Appendix to **Diagonal Discriminant Analysis with Feature Selection for High Dimensional Data** published in the Journal of Computational and Graphical Statistics, by Sarah E. Romanes, John T. Ormerod, and Jean Y.H. Yang.

A Proofs

Proof: Selection of penalty from M fixed probabilities

We are interested in showing there exists an explicit solution for the following set of equations, with the constraint that $\sum_{m=1}^{M} \rho_m = 1$ and $\rho_m \in (0, 1]$ for all m.

$$\begin{aligned} 2\log(\rho_1) - 2\log(\rho_1) &= C\nu_1 \\ 2\log(\rho_2) - 2\log(\rho_1) &= C\nu_2 \\ &\vdots \\ 2\log(\rho_M) - 2\log(\rho_1) &= C\nu_M. \end{aligned}$$

With ν_m representing the number of degrees of freedom for partition m (with $\nu_1 = 0$ (by definition), and $\nu_2 \leq \nu_3 \leq \cdots \leq \nu_M$), and with C representing any real number - in this case, corresponds to the penalty of choice (ie, for the AIC penalty, $C = -2$).

Now, it is clear that the first case holds no matter the choice of ρ_1, since we are left with a 0 on both the LHS and RHS. Rearranging the rest of the equations in terms of ρ_1, we are left with the following:

$$\rho_m = \rho_1 e^{C\nu_m} \qquad \text{for } m = 2, \dots M.$$

Since $\sum_{m=1}^{M} \rho_m$ = 1, we have explicit solutions for all ρ_v:

$$\begin{aligned} \rho_1 + \rho_2 + \cdots + \rho_M &= 1 \\ \rho_1 + \rho_1 e^{C\nu_2} + \cdots + \rho_1 e^{C\nu_M} &= 1 \\ \rho_1(1 + \rho_1 e^{C\nu_2} + \cdots + \rho_1 e^{C\nu_M}) &= 1 \\ \rho_1 \sum_{v=1}^{M} e^{C\nu_m} &= 1 \end{aligned}$$

so that

$$\rho_1 = \frac{1}{\sum_{m=1}^{M} e^{C\nu_m}}$$

Substituting this back into the expressions for $\rho_m, m > 1$, we have:

$$\rho_m = \frac{e^{C\nu_m}}{\sum_{\ell=1}^{M} e^{C\nu_\ell}}$$

which is guaranteed to be in $(0, 1)$ since $\nu_1 < \nu_2 \leq \nu_3 \leq \cdots \leq \nu_M$.

Proof of $\widetilde{E} = o_p(1)$

Let $X_{ij} \sim p_{jm_j}(\,\cdot\,; \boldsymbol{\theta}_{0jm})$ for some true parameter vector $\boldsymbol{\theta}_{0jm} \in \mathbb{R}^{d_m}$. Define the log-likelihood for variable j and hypothesis m as $\ell_{jm}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \ln p_{jm}(X_{ij}; \boldsymbol{\theta}_{jm})$ with corresponding MLE and "pseudo-true" value of $\boldsymbol{\theta}_{jm}$ as

$$\widehat{\boldsymbol{\theta}}_{jm} = \arg\max_{\boldsymbol{\theta}_{jm}} \{\ell_{jm}(\boldsymbol{\theta}_{jm})\} \qquad \text{and} \qquad \boldsymbol{\theta}^*_{jm} = \arg\max_{\boldsymbol{\theta}_{jm}} \left\{\mathbb{E}\left(n^{-1}\ell_{jm}(\boldsymbol{\theta}_{jm})\right)\right\},$$

respectively. We will assume conditions on the likelihood and parameter space such that $\mathbb{E}\left[n^{-1}\ell_{jm}(\boldsymbol{\theta}^*_{jm})\right] \to \ell^*_{jm}$ for $1 \leq j \leq p$, $1 \leq m \leq M$. Using the theory summarised in Ormerod et al. (2017) based on Vuong (1989) and van der Vaart (1998) we have two main cases to consider.

- **[Underfitting case]** – Suppose $\ell^*_{jm_j} > \ell^*_{jm}$ for some $m \neq m_j$. Then

  $$\tfrac{1}{2n}\left[\lambda_{jm_j}(\mathbf{X}_j) - \lambda_{jm}(\mathbf{X}_j)\right] \xrightarrow{P} \ell^*_{jm_j} - \ell^*_{jm} = \Delta_{jm} > 0$$

  and so $(1/2)\lambda_{jm}(\mathbf{X}_j) = n[\Delta_{jm} + o_p(1)]$.

- **[Overfitting case]** – Suppose $\ell^*_{jm} = \ell^*_{jm_j}$ for some $m \neq m_j$ and let $\nu_m = d_m - d_{m_j}$. Then $\lambda_{jm}(\mathbf{X}_j) - \lambda_{jm_j}(\mathbf{X}_j) \xrightarrow{D} \chi^2_{\widetilde{\nu}_{jm}}$.

The following lemma will be useful.

Lemma 1 (Gasull et al., 2015): If X_j, $j = 1, \ldots, p$, are independent χ^2_ν random variables, and $M_p = \max_{1 \leq j \leq p}\{X_j\}$, then $\frac{1}{2}M_p - [\ln(p) + (\nu/2 - 1)\ln\ln(p) - \ln\Gamma(\nu/2)] \xrightarrow{D} G$, as $p \to \infty$ where G is a Gumbel distributed random variable.

Define $\mathcal{J}_m = \{j \colon \gamma_{0jm} = 1\}$ and $\mathcal{T}_m = \{j \colon \ell^*_{jm} = \ell^*_{jm_j}\}$. Here $\mathcal{J}_k$ is the set of true variables over the kth set of hypotheses, and $\mathcal{T}_m$ is the union of over-fitting and true models over the kth set of hypotheses. We define and decompose the the error as

$$\begin{aligned} E &= \sum_{j=1}^{p} 1 - \widehat{\gamma}_{jm_j}(\mathbf{X}_j) + \sum_{j=1}^{p} \sum_{m \neq m_j} \widehat{\gamma}_{jm}(\mathbf{X}_j) \\ &= 2 \sum_{j=1}^{p} \sum_{m \neq m_j} \widehat{\gamma}_{jm}(\mathbf{X}_j) = 2 \sum_{m=1}^{M} \sum_{j \notin \mathcal{J}_m} \widehat{\gamma}_{jm}(\mathbf{X}_j) \\ &= \underbrace{2 \sum_{m=2}^{M} \sum_{j \in \mathcal{O}_m} \widehat{\gamma}_{jm}(\mathbf{X}_j)}_{\text{Overfitting models}} + \underbrace{2 \sum_{m=1}^{M} \sum_{j \in \mathcal{U}_m} \widehat{\gamma}_{jm}(\mathbf{X}_j)}_{\text{Underfitting models}} \\ &\triangleq E_{\mathcal{O}} + E_{\mathcal{U}}, \end{aligned}$$

where $\mathcal{O}_m = \mathcal{J}_m^c \cap \mathcal{T}_m$, and $\mathcal{U}_m = \mathcal{J}_m^c \cap \mathcal{T}_m^c$.

Note that for $E_{\mathcal{O}}$ the index m is summation does not include $m = 1$ since the null model cannot be an overfitting model. Next, we consider $E_{\mathcal{O}}$ where the true model is used as the null hypothesis and rewrite $E_{\mathcal{O}}$ as

$$E_{\mathcal{O}} = 2 \sum_{m=2}^{M} \sum_{j \in \mathcal{O}_m} \frac{\exp\left[\frac{1}{2}\widetilde{\lambda}_{jm}(\mathbf{X}_j) - \widetilde{\nu}_m \left\{\ln(n) + 2\ln(p)\right\}\right]}{\sum_{\ell=1}^{M} \exp\left[\frac{1}{2}\widetilde{\lambda}_{j\ell}(\mathbf{X}_j) - \widetilde{\nu}_\ell \left\{\ln(n) + 2\ln(p)\right\}\right]}$$

where $\widetilde{\lambda}_{jm}(\mathbf{X}_j) = \lambda_{jm}(\mathbf{X}_j) - \lambda_{jm_j}(\mathbf{X}_j)$. Using a chi-square approximation over the set of overfitting models in place of LRT statistics with $U_{jm} \overset{\text{iid}}{\sim} \chi^2_{\widetilde{\nu}_m}$ we obtain an approximation $\widetilde{E}_{\mathcal{O}}$ of $E_{\mathcal{O}}$ given by

$$\begin{aligned} \widetilde{E}_{\mathcal{O}} &= 2 \sum_{m=1}^{M} \sum_{j \in \mathcal{J}_m^c \cap \mathcal{T}_m} \frac{\exp\left[\frac{1}{2}U_{jm} - \widetilde{\nu}_m \left\{\ln(n) + 2\ln(p)\right\}\right]}{\sum_{\ell=1}^{M} \exp\left[\frac{1}{2}\lambda_{j\ell}(\mathbf{X}_j) - \widetilde{\nu}_\ell \left\{\ln(n) + 2\ln(p)\right\}\right]} \\ &\leq 2 \sum_{m=1}^{M} \sum_{j \in \mathcal{O}_m} \exp\left[\tfrac{1}{2}U_{jm} - \widetilde{\nu}_m \left\{\ln(n) + 2\ln(p)\right\}\right] \\ &\leq 2 \sum_{m=1}^{M} p_{1m} \exp\left[\max_{j \in \mathcal{O}_m} \tfrac{1}{2}Z_{jm} - \widetilde{\nu}_k \left\{\ln(n) + 2\ln(p)\right\}\right] \\ &\to \sum_{m=1}^{M} \left(\frac{p_{1m}^2}{p^{2\widetilde{\nu}_m} \ln(p_{1m})}\right) \left(\frac{\ln(p_{1m})^{1/2}}{n}\right)^{\widetilde{\nu}_m} \frac{2\exp(G_m)}{\Gamma(\widetilde{\nu}_m/2)} \end{aligned}$$

where $p_{1k} = |\mathcal{O}_m|$, the last line follows from Lemma 1 with $G_1, \ldots, G_m$ being independent Gumbel distributions. Note that $E_{\mathcal{O}} = o_p(1)$ provided $\ln(p)/n \to 0$. Similarly, for

$E_{\mathcal{U}}$ we have

$$\begin{aligned} E_{\mathcal{U}} &\leq 2\sum_{m=1}^{M}\sum_{j\in\mathcal{U}_m}\exp\left[\tfrac{1}{2}\widetilde{\lambda}_{jm}(\mathbf{X}_j) - \widetilde{\nu}_m\{\ln(n)+2\ln(p)\}\right] \\ &= 2\sum_{m=1}^{M}\sum_{j\in\mathcal{U}_m}\exp\left[-\tfrac{1}{2}n\{\widetilde{\Delta}_{jm}+o_p(1)\} - \widetilde{\nu}_m\{\ln(n)+2\ln(p)\}\right] \\ &\leq 2\sum_{m=1}^{M}\frac{p_{0m}}{p^{2\widetilde{\nu}_m}}\exp\left[-\frac{1}{2}n\left\{\min_{j\in\mathcal{U}_m}\widetilde{\Delta}_{jm}\right\} - \widetilde{\nu}_m\ln(n)\right] + \text{smaller terms} \\ &= o_p(1) \end{aligned}$$

where $\widetilde{\Delta}_{jm} = \Delta_{jm_j} - \Delta_{jm} > 0$, the above $\widetilde{\nu}_m$ may be positive or negative, and $p_{0m} = |\mathcal{U}_m|$. The only potentially problematic term occurs for when $m = 1$ since $\nu_1 = 0$. For this case $E_{\mathcal{U}} = o_p(1)$ provided

$$p_{01}\exp\left[-\frac{1}{2}n\left\{\min_{j\in\mathcal{U}_1}\widetilde{\Delta}_{j1}\right\}\right] = o(1).$$

Which is true provided $\ln(p)/n \to 0$. Hence, $\widetilde{E}_{\mathcal{O}} + E_{\mathcal{U}} = o_p(1)$.

B Competing Methods

Competing methods are listed in Table 1 below.

Method	**Paper**	`R` **Implementation**
DLDA/DQDA	Dudoit et al. (2002)	`sparsediscrim` - Ramey (2017)
Penalized LDA	Witten and Tibshirani (2011)	`penalizedLDA` - Witten (2015)
Nearest Shrunken Centroids	Tibshirani et al. (2003)	`pamr` - Hastie et al. (2014)
Random Forest	Breiman (2001)	`randomForest` - Liaw and Wiener (2002)
Support Vector Machine (SVM)	Cortes and Vapnik (1995)	`e1071` - Meyer et al. (2017)
Multinomial logistic regression with LASSO regularization	Tibshirani (1996)	`glmnet` - Friedman et al. (2010)
K nearest neighbours classifier (K=1)	Cover and Hart (1967)	`class` - Venables and Ripley (2002)

Table 1: ML methods used in our comparisons

For the ML methods that required hyper-parameter tuning we detail such tuning as follows:

- Random Forest was tuned following guidelines for Microarray data from Díaz-Uriarte and Alvarez de Andrés (2006).

- Default parameters were used for SVM.
- `cv.glmnet` was used to select the best value of the regularisation parameter λ for multinomial logistic regression with LASSO regularisation.
- $K = 1$ was used for KNN as this produced the best prediction results.
- In Nearest Shrunken Centroids, the shrinkage parameter Δ was tuned following procedures from Tibshirani et al. (2003) - we first trained using `pamr.train` before using `pamr.adaptthresh` to adaptively search for a set of good threshold scales to use in further retraining.
- Penalized LDA - optimal parameters K (number of discriminant vectors to be used) and λ (regularisation paramter) were determined running `PenalizedLDA.cv` before training (as guided by the PenalizedLDA vignette).

R code implementation

We include general format of the `R` code used in this paper for complete transparency.

- Discriminant Analysis methods
  - DLDA `sparsediscrim`

    ```
    res = dlda(mX.train, vy.train)
    vals = as.numeric(predict(res, newdata = mX.test)$class)
    ```
  - DQDA `sparsediscrim`

    ```
    res = dqda(mX.train, vy.train)
    vals = as.numeric(predict(res, newdata = mX.test)$class)
    ```
  - Penalized LDA `penalizedLDA`

    ```
    cv.out = PenalizedLDA.cv(mX.train,vy.train,type="standard",
    lambdas=c(1e-4,1e-3,1e-2,.1,1,10))
    vals = PenalizedLDA(mX.train,vy.train,type="standard",
    xte=mX.test, lambda=cv.out$bestlambda,K=cv.out$bestK)
    ```
  - NSC `pamr`

    ```
    mydata = list(x=t(mX.train),y=as.factor(vy.train), geneid=1:p)
    ```

```
res = pamr.train(mydata)
new.scales = pamr.adaptthresh(res)
res = pamr.train(mydata, threshold.scale=new.scales)
vals = as.numeric(pamr.predict(res, t(mX.test), threshold=new.scales))
```

- Random Forest `randomForest`

```
res = randomForest(mX.train, vy.train, ntree=500,
mtry=floor(sqrt(p)),nodesize=1)
vals = as.numeric(predict(res, newdata = mX.test))
```

- Support Vector Machine `e1071`

```
res = svm(mX.train, vy.train, probability=FALSE)
vals = predict(res,mX.test, decision.values=TRUE, probability=FALSE)
```

- Multinomial Response LASSO `glmnet`

```
res = cv.glmnet(mX.train,vy.train,family="multinomial")
vals = predict(cv, newx = mX.test, s = "lambda.min", type = "class")
```

- K nearest neighbours `class`

```
vals = knn(mX.train, mX.test, vy.train, k=1)
```

Generation of covariance matrix for simulations

```
genZeroMeanSparceCovNormal = function(n,nBlocks,blockSize,perc,symmetric,
                                      permute=TRUE,seed)
{
    set.seed(seed)
    p = nBlocks*blockSize
    nnz = round(nBlocks*blockSize*perc)
    mX = c()
    lmSigma = list()
    lmZ = list()

    for (b in 1:nBlocks)
```

```
    {
        randRows = sample(1:blockSize, nnz, replace = TRUE)
        randCols = sample(1:blockSize, nnz, replace = TRUE)

        if (symmetric) {
            vr = c(1:blockSize,randRows,randCols)
            vc = c(1:blockSize,randCols,randRows)
        } else {
            vr = c(1:blockSize,randRows)
            vc = c(1:blockSize,randCols)
        }

        vals = rnorm(length(vr))
        mA = sparseMatrix(x=vals, i=vr, j=vc)
        lmSigma[[b]] = t(mA)%*%mA
        mZ = t(mA%*%matrix(rnorm(n*blockSize),blockSize,n))
        mZ = matrix(mZ,n,blockSize)

        lmZ[[b]] = mZ
        mX = cbind(mX,mZ)

    }

    if (permute) {
        ord = sample(p)
        mX = mX[,ord]
    }
    return(list(mX=mX,lmSigma=lmSigma,lmZ=lmZ,ord=ord))
}
```

C Tables

Wilcoxin Signed Rank Test

The Wilcoxin Signed Rank Test was performed to see if the population mean ranks differed for classification performance between two competing ML methods. A one-sided test was performed to see if the mean ranks for the multiDA methods were lower than those of those competitors. p-values from these tests are reported in the tables below.

	multiLDA	**multiQDA**	**DLDA**	**DQDA**	**penLDA**	**NSC**	**RF**	**KNN**	**SVM**	**LASSO**
multiLDA	-	0.7264	3.607e-10	3.688e-10	4.165e-09	5.63e-10	6.037e-10	3.831e-10	3.747e-10	1.889e-06
multiQDA	0.271	-	3.607e-10	3.504e-10	3.093e-09	5.32e-10	7.54e-10	3.832e-10	3.72e-10	1.846e-07

Table 2: p-values for Wilcoxin Signed Rank test for TCGA dataset

	multiLDA	**multiQDA**	**DLDA**	**DQDA**	**penLDA**	**NSC**	**RF**	**KNN**	**SVM**	**LASSO**
multiLDA	-	0.02553	8.64e-10	3.175e-10	7.936e-10	1.77e-08	3.114e-10	0.8967	3.397e-10	0.9989
multiQDA	0.9755	-	7.508e-09	3.247e-10	3.162e-08	2.884e-5	3.077e-10	0.9963	5.419e-10	0.9998

Table 3: p-values for Wilcoxin Signed Rank test for SRBCT dataset

References

Peter J. Bickel and Elizaveta Levina. Some theory for Fisher's linear discriminant function, 'näive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6):989–1010, 2004. doi: 10.3150/bj/1106314847.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967. ISSN 0018-9448. doi: 10.1109/TIT.1967.1053964.

Ramon Diaz-Uriarte and Sara Alvarez de Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, Jan 2006.

Ramón Díaz-Uriarte and Sara Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7(1):3, Jan 2006.

Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

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

Armengol Gasull, José A. López-Salcedo, and Frederic Utzet. Maxima of Gamma random variables and other Weibull-like distributions and the Lambert W function. *TEST*, 24(4): 714–733, 2015. ISSN 1863-8260. doi: 10.1007/s11749-015-0431-9.

Henry Han and Xiao-Li Li. Multi-resolution independent component analysis for high-performance tumor classification and biomarker discovery. *BMC Bioinformatics*, 12(1):S7, 2011.

T. Hastie, R. Tibshirani, Balasubramanian Narasimhan, and Gil Chu. *pamr: PAM: prediction analysis for microarrays*, 2014. R package version 1.55.

Andy Liaw and Matthew Wiener. Classification and Regression by randomForest. *R News*, 2 (3):18–22, 2002.

David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2017. R package version 1.6-8.

J. T. Ormerod, M. Stewart, W. Yu, and S. E. Romanes. Bayesian hypothesis tests with diffuse priors: Can we have our cake and eat it too? 2017. URL `arxiv.org/pdf/1710.09146.pdf`.

John A. Ramey. *sparsediscrim: Sparse and Regularized Discriminant Analysis*, 2017. R package version 0.2.4.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246.

Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, 18(1):104–117, 02 2003. doi: 10.1214/ss/1056397488.

A. W. van der Vaart. *Asymptotic statistics, volume 3 of Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1998.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

Quang H. Vuong. Likelihood ratio tests for model selection and nonnested hypotheses. *Econometrica*, 57(2):307–333, 1989.

Daniela Witten. *penalizedLDA: Penalized classification using Fisher's Linear Discriminant*, 2015. R package version 1.1.

Daniela M. Witten and Robert Tibshirani. Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):753–772, 2011. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2011.00783.x.