Random KNN Classification and Regression

Shengqiao Li, Donald Adjeroh and E. James Harner

April 22, 2012

1 Random KNN Application to Golub Leukemia Data

1.1 Libraries and Data Sets

```
> require(rknn)
> require(Biobase)
> require(genefilter)
> require(golubEsets)
> require(chemometrics)
>
```

```
> data(Golub_Train)
> data(Golub_Test)
```

2 Random KNN Results

Here, we will apply the Random KNN classifier to the leukemia data sets. We use the first data set as training set and the second data set as the testing set. We choose $m = \sqrt{p} \approx 55$. To choose r, we set $\tilde{\eta} = 0.999$, thus $r = 821$ and $\nu = 14.8$. The following is the result:

```
> golub.rnn<- rknn(data=golub.train, newdata=golub.test, y=golub.train.cl,
+          r=821, mtry=55, seed=20081029);
> golub.rnn
```

```
Call:
rknn(data = golub.train, newdata = golub.test, y = golub.train.cl,
r = 821, mtry = 55, seed = 20081029)

Number of neighbors:  1
Number of knns:  821
No. of variables used for each knn:  55
Prediction:
 [1] ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL ALL
[20] ALL AML AML AML AML ALL AML AML AML AML AML ALL AML AML AML
> confusion(golub.test.cl, fitted(golub.rnn))
classified as-> ALL AML
            ALL  20   0
            AML   2  12
>
```

The confusion matrix of above random KNN classifier shows that of the 34 test samples, only two AML are misclassified as ALL.

In above example, every feature is used and the multiplicities is between 3 and 32 as shown by following output:

```
> length(varNotUsed(golub.rnn));
[1] 0
> golub.varUsed<- varUsed(golub.rnn);
> summary(golub.varUsed);
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    3.0    12.0    15.0    14.8    17.0    32.0
```

To check the distribution of feature multiplicities in these random KNNs, a histogram is plotted in Figure 1.

The multiplicity is approximately symmetric around 15.

As shown above, the classification is quite accurate. But we prefer a simpler model that use a few of genes, i.e., a *gene signature*. Next section will demonstrate feature selection with Random KNN.

```
> #flamehist(golub.varUsed, xlab="Multiplicity", main="")
> hist(golub.varUsed, xlab="Multiplicity", main="")
```

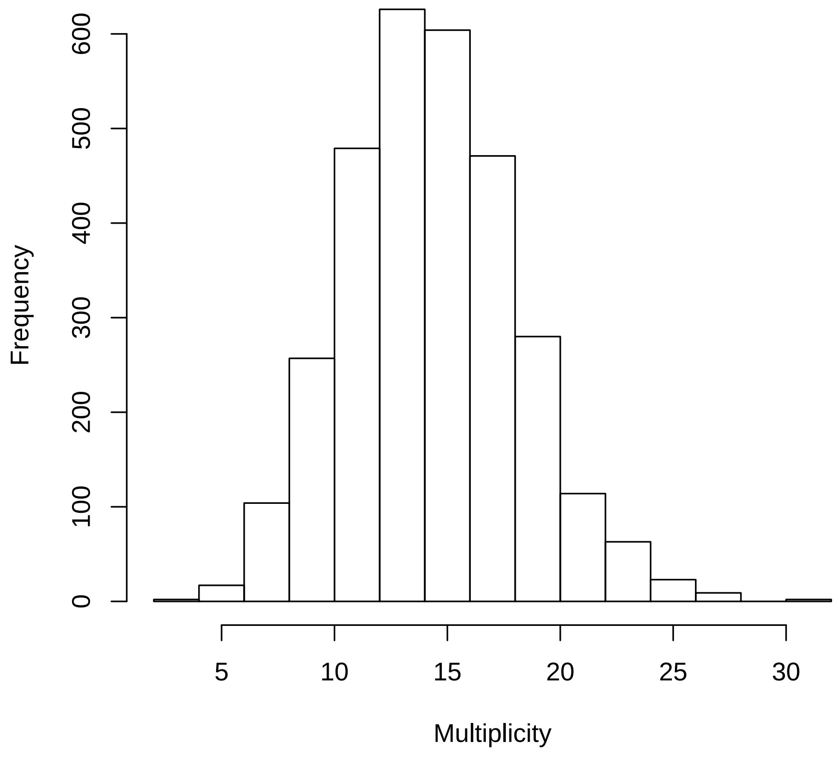

Figure 1: Histogram of the feature multiplicities

3 Feature Selection

Figure 3 shows the mean accuracy increment with decreasing number of features using the Golub leukemia data in the first stage of feature selection.

Figure 4 shows the mean accuracy increment with decreasing number of features of the Golub leukemia data in the second stage of feature selection. From this figure, when 4 genes are left in the model, a maximum mean accuracy is reached. These 4 genes for leukemia classification are:

```
> best.set<- bestset(golub.bel);
> cat(best.set, sep=", "); #remove[1] and quote, and comma.
M27891_at, X95735_at, U27460_at, L09209_s_at
```

Now we use these four genes and the ordinary KNN classifier to classify the 34 independent test samples, the confusion matrix is:

```
> test.class<- knn(golub.train[, best.set], golub.test[, best.set], golub.train.c
> #test.class;
> confusion(golub.test.cl, test.class);
classified as-> ALL AML
            ALL  18   2
            AML   1  13
```

Two ALL samples are classified as AML and one AML is classified as ALL. Total accuracy is as high as 91%. This model is very simple compared with others that use much more genes.

```
> golub.support<- rknnSupport(golub.train, golub.train.cl, k=3)
> golub.support

Call:
rknnSupport(data = golub.train, y = golub.train.cl, k = 3)

Number of knns:  500
No. of variables used for each knn:  55
Accuracy: 0.9473684
Confusion matrix:

classified as-> ALL AML
            ALL 27   0
            AML  2   9

> plot(golub.support, main="Support Criterion Plot")
```

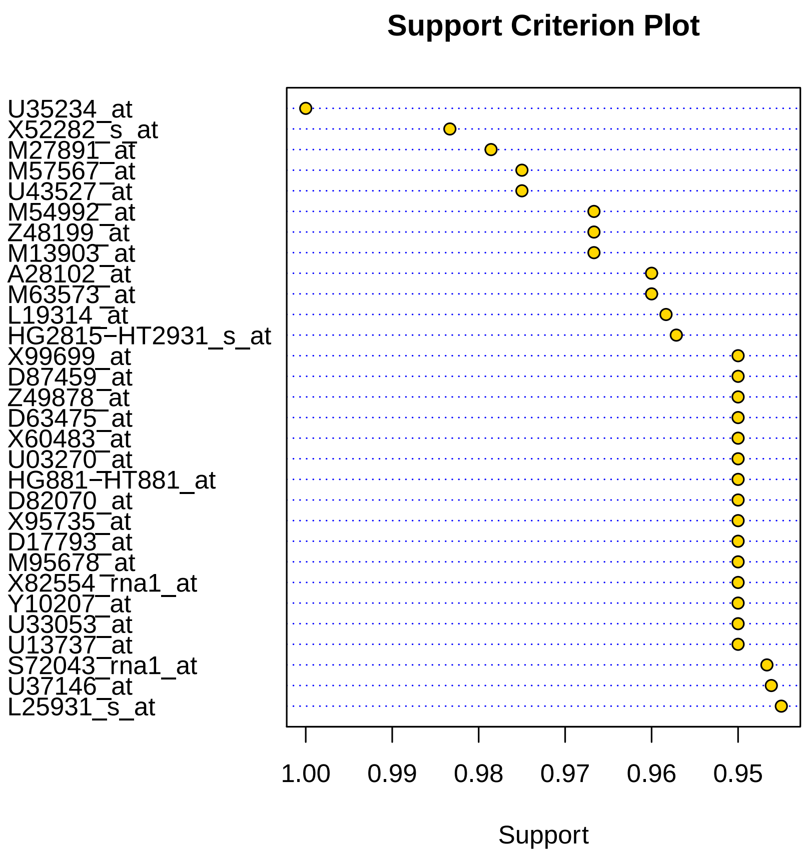

Figure 2: Support plot for Golub leukemia training data

```
> set.seed(20081031)
> golub.beg<- rknnBeg(golub.train, golub.train.cl);
> plot(golub.beg)
```

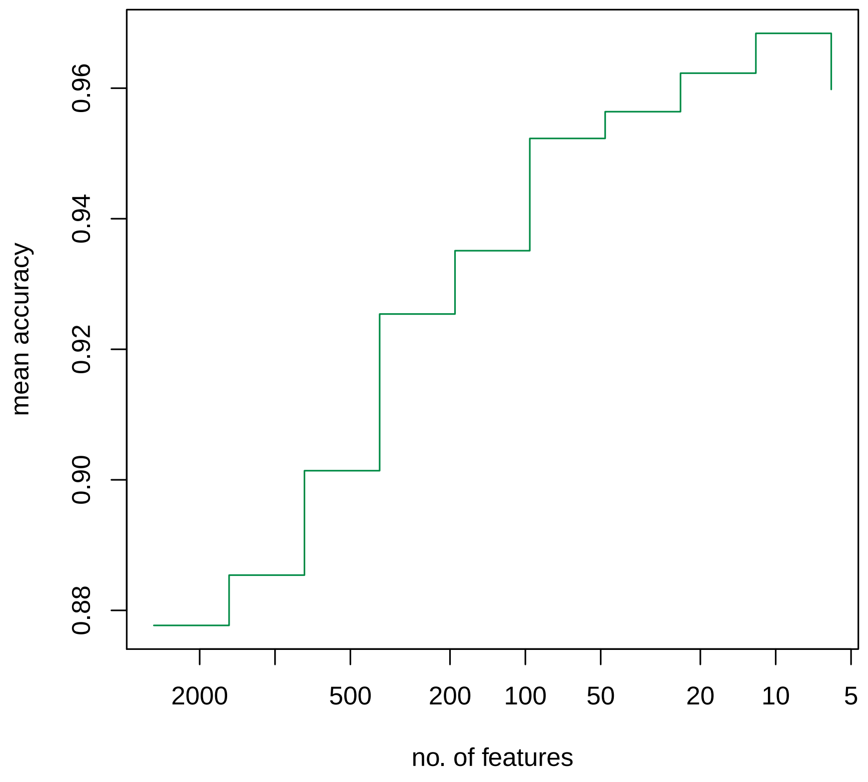

Figure 3: Mean accuracy change with the number of features for Golub leukemia data in first stage

```
> better.set<- prebestset(golub.beg);
> golub.bel<- rknnBel(golub.train[,better.set], golub.train.cl);
> plot(golub.bel, ylim=c(0.88, 1))
```

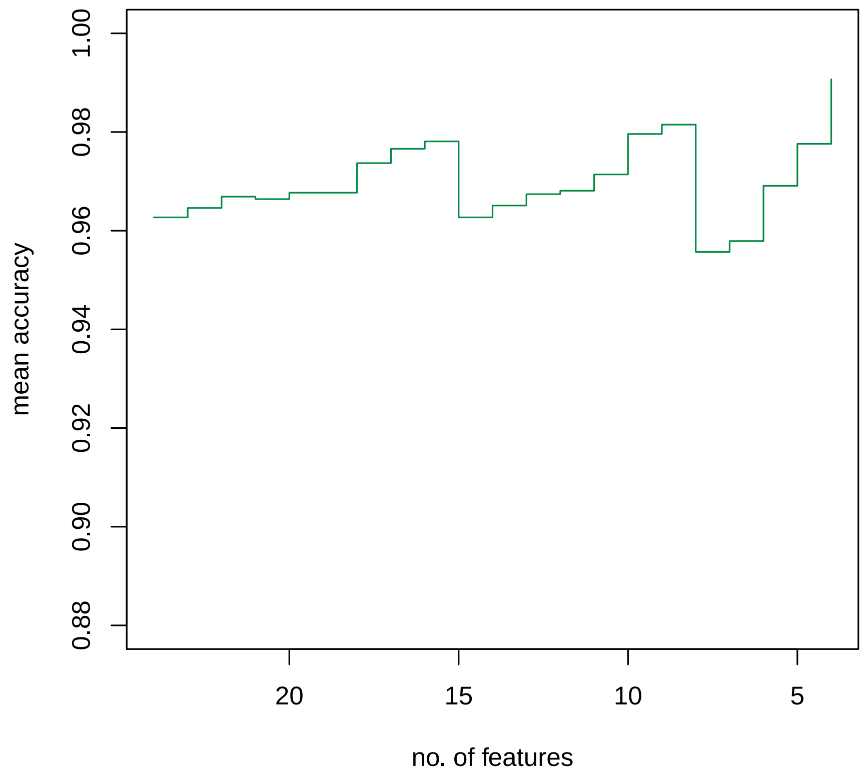

Figure 4: Mean accuracy change with the number of features for Golub leukemia data in second stage

4 Regression

```
> data(PAC)
> x<- scale(PAC$X);
> PAC.beg<- rknnBeg(data=x, y=PAC$y, k=3, r=500, pk=0.8)
> plot(PAC.beg)
> knn.reg(x[,bestset(PAC.beg)],  y=PAC$y, k=3)

PRESS =  37912.79
R2-Predict =  0.972057

>
```

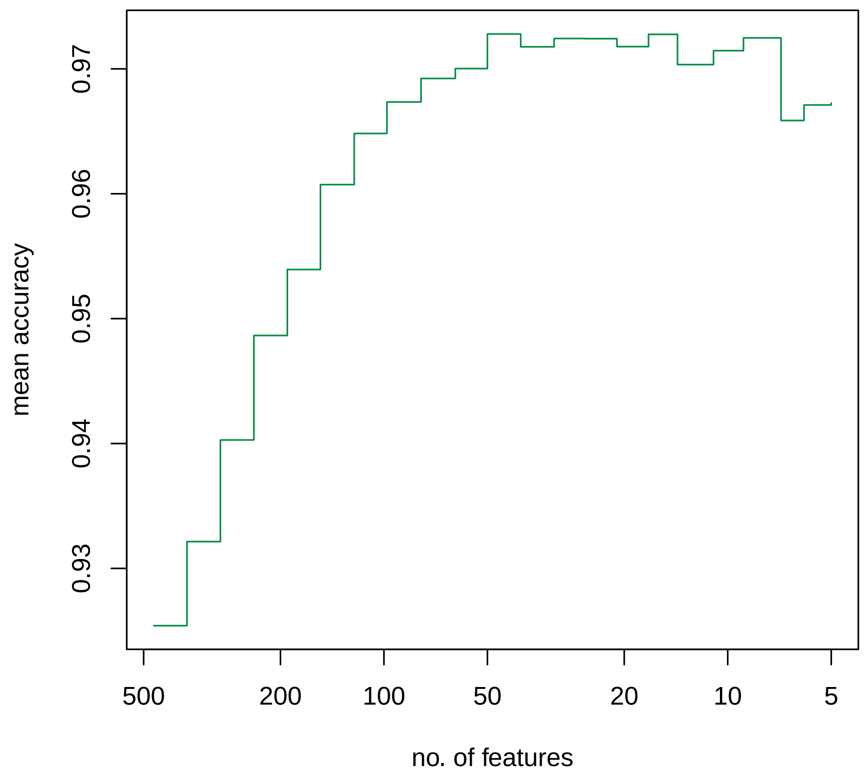

Figure 5: Mean accuracy change with the number of features for PAC data in second stage