

# latex-math Macros

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

## basic-math.tex

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

# basic-ml.tex

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

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

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

---

## ml-bagging.tex

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

## ml-bayesopt.tex

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

## ml-boosting.tex

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

## ml-gp.tex

Macro	Notation	Comment
<code>\gp</code>	$\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$	Gaussian Process Definition
<code>\mvec</code>	$\mathbf{m}$	Gaussian process mean vector
<code>\Kmat</code>	$\mathbf{K}$	estimated base learner
<code>\ls</code>	$\ell$	length-scale



## ml-mbo.tex

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

## ml-nn.tex

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

ml-rf.tex

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

# ml-svm.tex

Macro	Notation	Comment
<code>\sv</code>	$SV$	supportvectors
<code>\HS</code>	$\mathcal{H}$	H, hilbertspace
<code>\sl</code>	$\zeta$	
<code>\slvec</code>	$(\zeta^{(1)}, \zeta^{(n)})$	slack variables (SVM)
<code>\sli</code>	$\zeta^{(i)}$	slack variable (SVM)

# ml-trees.tex

Macro	Notation	Comment
<code>{\Np}</code>	$\mathcal{N}$	(Parent) node N
<code>{\Npk}</code>	$\mathcal{N}_k$	node N_k
<code>{\Nl}</code>	$\mathcal{N}_1$	Left node N_1
<code>{\Nr}</code>	$\mathcal{N}_2$	Right node N_2
<code>{\pikN}</code>	$\pi_k^{(\mathcal{N})}$	class probability node N
<code>{\pikNh}</code>	$\hat{\pi}_{\#1}^{(\mathcal{N})}$	estimated class probability node N
<code>{\pikNlh}</code>	$\hat{\pi}_{\#1}^{(\mathcal{N}_1)}$	
<code>{\pikNrhh}</code>	$\hat{\pi}_{\#1}^{(\mathcal{N}_2)}$	