10-301/601 Notation Guide

Matthew R. Gormley

May 17, 2020

1 Scalars, Vectors, Matrices

Scalars are either lowercase letters $x, y, z, \alpha, \beta, \gamma$ or uppercase Latin letters N, M, T. The latter are typically used to indicate a **count** (e.g. number of examples, features, timesteps) and are often accompanied by a corresponding **index** n, m, t (e.g. current example, feature, timestep). **Vectors** are bold lowercase letters $\mathbf{x} = [x_1, x_2, \ldots, x_M]^T$ and are typically assumed to be *column* vectors—hence the transposed row vector in this example. When handwritten, a vector is indicated by an over-arrow $\vec{x} = [x_1, x_2, \ldots, x_M]^T$. **Matrices** are bold uppercase letters:

$$\mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & \ldots & U_{1m} \\ U_{21} & U_{22} & & \\ \vdots & & \ddots & \vdots \\ U_{n1} & & \ldots & U_{nm} \end{bmatrix}$$

As in the examples above, subscripts are used as **indices** into structured objects such as vectors or matrices.

2 Sets

Sets are represented by caligraphic uppercase letters $\mathcal{X}, \mathcal{Y}, \mathcal{D}$. We often index a set by **labels** in parenthesized superscripts $\mathcal{S} = \{s^{(1)}, s^{(2)}, \ldots, s^{(S)}\}$, where $S = |\mathcal{S}|$. A shorthand for this equivalently defines $\mathcal{S} = \{s^{(s)}\}_{s=1}^{S}$. This shorthand is convenient when defining a set of **training examples**: $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$ is equivalent to $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$.

3 Random Variables

Random variables are also uppercase Latin letters X, Y, Z, but their use is typically apparent from context. When a random variable X_i and a scalar x_i are upper/lower-case versions of each other, we typically mean that the scalar is a **value** taken by the random variable.

When possible, we try to reserve Greek letters for **parameters** $\boldsymbol{\theta}, \boldsymbol{\phi}$ or **hyperparameters** α, β, γ.

For a random variable X, we write $X \sim \text{Gaussian}(\mu, \sigma^2)$ to indicate that X **follows** a 1D Gaussian distribution with mean μ and variance σ^2. We write $x \sim \text{Gaussian}(\mu, \sigma^2)$ to say that x is a value **sampled** from the same distribution.

A **conditional probability distribution** over random variable X given Y and Z is written $P(X|Y, Z)$ and its **probability mass function** (pmf) or **probability density function** (pdf) is $p(x|y, z)$. If the probability distribution has parameters α, β, we can write its pmf/pdf in at least three equivalent ways: A statistician might prefer $p(x|y, z; \alpha, \beta)$ to clearly demarcate the parameters. A graphical models expert prefer $p(x|y, z, \alpha, \beta)$ since said parameters are really just additional random variables. A typographer might prefer to save ink by writing $p_{\alpha,\beta}(x|y, z)$. To refer to this pmf/pdf as a function over possible values of a we would elide it as in $p_{\alpha,\beta}(\cdot|y,z)$. Using our $\sim$ notation from above, we could then write that X follows the distribution $X \sim p_{\alpha,\beta}(\cdot|y,z)$ and x is a sample from it $x \sim p_{\alpha,\beta}(\cdot|y,z)$.

The **expectation** of a random variable X is $\mathbb{E}[X]$. When dealing with random quantities for which the generating distribution might not be clear we can denote it in the expectation. For example, $\mathbb{E}_{x\sim p_{\alpha,\beta}(\cdot|y,z)}[f(x,y,z)]$ is the expectation of $f(x,y,z)$ for some function f where x is sampled from the distribution $p_{\alpha,\beta}(\cdot|y,z)$ and y and z are constant for the evaluation of this expectation.

4 Functions and Derivatives

Suppose we have a function $f(x)$. We write its partial derivative with respect to x as $\frac{\partial f(x)}{\partial x}$ or $\frac{df(x)}{dx}$.[1] We also denote its first derivative as $f'(x)$, its second derivative as $f''(x)$, and so on. For a multivariate function $f(\mathbf{x}) = f(x_1,\ldots,x_M)$, we write its gradient with respect to $\mathbf{x}$ as $\nabla_{\mathbf{x}} f(\mathbf{x})$ and frequently omit the subscript, i.e. $\nabla f(\mathbf{x})$, when it is clear from context—it might not be for a gradient such as $\nabla_{\mathbf{y}} g(\mathbf{x},\mathbf{y})$.

5 Common Conventions

The table below lists additional common conventions we follow:

Notation	Description
N	number of training examples
M	number of feature types
K	number of classes
n or i	current training example
m	current feature type
k	current class
$\mathbb{Z}$	set of integers
$\mathbb{R}$	set of reals
$\mathbb{R}^M$	set of real-valued vectors of length M
$\{0,1\}^M$	set of binary vectors of length M
$\mathbf{x}$	feature vector (input) where $\mathbf{x} = [x_1, x_2, \ldots, x_M]^T$; typically $\mathbf{x} \in \mathbb{R}^M$ or $\mathbf{x} \in \{0,1\}^M$
y	label / regressand (output); for classification $y \in \{1,2,\ldots,K\}$; for binary classification $y \in \{0,1\}$ or $y \in \{+1,-1\}$; for regression, $y \in \mathbb{R}$
$\mathcal{X}$	input space, i.e. $\mathbf{x} \in \mathcal{X}$
$\mathcal{Y}$	output space, i.e. $y \in Yc$
$\boldsymbol{x}^{(i)}$	the ith feature vector in the training data
$y^{(i)}$	the ith true output in the training data
$x_m^{(i)}$	the mth feature of the ith feature vector
$(\boldsymbol{x}^{(i)}, y^{(i)})$	the ith training example (feature vector, true output)

[1] Note that a more careful notation system would *always* use $\frac{\partial f(x)}{\partial x}$ for partial derivatives, since $\frac{\mathrm{d}f(x)}{\mathrm{d}x}$ is typically reserved for total derivatives. However, only partial derivatives make an appearance herein.

$\mathcal{D}$	set of training examples; for supervised learning $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^N$; for unsupervised learning $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^N$
$\mathbf{X}$	design matrix; the ith row contains the features of the ith training example $\mathbf{x}^{(i)}$; i.e the ith row contains $x_1^{(i)}, \ldots, x_M^{(i)}$
$X_1, \ldots, X_M$	random variables corresponding to feature vector $\mathbf{x}$; (note: we generally avoid defining a vector-valued random variable $\mathbf{X} = [X_1, X_2, \ldots, X_M]^T$ so that $\mathbf{X}$ is not overloaded with the design matrix)
Y	random variable corresponding to predicted class y
$P(Y = y \vert \mathbf{X} = \mathbf{x})$	probability of random variable Y taking value y given that random variable $\mathbf{X}$ takes value $\mathbf{x}$
$p(y \vert \mathbf{x})$	shorthand for $P(Y = y \vert \mathbf{X} = \mathbf{x})$
$\boldsymbol{\theta}$	model parameters
$\mathbf{w}$	model parameters (weights of linear model)
b	model parameter (bias term of linear model)
$\ell(\boldsymbol{\theta})$	log-likelihood of the data; depending on context, this might alternatively be the log- conditional likelihood *or* log-marginal likelihood
$J(\boldsymbol{\theta})$	objective function
$J^{(i)}(\boldsymbol{\theta})$	example i's contribution to the objective function; typically $J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N J^{(i)}(\boldsymbol{\theta})$
$\nabla J(\boldsymbol{\theta})$	gradient of the objective function with respect to model parameters $\boldsymbol{\theta}$
$\nabla J^{(i)}(\boldsymbol{\theta})$	gradient of $J^{(i)}(\boldsymbol{\theta})$ with respect to model parameters $\boldsymbol{\theta}$
λ	stepsize in numerical optimization
$\theta^T\mathbf{x}$ or $\mathbf{x}^T\boldsymbol{\theta}$ or $\boldsymbol{\theta} \cdot \mathbf{x}$	dot product of model parameters and features
$h_{\boldsymbol{\theta}}(\mathbf{x})$	decision function / decision rule / hypothesis
$\mathcal{H}$	hypothesis space; we say that $h \in \mathcal{H}$
$\hat{y}$	prediction of a decision function, e.g. $\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x})$
$\hat{\boldsymbol{\theta}}$	model parameters that result from learning
$\ell(\hat{y}, y)$	loss function
$p^*(\mathbf{x}, y)$	unknown data generating distribution of labeled examples
$p^*(\mathbf{x})$	unknown data generating distribution of feature vectors only
$c^*(\mathbf{x})$	true unknown hypothesis (i.e. oracle labeling function), e.g. $y = c^*(\mathbf{x})$
$\mathbf{z}$	Values of unknown variables (latent)
$Z_1, \ldots, Z_C$	random variables (latent) corresponding to $\mathbf{z}$
$\mathbf{y}$	predicted structure (output) for structured prediction
$Y_1, \ldots, Y_C$	random variables corresponding to predicted structure $\mathbf{y}$
$\mathbb{I}(a = b)$	indicator function which returns 1 when a equals b and 0 otherwise—other notations are also possible $\mathbb{I}(a = b) = \mathbf{1}(a = b) = \mathbf{1}_{a=b}$