`latex-math` Macros

Latex macros like `\frac{#1}{#2}` with arguments are displayed as $\frac{\#1}{\#2}$.

basic-math.tex

Macro	Notation	Comment
`{\N}`	$\mathbb{N}$	N defined by "siunitx" (which we use), for "NEWTON"
`{\N}`	$\mathbb{N}$	
`{\Z}`	$\mathbb{Z}$	Z, integers
`{\Q}`	$\mathbb{Q}$	Q, rationals
`{\R}`	$\mathbb{R}$	R, reals
`{\C}`	$\mathbb{C}$	C, complex
`{\C}`	$\mathbb{C}$	
`{\HS}`	H	H, hilbertspace
`{\continuous}`	$\mathcal{C}$	C, space of continuous functions
`{\M}`	$\mathcal{M}$	machine numbers
`{\epsm}`	ϵ_m	maximum error
`{\xt}`	$\tilde{x}$	x tilde
`{\sign}`	sign	sign, signum
`{\I}`	$\mathbb{I}$	I, indicator
`{\order}`	$\mathcal{O}$	O, order
`{\fp}`	$\frac{\partial \#1}{\partial \#2}$	partial derivative
`{\pd}`	$\frac{\partial \#1}{\partial \#2}$	partial derivative
`{\sumin}`	$\sum_{i=1}^n$	summation from i=1 to n
`{\sumkg}`	$\sum_{k=1}^g$	summation from k=1 to g
`{\meanin}`	$\frac{1}{n}\sum_{i=1}^n$	mean from i=1 to n
`{\meankg}`	$\frac{1}{g}\sum_{k=1}^g$	mean from k=1 to g
`{\prodin}`	$\prod_{i=1}^n$	product from i=1 to n
`{\prodkg}`	$\prod_{k=1}^g$	product from k=1 to g
`{\one}`	$\mathbf{1}$	1, unitvector
`{\zero}`	$\mathbf{0}$	0-vector
`{\id}`	$\boldsymbol{I}$	I, identity
`{\diag}`	diag	diag, diagonal
`{\trace}`	tr	tr, trace
`{\spn}`	span	span
`{\scp}`	$\langle \#1, \#2 \rangle$	<.,.>, scalarproduct
`{\Amat}`	$\mathbf{A}$	matrix A
`{\xv}`	$\mathbf{x}$	vector x (bold)
`{\yv}`	$\mathbf{y}$	vector y (bold)
`{\Deltab}`	$\mathbf{\Delta}$	error term for vectors
`{\P}`	$\mathbb{P}$	P, probability
`{\E}`	$\mathbb{E}$	E, expectation
`{\var}`	Var	Var, variance
`{\cov}`	Cov	Cov, covariance
`{\corr}`	Corr	Corr, correlation
`{\normal}`	$\mathcal{N}$	N of the normal distribution
`{\iid}`	$\overset{i.i.d}{\sim}$	dist with i.i.d superscript
`{\distas}`	$\overset{\#1}{\sim}$	... is distributed as ...

basic-ml.tex

Macro	Notation	Comment
`{\Xspace}`	$\mathcal{X}$	X, input space
`{\Yspace}`	$\mathcal{Y}$	Y, output space
`{\nset}`	$\{1,\ldots,n\}$	set from 1 to n
`{\pset}`	$\{1,\ldots,p\}$	set from 1 to p
`{\gset}`	$\{1,\ldots,g\}$	set from 1 to g
`{\Pxy}`	$\mathbb{P}_{xy}$	P_xy
`{\Exy}`	$\mathbb{E}_{xy}$	E_xy: Expectation over random variables xy
`{\xy}`	$(\mathbf{x}, y)$	observation (x, y)
`{\xvec}`	$(x_1,\ldots,x_p)^T$	(x1, ..., xp)
`{\xb}`	$\mathbf{x}$	x (bold) feature vector
`{\Xmat}`	$\mathbf{X}$	Design matrix
`{\D}`	$\mathcal{D}$	D, data
`{\ydat}`	$\mathbf{y}$	y (bold), vector of outcomes
`{\yvec}`	$\left(y^{(1)},\ldots,y^{(n)}\right)^T$	(y1, ..., yn), vector of outcomes
`{\xi}`	$\mathbf{x}^{(\#1)}$	x^i, i-th observed value of x
`{\yi}`	$y^{(\#1)}$	y^i, i-th observed value of y
`{\xyi}`	$\left(\mathbf{x}^{(\#1)}, y^{(\#1)}\right)$	(x^i, y^i), i-th observation
`{\xivec}`	$\left(x_1^{(i)},\ldots,x_p^{(i)}\right)^T$	(x1^i, ..., xp^i), i-th observation vector
`{\xj}`	x_j	x_j, j-th feature
`{\xjb}`	$\mathbf{x}_j$	x_j (bold), j-th feature vecor
`{\xjvec}`	$\left(x_j^{(1)},\ldots,x_j^{(n)}\right)^T$	(x^1_j, ..., x^n_j), j-th feature vector
`{\Dtrain}`	$\mathcal{D}_{\text{train}}$	D_train, training set
`{\Dtest}`	$\mathcal{D}_{\text{test}}$	D_test, test set
`{\phiv}`	ϕ	Basis transformation function phi
`{\phixi}`	$\phi^{(i)}$	Basis transformation of xi: phi^i := phi(xi)
`{\inducer}`	$\mathcal{I}$	Inducer, inducing algorithm, learning algorithm
`{\ftrue}`	f_{true}	True underlying function (if a statistical model is assumed)
`{\ftruex}`	$f_{\text{true}}(\mathbf{x})$	True underlying function (if a statistical model is assumed)
`{\fx}`	$f(\mathbf{x})$	f(x), continuous prediction function
`{\Hspace}`	$\mathcal{H}$	hypothesis space where f is from
`{\fh}`	$\hat{f}$	f hat, estimated prediction function
`{\fxh}`	$\hat{f}(\mathbf{x})$	fhat(x)
`{\fxt}`	$f(\mathbf{x} \mid \boldsymbol{\theta})$	f(x | theta)
`{\fxi}`	$f\left(\mathbf{x}^{(i)}\right)$	f(x^(i))
`{\fxih}`	$\hat{f}\left(\mathbf{x}^{(i)}\right)$	f(x^(i))
`{\fxit}`	$f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)$	f(x^(i) | theta)
`{\fhD}`	$\hat{f}_{\mathcal{D}}$	fhat_D, estimate of f based on D
`{\fhDtrain}`	$\hat{f}_{\mathcal{D}_{\text{train}}}$	fhat_Dtrain, estimate of f based on D
`{\hx}`	$h(\mathbf{x})$	h(x), discrete prediction function
`{\hxv}`	$h(\mathbf{x})$	h(x), discrete prediction function with x (vector) as input
`{\hh}`	$\hat{h}$	h hat
`{\hxh}`	$\hat{h}(\mathbf{x})$	hhat(x)
`{\hxt}`	$h(\mathbf{x}\|\boldsymbol{\theta})$	h(x | theta)
`{\hxi}`	$h\left(\mathbf{x}^{(i)}\right)$	h(x^(i))
`{\hxit}`	$h\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)$	h(x^(i) | theta)
`{\yh}`	$\hat{y}$	yhat for prediction of target
`{\yih}`	$\hat{y}^{(i)}$	yhat^(i) for prediction of ith targiet
`{\thetah}`	$\hat{\boldsymbol{\theta}}$	

`{\thetab}`	$\boldsymbol{\theta}$	theta vector
`{\thetabh}`	$\hat{\boldsymbol{\theta}}$	theta vector
`{\pdf}`	p	p
`{\pdfx}`	$p(\mathbf{x})$	p(x)
`{\pixt}`	$\pi(\mathbf{x} \mid \boldsymbol{\theta})$	pi(x|theta), pdf of x given theta
`{\pixit}`	$\pi\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)$	pi(x^i|theta), pdf of x given theta
`{\pdfxy}`	$p(\mathbf{x}, y)$	p(x, y)
`{\pdfxyt}`	$p(\mathbf{x}, y \mid \boldsymbol{\theta})$	p(x, y | theta)
`{\pdfxyit}`	$p\left(\mathbf{x}^{(i)}, y^{(i)} \mid \boldsymbol{\theta}\right)$	p(x^(i), y^(i) | theta)
`{\pdfxyk}`	$p(x\|y = k)$	p(x | y = k)
`{\lpdfxyk}`	$\log p(x\|y = k)$	log p(x | y = k)
`{\pdfxiyk}`	$p\left(\mathbf{x}^{(i)}\|y = k\right)$	p(x^i | y = k)
`{\pik}`	π_k	pi_k, prior
`{\lpik}`	$\log \pi_k$	log pi_k, log of the prior
`{\pit}`	$\pi(\boldsymbol{\theta})$	Prior probability of parameter theta
`{\post}`	$\mathbb{P}(y = 1 \mid \mathbf{x})$	P(y = 1 | x), post. prob for y=1
`{\pix}`	$\pi(\mathbf{x})$	pi(x), P(y = 1 | x)
`{\postk}`	$\mathbb{P}(y = k \mid \mathbf{x})$	P(y = k | y), post. prob for y=k
`{\pikx}`	$\pi_k(\mathbf{x})$	pi_k(x), P(y = k | x)
`{\pikxt}`	$\pi_k(\mathbf{x} \mid \boldsymbol{\theta})$	pi_k(x | theta), P(y = k | x, theta)
`{\pijx}`	$\pi_j(\mathbf{x})$	pi_j(x), P(y = j | x)
`{\pdfygxt}`	$p(y \mid \mathbf{x}, \boldsymbol{\theta})$	p(y | x, theta)
`{\pdfyigxit}`	$p\left(y^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta}\right)$	p(y^i |x^i, theta)
`{\lpdfygxt}`	$\log p(y \mid \mathbf{x}, \boldsymbol{\theta})$	log p(y | x, theta)
`{\lpdfyigxit}`	$\log p\left(y^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta}\right)$	log p(y^i |x^i, theta)
`{\pixh}`	$\hat{\pi}(\mathbf{x})$	pi(x) hat, P(y = 1 | x) hat
`{\pikxh}`	$\hat{\pi}_k(\mathbf{x})$	pi_k(x) hat, P(y = k | x) hat
`{\pixih}`	$\hat{\pi}(\mathbf{x}^{(i)})$	pi(x^(i)) with hat
`{\pikxih}`	$\hat{\pi}_k(\mathbf{x}^{(i)})$	pi_k(x^(i)) with hat
`{\eps}`	ϵ	residual, stochastic
`{\epsi}`	$\epsilon^{(i)}$	epsilon^i, residual, stochastic
`{\epsh}`	$\hat{\epsilon}$	residual, estimated
`{\yf}`	$yf(\mathbf{x})$	y f(x), margin
`{\yfi}`	$y^{(i)} f\left(\mathbf{x}^{(i)}\right)$	y^i f(x^i), margin
`{\Sigmah}`	$\hat{\Sigma}$	estimated covariance matrix
`{\Sigmahj}`	$\hat{\Sigma}_j$	estimated covariance matrix for the j-th class
`{\Lxy}`	$L\left(y, f(\mathbf{x})\right)$	L(y, f(x)), loss function
`{\Lxyi}`	$L\left(y^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right)$	L(y^i, f(x^i))
`{\Lxyt}`	$L\left(y, f(\mathbf{x} \mid \boldsymbol{\theta})\right)$	L(y, f(x | theta))
`{\Lxyit}`	$L\left(y^{(i)}, f\left(\mathbf{x}^{(i)} \mid \boldsymbol{\theta}\right)\right)$	L(y^i, f(x^i | theta)
`{\Lxym}`	$L\left(y^{(i)}, f\left(\tilde{\boldsymbol{x}}^{(i)} \mid \boldsymbol{\theta}\right)\right)$	L(y^i, f(tilde(x)^i | theta),
`{\Lhxy}`	$L\left(y, h(\mathbf{x})\right)$	L(y, h(x)), loss function on discrete classes
`{\risk}`	$\mathcal{R}$	R, risk
`{\riskf}`	$\mathcal{R}(f)$	R(f), risk
`{\riskt}`	$\mathcal{R}(\boldsymbol{\theta})$	R(theta), risk
`{\riske}`	$\mathcal{R}_{\text{emp}}$	R_emp, empirical risk (without factor 1 / n
`{\riskeb}`	$\bar{\mathcal{R}}_{\text{emp}}$	R_emp, empirical risk with factor 1 / n
`{\riskef}`	$\mathcal{R}_{\text{emp}}(f)$	R_emp(f)
`{\risket}`	$\mathcal{R}_{\text{emp}}(\boldsymbol{\theta})$	R_emp(theta)
`{\riskr}`	$\mathcal{R}_{\text{reg}}$	R_reg, regularized risk
`{\riskrt}`	$\mathcal{R}_{\text{reg}}(\boldsymbol{\theta})$	R_reg(theta)
`{\riskrf}`	$\mathcal{R}_{\text{reg}}(f)$	R_reg(f)
`{\riskrth}`	$\hat{\mathcal{R}}_{\text{reg}}(\boldsymbol{\theta})$	hat R_reg(theta)

`{\risketh}`	$\hat{\mathcal{R}}_{\text{emp}}(\boldsymbol{\theta})$	hat R_emp(theta)
`{\LL}`	$\mathcal{L}$	L, likelihood
`{\LLt}`	$\mathcal{L}(\boldsymbol{\theta})$	L(theta), likelihood
`{\ll}`	ℓ	l, log-likelihood
`{\llt}`	$\ell(\boldsymbol{\theta})$	l(theta), log-likelihood
`{\LS}`	$\mathfrak{L}$	???????????
`{\TS}`	$\mathfrak{T}$	?????????????
`{\errtrain}`	$\text{err}_{\text{train}}$	training error
`{\errtest}`	err_{test}	training error
`{\errexp}`	$\overline{\text{err}_{\text{test}}}$	training error
`{\GEf}`	$GE\left(\hat{f}\right)$	Generalization error of a fitted model
`{\GEind}`	$GE_n\left(\mathcal{I}_{L,O}\right)$	Generalization error of a fitted model
`{\GE}`	$GE_n\left(\hat{f}_{\#1}\right)$	Generalization error GE
`{\GEh}`	$\widehat{GE}_{\#1}$	Estimated train error
`{\GED}`	$GE_n\left(\hat{f}_{\mathcal{D}}\right)$	Generalization error GE
`{\EGEn}`	EGE_n	Generalization error GE
`{\EDn}`	$\mathbb{E}_{\|D\|=n}$	Generalization error GE
`{\costs}`	$\mathcal{C}$	costs
`{\Celite}`	θ^*	elite configurations
`{\instances}`	$\mathcal{I}$	sequence of instances
`{\budget}`	$\mathcal{B}$	computational budget
`{\np}`	n_+	no. of positive instances
`{\nn}`	n_-	no. of negative instances
`{\rn}`	π_-	proportion negative instances
`{\rp}`	π_+	proportion negative instances
`{\tp}`	#TP	
`{\fap}`	#FP	fp taken for partial derivs
`{\tn}`	#TN	
`{\fan}`	#FN	

ml-bagging.tex

Macro	Notation	Comment
`{\bl}`	$b^{[\#1]}(x)$	baselearner with argument for m
`{\blm}`	$b^{[m]}(x)$	baselearner without argument for m
`{\blmh}`	$\hat{b}^{[m]}(x)$	estimated base learner
`{\fM}`	$f^{[M]}(x)$	ensembled predictor
`{\fMh}`	$\hat{f}^{[M]}(x)$	estimated ensembled predictor
`{\ambifM}`	$\Delta\left(f^{[M]}(x)\right)$	ambiguity/instability of ensemble

ml-bayesopt.tex

Macro	Notation	Comment
`{\minit}`	init	Size of the initial design
`{\lambdai}`	$[i]$	input for black box optimization
`{\lambdaopt}`	λ^*	Minimum of the black box function Psi
`{\metadata}`	$\left\{\left(\boldsymbol{\lambda}^{[i]}, \Psi^{[i]}\right)\right\}$	Metadata for the Gaussian process
`{\lambdavec}`	$\left(\lambda^{[1]}, \dots, \lambda^{[m_{\text{init}}]}\right)$	Vector of different inputs
`{\gp}`	$\mathcal{GP}\left(m(x), k\left(x, x'\right)\right)$	Gaussian Process

ml-boosting.tex

Macro	Notation	Comment
{\fm}	$f^{[m]}$	prediction in iteration m
{\fmh}	$\hat{f}^{[m]}$	prediction in iteration m
{\fmd}	$f^{[m-1]}$	prediction m-1
{\fmdh}	$\hat{f}^{[m-1]}$	prediction m-1
{\bmm}	$b^{[m]}$	basemodel m
{\bmmh}	$\hat{b}^{[m]}$	basemodel m with hat
{\betam}	$\beta^{[m]}$	weight of basemodel m
{\betamh}	$\hat{\beta}^{[m]}$	weight of basemodel m with hat
{\betai}	$\beta^{[\#1]}$	weight of basemodel with argument for m
{\errm}	$\mathrm{err}^{[m]}$	weighted in-sample misclassification rate
{\wm}	$w^{[m]}$	weight vector of basemodel m
{\wmi}	$w^{[m](i)}$	weight of obs i of basemodel m
{\thetam}	$\theta^{[m]}$	parameters of basemodel m
{\thetamh}	$\hat{\theta}^{[m]}$	parameters of basemodel m with hat
{\rmm}	$r^{[m]}$	pseudo residuals
{\rmi}	$r^{[m](i)}$	pseudo residuals
{\Rtm}	$R_t^{[m]}$	terminal-region
{\Tm}	$T^{[m]}$	
{\ctm}	$c_t^{[m]}$	mean, terminal-regions
{\ctmh}	$\hat{c}_t^{[m]}$	mean, terminal-regions with hat
{\ctmt}	$\tilde{c}_t^{[m]}$	mean, terminal-regions
{\fxk}	$f_k(x)$	f_k(x)
{\Lp}	L'	
{\Ldp}	L''	
{\Lpleft}	L'_{left}	

ml-gp.tex

Macro	Notation	Comment
`{\gp}`	$\mathcal{GP}\left(m(\boldsymbol{x}), k\left(\boldsymbol{x}, \boldsymbol{x}'\right)\right)$	Gaussian Process Definition
`{\mvec}`	$\mathbf{m}$	Gaussian process mean vector
`{\Kmat}`	$\mathbf{K}$	estimated base learner
`{\ls}`	ℓ	length-scale

ml-mbo.tex

Macro	Notation	Comment
`{\sxh}`	$\hat{s}(x)$	uncertainty shat(x)
`{\vxh}`	$\hat{s}^2(x)$	
`{\matK}`	$\mathbf{K}$	
`{\kstarx}`	$\mathbf{k}_*(x)$	
`{\xpi}`	$x^{*(\#1)}$	
`{\vhx}`	$\hat{s}^2(\mathbf{x})$	local estimated variance at point x
`{\shx}`	$\hat{s}(\mathbf{x})$	local estimated uncertainty at point x
`{\sh}`	$\hat{s}$	local estimated uncertainty
`{\px}`	$\boldsymbol{x}^*$	
`{\equote}`	"#1″	
`{\vecx}`	$\boldsymbol{x}$	
`{\yx}`	$y(\boldsymbol{x})$	
`{\X}`	$\mathcal{X}$	domain / search space
`{\yv}`	$\boldsymbol{y}$	
`{\fhx}`	$\hat{f}(\mathbf{x})$	surrogate (x), better use \mhx for predicted value
`{\minit}`	m_{init}	Size of the initial design
`{\lambdai}`	$\boldsymbol{\lambda}^{[i]}$	input for black box optimization
`{\lambdanew}`	$\boldsymbol{\lambda}^{\text{new}}$	new proposed configuration
`{\metadata}`	$\left\{\left(\boldsymbol{\lambda}^{[i]}, \Psi^{[i]}\right)\right\}$	Metadata for the Gaussian process
`{\lambdavec}`	$\boldsymbol{\lambda}^{[1]}, ..., \boldsymbol{\lambda}^{[m_{\text{init}}]}$	Vector of different inputs
`{\lambdab}`	$\boldsymbol{\lambda}$	input
`{\lambdaopt}`	$\boldsymbol{\lambda}^*$	Minimum of the black box function Psi

ml-nn.tex

Macro	Notation	Comment
`{\neurons}`	$z_1, \dots, z_M$	vector of neurons
`{\hidz}`	$\mathbf{z}$	vector of hidden activations
`{\biasb}`	$\mathbf{b}$	bias vector
`{\biasc}`	c	bias in output
`{\wtw}`	$\mathbf{w}$	weight vector (general)
`{\Wmat}`	$\mathbf{W}$	weight vector (general)
`{\wtu}`	$\mathbf{u}$	weight vector of output neuron
`{\Oreg}`	$R_{reg}(\theta\|X, y)$	regularized objective function
`{\Ounreg}`	$R_{emp}(\theta\|X, y)$	unconstrained objective function
`{\Pen}`	$\Omega(\theta)$	penalty
`{\Oregweight}`	$R_{reg}(w\|X, y)$	regularized objective function with weight
`{\Oweight}`	$R_{emp}(w\|X, y)$	unconstrained objective function with weight
`{\Oweighti}`	$R_{emp}(w_i\|X, y)$	unconstrained objective function with weight w_i
`{\Oweightopt}`	$J(w^*\|X, y)$	unconstrained objective function withoptimal weight
`{\Oopt}`	$\hat{J}(\theta\|X, y)$	optimal objective function
`{\Odropout}`	$J(\theta, \mu\|X, y)$	dropout objective function
`{\Lmomentumnest}`	$L(y^{(i)}, f(x^{(i)}, \boldsymbol{\theta} + \varphi\boldsymbol{\nu}))$	momentum risk
`{\Lmomentumtilde}`	$L(y^{(i)}, f(x^{(i)}, \tilde{\boldsymbol{\theta}}))$	Nesterov momentum risk
`{\Lmomentum}`	$L(y^{(i)}, f(x^{(i)}, \boldsymbol{\theta}))$	
`{\Hess}`	$\mathbf{H}$	
`{\nub}`	$\boldsymbol{\nu}$	
`{\uauto}`	$L(x, g(f(x)))$	undercomplete autoencoder objective function
`{\dauto}`	$L(x, g(f(\tilde{x})))$	denoising autoencoder objective function

ml-rf.tex

Macro	Notation	Comment
`{\betam}`	$\beta^{[m]}$	baselearner with argument for m
`{\betaM}`	$\beta^{[M]}$	baselearner with argument for M
`{\betai}`	$\beta^{[1]}$	baselearner with argument for 1

ml-svm.tex

Macro	Notation	Comment
`{\sv}`	SV	supportvectors
`{\HS}`	$\mathcal{H}$	H, hilbertspace
`{\sl}`	ζ	
`{\slvec}`	$\left(\zeta^{(1)}, \zeta^{(n)}\right)$	slack variables (SVM)
`{\sli}`	$\zeta^{(i)}$	slack variable (SVM)

ml-trees.tex

Macro	Notation	Comment
{\Np}	$\mathcal{N}$	(Parent) node N
{\Npk}	$\mathcal{N}_k$	node N__k
{\Nl}	$\mathcal{N}_1$	Left node N__1
{\Nr}	$\mathcal{N}_2$	Right node N__2
{\pikN}	$\pi_k^{(\mathcal{N})}$	class probability node N
{\pikNh}	$\hat{\pi}_{\#}1^{(\mathcal{N})}$	estimated class probability node N
{\pikNlh}	$\hat{\pi}_{\#}1^{(\mathcal{N}_1)}$	
{\pikNrh}	$\hat{\pi}_{\#}1^{(\mathcal{N}_2)}$	