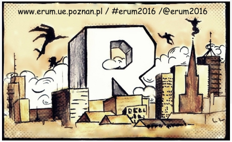
Unattended SVM parameters fitting for monitoring nonlinear profiles

Emilio L. Cano (University of Castilla-La Mancha)
Javier M. Moguerza (Rey Juan Carlos University)
Mariano Prieto (ENUSA Industrias Avanzadas)

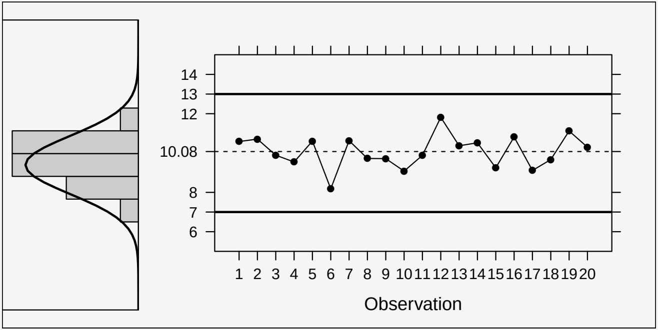


ELCano

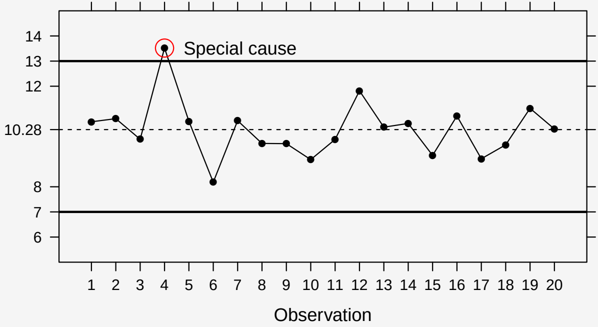
Statistical Process Control

Assignable causes of variation may be found and eliminated
Walter A. Shewhart

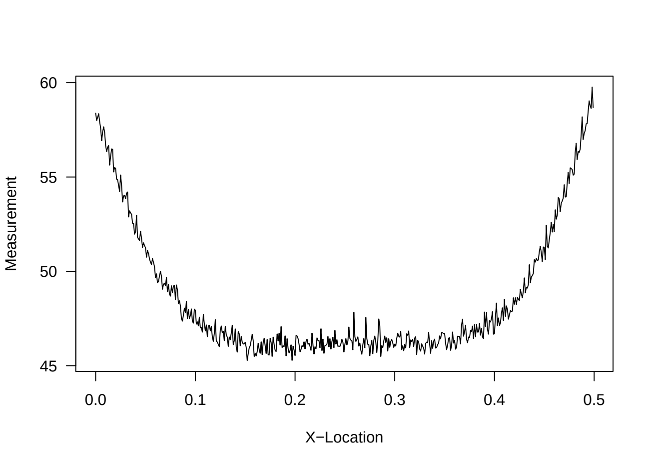
Statistical Process Control (cont.)



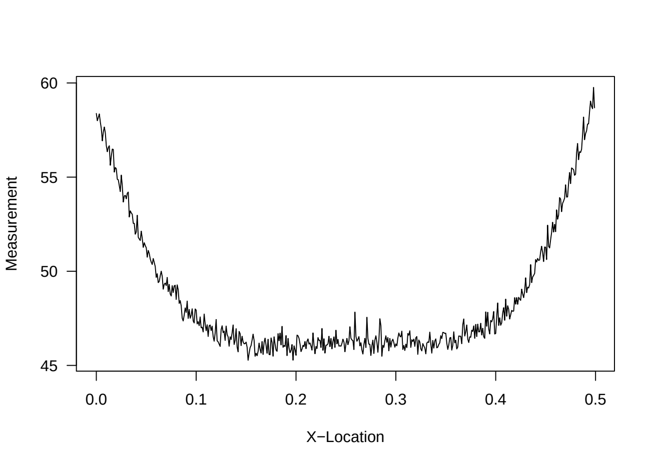
Statistical Process Control (cont.)



Nonlinear profiles



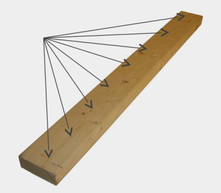
Nonlinear profiles



One function per sample (instead of a data point)

Illustrative example

- Engineered woodboards
- Data set of 50 boards
- Sample of 5 boards per shift



- Quality characteristic: density
- Total measurements: 500
- Every 0.001 in along the board

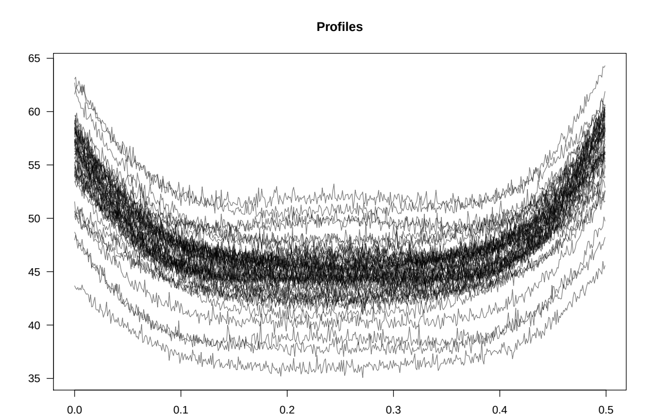
Illustrative example (cont.)

Data

```
library(SixSigma)
str(ss.data.wbx)

##  num [1:500] 0 0.001 0.002 0.003 0.004 0.005 0.006 0.007

str(ss.data.wby)

##  num [1:500, 1:50] 58.4 58 58.2 58.4 57.9 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:50] "P1" "P2" "P3" "P4" ...
```

Illustrative example (cont.)

```
plotProfiles(profiles = ss.data.wby,
    x = ss.data.wbx)
```



Support Vector Machines (SVM)

SVM regression model

Given response $\mathbf{y}$, input space $\mathbf{x}$:

$$\mathbf{y} = r(\mathbf{x}) + \delta$$

$r(\mathbf{x})$ feature space (higher dimension than $\mathbf{x}$)

$r(\mathbf{x})$: non-linear, high dimensional transformation of $\mathbf{x}$ input vector
Then, a linear combination over the feature space is the prediction model:

$$f(\mathbf{x}, \omega) = \sum_j \omega_j, \mathbf{g}_j(\mathbf{x})$$

Support Vector Machines (SVM) (cont.)

SVM regression parameters

Regression estimates are obtained minimizing the ε-intensive loss function. This function contains two input parameters: ε and C (regularization parameter)

Details: *Vapnik, V. (1998). Statistical learning theory. New York: Wiley; Vapnik, V. (1999). The nature of statistical learning theory (2nd ed). Berlin: Springer.*

Support Vector Machines (SVM) (cont.)

Parameters selection

- C trade off between model complexity and deviations larger than ε in optimization
- ε controls the width of the ε-insensitive zone
- Several practical approaches (cross validation, experts opinion, $\ldots$)

(unattended) Parameters selection

Regularization parameter C

$$C = \max\{|\overline{y} + 3\sigma_y|, |\overline{y} - 3\sigma_y|\}$$

```
max(c(abs(mean(y) + 3*sd(y)), abs(mean(y) - 3*sd(y))))
```

ε parameter

$$\varepsilon = 3\sigma\sqrt{\frac{\log n}{n}}$$

```
3*par.sigma*sqrt(log(nrowprofiles)/nrowprofiles)
```

(unattended) Parameters selection (cont.)

Input noise level σ

An approximation using polynomials

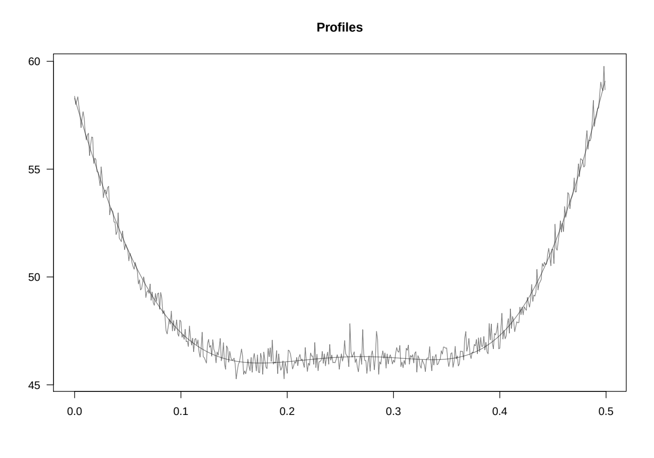
```
mloess <- loess(y ~ x)
yhat <- predict(mloess, newdata = x)
deltas <- y - yhat
par.sigma <- sd(deltas)
```

Details: *Cherkassky, V and Ma, Y (2004). Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks, **17**(1), 113-126*

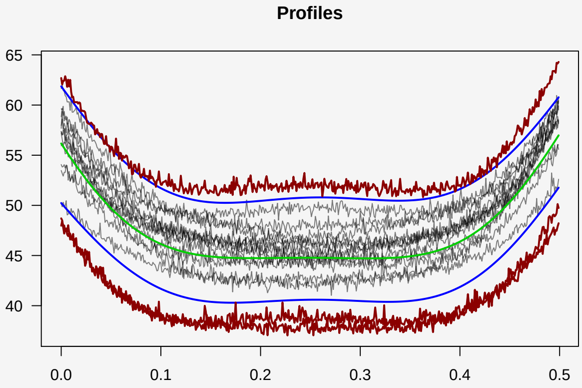
Regularization of nonlinear profiles via SVM

```
P1.smooth <- smoothProfiles(
    profiles = ss.data.wby[, "P1"],
    x = ss.data.wbx)
plotProfiles(profiles = cbind(P1.smooth,
        ss.data.wby[, "P1"]),
    x = ss.data.wbx)
```

Regularization of nonlinear profiles via SVM (cont.)



Smoothed prototype and confidence bands



Smoothed prototype and confidence bands (cont.)

```
wby.phase1 <- ss.data.wby[, 1:35]
wb.limits <- climProfiles(profiles = wby.phase1[, -28],
    x = ss.data.wbx,
    smoothprof = TRUE,
    smoothlim = TRUE)
wby.phase2 <- ss.data.wby[, 36:50]
wb.out.phase2 <- outProfiles(profiles = wby.phase2,
    x = ss.data.wbx,
    cLimits = wb.limits,
    tol = 0.8)
plotProfiles(wby.phase2,
    x = ss.data.wbx,
    cLimits = wb.limits,
    outControl = wb.out.phase2$idOut,
    onlyout = FALSE)
```

Questions

Thanks!

emilio.lcano@uclm.es