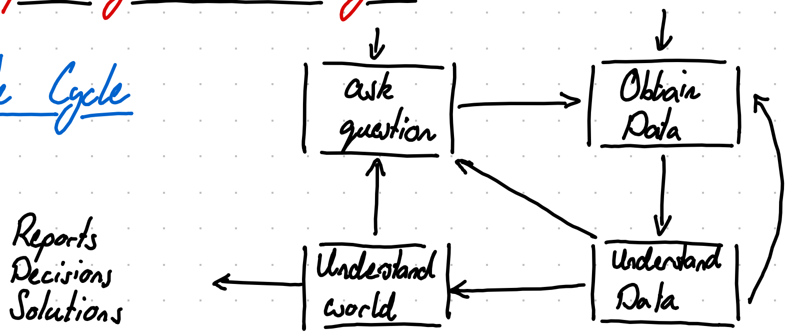
Exploratory Data Analysis

Life Cycle



EDA Properties

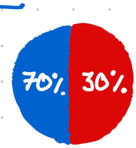
- Structure : Shape of Files
- Granularity : Fine / Course Data
- Scope : Completeness
- Temporality : Time situation
- Faithfulness : To reality

Handling Missing Values

- Remove records
- Fill in manually
- Fill with mean
- Fill with prediction from model

Effective Visualizations

1. Graphical Integrity



2. Keep it simple – No 3D!
3. Use a sensible design – Not: × • Δ ☆

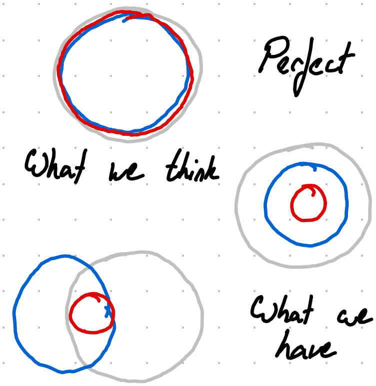
. Use the right display.

↳ Distribution
↳ Relationship
↳ Composition
↳ Comparison

Collection & Sampling

Key Concepts

- Population
- Sampling Frame
- Sample



Types of Error

- Chance Error
- Bias

Samples

Convenience Sample : Whoever is there

Quota Sample : Break down groups (50% male / female)

example:

"Raise your hand if you are familiar with sampling bias"

100% at stats conference

Common Biases

- Selection Bias

"90% of people think Skydiving is safe" asked at a DZ

- Response Bias

"100% of men report having larger than average penis"

- Non-Response Bias

high computer Literacy from email survey

Regression

Modelling Process

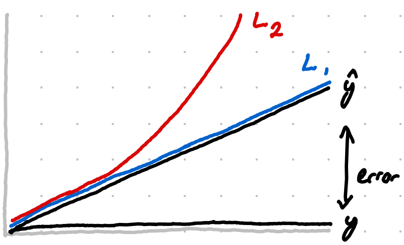
1. Choose a model
2. Choose a loss function
3. Fit the model
4. Evaluate Performance

Loss Functions

$L(y, \hat{y}) =$

$L1$ loss $= |y - \hat{y}|$

$L2$ loss $= (y - \hat{y})^2$



Performance Measures

- $R^2 = \dfrac{\text{Variance of Model}}{\text{Total Variance}}$

Mean Squared Error

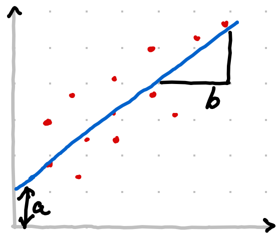
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Simple Linear Regression

$$\hat{y} = \hat{a}x + \hat{b}$$

$$\hat{b} = r\frac{\sigma_y}{\sigma_x}$$

$$\hat{a} = \bar{y} - \hat{b}\bar{x}$$



σ = std dev

$\bar{x}$ = mean x

r = sample correlation coefficient

minimize mean squared error

$$e_i = y_i - \hat{y}_i$$

Empirical Risk

Average loss over entire dataset

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

Parameters

Minimize MSE for SLR

$$R(a,b) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - (a + bx_i)\right)^2$$

Sum of Squared Residuals (SSR)

$$SSR = \sum_{i=1}^{n}(y_i - f(x_i))^2$$

Root Mean Squared Error

$$RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - y_i)^2}{n}}$$

Multiple Regression

Matrix Notation

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & & x_{2p} \\ \vdots & & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{bmatrix}$$

Prediction Vector $\mathbb{R}^n$ → $Y = X\Theta$

Design Vector $\mathbb{R}^{n \times (p+1)}$

Parameter Vector $\mathbb{R}^{(p+1)}$

Process

① $\hat{Y} = X\Theta$

② Loss function

$$R(\Theta) = \frac{1}{n} \| Y - X\Theta \|_2^2$$

③ Fit model

$$\hat{\Theta} = (X^T X)^{-1} X^T Y$$

④ Evaluate

Gradient Descent

$$\Theta^{t+1} = \Theta^t - \alpha \nabla_{\Theta} L(\Theta, X, y)$$

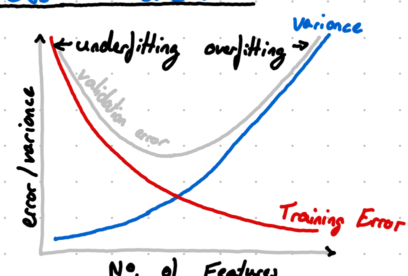
Θ = model weights
L = Loss function
α = Learning rate
y = true values

Feature Engineering

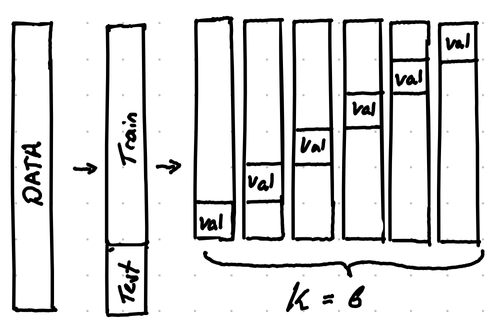
$$y = \Theta_0 + \Theta_1 x + \Theta_2 x^2 \rightarrow y = \Theta_0 + \Theta_1 \phi_1 + \Theta_2 \phi_2$$

$$Y = X\Theta \rightarrow Y = \Phi\Theta$$

Cross Validation



K-fold Cross Validation

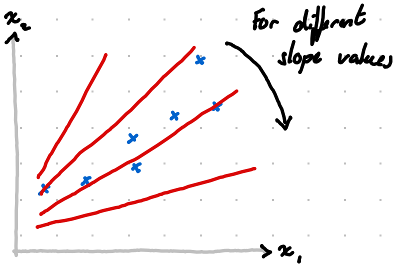


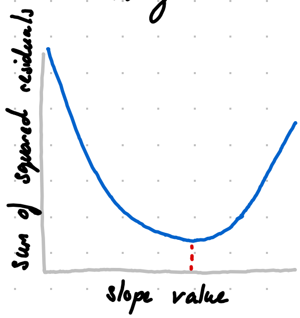
Regularization

... controlling for and reducing overfitting

Sum of Squared Residuals

works by "prefering" certain parameter values shrinking them towards zero





applying may:

Reduce variance

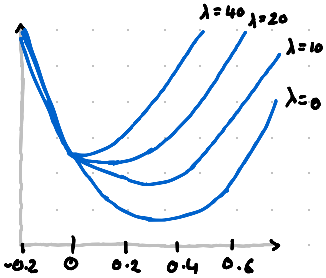
Increase bias

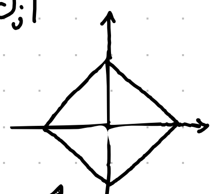
Lasso (L1)

Absolute value penalty

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i-(\theta_0+\theta_1 x_{i1}+\ldots+\theta_p x_{ip})\right)^2+\lambda\sum_{j=1}^{p}|\theta_j|$$

Increasing the lasso penalty the optimal value shifts towards and becomes zero (where the kink is on the graph)



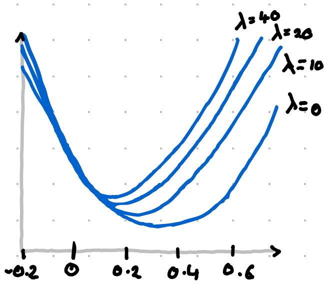


Isosurface

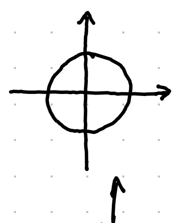
Ridge Regression (L2)

Squared Value penalty

$$\frac{1}{n}\sum_{i=1}^{n}\left(y_i-(\theta_0+\theta_1 x_{i1}+\ldots+\theta_p x_{ip})\right)^2+\lambda\sum_{j=1}^{p}\theta_j^2$$



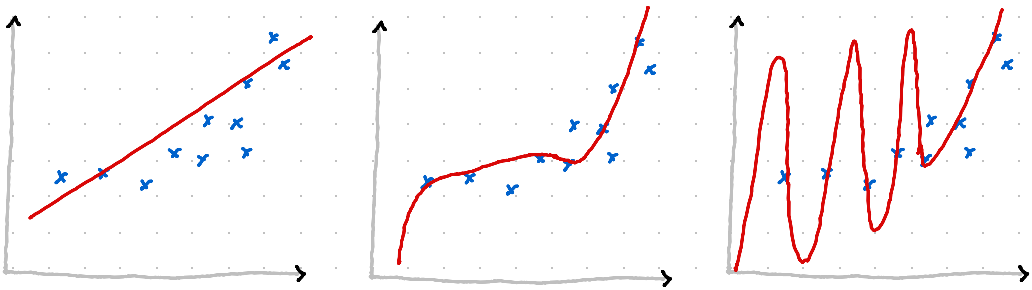
As we increase λ for the ridge regression penalty the optimal slope approaches but does not reach zero



Isosurface

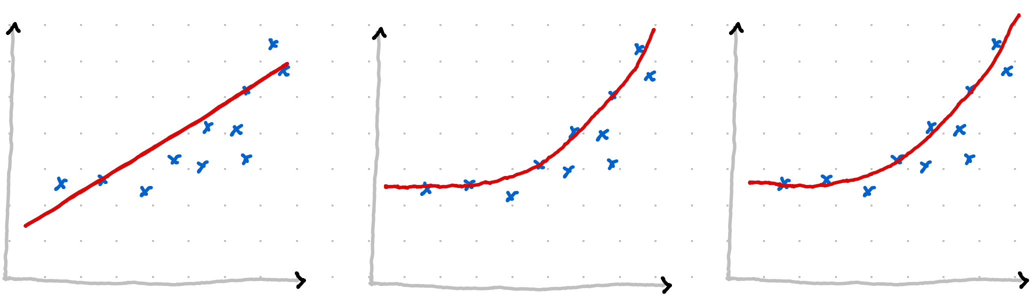
Regularization Intuition

For higher order polynomial functions, regularization becomes far more obvious:



Linear | Low order polynomial | High order polynomial

As the order of the polynomial increases, the model wants to overfit the data to reduce the residuals to zero

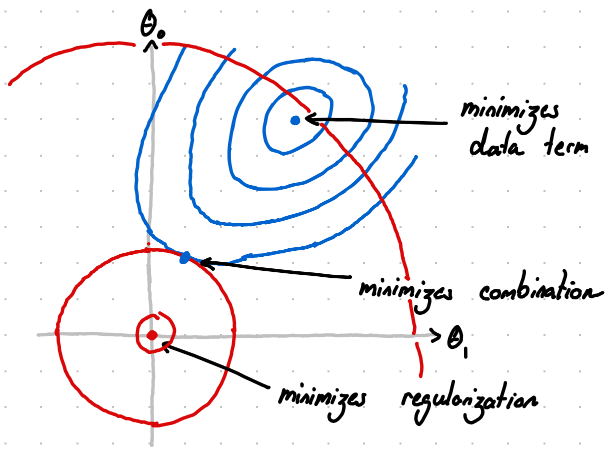


Regularization mean adding a penalty for using so many parameter values to reduce the MSE so the model is dissuaded from overfitting!

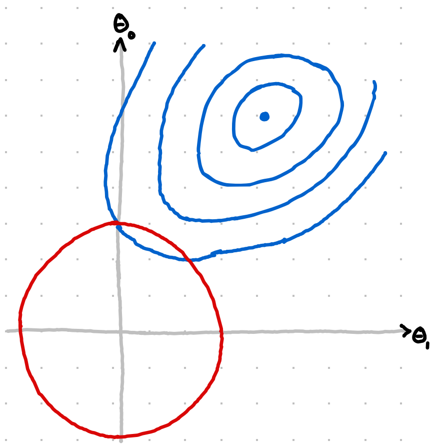
$\Rightarrow$ more complex model without overfitting

Isosurfaces

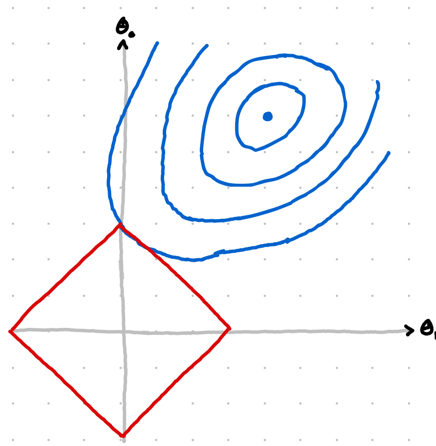
what are these funky shapes?



If the isoshape becomes very large, the regularization has no effect, very small and the model becomes a constant model that only returns zero



Sphere i.e. L_2 Regularization



Hypercube i.e. L_1 Regularization

Classification

Signoid Function

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Logistic Regression

$$\hat{p}_{\theta}(Y = 1 \mid x) = \sigma(x^T \theta)$$

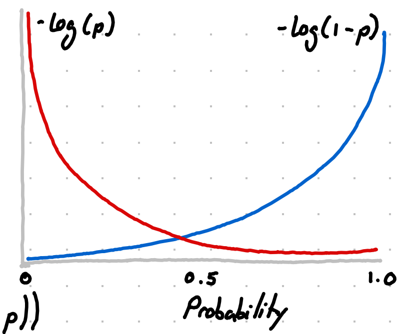
Probability y is 1 given x

Loss

Mean Squared Error ✗ has many issues!

Cross-Entropy Loss ✓

$$L(p) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{if } y = 0 \end{cases}$$



Re-written as:

$$-\left(y \log(p) + (1-y)\log(1-p)\right)$$

Accuracy

$$= \frac{\#\text{ correctly classified}}{\#\text{ total points}}$$

Evaluation Metrics

Class Imbalance Problem

With a dataset of 100 with only 5 negative examples we can get 95% accuracy using just a constant model that only outputs "positive"

Confusion Matrix

		Prediction $\hat{y}$	
		0	1
Actual y	0	True negative	False positive
	1	False negative	True positive

Other Metrics

$$\text{accuracy} = \frac{TP + TN}{n}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

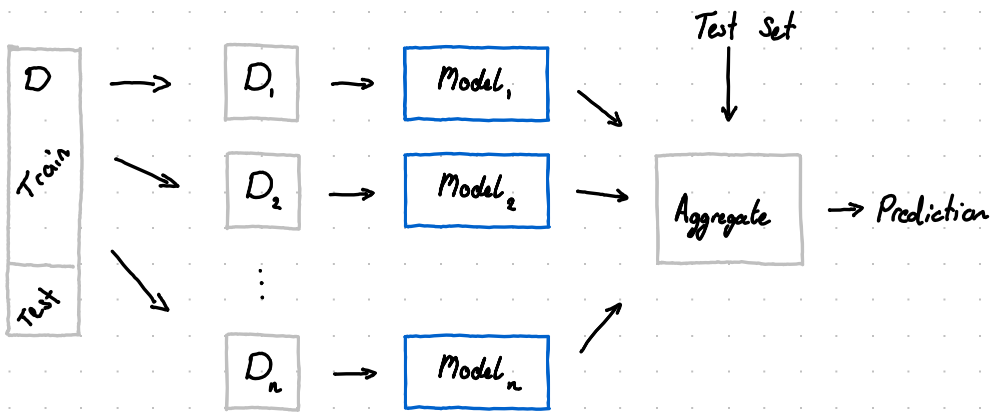
Decision Trees

addressed by bagging & boosting!

+ Simple to explain
+ Nice graphical representation
+ Easy to handel categorical vars

- Tend to overfit data
- Sensitive to small changes (tend to have high variance)

Bagging (bootstrap aggregating)



1. create n random subsets (with replacement)
2. Train n different models
3. Test aggregated models on test set

Pros

- High expressiveness; approximate complex functions
- Low variance; averaging over many models reduces variance

Cons

- Not easily explainable or interpretable
- Trees tend to be highly correlated on the same data, so will split on similar variables in turn

Random Forrest

A modified form of bagging where trees are split on a "random" subset of predictors.

Hyperparameters

- # of predictors to randomly select at each split
- # of trees in forest
- Minimum leaf node size / min samples
- Max depth

Evaluation of RFs

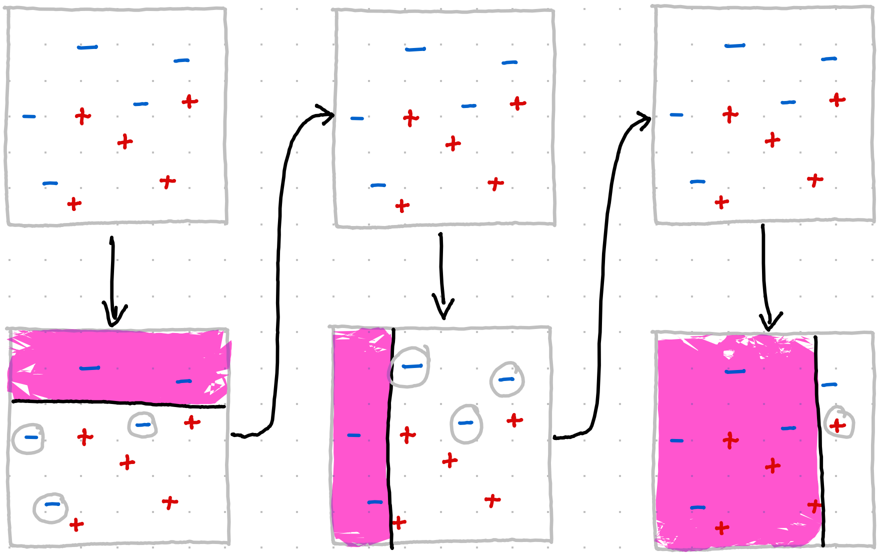
When # of predictors is large but many are not relevant $\Longrightarrow$ Poor Performance

Increasing the # of trees $\Rightarrow$ Not as high a risk of overfitting

# of trees is too large $\Rightarrow$ Increased variance

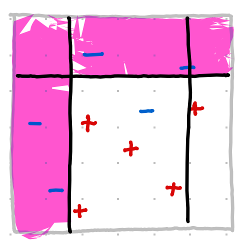
Boosting

Training lots of shitty predictors that together form a good predictor



At each stage we update the weights and retrain the classifier

Then the classifiers are combined



Parameters

q = the number of trees ← select using cross validation

λ = the shrinkage param (or learning rate) ← 0.01 or 0.001 very small requires large q

d = the number of splits ← often $d = 1$ works well

Temporal Analysis

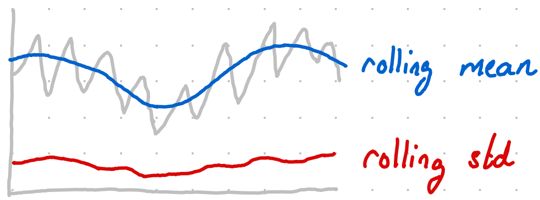
Stationarity

a time series is stationary if its statistical properties are constant over time

- No trend
- variations about mean have constant amplitude
- Short term patterns always look the same

Checks

1. Visual

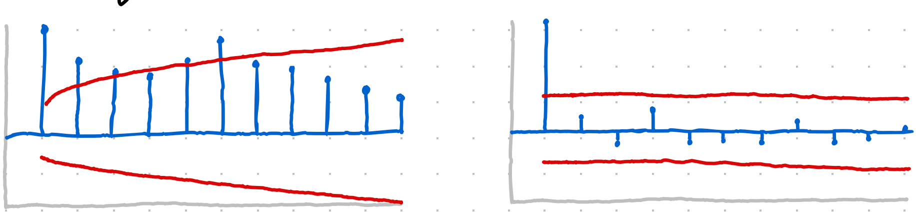


2. Statistical

Unit root test - $y_t = \alpha \cdot y_{t-1} + \epsilon$ a hypothesis test where if $\alpha = 1$ then non-stationary

KPSS test - Checks if the time series is stationary about the mean

3. Correlogram



Values outside the red bands are significant, three or more values well outside the bands indicate a departure from randomness. If the blue bars "die out" this indicates a strong trend.

Stationarization

Removing trends

Differencing

$$y_t = x_t - x_{t-1} \simeq r_t - r_{t-1}$$

To correct for a trend, difference observations from prior observations

Assume a series with an additive trend but no seasonal variation.

Box-Cox Transformation

The parameter λ must be estimated from the data

$$w_t = \begin{cases} \log(x_t) & \text{if } \lambda = 0 \\ \dfrac{x_t^\lambda - 1}{\lambda} & \text{else} \end{cases}$$

Smoothing (moving average)

The q day moving average at time t is the average of x_t over the past q days

$$MA_t^q = \frac{1}{q} \sum_{i=0}^{q-1} x_{t-i}$$

Storytelling

I Inferential Goal (question)
M Model
A Algorithms
C Conclusions and Checking

I'm not going to bother
writing the other shite ...

Dimensionality Reduction

Matrix Decomposition

Singular Value Decomposition

$$\underset{n\times p}{\begin{bmatrix} X \end{bmatrix}} = \underset{n\times p}{\begin{bmatrix} U \end{bmatrix}} \times \underset{p\times p}{\begin{bmatrix} \Sigma \end{bmatrix}} \times \underset{p\times p}{\begin{bmatrix} V^T \end{bmatrix}}$$

orthonormal set

diagonal matrix

orthonormal set

Diagonal matricies and Singular values

$$\Sigma = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix}$$

where $a \geq b \geq c \geq d$

i.e. decreasing order