Statistical Machine Learning (STAT W4400)

Homework 2

Linan Qiu
lq2137

October 13, 2015

1 Linear Classification

1.1 Classification Results

$\mathbf{v}_h$ is a unit vector.

```
vh = matrix(c(1/sqrt(2), 1/sqrt(2)), nrow=2)
c = 1/(2 * sqrt(2))
x1 = matrix(c(-3, 0), nrow=2)
x2 = matrix(c(1/2, 1/2), nrow=2)

> sign(crossprod(x1, vh) - c)
     [,1]
[1,]   -1
> sign(crossprod(x2, vh) - c)
     [,1]
[1,]    1
```

$$r_1 = \text{sgn}(< \mathbf{x}_1, \mathbf{v}_h > -c) = -1$$
$$r_1 = \text{sgn}(< \mathbf{x}_2, \mathbf{v}_h > -c) = 1$$

1.2 SVM with Margin

The values of the cross products with offset c are as follows:

```
> crossprod(x1, vh) - c
          [,1]
[1,] -2.474874
> crossprod(x2, vh) - c
          [,1]
[1,] 0.3535534
```

$\mathbf{x}_1$ is still on the negative side and classified the same. However, $\mathbf{x}_2$ is not greater than the margin of 1, hence will not be classified as positive. If $\mathbf{x}_2$ was in the training set, $\mathbf{v}_h$ and c would not have been selected as the classifier as the classifier for a SVM with margin would have ensured that $< \mathbf{v}_h, \mathbf{x} > -c > 1$ or < -1 for all training data.

1.3 Cost Function Approximated by Perceptron Cost Function

It approximates the empirical risk function. Empirical risk function is piece-wise constant, hence would not allow us to gradient descent optimally. The perceptron cost function, by using $\left|\left\langle \mathbf{z}, \begin{pmatrix} 1 \\ \tilde{\mathbf{x}}_i \end{pmatrix} \right\rangle\right|$ instead of just the loss function.

2 Perceptron

2.1 Classify

Included in file `problem2.R`

2.2 Perceptron Training Algorithm

Included in file `problem2.R`

2.3 Train and Test

Included in file `problem2.R`

2.4 2D Representation

Slope can be obtained from $\mathbf{v}_h$ by

$$v_x x + v_y y - c = 0$$

and solving accordingly.

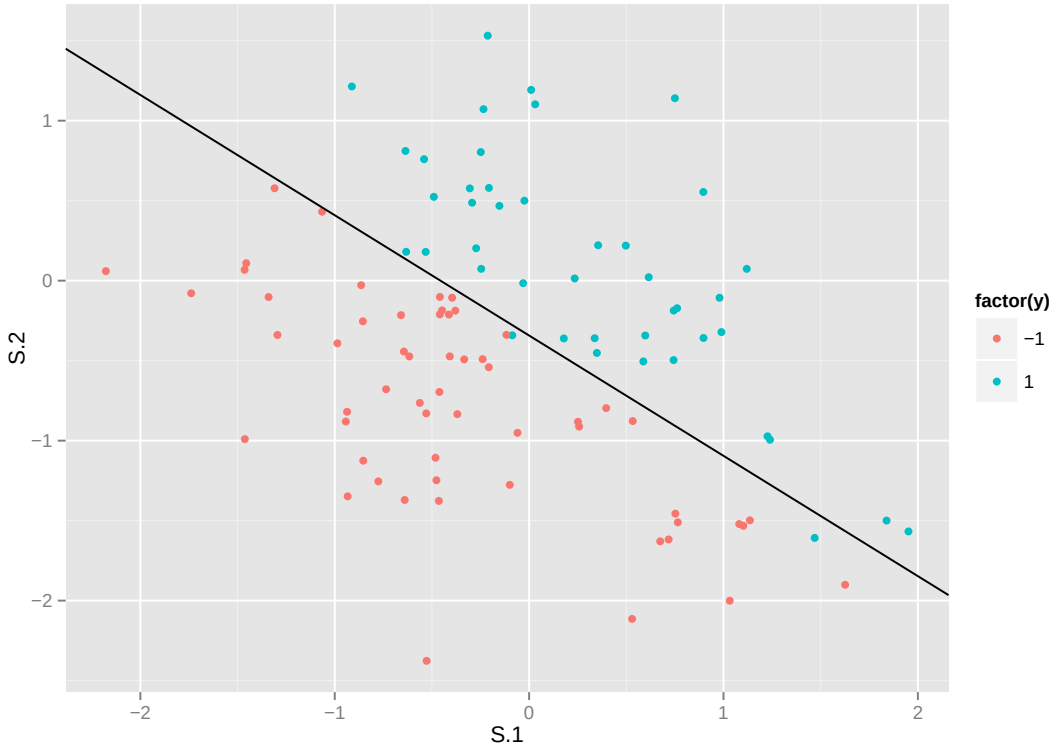

Figure 1: Plot of test data against obtained classifier from `z`.

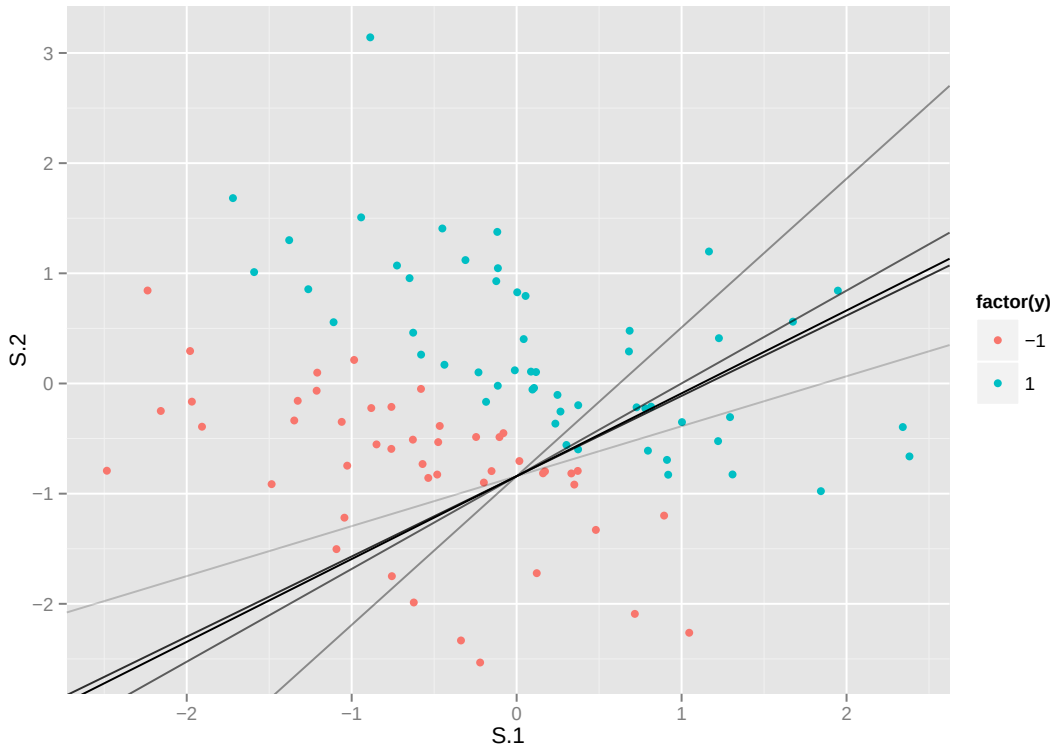

Figure 2: Plot of training data against history of classifiers from `z_history`. Earlier iterations have lower alpha.

3 SVM

3.1 Cross Validation Estimates

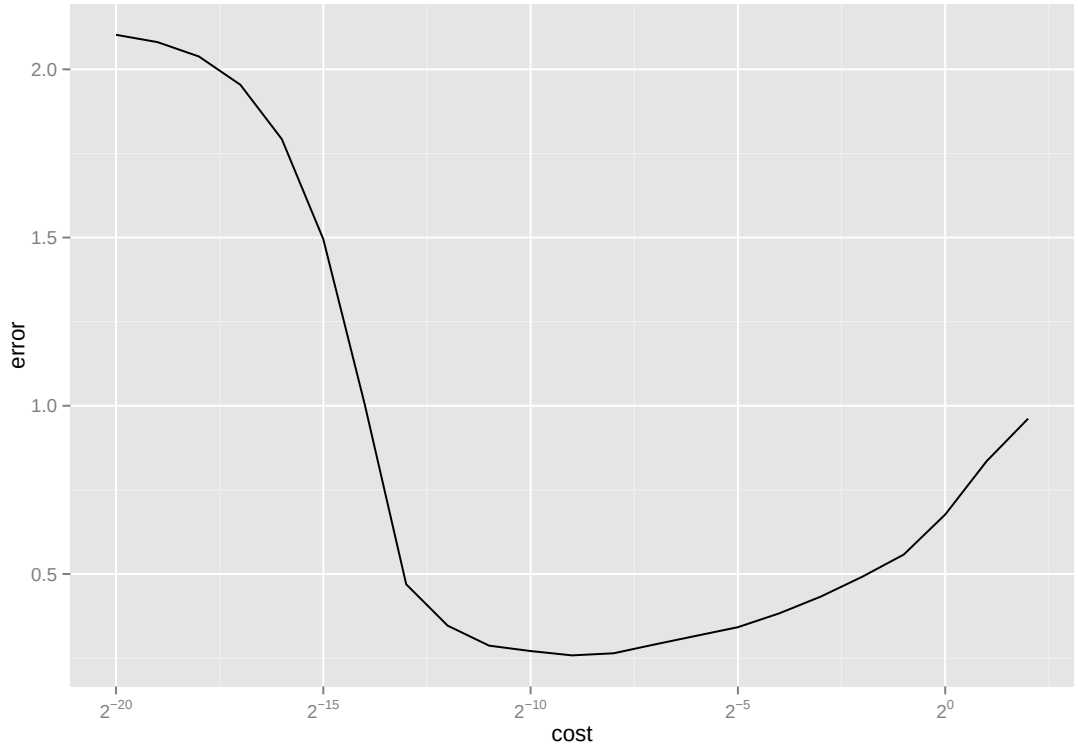

Figure 3: Plot of error against margin parameter (cost) for linear kernel SVM

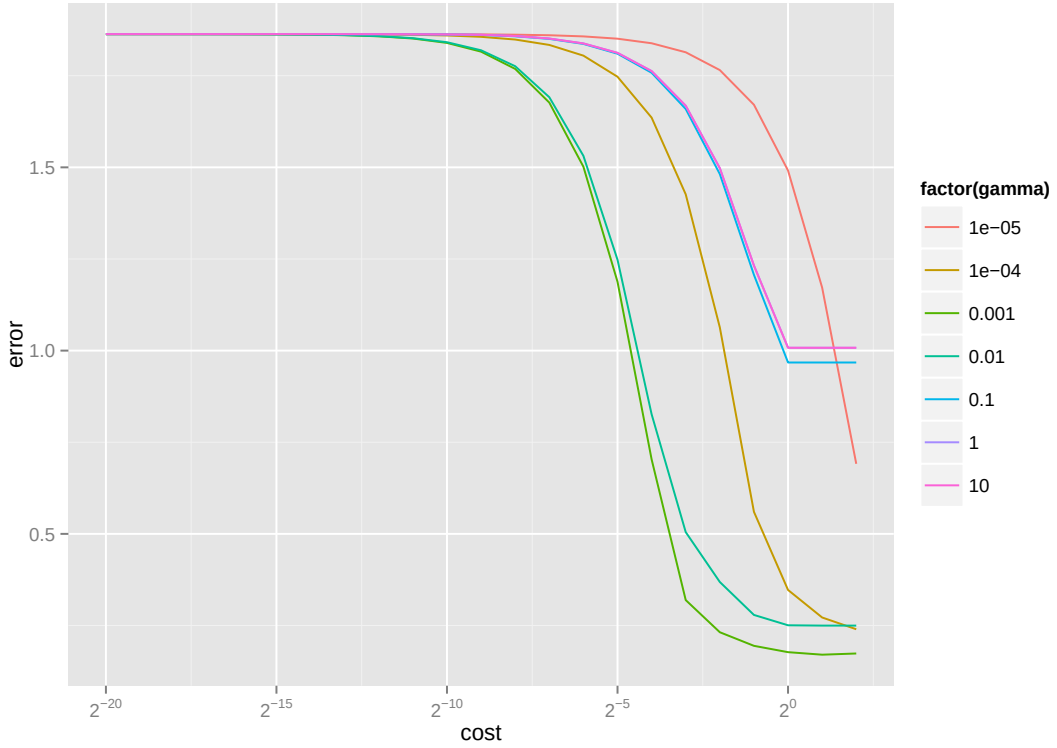

Figure 4: Plot of error against margin parameter (cost) for different kernel bandwidth (gamma) for RBF kernel SVM

3.2 Test

Selected parameter values are:

```
> summary(tuned_linear)

Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
        cost
 0.001953125

- best performance: 0.258095

> summary(tuned_rbf)

```

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 gamma cost
 0.001    2

- best performance: 0.1704325
```

It seems from the training data tuning that the RBF kernel has a better performance. However, upon testing, this is not immediately clear.

```
> classAgreement(table(pred = linear_predict, true = testset
   [, ncol(data)]))
$diag
[1] 0.025

$kappa
[1] 0

$rand
[1] 0.5076923

$crand
[1] 0

> classAgreement(table(pred = rbf_predict, true = testset[,
   ncol(data)]))
$diag
[1] 0.025

$kappa
[1] 0

$rand
[1] 0.5076923

$crand
[1] 0
```