Documentation for `msoelch.sty`

Maximilian Soelch

June 16, 2016

The `msoelch` package provides convenient and readable macros for commonly used syntax when writing scientific articles about machine learning.

1 The `\variables{}` macro

A clear nomenclature is a necessity for good scientific writing. Macros can help a great deal to stick to one convention while maintaining readability of the source file. However, with an increasing number of conventions, maintenance of the macros becomes just as hard.

The `\variables{}` macro eases some of that pain. It is called in the preamble with a list of variables and dynamically creates macros based on this list. For example, the call `\variables{x,y,A}` automatically provides the following macros:

x	`\bx`	`\xt`	`\bxt`	`\xT`	`\bxT`	`\xts`	`\bxts`	`\xTs`	`\bxTs`
	$\mathbf{x}$	x_t	$\mathbf{x}_t$	x_T	$\mathbf{x}_T$	$x_{1:t}$	$\mathbf{x}_{1:t}$	$x_{1:T}$	$\mathbf{x}_{1:T}$
y	`\by`	`\yt`	`\byt`	`\yT`	`\byT`	`\yts`	`\byts`	`\yTs`	`\byTs`
	$\mathbf{y}$	y_t	$\mathbf{y}_t$	y_T	$\mathbf{y}_T$	$y_{1:t}$	$\mathbf{y}_{1:t}$	$y_{1:T}$	$\mathbf{y}_{1:T}$
A	`\bA`	`\At`	`\bAt`	`\AT`	`\bAT`	`\Ats`	`\bAts`	`\ATs`	`\bATs`
	$\mathbf{A}$	A_t	$\mathbf{A}_t$	A_T	$\mathbf{A}_T$	$A_{1:t}$	$\mathbf{A}_{1:t}$	$A_{1:T}$	$\mathbf{A}_{1:T}$

The driving idea behind these macros is the application to time series of scalars (normal font), vectors (bold font) and matrices (upper-case bold font). These macros can easily be edited and/or extended by adjusting the `msoelch.sty` file.

2 The `\probdists{}` macro

The `\probdists{}` macro provides a rich, overloaded macro for probability distributions.

The call `\probdists{p}` provides a command that can be used in any of the following ways:

`\p`	`\p{x}`	`\p{x}{z}`	`\p{x}{z}{y}`	`\p{x}{}{y}`	`\p{}{}{y}`
p	$p(x)$	$p(x \mid z)$	$p_y(x \mid z)$	$p_y(x)$	p_y

The `\p` macro dynamically decides which parts to display. This allows for rapid adjustment of the syntax. The parentheses adjust dynamically via internal usage of `\mleft` and `\mright`:

$$p\left(\frac{x}{y}\right)$$

Moreover, the `\p` macro ignores any argument not given in set parentheses, e. g., `\p{x}{z}y` yields $p(x \mid z)y$.

Of course, `probdists` can be called with a list of letters, e. g., `\probdists{p,q}`. At this point, more complicated constructions like $\mathbb{P}$ etc. are not supported since the letter is directly used for macro name construction.

3 The `\expc` and `\var` macros

For expectation and variance, this package provides overloaded macros similar to the dynamic macros provided by `\probdists{}` in section 2.

The applicaion is straighforward:

`\expc{x}`	`\expc{x}{z}`	`\expc{x}{z}{y}`	`\expc{x}{}{y}`
$\mathbb{E}[x]$	$\mathbb{E}[x \mid z]$	$\mathbb{E}_y[x \mid z]$	$\mathbb{E}_y[x]$
`\var{x}`	`\var{x}{z}`	`\var{x}{z}{y}`	`\var{x}{}{y}`
$\operatorname{Var}[x]$	$\operatorname{Var}[x \mid z]$	$\operatorname{Var}_y[x \mid z]$	$\operatorname{Var}_y[x]$

All comments from section 2 apply here as well.

4 Syntactic sugar

4.1 Math

Some general, commonly used math expressions are provided by short, readable macros:

- `mathbb` wrappers: `\NN` for $\mathbb{N}$, `\RR` for $\mathbb{R}$, `\PP` for $\mathbb{P}$, `\EE` for $\mathbb{E}$.
- `mathcal` wrappers: `\mcX` for $\mathcal{X}$, `\mcU` for $\mathcal{U}$, `\mcZ` for $\mathcal{Z}$.

- `\left\right` wrappers:

  `\abs{}`: $\left|\frac{a}{b}\right|$ `\set{}`: $\left\{\frac{a}{b}\right\}$ `\interval{}`: $\left[\frac{a}{b}\right]$

- Loss functions: `\loss[index]{arg}` for $\mathcal{L}_{index}(arg)$. The parentheses adjust to the height of the arguments.
- Kullback-Leibler divergence: `\kl{p}{q}` for $\mathrm{KL}(p \parallel q)$. The parentheses adjust to the height of the arguments.
- Gaussian/Normal distributions: `\gauss{0,1}` for $\mathcal{N}(0,1)$. The parentheses adjust to the height of the arguments.
- Integrals: `\dint` for nicer integrals:

  `\int x \dint x`: $\int x\,dx$ `\int x dx`: $\int x dx$

- Equations that are to be proven: `x \shallbe y` for $x \overset{!}{=} y$.

4.2 Other

One of the most cumbersome typesetting issues is the correct spacing after abbreviations.

This packages provides macros that automatically apply correct spacing and follow →this guide for

- *with respect to*: `derivative \wrt input` ⇝ derivative w. r. t. input,
- *exempli gratia*: `algrorithms, \eg EM` ⇝ algorithms, e. g., EM; also `\Eg` for E. g.,
- *id est*: `156 members, \ie almost all` ⇝ 156 members, i. e., almost all; also `\Ie` for I. e.,
- *et cetera*: `Germany, France, \etc` ⇝ Germany, France, etc.

The latter command checks if a period follows, i. e., `\dots Germany, France, \etc. Other countries \dots` yields ... Germany, France, etc. Other countries ...

5 Style of this document

The default style of this document uses Palantino as the main font via

```
\usepackage[osf,sc]{mathpazo}
 \linespread{1.05}\selectfont
```

with some extra spacing between lines. Moreover, the Euler math font is used:

```
\usepackage[euler-digits]{eulervm}
```

The `amsmath` package is loaded with the `fleqn` for left flush of equations.

All these changes were taken from the highly recommended →`classicthesis` package by André Miede.

Other recommended nice-to-have packages for convenient and good-looking typewriting are `cleveref` for extremely clever referencing, `nicefrac` for better handling of inline fractions, and `microtype`, which takes care of small typesetting issues that improve the overall document appearance.

6 TEXstudio autocomplete

Along with this style file and example document, a `cwl` file is provided. By adding it to your TEXstudio, autocomplete for the static macros is provided. At this point, autocomplete of the dynamically produced `variables` and `probdists` is not included.

Under Windows, it has to be copied to the directory

`%appdata%\texstudio\completion\user`.