Biostatistics (2012), **0**, 0, *pp.* 1–11
doi:10.1093/biostatistics/classificationsupplementary

Supplementary for "Classification of patients from time-course gene expression"

YUPING ZHANG*

Stanford Genome Technology Center, Palo Alto, CA 94306

yupingz@stanford.edu

ROBERT TIBSHIRANI

Departments of Health, Research and Policy, and Statistics, Stanford University, CA 94305

RONALD DAVIS

Stanford Genome Technology Center, Palo Alto, CA 94306

1. An underlying model

We now consider a model to support our method – TPAM (Time-course Prediction Analysis using Microarray). Suppose we have a $p \times T \times N$ dimensioned variable X with N observations x_{gti}, where $g \in \{1, \ldots, p\}$, $t \in \{1, \ldots, T\}$, $i \in \{1, \ldots, N\}$. Let X_{**i} be the $p \times T$ matrix indicating time course gene expression measurements of patient i. For simplification, we write X_{**i} as X_i. Assume $X_i \sim MVN(\mu_k, \Sigma_k)$ from class k, where μ_k is a $p \times T$ matrix and estimated by $\hat{\mu}_k(g,t) = \frac{1}{N_k} \sum_{i \in C_k} x_{gti}$. Σ_k is a $T \times T \times p$ array. For a given g, $\Sigma_k(|g)$ is a $T \times T$ matrix. We assume $\Sigma_k(|g)$, $k \in \{1, \ldots, K\}$, have common covariance matrix $\Sigma(|g)$, where K is the number of classes. Let $\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$ be the sample mean matrix ($p \times T$). $\bar{X}_g$ and X_{gi} are the column

*To whom correspondence should be addressed.

vectors with length of T for gene g. The total sample variance matrix S_T for gene g is defined as

$$S_{Tg} = \frac{1}{N}\sum_{i=1}^{N}(X_{gi} - \bar{X}_g)(X_{gi} - \bar{X}_g)^T.$$

The within-class k covariance matrix S_{Wkg} is given by

$$\hat{\Sigma}_k(|g) = S_{W_kg} = \frac{1}{N_k}\sum_{i\in C_k}(X_{gi} - \hat{\mu}_{gk})(X_{gi} - \hat{\mu}_{gk})^T.$$

The overall pooled within-class covariance matrix S_{wg} is given by

$$S_{Wg} = \frac{1}{N}\sum_{k=1}^{K}\sum_{i\in C_k}(X_{gi} - \hat{\mu}_{gk})(X_{gi} - \hat{\mu_{gk}})^T.$$

The between-class covariance matrix is given by

$$S_{Bg} = S_{Tg} - S_{Wg}$$

For every gene g, X_{gi} is a T dimensional observation vector. Let θ_g be a non-singular linear transformation that transforms the data variables X_g into new variables χ_g. We assume that all the class discrimination information resides in a 1 dimensional sub-space of the T dimensional observation space. This is equivalent to assuming that only the first component of χ_g carries any class discrimination information. Thus, based on this assumption, the class means lie in a one-dimensional subspace, and the remaining $T-1$ dimensional subspace is homogenous with respect to class means and variances. We partition the parameter space of θ_g as $\theta_g = [a_g, \theta_{/ag}]$. Let Z_g denote the first row of χ_g and $\chi_{/Zg}$ denote the remaining $T-1$ rows of χ_g. The Z_g, $g \in \{1,\ldots,p\}$ constitutes the $p \times N$ transformed matrix Z with $z_{gi} = a_g^T X_{gi}$, $i \in \{1,\ldots,N\}$. For sample i from class k, the transformed vector $Z_i = (z_{1i},\ldots,z_{pi})$ follows a multivariate normal distribution $MVN(\nu_k, D_k)$, where $\nu_k = (a_1^T\mu_k(1),\ldots,a_p^T\mu_k(p))$ is the mean vector for class k, and D_k is the covariant matrix for class k. The covariance $Cov(a_{g_1}^T X_{g_1,C_k}, a_{g_2}^T X_{g_2,C_k})$ is equal to

$a_{g_1}^T \Sigma(|g_1, g_2, C_k) a_{g_2}$. We further assume that the covariance matrices are the same across different classes and are diagonal, that is,

$$Cov(a_{g_1}^T X_{g_1,C_k}, a_{g_2}^T X_{g_2,C_k}) = \begin{cases} a_{g_1}^T \Sigma(|g) a_{g_2} = \sigma_{kg} = \sigma_g, & \text{if } g_1 = g_2 = g; \\ 0, & \text{else.} \end{cases}$$

The $p \times T$ mean matrix of $\chi_{/Z}$ is denoted as $\upsilon_{/\nu}$, with estimation $\hat{\nu}_0 = \theta_{/a}^T \bar{X}$, where $\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$. The $(T-1) \times (T-1)$ variance matrix of $\chi_{/Zg}$ for gene g is denoted as $\Omega_{/Dg} = Diag(\sigma_{0g}^2, \ldots, \sigma_{0g}^T)$, with the estimation $\hat{\Omega}_{/Dg} = \theta_{/ag}^T S_{Tg} \theta_{/ag}$.

In summary, the means $\upsilon_k(g)$ and variances $\Omega_k(g)$ of χ_g for each class k and each gene g are as follows

$$\upsilon_k(g) = \begin{pmatrix} \nu_k(g) \\ \upsilon_{/\nu}(g) \end{pmatrix},$$

$$\hat{\upsilon}_k(g) = \begin{pmatrix} a_g^T \hat{\mu}_k(g) \\ \theta_{/ag}^T \bar{X}_g \end{pmatrix},$$

$$\Omega_k(g) = \begin{pmatrix} D_k(g) & 0 \\ 0 & \Omega_{/Dk}(g) \end{pmatrix} = \begin{pmatrix} D(g) & 0 \\ 0 & \Omega_{/D}(g) \end{pmatrix},$$

$$\hat{\Omega}_k(g) = \begin{pmatrix} a_g^T S_{W_k}(g) a_g & 0 \\ 0 & \theta_{/ag}^T S_T(g) \theta_{/ag} \end{pmatrix} = \begin{pmatrix} a_g^T S_W(g) a_g & 0 \\ 0 & \theta_{/ag}^T S_T(g) \theta_{/ag} \end{pmatrix}$$

The probability density function of variable X_{gi}, $i \in C_k$ under the model can be written as

$$p(X_{gi}) = \frac{|\theta|}{(2\pi)^{\frac{T}{2}} |D_k|^{\frac{1}{2}}} exp\{\frac{(\chi_{gi} - \upsilon_k)^T \Omega_k (\chi_{gi} - \upsilon_k)}{2}\}$$

The log-likelihood of the data can be written as

$$\log L(\upsilon,\Omega,\theta|X) = -\frac{pNT}{2}\log 2\pi + pN\log|\theta| - \sum_{g=1}^{p}\{\frac{N}{2}\sum_{j=2}^{T}\log|\sigma_{0g}^{j}|- \sum_{k=1}^{K}\frac{N_k}{2}\log|\sigma_{kg}| - \frac{1}{2}\sum_{k=1}^{K}\sum_{i\in C_k}\frac{(a_g^T X_{gi}-\nu_k(g))^2}{\sigma_{kg}} + \frac{1}{2}\sum_{k=1}^{K}\sum_{i\in C_k}\sum_{j=2}^{T}\frac{(\theta_{/ag}^T X_{gi}-\upsilon_{/v}(g))^2}{\sigma_{0g}^{j}}\}$$

The maximum likelihood estimation of a_g, $g \in \{1,\dots p\}$ can be obtained by maximizing $\log L(\theta|\hat{\upsilon},\hat{\Omega},X)$, that is,

$$\hat{\theta} = argmax_\theta\{N\sum_{g=1}^{p}[-\frac{1}{2}\log|Diag(\theta_{/ag}^T S_{T_g}\theta_{/ag})| - \frac{1}{2}\log|Diag(a_g^T S_{W_g}a_g)| + \log|\theta|]\} \tag{1.1}$$

It can be proved that $\hat{\theta_g}$ corresponds to the right eigenvectors of $S_{W_g}^{-1}S_{T_g}$ and a_g is the first eigenvector of $S_{W_g}^{-1}S_{T_g}$ with the largest eigenvalue. This maximization problem is equivalent to the following maximization problem.

$$J(a_g) = \frac{a_g^T S_{Bg} a_g}{a_g^T S_{Wg} a_g}, \tag{1.2}$$

Let Υ be a $N \times K$ matrix with elements γ_{ik} indicating whether patient i is in class k, where $i \in \{1,\dots,p\}$ and $k \in \{1,\dots,K\}$. We first center and scale each element z_{gi} of Z to be

$$y_{gi} = (z_{gi}-\bar{z}_g)/(c_k s_g + s_0) = (a_g^T X_{gi} - \frac{1}{N}\sum_{i=1}^{N} a_g^T X_{gi})/(c_k s_g + s_0) \tag{1.3}$$

and consider the linear regression:

$$y_{gi} = \sum_{k=1}^{K}\gamma_{ik}\omega_{gk} + \epsilon_{gi}, \tag{1.4}$$

where $\omega_{gk} = (\nu_{gk}-\bar{z}_g)/(c_k s_g + s_0) = (a_g^T\mu_k(g) - \frac{1}{N}\sum_{i=1}^{N} a_g^T X_{gi})/(c_k s_g + s_0)$, $c_k = \sqrt{1/N_k - 1/N}$, $s_g = \sqrt{\frac{1}{N-K}\sum_{k=1}^{K}\sum_{i\in C_k}(z_{gi}-\bar{z}_{gk})^2}$, and ϵ_{gi} are independent of each other and follow $N(0,\sigma_{\epsilon k})$, if sample i belongs to class k. We add an L_1 penalty to the above regression model to select the predictors. The estimator of ω_{gk} can be obtained by minimizing the following objective function.

The score function is

$$\begin{aligned}\hat{\omega}_{gk} &= arg\min_{\omega_{gk}} \frac{1}{2}\sum_{i=1}^{N}\sum_{g=1}^{p}\sum_{k=1}^{K}\frac{\gamma_{ik}}{N_k}(y_{gi}-\omega_{gk})^2 + \triangle\sum_{g=1}^{p}\sum_{k=1}^{K}|\omega_{gk}| \\ &= sign(\sum_i \gamma_{ik}y_{gi})(|\frac{\sum_i \gamma_{ik}y_{gi}}{\sum_i \gamma_{ik}}| - \triangle)_+ \\ &= sign(d_{gk})(|d_{gk}| - \triangle)_+\end{aligned}$$

This is equivalent to shrinking the centroids of each gene towards zero as addressed in Wang and Zhu (2007); Wu (2005).

2. Simulation Studies

2.1 *Simulation study I*

We performed a simulation study to validate the performance of our method. The simulated data set of time point t - X^t consisted of 1000 "genes" (rows) and 100 "patients" (columns). Patients $1 \sim 50$ belong to "type I", while patients $51 \sim 100$ belong to "type II". Here we considered three time points. Let x^t_{gi}, $t \in 1, 2, 3$ denote the "expression level" of the gene g and patient i at time point t. We generated the data as follows:

$$x^0_{gi} = \begin{cases} 6 + \varepsilon_b & g \leqslant 100, i \leqslant 50 \\ 6.1 + \varepsilon_b & g \leqslant 100, i > 50 \\ 3.5 + 0.2I(\mu_i < 0.4) + \varepsilon_b & 200 < g \leqslant 300 \\ 3.5 + \varepsilon_b & else \end{cases}$$

$$x^1_{gi} = \begin{cases} 0.5 + \varepsilon_t + x^0_{gi} & g \leqslant 100, i \leqslant 50 \\ -0.3 + \varepsilon_t + x^0_{gi} & g \leqslant 100, i > 50 \\ \varepsilon_t + x^0_{gi} & else \end{cases}$$

and

$$x^2_{gi} = \begin{cases} -0.8 + \varepsilon_t + x^1_{gi} & g \leqslant 100, i \leqslant 50 \\ -0.7 + \varepsilon_t + x^1_{gi} & g \leqslant 100, i > 50 \\ \varepsilon_t + x^1_{gi} & else \end{cases}$$

Here, $\varepsilon_b \in N(0, 2.5)$ and $\varepsilon_t \in N(0, 2)$, and both are normal distributions. The μ_i is the uniform random variable on $(0, 1)$ and $I(x)$ is an indicator function. We introduced the time

course structure in the $1 \sim 100$ genes.

After generating the training data sets, we constructed the test data sets by the same manner independently. We performed the simulation data sets 10 times independently, and applied the prediction on each simulation. First we performed the class prediction. To compare, for gene expression data from each time point, we applied the PAM to build the predictors using training data and make the prediction on the test data set. The performance of prediction was characterized by the error rate in the test data. We repeated the whole process 10 times and calculated the average performance of the 10 time simulations. Then we generated one big matrix of gene expression for each simulation by simply combining the gene expression from three time points. We performed the regular PAM on the pooled matrix directly. For each tuning parameter, we recorded prediction performance and averaged them across the 10 independent simulations. We also applied the PC-PAM with the first principal component as the unsupervised direction at the first stage and PAM at the second stage. Finally, we applied our prediction methods for longitudinal data (TPAM) with the linear optimal projection at the first stage and PAM at the second stage respectively. We use the number of features as the tuning parameter. As shown in the top-left panel of Figure 1, the TPAM using longitudinal gene expression has the best performance.

2.2 *Simulation study II*

We performed a simulation study to validate the performance of our method. The simulated data set of time point k - X_k consisted of 1000 "genes" (rows) and 100 "patients" (columns). Here we considered three time points. Let x_{ijk}, $k \in \{1, 2, 3\}$ denote the "expression level" of the gene i and patient j at time point k. We generated the data as follows:

$$x_{ij0} = \begin{cases} 6 + \varepsilon_b & i \leqslant 100, j \leqslant 50 \\ 6.1 + \varepsilon_b & i \leqslant 100, j > 50 \\ 3.5 + \varepsilon_b & else \end{cases}$$

$$x_{ij1} = \begin{cases} 0.5 + \varepsilon_t + x^0_{ij} & i \leqslant 100, j \leqslant 50 \\ -0.3 + \varepsilon_t + x^0_{ij} & i \leqslant 100, j > 50 \\ \varepsilon_t + x^0_{ij} & else \end{cases}$$

and

$$x_{ij2} = \begin{cases} -0.8 + \varepsilon_t + x^1_{ij} & i \leqslant 100, j \leqslant 50 \\ -0.7 + \varepsilon_t + x^1_{ij} & i \leqslant 100, j > 50 \\ \varepsilon_t + x^1_{ij} & else \end{cases}$$

Here, $\varepsilon_b \in t(3)$, $\varepsilon_t \in t(3)$, which is Students t-distribution with the degree of freedom 3. We introduced the time course structure in the $1 \sim 100$ genes.

Patients $1 \sim 50$ belong to "type I", while patients $51 \sim 100$ belong to "type II". After generating the training data sets, we generated the test data sets by the same manner independently. We generated the simulation data sets 10 times independently, and applied the prediction on each simulation. First we performed the class prediction. For gene expression data from each time point, we applied the PAM to build the predictors using training data and make the prediction on the test data set. The performance of prediction was characterized by the error rate. We performed the whole process 10 times and calculate the average performance of 10 times simulation. Then we generated one big matrix of gene expression for each simulation by simply combining the gene expression from three time points. We performed the PAM on the pooled matrix directly. For each tuning parameter, we recorded prediction performance and averaged them across 10 times independent simulation. At last, we applied our prediction methods for longitudinal data with linear optimal projection at the first stage and PAM as the second stage respectively. One can notice that in the original PAM method, the tuning parameter has different range across the above five prediction scenario. To make the performance across the five scenario comparable, we use the number of features as the tuning parameter. As shown in the top-right panel of Figure 1, our approach using longitudinal gene expression has the best performance.

2.3 *Simulation study III*

We performed another simulation to further validate the performance with correlated genes. We simulate expression for 1000 genes, 100 patients and three time points. We first generate a 1000×1000 non-diagonal correlation matrix Σ as below:

$$\Sigma_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0.5, & \text{if } i \leqslant 50, j < 50, i \neq j \\ 0 & \text{else} \end{cases}$$

The formula for the simulation is below:

$$x_{gi0} = \begin{cases} 6 + \varepsilon_b & g \leqslant 100, i \leqslant 50 \\ 6.1 + \varepsilon_b & g \leqslant 100, i > 50 \\ 3.5 + \varepsilon_b & else \end{cases}$$

$$x_{gi1} = \begin{cases} 0.5 + \varepsilon_t + x_{gi0} & g \leqslant 100, i \leqslant 50 \\ -0.3 + \varepsilon_t + x_{gi0} & g \leqslant 100, i > 50 \\ \varepsilon_t + x_{gi0} & else \end{cases}$$

and

$$x_{gi2} = \begin{cases} -0.8 + \varepsilon_t + x_{gi1} & g \leqslant 100, i \leqslant 50 \\ -0.7 + \varepsilon_t + x_{gi1} & g \leqslant 100, i > 50 \\ \varepsilon_t + x_{gi1} & else \end{cases}$$

Here, $\varepsilon_b \in MVN(0, \Sigma)$, $\varepsilon_t \in N(0, 2)$ which are multi-variate normal distribution and normal distribution respectively. The μ_j is the uniform random variables on $(0, 1)$ and $I(x)$ is an indicator function. We introduced the time course structure in the $1 \sim 100$ genes.

Patients $1 \sim 50$ belong to "type I", while patients $51 \sim 100$ belong to "type II". After generating the training data sets, we generated the test data sets by the same manner independently. We generated the simulation data sets 10 times independently, and applied the prediction on each simulation. First we performed the class prediction. For gene expression data from each time point, we applied the PAM to build the predictors using training data and make the prediction on the test data set. The performance of prediction was characterized by the error rate. We performed the whole process 10 times and calculate the average performance of 10 times simulation.

Then we generated one big matrix of gene expression for each simulation by simply combining the gene expression from three time points. We performed the PAM on the pooled matrix directly. For each tuning parameter, we recorded prediction performance and averaged them across 10 times independent simulation. At last, we applied our prediction methods for longitudinal data with optimal projection at the first stage and PAM as the second stage respectively. One can notice that in the original PAM method, the tuning parameter has different range across the above five prediction scenario. To make the performance across the five scenario comparable, we use the number of features as the tuning parameter. As shown in the bottom-left panel of Figure 1, our approach using longitudinal gene expression has the best performance.

2.4 *Simulation study IV*

We performed another simulation to further validate the performance using the non-linear projection at the first stage of our method. We simulate expression for 1000 genes, 100 patients and 2 time points. Patients $1 \sim 50$ belong to the first class, while patients $51 \sim 100$ belong to the second class. The formula for the simulation is below:

$$x^0_{gi} = \begin{cases} 6 + \varepsilon_b & g \leqslant 100, i \leqslant 50 \\ 6.1 + \varepsilon_b & g \leqslant 100, i > 50 \\ 3.5 + 0.2I(\mu_j < 0.4) + \varepsilon_b & 200 < g \leqslant 300 \\ 3.5 + \varepsilon_b & else \end{cases}$$

$$x^1_{gi} = \begin{cases} 0.1x^0_{gi}x^0_{gi} - 0.1x^0_{gi} + 4 + \varepsilon_t & g \leqslant 100, i \leqslant 50 \\ 0.05x^0_{gi}x^0_{gi} - 0.1x^0_{gi} + 4 + \varepsilon_t & g \leqslant 100, i > 50 \\ \varepsilon_t + x^0_{gi} & else \end{cases}$$

Here, $\varepsilon_b \in N(0, 2.5)$, $\varepsilon_t \in N(0, 2)$ which are normal distributions. The μ_i is the uniform random variables on $(0, 1)$ and $I(x)$ is an indicator function. We introduced the time course structure in the $1 \sim 100$ genes.

After generating the training data sets, we generated the test data sets by the same manner independently. We generated the simulation data sets 10 times independently, and applied the

prediction on each simulation. First we performed the class prediction. For gene expression data from each time point, we applied the PAM to build the predictors using training data and make the prediction on the test data set. The performance of prediction was characterized by the error rate. We performed the whole process 10 times and calculate the average performance of the 10 simulations. Then we generated one large matrix of gene expression for each simulation by simply combining the gene expression from three time points. We performed the PAM on the pooled matrix directly. For each tuning parameter, we recorded prediction performances and averaged them across the 10 independent simulations. Last, we applied our prediction methods for longitudinal data with nonlinear optimal projection at the first stage and PAM at the second stage respectively. We use the number of features as the tuning parameter. As shown in the bottom-right panel of Figure 1, our approach using longitudinal gene expression has the best performance in both situations.

References

Wang, S. and Zhu, J. (2007). Improved centroids estimation for the nearest shrunken centroid classifier. *Bioinformatics* **23**(8), 972.

Wu, B. (2005). Differential gene expression detection and sample classification using penalized linear regression models. *Bioinformatics* **22**(4), 472.

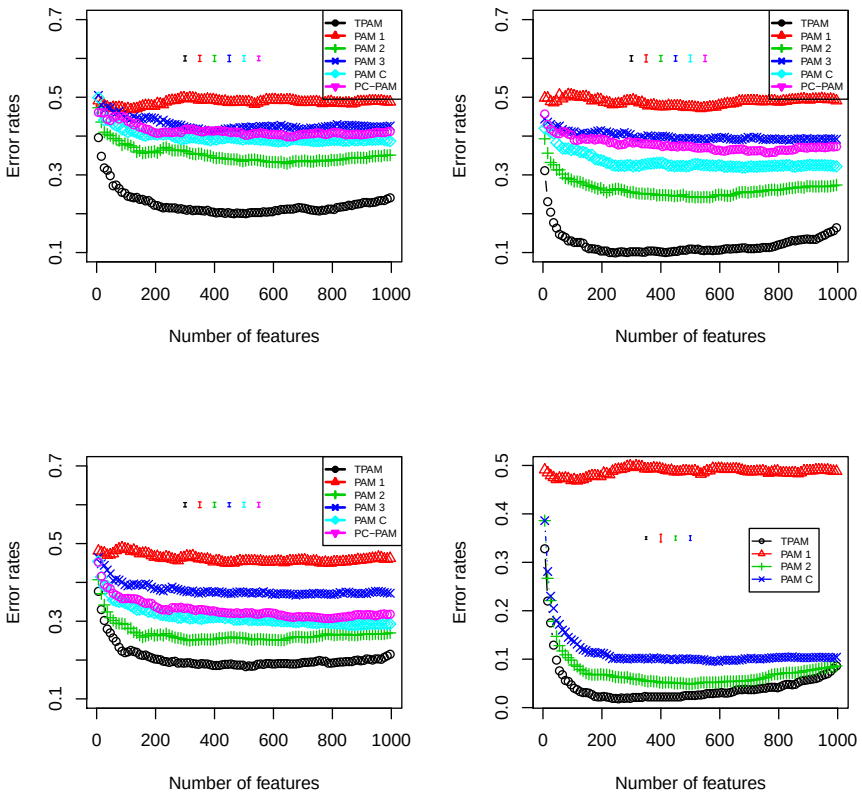

Fig. 1. Classification performances of TPAM and PAM on simulation data with different types of distribution assumptions. Top-left: Simulation study I (independent normal distributions, section 2.1); top-right: Simulation study II (independent t-distribution, section 2.2); bottom-left: Simulation study III (normal distribution with correlated genes, section 2.3); bottom-right: Simulation study IV (optimal direction is non-linear, section 2.4). Black $\circ$: TPAM with linear "optimal direction" projection at the first stage; red $\triangle$: PAM 1, PAM using the first time point; green $+$: PAM 2, PAM using the second time point; blue $\times$: PAM 3, using the third time point; cyan $\diamond$: PAM C, applying PAM to the data set combining the early and middle time points; pink: PC-PAM with the first principal component as the direction of projection at the first stage. Y-axis is the error rate on the test data.