Proceedings of Machine Learning Research 1:1–18, 2019

Conformal Prediction for Students' Grades in a Course Recommender System*

Raphaël Morsomme† RAPHAEL.MORSOMME@MAASTRICHTUNIVERSITY.NL
Zwingelput 4, 6211 KH Maastricht, the Netherlands

Evgueni Smirnov SMIRNOV@MAASTRICHTUNIVERSITY.NL
DKE's address, 6211 KH Maastricht, the Netherlands

Abstract

Course selection can be challenging for student of Liberal Arts programs. In particular, due to the highly personalized curricula of liberal arts students, it is often difficult to assess whether or not a certain course is too advanced given their academic background. To assist students of the liberal arts program of the University College Maastricht, Morsomme and Vazquez (2019) developed a course recommender system that suggests courses whose content matches the student's academic interests, and issues warnings for courses that it deems too advanced. To accomplish the latter, we fit a sparse regression model for grade. In this paper, we use the Inductive Confidence Machine (Papadopoulos, 2002) to obtain predictive regions for future grades. We compare the results obtained with the standard regression nonconformity measure and the normalized nonconformity measure. We find that both nonconformity measures produce valid predictive regions and that the width of the predictive regions is correlated with the accuracy of the regression model.

Keywords: Conformal Prediction, Recommender System, Grade Prediction, Lasso, Education

Raphaël: make all figures black and white

question: make flowchart with two pillars (course recommendation and warnings) instead of input/model/intermediate results/output?

connect with bibtex bibliography from citavi

1. Introduction

difficult to assess if student ready. Morsomme and Vazquez include warnings based on grade prediction to help student identify course too advanced for them. Provide warning if predicted grade is a fail.

In this paper, we present an applictaion of the ICM to provide students with prediction regions instead of warnings based on a point estimate for the grade. The advantage of PR over point estimates is that it improves info position of student, tehreby enabling them to make better-informed course selection. We decide to use conformal prediction prediction to tailor the PR to the students and to use an ICM to alleviate the computational costs of transductive.

* XXX

† with a note

© 2019 R. Morsomme & E. Smirnov.

Liberal Arts program are often characterized by their open curriculum which allows student to tailor their own study program to their academic objectives. These highly personalized curricula make it difficult for students, academic advisors and course coordinators to assess whether a particular course is too advanced given the student's current academic background or whether she/he has acquired the prerequisite skills via a combination of courses. To alleviate this problem, Morsomme and Vazquez (2019) developed a course recommender system which suggest courses whose content matches the academic interests of the student and issues warnings for courses that it deems too advanced given the student's academic background. To accomplish the latter, they fit a sparse regression model to each course which takes as input a model of the student indicating the skills that she/he has acquired in previous courses and her/his academic performance in different academic disciplines in order to predict grades. Warnings are issued when the system predicts a fail grade.

This paper presents the application of an Inductive Confidence Machine (ICM) (Papadopoulos, 2002) to complement the existing course recommender system of [ref: Morsomme and Vazquez, 2019] with predictive regions for future grades tailored to the student.

To improve student info position, we complement point estimates predictions with region predictions. To tailor the prediction to each student, we use conformal prediction.

This way, the information position of students is improved; instead of providing students with a binary advice, i.e. a warning if the predicted grade is a fail and nothing if it is a passing grade, students are provided with a predictive regions indicating a range of grades they are likely to receive in the courses that they are, for instance, considering for the following term particular course.

We present previous research on grade prediction in Section 2 and the data in Section 3 We introduce the methodology related to the course recommender system and the ICM in Section 4 and present the results in Section 5

2. Related Work (Evgueni)

Conclusion of the section: no previous research has applied the conformal prediction framework to students' grade prediction.

3. Data

We employ two sets of data: student data and course data.

The student data consists of anonymized course enrollment information. We use the transcripts of the 2,526 students of the liberal arts program between 2008 and 2019 with a total of 79,245 course enrollments. We exclude enrollments with a missing grade which indicates that the student either dropped the course or fail the attendance requirement. In the latter case, the data set contains an observation corresponding to the resit. Table 1 presents the student data. Each row contains an anonymized student ID, a course ID, a year and semester, and the obtained grade.

The course data consists of the 2018-2019 course catalogues of five departments of Maastricht University: European Studies, University College Maastricht, University College Venlo, Psychology and Science Program. These catalogues contain a one-page description of

490 courses. Table 2 presents the textual data in the tidy format with one row per document-term [ref: Wickham 2014]. We process the data following common cleaning procedures [ref: Meyer, 2008]: we tokenize the individual terms, stem them with the Hunspell dictionary and remove common stop words, numbers between 1 and 1,000, and terms occurring less than 3 times in the data set.

Table 1: Example of Student Data

Student ID	**Course ID**	**Academic Year**	**Period**	**Grade**
44940	CAP3000	2009-2010	4	8.8
37490	SSC2037	2009-2010	4	8.4
71216	HUM1003	2010-2011	4	6.8
44212	SSC2049	2010-2011	2	8.4
85930	SSC2043	2011-2012	1	4.3
14492	COR1004	2012-2013	2	8.5
34750	HUM2049	2013-2014	5	6.0
32316	SSC1001	2013-2014	1	8.5
22092	SCI1009	2014-2015	1	6.4
19512	COR1004	2016-2017	5	7.0

Table 2: Example of Course Data

Course ID	**Course Title**	**Department**	**Term**
HUM3034	World History	UCM	understand
HUM3034	World History	UCM	major
HUM3034	World History	UCM	issue
HUM3034	World History	UCM	episode
HUM3034	World History	UCM	shape
HUM3034	World History	UCM	history
HUM3034	World History	UCM	mankind
HUM3034	World History	UCM	focus
HUM3034	World History	UCM	theme
HUM3034	World History	UCM	topic

4. Methodology

4.1. The Course Recommender System

4.1.1. Overview

The course recommender system has two goals: suggesting courses whose content matches the student's academic interests and help the student assess whether she/he has acquired the necessary knowledge to succeed in the course she /he is considering for the following term. Although the former is not of direct interest for this paper, we briefly describe

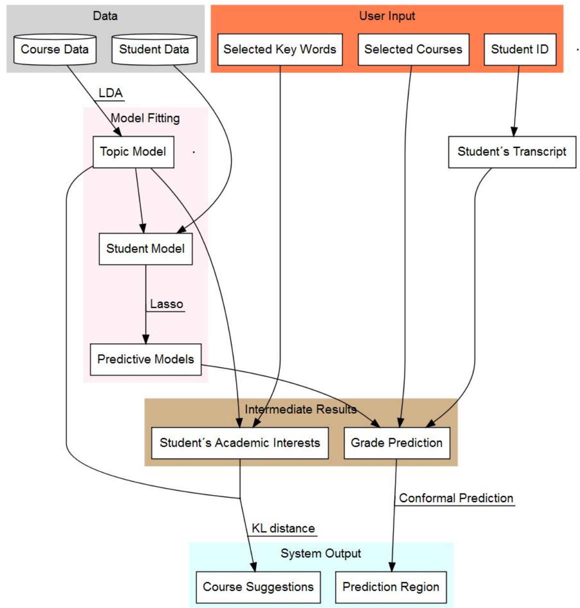


Figure 1: Flowchart of the Course Recommender System

4.1.2. Topic Model

We start by fitting a topic model to the course data using the Latent Dirichlet Allocation generative model and the Gibbs sampling algorithm (ref: Blei, 2003; Phan, 2008). The LDA model views a topic as a mixture of words and a document as a mixture of topics. More precisely, topics are conceptualize as a probability distribution over the vocabulary of the data, and document as a set of words, each drawn from a probability distribution over topics specific to that document. The term *Dirichlet* comes from the fact that the word distribution β_t for a topic t is generated from a Dirichlet distribution $\beta_t \sim Dirichlet(\delta)$ and the topic distribution θ_d for a document d is generated from a Dirichlet distribution $\theta_d \sim Dirichlet(\alpha)$ where δ and α act as hyper-parameters determining how concentrated the distributions of words in topics and the distributions of topics in documents are.

We follow Phan (ref: 2008) who used a Gibbs sampler to learn the distributions β and θ. The obtained topic model consists of a term distribution for each topic and a topic distribution for each course description. Figure 2 shows the importance of the main topics of the core course COR1004 Political Philosophy. Figure 3 indicates that the two main topics of the course (topics 4 and 19) respectively correspond to the themes of international politics and philosophy.

The number of topics T needs to be specified *a-priori*. We trained 30 models with $K = 5, 10, \cdots, 150$. As suggested by Griffiths and Steyvers (ref: 2008), we set $\alpha = 50/K$ and $\delta = 0.1$. We run 6,000 iterations of the Gibbs sampler with a burn in of 1,000 iterations and sample every 100 iterations. To prevent being stuck in a local optimum, we use 10 random initialization and keep the best model with respect to the log-likelihood. We choose the number of topics K that yields the model with the maximum likelihood. Figure X shows that the model with 65 topics has the maximum likelihood. To increase the quality of the model, we retrain the model with 16,000 iterations of the the Gibbs sampler, a burn in of 2,000 iterations and 20 random starts and the other parameters at their original values.

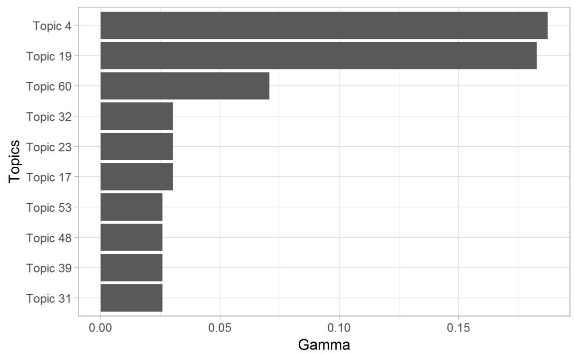
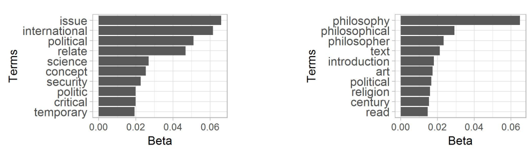


Figure 2: Topic Distribution in the course COR1004 Political Philosophy.



(a) Topic 4 (b) Topic 19

Figure 3: Word Distribution in the main two topics of COR1004 Political Philosophy. Topic 4 corresponds to international politics and Topic 19 to philosophy.

4.1.3. Student Model

The student model consists of two elements: academic performance and level of topic-specific expertise. Academic performance corresponds to the student's general GPA as well as her/his GPA in humanities, natural sciences, social sciences, skills and projects. These are derived from the student's transcript in a straightforward way. Topic-specific expertise corresponds to the knowledge that the student has acquired in each of the topics present in the topic model in previous courses. We posit that when students takes a course they acquire knowledge about its content and that the amount of knowledge they acquire is proportional to their grade. More precisely, we assume that students who obtain 10/10 in a course acquire all the knowledge related to its content and that those who obtain 5/10 only acquire half of it. The content of a course is determined by its topic distribution in the topic model and the grade are retrieved from the student's transcript. Hence if a student

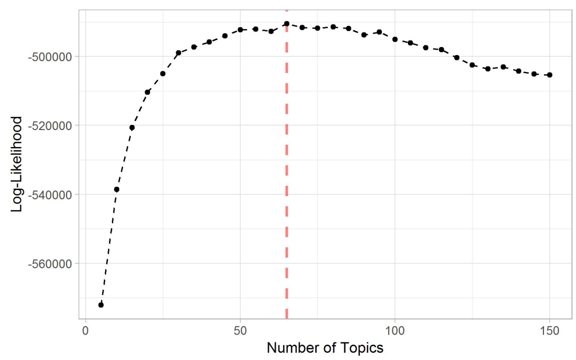


Figure 4: Maximum Likelihood Model Selection: we choose the model with 65 topics.

has taken n courses and g_i corresponds to her/his grade in course i, for $i = 1, \cdots, n$, then the expertise of the student e_t in topic t simply corresponds to

$$e_t = \sum_{i=1}^{n} g_i \gamma_{i,t}$$

where $\gamma_{i,t}$ corresponds to the importance of topic t in course i in the topic model. Figure 5 presents a toy example of the contribution of individual courses to a student's expertise in 5 topics.

Table 3: Toy Example of Topic Expertise

(a) Topic Distribution γ

Course	**Topic 1**	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**
Course 1	0.0	0.7	0.2	0.1	0.0
Course 2	0.2	0.2	0.2	0.2	0.2
Course 3	0.0	0.4	0.2	0.1	0.2

(b) Transcript

Course	**Grade**
Course 1	6/10
Course 2	9/10
Course 3	2.5/10

(c) Course Contribution towards Topic Expertise

Course	**Topic 1**	**Topic 2**	**Topic 3**	**Topic 4**	**Topic 5**
Course 1	0.00	0.42	0.12	0.060	0.00
Course 2	0.18	0.18	0.18	0.180	0.18
Course 3	0.00	0.10	0.05	0.025	0.05

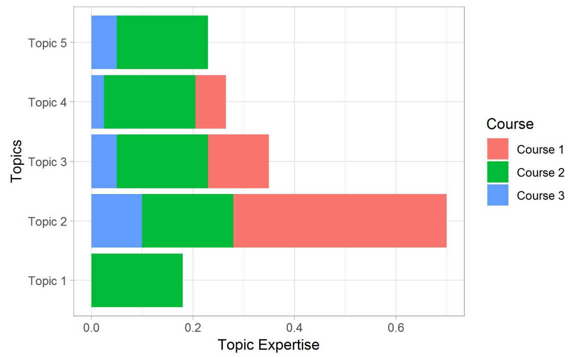


Figure 5: Contribution of Individual Courses towards a Student's Topic Expertise Profile.

4.1.4. Underlying Algorithm: Sparse Regression Model

We fit a sparse linear regression model for grade prediction to each of the 132 courses currently offered at the college with at least 20 student enrollments since 2008. The input consists of the 71 variables present in the student model at the start of the course: her/his 6 GPA's (1 general and 5 discipline-specific) and 65 levels of topic-specific expertise. Since the number of predictors is relatively large, we regularize the models with the lasso penalty (ref: Tibshirany, 1998). For each course, we use 10-fold cross-validation to find the value of the lasso tuning parameter λ that minimizes the CV mean absolute error (mae), a more robust loss function than the squared absolute error (ref: els, 2008). Figure **??** presents the distribution of the CV mae for the 132 prediction models. The model for the course *PRO2004 Academic Debate* has the smallest prediction error (0.38 grade point) and the model for *SCI3006 Mathematical Modelling* the largest (1.80 grade point). The mean CV mae weighted by the number of students enrolled in the course is 0.78, the median is 0.78 and the standard deviation is 0.28.

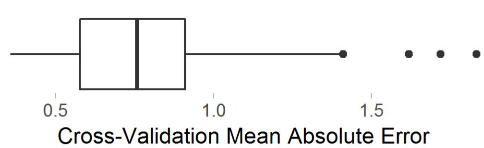


Figure 6: Distribution of Cross-Validation Error

4.1.5. Prediction Region for Grade

Import from EDM2019

4.2. The Inductive Confidence Machine (Evgueni)

5. Results

Make figures

6. Conclusion

valid

width correlated to accuracy of underlying algorithm i.e. CV error of lasso

7. Comments to Evgueni

Focus on conformal prediction instead of recommender system, lasso, topic model, etc. This way, the paper is simpler to follow.

If we want to expand on lasso, we should write a paper on lasso itself; just like we are doing with grade prediction. The same goes for topic model.

8. others

This is a sample article that uses the `jmlr` class with the `wcp` class option. Please follow the guidelines in this sample document as it can help to reduce complications when combining the articles into a book. Please avoid using obsolete commands, such as `\rm`, and obsolete packages, such as `epsfig`.[1]

Please also ensure that your document will compile with PDFLaTeX. If you have an error message that's puzzling you, first check for it at the UK TUG FAQ http://www.tex.ac.uk/cgi-bin/texfaq2html?label=man-latex. If that doesn't help, create a minimal working example (see http://theoval.cmp.uea.ac.uk/~nlct/latex/minexample/) and post to somewhere like TeX on StackExchange (http://tex.stackexchange.com/) or the LaTeX Community Forum (http://www.latex-community.org/forum/).

Note:
This is an numbered theorem-like environment that was defined in this document's preamble.

8.1. Sub-sections

Sub-sections are produced using `\subsection`.

8.1.1. Sub-sub-sections

Sub-sub-sections are produced using `\subsubsection`.

Sub-sub-sub-sections Sub-sub-sub-sections are produced using `\paragraph`. These are unnumbered with a running head.

Sub-sub-sub-sub-sections Sub-sub-sub-sub-sections are produced using `\subparagraph`. These are unnumbered with a running head.

1. See http://www.ctan.org/pkg/l2tabu

9. Cross-Referencing

Always use `\label` and `\ref` (or one of the commands described below) when cross-referencing. For example, the next section is Section 10. The jmlr class provides some convenient cross-referencing commands: `\sectionref`, `\equationref`, `\tableref`, `\figureref`, `\algorithmref`, `\theoremref`, `\lemmaref`, `\remarkref`, `\corollaryref`, `\definitionref`, `\conjectureref`, `\axiomref`, `\exampleref` and `\appendixref`. The argument of these commands may either be a single label or a comma-separated list of labels. Examples:

Referencing sections: Section 10 or Sections 1 and 10 or Sections 1, 10, 12.1 and 12.2.

Referencing equations: Equation (1) or Equations (1) and (3) or Equations (1), (2), (3) and (4).

Referencing tables: Table 4 or Tables 4 and 6 or Tables 4, 6 and **??**.

Referencing figures: Figure **??** or Figures **??** and 8 or Figures **??**, 8 and 9 or Figures 9(a) and 9(b).

Referencing algorithms: Algorithm 1 or Algorithms 1 and 2 or Algorithms 1, 2 and 3.

Referencing theorem-like environments: Theorem 1, Lemma 2, Remark 3, Corollary 4, Definition 5, Conjecture 6, Axiom 7 and Example 1.

Referencing appendices: Appendix A or Appendices A and B.

10. Equations

The jmlr class loads the amsmath package, so you can use any of the commands and environments defined there. (See the amsmath documentation for further details.[2])

Unnumbered single-lined equations should be displayed using `\[` and `\]`. For example:

$$E = mc^2$$

Numbered single-line equations should be displayed using the `equation` environment. For example:

$$\cos^2\theta + \sin^2\theta \equiv 1 \tag{1}$$

This can be referenced using `\label` and `\equationref`. For example, Equation (1).

Multi-lined numbered equations should be displayed using the `align` environment.[3] For example:

$$\begin{align} f(x) &= x^2 + x \tag{2} \\ f'(x) &= 2x + 1 \tag{3} \end{align}$$

Unnumbered multi-lined equations should be displayed using the `align*` environment. For example:

$$\begin{align*} f(x) &= (x+1)(x-1) \\ &= x^2 - 1 \end{align*}$$

2. Either `texdoc amsmath` or http://www.ctan.org/pkg/amsmath
3. For reasons why you shouldn't use the obsolete `eqnarray` environment, see Lars Madsen, *Avoid eqnarray!* TUGboat 33(1):21–25, 2012.

If you want to mix numbered with unnumbered lines use the align environment and suppress unwanted line numbers with `\nonumber`. For example:

$$\begin{aligned} y &= x^2 + 3x - 2x + 1 \nonumber \\ &= x^2 + x + 1 \tag{4} \end{aligned}$$

An equation that is too long to fit on a single line can be displayed using the `split` environment. Text can be embedded in an equation using `\text` or `\intertext` (as used in Theorem 1). See the amsmath documentation for further details.

10.1. Operator Names

Predefined operator names are listed in Table 4. For additional operators, either use `\operatorname`, for example $\operatorname{var}(X)$ or declare it with `\DeclareMathOperator`, for example

```
\DeclareMathOperator{\var}{var}
```

and then use this new command. If you want limits that go above and below the operator (like `\sum`) use the starred versions (`\operatorname*` or `\DeclareMathOperator*`).

Table 4: Predefined Operator Names (taken from amsmath documentation)

`\arccos`	$\arccos$	`\deg`	$\deg$	`\lg`	$\lg$	`\projlim`	$\projlim$
`\arcsin`	$\arcsin$	`\det`	$\det$	`\lim`	$\lim$	`\sec`	$\sec$
`\arctan`	$\arctan$	`\dim`	$\dim$	`\liminf`	$\liminf$	`\sin`	$\sin$
`\arg`	$\arg$	`\exp`	$\exp$	`\limsup`	$\limsup$	`\sinh`	$\sinh$
`\cos`	$\cos$	`\gcd`	$\gcd$	`\ln`	$\ln$	`\sup`	$\sup$
`\cosh`	$\cosh$	`\hom`	$\hom$	`\log`	$\log$	`\tan`	$\tan$
`\cot`	$\cot$	`\inf`	$\inf$	`\max`	$\max$	`\tanh`	$\tanh$
`\coth`	$\coth$	`\injlim`	$\injlim$	`\min`	$\min$		
`\csc`	$\csc$	`\ker`	$\ker$	`\Pr`	$\Pr$		
		`\varlimsup`	$\varlimsup$	`\varinjlim`	$\varinjlim$		
		`\varliminf`	$\varliminf$	`\varprojlim`	$\varprojlim$		

11. Vectors and Sets

Vectors should be typeset using `\vec`. For example $\boldsymbol{x}$. The jmlr class also provides `\set` to typeset a set. For example $\mathcal{S}$.

12. Floats

Floats, such as figures, tables and algorithms, are moving objects and are supposed to float to the nearest convenient location. Please don't force them to go in a particular place. In general it's best to use the `htbp` specifier and don't put the figure or table in the middle of a paragraph (that is make sure there's a paragraph break above and below the float).

Table 5: Student Data

Student ID	Course ID	Academic Year	Period	Grade
44940	CAP3000	2009-2010	4	8.8
37490	SSC2037	2009-2010	4	8.4
71216	HUM1003	2010-2011	4	6.8
44212	SSC2049	2010-2011	2	8.4
85930	SSC2043	2011-2012	1	4.3
14492	COR1004	2012-2013	2	8.5
34750	HUM2049	2013-2014	5	6.0
32316	SSC1001	2013-2014	1	8.5
22092	SCI1009	2014-2015	1	6.4
19512	COR1004	2016-2017	5	7.0

Floats are supposed to have a little extra space above and below them to make them stand out from the rest of the text. This extra spacing is put in automatically and shouldn't need modifying.

To ensure consistency, please *don't* try changing the format of the caption by doing something like:

```
\caption{\textit{A Sample Caption.}}
```

or

```
\caption{\em A Sample Caption.}
```

You can, of course, change the font for individual words or phrases, for example:

```
\caption{A Sample Caption With Some \emph{Emphasized Words}.}
```

12.1. Tables

Tables should go in the `table` environment. Within this environment use `\floatconts` (defined by jmlr) to set the caption correctly and center the table contents.

Table 6: An Example Table

Dataset	**Result**
Data1	0.12345
Data2	0.67890
Data3	0.54321
Data4	0.09876

If you want horizontal rules you can use the booktabs package which provides the commands `\toprule`, `\midrule` and `\bottomrule`. For example, see Table **??**.

If you want vertical lines as well, you can't use the booktabs commands as there'll be some unwanted gaps. Instead you can use LaTeX's `\hline`, but the rows may appear

Table 7: A Table With Horizontal Lines

Dataset	**Result**
Data1	0.12345
Data2	0.67890
Data3	0.54321
Data4	0.09876

a bit cramped. You can add extra space above or below a row using `\abovestrut` and `\belowstrut`. For example, see Table 8.

Table 8: A Table With Horizontal and Vertical Lines

Dataset	**Result**
Data1	0.12345
Data2	0.67890
Data3	0.54321
Data4	0.09876

If you want to align numbers on their decimal point, you can use the siunitx package. For example, see Table 9. For further details see the siunitx documentation[4].

Table 9: A Table With Numbers Aligned on the Decimal Point

Dataset	**Result**
Data1	0.123 45
Data2	10.6789
Data3	50.543
Data4	200.098 76

If the table is too wide, you can adjust the inter-column spacing by changing the value of `\tabcolsep`. For example:

```
\setlength{\tabcolsep}{3pt}
```

If the table is very wide but not very long, you can use the `sidewaystable` environment defined in the rotating package (so use `\usepackage{rotating}`). If the table is too long to fit on a page, you should use the `longtable` environment defined in the longtable package (so use `\usepackage{longtable}`).

12.2. Figures

Figures should go in the `figure` environment. Within this environment, use `\floatconts` to correctly position the caption and center the image. Use `\includegraphics` for external

4. Either `texdoc siunitx` or http://www.ctan.org/pkg/siunitx

graphics files but omit the file extension. Do not use `\epsfig` or `\psfig`. If you want to scale the image, it's better to use a fraction of the line width rather than an explicit length. For example, see Figure ??.

Figure 7: Example Image

If your image is made up of LaTeX code (for example, commands provided by the `pgf` package) you can include it using `\includeteximage` (defined by the `jmlr` class). This can be scaled and rotated in the same way as `\includegraphics`. For example, see Figure 8.

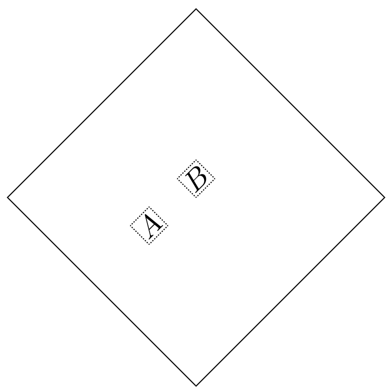


Figure 8: Image Created Using LaTeX Code

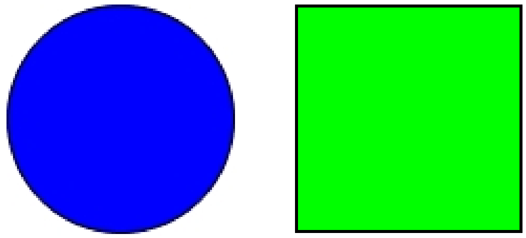
If the figure is too wide to fit on the page, you can use the `sidewaysfigure` environment defined in the `rotating` package.

Don't use `\graphicspath`. If the figures are contained in a subdirectory, specify this when you include the image, for example `\includegraphics{figures/mypic}`.

12.2.1. Sub-Figures

Sub-figures can be created using `\subfigure`, which is defined by the `jmlr` class. The optional argument allows you to provide a subcaption. The label should be placed in the mandatory argument of `\subfigure`. You can reference the entire figure, for example Figure 9, or you can reference part of the figure using `\figureref`, for example Figure 9(*a*). Alternatively you can reference the subfigure using `\subfigref`, for example (*a*) and (*b*) in Figure 9.

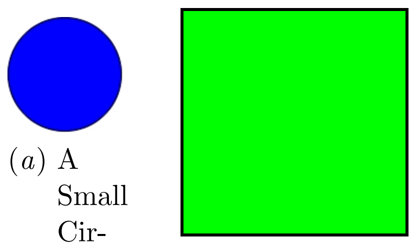
By default, the sub-figures are aligned on the baseline. This can be changed using the second optional argument of `\subfigure`. This may be `t` (top), `c` (centered) or `b` (bottom).



(*a*) A Circle (*b*) A Square

Figure 9: An Example With Sub-Figures.

For example, the subfigures (*a*) and (*b*) in Figure 10 both have `[c]` as the second optional argument.



(*a*) A Small Circle (*b*) A Square

Figure 10: Another Example With Sub-Figures.

12.3. Sub-Tables

There is an analogous command `\subtable` for sub-tables. It has the same syntax as `\subfigure` described above. You can reference the table using `\tableref`, for example Table 10 or you can reference part of the table, for example Table 10(*a*). Alternatively you can reference the subtable using `\subtabref`, for example (*a*) and (*b*) in Table 10.

Table 10: An Example With Sub-Tables

(*a*)

A	**B**
1	2

(*b*)

C	**D**
3	4
5	6

By default, the sub-tables are aligned on the top. This can be changed using the second optional argument of `\subtable`. This may be `t` (top), `c` (centered) or `b` (bottom). For example, the sub-tables (*a*) and (*b*) in Table 11 both have `[c]` as the second optional argument.

Table 11: Another Example With Sub-Tables

(a)

A	**B**
1	2

(b)

C	**D**
3	4
5	6

12.4. Algorithms

Enumerated textual algorithms can be displayed using the `algorithm` environment. Within this environment, use use an `enumerate` or nested `enumerate` environments. For example, see Algorithm 1. Note that algorithms float like figures and tables.

Algorithm 1: The Gauss-Seidel Algorithm

1. For $k = 1$ to maximum number of iterations
   (a) For $i = 1$ to n
      i. $x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}$
      ii. If $\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)} < \epsilon\|$, where ϵ is a specified stopping criteria, stop.

You can use `\caption` and `\label` without using `\floatconts` (as in Algorithm 2).

If you'd rather have the same numbering throughout the algorithm but still want the convenient indentation of nested `enumerate` environments, you can use the `enumerate*` environment provided by the jmlr class. For example, see Algorithm 2.

Algorithm 2: Moore's Shortest Path
Given a connected graph G, where the length of each edge is 1:

1. Set the label of vertex s to 0
2. Set $i = 0$
   3. Locate all unlabelled vertices adjacent to a vertex labelled i and label them $i+1$
   4. If vertex t has been labelled,
         the shortest path can be found by backtracking, and the length is given by the label of t.
      otherwise
         increment i and return to step 3

Pseudo code can be displayed using the `algorithm2e` environment. This is defined by the algorithm2e package (which is automatically loaded) so check the algorithm2e documentation for further details.[5] For an example, see Algorithm 3.

Algorithm 3: Computing Net Activation

Input: $x_1, \ldots, x_n, w_1, \ldots, w_n$
Output: y, the net activation
$y \leftarrow 0$;
for $i \leftarrow 1$ **to** n **do**
  $y \leftarrow y + w_i * x_i$;
end

13. Description Lists

The jmlr class also provides a description-like environment called `altdescription`. This has an argument that should be the widest label in the list. Compare:

add A method that adds two variables.

differentiate A method that differentiates a function.

with

add A method that adds two variables.

differentiate A method that differentiates a function.

14. Theorems, Lemmas etc

The following theorem-like environments are predefined by the jmlr class: `theorem`, `example`, `lemma`, `proposition`, `remark`, `corollary`, `definition`, `conjecture` and `axiom`. You can use the `proof` environment to display the proof if need be, as in Theorem 1.

Theorem 1 (Eigenvalue Powers) *If λ is an eigenvalue of $\boldsymbol{B}$ with eigenvector $\boldsymbol{\xi}$, then λ^n is an eigenvalue of $\boldsymbol{B}^n$ with eigenvector $\boldsymbol{\xi}$.*
Proof *Let λ be an eigenvalue of $\boldsymbol{B}$ with eigenvector ξ, then*

$$\boldsymbol{B\xi} = \lambda\boldsymbol{\xi}$$

premultiply by $\boldsymbol{B}$:

$$\begin{aligned} \boldsymbol{BB\xi} &= \boldsymbol{B}\lambda\boldsymbol{\xi} \\ \Rightarrow \boldsymbol{B}^2\boldsymbol{\xi} &= \lambda\boldsymbol{B\xi} \\ &= \lambda\lambda\boldsymbol{\xi} \qquad \textit{since } \boldsymbol{B\xi} = \lambda\boldsymbol{\xi} \\ &= \lambda^2\boldsymbol{\xi} \end{aligned}$$

5. Either `texdoc algorithm2e` or http://www.ctan.org/pkg/algorithm2e

Therefore true for $n = 2$*. Now assume true for* $n = k$*:*

$$\boldsymbol{B}^k \boldsymbol{\xi} = \lambda^k \boldsymbol{\xi}$$

premultiply by $\boldsymbol{B}$*:*

$$\begin{aligned} \boldsymbol{B}\boldsymbol{B}^k \boldsymbol{\xi} &= \boldsymbol{B}\lambda^k \boldsymbol{\xi} \\ \Rightarrow \boldsymbol{B}^{k+1} \boldsymbol{\xi} &= \lambda^k \boldsymbol{B}\boldsymbol{\xi} \\ &= \lambda^k \lambda \boldsymbol{\xi} \qquad \text{since } \boldsymbol{B}\boldsymbol{\xi} = \lambda \boldsymbol{\xi} \\ &= \lambda^{k+1} \boldsymbol{\xi} \end{aligned}$$

Therefore true for $n = k + 1$*. Therefore, by induction, true for all* n*.* ■

Lemma 2 (A Sample Lemma) *This is a lemma.*

Remark 3 (A Sample Remark) *This is a remark.*

Corollary 4 (A Sample Corollary) *This is a corollary.*

Definition 5 (A Sample Definition) *This is a definition.*

Conjecture 6 (A Sample Conjecture) *This is a conjecture.*

Axiom 7 (A Sample Axiom) *This is an axiom.*

Example 1 (An Example) *This is an example.*

15. Color vs Grayscale

It's helpful if authors supply grayscale versions of their figures in the event that the article is to be incorporated into a black and white printed book. With external PDF, PNG or JPG graphic files, you just need to supply a grayscale version of the file. For example, if the file is called `myimage.png`, then the gray version should be `myimage-gray.png` or `myimage-gray.pdf` or `myimage-gray.jpg`. You don't need to modify your code. The jmlr class checks for the existence of the grayscale version if it is print mode (provided you have used `\includegraphics` and haven't specified the file extension).

You can use `\ifprint` to determine which mode you are in. For example, in Figure **??**, the purple ellipse represents an input and the yellow ellipse represents an output. Another example: important text!

You can use the class option `gray` to see how the document will appear in gray scale mode. Colored text will automatically be converted to gray scale.

The jmlr class loads the xcolor package, so you can also define your own colors. For example: XYZ.

The xcolor class is loaded with the `x11names` option, so you can use any of the x11 predefined colors (listed in the xcolor documentation[6]).

6. either `texdoc xcolor` or `http://www.ctan.org/pkg/xcolor`

16. Citations and Bibliography

The jmlr class automatically loads natbib. This sample file has the citations defined in the accompanying BibTeX file `jmlr-sample.bib`. For a parenthetical citation use `\citep`. For example (Guyon and Elisseeff, 2003). For a textual citation use `\citet`. For example Guyon et al. (2007). Both commands may take a comma-separated list, for example Guyon and Elisseeff (2003); Guyon et al. (2007).

These commands have optional arguments and have a starred version. See the natbib documentation for further details.[7]

The bibliography is displayed using `\bibliography`.

Acknowledgments

Our thanks to the University College Maastricht, Maastricht University, the Institute of Data Science, and the Department of Data Science and Knowledge Engineering, in particular to Peter Vermeer for initiating the project and enabling collaboration with the University College Maastricht.

References

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3: 1157–1182, March 2003.

I. Guyon, C. Aliferis, and A. Elisseeff. Causal feature selection. Technical report, Clopinet, 2007.

Appendix A. First Appendix

This is the first appendix.

Appendix B. Second Appendix

This is the second appendix.

7. Either `texdoc natbib` or http://www.ctan.org/pkg/natbib