Hands on cusp package tutorial

Raoul P. P. P. Grasman

November 5, 2009

1 Introduction

The cusp package provides routines for fitting a cusp catastrophe model as suggested by (Cobb, 1978). The full documentation of the package can be found in the vignette "Fitting the Cusp Catastrophe in R: A **cusp**-Package Primer" included with the package.

2 Getting started

Load the library with the statement

```
> library(cusp)
```

3 The Zeeman data

We analyze three data sets that were obtained with three different instances of a Zeeman catastrophe machine. This device consists of a wheel is tethered by an elastic chord to a fixed point. For sake of reference let the *central axis* be defined as the axis parallel to the line between the fixed point and the center of the wheel. Another elastic is also attached to the wheel on one end, while the other end is moved about in the 'control plane' area opposite to the fixed point. The shortest distance between the strap point on the wheel and the central axis is recorded as a function of the position in the control plane. (In the original machine the angle between this axis and the line through the wheel center and the strap point is used.) See http://www.math.sunysb.edu/ tony/whatsnew/column/catastrophe-0600/cusp4.html for a vivid demonstration. The behavior of the Zeeman catastrophe machine is archetypal for the deterministic Cusp Catastrophe.

The data sets were obtained from 3 different physical instances of this machine, made by different people.

Measurements from this machine will be made with measurement errors, e.g., due to parallax, friction, writing mistakes etc. The measurements will fit the following model, the

Figure 1: Catastrophe machine that was used to collect the data in `zeeman2`

we will dub the measurement error model,

$$y_i = z_i + \epsilon_i, \tag{1}$$

where ϵ_i is a zero mean random variable, e.g., $\epsilon \sim N(0, \sigma^2)$ for some σ^2, and z_i is one of the extremal real roots of the cusp catastrophe equation

$$\alpha_i + \beta_i z - z^3 = 0.$$

Note that this is quite different from Cobb's conceptualization of the stochastic cusp catastrophe, in which

$$y_i \sim \Psi \, e^{\alpha_i y + \beta_i y^2 - \frac{1}{4} y^4}.$$

where the right hand side is the state density of the *stochastic* differential equation

$$dY = (\alpha_i + \beta_i z - z^3)dt + dW(t), \tag{2}$$

where $W(t)$ is a Wiener process.

The cusp package is intended for the latter model (i.e., for Cobb's stochastic catastrophe model). However, the data from the Zeeman catastrophe machine below show that it is also quite suitable for the former model.

3.1 `zeeman1`

The first data set

```
> data(zeeman1)
> nrow(zeeman1)
```

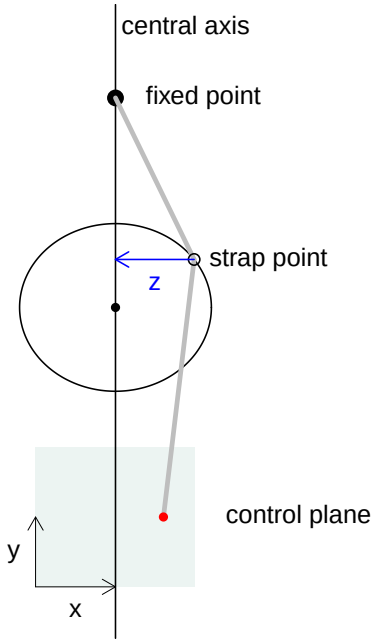

Figure 2: Catastrophe machine used for the data set `zeeman1` (left). Schematic Zeeman machine and the variables available in the data frames (right).

```
[1] 150
```

```
> head(zeeman1)
```

```
   x y    z
1 -7 2 -2.0
2 -6 2 -1.8
3 -5 2 -1.8
4 -4 2 -1.3
5 -3 2 -0.8
6 -2 2 -0.5
```

consists of 150 observations from experiments with the machine.

The columns of `zeeman1` respectively contain the value on the asymmetry axis which is orthogonal to the central axis (`x`), the value of the bifurcation variable which is paralel to the central axis (`y`), and state variable, the shortest distance from the wheel strap point to the central axis. Figure 2 displays these coordinates.

While thus the asymmetry and bifurcation variables are known, namely `x` and `y` respectively, we fit the model where each parameter is regressed on both of the control variables:

```
> fit1.1 = cusp(y ~ z, alpha ~ x + y, beta ~ x + y, zeeman1)
> summary(fit1.1)
```

```
Call:
cusp(formula = y ~ z, alpha = alpha ~ x + y, beta = beta ~ x +
    y, data = zeeman1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.26705  -0.11605  -0.01376   0.10961   2.04443

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
a[(Intercept)]  0.21469    0.25516   0.841  0.40012
a[x]            1.15341    0.15558   7.414 1.23e-13 ***
a[y]           -0.16460    0.11996  -1.372  0.17002
b[(Intercept)]  1.04763    0.37401   2.801  0.00509 **
b[x]            0.02851    0.14058   0.203  0.83927
b[y]           -1.48202    0.13634 -10.870  < 2e-16 ***
w[(Intercept)] -0.02387    0.13178  -0.181  0.85625
w[z]            0.90316    0.02723  33.168  < 2e-16 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance: 539.492  on 149  degrees of freedom
Linear deviance:  85.233  on 146  degrees of freedom
Logist deviance:      NA  on  NA  degrees of freedom
 Delay deviance:  24.228  on 142  degrees of freedom

             R.Squared     logLik npar       AIC      AICc       BIC
Linear model 0.8711298 -170.44770    4  348.8954  349.1713  360.9379
Cusp model   0.9554786   76.23194    8 -136.4639 -135.4426 -112.3788
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.

        Chi-square test of linear vs. cusp model

X-squared = 493.4, df = 4, p-value = 0

Number of optimization iterations: 72
```

The summary table indicates that the regression coefficients `a[(Intercept)]`, `a[y]`, `b[x]`, and `w[(Intercept)]` are non-significant suggesting that they are not required in the model. We therefore fit the model again, leaving these regressors out of the model:

```
> fit1.2 = cusp(y ~ z - 1, alpha ~ x - 1, beta ~ y, zeeman1)
> summary(fit1.2)

Call:
cusp(formula = y ~ z - 1, alpha = alpha ~ x - 1, beta = beta ~
    y, data = zeeman1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.15809  -0.04248   0.04943   0.16451   2.11321

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
a[x]            1.15341    0.15540   7.422 1.15e-13 ***
b[(Intercept)]  0.96994    0.37551   2.583   0.0098 **
b[y]           -1.47496    0.13656 -10.801  < 2e-16 ***
w[z]            0.89487    0.02723  32.862  < 2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance: 529.634  on 149  degrees of freedom
Linear deviance:  85.233  on 146  degrees of freedom
Logist deviance:      NA  on  NA  degrees of freedom
 Delay deviance:  23.365  on 146  degrees of freedom

             R.Squared    logLik npar       AIC      AICc       BIC
Linear model 0.8716421 -170.4477    4  348.8954  349.1713  360.9379
Cusp model   0.9568910   70.7295    4 -133.4590 -133.1831 -121.4165
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.

        Chi-square test of linear vs. cusp model

X-squared = 482.4, df = 0, p-value = 0

Number of optimization iterations: 18
```

Now all the regression coefficients 'are significant'. If we turn attention to the information criteria however, we see that the AIC for the first model (-136.464) is smaller than the AIC for the second model (-133.459). The BIC on the other hand, is larger for the first model (-112.379) compared to the second model (-133.459). Because in the first model especially the intercept for that state model equation is quite small (`w[(Intercept)]` = -0.0239), we relax the second model a bit by allowing for intercept in model equation for α to see if the AIC is smaller

```
> fit1.3 = cusp(y ~ z - 1, alpha ~ x, beta ~ y, zeeman1)
> (sf1.3 <- summary(fit1.3, logist = TRUE))

Call:
cusp(formula = y ~ z - 1, alpha = alpha ~ x, beta = beta ~ y,
    data = zeeman1)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-2.21248  -0.10346   0.00771   0.12443   2.07045

Coefficients:
```

```
               Estimate Std. Error z value Pr(>|z|)
a[(Intercept)]  0.45555    0.17266   2.638  0.00833 **
a[x]            1.15341    0.15471   7.455 8.98e-14 ***
b[(Intercept)]  0.99190    0.37313   2.658  0.00785 **
b[y]           -1.47891    0.13597 -10.877  < 2e-16 ***
w[z]            0.89766    0.02711  33.114  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance: 532.944  on 149  degrees of freedom
Linear deviance:  85.233  on 146  degrees of freedom
Logist deviance:  17.061  on 146  degrees of freedom
 Delay deviance:  24.683  on 145  degrees of freedom

             R.Squared     logLik npar       AIC      AICc       BIC
Linear model 0.8716421 -170.44770    4  348.8954  349.1713  360.9379
Logist model 0.9742049  -49.80112    4  107.6022  107.8781  119.6448
Cusp model   0.9540613   74.72029    5 -139.4406 -139.0239 -124.3874
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.

        Chi-square test of linear vs. cusp model

X-squared = 490.3, df = 1, p-value = 0

Number of optimization iterations: 31
```

The AIC, AICc and BIC are now all smaller than both the first and the second model, and all intercept coefficients 'are significant'.

In the call to `summary`, this time we have set the optional parameter `logist` to `TRUE`, so that also the logistic model is fitted for a more critical test of the cusp model (see the main vignette and `help(cusp.logist)` for details). The information criteria all indicate that the cusp model is to be preferred over the logistic model, even though the *pseudo-*R^2 is larger for the logistic model (0.9742) than for the cusp model (0.9541). (See for a discussion of the *pseudo-*R^2 the main vignette.)

Figure 3 displays a diagnostic plot of the last model that was generated with

```
> plot(fit1.3)
```

```
> plot(fit1.3)
```

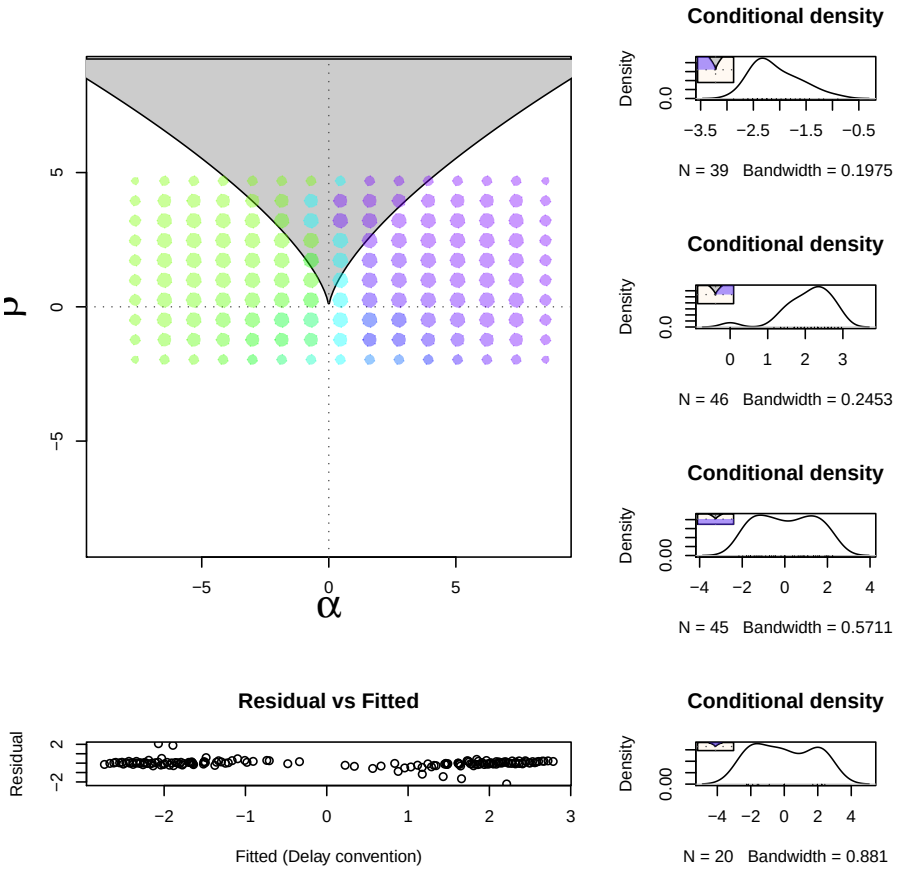

Figure 3: Diagnostic plot for the second model fitted to `zeeman1`.

3.2 zeeman2

We fit the same models as in the previous case

```
> data(zeeman2)
> fit2.1 = cusp(y ~ z, alpha ~ x + y, beta ~ x + y, zeeman2)
> summary(fit2.1)

Call:
cusp(formula = y ~ z, alpha = alpha ~ x + y, beta = beta ~ x +
    y, data = zeeman2)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.589e+00  -1.839e-01  -3.242e-06   1.839e-01   1.589e+00

Coefficients:
                 Estimate Std. Error  z value Pr(>|z|)
a[(Intercept)] -1.285e-04  2.084e-01   -0.001  0.99951
a[x]            7.892e-01  9.924e-02    7.952 1.84e-15 ***
a[y]            1.374e-05  4.994e-02 2.75e-04  0.99978
b[(Intercept)]  3.891e+00  2.864e-01   13.588  < 2e-16 ***
b[x]            6.326e-06  4.585e-02 1.38e-04  0.99989
b[y]            1.007e-01  3.448e-02    2.921  0.00349 **
w[(Intercept)]  6.862e-07  4.799e-02 1.43e-05  0.99999
w[z]            7.789e-01  1.791e-02   43.495  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance: 1100.103  on 197  degrees of freedom
Linear deviance:  426.846  on 194  degrees of freedom
Logist deviance:       NA  on  NA  degrees of freedom
 Delay deviance:   25.421  on 190  degrees of freedom

             R.Squared    logLik npar       AIC      AICc       BIC
Linear model 0.7645990 -356.9972    4  721.9944  722.2016  735.1475
Cusp model   0.9770091  121.4310    8 -226.8621 -226.1002 -200.5559
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.
```

```
        Chi-square test of linear vs. cusp model

X-squared = 956.9, df = 4, p-value = 0

Number of optimization iterations: 96
```

Again, the intercept coefficients `a[(Intercept)]` and `w[(Intercept)]`) are non-significant, as are `a[y]` and `b[x]`. We leave these coefficients out of the model in the followup fit

```
> fit2.2 = cusp(y ~ z - 1, alpha ~ x - 1, beta ~ y, zeeman2)
> summary(fit2.2)

Call:
cusp(formula = y ~ z - 1, alpha = alpha ~ x - 1, beta = beta ~
    y, data = zeeman2)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-1.589e+00  -1.839e-01   3.331e-15   1.839e-01   1.589e+00

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
a[x]            0.78913    0.09924   7.952 1.84e-15 ***
b[(Intercept)]  3.89115    0.28636  13.588  < 2e-16 ***
b[y]            0.10071    0.03448   2.921  0.00349 **
w[z]            0.77891    0.01791  43.495  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance: 1100.104  on 197  degrees of freedom
Linear deviance:  426.846  on 194  degrees of freedom
Logist deviance:       NA  on  NA  degrees of freedom
 Delay deviance:   25.421  on 194  degrees of freedom

             R.Squared    logLik npar       AIC      AICc       BIC
Linear model 0.7645990 -356.9972    4  721.9944  722.2016  735.1475
Cusp model   0.9770089  121.4310    4 -234.8621 -234.6548 -221.7090
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.
```

```
        Chi-square test of linear vs. cusp model

X-squared = 956.9, df = 0, p-value = 0

Number of optimization iterations: 34
```

The AIC of this reduced model (-234.8621) is indeed smaller than the AIC of the previous model (-226.8621). If for comparison with the model selected for `zeeman1` we include `a[(Intercept)]` again, this barely reduces the negative log-likelihood, while increasing the AIC. More importantly this coefficient is estimated to be zero.

```
> fit2.3 = cusp(y ~ z - 1, alpha ~ x, beta ~ y, zeeman2)
> fit2.3

Call:  cusp(formula = y ~ z - 1, alpha = alpha ~ x, beta = beta ~ y,      data = zeeman2)

Coefficients:
a[(Intercept)]            a[x]  b[(Intercept)]            b[y]            w[z]
    -5.934e-18       7.891e-01       3.891e+00       1.007e-01       7.789e-01

Degrees of Freedom: 197 Total (i.e. Null);  193 Residual
Null Deviance:            1100
Delay Deviance:         25.42         AIC: -232.9
```

3.3 zeeman3

We do the same analysis for `zeeman3`.

```
> data(zeeman3)
> fit3.1 <- cusp(y ~ z, alpha ~ x + y, beta ~ x + y, zeeman3)
> summary(fit3.1)

Call:
cusp(formula = y ~ z, alpha = alpha ~ x + y, beta = beta ~ x +
    y, data = zeeman3)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.28904  -0.14995   0.03348   0.17721   2.44220

Coefficients:
                Estimate Std. Error  z value Pr(>|z|)
```

```
a[(Intercept)] -0.0540490  0.0033964  -15.914   <2e-16 ***
a[x]            0.2437644  0.0006382  381.935   <2e-16 ***
a[y]           -0.0185760  0.0008130  -22.848   <2e-16 ***
b[(Intercept)]  4.8832846  0.0034005 1436.034   <2e-16 ***
b[x]           -0.0454644  0.0006383  -71.223   <2e-16 ***
b[y]           -0.6766923  0.0008136 -831.715   <2e-16 ***
w[(Intercept)] -0.0564834  0.0219623   -2.572   0.0101 *
w[z]            0.4664489  0.0044566  104.666   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance:  987.386  on 281  degrees of freedom
Linear deviance: 2016.486  on 278  degrees of freedom
Logist deviance:       NA  on  NA  degrees of freedom
 Delay deviance:   55.197  on 274  degrees of freedom

             R.Squared     logLik npar        AIC     AICc       BIC
Linear model 0.5556587 -677.51652    4 1363.03304 1363.177 1377.6007
Cusp model   0.9451417  -34.08026    8   84.16052   84.688  113.2958
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.

        Chi-square test of linear vs. cusp model

X-squared = 1287, df = 4, p-value = 0

Number of optimization iterations: 114
```

This time all regression coefficients are significant. We still compare with the simplified model used in the other cases to see if the AIC, AICc and BIC might be lower still

```
> fit3.2 <- cusp(y ~ z - 1, alpha ~ x - 1, beta ~ y, zeeman3)
> summary(fit3.2)

Call:
cusp(formula = y ~ z - 1, alpha = alpha ~ x - 1, beta = beta ~
    y, data = zeeman3)

Deviance Residuals:
      Min         1Q     Median         3Q        Max
```

```
-2.496125  -0.182561  -0.005285   0.158835   1.054552

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
a[x]            0.24273    0.02918   8.317   <2e-16 ***
b[(Intercept)]  4.87001    0.25664  18.976   <2e-16 ***
b[y]           -0.68031    0.04054 -16.781   <2e-16 ***
w[z]            0.46586    0.01051  44.334   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance:  984.900  on 281  degrees of freedom
Linear deviance: 2016.486  on 278  degrees of freedom
Logist deviance:       NA  on  NA  degrees of freedom
 Delay deviance:   56.263  on 278  degrees of freedom

             R.Squared     logLik npar        AIC      AICc        BIC
Linear model 0.5563031 -677.51652    4 1363.03304 1363.1774 1377.60067
Cusp model   0.9445425  -36.21439    4   80.42879   80.5732   94.99642
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.

        Chi-square test of linear vs. cusp model

X-squared = 1283, df = 0, p-value = 0

Number of optimization iterations: 25
```

Compared to the previous fit, this model indeed yields a lower AIC, AICc, and BIC and hence is preferred according to the information criteria.

4 Simulation

Sometimes it's useful to be able to simulate data from a model. Two cases are discussed here.

4.1 Generating data in accordance with Cobb's cusp SDE

To obtain observations in accordance with Cobb's stochastic cusp catastrophe model in equation (2), the function `rcusp` can be used. If for all the observations α and β are equal, this is simply done with the statement. Figure 4 displays a histogram.

```
> set.seed(423)
> alpha = 0.25
> beta = 2
> n = 1000
> y = rcusp(n, alpha, beta)
> hist(y, 80, freq = FALSE)
> curve(dcusp(x, 0.25, 2), min(y) - 1, max(y) + 1, add = TRUE,
+     col = 2)
```

The parameters α and β can simply be estimated with the statement

```
> cusp(y ~ y - 1, alpha ~ 1, beta ~ 1)

Call:  cusp(formula = y ~ y - 1, alpha = alpha ~ 1, beta = beta ~ 1)

Coefficients:
a[(Intercept)]  b[(Intercept)]            w[y]
        0.1916          2.0654          1.0081

Degrees of Freedom: 999 Total (i.e. Null);  997 Residual
Null Deviance:            1641
Delay Deviance:         361.2         AIC: 2278
```

If the α's and β's are different, e.g., depending on predictive variables, they can be generated for example with the statements

```
> x1 = runif(150)
> x2 = runif(150)
> a = c(-2, 4)
> b = c(-1, 4)
> alpha = a[1] + a[2] * x1
> beta = b[1] + b[2] * x2
> z = Vectorize(rcusp)(1, alpha, beta)
> data <- data.frame(x1, x2, z)
```

Estimating the coefficients in `a` and `b` can then be carried out with, e.g.,

```
> fit <- cusp(y ~ z, alpha ~ x1 + x2, beta ~ x1 + x2, data)
```

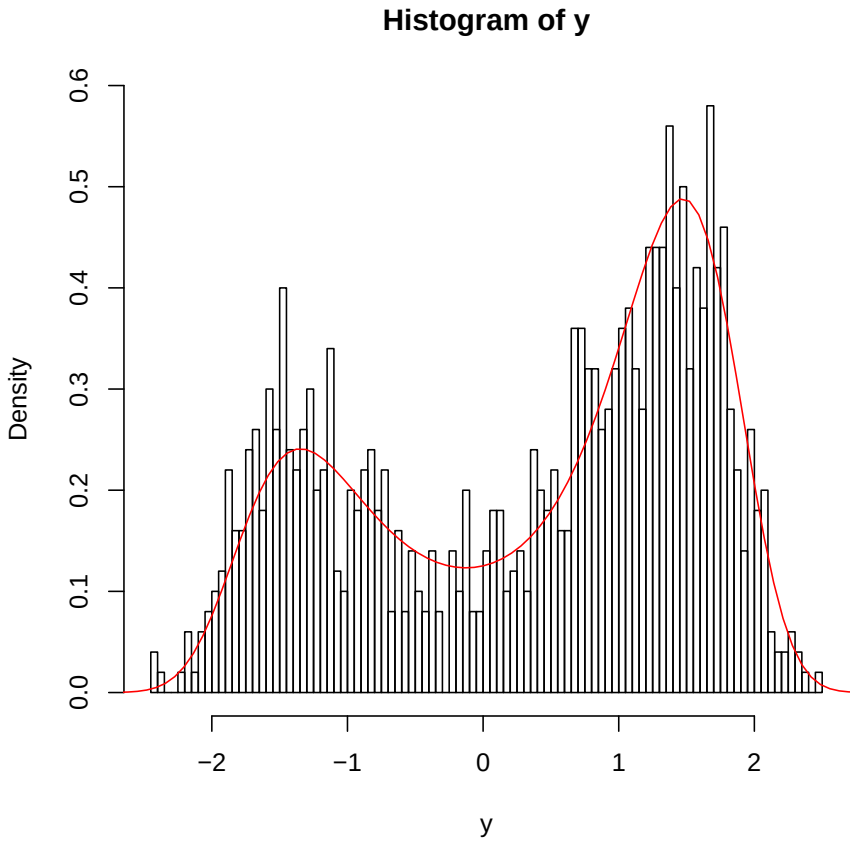

Figure 4: Histogram of 1000 observations from Cobb's stochastic cusp catastrophe SDE (as opposed to the deterministic cusp catastrophe) with for each observation $\alpha = 0.25$ and $\beta = 2$.

4.2 Generating data in accordance with the measurement error model

We can generate data in accordance with the measurement error model in equation (1) as follows

```
> set.seed(423)
> g = expand.grid(seq(-3, 3, len = 15), seq(-3, 3, len = 15))
> a = g[, 1]
> b = g[, 2]
> idx = cbind(sample(c(1, 3), length(a), TRUE), seq(along = a))
> s = Vectorize(cusp.extrema)(a, b)[idx]
> y = s + rnorm(length(s), , 0.3)
```

Here `g` is a grid of points on the control surface, and `cusp.extrema` is used to compute the roots of the cusp equilibrium equation

$$\alpha + \beta y^2 - y^3 = 0.$$

The vector of states `s` is then constructed by sampling from the smallest and largest roots. The observations `y` are then created in accordance with the measurement error model.

It instructive to see a 3D plot of the generated data points generated in this way

```
> if (require(onion)) {
+     p3d(a, b, y, theta = 200, phi = 10, zlim = c(-4, 4))
+ }
```

Note that this measurement model is *not* cusp SDE model of Cobb as described in the paper (the main vignette that comes with the cusp package). Even so, the maximum likelihood method of Cobb, does fit this surface quite well:

```
> fit <- cusp(y ~ y, alpha ~ a, beta ~ b)
> summary(fit)

Call:
cusp(formula = y ~ y, alpha = alpha ~ a, beta = beta ~ b)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.81307  -0.20738   0.02675   0.24115   1.05391

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
a[(Intercept)] -0.010426   0.147637  -0.071    0.944
a[a]            1.384265   0.162343   8.527   <2e-16 ***
```

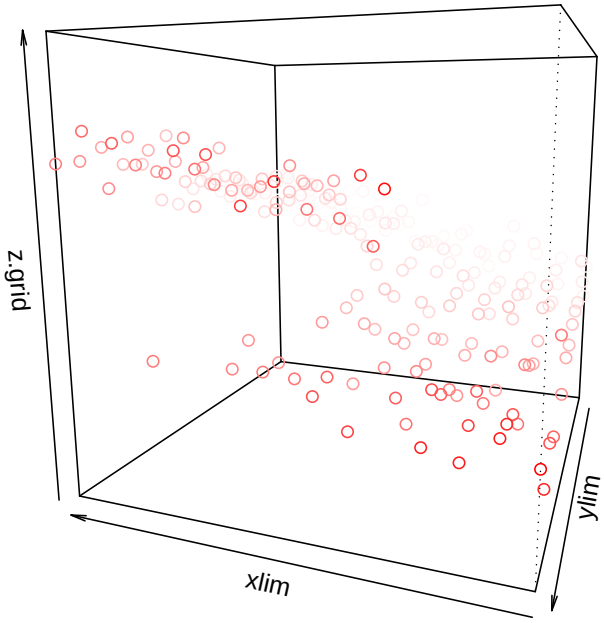

Figure 5: 3D scatter plot of cusp equilibrium surface with measurement error

```
b[(Intercept)]  0.296099   0.365037   0.811    0.417
b[b]            1.322078   0.095412  13.857   <2e-16 ***
w[(Intercept)]  0.004184   0.036282   0.115    0.908
w[y]            1.131029   0.034713  32.583   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


  Null deviance: 480.922  on 224  degrees of freedom
Linear deviance: 127.009  on 221  degrees of freedom
Logist deviance:      NA  on  NA  degrees of freedom
 Delay deviance:  26.036  on 219  degrees of freedom

             R.Squared     logLik npar      AIC     AICc      BIC
Linear model 0.6621645 -254.92849    4 517.8570 518.0388 531.5214
Cusp model   0.9462034  -62.04378    6 136.0876 136.4729 156.5842
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.

        Chi-square test of linear vs. cusp model

X-squared = 385.8, df = 2, p-value = 0

Number of optimization iterations: 43
```