Anomaly Detection
One-Class SVM

Stephen O'Connell

August 12, 2011

Anomaly Detection

- Detecting "samples" that don't fit, the outliers
- Many uses - credit card, loan app, sensor monitoring, network monitoring
- There are different approaches
- Form of classification, only there is one-class
- Model is trained on data relating to only "TRUE" conditions
- Testing models can be difficult, especially as dimensions increase

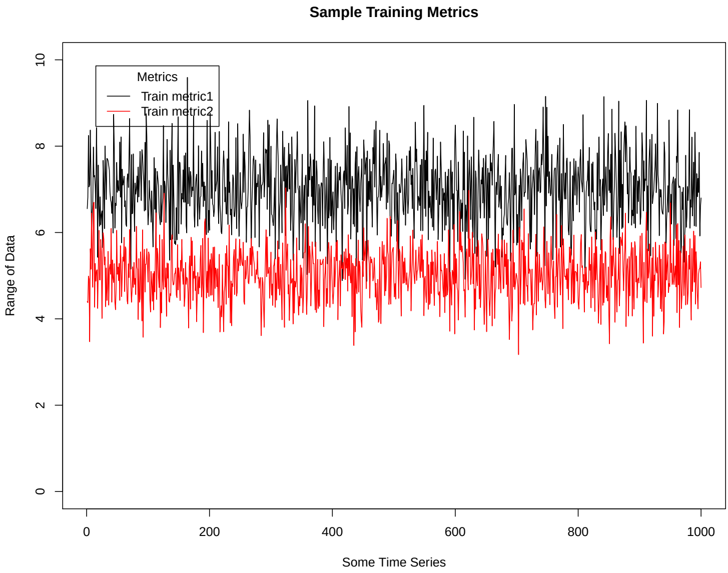
Example: One-Class SVM

- Feature space size = 2
- Training and test data is generated
- Build an SVM with default parameters
- Measure results - False Positives
- Tune the model and measure results
- Test model and measure results

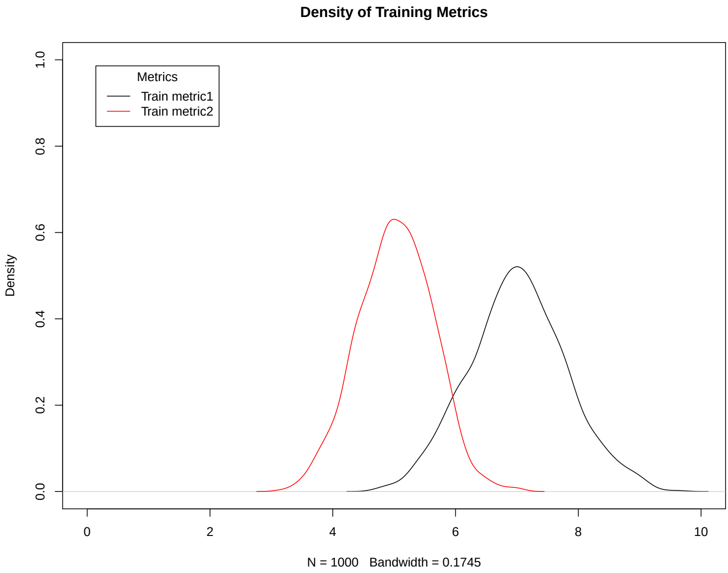
Create Training Data

```
##------------------------------------------------------------------------
## CREATE TRAINING DATA

N <- 1000

x1 <- rnorm(N, mean=7, sd=.8)
x2 <- rnorm(N, mean=5, sd=.6)

train <- data.frame(x1, x2)
```

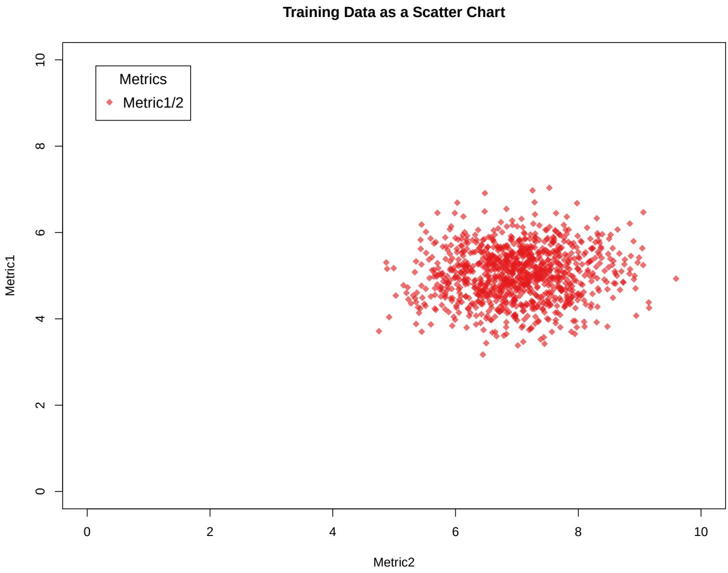
Training Data



Training Data - Density



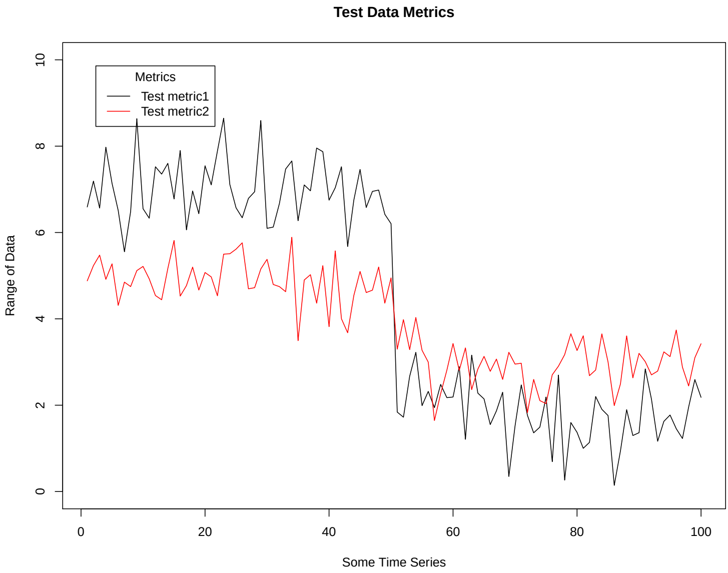
Training Data - Scatter



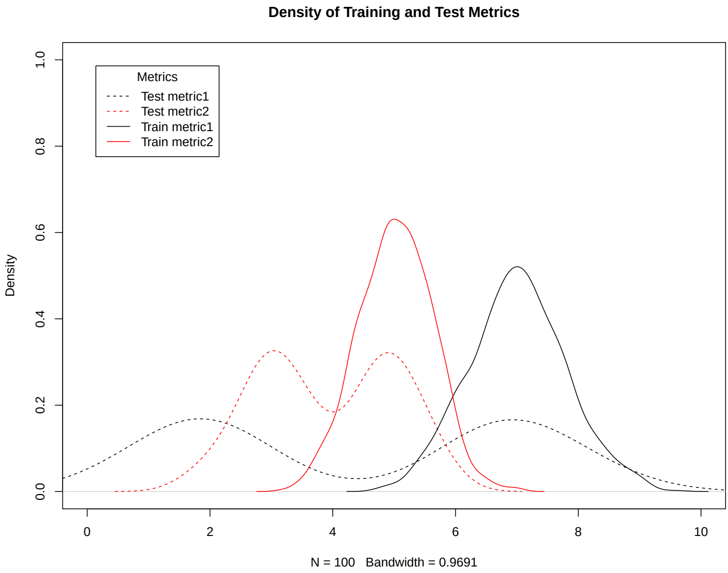
Create Test Data

```r
##-----------------------------------------------------------------------
## CREATE TEST DATA
N_test <- 100

T1 <- c(rnorm(N_test/2, mean=7, sd=.8),
        rnorm(N_test/2, mean=2, sd=.8))

T2 <- c(rnorm(N_test/2, mean=5, sd=.6),
        rnorm(N_test/2, mean=3, sd=.6))

test <- data.frame(T1,T2)

## GROUND TRUTH
testGroundTruth <- c(rep(TRUE, N_test/2), rep(FALSE, N_test/2))
```

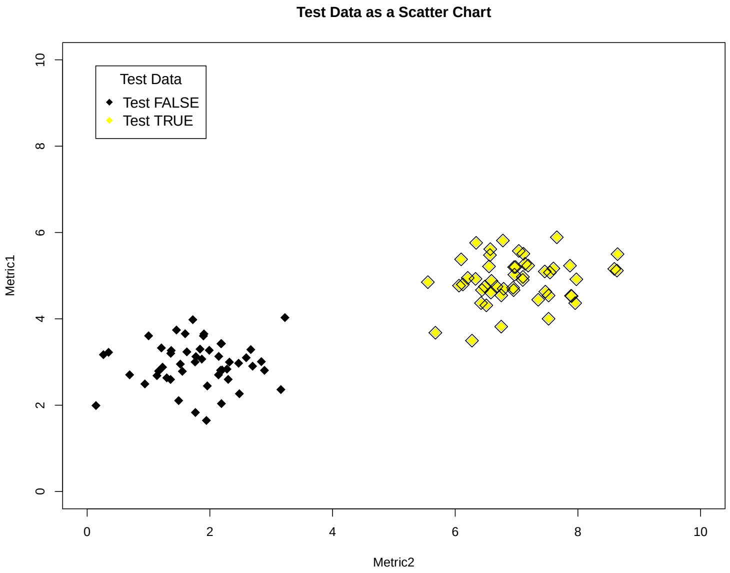
Test Data


Test Data Metrics
Metrics
Test metric1
Test metric2
Range of Data
0
2
4
6
8
10
Some Time Series
0
20
40
60
80
100

Training and Test - Density



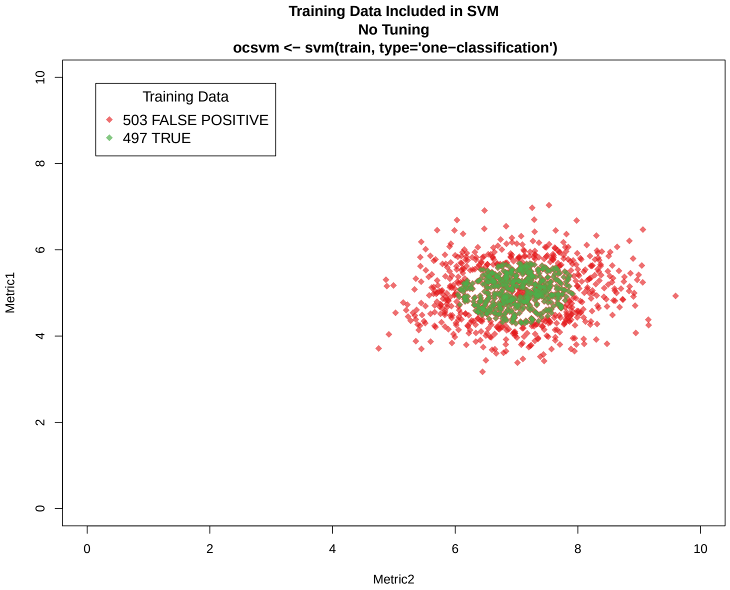
Test Data - Scatter



One-Class SVM, Default Parameters

```
##-------------------------------------------------------------------
## TRAIN THE MODEL

ocsvm <- svm(train, type='one-classification')


train_indx <- predict(ocsvm, train)


table(train_indx, rep(TRUE, N))


## PERFORMANCE:

train_indx TRUE
     FALSE  503
     TRUE   497
```

One-Class SVM - No Tuning



Training Data Included in SVM
No Tuning
ocsvm <− svm(train, type='one−classification')

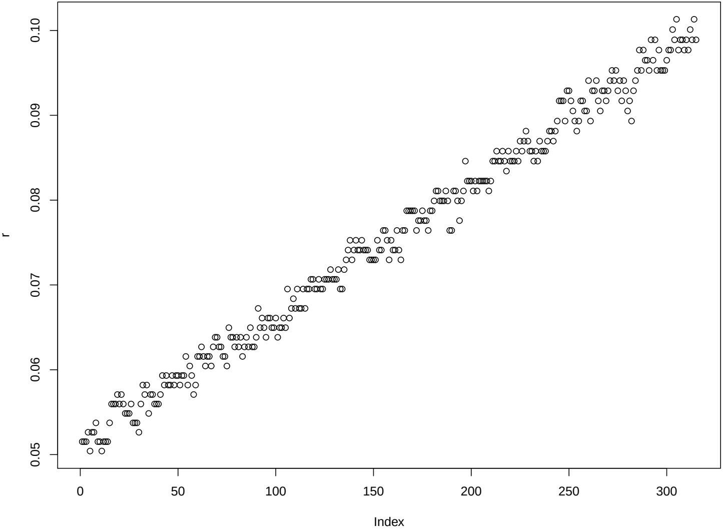
One-Class SVM Tuning

Grid Search – Experiment

```
##----------------------------------------------------------------------
## TUNING THE MODEL
results <- list()
for (nu in seq(from=.05, to=.09, by=.002)) {
    for (gamma in seq(from=.12, to=.19, by=.005)) {
        key <- paste("nu_", as.character(nu), "_gamma_", as.character(gamma), sep='')
        ocsvm <- svm(train, type='one-classification', nu=nu, gamma=gamma)
        train_indx <- predict(ocsvm, train)
        t <- table(train_indx, rep(TRUE, N))
        results[key] <- t[1] / t[2]
    }
}

> r <- do.call("rbind",results)
> dimnames(r)[[1]][which.min(r)]
 [1] "nu_0.05_gamma_0.14"

> min(r)
 [1] 0.05042017
```

Careful not to over fit.

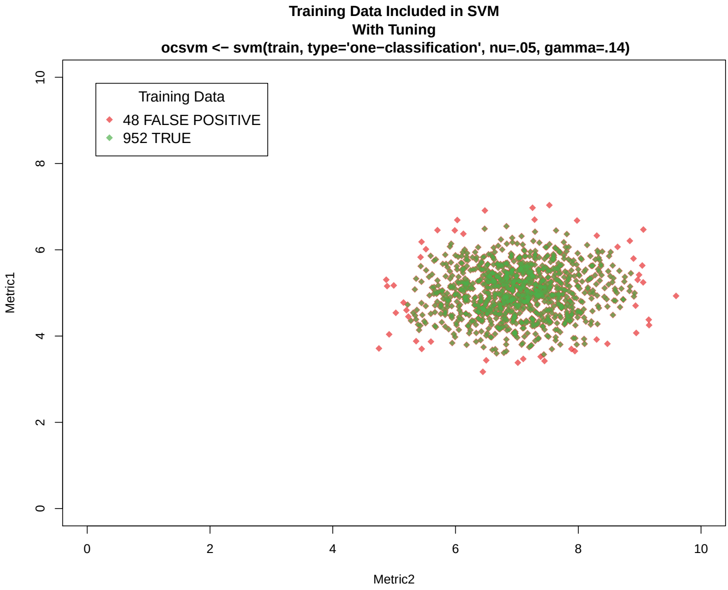
One-Class SVM - Tuning



One-Class SVM, Tuned Parameters

```
##------------------------------------------------------------------
## TRAIN THE MODEL

ocsvm <- svm(train, type='one-classification', nu=.05, gamma=.14)

train_indx <- predict(ocsvm, train)

table(train_indx, rep(TRUE, N))


## PERFORMANCE:

train_indx TRUE
     FALSE   48
     TRUE   952
```

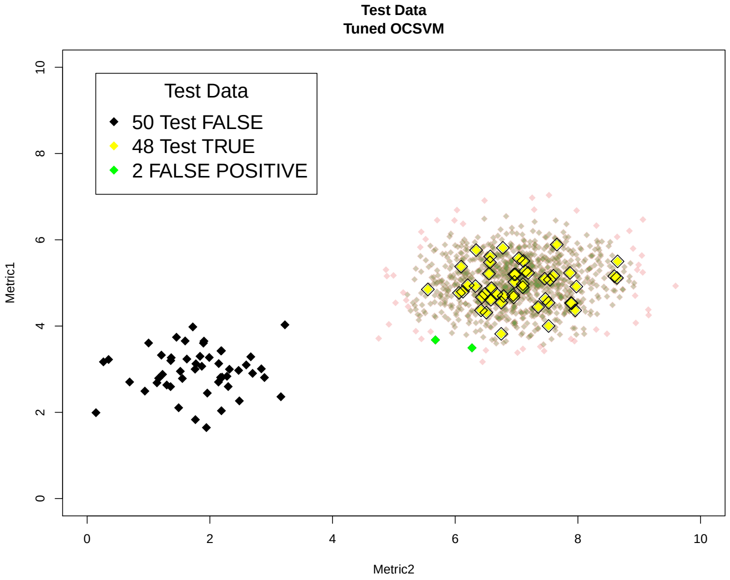
One-Class SVM - Tuned



One-Class SVM - Testing

```
##-------------------------------------------------------------------------
## TESTING THE MODEL

test_indx <- predict(ocsvm, test)

table(test_indx, testGroundTruth)


## PERFORMANCE:

         testGroundTruth
test_indx FALSE TRUE
    FALSE    50    2
    TRUE      0   48
```

One-Class SVM - Test



Deployment Considerations

- High dimensional data - can be hard to build and test, and training time can become an issue
- Training Data: hours/days/weeks/months/years ?
- Time of day, day of week - could have multiple levels of models
- Cycle time for running the model - data stream processing would be best
- Volume of data, distribution of data
- Model Maintenance - anomalies or a new TRUTH?

Thank You!

E-mail:	sao@saoconnell.com
Phone:	925-330-4350

Example code and slides available: ??