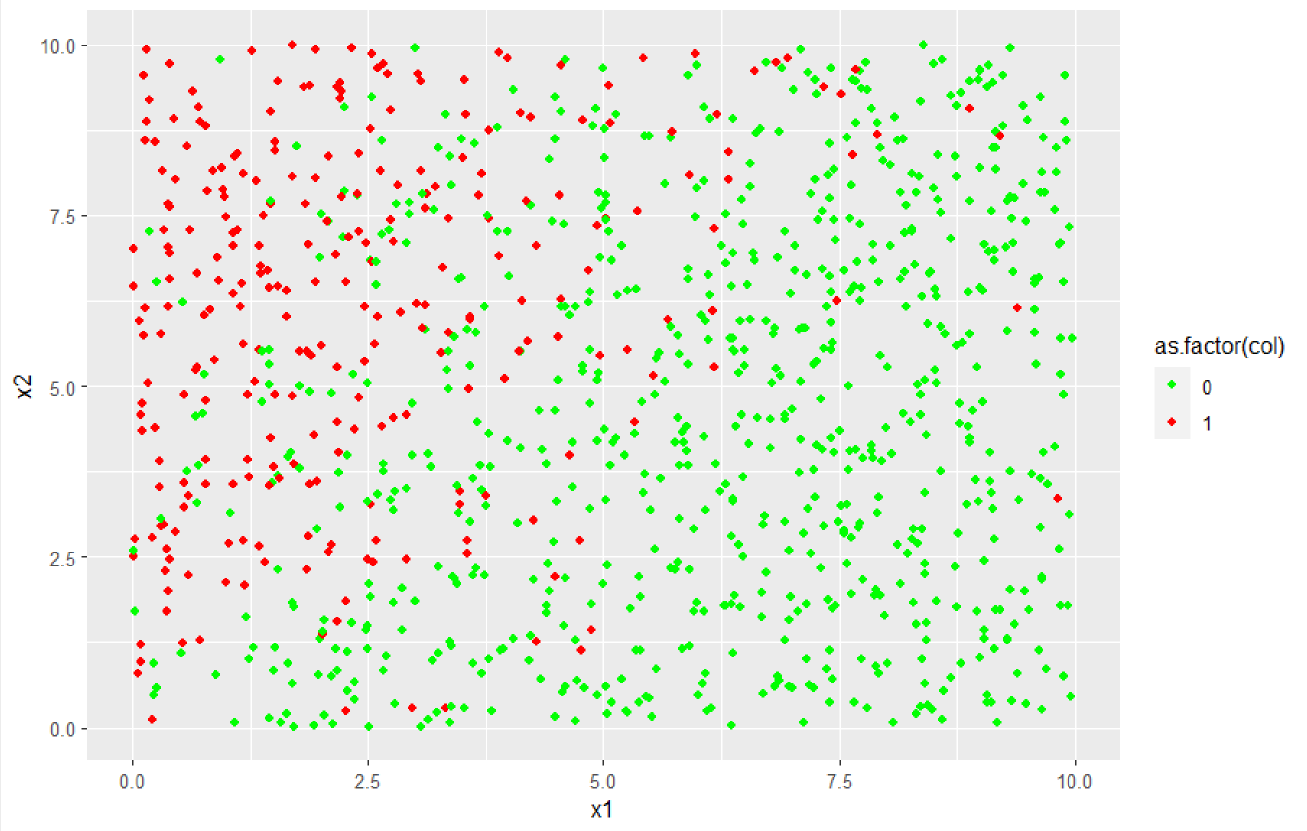
SUPPORT VECTOR CLASSIFIERS

Paul Speaker

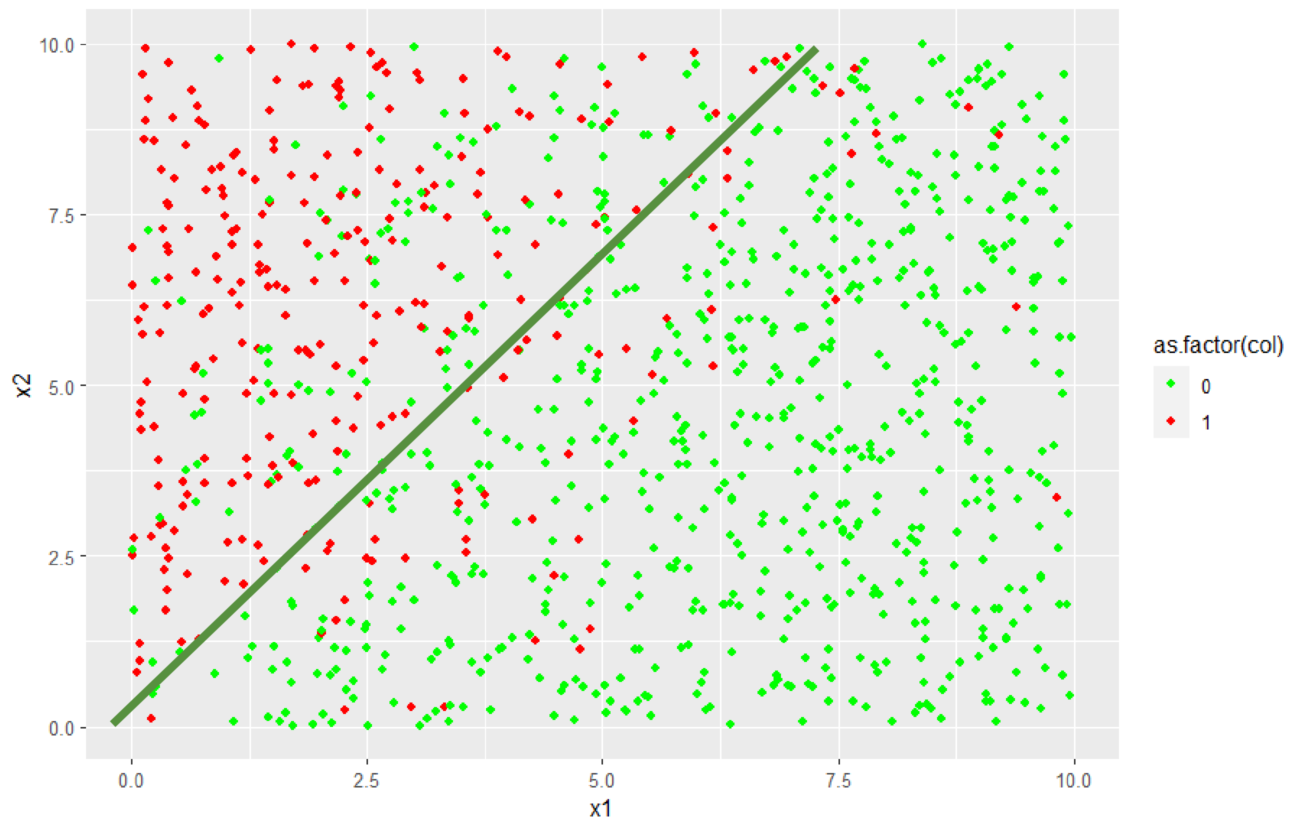
Motivating Example

- Suppose we are using x1 and x2 to predict the color variable, with the data shown to the right
- We sets of decision trees do very well, but they will be awkward in this case
- Decision trees look only at one x at the time.



Motivating Example

- Decision trees look only at one x at the time.
- Shouldn't be instead have a split more like this one?
- So intuitive
- Enter Support Vector Classifiers



Separating boundary

- A linear classifier for a boundary is called a **hyperplane**
- With 2 x's this is just a line
  - $\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$
  - Left hand side $\beta_0 + \beta_1 x_1 + \beta_2 x_2$ is considered the **output of the classifier**
    - On one side the output is positive, the other side the output is negative
- With 3 x's this is a plane
  - $\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 = 0$
- Other classifers that are non-linear
  - Polynomial
  - Radial (gaussian-like)
  - Logistic (like logistic function)
- Consequence of this structure: only numerical variables as inputs for SVM. With categorical, either recode or (for smaller numbers of categorical values) model separately

Classifier Margin

- The **margin** of a linear classifier
  - the width that the boundary could be increased by before hitting a datapoint.
- The **maximum margin linear classifier** is the linear classifier with the maximum margin.
  - Simplest kind of Support Vector Classifier
  - Special case—linear classifier
  - The **support vectors** are the points where are the boundary of the margin
- But what about misses? Maximal margin has nothing to do with minimizing error (unless a perfect split can be found).
- Need loss metric which incorporates both maximal margin and minimizing loss

Support Vector Classifier Loss Function

- For each point, a **hinge loss** is defined
  - Hinge loss = max(0, 1 – ty), where
    - t = actual value of point (for binary target equals 1 or -1, not 0, 1)
    - y = output of the classifier (such as $\beta_0 + \beta_1 x_1 + \beta_2 x_2$ in 2 dimensions)
- Overall minimization is over sum of hinge losses + extra penalty term to make parameters small
- Minimization is taken over parameters of classifiers
  - A hyperplane in n dimensions has n parameters
- Hyperparameter which controls relative weight of hinge losses and penalty terms (just like with Lasso/Ridge Regression)

Support Vector Classifiers in R

- Use e1071 library
- Familiar format: svm(formula, data = <dataset>, type = <>, kernel = <> , cost = <>, gamma = <>)
- Type: type of problem. Set to ‘C-Classification’
- Kernel: linear, polynomial, radial, logistic
- Cost and gamma are hyperparameters
- Defaults for SVM in R:
  - cost = 1,
  - gamma = 1/dim(x)

Support Vector Classifiers in R

- Cost: c hyperparameter
- Gamma: gives distance scale of training point from classifier before it has an effect (only for non-linear kernels)
  - Large gamma values tend to resemble KNN for radial kernel
- Both gamma and c can range over orders of magnitude (between 0.01 and 100)
- For larger values of gamma, c does not have effect on model
- Defaults for SVM in R: cost = 1, gamma = 1/dim(x)
- Caret can be used to tune hyperparameters
  - Define grid search carefully, since there are potentially too many values (log scale)
- What the svm classifier does not output: the parameters (why??????????)