Feature Selection, Shrinkage, and Dimension Reduction

Alexander Brenning

Department of Geography, Friedrich Schiller University Jena

Geo 408B

Feature Selection

Could there be reasons to use only a subset of the available features?

- **Prediction accuracy**: "Curse of dimensionality" (Bellman, 1957) → many parametric methods perform poorly when there is a large number of predictors, p
- **Model interpretability**: Irrelevant features lead to unnecessary complexity
- **Computational cost**: Irrelevant features increase the computing time

Feature Selection

Which approaches to features selection (and related methods) are you already familiar with?

Overview and Terminology

- **Filter methods**
  - Classroom example: AUROC ranking
- **Feature selection**
  - E.g. stepwise selection
- **Shrinkage methods**
  - Classroom example: LDA with ridge penalty
- **Dimension reduction**
  - Classroom example: principal component analysis combined with LDA

(There's some overlap between these broad classes of methods. In particular, all feature selection and filter methods reduce dimensionality; and some shrinkage methods select features.)

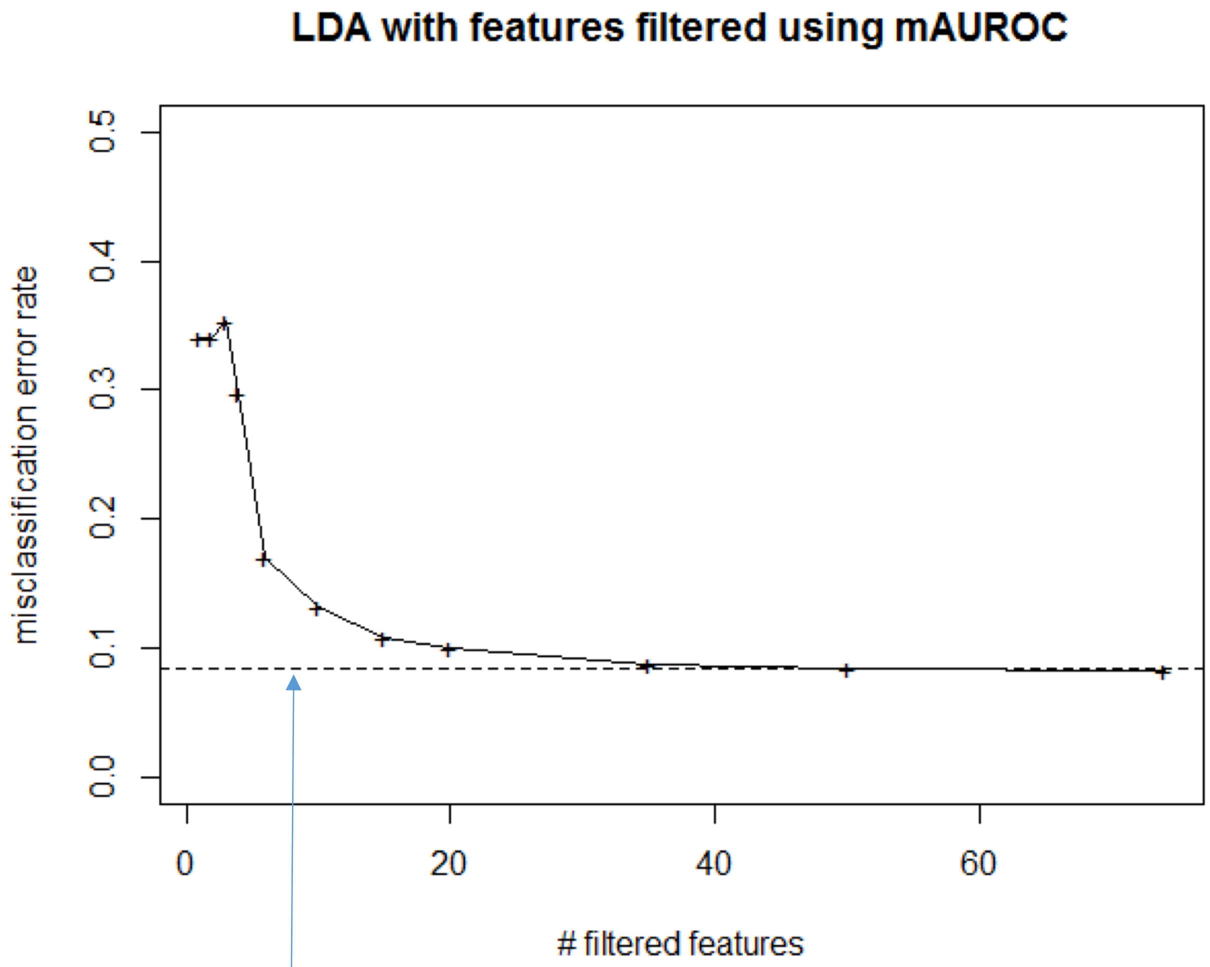
Filter Methods

- Filter features that seem promising without consideration of the specific modelling technique to be used
    1. Calculate e.g. AUROC for each feature.
    2. Fit the model using only features with AUROC greater than a specified threshold, or a specified number of top-ranking predictors.
- Note: Some of the methods available in `mlr` (and in the literature) are identical or nearly identical. Understand the maths behind the methods before using them!
- The threshold, or the number of features to be included, becomes a **hyperparameter**.
- Advantage:
    - Very efficient. Features can be filtered without fitting a model.
- Disadvantage:
    - There is no guarantee that the filtered features are "optimal" or even useful at all with the chosen model. This method likely makes poor use of the available data.
    - The filtered features may indeed be strongly correlated with each other.

Filter Methods

Classroom example:

- Rank features based on multi-class AUROC.
- Pick top-ranked *k* features.
- Use LDA with (only) these features.



LDA using all features

Feature Selection: Wrapper Methods

- Compare model performances achieved with different subsets of the available features.
  - Several strategies are available.
- Usually involves a **hyperparameter** that tells the algorithm "where to stop."
- Advantage:
  - Although an **exhaustive search** among all possible feature subsets is usually not possible, wrapper methods may achieve a good trade-off between model complexity and accuracy.
- Disadvantage:
  - Computationally expensive. Often a huge number of models must be fitted.
  - Tuning of hyperparameter increases computational cost.

Feature Selection: Wrapper Methods
Best Subset Selection

- Check the test-set performances of all possible models that use a subset or possibly all of the available *p* predictors.
  - Often used with a pre-specified number of features to be selected.
  - There are $\binom{p}{k}$ possible models with exactly *k* out of *p* features, and 2^p possible models overall.
  - E.g. 40 available features: >847 million possible models using exactly 10 features, or >10^{12} possible models overall.

- **Exhaustive search** would be prohibitively slow in most relevant situations.

Feature Selection: Wrapper Methods
Stepwise (Forward) Selection

- Start with the ‘empty model’ (no features), and keep adding features, one at a time, as long as the test-set performance (or some penalized measure of goodness-of-fit such as AIC) improves.
- Can also be applied to groups of features.
  - E.g. stepwise selection of image date, i.e. add all features that belong to the added image date (Peña & Brenning, 2015 in *RSE*)
- Checks “only” up to $n(n+1)/2$ candidate models out of the 2^p possible models.
  - E.g. for p = 40 only an unimaginably small fraction of all possible feature sets
- Limiting model size (i.e. number of steps) reduces computational cost, but introduces a new hyperparameter.
- Simple and pragmatic method, but better methods are available…

Feature Selection: Wrapper Methods
Other Search Strategies / Optimization Techniques

- Since best subset selection is computationally untractable and stepwise selection insufficient, several strategies have been developed to obtain a nearly optimal feature set with a high probability, using a limited amount of computing time.

- General-purpose combinatorial optimization techniques can be used, e.g. **genetic algorithms**.

Shrinkage Methods

- Shrinkage methods modify mathematical models such as LDA and linear regression (but also SVM) in order to obtain simpler, more parsimonious models.
- They have in common that model coefficients tend to be pushed closer to zero, i.e. they are shrunk.
- In penalized regression, ordinary least squares is modified to include a term that measures the size of the coefficient vector.
  - Slightly larger residuals will be accepted if this reduces the size of the coefficients.
- Depending on the criterion used, coefficients can effectively be shrunk to zero.
- This is the case for the **lasso penalty** in LDA and linear regression.
  - This eliminates a feature from the model → the lasso is a subset selection method!
  - A hyperparameter $\lambda \geq 0$ controls the weight of the penalty.
  - For $\lambda = 0$, standard LDA or MLR is obtained; for $\lambda \rightarrow \infty$, only the intercept is modelled.

Shrinkage Methods

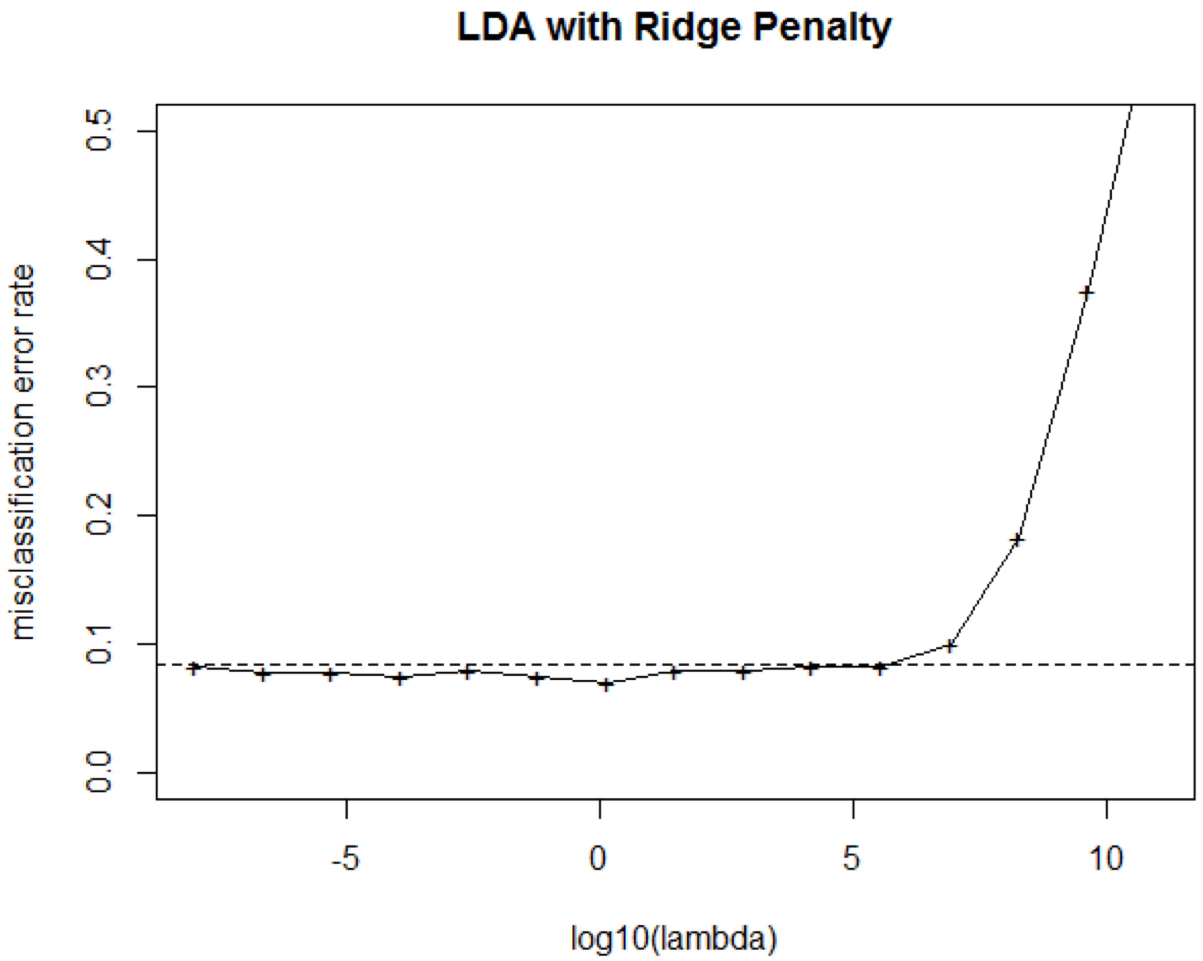
Other available penalties include:

- **Ridge penalty**
  - Coefficients will not normally be shrunk to zero, and therefore all features remain in the model.
  - This may be better at suppressing noise in the features than lasso.
  - I'd expect ridge to be more promising than lasso in hyperspectral remote sensing applications due to strong correlations between spectral bands.
- **Elastic net**
  - Uses an additional hyperparameter to blend the lasso and ridge penalties.
  - Only available in regression analysis, not in LDA yet (to my knowledge).

Shrinkage Methods: LDA with Ridge Penalty

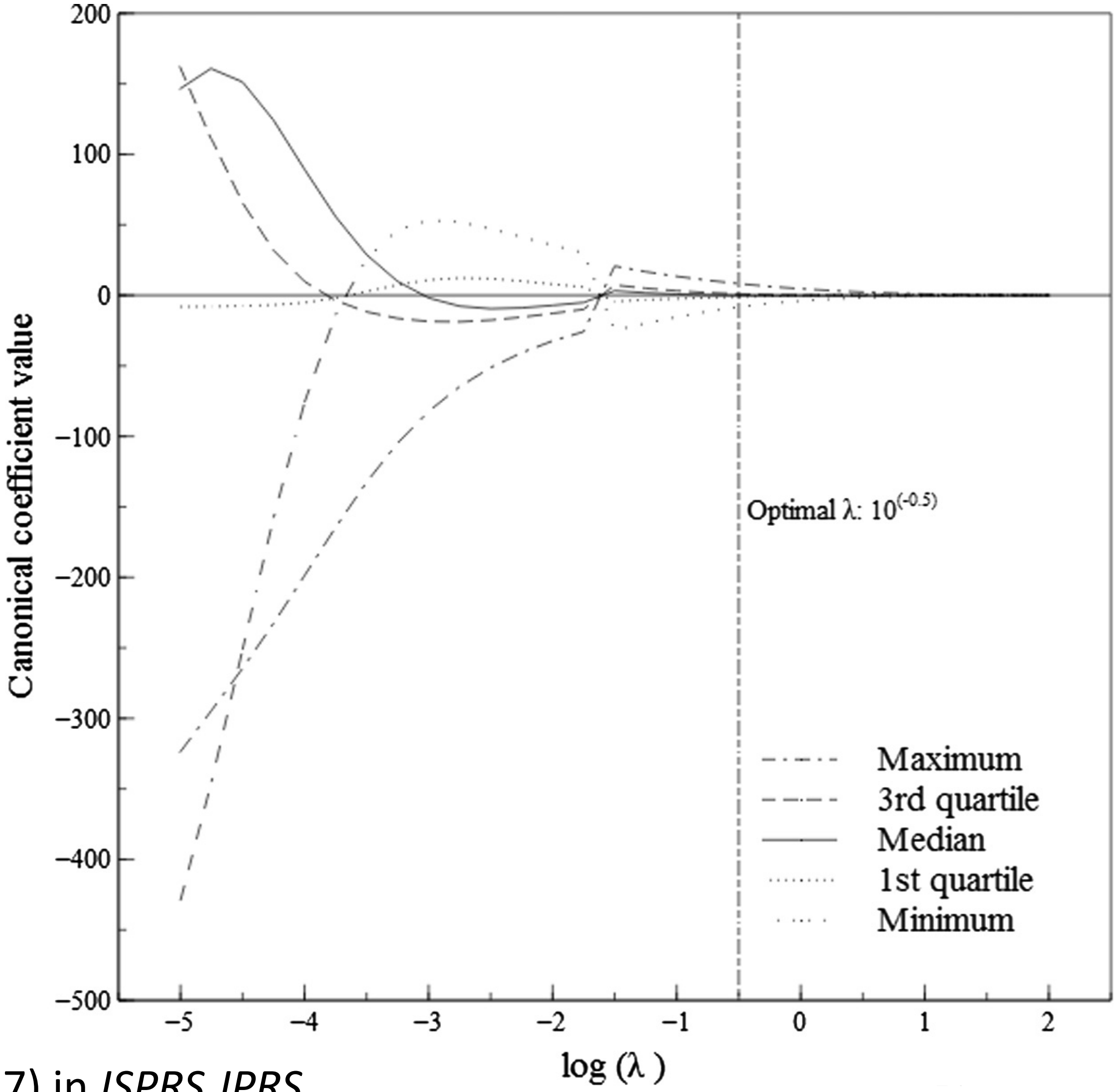
Classroom example:

- LDA with ridge penalty as implemented in the ‘mda’ package, function ‘fda’ with method = gen.ridge



Shrinkage Methods: LDA with Ridge Penalty

- Several thousand spectro-temporal Landsat features were used in the study by Peña et al. (2017).
- LDA with ridge penalty performed much better than lasso penalty.
- It also performed better than using a smaller feature set and plain LDA.
- Coefficient estimates were highly variable when a small penalty were used.



Peña et al. (2017) in *ISPRS JPRS*

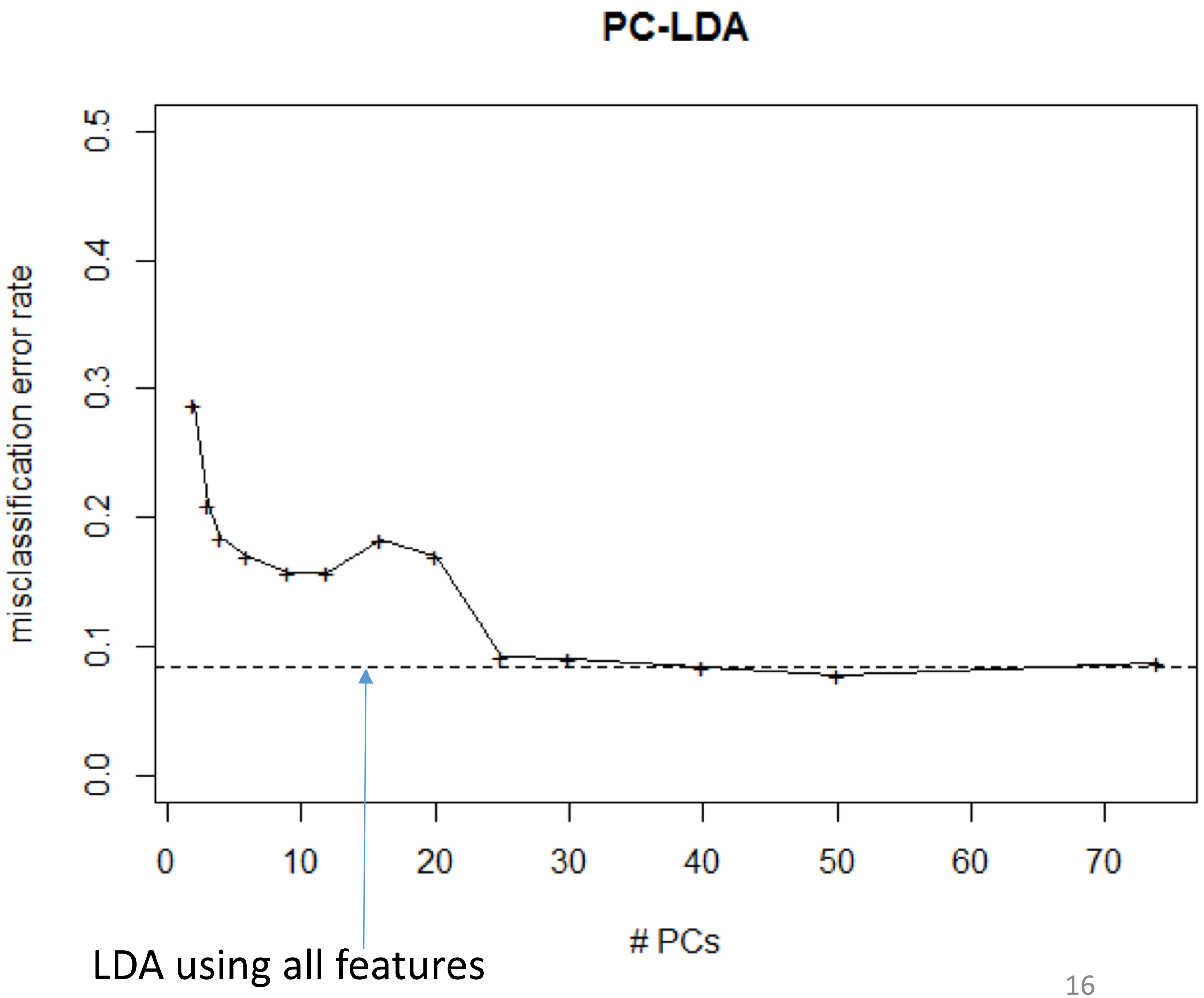
Dimension Reduction

- Subset selection reduces the dimensionality of the features space – but other approaches are also available.
- Most well-known: **principal component analysis** (**PCA**)
- PCA linearly re-combines the features into new variables, called "principal components" (PC).
  - The first PC represents the largest amount of variance.
  - The second PC is orthogonal on the first on and represent the second largest amount of variance. Etc.
- The first *k* PCs are then used as a new feature set.
  - Although the model will then only have *k* predictors, technically all available features went into calculating the *k* PCs.
- This can be combined with any classification or regression model. But it makes more sense in combination with linear models (LDA, GLM).

Dimension Reduction: PC-LDA

Classroom example:

- Calculate principle components (PCs) from feature set
- Rank the PCs based on the explained variance.
- Pick the top-ranked *k* PCs.
- Fit an LDA using only these PCs as features.



What Have We Learned

- There are many approaches that allow us to reduce model complexity through feature selection, shrinkage, or dimension reduction.
- It can be difficult to tell in advance which (if any) of these methods is most appropriate in a particular application.
  - In some cases, feature selection makes our analysis more complex and computationally more expensive. The choices we make in choosing a feature selection technique may seem arbitrary...
- Ridge penalties seem very promising to me, and very efficient implementations of ridge LDA and MLR are available in R.
- Penalized models can be faster and perform better, but additional hyperparameters may need to be tuned.
- Make sure that (an outer) cross-validation is properly performed, or a separate hold-out set is used only for testing the model *after* tuning any parameter.