Classification with Support Vector Machines

Maximal Margin Classifier

Maximal Margin

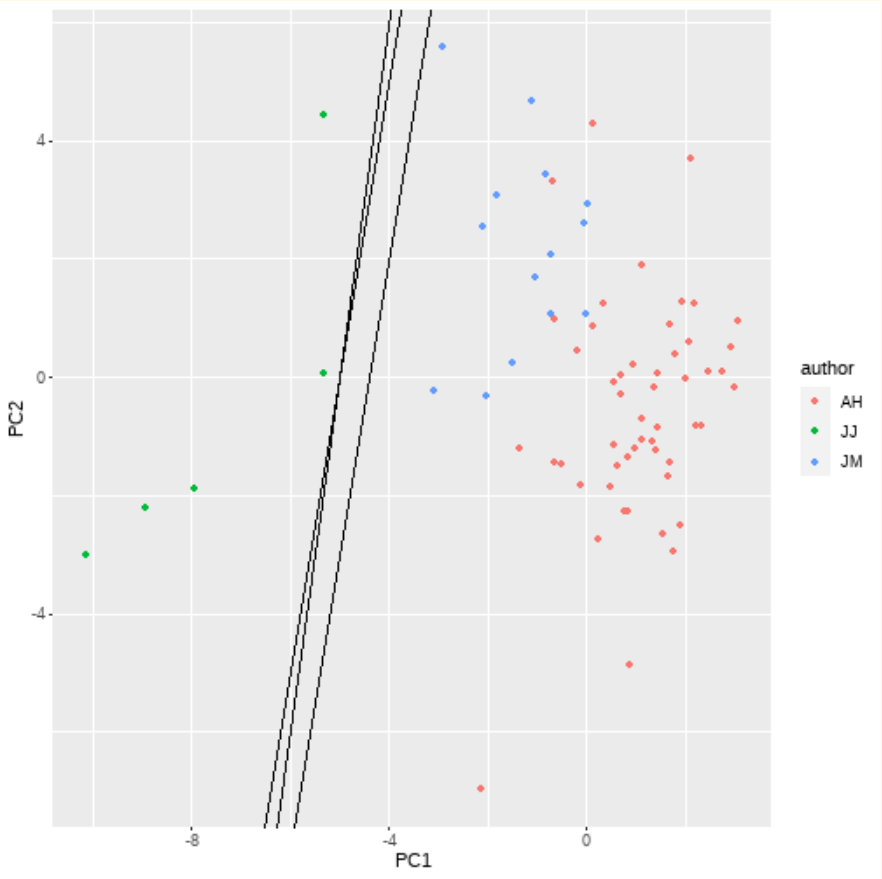
Let's revisit the Federalist papers data.

Recall that we plotted the essays in the first two PC dimensions, and saw that these separated the authors reasonably well:

Maximal Margin

Suppose we are interested in classifying a new observation as "John Jay" or "Not John Jay".

There are many lines we could draw that split the *training* data perfectly between JJ and not JJ

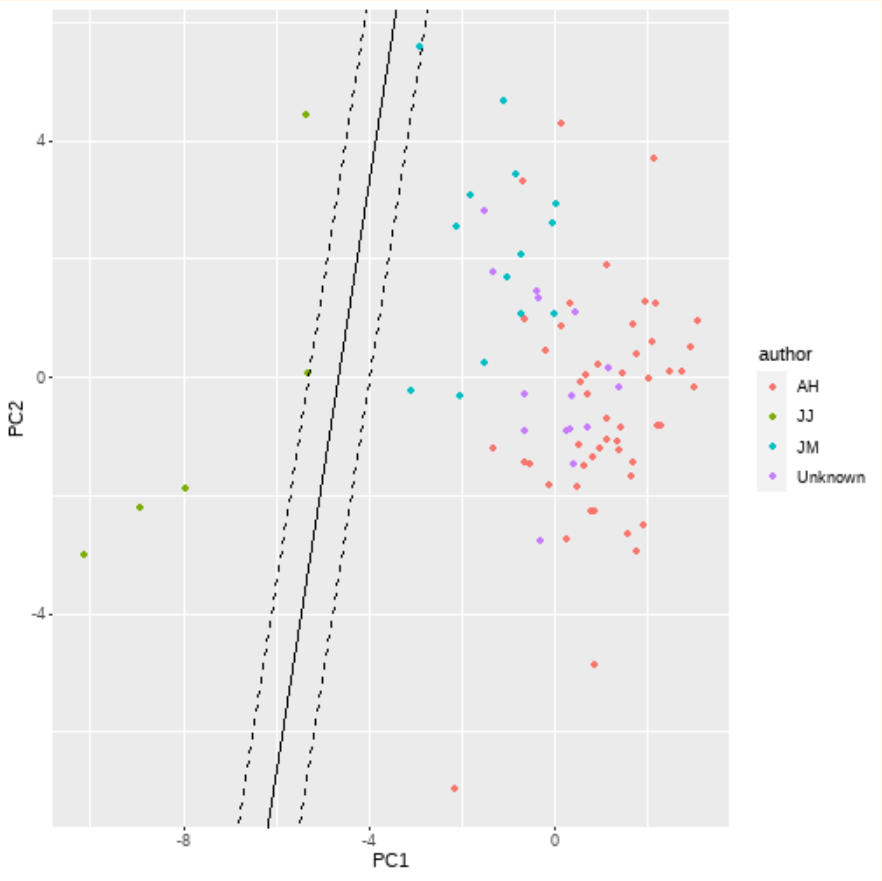

4
0
-4
PC2
-8
-4
0
PC1
author
AH
JJ
JM

Maximal Margin

The "best" one is the one that is furthest from the nearest observation on either side.

Maximal Margin

Let's check out where the essays with unknown authorship fall on this plot:


4
0
-4
PC2
-8
-4
0
PC1
author
AH
JJ
JM
Unknown

Maximal Margin

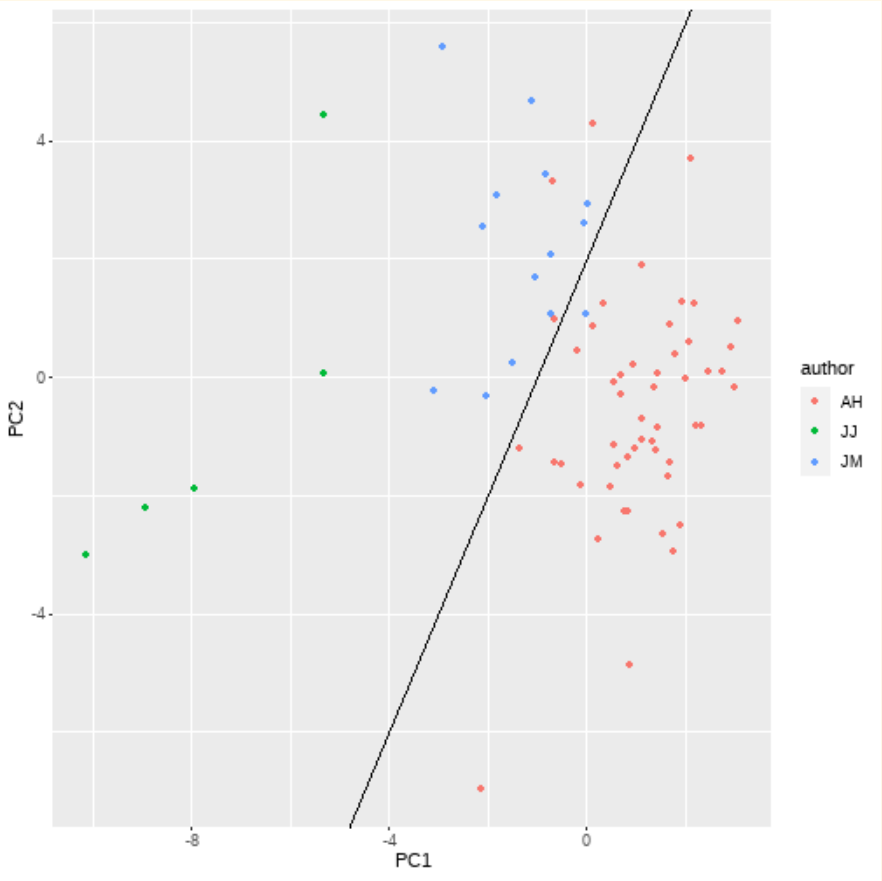
Okay, so what's the problem?

In real situation we rarely have observations that *perfectly* fall on either side of a line/plane.

Adding one more observation could *totally change* our classification line!

None of the unknown essays are John Jay.

Suppose we wanted instead to separate "Hamilton" from "Not Hamilton"


4
0
-4
PC2
-8
-4
0
PC1
author
AH
JJ
JM

Soft Margin

A soft margin is a margin with only a certain number of misclassified points.

There are two decisions to make here:

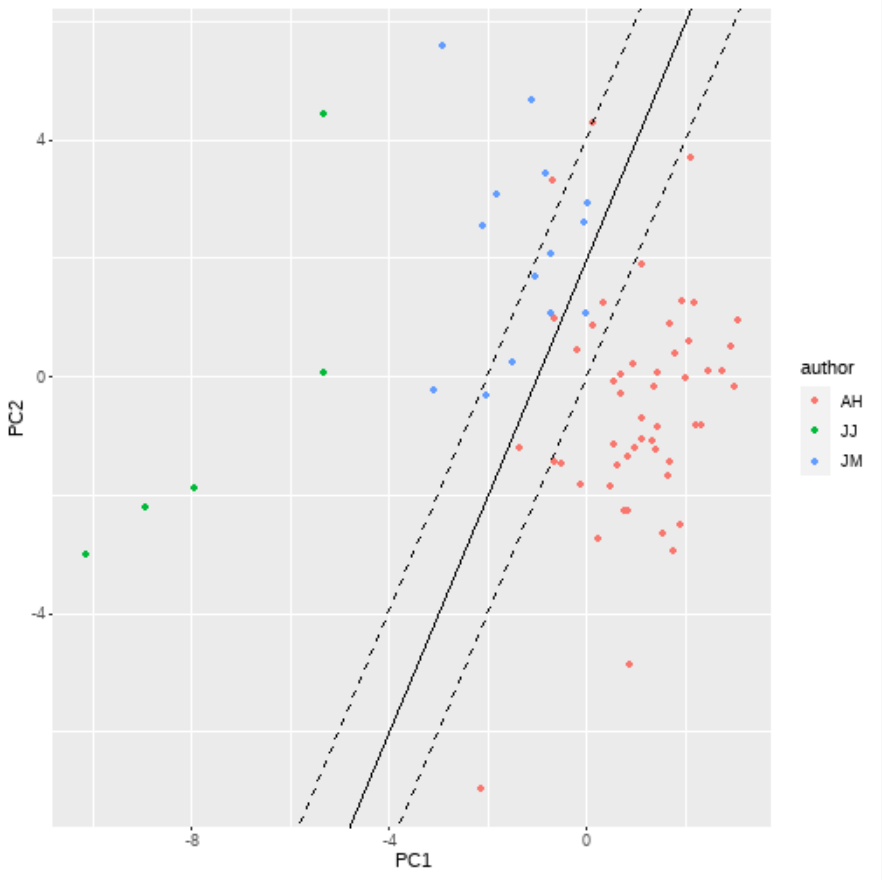
1. How big is our margin?

(M = width of margin)

1. How many misclassified observations are we willing to have?

(C = cost of a misclassified point)

Width of margin: 2 Misclassified points in margin: 3



Support Vector Classifier

The **support vector** is the set of all observations that falling within the **soft margin** that are **misclassified**.

A **support vector classifier** tries to find:

a *line/plane* that will be used to classify future observations ...

... that give us the biggest *margin width*...

... while still respecting the cost, C.

Support Vector Classifier

```
svm_spec <- svm_poly(cost = 2, degree = 1) %>%
  set_mode("classification") %>%
  set_engine("kernlab")

fed_recipe <- recipe(author ~ PC1 + PC2, data = fed_pca_df)

fed_wflow <- workflow() %>%
  add_model(svm_spec) %>%
  add_recipe(fed_recipe)

my_svm <- fed_wflow %>%
  fit(fed_pca_df)
```

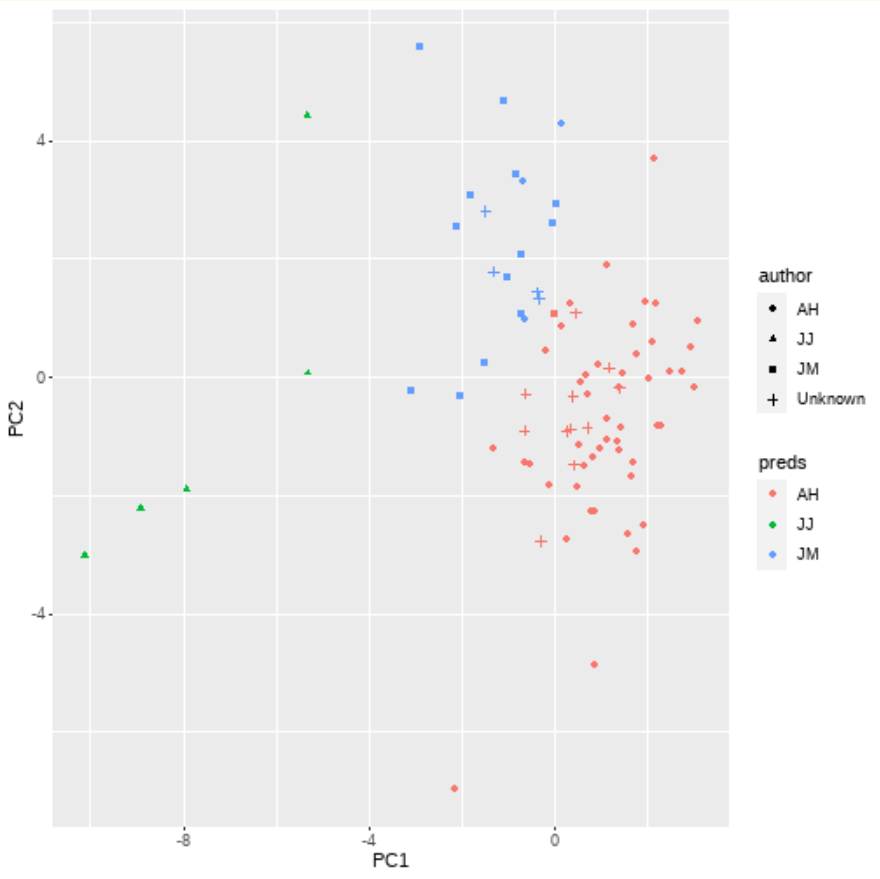
```
fit <- my_svm %>% pull_workflow_fit()
fit
```

```
## parsnip model object
##
## Fit time:  611ms
## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc  (classification)
##  parameter : cost C = 2
##
## Polynomial kernel function.
##  Hyperparameters : degree =  1  scale =  1  offset =  1
##
## Number of Support Vectors : 21
##
## Objective Function Value : -0.7777 -26.67 -2.6
## Training error : 0.057143
## Probability model included.
```

Support Vector Classifier

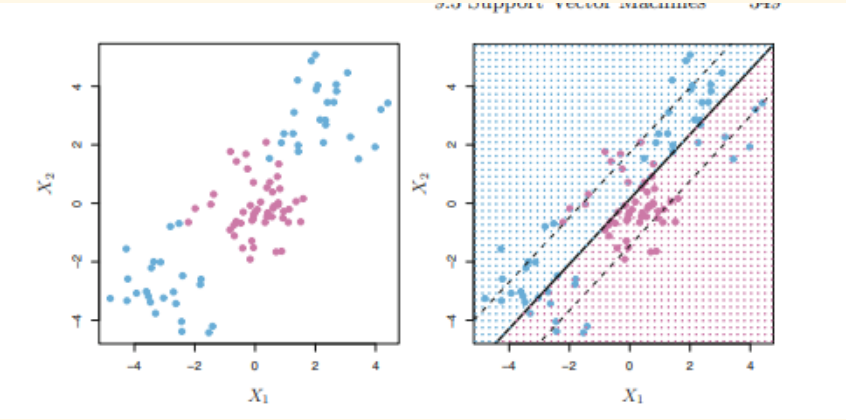
```
predict(my_svm, fed_pca_df_uk)
```

```
## # A tibble: 15 x 1
##    .pred_class
##    <fct>
##  1 JM
##  2 JM
##  3 JM
##  4 AH
##  5 JM
##  6 AH
##  7 AH
##  8 AH
##  9 AH
## 10 AH
## 11 AH
## 12 AH
## 13 AH
## 14 AH
## 15 AH
```


4
0
-4
PC2
-8
-4
0
PC1
author
AH
JJ
JM
Unknown
preds
AH
JJ
JM

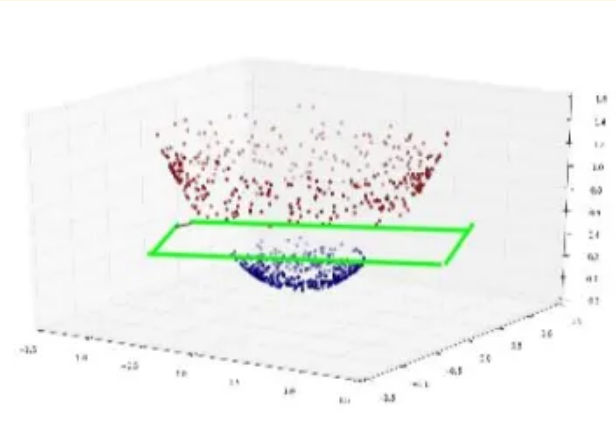
Kernels

What if we simply couldn't separate our data with a line/plane?



Kernels

What if we imagine our points exist in three dimensions?



Support Vector Machine

A support vector machine classifies observations using dividers in extra dimensions!

In this class, we will only implement polynomial svms.

Try it!

Open Activity-SVM.Rmd

Fit a support vector classifier, tuning the cost parameter

Fit a support vector machine, tuning the cost parameter AND the degree parameter.