that shares embedded k dependence, i.e., $i \leq N_1$ or $j \leq M_1$.

In order to see if $u_{1i}^{(i)}$ and $u_{1j}^{(j)}$ are useful for the feature selection, P-values are attributed to i as Eq. (5.4) and j as

$$P_j = P_{\chi^2}\left[> \left(\frac{u_{1j}^{(j)}}{\sigma_1'} \right)^2 \right] \tag{5.20}$$

where σ_1' is the standard deviation of $u_{1j}^{(j)}$. Then is and js associated with adjusted P-value less than 0.1 are selected (performances are averaged over 100 independent trials). Table 5.4 shows the corresponding confusion matrices. Although the perfor-

mance cannot be said very good, it is remarkable that there are no FP which are as many as 38.49 in pairwise correlation analysis (Table 5.4). TD based unsupervised FE also has more TPs than correlation analysis; 6.20 or 6.14 TPs among 10 positives versus 15.47 TP among 100 positives.

Only from this specific example, we cannot conclude that TD based unsupervised FE can always outperform the conventional methods. Nevertheless, in the application to the real data set that will be shown later, we will see that TD based unsupervised FE can achieve better performances than conventional supervised methods.

5.6 Identification of Correlated Features Using Type II Tensor

In the previous section, we can see that TD based unsupervised FE can correctly recognize the features with mutual correlation that cannot be recognized by conventional pairwise correlation analysis. In this section, we would like to see if type II tensor, Eq. (5.12), can samely identify features with mutual correlations using the same data set 8, Eqs. (5.17) and (5.18). In the present specific case, type II tensor can be defined as

$$\tilde{x}_{ij} = \sum_{k=1}^{K} x_{ijk} = \sum_{k=1}^{K} x_{ik} x_{jk}. \tag{5.21}$$

TD, or essentially it is SVD because HOSVD is equivalent to SVD when it is applied to matrix, is applied to $\tilde{x}_{ij}$. Figure 5.6 shows the comparison of $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ between type I and type II tensors. Although slight deviation can be observed, they

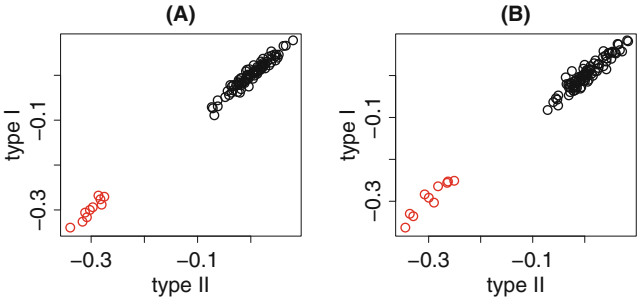


Fig. 5.6 Comparison between $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ in Fig. 5.4 and those when SVD is applied to type II tensor (matrix), $\tilde{x}_{ij}$, defined in Eq. (5.21). (**a**) $\boldsymbol{u}_1^{(i)}$, red and black open circles correspond to $i \leq N_1$ and $i > N_1$, respectively. (**b**) $\boldsymbol{u}_1^{(j)}$, red and black open circles correspond to $j \leq M_1$ and $j > M_1$, respectively

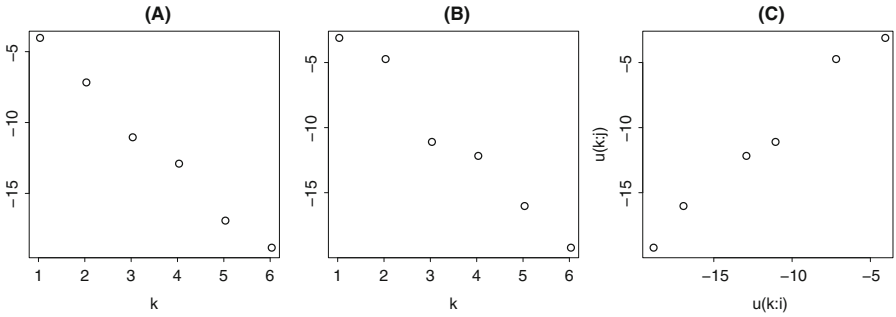


Fig. 5.7 Comparison between $\boldsymbol{u}_1^{(k:i)}$ and $\boldsymbol{u}_1^{(k:j)}$ computed by Eqs. (5.22) and (5.23), respectively. (**a**) $\boldsymbol{u}_1^{(k:i)}$ (**b**) $\boldsymbol{u}_1^{(k:j)}$, (**c**) scatterplot of (**a**) and (**b**)

are coincident enough to recognize features with mutual correlations, i.e., $i \leq N_1$ and $j \leq M_1$, respectively. Thus as long as considering feature selection, replacing type I tensor with type II tensor does not cause any problems.

Then we need to see if two vectors,

$$\boldsymbol{u}_1^{(k;i)} = X^{(ik)} \times_i \boldsymbol{u}_1^{(i)} \tag{5.22}$$

$$\boldsymbol{u}_1^{(k;j)} = X^{(jk)} \times_j \boldsymbol{u}_1^{(j)} \tag{5.23}$$

are coincident with each other and reflect k dependence when $\boldsymbol{u}_1^{(i)}$ and $\boldsymbol{u}_1^{(j)}$ are computed from type II tensor (matrix), Eq. (5.21). Figure 5.7 shows $\boldsymbol{u}_1^{(k:i)}$ and $\boldsymbol{u}_1^{(k:j)}$. They are not only coincident with each other, but also reflecting k dependence in Eqs. (5.17) and (5.18), respectively. Thus, replacing type I tensor with type II, at least in the present case, does not likely cause any problems.

5.7 Summary

In this chapter, we proposed feature section using TD, named TD based unsupervised FE. TD based unsupervised FE can outperform conventional supervised method when the number of samples is much less than the number of features and true classification is a complex function of apparent labeling. We also further extended the concept of tensor such that we can make use of TD based unsupervised FE even when only matrices are given. As a bi-product, we come to be able to select features with mutual correlations even when conventional pairwise correlation analysis fails. Nothing shown in this chapter are proven, but are only demonstrated by synthetic data set. Nonetheless, we will see that TD based unsupervised FE can work very well when it is applied to real examples, i.e., the applications toward bioinformatics in the later part of this book.

Reference

1. Taguchi, Y.H.: Tensor decomposition-based unsupervised feature extraction applied to matrix products for multi-view data processing. PLoS One **12**(8), e0183933 (2017). https://doi.org/10.1371/journal.pone.0183933