Supplementary Material for "A penalized likelihood method for classification with matrix-valued predictors"

Aaron J. Molstad* and Adam J. Rothman†
Biostatistics Program, Fred Hutchinson Cancer Research Center*
School of Statistics, University of Minnesota†

7 Sensitivity to weight specification

In this section, we repeat the simulations from Section 4 for Model 1 to assess our method's sensitivity to the choice of weight matrices $w_{j,m}$. The weighting choices we compare are the following:

- Adaptive. The weights we suggest in Section 2.1 (Guo, 2010), i.e., $w_{j,m}^{-1} = |\bar{x}_j - \bar{x}_m|$ for $1 \leq j < m \leq J$;
- Constant. The weights are set so that every element of $w_{j,m}$ equals one for all $1 \leq j < m \leq J$;
- Random. The weights are chosen so that $w_{j,m}$ has entries which are a realization of rc independent copies of a Uniform(0,1) random variable;
- Oracle. The weights are chosen with knowledge of the population sparsity pattern: $w_{j,m} = 1_{r\times c}\left(\mu_{*j} = \mu_{*m}\right)$, where $1_{r\times c} : \mathbb{R}^{r\times c} \to \mathbb{R}^{r\times c}$ is the indicator function.

We also include the matrix-normal maximum likelihood estimator (MN) and the Bayes rule (Bayes) described in Section 4.2.

Average misclassification rates are displayed in Figure 1. We found that the Adaptive weights did only slightly worse than PMN with Oracle weights. In terms of misclassification accuracy, the Adaptive weights were similar to the Constant weights, however, when we compare the TNR and TPR of the Adaptive and Constant weights in Table 1, we saw that the Adaptive weights tend to have lower TPR and higher TNR.

*Corresponding author: amolstad@fredhutch.org

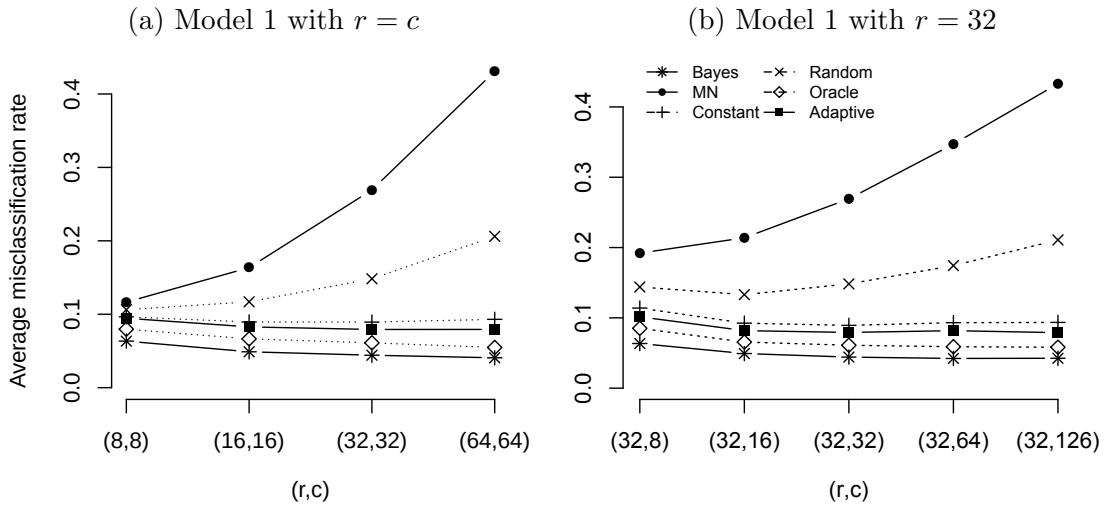


Figure 1: Misclassification rates averaged over 100 replications; (a) and (b) for Model 1 using the weight choices described in Section 7.

Table 1: TNR/TPR percentages averaged over the 100 replications for Model 1 using the weight choices described in Section 7.

	Model 1 (r, c)							
Weights	(8,8)	(16,16)	(32,32)	(64,64)	(32,8)	(32,16)	(32,64)	(32,126)
Adpative	89.4/81.4	96.3/81.0	98.9/75.8	99.5/73.0	98.2/73.4	98.3/76.5	99.4/73.4	99.7/70.9
Constant	79.7/91.9	89.3/87.8	95.8/86.6	98.4/84.8	88.7/87.5	93.0/88.6	97.6/85.2	98.3/85.1
Random	59.9/92.0	75.1/84.4	84.6/82.8	89.6/73.5	72.9/85.8	80.4/82.6	86.8/79.1	89.1/73.0
Oracle	93.9/100	93.7/100	98.4/100	99.4/100	95.2/100	96.3/100	99.3/100	99.4/100

8 Comparison to alternative methods

8.1 Comparison to vector-valued sparse linear discriminant methods

In this section, we compare our proposed method to multiple vector-valued sparse linear discriminant methods applicable for arbitrary number of response categories J. For the methods that are not specifically designed for matrix-valued predictors, we treat the predictor as an rc-variate vector. In addition to Bayes, MN, and PMN defined in Section 4.2, we use the following methods for classification:

- Witten. The L_1-penalized Fisher-linear-discriminant method proposed by Witten and Tibshirani (2011) with tuning parameter and dimension chosen to minimize the misclassification rate on the validation set;
- Clemmensen. The sparse optimal scoring method proposed by Clemmensen et al. (2011) with tuning parameter chosen to minimize the misclassification rate on the validation set;
- Mai. The multiclass sparse linear discriminant analysis method proposed by Mai et al. (2018) with tuning parameter chosen to minimize the misclassification rate on the validation set.

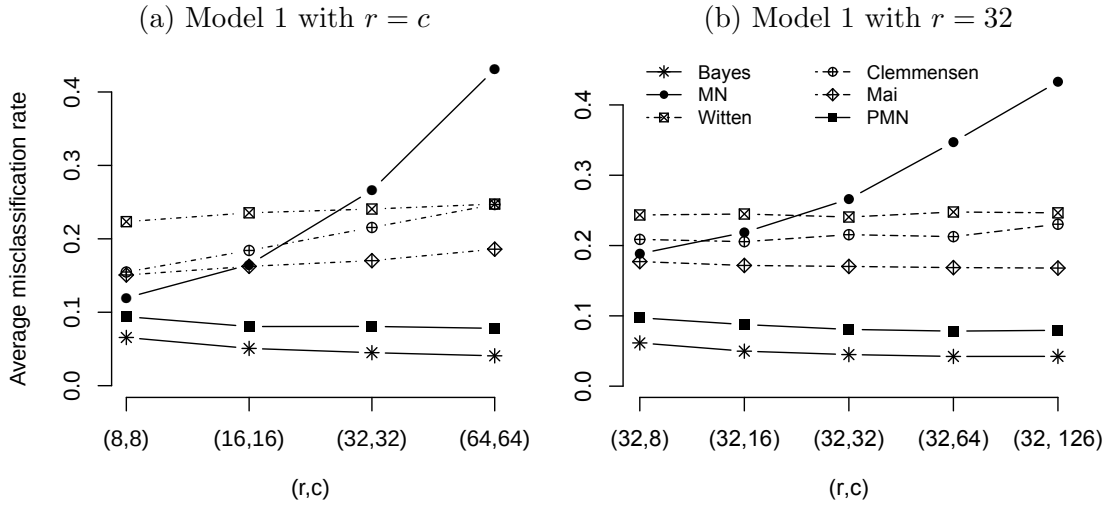


Figure 2: Misclassification rates averaged over 100 replications for Model 1.

Mai et al. (2018) provide a brief review of these three methods and how they differ. We emphasize that these three methods were not designed specifically for classification when the predictor is matrix-valued.

We are aware of only one other method for penalized linear discriminant analysis with matrix-valued predictors: the penalized matrix-variate discriminant analysis (PMDA) method proposed by Zhong and Suslick (2015). However, PMDA treats row and column variables differently by either including or excluding entire rows of the predictor. Since the majority of both row and column variables are irrelevant for Model 1, we would not expect PMDA to perform well and thus, we exclude it from comparison in this section. In the next section, we compare our method to the methods proposed by Zhong and Suslick (2015), including PMDA, under different data generating models.

We compare methods in terms of classification accuracy for Model 1 described in Section 4.1. In Figure 2, we display misclassification rates averaged over 100 replications for each of the competing methods with varying r and c. When $r = c$, the matrix-normal maximum likelihood estimator outperformed some of the vector-valued sparse linear discriminant analysis methods when $r = 8$ and $r = 16$. In general, the method of Mai et al. (2018) tended to perform best amongst the vector-valued sparse linear discriminant methods, all of which were outperformed by our proposed method, PMN. We did not measure TPR or TNR since these methods do not actually fit (1).

8.2 Comparison to Zhong and Suslick (2015)

In this section, we compare our proposed method to both the matrix linear discriminant method (MDA) and penalized matrix linear discriminant method (PMDA) proposed by Zhong and Suslick (2015). For 100 independent replications, we generated a realization of $n = n_{\text{train}} + n_{\text{validate}} + n_{\text{test}}$ independent copies of (X, Y), where we set $n_{\text{train}} = 120$, $n_{\text{validate}} = 120$, and $n_{\text{test}} = 500$. We use the same data generating model as in Section 5.2 of Zhong and Suslick (2015): the categorical response Y had support $\{1, 2\}$ with probabilities $\pi_{*1} = \pi_{*2} = 1/2$, the predictor $X = (U_1, U_2, U_3)^{\mathrm{T}}$, where

$U_1 \mid Y = j \sim \mathrm{N}_{36}\left\{0_{36\times 1}, \Sigma_{*1}\right\}$, where $\Sigma_{*1,s,t} = .5^{|s-t|}$;

$U_2 \mid Y = j \sim \mathrm{N}_{36}\left\{0_{36\times 1}, \Sigma_{*2}\right\}$, where $\Sigma_{*2} = I_{36}$;

and we used two distinct models for U_3:

- Model 5. We set

$$U_3 \mid Y = j \sim \mathrm{N}_{36}\left\{|U_1| - 0.3(j+1), \Sigma_{*3}\right\}, \text{ where } \Sigma_{*3} = .5I_{36},$$

  which is exactly the model from Section 5.2 of Zhong and Suslick (2015).

- Model 6. We set $U_3 = (U_{3,1}, \ldots, U_{3,36})^{\mathrm{T}}$

$$U_{3,k} \mid Y = j \sim \mathrm{N}_1\left\{|U_{3,k}| - 0.5 \cdot 1\left[k \in \{(j-1)18+1, \ldots, j18\}\right], 0.5\right\}, \quad k = 1, \ldots, 36,$$

  where $U_{3,l}$ and $U_{3,m}$ were independent when $l \neq m$.

Model 6 has a stronger signal than Model 5, meaning that more accurate classification is theoretically possible.

We compare our method to the same methods considered in Zhong and Suslick (2015). In addition to MN and PMN defined in Section 4.2; and Clemmensen as defined in Section 8.1, we consider the following methods:

- MDA. The matrix-valued Fisher-linear-discriminant method proposed by Zhong and Suslick (2015);
- PMDA. The matrix-valued penalized Fisher-linear-discriminant method proposed by Zhong and Suslick (2015) with tuning parameter chosen to minimize the misclassification rate on the validation set;
- RDA. Vector-valued regularized linear discriminant analysis with tuning parameters chosen to minimize the misclassification rate on the validation set;
- Lasso. Vector-valued L_1-penalized logistic regression (Friedman et al., 2010) with tuning parameter chosen to minimize the misclassification rate on the validation set.

To implement MDA and PMDA, we used the code provided in the Supplementary Material of Zhong and Suslick (2015). We tried both discrete and continuous tuning parameters for PMDA, and used two different initial values for both MDA and PMDA using both tuning approaches. We used the tuning method and initializer that minimized the misclassification rate on the validation set. We found that for Model 5, PMDA performed slightly better than PMN and RDA. However, for Model 6, PMN slightly outperformed both RDA and PMDA.

9 Efficiency of joint estimation

We also investigated the efficiency of our estimator relative to two alternative estimators of (1) under assumption (2). The first alternative is the matrix-normal maximum likelihood estimator, i.e., MN defined in Section 4.2. The second alternative is a one-step approximation to (3): we initialize Algorithm 1 as described in Section 3.4, we perform Step 1 of Algorithm 1, then we repeat Steps 2–4 of Algorithm 1 until convergence. The algorithm used for this one-step approximation is given in Algorithm 2.

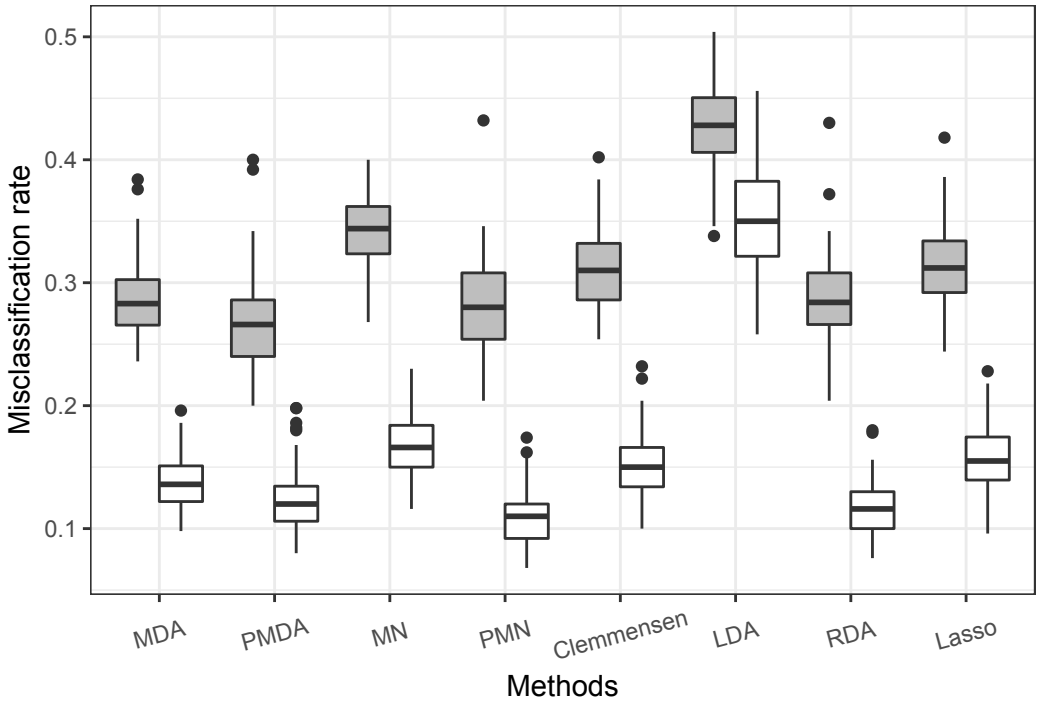


Figure 3: Misclassification rates for 100 independent replications under Model 5 (grey boxplots) and Model 6 (white boxplots).

Algorithm 2. *Given* $\epsilon > 0$*, set* $m = 0$:

Step 1: Iterate the "flip-flop" algorithm (Dutilleul, 1999) until convergence and set $\Phi^{(0)} = \text{diag}(\Phi^{\text{MLE}})$ *and* $\Delta^{(0)} = \text{diag}(\Delta^{\text{MLE}})$ *where* $\left(\Phi^{\text{MLE}}, \Delta^{\text{MLE}}\right)$ *are the final iterates.*

Step 2: Compute $\mu^{(1)} = \underset{\mu \in \mathbb{R}^{(r\times c)J}}{\arg\min}\ g\left(\mu, \Phi^{(0)}, \Delta^{(0)}\right) + \lambda_1 \sum_{j<m} \|w_{j,m} \circ (\mu_j - \mu_m)\|_1$ *using the algorithm from Section 3.3.*

Step 3: Compute $\tilde{\Delta} = \text{GL}\left\{S_\delta\left(\mu^{(1)}, \Phi^{(m)}\right), \lambda_2\right\}$.

Step 4: Compute $\tilde{\Phi} = \text{GL}\left\{S_\phi\left(\mu^{(1)}, \tilde{\Delta}\right), \frac{\lambda_2}{c}\|\tilde{\Delta}\|_1\right\}$.

Step 5: Compute $\Delta^{(m+1)} = \frac{\|\tilde{\Phi}\|_1}{r}\tilde{\Delta}$, $\Phi^{(m+1)} = \frac{r}{\|\tilde{\Phi}\|_1}\tilde{\Phi}$

Step 6: If $f\left(\mu^{(1)}, \Phi^{(m)}, \Delta^{(m)}\right) - f\left(\mu^{(1)}, \Phi^{(m+1)}, \Delta^{(m+1)}\right) < \epsilon |f\left(\bar{x}, \Phi^{(0)}, \Delta^{(0)}\right)|$*, return* $(\mu^{(1)}, \Phi^{(m+1)}, \Delta^{(m+1)})$*. Otherwise, replace* m *by* $m+1$ *and go to Step 3.*

In comparison to Algorithm 1, the one step-approximation does not jointly estimate the mean and precision matrix parameters; however, it does jointly estimate Φ_* and Δ_*. A similar one-step approximation was proposed by Rothman et al. (2010) for multivariate response linear regression with precision matrix estimation.

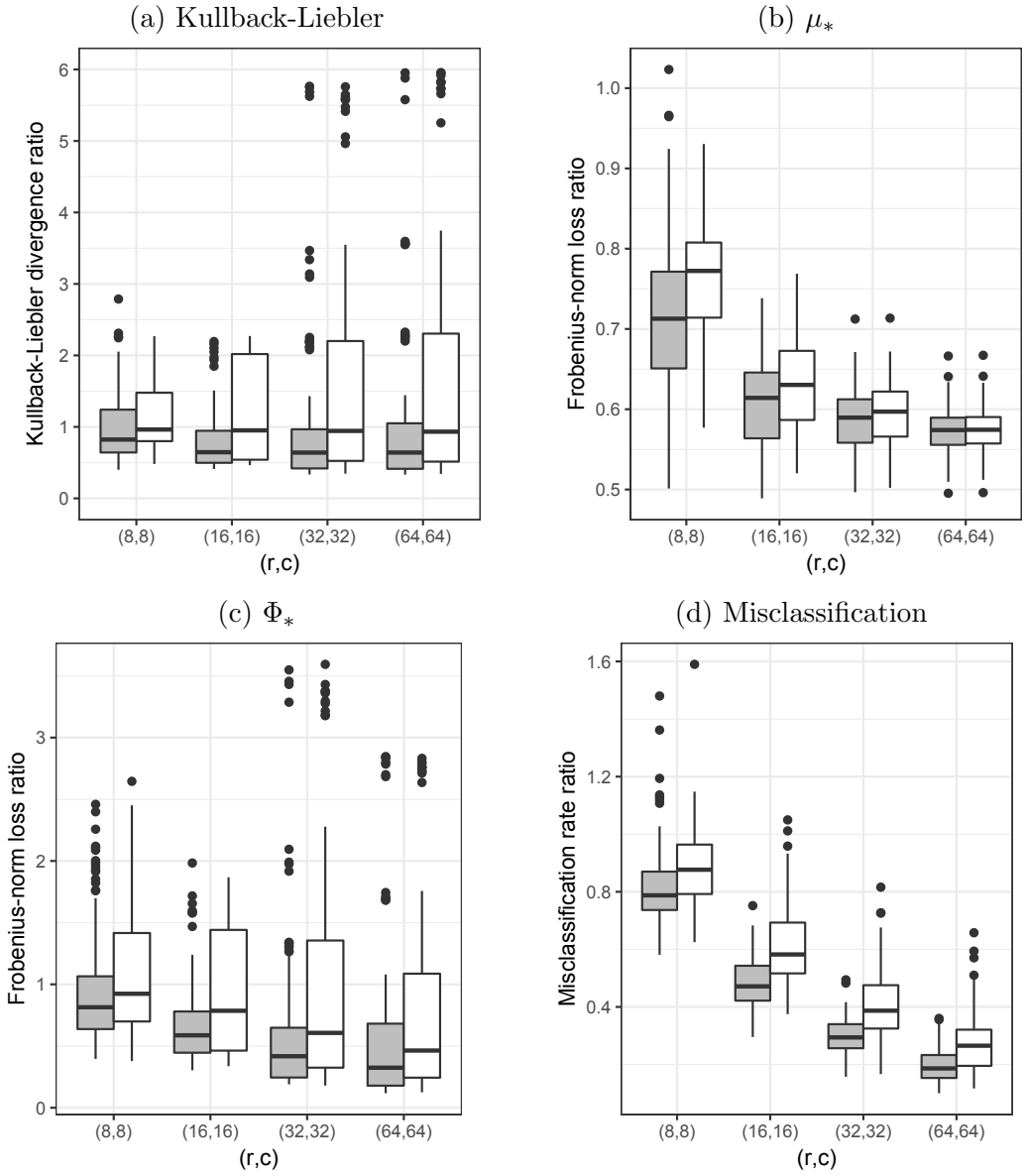
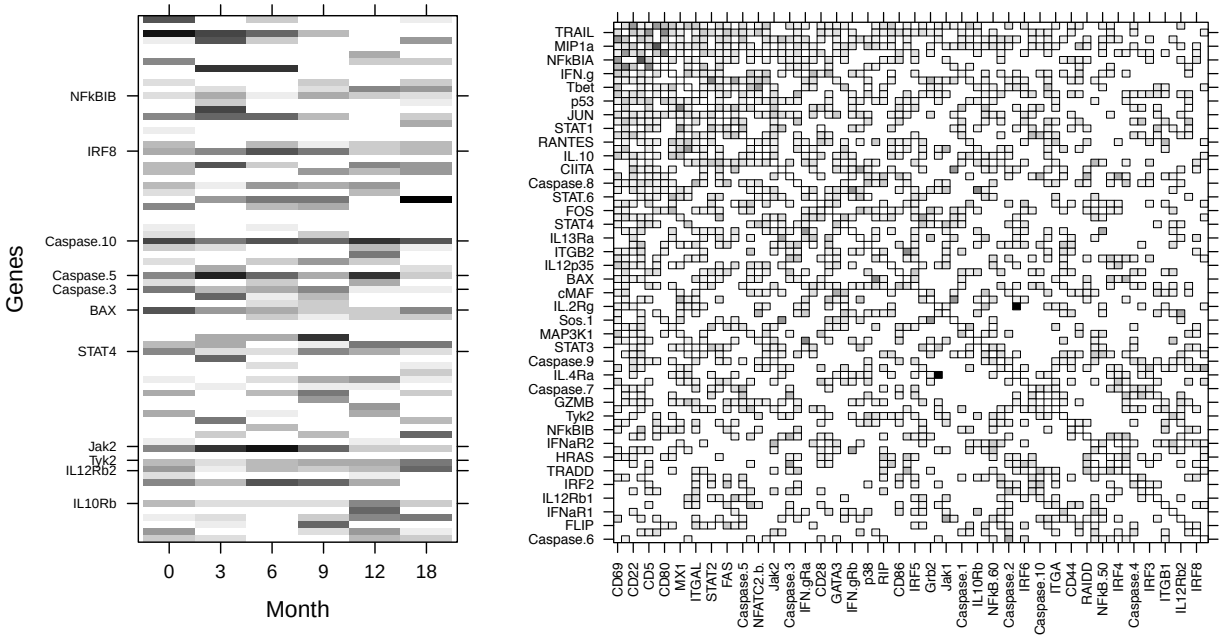


Figure 4: Loss ratios (relative to maximum likelihood) for 100 replications using the estimator from Algorithm 1 (grey boxplots) and the one-step approximation from Algorithm 2 (white boxplots). (a) displays Kullback-Liebler loss ratios, (b) displays Frobenius-norm loss ratios for estimating μ_*, (c) displays Frobenius-norm loss ratios for estimating Φ_*, and (d) displays the misclassification rate ratios.

To compare the performance of the maximum likelihood estimator, the joint estimator from Algorithm 1, and the one-step approximation from Algorithm 2, we measure multiple losses for Model 1 using the same settings as in Section 4.1. For each of the three estimators, we measured the Frobenius norm loss for estimating Φ_*, the Frobeinus norm loss for estimating μ_*, the Kullback-Liebler divergence, and the classification accuracy. For the sake of comparison, we keep losses on the same scale for different (r, c) by displaying



(a) (b)

Figure 5: (a) The absolute value of the estimated mean matrix difference between the positive and negative responders using (3) with a tuning parameter pair with leave-one-out misclassification accuracy of 47/53. Genes which had all six time points estimated to have nonzero mean differences are labeled. White entries correspond to zero, whereas darker entries indicate a larger magnitude. (b) The estimated absolute precision matrix (on the correlation scale) between genes using (3). White entries correspond to zero, whereas darker entries indicate a larger magnitude. The diagonal was set to zero for clarity of display.

the ratio the penalized estimators' losses to the maximum likelihood estimator's loss. For example, for estimating μ_* with estimator $\hat{\mu}$, we display the ratio

$$\frac{\|\hat{\mu} - \mu_*\|_F}{\|\bar{x} - \mu_*\|_F},$$

where $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_J) \in \mathbb{R}^{(r\times c)J}$ are the usual sample estimators of the J response category mean matrices. As in Section 4.2, we select tuning parameters by minimizing the misclassification rate on the validation set for both penalized estimators we consider. Ratios for the four losses are shown in Figure 4. Grey boxplots are the ratios from (3) and white boxplots are from the one-step approximation to (3). We see that jointly estimating the mean and precision matrix parameters generally led to better estimates of Φ_* compared to the maximum likelihood estimator and the one-step approximation. It is possible that we could have improved Kullback-Liebler loss and both Frobienus-norm losses if we had selected tuning parameters by maximizing a validation likelihood. However, even when we select tuning parameters to maximize validation classification accuracy, (3) is still more efficient than the the maximum likelihood estimator and one-step approximation.

10 Figures for longitudinal genomic data example

In Figure 5a, we display the estimated mean matrix absolute difference based on our fitted model using (3). Based the mean difference estimate displayed in 5a, there does not appear to be any clear evidence of a change in gene expression over time. Genes which had all six time point estimated to have nonzero mean matrix differences are labeled along the vertical axis. Of the labeled genes, Caspase 10 appears to have the largest mean differences across time. In Figure 5b, we display the absolute correlation matrix corresponding to the precision matrix estimate for the genes.

References

Clemmensen, L., Hastie, T., Witten, D., and Ersbøll, B. (2011). Sparse discriminant analysis. *Technometrics*, 53(4):406–413.

Dutilleul, P. (1999). The MLE algorithm for the matrix normal distribution. *Journal of Statistical Computation and Simulation*, 64(2):105–123.

Friedman, J. H., Hastie, T. J., and Tibshirani, R. J. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Guo, J. (2010). Simultaneous variable selection and class fusion for high-dimensional linear discriminant analysis. *Biostatistics*, 11(4):599–608.

Mai, Q., Yang, Y., and Zou, H. (2018). Multiclass sparse discriminant analysis. *Statistica Sinica*. to appear, doi:10.5705/ss.202016.0117.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rothman, A. J., Levina, E., and Zhu, J. (2010). Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962.

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, New York.

Witten, D. M. and Tibshirani, R. (2011). Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B*, 73(5):753–772.

Zhong, W. and Suslick, K. S. (2015). Matrix discriminant analysis with application to colorimetric sensor array data. *Technometrics*, 57(4):524–534.