

abaisero.sty

Andrea Baisero

October 21, 2024

1 Options

Options are processes left-to-right. If no options are provided, or none are enabled by the end of the processing, then by default they are all considered to be enabled.

Option	Description
<code>all</code>	Enable all commands
<code>(no-)math</code>	Disable/enable mathematical commands
<code>(no-)linalg</code>	Disable/enable linear algebra commands
<code>(no-)optim</code>	Disable/enable optimization commands
<code>(no-)stats</code>	Disable/enable statistics commands
<code>(no-)dists</code>	Disable/enable distributions commands
<code>(no-)ml</code>	Disable/enable machine learning commands
<code>(no-)rl</code>	Disable/enable reinforcement learning commands
<code>(no-)marl</code>	Disable/enable multi-agent reinforcement learning commands
<code>(no-)theorem</code>	Disable/enable theorem commands
<code>(no-)misc</code>	Disable/enable miscellanea commands

2 Commands

Option [math]

Symbol	Command	Description	Example
\mathbb{N}	<code>\naturalset</code>	the set of natural numbers	$\mathbb{N} \doteq \{1, 2, 3, \dots\}$
\mathbb{Z}	<code>\integerset</code>	the set of integer numbers	$\mathbb{Z} \doteq \{0, 1, -1, 2, -3, \dots\}$
\mathbb{R}	<code>\realset</code>	the set of real numbers	$\sqrt{2} \in \mathbb{R}$
$*$	<code>\kstar</code>	the Kleene star operator	$\mathcal{X}^* \doteq \bigcup_{k=0}^{\infty} \mathcal{X}^k$
$+$	<code>\kplus</code>	the Kleene plus operator	$\mathcal{X}^+ \doteq \bigcup_{k=1}^{\infty} \mathcal{X}^k$
\mathcal{P}	<code>\powerset</code>	power set	$\mathcal{P}(\mathcal{X}) \doteq \{\hat{\mathcal{X}} \mid \hat{\mathcal{X}} \subseteq \mathcal{X}\}$
\mathbb{I}	<code>\Ind</code>	Indicator function	$\Pr(x = 0) = \mathbb{E}[\mathbb{I}[x = 0]]$
$\llbracket \cdot \rrbracket$	<code>\iverson{\cdot}</code>	Iverson brackets	$\llbracket x \geq 0 \rrbracket$ maps x to binaries
softmax	<code>\softmax</code>	a.k.a. logsumexp, realsoftmax ¹	$\text{softmax}(x_1, \dots, x_n) \doteq \log \sum_i \exp(x_i)$
softmaxmin	<code>\softmaxmin</code>		$\text{softmaxmin}(x_1, \dots, x_n) \doteq -\log \sum_i \exp(-x_i)$
softargmax	<code>\softargmax</code>	a.k.a. softmax ¹	$\text{softargmax}(x_1, \dots, x_n)_i \doteq \frac{\exp(x_i)}{\sum_k \exp(x_k)}$
sign	<code>\sign</code>		$x = \text{sign } x \cdot x $
supp	<code>\supp</code>	support operator	$\text{supp}(f) \doteq \{x \mid f(x) \neq 0\}$

Option [linalg]

Symbol	Command	Description	Example
diag	<code>\diag</code>		
rank	<code>\rank</code>		
tr	<code>\trace</code>		$\text{tr}(M) \doteq \sum_{i=1}^n M_{ii}$
col	<code>\colspace</code>		
ker	<code>\nullspace</code>	Nullspace (a.k.a kernel) of a linear mapping	
span	<code>\spanspace</code>		
$^\top$	<code>\T</code>	Transpose superscript	symmetric $M \implies M = M^\top$
$^{-1}$	<code>\I</code>	Inverse superscript	invertible $M \implies MM^{-1} = I$
$+$	<code>\PI</code>	Pseudo-inverse superscript	$MM^+M = M$
$^{-\top}$	<code>\IT</code>	Inverse transpose superscript	$M^{-\top} = (M^{-1})^\top = (M^\top)^{-1}$
$^{+\top}$	<code>\PIT</code>	Pseudo-inverse transpose superscript	$M^{+\top} = (M^+)^\top = (M^\top)^+$

Option [optim]

Symbol	Command	Description	Example
argmax	<code>\argmax</code>		$\text{argmax}_a Q^\pi(s, a)$
argmin	<code>\argmin</code>		$\theta^* \doteq \text{argmin}_\theta \mathcal{L}(\theta)$
$*$	<code>\opt</code>	Optimality superscript	$\pi^*(s) = \text{argmax}_a Q^*(s, a)$
stop	<code>\stopg</code>	Stop-gradient operator	$\nabla_x \text{stop}[f(x)] = 0$

¹The functions that in this document are called “softmax” and “softargmax” are poorly and inaccurately named in the broader math and ML fields (see https://en.wikipedia.org/wiki/Softmax_function and <https://en.wikipedia.org/wiki/LogSumExp>). Rather than stick to the more common naming conventions, I opt to rename the functions more accurately to appropriately reflect their actual properties. In any document where I would use these functions, I would need to define them anyway, so the risk of misunderstandings are minimal.

Option [stats]

Symbol	Command	Description	Example
\perp	<code>\indep</code>	Independence	$X \perp Y \mid Z$
do	<code>\causaldo</code>	Pearl's <i>do</i> operator	$\Pr(Y \mid \text{do}(X = x)) \neq \Pr(Y \mid X = s)$
\mathbb{C}	<code>\Cov</code>	Covariance	$\mathbb{C}(x, y) = \mathbb{E}[xy] - \mathbb{E}[x] \mathbb{E}[y]$
\mathbb{H}	<code>\Ent</code>	Entropy	$\mathbb{H}[x] = -\mathbb{E}[\log \Pr(x)]$
\mathbb{E}	<code>\Exp</code>	Expectation	$\mathbb{E}[f(x)] = \sum_x \Pr(x) f(x)$
KL	<code>\KL</code>	KL-divergence	$\text{KL}(p \parallel q) \doteq \mathbb{E}_{x \sim p}[\log p(x) - \log q(x)]$
D_{KL}	<code>\DKL</code>	KL-divergence (alternative)	
\mathbb{I}	<code>\MI</code>	Mutual Information	
\mathbb{B}	<code>\Bias</code>	Bias	$\mathbb{B}[\hat{f}(x)]$ is the bias of estimator \hat{f}
\mathbb{V}	<code>\Var</code>	Variance	$\mathbb{V}[\hat{f}(x)] = \mathbb{E}[\hat{f}(x)^2] - \mathbb{E}[\hat{f}(x)]^2$

Option [dists]

Symbol	Command	Description	Example
Categorical	<code>\Categorical</code>	Categorical	
Dirichlet	<code>\Dirichlet</code>	Dirichlet	
Geometric	<code>\Geometric</code>	Geometric	
Normal	<code>\Normal</code>	Normal	
Uniform	<code>\Uniform</code>	Uniform	

Option [ml]

Symbol	Command	Description	Example
\mathcal{D}	<code>\data</code>	Data set	$\mathcal{D} \doteq \{(x_i, y_i)\}_{i=1}^N$
\mathcal{L}	<code>\loss</code>	Loss function	$\mathcal{L}(\theta; x, y) = \frac{1}{2} \ y - f(x; \theta)\ ^2$
nll	<code>\nll</code>	Neg-log-likelihood	$\text{nll}(x) \doteq -\log \Pr(x)$
MSE	<code>\mse</code>	Mean-squared-error	
\rightsquigarrow	<code>\trainedto</code>	Model is trained to approximate some value	$\hat{V}(s) \rightsquigarrow V^\pi(s) \xrightarrow{\text{hope}} \hat{V}(s) \approx V^\pi(s)$

Option [rl]

Symbol	Command	Description	Example
\mathcal{A}	<code>\aset</code>	Action set	
\mathcal{B}	<code>\bset</code>	Belief set	
\mathcal{H}	<code>\hset</code>	History set	
\mathcal{O}	<code>\oset</code>	Observation set	
\mathcal{R}	<code>\rset</code>	Reward set	
\mathcal{S}	<code>\sset</code>	State set	
ε	<code>\nohistory</code>	Empty history	
π	<code>\policy</code>	policy	
Q^π	<code>\qpolicy</code>	Q policy values	
Q^μ	<code>\qpolicy[\mu]</code>	Q policy values	
\hat{Q}	<code>\qmodel</code>	Parametric model	
V^π	<code>\vpolicy</code>	V policy values	
V^μ	<code>\vpolicy[\mu]</code>	V policy values	
\hat{V}	<code>\vmodel</code>	Parametric model	
A^π	<code>\apolicy</code>	A policy values	
A^μ	<code>\apolicy[\mu]</code>	A policy values	
\hat{A}	<code>\amodel</code>	Parametric model	
U^π	<code>\upolicy</code>	U policy values	
U^μ	<code>\upolicy[\mu]</code>	U policy values	
\hat{U}	<code>\umodel</code>	Parametric model	
B_π	<code>\bpolicy</code>	Policy Bellman operator	
B_μ	<code>\bpolicy[\mu]</code>	Policy Bellman operator	

Option [marl]

Symbol	Command	Description	Example
x	<code>\joint{x}</code>	Joint formatting (redefinable)	
\mathcal{A}	<code>\jaset</code>	Joint action set	
\mathcal{O}	<code>\joset</code>	Joint observation set	
\mathcal{H}	<code>\jhset</code>	Joint history set	
A, a	<code>\jA, \ja</code>	Joint action	$A = a \in \mathcal{A}$
O, o	<code>\jO, \jo</code>	Joint observation	$O = o \in \mathcal{O}$
H, h	<code>\jH, \jh</code>	Joint history	$H = h \in \mathcal{H}$
b	<code>\jb</code>	Joint belief	$b: \mathcal{H} \rightarrow \Delta \mathcal{S}$
π	<code>\jpolicy</code>	Joint policy	$\pi(h, a) \doteq \prod_i \pi_i(h_i, a_i)$
Q^π	<code>\jqpolicy</code>	Q joint-policy values	
V^π	<code>\jvpolicy</code>	V joint-policy values	
A^π	<code>\japolicy</code>	A joint-policy values	
U^π	<code>\jupolicy</code>	U joint-policy values	
B_π	<code>\jbpolicy</code>	Joint policy Bellman operator	
B_μ	<code>\jbpolicy[\joint\mu]</code>	Joint policy Bellman operator	
f_{mix}	<code>\fmix</code>	Mixing function	$f_{\text{mix}}(h, a, Q_1(h_1), \dots, Q_n(h_n))$
f_{mono}	<code>\fmono</code>	Monotonic mixing function	$f_{\text{mono}}(u_1, \dots, u_n)$
f_{IGM}	<code>\figm</code>	IGM mixing function	$f_{\text{IGM}}(h, a, Q_1(h_1), \dots, Q_n(h_n))$
\hat{Q}_{VDN}	<code>\qvdn</code>	VDN Q-model	
\hat{V}_{VDN}	<code>\vvdn</code>	VDN V-model	
\hat{A}_{VDN}	<code>\avdn</code>	VDN A-model	
\hat{Q}_{MIX}	<code>\qmix</code>	QMIX Q-model	
\hat{V}_{MIX}	<code>\vmix</code>	QMIX V-model	
\hat{A}_{MIX}	<code>\amix</code>	QMIX A-model	
\hat{Q}_{TRAN}	<code>\qtran</code>	QTRAN Q-model	
\hat{V}_{TRAN}	<code>\vtran</code>	QTRAN V-model	
\hat{A}_{TRAN}	<code>\atran</code>	QTRAN A-model	
\hat{Q}_{WMIX}	<code>\wqmix</code>	WQMIX Q-model	
\hat{V}_{WMIX}	<code>\wvmix</code>	WQMIX V-model	
\hat{A}_{WMIX}	<code>\wamix</code>	WQMIX A-model	
\hat{Q}_{PLEX}	<code>\qplex</code>	QPLEX Q-model	
\hat{V}_{PLEX}	<code>\vplex</code>	QPLEX V-model	
\hat{A}_{PLEX}	<code>\aplex</code>	QPLEX A-model	
\hat{Q}_{FIX}	<code>\qfix</code>	QFIX Q-model	
\hat{V}_{FIX}	<code>\vfix</code>	QFIX V-model	
\hat{A}_{FIX}	<code>\afix</code>	QFIX A-model	

Option [theorem]

Symbol	Command	Description	Example
	<code>\begin{definition}, \end{definition}</code> <code>\begin{assumption}, \end{assumption}</code> <code>\begin{example}, \end{example}</code>		
	<code>\begin{axiom}, \end{axiom}</code> <code>\begin{conjecture}, \end{conjecture}</code> <code>\begin{proposition}, \end{proposition}</code> <code>\begin{lemma}, \end{lemma}</code> <code>\begin{theorem}, \end{theorem}</code> <code>\begin{corollary}, \end{corollary}</code> <code>\begin{generalization}, \end{generalization}</code>		

Option [misc]

Symbol	Command	Description	Example
\dagger	<code>\D</code>	Dagger superscript	
(k)	<code>\iter{k}</code>	Superscript indicating iteration	

Option [utils]

Symbol	Command	Description	Example
$$	<code>\phantomeq</code>	The width of an =, for alignment purposes (bounding box shown)	