ML2R Coding Nuggets
Linear Programming for Robust Regression

Pascal Welke*
Machine Learning Rhine-Ruhr
University of Bonn
Bonn, Germany

Christian Bauckhage†
Machine Learning Rhine-Ruhr
Fraunhofer IAIS
St. Augustin, Germany

ABSTRACT

The ML2R Coding Nugget template defines a common look and feel for ML2R coding nuggets. The source code of this document shows how to use the ML2R Coding Nugget Template to produce nice documents.

1 INTRODUCTION

Coding nuggets are currently (short) how-tos that show how to solve problems "by foot" in practice. The typical structure of such a paper consists of motivating a (machine learning) problem, showing how to formulate this problem in a particular way and then showing how to solve it practically using existing methods.

Examples include "Solving Linear Programming Problems" [2] and "Linear Programming for Robust Regression" [1]. In the latter example, e.g. we show how to fit a linear model to data by minimizing least absolute deviations, instead of the more common least squares. To this end, we formulate the optimization problem, transform it to show that it can be solved by linear programming and then solve it in python using methods from numpy and scipy alone.

2 TECHNICAL DETAILS

preamble.tex contains several imports and definitions. There is a part marked optional that you can change, remove or add to as you please. Please refrain from changing the part, however, that is marked as non-optional. Below we give a short introduction to

2.1 Colors

The preamble defines the ML2R corporate design colors that should be used throughout the document and within figures (cf. Fig. 1) to provide a common look and feel:

blue	:	ml2rblu
yellow	:	ml2ryel
green	:	ml2rgrn
turquoise	:	ml2rtrq

2.2 Python Code and keywords

Python code with pretty printing is supported. Inline code looks like this:

```
vecWLAD = result.x[:m]
print('w0 =',vecWLAD[0])
print('w1 =',vecWLAD[1])
```

* 0000-0002-2123-3781
† 0000-0001-6615-2128

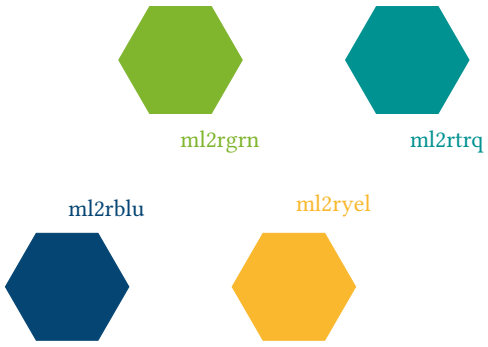

Figure 1: A figure with our corporate design colors

Listing 1: setting up a linear programming problem

```
matA = np.zeros((2*n, m+n))
matA[:n,:m] = +matF.T
matA[n:,:m] = -matF.T
matA[:n,m:] = -np.eye(n)
matA[n:,m:] = -np.eye(n)

vecB = np.zeros(2*n)
vecB[:n] = +vecY
vecB[n:] = -vecY

vecC = np.hstack((np.zeros(m), np.ones(n)))
```

An example of a longer, floating python code environment can be found in Listing 1.

ACKNOWLEDGMENTS

This material was produced within the Competence Center for Machine Learning Rhine-Ruhr (**ML2R**) which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01|S18038C). The authors gratefully acknowledge this support.

REFERENCES

[1] P. Welke and C. Bauckhage. 2020. *ML2R Coding Nuggets: Linear Programming for Robust Regression.* Technical Report. MLAI, University of Bonn.
[2] P. Welke and C. Bauckhage. 2020. *ML2R Coding Nuggets: Solving Linear Programming Problems.* Technical Report. MLAI, University of Bonn.