GENERAL ASSEMBLY

DATA SCIENCE

SYD DAT 6

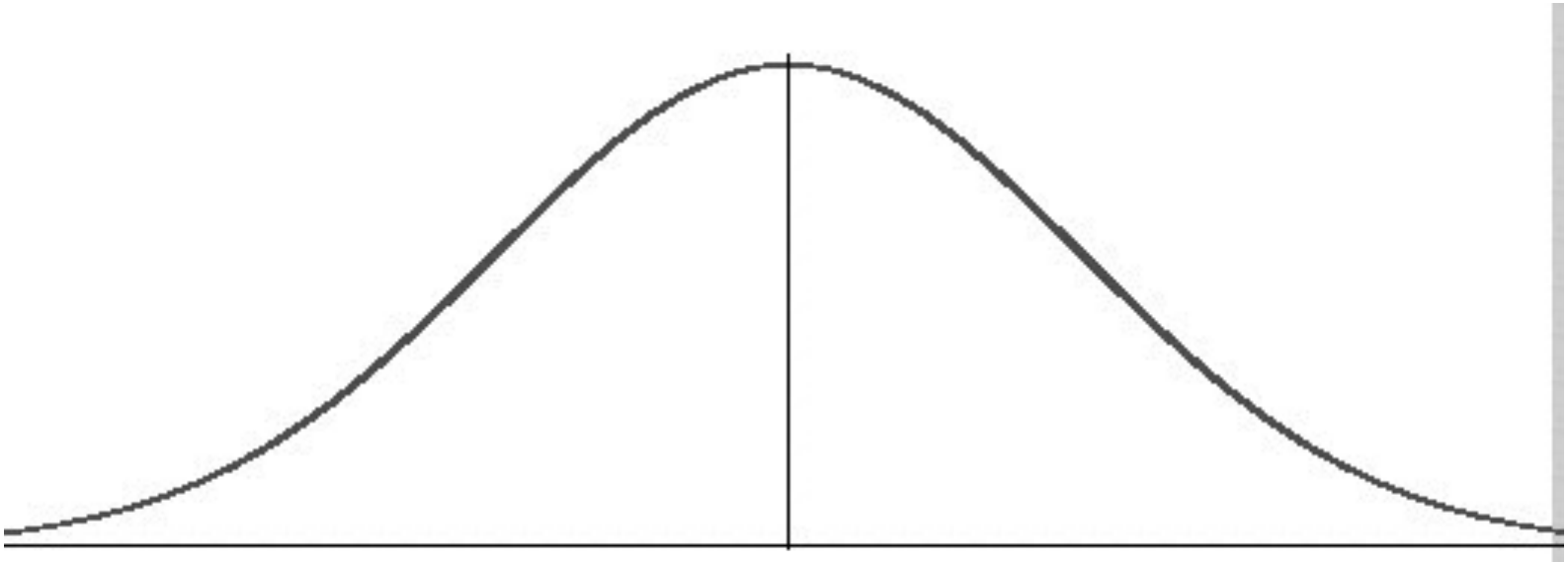
Week 4 – Regularisation & Dimensionality Reduction
Monday 31st October

AGENDA

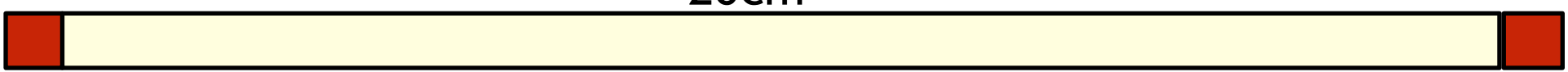
1. Review of Model Selection
2. What does dimensionality mean
3. What is Regularization?
4. Why use Regularization
5. Lab 1
6. What is Dimensionality Reduction?
7. Why use Dimensionality Reduction
8. Lab 2
9. Discussion

DATA SCIENCE PART TIME COURSE

DIMENSIONALITY

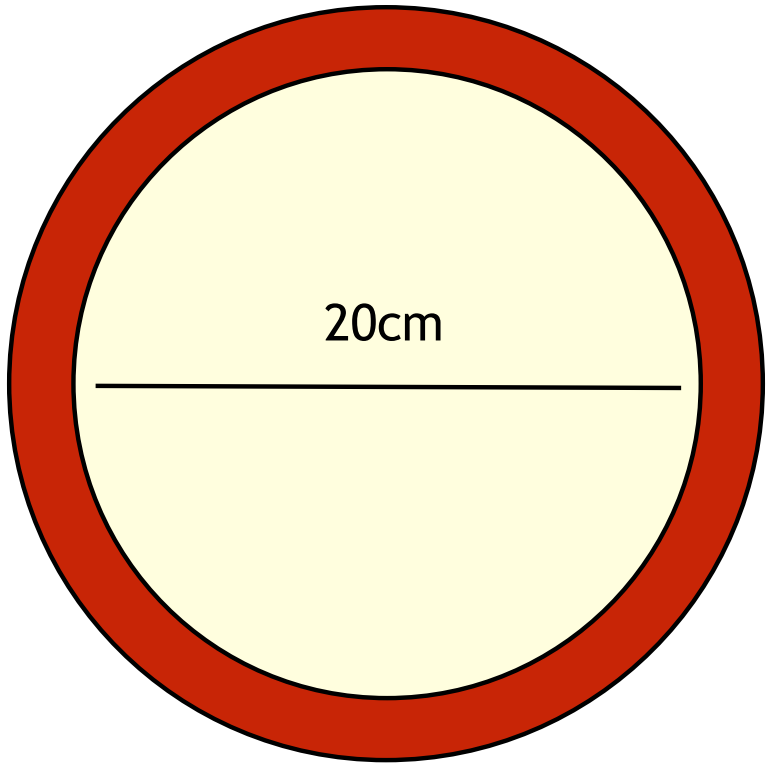


20cm



Discard **Keep** **Discard**

In 1 dimension we keep 20/22 ≈ 0.91% of the data



In 2 dimensions we keep (10^2) ∏ / (11^2)∏ ≈ 82% of the data

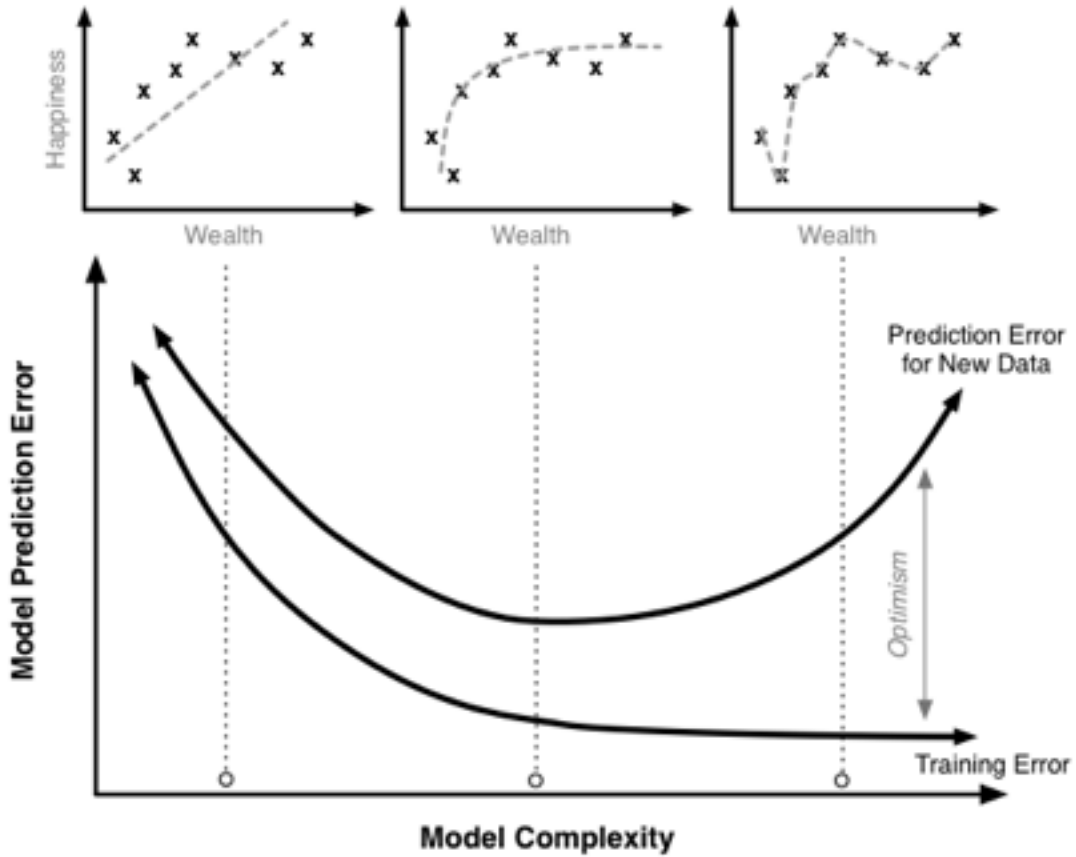
DATA SCIENCE PART TIME COURSE

FEATURE SELECTION

Prediction Accuracy: especially when $p > n$, to control the variance.

Model Interpretability: By removing irrelevant features — that is, by setting the corresponding coefficient estimates to zero — we can obtain a model that is more easily interpreted. We will present some approaches for automatically performing feature selection.

BIAS – VARIANCE TRADEOFF



- Subset Selection
- Shrinkage
- Dimension Reduction

We could fit a separate linear regression model for every combination of our features.

But what happens when we have a large number of features?

Computation time becomes a factor and we also need to consider that as we include more features we are increasing the chance we include a variable that doesn't add any predictive power for future data.

SIMPLE

1. Select a subset of the available features
2. Build a model with those features
3. Evaluate the model by your chosen metric and use cross validation
4. Repeat for each combination of features

Problematic with a large number of predictors

- Computationally expensive
- The model overfit problem creeps in as

FORWARD STEPWISE

- Same process as subset selection, but adds a predictor to a model, one-at-a-time, until all of the predictors are in a model
- Then, for each predictor that made the greatest difference to a model, that features is added to the proper model

BACKWARDS STEPWISE

- Same as Forwards, except removing one feature at a time
- Disadvantage in that it requires that there be more records than features.

DATA SCIENCE PART TIME COURSE

REGULARISATION

Shrinkage

We fit a model involving all p predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as regularization) has the effect of reducing variance and can also perform variable selection.

- A tuning parameter lambda (or sometimes alpha) imposes a penalty on the size of coefficients.

- Instead of minimizing the "loss function" (mean squared error), it minimizes the "loss plus penalty".
- A tiny alpha imposes no penalty on the coefficient size, and is equivalent to a normal linear model.
- Increasing the alpha penalizes the coefficients and shrinks them toward zero.

Ridge Regression is similar to least squares, except we include a penalty term,

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2,$$

the λ term is a tuning parameter. When it is zero we get least squares, as it increases the term, $\lambda\sum_{j=1}^{p}\beta_j^2$ (the shrinkage penalty) has more of an

impact and the coefficients will *approach* zero.

Ridge Regression is similar to least squares, except we include a penalty term,

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2,$$

the λ term is a tuning parameter. When it is zero we get least squares, as it increases the term, $\lambda\sum_{j=1}^{p}\beta_j^2$ (the shrinkage penalty) has more of an

impact and the coefficients will *approach* zero.

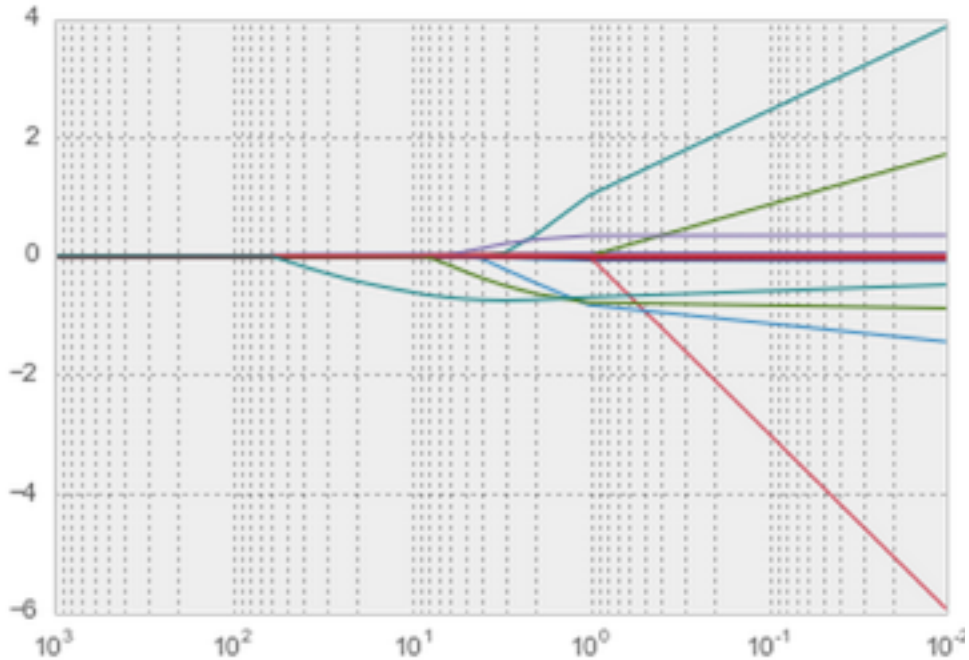
Lasso Regression is similar to Ridge Regression, except we have the absolute value of beta in our penalty term,

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

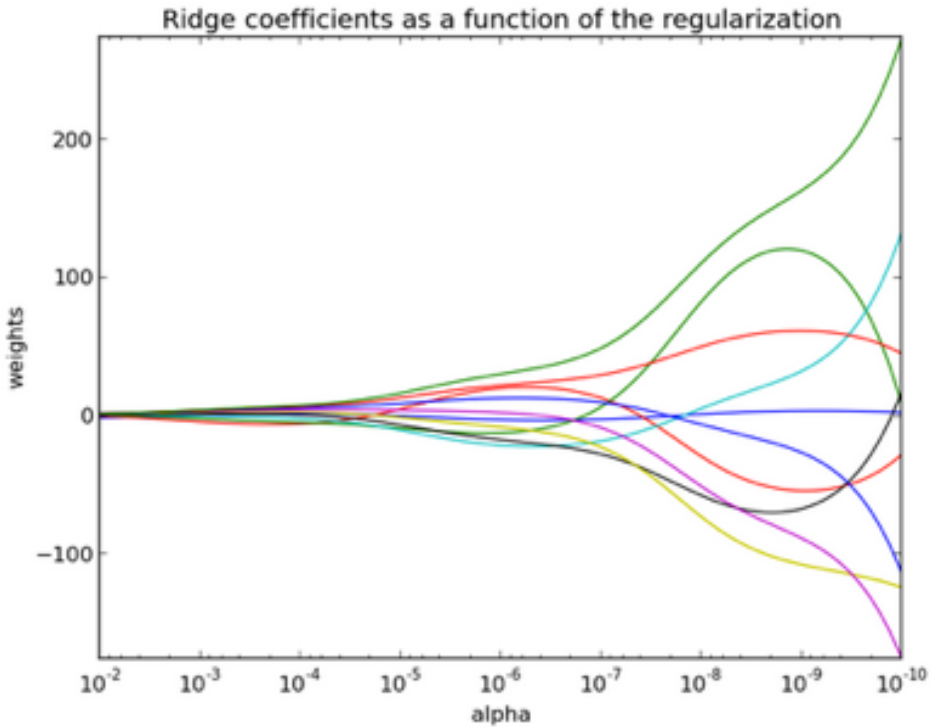
the λ term is a tuning parameter. When it is zero we get least squares, as it increases the term, $\lambda\sum_{j=1}^{p}|\beta_j|$ (the shrinkage penalty) has more of an

impact and the coefficients will **equal** zero.

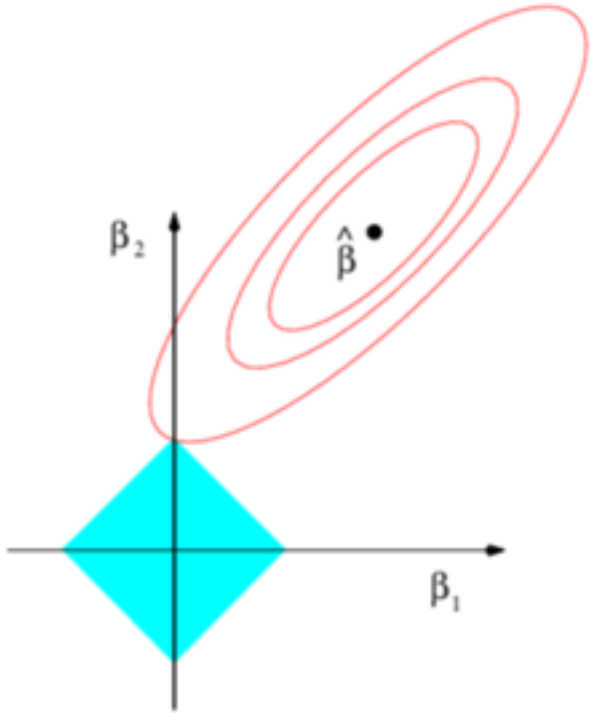
Lasso (L1)



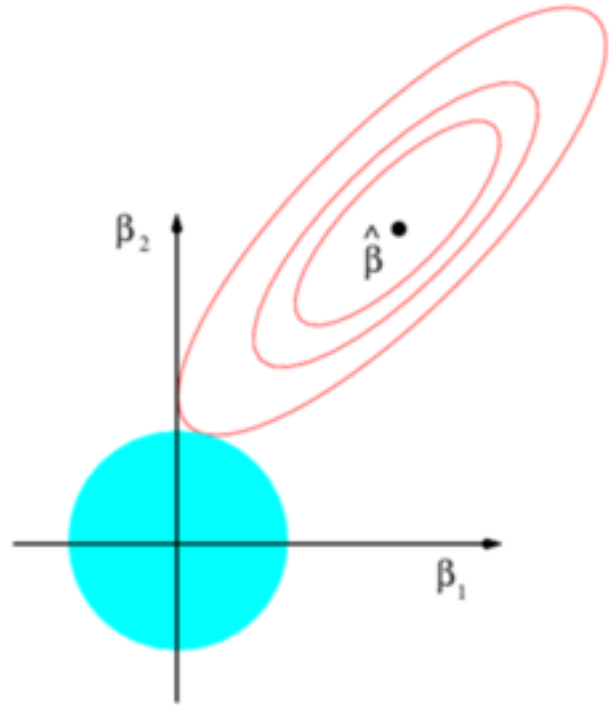
Ridge (L2)

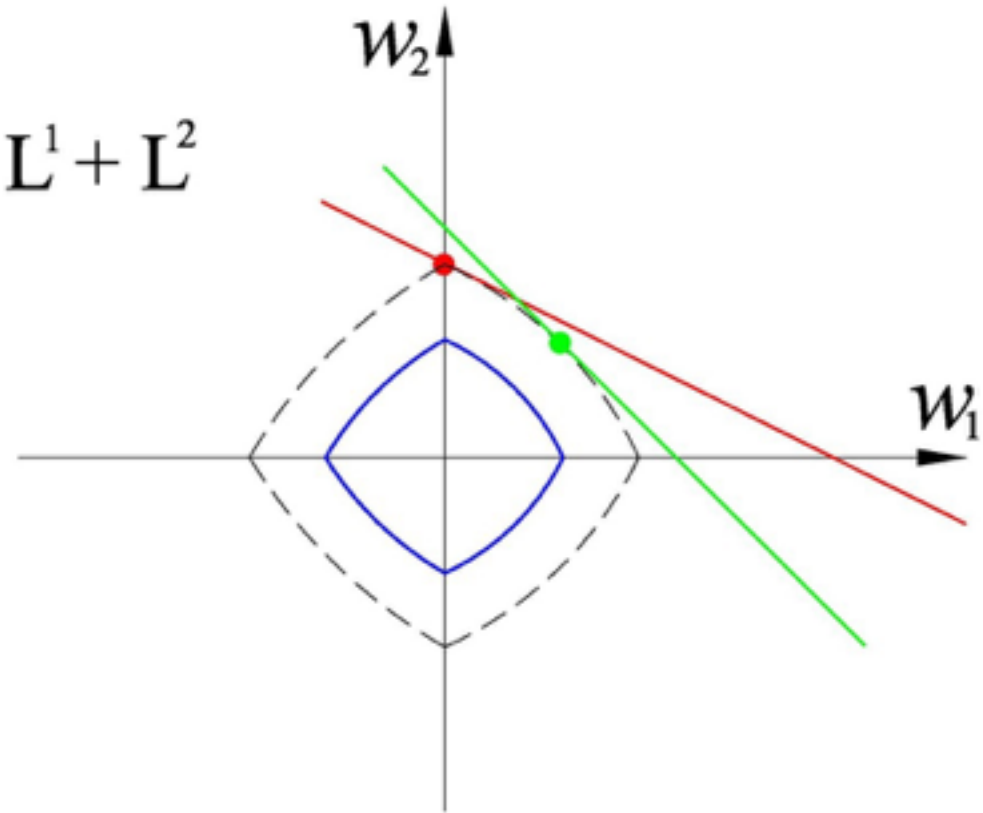


Lasso (L1)



Ridge (L2)




w_2
$L^1 + L^2$
w_1

Lasso regularization is useful if we believe many features are irrelevant, since a feature with a zero coefficient is essentially removed from the model. Thus, it is a useful technique for feature selection.

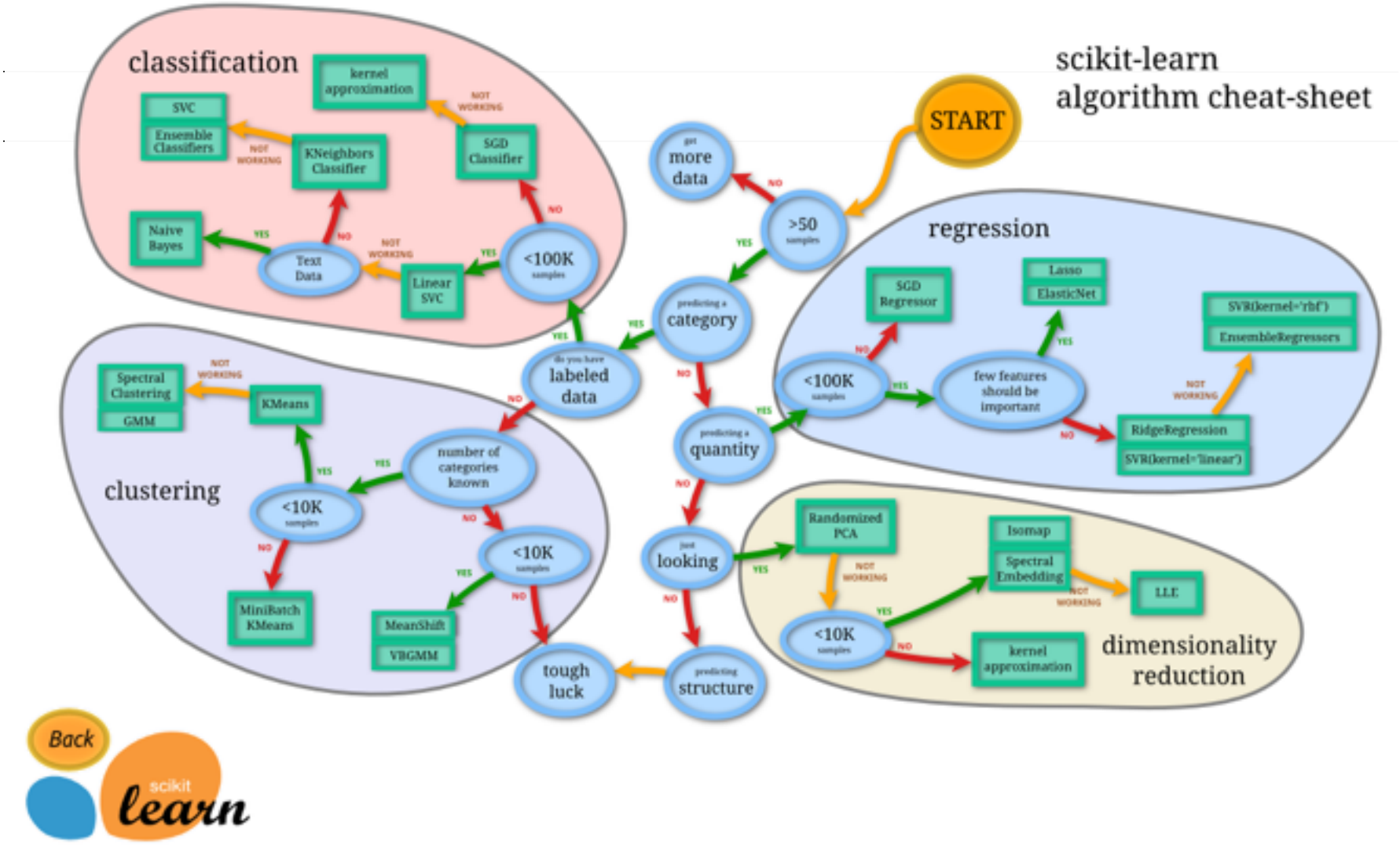
LAB

SYNCHING YOUR FORK WITH THE COURSE REPO

1. re-name your labs with lab_name.<yourname>.ipynb (to prevent a conflict)
2. cd <path to the root of your SYD_DAT_6 local repo>
3. commit your changes ahead of sync
   - git status
   - git add .
   - git commit -m "descriptive label for the commit"
   - git status
4. download new material from official course repo (upstream) and merge it
   - git checkout master (ensures you are in the master branch)
   - git fetch upstream
   - git merge upstream/master

DATA SCIENCE PART TIME COURSE

DIMENSIONALITY REDUCTION


scikit-learn
algorithm cheat-sheet
START
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
get more data
NO
>50 samples
YES
regression
SGD Regressor
Lasso
ElasticNet
SVR(kernel='rbf')
EnsembleRegressors
NO
<100K samples
YES
few features should be important
YES
NOT WORKING
RidgeRegression
NO
SVR(kernel='linear')
predicting a category
YES
NO
do you have labeled data
YES
NO
predicting a quantity
YES
NO
clustering
Spectral Clustering
NOT WORKING
GMM
KMeans
YES
number of categories known
YES
NO
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
MeanShift
VBGMM
NO
just looking
YES
NO
dimensionality reduction
Randomized PCA
NOT WORKING
Isomap
Spectral Embedding
NOT WORKING
LLE
YES
<10K samples
NO
kernel approximation
tough luck
predicting structure
Back
scikit learn

A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

In general, the idea is to regard the dataset as a matrix and to decompose the matrix into simpler, meaningful pieces.

- The number of features in our dataset can be difficult to manage, or even misleading (eg, if the relationships are actually simpler than they appear).

- Ideally, we would like to eliminate redundancy and consolidate the number of variables we're looking at.

- The complexity that comes with a large number of features is due in part to the curse of dimensionality.

- The complexity that comes with a large number of features is due in part to the curse of dimensionality.

- Namely, the sample size needed to accurately estimate a random variable taking values in a d-dimensional feature space grows exponentially with d (almost).

- (More precisely, the sample size grows exponentially with $l \leq d$, the dimension of the manifold embedded in the feature space).

- We'd like to analyze the data using the most meaningful basis (or coordinates) possible.

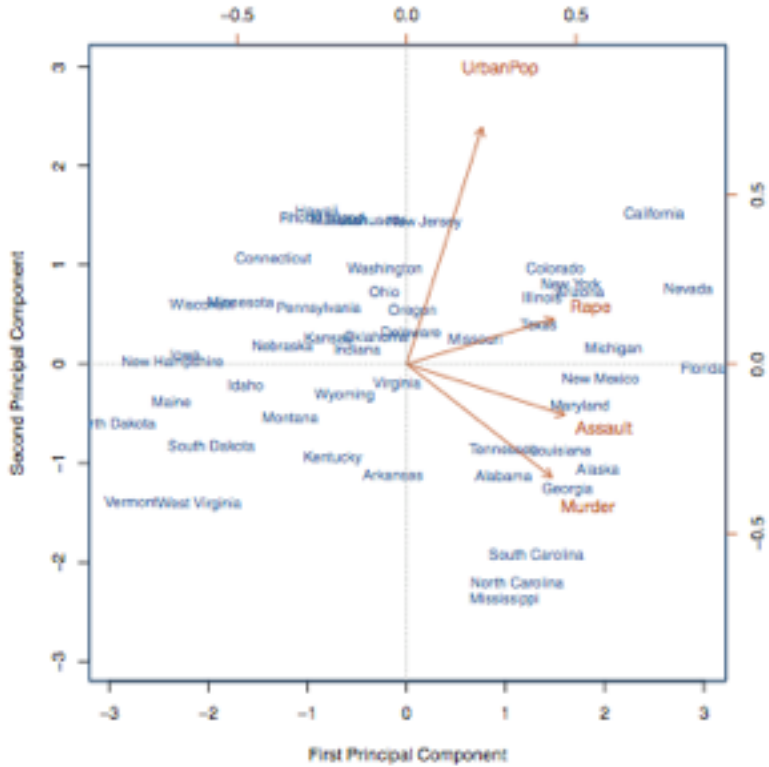
- More precisely: given an n x d matrix X (encoding n observations of a d-dimensional random variable), we want to find a k-dimensional representation of X ($k < d$) that captures the information in the original data, according to some criterion.

DATA SCIENCE PART TIME COURSE

PRINCIPAL COMPONENTS

'It finds a low-dimensional representation of a data set that contains as much as possible of the variation. The idea is that each of the n observations lives in p-dimensional space, but not all of these dimensions are equally interesting. PCA seeks a small number of dimensions that are as interesting as possible, where the concept of interesting is measured by the amount that the observations vary along each dimension. Each of the dimensions found by PCA is a linear combination of the p features.'

- Introduction to Statistical Learning


UrbanPop
Rape
Assault
Murder
California
Nevada
Florida
Michigan
New Mexico
Maryland
Alaska
Georgia
Louisiana
Alabama
Tennessee
Kentucky
Arkansas
South Dakota
Montana
Maine
Idaho
Wyoming
Virginia
Nebraska
Indiana
Ohio
Oregon
Pennsylvania
Connecticut
Washington
Colorado
New Jersey
Vermont
West Virginia
South Carolina
North Carolina
Mississippi
First Principal Component
Second Principal Component

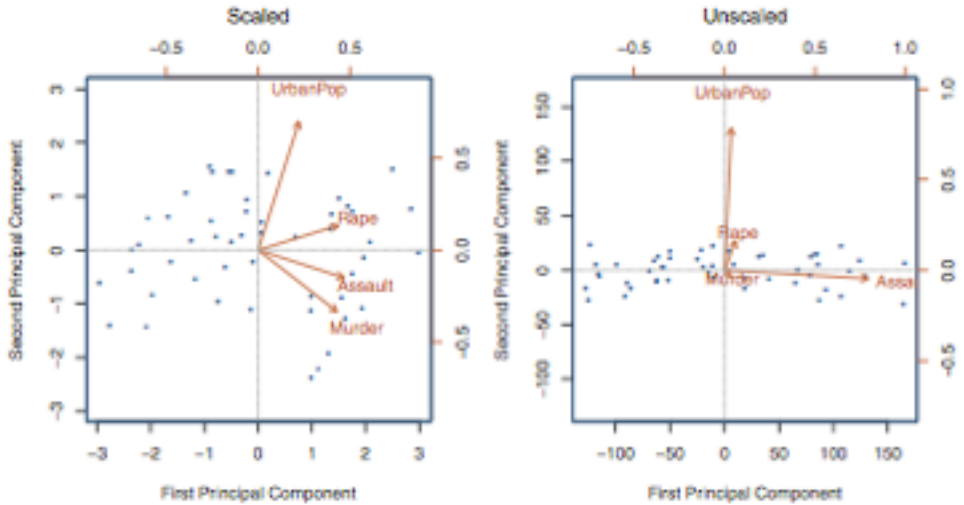
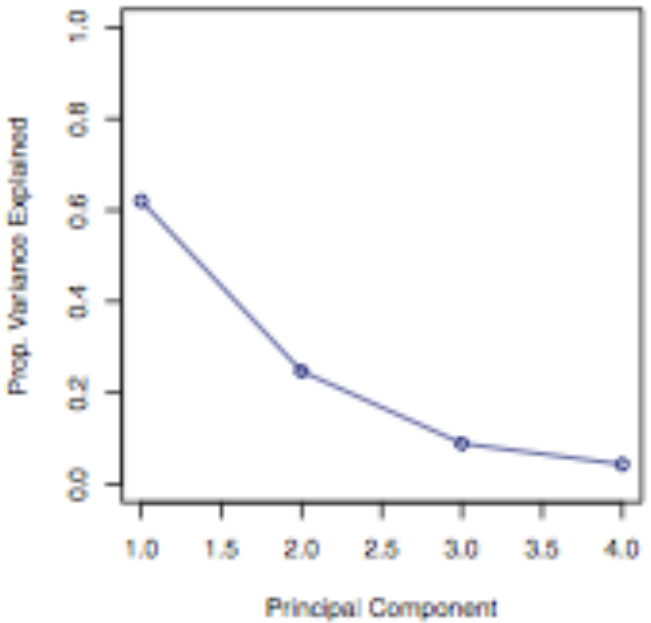
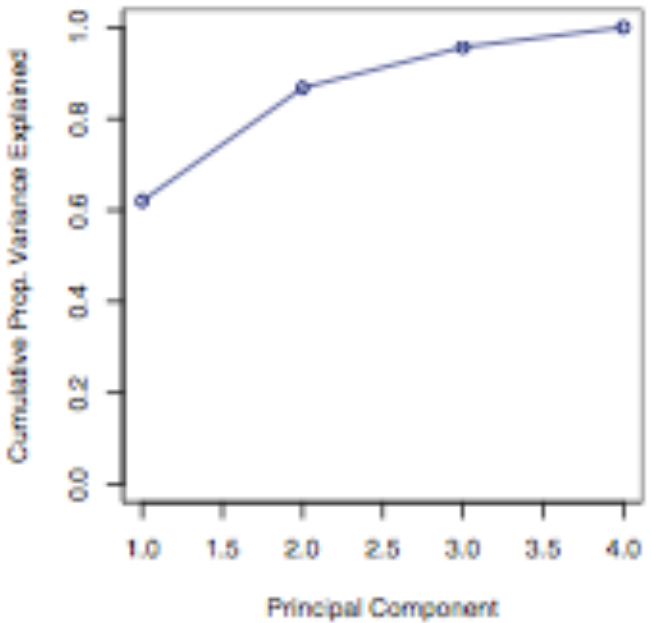


FIGURE 10.3. *Two principal component biplots for the* `USArrests` *data.* Left: *the same as Figure 10.1, with the variables scaled to have unit standard deviations.* Right: *principal components using unscaled data.* `Assault` *has by far the largest loading on the first principal component because it has the highest variance among the four variables. In general, scaling the variables to have standard deviation one is recommended.*


Prop. Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Principal Component


Cumulative Prop. Variance Explained
0.0
0.2
0.4
0.6
0.8
1.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Principal Component

WEEK 4

READINGS

Read the following

- Chapter 6 of Introduction to Statistical Learning, Linear Model Selection and Regularization
- Clustering Methods in Introduction to Statistical Learning, Chapter 10.3 (15 pages)

DISCUSSION TIME