Advanced Statistical Modelling: Handling High Dimensional Data

Nikolas Krstic
Applied Statistics and Data Science Group (ASDa)
Department of Statistics, UBC

July 31, 2024

Outline

- Intro to high dimensional problems
- Principal Component Analysis (PCA) and Multidimensional Scaling (MDS)
- Feature/Variable Selection
- Ridge Regression
- LASSO
- Group LASSO

Introduction to High Dimensional Problems

Typically, datasets have two dimensions:

- Number of Rows/Observations (denoted with n)
- Number of Columns/Variables (denoted with p)

Datasets with more dimensions can usually be transformed into ones with two dimensions.

Example: Collecting longitudinal data of participants at different timepoints, third dimension is timepoint, can simply aggregate data together and include another variable (column) to indicate observation's timepoint.

High dimensional problems specifically refer to large p or even $p > n$. Can be difficult to address when modelling.

Why are High Dimensional Problems "Problematic"? Part I

May have heard this in Machine Learning referred to as the "Curse of Dimensionality".

MAIN IDEA: The more variables/features there are (i.e. large p) compared to the sample size (n), the harder it is for the model to **generalize** to new unseen data.

Consider basic example of binary predictors and a continuous outcome:

- To have an observation for every possible combination of predictor value, we need at least 2^p observations.
- $p = 20$ means we need more than 1 million observations
- Considering each combination of predictor values could have varying responses, means we generally need even more data than just one per combination.

Why are High Dimensional Problems "Problematic"? Part II

In practice, we don't necessarily need data for every single combination, because some variable value combinations are rare or even non-existant in our population.

However, the previous example still gives some perspective that we generally need plenty of data (large n, that is) to account for a large number of variables/features.

Multiple observations with the same or similar combination of values can help us characterize the uncertainty/variability in the outcome.

- For example with a binary outcome, one combination of values might result in an outcome of **0** 80% of the time and an outcome of **1** 20% of the time in our original population.

Why are High Dimensional Problems "Problematic"? Part III

Connection with Overfitting - The increased model complexity due to many additional predictors in our model means we begin to overfit on our data, worsening generalization.

Very, very rough general rule of thumb for linear regression models is that you have at least 10 observations for each term in your model:

- Ultimately depends on your problem and your objective (inference, prediction, etc.).
- Might not be good enough for large datasets (large p and large n), since we can see in the previous example that the problem can become exacerbated exponentially.

Why are High Dimensional Problems "Problematic"? Part IV

Relationship with the signal to noise ratio in data:

- Signal refers to the meaningful part/pattern of the data (contributes to predicting the outcome).
- Noise refers to the meaningless part of the data (hinders predicting the outcome, the "background noise", irreducible error).
- Can have important predictors that help capture "signal", but can also have predictors that could either be unhelpful or even hinder the quality of the model.

Although incorporating more variables in our model can help capture more of the "signal", in some cases many of the predictors could be "noisy" or have minimal signal (i.e. relationship with the response).

Why are High Dimensional Problems "Problematic"? Part V

Some models just actually won't fit if there is $p > n$:

- Linear regression models can't be fit because:
  - X has rank of at most n
  - $\Rightarrow X$ columns are not all linearly independent
  - $\Rightarrow X^T X$ has determinant of 0
  - $\Rightarrow X^T X$ is singular, and therefore non-invertible
  - $\Rightarrow$ Can't solve $\hat{\beta} = (X^T X)^{-1} X^T y$
- K-nearest neighbours (kNN) suffers tremendously and its primary assumption (that observations in the same neighbourhood are similar to each other) essentially "breaks down" when p is large compared to n.
  - The reason for this breakdown is that observations are now quite distant from one another on average (so a test point's k nearest neighbours are not actually that near at all).

Multicollinearity Impacts

Even if p is just very large (i.e. $p < n$), another problem that can still surface is multicollinearity (i.e. high correlation of columns).

Multicollinearity - Variables or combinations of variables are highly correlated/linearly dependent. Severe multicollinearity can lead to the same problems as $p > n$.

Linear Regression: Can somewhat safely "disregard" this problem if focus is solely on prediction, but for inference this can be very problematic because standard errors of coefficients inflate (i.e. significant loss of statistical power).

Random Forests: Generally robust, but if many of the variables are highly correlated, there is "dilution" when randomly selecting predictors for splits in the tree.

- Extreme Example: If 100 predictors are highly correlated and 1 predictor is "unique" and informative, can be a problem because that last predictor will rarely get randomly selected.

Dimension Reduction methods

There are a few methods that aim to reduce the dimension of datasets:

- Help maximize "information" available in the dataset to only a few dimensions.
- Can help for visualization of the data when selecting to reduce to 2 or 3 dimensions (e.g. examine separability of points for classification).

Two popular methods include:

- Principal Components Analysis (PCA)
- Multidimensional Scaling (MDS)

MDS is typically used for data visualization while PCA is often used for both data visualization and modelling (principal components "regression").

PCA Method

Objective of PCA is to transform the original data into a collection of "principal components", which sequentially maximally capture the variation available in the data.

- First principal component (PC) "summarizes" the most variance present in the dataset.
- Second PC "summarizes" the most **leftover** variance.
- Third PC "summarizes" the most variance **still leftover**.

Given the goal, we need to first standardize all of the variables. Otherwise, the results of PCA will be dependent on the variable scales (therefore inconsistent).

PCA ultimately produces a maximum of $\min(n, p)$ PCs (any additional components, in the case of $p > n$, essentially have 0 variance).

First Principal Component

The first PC (z_1) is a linear combination of all p variables that "captures" the most variance:

- Need to find the weight vector of the linear combination that does this by computing:

$$w_1 = \underset{w_1}{\text{argmax}} \left\{ \frac{{w_1}^T X^T X w_1}{{w_1}^T w_1} \right\}$$

- Solution to this is fairly simple, it is the eigenvector of $X^T X$ corresponding to its largest eigenvalue.
- Can now obtain the first PC by simply computing:

$$z_1 = X w_1$$

Further Principal Components Part I

The second PC (z_1) is a linear combination of all p variables that "captures" the most "leftover" variance:

- Need to first remove from the data what has already been explained/captured by the first PC:

$$X_{(2)} = X - X w_1 w_1^T = X - z_1 w_1^T$$

- Proceed with the same as before, find the weight vector of the desired linear combination:

$$w_2 = \underset{w_2}{\text{argmax}} \left\{ \frac{w_2^T X_{(2)}^T X_{(2)} w_2}{w_2^T w_2} \right\}$$

Further Principal Components Part II

- Solution to the above is also simple, it is the eigenvector of $X^T X$ corresponding to its second largest eigenvalue.
- Can now obtain the second PC by simply computing:

$$z_2 = X w_2$$

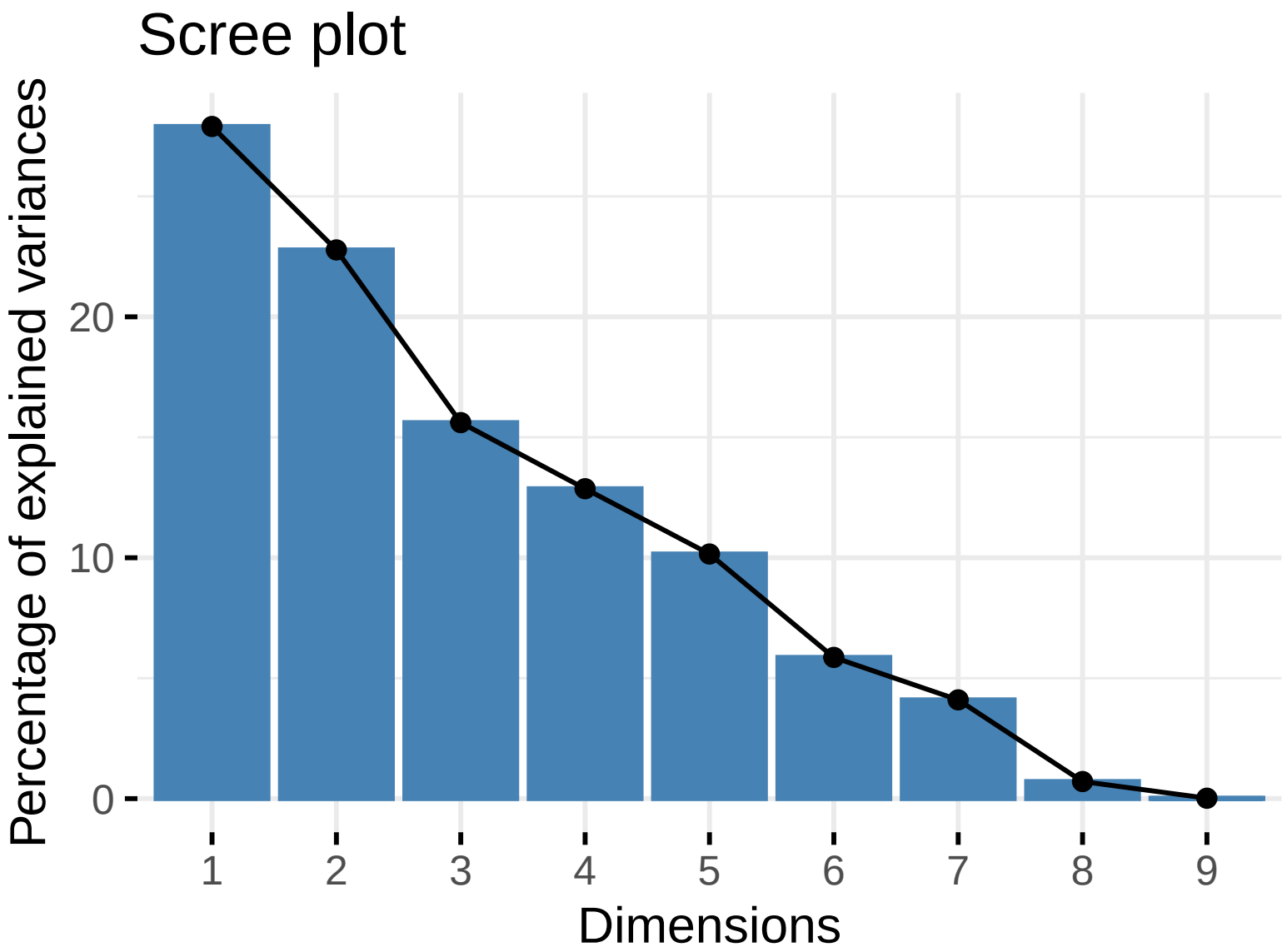
Repeat the exact same process for all other PCs. For third PC, remove variance explained by second PC:

$$X_{(3)} = X_{(2)} - X_{(2)} w_2 w_2^T = X_{(2)} - z_2 w_2^T$$

and repeat above steps using $X_{(3)}$ now.

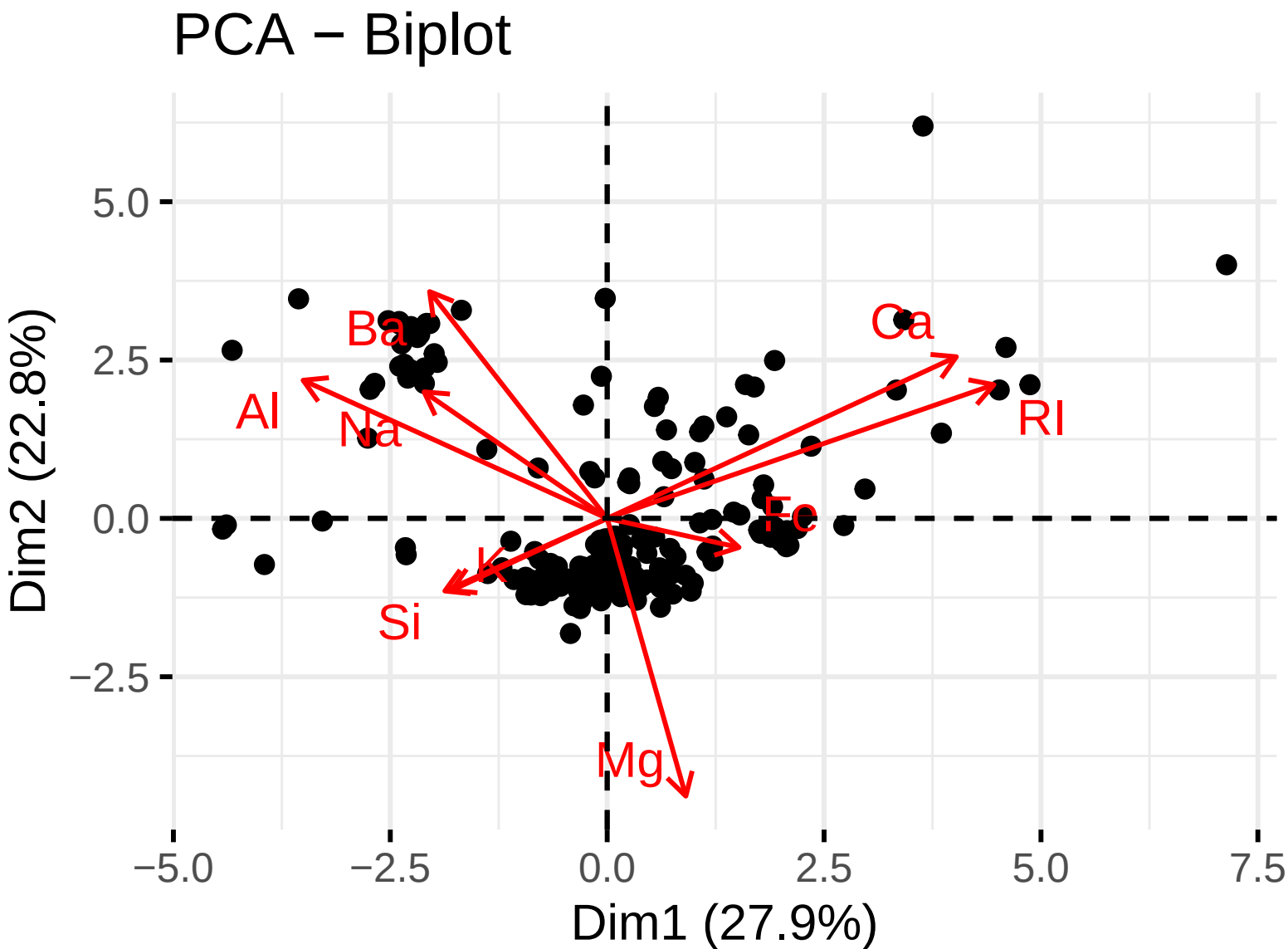
PCA Scree Plot

Useful plot to understand the amount/proportion of variance explained by each of the first few principal components.



PCA Biplot

Can examine the relationship between the first two PCs and the original variables' contributions.



MDS Method

Objective of MDS differs from PCA in that the goal is simply to preserve the "distances" or "dissimilarity" (as best as possible) between observations when reducing their dimensionality.

- Compute a distance/dissimilarity matrix D with entries d_{ij}, quantifying how different observations are:
  - Euclidean distances ($d_{ij} = ||x_i - x_j||$)
  - Custom distance measure (useful for mixed data)
- Select the dimension k you would like to reduce the data to
- Generate new observations z_i ($i = \{1, ..., n\}$) in the k-dimensional space that mimics the distances between original observations x_i as much as possible (i.e. loss function)

Example loss function to use for MDS:

$$\sum_{i \neq j} (d_{ij} - ||z_i - z_j||)^2$$

Final Comments on Direct Dimension Reduction Methods

PCA only works on data with continuous predictors, while MDS could potentially be used on mixed data as well (requires specifying what the "distance" is for categories, which could be unclear).

PCA operates under the assumption of linear transformations (namely, that there are linear relationships between the variables), though there have been extensions (non-linear PCA).

These methods significantly impair interpretability of the final variables used in modelling:

- PCA needs to standardize variables before use.
- Resulting data from MDS is very uninterpretable (given method's nature).
- Generally not recommended to use PCA when doing inference, but might salvage some interpretability by understanding contribution of variables to PCs.

Direct Variable Selection Approaches

Another option is to select variables to use in the model (as opposed to aggregating information together like with the approaches above).

Advantages:

- Can preserve interpretability of variables.
- Limit data that needs to be collected in the future (e.g., only need to collect half of the variables instead of all).
- Eliminate variables that are poor predictors of the response.
- Simple to understand.

Disadvantages:

- Can be computationally intensive (depending on method)
- Post-Selection Inference problem
- Complex predictive performance assessment pipelines
- Risks of overfitting

Subset Selection

Objective is to find a subset of variables that "best" models the outcome.

How to figure out what subset is best choice?

- Need to figure out how to find this subset (search method)
- Need to figure out how to assess quality of fit using subset (criterion)

Search methods include forward selection, backward selection, best subset, etc.

Criteria include measures such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Mallows's C_p, etc.

Forward Selection

Let's consider the linear regression model context, where we want to select predictors for our model.

Forward Selection Algorithm:

1. Start with the null model containing no predictors
2. Add one of the available predictors to the model and compute the criterion (e.g., AIC)
3. Remove the selected predictor
4. Repeat Steps 2 and 3 for every unique predictor
5. Review the criterions computed and select the predictor that best optimizes the criterion
6. Add the selected predictor to the model and no repeat Steps 2-5 again using this new model as the "baseline" model
7. Stopping rule is when the addition of any new predictors does not improve the criterion any further.

Backward and Best Subset Selection

Backward Selection algorithm is essentially identical to forward selection, except that we start with the full model (put all predictors in the model) and work our way backwards (removing predictors one at a time).

Best Subset Selection involves trying every single possible combination of variables in the model and selecting the best subset according to the criterion used.

Stepwise Selection is essentially a combination of Forward and Backward selection:

- Start like Forward Selection, using null model.
- Try adding or removing a predictor from the current model, see which option best improves the criterion.
- Repeat until criterion is not improved.

Disadvantages to These Search Methods

Forward Selection is potentially overly simple (only really trying a few combinations, once a predictor is selected it's always in the model).

Backward Selection is problematic when there are many predictors (model can't even properly fit, like $p > n$).

Best Subset Selection is computationally intensive and increases risk of overfitting.

Stepwise Selection is a good compromise of all above three, but still has some of the general limitations as previously listed (overfitting risk, post-selection inference, etc.).

AIC

AIC is often used because this criterion balances the quality of the model fit with the number of predictors in the model:

- $\text{AIC} = 2(k - \ln(\hat{L}))$
  - k is the number of model parameters.
  - $\hat{L}$ is the maximum likelihood function of the model.
- Objective is to minimize AIC as much as possible.
- AIC can be used on a variety of statistical models.
- Nature of AIC somewhat helps address the risk of overfitting when using above search methods.

When considering linear regression models, AIC is asymptotically the same as conducting leave-one-out cross-validation to select the model.

BIC and Mallows's C_p

BIC behaves very similarly to AIC, except instead of using $2k$ to penalize overfitting it uses $\ln(n)k$ instead.

Given this extra penalization (reliance on n) and since BIC is originally derived as an asymptotic approximation $\Rightarrow$ BIC is not a good measure to use for high-dimensional modelling.

- Very conservative, so only works well for large enough n compared to k

Mallows's C_p is equivalent to AIC for linear regression models, but can't be used on high dimensional problems (requires estimation of residuals variance on "complete" model).

Regularized Modelling Methods

So the problem with "search method" type variable selection is that there remains a risk of overfitting in high dimensional case (even with AIC or even extended versions of AIC).

Search methods also involve significant computation time for complex models (need to build a model for every... single... predictor combination of interest).

One different solution is to implement "regularization" terms to our method's objective function, to help alleviate high-dimensional problem.

Idea is often to penalize the magnitudes of the model parameters to avoid overfitting and to also potentially allow the modelling method **itself** to select the predictors.

Regularization - Bias-Variance Tradeoff

Why does regularization help?

- Regularization penalizes parameter sizes, and thus introduce bias to the parameters
- $\Rightarrow$ Even though the parameters are biased, they have less variance (i.e. are not super sensitive to the contents of the training dataset).
- $\Rightarrow$ The overall combination of bias+variance in our prediction errors are generally lower on new and unobserved data
- $\Rightarrow$ Introducing regularization helps optimize bias-variance tradeoff, and thus avoid both underfitting and overfitting.

TECHNICALLY, AIC search method is a type of regularization, but doesn't introduce bias (predictor is either "selected and unbiased" OR "not selected").

Regularized Linear Regression Model

Linear regression has least squares objective:

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \, ||Y - X\beta||_2^2$$

Want to introduce a new penalty term $R(\beta)$ to perform regularization:

$$\hat{\beta} = \underset{\beta}{\text{argmin}} \left\{ ||Y - X\beta||_2^2 + R(\beta) \right\}$$

Have a lot of flexibility on what to choose for the penalty term, and usually we want to choose one that reflects some information/expectations we have about the current problem.

Typically $R(\beta)$ is some norm function.

Ridge Regression (L_2-Regularization)

One type of penalty is $R(\beta) = \lambda||\beta||_2^2 = \lambda(\beta_1^2 + \beta_2^2 + ... + \beta_p^2)$.
Called L_2-Regularization since it uses the L_2-norm.

Note that λ here is a hyperparameter (Lagrangian multiplier), controlling how intensely we are penalizing/regularizing the coefficient magnitudes.

- Large $\lambda \Rightarrow$ more penalization $\Rightarrow$ more biased coefficients
- Each λ corresponds to a constant c such that $\beta_1^2 + \beta_2^2 + ... + \beta_p^2 \leq c$
- Can be selected for the final model by selecting from a grid of values, often through cross-validation (select one corresponding to minimum RMSE).

Ridge regression developed to address multicollinearity/high dimensional problems in the linear regression context.

Properties of Ridge Regression

Note that the intercept is not included in the penalty (a bit nonsensical to do since it simply dictates the "starting point" of our prediction).

When $\lambda = 0$, then none of the coefficients are penalized (we get the least squares coefficient solution).

When λ is extremely large, the coefficients approach zero rapidly but are very unlikely to be zero exactly.

Need to standardize the predictors before use because we need to ensure that the variables are "fairly" penalized.

- Example: If one predictor is measured in the 1000s and another predictor is measured in the 1s, the second predictor will more likely have a larger coefficient and thus will get penalized more.

Advantages and Disadvantages of Using Ridge Regression

Advantages:

- Addresses multicollinearity issues.
- Addresses high dimensionality problems, thus preventing overfitting.
- Selecting λ is fairly straightforward, just use cross-validation (still need separate test or nested cross-validation for performance assessment).
- Can be generalized to many different types of regression models (logistic, etc.)

Disadvantages:

- Can't really do statistical inference on coefficients (no standard errors because coefficients are biased, so underlying distribution is unclear)
- No variable selection (model contains ALL variables, almost always)
- Can occasionally be computationally intensive

LASSO (L_1-Regularization)

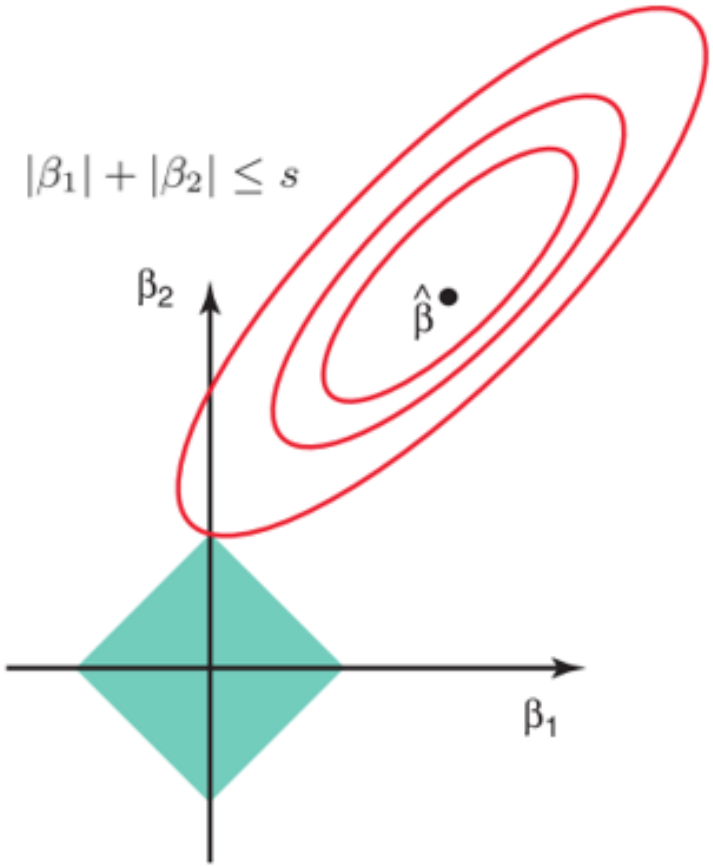
Least Absolute Shrinkage and Selection Operator (LASSO), uses the penalty $R(\beta) = \lambda||\beta||_1 = \lambda(|\beta_1| + |\beta_2| + ... + |\beta_{\boldsymbol{p}}|)$. Called L_1-Regularization since it uses the L_1-norm.

Extremely popular regularized regression method because it can achieve sparsity (i.e. variable selection), by setting coefficients to zero when penalization is high.

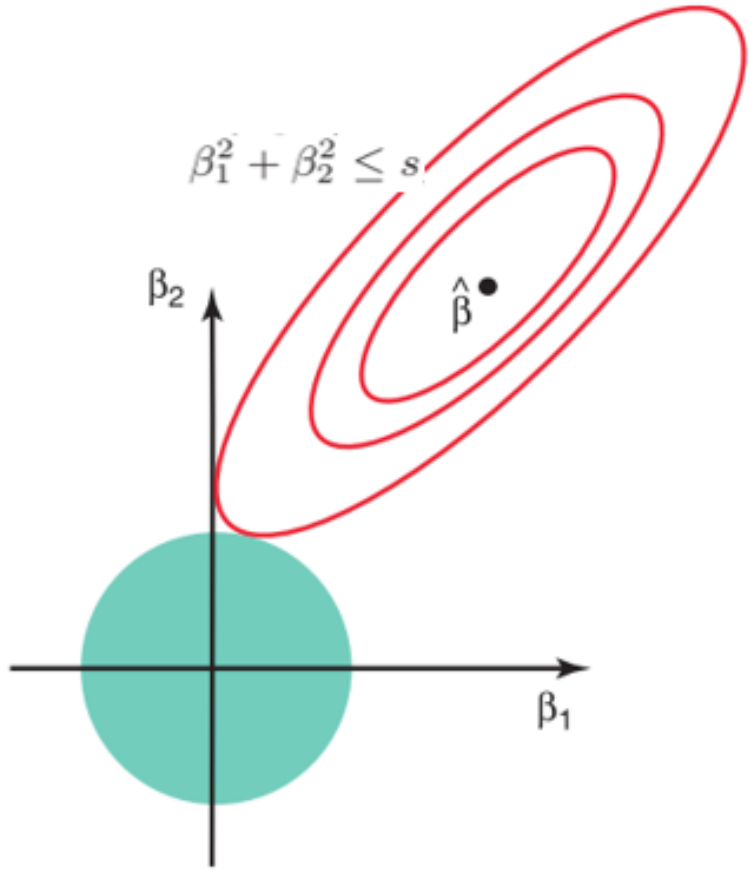
Thus, important predictors are kept while poor predictors are eliminated from the model.

LASSO relies on sparsity assumption, that some coefficients are truly zero (which often makes sense, given some predictors might be unrelated to the response).

Why Does LASSO Offer Sparsity?



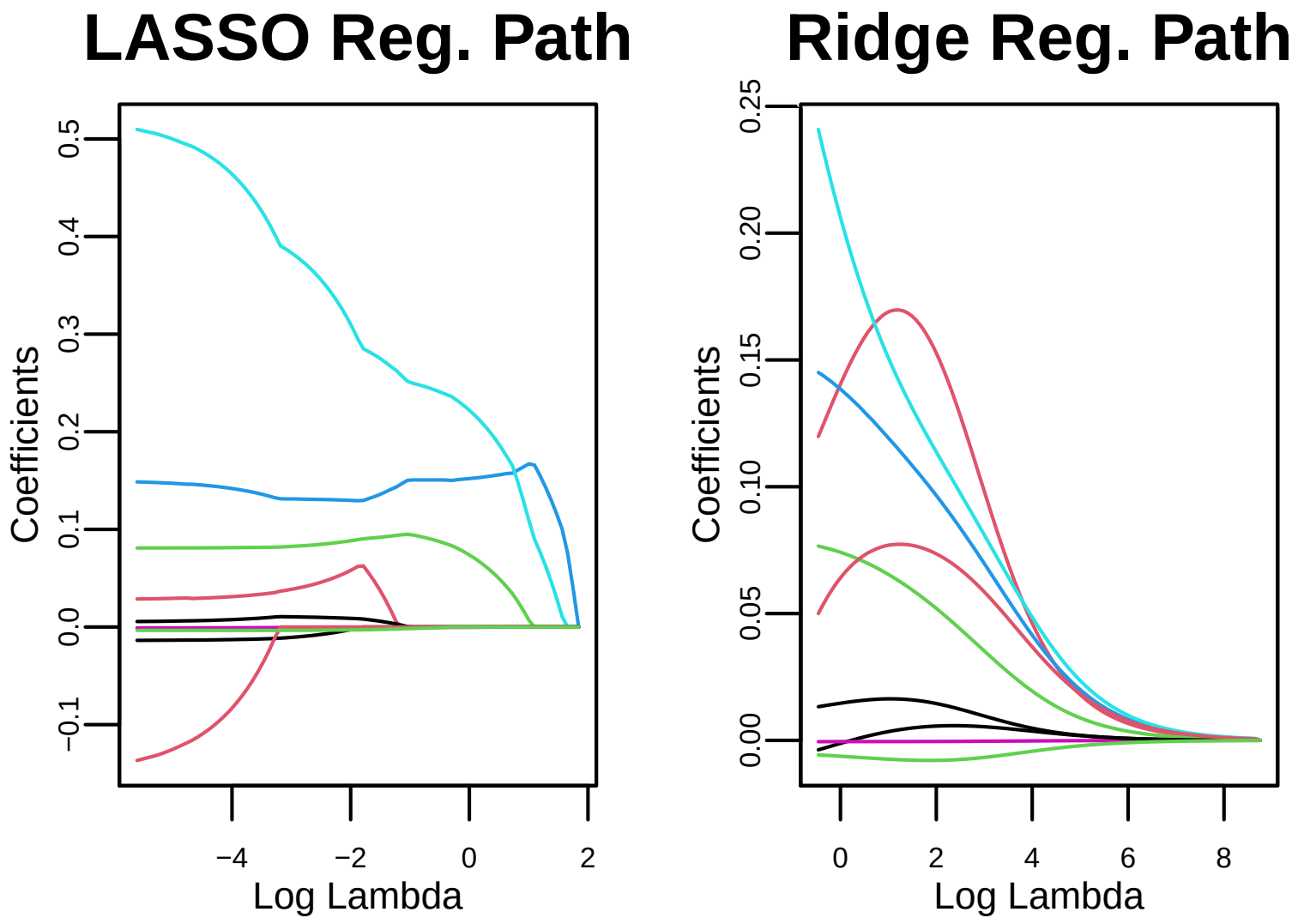
Lasso Regression



Ridge Regression

[0] https://www.datacamp.com/tutorial/tutorial-lasso-ridge-regression

Regularization Paths



Properties of LASSO

When $p > n$, then the solution of LASSO is actually not unique (infinite valid solutions) AND the number of non-zero coefficients is maximum n.

If there is a subset of highly correlated predictors, only a small number (often one) of them will ultimately be selected.

- Implies those predictors are unimportant when they could have similar importance

As λ increases, the coefficients reduce in size ("shrinkage") until they individually hit some "threshold", under which they are immediately set to zero

- Solution of LASSO is often referred to as "soft-thresholding" the ordinary least squares solution.

There is a finite λ for which $\beta = 0$ (all coefficients are zero).

Elastic Net

Can overcome problem of LASSO selecting only a few correlated predictors or only *n* predictors by using Elastic Net.

Think of it as combination of both Ridge and LASSO at the same time (we use both $||\beta||_1$ and $||\beta||_2^2$).

Overall penalty looks like: $\lambda((1-\alpha)||\beta||_2^2 + \alpha||\beta||_1)$

Basic extension, but we have an additional hyperparameter α that needs to be selected as well.

Can recover Ridge or LASSO simply by setting $\alpha = 0$ or $\alpha = 1$, respectively.

General Comments

Choice of method depends on the problem at hand, but often LASSO or Elastic Net are good choices.

LASSO/Elastic Net are related to Support Vector Machines.

Given the nature of LASSO's penalty, there have been many extensions for different circumstances:

- Categorical Data
- Grouped Data
- Model Interactions

Mixed Data Scenario

What happens if we also have categorical/grouped data in our dataset?

We typically use one-hot encoding with categorical predictors (one category acts as reference, create a bunch of binary predictors for each of the other levels).

Technically what LASSO would do is regularize each of the coefficient levels toward 0 independently.

Doesn't really make sense to do use LASSO, we want to determine whether we are going to select the categorical predictor itself or not.

Categorical Predictor Example

Human-Readable

Pet
Cat
Dog
Turtle
Fish
Cat

Machine-Readable

Cat	Dog	Turtle	Fish
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

[0] https://medium.com/analytics-vidhya/stop-one-hot-encoding-your-categorical-variables-bbb0fba89809

Group LASSO

If we want to regularize groups of predictors collectively, then we can use Group LASSO instead.

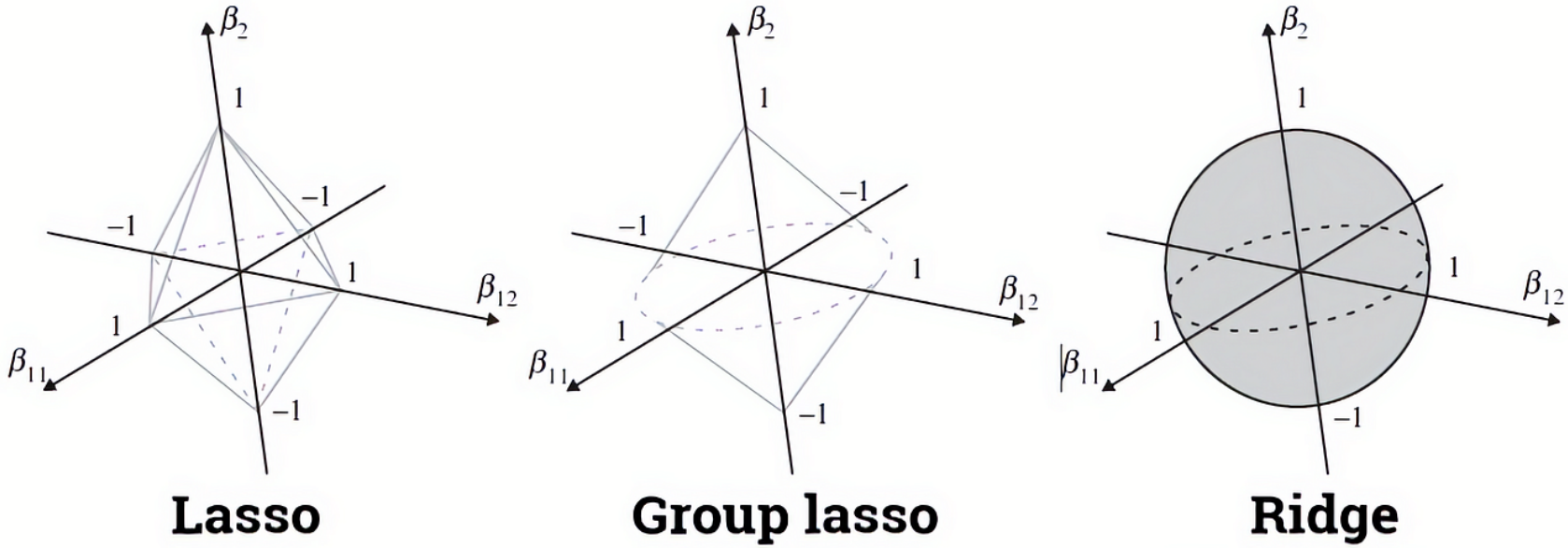
Penalty is of the form $||\beta^{G_i}||_2$, where β^{G_i} represents the coefficients for group $i = \{1, ..., k\}$.

Thus, there are multiple penalty terms, one for each group.

With high penalization, $\beta^{G_i} = 0$ (either all group coefficients are set to 0, or none are).

Can think of Group LASSO as doing Ridge "within-group" and LASSO "between-group"

Group LASSO Penalty Behaviour



[0]https://towardsdatascience.com/sparse-group-lasso-in-python-255e379ab892

Group LASSO Properties and Assumptions

Since there are multiple penalty terms (one for each group), then each term can receive a unique weight w_i:

$$\lambda \sum_{i=1}^{k} w_i ||\beta^{G_i}||_2$$

Common choice of weighting scheme is to set $w_i = \sqrt{|G_i|}$.

STRONG ASSUMPTION: To apply Group LASSO, we need that the group matrices X_{G_i} are orthonormal ($X_{G_i}^T X_{G_i} = I$).

Need to either conduct "sphering" transformation on the data first or use penalty $\lambda \sum_{i=1}^{k} w_i ||X^{G_i} \beta^{G_i}||_2$ instead (Simon and Tibshirani, 2011).

Further Relevant Methods/Estimators

Factor Analysis:

- Similar to PCA, but objective and methodologies differ.
- Goal is to recover "latent" variables (unobserved variables that actually explain the variation in the vast array of observed variables).
- Exploratory and Confirmatory Types of Analysis

James-Stein Estimator (James and Stein, 1992):

- Have a single multivariate Normal/Gaussian observation ($n = 1$, $p > 1$), how to best estimate the mean vector.
- Shocking conclusion that their **shrinkage** estimator is at least as good or always better than the least-squares estimator (i.e. sample mean) in terms of mean squared-error (for $p > 2$).
- Part of motivation for above regularized regression methods.

References

1. James, W., & Stein, C. (1992). Estimation with quadratic loss. In Breakthroughs in statistics: Foundations and basic theory (pp. 443-460). New York, NY: Springer New York.

2. Simon, N., & Tibshirani, R. (2012). Standardization and the group lasso penalty. Statistica Sinica, 22(3), 983.

3. Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society Series B: Statistical Methodology, 68(1), 49-67.