MSA200/TMS041 Multivariate Analysis

Lecture 8

Rebecka Jörnsten

Mathematical Sciences
University of Gothenburg and Chalmers University of Technology

Back to Discriminant analysis

As mentioned in the previous lecture, one problem with LDA stems from the instability of the estimate $\hat{\Sigma}$ when n is small and/or p is large and/or x-features are correlated.
Let's look at the source of this problem in more detail.

Inverse of Σ

Consider the eigendecomposition of $\hat{\Sigma} = UDU'$, where U are the eigenvectors and $U'U = I$ and D is a diagonal matrix containing the eigenvalues

$$\begin{pmatrix} d_1^2 & 0 & \cdots \\ 0 & d_2^2 & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & 0 & d_p^2 \end{pmatrix}$$

where $d_1^2 > d_2^2 > \cdots$

Inverse of Σ

We can then write the inverse as

$$\hat{\Sigma}^{-1} = UD^{-1}U'$$

and plugging this into the expression for the DA rule

$$\hat{c}(x) = \arg\min_c (x - \hat{\mu}_c)' \hat{\Sigma}^{-1} (x - \hat{\mu}_c) =$$

$$= (x - \hat{\mu}_c)' UD^{-1}U' (x - \hat{\mu}_c) =$$

$$= [U'(x - \hat{\mu}_c)]' D^{-1} [U'(x - \hat{\mu}_c)] =$$

$$= (\tilde{x} - \tilde{\mu})' D^{-1} (\tilde{x} - \tilde{\mu}) =$$

$$= \sum_{j=1}^{p} \frac{(\tilde{x}_j - \tilde{\mu}_j)^2}{d_j^2}$$

which is a weighted euclidean distance between x and μ in the new coordinate system corresponding to the principal component directions U of $\hat{\Sigma}$.

Inverse of Σ

So LDA is really just nearest centroids in the new coordinate system that you get by rotating the data by U and scaling it by D. Writing $\hat{\Sigma} = UDU' = UD^{1/2}D^{1/2}U'$, we have that

$$\hat{\Sigma}^{-1} = UD^{-1/2}D^{-1/2}U' = \hat{\Sigma}^{-1/2}\hat{\Sigma}^{-1/2}$$

where we define $\hat{\Sigma}^{-1/2} = D^{-1/2}U'$ (square root of a diagonal matrix is just the square root of the elements).
Therefore we can write

$$(x - \hat{\mu}_c)'\hat{\Sigma}^{-1}(x - \hat{\mu}_c) = [\hat{\Sigma}^{-1/2}(x - \hat{\mu}_c)]'[\hat{\Sigma}^{-1/2}(x - \hat{\mu}_c)].$$

Inverse of Σ

The operation $\hat{\Sigma}^{-1/2}$ on x is called *sphering* the data. Why?

$$Cov(\hat{\Sigma}^{-1/2}X) = E[\hat{\Sigma}^{-1/2}X(\hat{\Sigma}^{-1/2}X)'] =$$

$$= E[\hat{\Sigma}^{-1/2}XX'\hat{\Sigma}^{-1/2}] = \hat{\Sigma}^{-1/2}E[XX']\hat{\Sigma}^{-1/2} = \hat{\Sigma}^{-1/2}\hat{\Sigma}\hat{\Sigma}^{-1/2} = I$$

I.e., in the new coordinate system Xs are uncorrelated and all features have variance 1.

Inverse of Σ

When $\hat{\Sigma}$ is near singular, $\hat{\Sigma}^{-1}$ behaves poorly (or may not even exist). The estimate is numerically unstable and small changes to the data can lead to big change for the inverse (and thus how you rotate the data before applying nearest centroids $\rightarrow$ poor classification performance.

The source of the problems lie in the direction u_j corresponding to small eigenvalues d_j since d_j appears in the denominator in the weighted euclidean distance computation. Small ds "blows up" the distance computation.

How do we fix this? The solution is to stabilize the inverse by reducing the influence of these small eigenvalues. This is done quite easily by simply adding something to the diagonal of $\hat{\Sigma}$ before you take the inverse.

Inverse of Σ

Use $\tilde{\Sigma} = (\hat{\Sigma} + \lambda I)$ and its inverse $\tilde{\Sigma}^{-1} = (\hat{\Sigma} + \lambda I)^{-1}$.
The impact of this is mainly limited to the small eigenvalues as we can see from the following

$$\hat{\Sigma} + \lambda I = UDU' + \lambda I = UDU' + \lambda UU' = U(D + \lambda I)U'$$

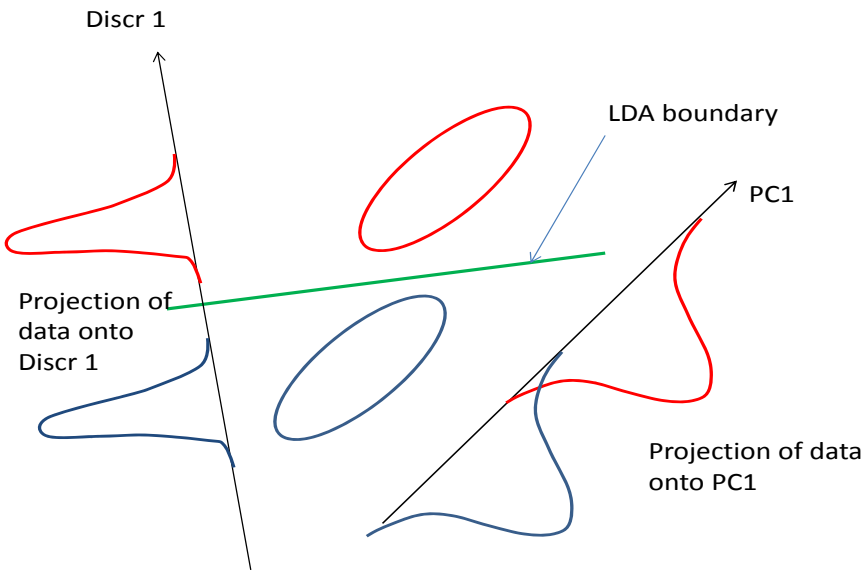
For large d_j the contribution λ is negligible.
Using $\tilde{\Sigma}^{-1}$ in your DA rule is called *penalized DA* (or regularized DA). When $\lambda = 0$ PDA is the same as LDA. If you make λ really big it starts to dominate the $d_j, \forall j$ which essentially means you start ignoring the correlation and scale structure in the data (get closer and closer to nearest centroids).

Flexible DA methods

Penalized DA addresses one problem with LDA, poor performance due to unstable estimates of $\hat{\Sigma}^{-1}$ (high variance). We also need to be concerned with potential BIAS, meaning the linear boundaries that LDA implicitly assumes are too simplistic to separate the classes from eachother.
One extension is then to use QDA (quadratic DA) we already looked at. This assumes that each class has its own correlation and scale structure. It leads to quadratic boundaries in x-space and is quite costly in terms of the number of parameters you need to estimate. This can reduce BIAS but lead to a large increase in VARIANCE so the end result is little or no improvement over LDA (or even worse performance if VARIANCE grows quickly as would be the case for very large p).

PDA tries to fix the problem with LDA's high variance by penalizing the estimate $\hat{\Sigma}$ as $\hat{\Sigma} + \lambda I$.
Another way to deal with this problem is to use a reduced rank approximation of Σ instead. That is, use the leading principal components only!
Is this a good idea? Often, yes - BUT you have to be very careful. It is quite possible that the class means are not well separated along the leading PC directions of $\hat{\Sigma}$.

Look at the distribution of the data when projected onto the first principal component. All the variation is carried with the data onto the projection and the distributions overlap, meaning classification is not going to be perfect.



What does LDA do then? We utilize the distance

$$(x - \hat{\mu}_c)'\hat{\Sigma}^{-1}(x - \hat{\mu}_c) = (U'(x - \hat{\mu}_c))'D^{-1}(U'(x - \hat{\mu}_c))$$

where $\hat{\Sigma} = UDU'$.
As discussed before, this means we *sphere* the data and use the nearest centroid in the new coordinate system with data $\tilde{x} = D^{-1/2}U'x$.
If we use only the leading principal components that is equivalent to taking u_1, u_2 from $U = (u_1, u_2, \cdots, u_p)$ and creating a new, lower-dimensional data set comprising $\tilde{x}_1 = d_1^{-1}xu_1$ and $\tilde{x}_2 = d_2^{-1}xu_2$.

The above procedure is blind to the placement of the centroids, μ_c. The μ_c are p-dimensional but they lie in a subspace of dimension $\leq C - 1$ (just consider two classes, a one-dimensional line connects those, for three classes a 2D-plane connects the three centroids, or if they are placed along a one-dimensional line in 3D space a 1D-line spans the centroids.).
R.A. Fisher had the following idea: What if you project onto a lower dimensional space and do the classification there?
The goal: *Find the optimal projection in* $\leq C - 1$*-space such that the centroids are as spread out as possible while the within-class variance (variance around centroids) is as small as possible*.
The primary goal is the centroid spread and the secondary goal you are also trying to fulfill is the limitation of the within-class spread.
Note - this is NOT equivalent to PCA! (see figure on earlier slide).

Fisher's LDA - FLDA

Some notation:

- $\Sigma = \Sigma_W$ is the within-class variance.
- Σ_B is the *between-class* variance, the spread of the centroids

We define these as

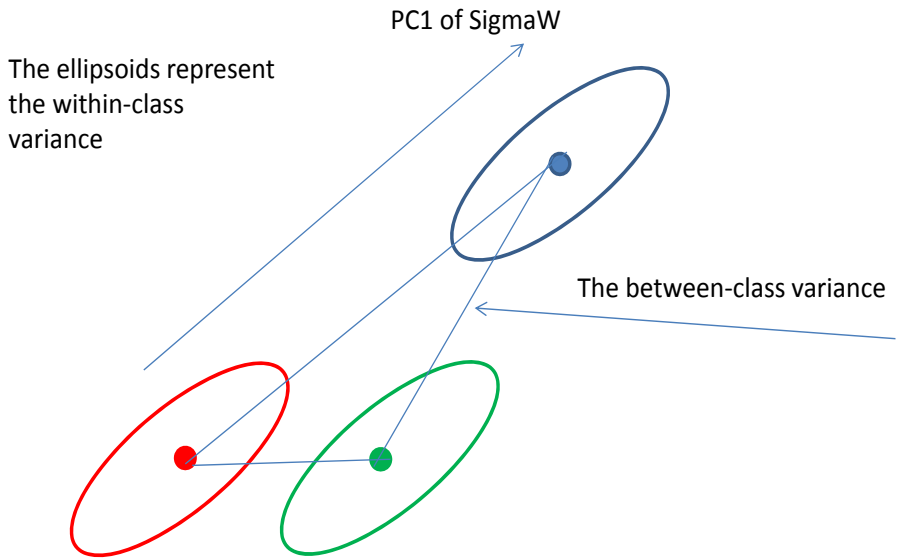
$$\Sigma_W = \sum_{c=1}^{C} \sum_{y_i=c} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)'/(N - C)$$

and

$$\Sigma_B = \sum_{c=1}^{C} (\hat{\mu}_c - \bar{x})(\hat{\mu}_c - \bar{x})'/(C - 1)$$

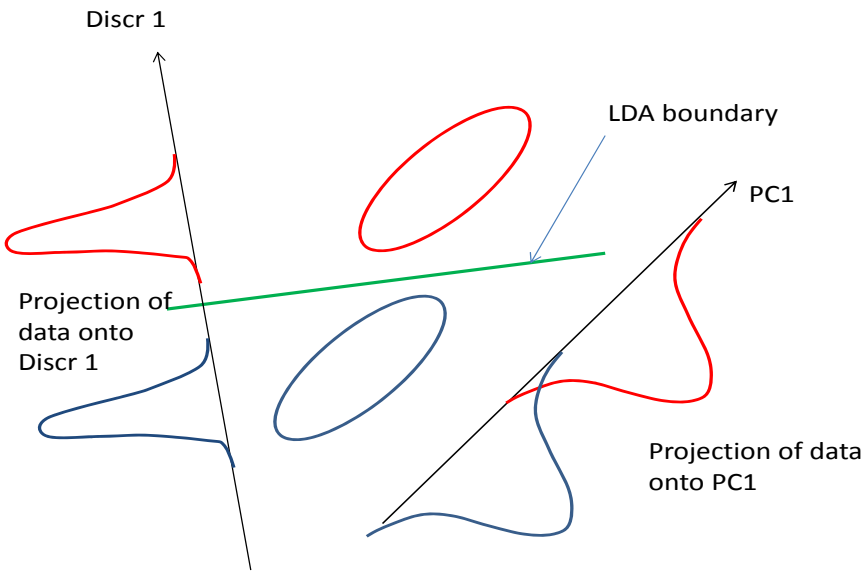
Fisher's LDA - FLDA

Here's an illustration of the between-class variance Σ_B and the within-class variance Σ_W



FLDA

FLDA means finding a projection other than the leading PCs such that the centroids are spread when taking the within-class variance into account.



Mathematically, Fisher's problem can be written as
Find directions a such that

$$\max_a \frac{a'\Sigma_B a}{a'\Sigma_W a}$$

Why?
Well, consider that we have projected data x onto directions a, then

$$V(a'X) = a'\Sigma_W a$$

so this is how much within-class variance we have in the new coordinate system.
Denoting the centroids by M we have

$$V(a'M) = a'\Sigma_B a$$

so this is how spread out the centroids are in the new coordinate system.
Fisher's problem simply states we want to maximize the centroids spread compared to the within-class spread.

The ratio

$$\frac{a'\Sigma_B a}{a'\Sigma_W a}$$

is called the Rayleigh quotient.
How do we go about maximizing this for not one, but several directions a (since 1-D projections may not suffice to separate $C > 2$ classes.

We write the problem as

$$\max_a a'\Sigma_B a \quad \textit{subject to} \quad a'\Sigma_W a = I$$

Note, our primary goal is maximizing the between-class spread. Our secondary goal is represented as a contraint where we say the directions should sphere the data.
We can rewrite this using Lagrangian methods as

$$\min_a -\frac{1}{2}a'\Sigma_B a + \frac{1}{2}\lambda(a'\Sigma_W a - I) = ***$$

where λ is the Lagrangian parameter.

We find the minimizer by taking derivatives and setting to 0:

$$\frac{\partial * **}{\partial a} = -\Sigma_B a + \lambda \Sigma_W a = 0$$

We can write

$$\Sigma_B a = \lambda \Sigma_W a$$

or

$$(\Sigma_W^{-1} \Sigma_B) a = \lambda a$$

$$(\Sigma_W^{-1}\Sigma_B)a = \lambda a$$

looks just like an eigenvalue problem!
That means that the optimal directions for separating the class centroids are the vectors a that are eigenvectors of the matrix $\Sigma_W^{-1}\Sigma_B$.
There's one problem - this matrix is not symmetric. Therefore this is not a standard eigenvalue problem but requires a *generalized eigendecomposition*.

Solution to

$$(\Sigma_W^{-1}\Sigma_B)a = \lambda a$$

Write $\Sigma_W = \Sigma_W^{1/2}\Sigma_W^{1/2}$ (which you do by defining $\Sigma = UDU'$ and $D = D^{1/2}D^{1/2}$ as before).

Plug in above to obtain

$$(\Sigma_W^{-1/2}\Sigma_B\Sigma_W^{-1/2})(\Sigma_W^{1/2}a) = \lambda(\Sigma_W^{1/2}a) =$$

$$(\Sigma_W^{-1/2}\Sigma_B\Sigma_W^{-1/2})w = \lambda w$$

which is a standard eigenproblem for w since the matrix $(\Sigma_W^{-1/2}\Sigma_B\Sigma_W^{-1/2})$ is symmetric.

You solve for the eigenvectors w and compute the original vectors that solve Fisher's problem as

$$v = \Sigma_W^{-1/2} w$$

These are the directions to project onto to separate the class means as far as possible given the within-class variance!

FLDA

- FLDA finds the optimal subspace separation of the centroids given within-class variance
- It corresponds to an eigendecomposition of $\Sigma_W^{-1}\Sigma_B$
- The data projected onto these eigenvectors are called *discriminant variables*
- Reduced dimension or reduced rank LDA means that you use only the leading eigenvectors of $\Sigma_W^{-1}\Sigma_B$

Flexible discriminant analysis, FDA

Why did we bother presenting LDA in so many different ways?
Well, you got so see that one arrives at the same classifiers from very different working assumptions: assumed data distribution or eigendecomposition idea.
Another reason is that introducing Fisher's version of LDA, which is more geometric, allow for generalizations.

Flexible discriminant analysis, FDA

In the mid-90's the Stanford group (Trevor Hastie, Robert Tibshirani and Andreas Buja and others) used Mardia's *Optimal Scoring* to reformulate DA as a regression problem.
This had two huge benefits:

- Flexible extensions: use polynomial and other more complex regression models to augment LDA
- Regularization: use penalized and sparse regression methods (feature selection) to reduce the variance of LDA

Flexible discriminant analysis, FDA

First, let's review how the FLDA discriminant variables are computed.

- Compute the centroid matrix $M = (\hat{\mu}_1, \hat{\mu}_2, \cdots, \hat{\mu}_C)$ which is a $C \times p$ matrix.
- "Sphere" the μs with the within-class covariance, Σ_W: $M^* = M\Sigma_W^{-1/2}$
- Compute the between-variance of the sphered centroids:

$$\Sigma_B^* = Cov(M^*) = (M^* - \bar{x}^*)(M^* - \bar{x}^*)'/C - 1$$

- The eigendecomposition of $\Sigma_B^* = V^* D_B V^{*'}$ provide the optimal discriminant vectors (eigenvectors of $\Sigma_W^{-1}\Sigma_B$)

Flexible discriminant analysis, FDA

- The discriminant variables (data projections to use) are obtained as

$$l - th\ variable: \ x_l^d = V_l^{*'}(\Sigma_W^{-1/2}x)$$

(you have to rotate the data to work in the same coordinate space)
Equivalently:

$$l - th\ variable: \ x_l^d = (\Sigma_W^{-1/2}V_l^{*'})x = V_l'x$$

where V are the discriminant directions in the original coordinate system)

Optimal Scoring

How do we turn this into a regression problem?

- We already know from previous lectures that LDA is pretty well approximated by 0/1 regression for a 2-class problem
- The problem was that we may "mask" some classes if we attempt 0/1 regression when we have $C > 2$ classes
- However, if we do several regression fits, when the classes takes turn to be the "1" in 0/1 regression, we will be able to separate all the classes.

Optimal Scoring

1. Create a $N \times C$ matrix Y where column k has 1s for observations i belonging to class k and 0 elsewhere.
2. Regress Y on the data matrix X $(N \times p)$ using least squares.
   - Results in a least-squares coefficient matrix B $(p \times C)$ where $\hat{B} = (X'X)^{-1}X'Y$
   - Quick recap of regression: want to find B to minimize $|| Y - XB ||^2 = (Y - XB)'(Y - XB)$
     Take derivatives with respect to B and set to 0:
     $-2X'(Y - XB) = 0$ and solve for B
   - The fitted values (values on the regression lines) are given by $\hat{Y} = X\hat{B} = X(X'X)^{-1}X'Y = HY$ where the hat-matrix $H = X(X'X)^{-1}X'$ $(N \times N)$ is a project of Y onto the space spanned by X.

3 Perform an eigendecomposition of $Y'\hat{Y} = Y'HY$

$$(Y'HY)\theta = \lambda\theta$$

One can show that θ is directly proportional to the discriminant vectors V we defined before!

To see this, compute $X'X$ and $X'Y$.
$X'X = \Sigma_W$ except for a normalizing factor.
$X'Y = M$, the $p \times C$ matrix of class centroids (except for a normalizing factor).
(Try this yourself by writing out the 0/1 Y-matrix and the data matrix X).
Now, $\hat{B} = (X'X)^{-1}X'Y = \Sigma_W^{-1}M$ and therefore it follows that

$$Y'\hat{Y} = Y'X(X'X)^{-1}X'Y = M'\Sigma_W^{-1}M =$$

$$= (\Sigma_W^{-1/2}M)'(\Sigma_W^{-1/2}M) = M^{*'}M^* = \Sigma_B^*$$

That is, the eigendecomposition problem in optimal scoring is the eigendecomposition of the between-centroid spread in the new coordinate system (taking within-class variance into account) = Same thing as FLDA!!!

Optimal Scoring

Why bother?
The point is that once we turn this into a regression problem it is easy to see how we can come up with extensions t the method.
You can use polynomial regression, nonlinear regression, semi-parametric regression, regressions with priors.... anything you want!
This idea is called *Flexible Discriminant Analysis*

Regularized Discriminant Analysis

We already talked about one particular kind of penalized DA: when we used the inverse of $\hat{\Sigma}_W + \lambda I$ to rotate/sphere our data.
Our regression analogue above connects this approach to penalized (or ridge) regression. One can run other kinds of penalized regression schemes also. For example, you could do feature selection at the regression step via e.g. lasso.
This method, and variants on the same theme, is called *sparse discriminant analysis*.

Sparse Discriminant Analysis

In sparse LDA we apply an $L1$ penalty to the regression coefficients in the optimal scoring problem. The final rule will now consist of discriminant vectors based on a subset of variables.
As in standard FLDA we can plot the data and classify the data in a reduced dimensional space based on these sparse discriminant vectors.
`sparseLDA` package

Regularized Discriminant Analysis

Several methods have been proposed for regularizing the within-covariance estimates.

- In QDA we can penalize each individual within-class covariance toward a common covariance (LDA)
- We can regularize the common within-class covariance toward a diagonal matrix (RDA)
- We can assume that the within-covariance matrix *is* diagonal (naive bayes)
- We can use a ridge-penalized estimate of the covariance matrix (PDA)

NB and Shrunken centroids

A special case of Naive Bayes is to replace the within-covariance estimate by its diagonal component.
This means we assume that features are independent.
In high-dimensional settings this tends to work quite well! The classifier now works on each variable at a time

$$k(i) = \arg\min_l \sum_{j=1}^{p} \frac{(x_{ij} - \mu_{lj})^2}{\sigma_l^2}$$

where $k(i)$ is the optimal class for observation i.
Tibshirani et al (2002) proposed we not use all the variables for classification.

- Shrink the class means (centroids) toward a common value (after standardizing by the within-class standard deviation)
- We can regularize the common within-class covariance toward a diagonal matrix (RDA)
- We can assume that the within-covariance matrix *is* diagonal (naive bayes)

Shrunken centroids

- Use a diagonal covariance estimate $diag(\Sigma + s_0^2 I)$ (where a small s_0 is used to avoid having really small standard deviations in the denominator later on)
- Compute for each variable j

$$t_{kj}^* = \frac{\hat{\mu}_{kj} - \hat{\mu}_j}{m_k(s_j + s_0)}$$

  where $\hat{\mu}_j$ is the overall mean for variable j, $s_j = \hat{\Sigma}_{jj}$ and $m_k = \sqrt{\frac{1}{n_k} + \frac{1}{n}}$
- Apply a soft-threshold to t_{kj}^*: $t_{kj} = ST(t_{kj}^*, \Delta)$
- Define $\mu_{kj}^s = \mu_j + m_k(s_j + s_0)t_{kj}$
- Use these *shrunken centroids* in your classifier!
- `pamr` package

SC-RDA

In 2005, Guo et al, proposed a combination of RDA and shrunken centroids

- We can use the SC method to shrink the centroids, either in the original data space
- or in the rotated data space!
- $\arg\min_k (x - \mu_k^s)' \hat{\Sigma}_R^{-1} (x - \mu_k^s)$
- `rda` package

Going back to the original "fix" - trying the regularize the within-class covariance estimates. There have been other types of proposals here as well that also uses sparse modeling.

- Estimate the inverse covariance using sparse modeling (`graphical lasso`)
- Use sparse inverse covariance estimation and approximate this with a block-diagonal matrix (Pavlenko et al, 2008). This is like a less severe approximation than naive bayes.
- Use your regularized estimate of the within-covariance in your DA rule.

PROJECT 2

- You can work individually or in pairs - but NOT THE SAME PAIRS as last time.
- In-class presentations on WEDNESDAY MAY 4th.
- Prepare slides to post or present.
- Submit the report no later than May 6th.

PROJECT 2 - TOPICS

1. Large-scale simulations: investigate how sample size, dimensionality, correlation structure affect high-dimensional modeling.
   - A. Regression modeling (e.g. lasso, adaptive lasso, elastic net)
   - B. p-values and confidence intervals, group testing

PROJECT 2 - TOPICS

2 Method comparisons in regression.

- The `caret` package collects many of the methods we have discussed in class in addition to a huge set of additional methods. Check out this impressive list.
- Choose at least two methods from the list that we have not discussed in class and one method from class. Compare on at least two data sets (see later slide for data specification). Provide a short description of the methods.
- Compare in terms of *prediction performance on a test set*, *model complexity*, *features selection/variable importance* and say something about interpreting the results.

Suggested methods: bayesian lasso (blasso) and similar, ridge regression with feature selection (foba), independent component regression (icr), principal component regression (pcr), knn regression, supervised pc regression (superpc), spike and slab regression (spikeslab), sparse partial least squares (spls).
You can of course use functions not in the `caret` package as well!

PROJECT 2 - TOPICS

3 Method comparisons for classification

- The `caret` package collects many of the methods we have discussed in class in addition to a huge set of additional methods. Check out this impressive list.
- Choose at least two methods from the list that we have not discussed in class and one method from class. Compare on at least two data sets (see next slide for data specification). Provide a short description of the methods.
- Compare in terms of *misclassification error*, *model complexity*, *features selection/variable importance* and say something about interpreting the results.

Suggested methods: discriminant analysis with stepwise feature selection, sparse mixture da (smda), shrinkage discriminant analysis (sda), localized lda (loclda), highdimensional da (hdda), bagged flexible da (bagfda), support vector machines (multiple), partial least squares generalized linear models (plsRglm).
You can of course use functions not in the `caret` package as well!

PROJECT 2 - DATA

- For the comparison to be interesting, work with fairly high-dimensional data sets where $n \simeq p$ or $p > n$.
- You can either download such data directly or...
- You can generate such data, with more or less complicated structure, from low-dimensional data sets as follows:
  - Add 10-100s of noise features that are uncorrelated or correlated between themselves
  - Add noisy copies of the features in your low-dimensional data set ($x_new = x + noise$).
  - Subsample the sample size

PROJECT 2 - Cautionary remark

- `caret` is good for comparisons of methods but it's not as easy to get familiar with a method through caret.
- Therefore, I recommend you use caret for method comparisons but to discuss each individual method, how tuning parameters etc play a role, you use the original functions.
- Remember, you are not only to compare methods in terms of performance but also provide a description of each method and interpret the results (which means looking at what each method actually does when it models the data).

PROJECT 2 - SIGN-UP!

- Here is the doodle
- Spread out across the 3 projects or I will re-assign those that sign up later