Support Vector Machines: 3/22/2018

Eric Rupert

March 21, 2018

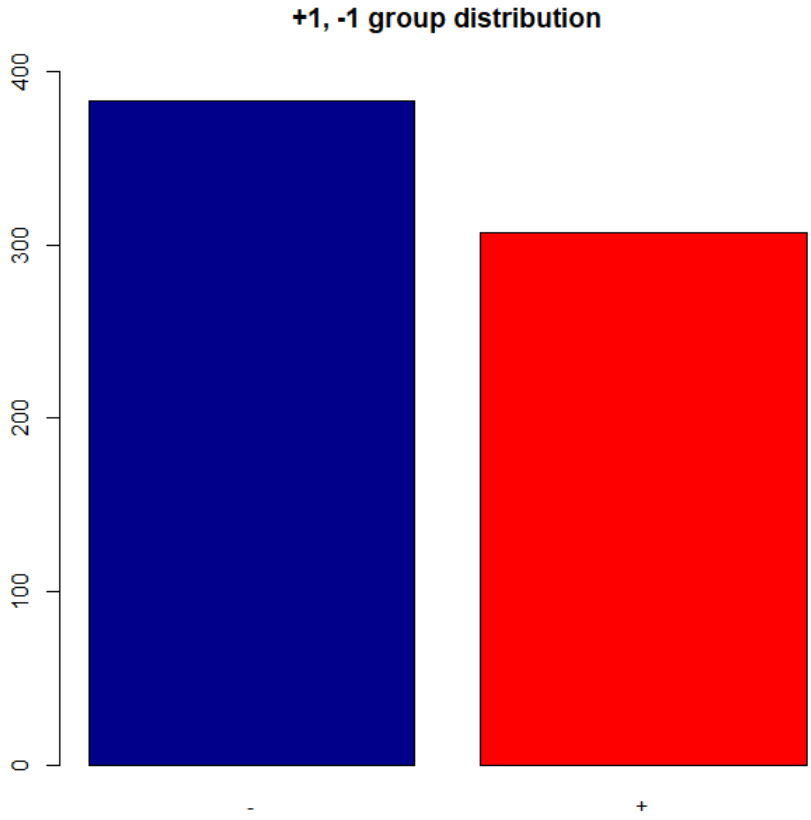
Exploratory data analysis:
Using R's summary function on the credit data frame from the UCI Repository, I found that due to missing values being classified as the character '?', some columns which would normally be classified as numeric, are instead classified as Factors:

```
> str(credit.df)
'data.frame':   690 obs. of  16 variables:
 $ b     : Factor w/ 3 levels "?","a","b": 3 2 2 3 3 3 3 2 3 3 ...
 $ X30.83: Factor w/ 350 levels "?","13.75","15.17",..: 158 330 91 127 45 170 181 76 312 257 ...
 $ X0    : num  0 4.46 0.5 1.54 5.62 ...
 $ u     : Factor w/ 4 levels "?","l","u","y": 3 3 3 3 3 3 3 3 4 4 ...
 $ g     : Factor w/ 4 levels "?","g","gg","p": 2 2 2 2 2 2 2 2 4 4 ...
 $ w     : Factor w/ 15 levels "?","aa","c","cc",..: 14 12 12 14 14 11 13 4 10 14 ...
 $ v     : Factor w/ 10 levels "?","bb","dd",..: 9 5 5 9 9 9 5 9 5 9 ...
 $ X1.25 : num  1.25 3.04 1.5 3.75 1.71 ...
 $ t     : Factor w/ 2 levels "f","t": 2 2 2 2 2 2 2 2 2 2 ...
 $ t.1   : Factor w/ 2 levels "f","t": 2 2 1 2 1 1 1 1 1 1 ...
 $ X01   : int  1 6 0 5 0 0 0 0 0 0 ...
 $ f     : Factor w/ 2 levels "f","t": 1 1 1 2 1 2 2 1 1 2 ...
 $ g.1   : Factor w/ 3 levels "g","p","s": 1 1 1 1 3 1 1 1 1 1 ...
 $ X00202: Factor w/ 171 levels "?","00000","00017",..: 70 13 98 33 39 117 56 25 64 17 ...
 $ X0.1  : int  0 560 824 3 0 0 31285 1349 314 1442 ...
 $ X.    : Factor w/ 2 levels "-","+": 2 2 2 2 2 2 2 2 2 2 ...
```

and are summarized as follows:

```
> summary(credit.df)
 b          X30.83          X0              u          g             w             v          X1.25       
 ?: 12   ?      : 12   Min.   : 0.000   ?:  6   ? :  6   c      :137   v      :399   Min.   : 0.000  
 a:210   22.67  :  9   1st Qu.: 1.000   l:  2   g :519   q      : 78   h      :138   1st Qu.: 0.165  
 b:468   20.42  :  7   Median : 2.750   u:519   gg:  2   w      : 64   bb     : 59   Median : 1.000  
         18.83  :  6   Mean   : 4.759   y:163   p :163   i      : 59   ff     : 57   Mean   : 2.223  
         19.17  :  6   3rd Qu.: 7.207                    aa     : 54   ?      :  9   3rd Qu.: 2.625  
         20.67  :  6   Max.   :28.000                    ff     : 53   j      :  8   Max.   :28.500  
         (Other):644                                     (Other):245   (Other): 20                   
 t       t.1          X01        f       g.1         X00202         X0.1           X.      
 f:329   f:395   Min.   : 0.0   f:374   g:625   00000  :132   Min.   :     0.0   -:383  
 t:361   t:295   1st Qu.: 0.0   t:316   p:  8   00120  : 35   1st Qu.:     0.0   +:307  
                 Median : 0.0           s: 57   00200  : 35   Median :     5.0          
                 Mean   : 2.4                   00160  : 34   Mean   :  1017.4          
                 3rd Qu.: 3.0                   00080  : 30   3rd Qu.:   395.5          
                 Max.   :67.0                   00100  : 30   Max.   :100000.0          
                                                (Other):394                             
```

with Y label distribution:



Model Build and Grid Search:
The following code was used in a manual grid search:

```
>svm.model<-svm(X. .,data=credit.df,type="C-classification",cost=1,kernel="linear",
cross=10)
>predict<-fitted(svm.model)
>cm<-table(credit.df$X.,predict)
>summary(svm.model)
```

Where cost and kernel (and kernel variables) were varied. The following are examples of the results:

Kernel	Cost	Cross Validated Accuracy
Linear	1.0	87.1
	0.1	85.6
	10	82.9
	100	82.0
Poly-2		
	1	55.6

	0.1	55.6
	10	56.0
	100	74.6
	1000	86.4
coef0=1	1	81.4
	10	85.6
	100	86.8
	1000	85
Poly-3	1	55.6
	0.1	55.6
	10	55.6
	100	56
	1000	58.1
coef0=4	1	85
	10	88.1
	100	83
radial	1	81.4
default gamma	0.1	55.7
gamma=.0018	10	85.6
	100	85.3
gamma=.01	1	85.2
	10	86.1
	100	86.8
	50	86.9
gamma=0.1	1	85.2
	10	85.8
sigmoid	1	73.4
	.1	55.6
	10	86.1
	100	85.5
coef0=1	1	58.5
	10	84.3
	100	85.6

The best models are:
1)Polynomial, degree=3, coef0=4, cost=10 (Cross Validated accuracy=88.1)
2)radial, gamma=.01, cost=50, cross validated accuracy=87.4
3)linear, cost=1, cross validated accuracy=87.1

Confusion Matricies:
Oddly, two of the three models returned identical confusion matricies, which demonstrates the randomness built-in to the svm funciton.
Models 1 & 3:

Actual\predict	-	+
-	372	11
+	16	290

Model 2:

Actual\predict	-	+
-	380	3
+	8	298

The credit data has been heavily anonymized, so it's impossible to truly know if false positives or false negatives are the lesser offence. In this case, it is unimportant, as they both predict more false negatives than false positives (If I were investing, I'd prefer to miss a good opportunity, keeping my money, than invest in a bad one, losing money).

Bootstrapping for Confidence Intervals:
The following code was used, with each of the three best models represented once:

```
>err<-integer(200)
>for (i in 1:200) {
>Btrain.df<-credit.df[sample.int(690, size=558, replace=TRUE),]
>Btest.df<-credit.df[sample.int(690, size=138, replace=TRUE),]
>svm.Bmodel<-svm(X. .,data=Btrain.df,type="C-classification",cost=10,
kernel="polynomial", degree=3,coef0=4)
>pred<-predict(svm.Bmodel, Btest.df)
>cm<-table(Btest.df$X.,pred)
>err[i]=(cm[1,2]+cm[2,1])/length(pred)*100}
>serr<-sort(err)
>ub<-serr[195]
>lb<-serr[5]
```

With the bounds found to be:

Kernel	ub	lb
Radial	10.9	1.45
Linear	12.3	3.62
Poly-3	13.0	3.62

Here it can be sen that the Confidence Intervals almost entirely overlap. With the closeness of Cross Validated accuracy, the same false positive/false negative bias, and such overlapping CIs, there is no significant differance in models. Given the lack of differentiation in the models, I would chose the one that runs the fastest (although I didn't time them), or default to linear, because it's the simplest.