Model Evaluation - Tuning

# Notes

* Many models have important parameters which cannot be directly estimated from the data.
* Poor choices for these parameters can result in over-fitting.

## Approaches for tuning

A general approach is to define a set of candiate values, generate reliable estimates of model utility across the candidate values, then choose the optimal settings.

Another approach is genetic algorithms. (APM - Mitchel 1998)

Another approach is simplex search methods. (APM - Olsson and Nelson 1975)

# Computation

Taken from APM Chapter 4. Load data.

source("~/MCD/MCD/m.R")  
  
library(caret)  
data(GermanCredit)  
  
GermanCredit <- GermanCredit[, - caret::nearZeroVar(GermanCredit)]  
GermanCredit$CheckingAccountStatus.lt.0 <- NULL  
GermanCredit$SavingsAccountBonds.lt.100 <- NULL  
GermanCredit$EmploymentDuration.lt.1 <- NULL  
GermanCredit$EmploymentDuration.Unemployed <- NULL  
GermanCredit$Personal.Male.Married.Widowed <- NULL  
GermanCredit$Property.Unknown <- NULL  
GermanCredit$Housing.ForFree <- NULL  
  
gc\_p <- m\_data(GermanCredit) %>% m\_response("Class") %>% m\_stratified\_split(p = 0.8) %>% m\_seed(1056)  
  
print("Data Loaded")

## [1] "Data Loaded"

## Tune length

Set the tune length. A set of tuing parameters was automatically selected.

gc\_tune\_m <- gc\_p %>%  
 m\_accuracy\_metric() %>%  
 m\_standardize\_transform() %>%  
 m\_cross\_validation() %>%  
 m\_support\_vector\_machine\_radial\_basis(tune\_length = 10) %>%  
 m\_train()

## [1] "GermanCredit"

gc\_tune\_m %>% m\_get\_results\_data() %>% m\_linechart(x = ~C, y = ~Accuracy, sd = ~AccuracySD, plot = ~base\_plot + scale\_x\_log10())

