

**rebmix**: Finite Mixture Modeling, Clustering & Classification

Marko Nagode

September 26, 2018

Abstract

The *rebmix* package provides R functions for random univariate and multivariate finite mixture model generation, estimation, clustering, latent class analysis and classification. Variables can be continuous, discrete, independent or dependent and may follow normal, lognormal, Weibull, gamma, binomial, Poisson, Dirac or von Mises parametric families.

1 Introduction

To cite the REBMIX algorithm please refer to (Nagode and Fajdiga, 2011a,b; Nagode, 2015, 2018). For theoretical backgrounds please upload also http://doi.org/10.5963/JAO0302001.

2 What’s new in version 2.10.3

Methods $Z_p$ and $\text{coef}$ are replaced by $a.Z_p$, $a.theta1.all$ and $a.theta2.all$ getters. All slots can now be accessed via accessors. Their names are generally composed of $a.$ followed by the slot name and are used to read the slots. Class "RGMIX.Theta" is added to simplify random finite mixture model generation. Method $\text{show}$ is added for "RCLS.chunk" class. Minimum number of component $cmin$ is added to REBMIX arguments and to the "REBMIX" class. The "Parzen window" preprocessing is renamed to more well known "kernel density estimation". Rough parameter estimation for binomial and Poisson parametric families is improved. The package is broadened to latent class analysis in version 2.10.3, too. Method $\text{split}$ is improved and examples for its proper use are added. GCC 8.1 notes and warnings in C++ functions are eliminated in version 2.10.2, too. Cholesky decomposition is now used to calculate logarithm of determinant and inverse of variance-covariance matrices instead of LU decomposition. Special attention is paid to resolving numerical problems related to high dimensional datasets. Version 2.10.1 is the further debugged version 2.10.0. Large $K$ in combination with large dimension $d$ can lead to histograms with numerous nonempty bins $v$. In order to restrain $v$, the well known RootN rule (Velleman, 1976) may intuitively be extended to multidimensions

$$v_{\text{max}} = \frac{1 + d}{d} n^{\frac{d}{d+1}}.$$  

(1)

If $d = \infty$, then $v_{\text{max}} = n$. If $d = 1$, then $v_{\text{max}} = 2\sqrt{n}$. Minor debugging and function improvements are done in version 2.10.0, too. Acceleration rate is now progressively increasing. Each time the inner loop starts, counter $I_2$ (see Nagode, 2015, for details) is initiated and constant

$$A = \left. \frac{1 - a_r}{a_r(D_l w_l - D_{\text{min}})} \right|_{I_2=1}$$  

(2)

is calculated. Acceleration rate $a_r$ at $I_2 = 1$ always equals the value stored in the input argument $\text{ar}$. Otherwise

$$a_r = \left. \frac{1}{A(D_l w_l - D_{\text{min}}) + 1} \right|_{I_2>1}.$$  

(3)

The Newton-Raphson root finding in C++ functions is improved in version 2.9.3, too. This affects only Weibull, gamma and von Mises parametric families. Circular von Mises parametric family is added.
and further debugging is done in version 2.9.2. Version 2.9.1 is further debugged version 2.8.4. The R code is extended and rewritten in S4 class system. The background C code is extended and rewritten as object-oriented C++ code, too. The package can easier be extended to other parametric families. Multivariate normal mixtures with unrestricted variance-covariance matrices are added. Clustering is added and classification is improved.

3 Examples

To illustrate the use of the REBMIX algorithm, univariate and multivariate datasets are considered. The `rebmix` is loaded and the prompt before starting new page is set to \texttt{TRUE}.

\begin{verbatim}
R> library("rebmix")
R> devAskNewPage(ask = TRUE)
\end{verbatim}

3.1 Gamma datasets

Three gamma mixtures are considered (Wiper et al., 2001). The first has four well-separated components with means 2, 4, 6 and 8, respectively

\[
\begin{align*}
\theta_1 &= 1/100 \quad \beta_1 = 200 \quad n_1 = 100 \\
\theta_2 &= 1/100 \quad \beta_2 = 400 \quad n_2 = 100 \\
\theta_3 &= 1/100 \quad \beta_3 = 600 \quad n_3 = 100 \\
\theta_4 &= 1/100 \quad \beta_4 = 800 \quad n_4 = 100.
\end{align*}
\]

The second has equal means but different variances and weights

\[
\begin{align*}
\theta_1 &= 1/27 \quad \beta_1 = 9 \quad n_1 = 40 \\
\theta_2 &= 1/270 \quad \beta_2 = 90 \quad n_2 = 360.
\end{align*}
\]

The third is a mixture of a rather diffuse component with mean 6 and two lower weighted components with smaller variances and means of 2 and 10, respectively

\[
\begin{align*}
\theta_1 &= 1/20 \quad \beta_1 = 40 \quad n_1 = 80 \\
\theta_2 &= 1 \quad \beta_2 = 6 \quad n_2 = 240 \\
\theta_3 &= 1/20 \quad \beta_3 = 200 \quad n_3 = 80.
\end{align*}
\]

3.1.1 Finite mixture generation

\begin{verbatim}
R> n <- c(100, 100, 100, 100)
R> Theta <- new("RNGMIX.Theta", c = 4, pdf = "gamma")
R> a.theta1(Theta) <- rep(1/100, 4)
R> a.theta2(Theta) <- c(200, 400, 600, 800)
R> gamma1 <- RNGMIX(Dataset.name = "gamma1", n = n, Theta = a.Theta(Theta))
R> n <- c(40, 360)
R> Theta <- new("RNGMIX.Theta", c = 2, pdf = "gamma")
R> a.theta1(Theta) <- c(1/27, 1/270)
R> a.theta2(Theta) <- c(9, 90)
R> gamma2 <- RNGMIX(Dataset.name = "gamma2", n = n, Theta = a.Theta(Theta))
R> n <- c(80, 240, 80)
R> Theta <- new("RNGMIX.Theta", c = 3, pdf = "gamma")
R> a.theta1(Theta) <- c(1/20, 1, 1/20)
R> a.theta2(Theta) <- c(40, 6, 200)
R> gamma3 <- RNGMIX(Dataset.name = "gamma3", rseed = -4, n = n, + Theta = a.Theta(Theta))
\end{verbatim}
3.1.2 Finite mixture estimation

```r
R> gamma1est <- REBMIX(Dataset = a.Dataset(gamma1), Preprocessing = "kernel density estimation + cmax = 8, Criterion = c("AIC", "BIC"), pdf = "gamma")
R> gamma2est <- REBMIX(Dataset = a.Dataset(gamma2), Preprocessing = "histogram", + cmax = 8, Criterion = "BIC", pdf = "gamma")
R> gamma3est <- REBMIX(Dataset = a.Dataset(gamma3), Preprocessing = "histogram", + cmax = 8, Criterion = "BIC", pdf = "gamma", K = 23:27)
```

3.1.3 Plot method

```r
R> plot(gamma3est, pos = 1, what = c("den", "dis"), ncol = 2, npts = 1000)
```

![Gamma 3 dataset. Empirical density (circles) and predictive gamma mixture density in black solid line.](image)

3.1.4 Summary, a.theta1.all and a.theta2.all methods

```r
R> summary(gamma2est)

Dataset Preprocessing Criterion c v/k IC logL M
1 gamma2 histogram BIC 2 16 -1321 676 5
Maximum logL = 676 at pos = 1.

R> a.theta1.all(gamma1est, pos = 2)

[,1]
theta2.1 0.01027
theta2.2 0.00921
theta2.3 0.00870
theta2.4 0.01118

R> a.theta2.all(gamma1est, pos = 2)

[,1]
theta2.1 195
theta2.2 437
theta2.3 918
theta2.4 535
```
3.1.5 Bootstrap methods

```r
R> gamma3boot <- boot(x = gamma3est, pos = 1, Bootstrap = "p", B = 10)
R> gamma3boot

An object of class "REBMIX.boot"
Slot "c":
[1] 3 3 3 3 3 3 3 3 3 3
Slot "c.se":
[1] 0
Slot "c.cv":
[1] 0
Slot "c.mode":
[1] 3
Slot "c.prob":
[1] 1
R> summary(gamma3boot)

w.cv
[1] 0.122 0.200 0.175
[,1]
theta1.1.cv 0.457
theta1.2.cv 0.840
theta1.3.cv 0.508
[,1]
theta2.1.cv 2.149
theta2.2.cv 0.697
theta2.3.cv 1.083
Mode probability = 1 at c = 3 components.
```

3.2 Poisson dataset

Dataset consists of \( n = 600 \) two dimensional observations obtained by generating data points separately from each of three Poisson distributions. The component dataset sizes and parameters, which are those studied in Ma et al. (2009), are displayed below

\[
\theta_1 = (3, 2)^T \quad n_1 = 200 \\
\theta_2 = (9, 10)^T \quad n_2 = 200 \\
\theta_3 = (15, 16)^T \quad n_3 = 200 
\]

For the dataset Ma et al. (2009) conduct 100 experiments by selecting different initial values of the mixing proportions. In all the cases, the adaptive gradient BYY learning algorithm leads to the correct model selection, i.e., finally allocating the correct number of Poissons for the dataset. In the meantime, it also results in an estimate for each parameter in the original or true Poisson mixture which generated the dataset. As the dataset of Ma et al. (2009) can not exactly be reproduced, 10 datasets are generated with random seeds \( r_{seed} \) ranging from \(-10\) to \(-1\).

3.2.1 Finite mixture generation

```r
R> n <- c(200, 200, 200)
R> Theta <- new("RNGMIX.Theta", c = 3, pdf = rep("Poisson", 2))
R> a.theta1(Theta, 1) <- c(3, 2)
R> a.theta1(Theta, 2) <- c(9, 10)
R> a.theta1(Theta, 3) <- c(15, 16)
R> poisson <- RNGMIX(Dataset.name = paste("Poisson", 1:10, sep = ""),
+                   n = n, Theta = a.Theta(Theta))
```
3.2.2 Finite mixture estimation

\[
R > \text{poissonest} <- \text{REBMIX(Dataset = a.Dataset(poisson), Preprocessing = "histogram", + cmax = 10, Criterion = "MDL5", pdf = rep("Poisson", 2), K = 1)}
\]

3.2.3 Plot method

\[
R > \text{plot(poissonest, pos = 9, what = c("dens", "marg", "IC", "D", + "logL"), nrow = 2, ncol = 3, npts = 1000)}
\]

![Graph showing empirical densities, predictive multivariate Poisson-Poisson mixture density, empirical densities, predictive univariate marginal Poisson mixture densities, and progress charts.](image)

Figure 2: Poisson dataset. Empirical densities (coloured large circles), predictive multivariate Poisson-Poisson mixture density (coloured small circles), empirical densities (circles), predictive univariate marginal Poisson mixture densities and progress charts (solid line).

3.2.4 Clustering

3.2.5 Summary, a.theta1.all and a.theta2.all methods

\[
R > \text{summary(poissonest)}
\]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Preprocessing</th>
<th>Criterion</th>
<th>c</th>
<th>v/k</th>
<th>IC</th>
<th>logL</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson_1</td>
<td>histogram</td>
<td>MDL5</td>
<td>3</td>
<td>1</td>
<td>7040</td>
<td>-3392</td>
<td>8</td>
</tr>
<tr>
<td>Poisson_2</td>
<td>histogram</td>
<td>MDL5</td>
<td>5</td>
<td>1</td>
<td>7245</td>
<td>-3399</td>
<td>14</td>
</tr>
<tr>
<td>Poisson_3</td>
<td>histogram</td>
<td>MDL5</td>
<td>4</td>
<td>1</td>
<td>7174</td>
<td>-3411</td>
<td>11</td>
</tr>
<tr>
<td>Poisson_4</td>
<td>histogram</td>
<td>MDL5</td>
<td>3</td>
<td>1</td>
<td>7004</td>
<td>-3374</td>
<td>8</td>
</tr>
<tr>
<td>Poisson_5</td>
<td>histogram</td>
<td>MDL5</td>
<td>3</td>
<td>1</td>
<td>7074</td>
<td>-3409</td>
<td>8</td>
</tr>
<tr>
<td>Poisson_6</td>
<td>histogram</td>
<td>MDL5</td>
<td>4</td>
<td>1</td>
<td>7133</td>
<td>-3391</td>
<td>11</td>
</tr>
<tr>
<td>Poisson_7</td>
<td>histogram</td>
<td>MDL5</td>
<td>2</td>
<td>1</td>
<td>7303</td>
<td>-3572</td>
<td>5</td>
</tr>
<tr>
<td>Poisson_8</td>
<td>histogram</td>
<td>MDL5</td>
<td>3</td>
<td>1</td>
<td>7124</td>
<td>-3434</td>
<td>8</td>
</tr>
</tbody>
</table>
R> poissonclu <- RCLRMIX(x = poissonest, pos = 9, Zt = a.Zt(poisson))
R> plot(poissonclu)

Figure 3: Poisson dataset. Predictive cluster membership (coloured circles), error (black circles).

R> a.theta1.all(poissonest, pos = 9)

[,1]  [,2]
theta2.1  3.65  2.94
theta2.2  10.84 10.87
theta2.3  13.90 16.75

R> a.theta2.all(poissonest, pos = 9)

[,1]  [,2]
theta2.1  0    0
theta2.2  0    0
theta2.3  0    0

3.3 Multivariate normal wreath dataset

A wreath dataset (Fraley et al., 2005) consist of 1000 observations drawn from a 14-component normal mixture in which the covariances of the components have the same size and shape but differ in orientation.

R> data("wreath", package = "mclust")
3.3.1 Finite mixture estimation

\[ R> \text{wreathest} <- \text{REBMIX}(\text{model = "REBMVNORM"}, \text{Dataset = list(as.data.frame(wreath))}, + \text{Preprocessing = "histogram", cmax = 20, Criterion = "BIC"}) \]

3.3.2 Plot method

\[ R> \text{plot(wreathest)} \]

![Plot of dataset wreath](image)

Dataset = dataset1, Preprocessing = histogram, Restraints = loose, c_{max} = 20, a_r = 0.1, c = 14, v = 21, BIC = 11174, log L = -5300.

Figure 4: Dataset wreath. Empirical densities (coloured circles), predictive multivariate normal mixture density (coloured lines).

3.3.3 Clustering

3.3.4 Summary method

\[ R> \text{summary(wreathest)} \]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Preprocessing</th>
<th>Criterion</th>
<th>c</th>
<th>v/k</th>
<th>IC</th>
<th>logL</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataset1</td>
<td>histogram</td>
<td>BIC</td>
<td>14</td>
<td>21</td>
<td>11174</td>
<td>-5300</td>
<td>83</td>
</tr>
</tbody>
</table>

Maximum logL = -5300 at pos = 1.

3.3.5 Summary method

\[ R> \text{summary(wreathclu)} \]
R> wreathclu <- RCLRMIX(model = "RCLRMVNORM", x = wreathest)
R> plot(wreathclu, s = 14)

![Dataset plot](image)

Figure 5: Dataset `wreath`. Predictive cluster membership (coloured circles).

<table>
<thead>
<tr>
<th>Number of clusters</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>From cluster</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>To cluster</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Entropy</td>
<td>1.50e-14</td>
<td>9.88e-04</td>
<td>2.23e-03</td>
<td>4.09e-03</td>
<td>6.43e-03</td>
</tr>
<tr>
<td>Entropy decrease</td>
<td>0.000988</td>
<td>0.001239</td>
<td>0.001868</td>
<td>0.002335</td>
<td>0.005271</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of clusters</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>From cluster</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>To cluster</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>Entropy</td>
<td>1.17e-02</td>
<td>2.13e-02</td>
<td>3.24e-02</td>
<td>4.96e-02</td>
<td>7.39e-02</td>
</tr>
<tr>
<td>Entropy decrease</td>
<td>0.009594</td>
<td>0.011101</td>
<td>0.017251</td>
<td>0.024271</td>
<td>0.027832</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of clusters</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>From cluster</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>To cluster</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Entropy</td>
<td>1.02e-01</td>
<td>1.31e-01</td>
<td>2.82e-01</td>
</tr>
<tr>
<td>Entropy decrease</td>
<td>0.029363</td>
<td>0.150611</td>
<td>0.607498</td>
</tr>
</tbody>
</table>

### 3.4 Multivariate normal ex4.1 dataset

A ex4.1 dataset (Baudry et al., 2010; Fraley et al., 2016) consist of 600 two dimensional observations.

R> data("Baudry_etal_2010_JCGS_examples", package = "mclust")
3.4.1 Finite mixture estimation

```r
R> ex4.1est <- REBMIX(model = "REBMVNORM", Dataset = list(as.data.frame(ex4.1)),
+         Preprocessing = "kernel density estimation", cmax = 10, Criterion = "AIC")
```

3.4.2 Plot method

```r
R> plot(ex4.1est, pos = 1, what = c("dens"), nrow = 1, ncol = 1)
```

![Graph of dataset ex4.1](image)

Figure 6: Dataset ex4.1. Empirical densities (coloured circles), predictive multivariate normal mixture density (coloured lines).

3.4.3 Clustering

```r
R> ex4.1clu <- RCLRMIX(model = "RCLRMVNORM", x = ex4.1est)
R> plot(ex4.1clu)
```

![Cluster plot of dataset ex4.1](image)

Figure 7: Dataset ex4.1. Predictive cluster membership (coloured circles).
3.4.4 Summary method

R> summary(ex4.1est)

Dataset Preprocessing Criterion c v/k IC logL M
1 dataset1 kernel density estimation AIC 6  28 4084 -2007 35
Maximum logL = -2007 at pos = 1.

3.5 Multivariate iris dataset

The well known set of iris data as collected originally by Anderson (1936) and first analysed by Fisher (1936) is considered here. It is available at Asuncion and Newman (2007) consisting of the measurements of the length and width of both sepals and petals of 50 plants for each of the three types of iris species setosa, versicolor and virginica. The iris dataset is loaded, split into three subsets for the three classes and the Class column is removed.

R> data("iris")
R> levels(iris[["Class"]])

[1] "iris-setosa" "iris-versicolor" "iris-virginica"

R> set.seed(5)
R> Iris <- split(p = 0.75, Dataset = iris, class = 5)

3.5.1 Finite mixture estimation

R> irisest <- REBMIX(model = "REBMVNORM", Dataset = a.train(Iris),
+ Preprocessing = "kernel density estimation", cmax = 10, Criterion = "ICL-BIC")

3.5.2 Classification

R> iriscla <- RCLSMIX(model = "RCLSMVNORM", x = list(irisest), Dataset = a.test(Iris),
+ Zt = a.Zt(Iris))

3.5.3 Show and summary methods

R> iriscla

An object of class "RCLSMVNORM"
Slot "CM":

   1  2  3
 1 13  0  0
 2  0 13  0
 3  0  1 12
Slot "Error":
[1] 0.0256
Slot "Precision":
[1] 1.000 1.000 0.923
Slot "Sensitivity":
[1] 1.000 0.929 1.000
Slot "Specificity":
[1] 1.000 1.040 0.963
Slot "Chunks":
[1] 1
R> summary(iriscla)
Test Predictive Frequency

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>3</td>
<td>12</td>
</tr>
</tbody>
</table>

Error = 0.0256.

3.5.4 Plot method

R> plot(iriscla, nrow = 3, ncol = 2)

Figure 8: Dataset iris. Predictive class membership (coloured circles), error (black circles).

3.6 Multivariate adult dataset

The adult dataset containing 48842 instances with 16 continuous, binary and discrete variables was extracted from the census bureau database Asuncion and Newman (2007). Extraction was done by Barry Becker from the 1994 census bureau database. The adult dataset is loaded, complete cases are extracted and levels are replaced with numbers.
R> data("adult")
R> adult <- adult[complete.cases(adult), ]
R> adult <- as.data.frame(data.matrix(adult))

Numbers of unique values for variables are determined and displayed.

R> cmax <- unlist(lapply(apply(adult[, c(-1, -16)], 2, unique), + length))
R> cmax

Age Workclass Fnlwgt Education Education.Num
74 7 26741 16 16
Marital.Status Occupation Relationship Race Sex
7 14 6 5 2
121 97 96 41

The dataset is split into train and test subsets for the two incomes and the Type and Income columns are removed.

R> Adult <- split(p = list(type = 1, train = 2, test = 1), Dataset = adult, + class = 16)

3.6.1 Finite mixture estimation

Number of components, component weights and component parameters are estimated assuming that the variables are independent for the set of chunks \( y_{1j}, y_{2j}, \ldots, y_{14j} \).

R> adultest <- list()
R> for (i in 1:14) {
+    adultest[[i]] <- REBMIX(Dataset = a.train(chunk(Adult, i)), 
+ Preprocessing = "histogram", cmax = min(120, cmax[i]), 
+ Criterion = "BIC", pdf = "Dirac", K = 1)
+ }

3.6.2 Classification

The class membership prediction is based upon the best first search algorithm.

R> adultcla <- BFSMIX(x = adultest, Dataset = a.test(Adult), Zt = a.Zt(Adult))

3.6.3 Show and summary methods

R> adultcla

An object of class "RCLSMIX"
Slot "CM":

<table>
<thead>
<tr>
<th>1 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 10649 711</td>
</tr>
<tr>
<td>2 1397 2303</td>
</tr>
</tbody>
</table>
Slot "Error":
[1] 0.14
Slot "Precision":
[1] 0.937 0.622
Slot "Sensitivity":
[1] 0.884 0.764
Slot "Specificity":
[1] 1.228 0.943
Slot "Chunks":
[1] 11 12 4 8 1

R> summary(adultcla)

Test Predictive Frequency
1 1 1 1 10649
2 2 1 1 1397
3 1 2 711
4 2 2 2 2303
Error = 0.14.

3.6.4 Plot method

R> plot(adultcla, nrow = 5, ncol = 2)

Figure 9: Dataset adult. Predictive class membership (coloured circles), error (black circles).

4 Summary

The users of the rebmix package are kindly encouraged to inform the author about bugs and wishes.
5 Acknowledgement

The author thanks Branislav Panič for his contribution to the Velleman (1976) rule extension to multidimensions and for his other valuable suggestions on the package from version 2.9.0 on.

References


Marko Nagode
University of Ljubljana
Faculty of Mechanical Engineering
Aškerčeva 6
1000 Ljubljana