Notation

Textbooks, papers, code, and other courses will all use different names and notations for the things covered in this course. While learning a subject, these differences can be confusing. However, dealing with different notations is a necessary research skill. There will always be differences in presentation in different sources, due to different trade-offs and priorities.

We try to make the notation fairly consistent within the course. This note lists some of the conventions that we've chosen to follow that might be unfamiliar.

You can probably skip over this note at the start of the class. Most notation is introduced in the notes as we use it. However, everything mentioned here is something that has been queried by previous students of this class. So please refer back to this note if you find any unfamiliar notation later on.

1 Vectors, matrices, and indexing

Indexing: Where possible, these notes use lower-case letters for indexing, with the corresponding capital letters for the numbers of settings. Thus the D 'dimensions' or elements of a vector are indexed with $d = 1 \ldots D$, and N training data-points are indexed with $n = 1 \ldots N$.

As a result, the notation won't match some textbooks. In statistics it's common to index data-items with $i = 1 \ldots n$, and parameters with $j = 1 \ldots p$.

Vectors in this course are all column-vectors — these are just columns of numbers, you don't need to know about more abstract vector-spaces in this course. We use bold-face lower-case letters to denote these column vectors in the typeset notes. For example, we write a D-dimensional vector as:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_D \end{bmatrix} = [x_1 \;\; x_2 \;\; \cdots \;\; x_D]^\top. \tag{1}$$

If we need to show the contents of a column vector, we often create it from a row-vector with the transpose $^\top$ to save vertical space in the notes. We use subscripts to index into a vector (or matrix).

Depending on your background, you might be more familiar with $\vec{x}$ than $\mathbf{x}$.

In handwriting, we underline vectors: $\underline{x}$ because it's difficult to handwrite in bold! While not everyone underlines vectors, and some authors use arrows, we recommend you write with the same notation in this class to help communication. But as long is it's clear from context what you are doing, it's not critical. We often forget to underline vectors when writing, so we understand.

Matrices (for us rectangular arrays of numbers) are upper-case letters, like A and Σ. We've chosen not to bold them, even though there are sometimes numbers represented by upper-case letters floating around (such as D for number of dimensions). It should usually be clear from context which quantities are matrices, and what size they are. See the maths cribsheet for details on indexing matrices, and how sizes match in matrix-vector multiplication.

Addition: sizes of vectors or matrices should match when adding quantities: $\mathbf{a} + \mathbf{b}$, or $A + B$. As an exception, to add a scalar c to every element, we'll just write $\mathbf{a} + c$ or $A + c$.

Indexing items: Sometimes we use superscripts to identify items, such as $\mathbf{x}^{(n)}$ for the nth D-dimensional input vector in a training set with N examples. We can (and often do) stack these vectors into an $N \times D$ matrix X, so we could use a notation such as $X_{n,:}$ to access the nth row. In this case we chose to introduce the extra superscript notation instead. In the past, many students have found it hard to follow the distinction between datapoints and feature dimensions when studying "kernel methods" such as Gaussian processes. We hope a notation where datapoint identity and feature dimensions look different will help avoid confusion later.

2 Probabilities

The **probability mass** of a discrete outcome x is written $P(x)$.

When it doesn't seem necessary (nearly always) we don't introduce notation for a corresponding random variable X, and write more explicit expressions like $P_X(x)$ or $P(X\!=\!x)$. Notation is a trade-off, and more explicit notation can be more of a burden to work with.

Joint probabilities: $P(x, y)$. **Conditional probabilities:** $P(x \,|\, y)$. Conditional probabilities are sometimes written in the literature as $P(x;\, y)$ — especially in frequentist statistics rather than Bayesian statistics. The '|' symbol, introduced by Jeffreys, is historically associated with Bayesian reasoning. Hence for arbitrary functions, like $f(\mathbf{x};\, \mathbf{w})$, where we want to emphasize that it's primarily a function of $\mathbf{x}$ controlled by parameters $\mathbf{w}$, we've chosen not to use a '|'.

The **probability density** of a real-valued outcome x is written with a lower-case $p(x)$, such that $P(a < X < b) = \int_a^b p(x)\, \mathrm{d}x$. We tend not to introduce new symbols for density functions over different variables, but again overload the notation: we call them all "p" and infer which density we are talking about from the argument.

Gaussian distributions are reviewed later in the notes. We will write that an outcome x was sampled from a Gaussian or normal distribution using $x \sim \mathcal{N}(\mu, \Sigma)$. We write the probability density associated with that outcome as $\mathcal{N}(x;\, \mu, \Sigma)$. We could also have chosen to write $\mathcal{N}(x \,|\, \mu, \Sigma)$, as Bishop and Murphy do. The ';' was force of habit, because the Gaussian (outside of this course) is used in many contexts, and not just Bayesian reasoning.

3 Integrals and expectations

All of the integrals in this course are definite integrals corresponding to expectations. For a real-valued quantity x, we write:

$$\mathbb{E}_{p(x)}[f(x)] = \mathbb{E}[f(x)] = \int f(x)\, p(x)\, \mathrm{d}x. \tag{2}$$

This is a definite integral over the whole range of the variable x. We might have written $\int_{-\infty}^{\infty} \ldots$ or $\int_X \ldots$, but because our integrals are always over the whole range of the variable, we don't bother to specify the limits.

The expectation notation is often quicker to work with than writing out the integral. As above, we sometimes don't specify the distribution (especially when handwriting), if it can be inferred from context.

Please do review the background note on expectations and sums of random variables. Throughout the course we will see generalizations of those results to real-valued variables (as above) and expressions with matrices and vectors. You need to have worked through the basics.

You may have seen multiple dimensional integrals written with multiple integral signs, for example for a 3-dimensional vector:

$$\iiint f(\mathbf{x})\, \mathrm{d}x_1\, \mathrm{d}x_2\, \mathrm{d}x_3. \tag{3}$$

Our integrals are often over high-dimensional vectors, so rather than writing potentially hundreds of integral signs, we simply write:

$$\int f(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{4}$$

And if we are integrating over multiple vectors we still might only write one integral sign:

$$\int f(\mathbf{x}, \mathbf{z})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{z}. \tag{5}$$

4 Derivatives

Partial derivative of a scalar with respect to another scalar: $\frac{\partial f}{\partial x}$. Example: $\frac{\partial \sin(yx)}{\partial x} = y\cos(yx)$.

Column vector of partial derivatives: $\nabla_{\mathbf{w}} f = \left[\frac{\partial f}{\partial x_1}\;\; \frac{\partial f}{\partial x_2}\;\; \cdots\;\; \frac{\partial f}{\partial x_D}\right]^\top$.

These notes avoid writing derivatives involving vectors as $\frac{\partial \mathbf{y}}{\partial \mathbf{z}}$. Usually this expression would be a matrix with $\left(\frac{\partial \mathbf{y}}{\partial \mathbf{z}}\right)_{ij} = \frac{\partial y_i}{\partial z_j}$. Under this common convention, the derivative of a scalar $\frac{\partial f}{\partial \mathbf{w}}$ is a row vector, $(\nabla_{\mathbf{w}} f)^\top$.

We also don't write $\frac{\partial A}{\partial B}$ for matrices A and B. While we could put all of the partial derivatives $\left\{\frac{\partial A_{ij}}{\partial B_{kl}}\right\}$ into a 4-dimensional array, that's often not a good idea in machine learning.

Later in the course we will review a notation more suitable for computing derivatives, where derivative quantities are stored in arrays of the same size and shape as the original variables. All will be explained in the note on *Backpropagation of Derivatives*.

5 Frequently used symbols

We try to reserve some letters to mean the same thing throughout the course, so you can recognize what the terms in equations are at a glance.

- D number of dimensions in input vector $\mathbf{x}$, number of features.
- N number of (training) data-points.
- M number of test or validation points, but used as some matrix in parts of the notes…
- K number of components in a model, or number of transformed features in $\boldsymbol{\phi}(\mathbf{x})$.
- f a scalar function, usually the output of a machine learning function. The vector $\mathbf{f}$ is often a vector containing the N function values for N training inputs. In some contexts later on, it could be a vector of outputs from a vector-valued function applied to a single input.
- $f(\mathbf{x}; \mathbf{w})$ depends on both the input feature vector $\mathbf{x}$, and the parameters $\mathbf{w}$. The semicolon (rather than a comma) is somewhat arbitrary, but separates the input that we will provide when we use the function later, from the parameters that we fit at training time.
- $\mathbf{x}$ a vector input to a machine learning system, also called the *features* or a *feature vector*.
- X an $N\times D$ matrix of (training) inputs. Occasionally code (especially when using the Fortran memory layout) expects your data to be stored in a $D\times N$ array. It is worth double-checking that your matrices are the correct way around when calling library routines, and leave comments in your own code to document the intended sizes of arrays. Sometimes I leave a comment at the end of a line that simply gives the size or shape of the result: for example, "`# (N,D)`".
- y is a target output that we predict with $f(\mathbf{x})$. A vector $\mathbf{y}$ could be a vector of N targets for all the training examples, *or* a vector target output for single input, depending on context.
- $\mathbf{w}$ (or W) a vector (or matrix, or collection) of parameters or 'weights'. In statistics textbooks and papers, linear regression weights are often called β.

- b a constant offset or intercept in linear regression, in neural networks language a "bias weight" or simply "bias". There's an unfortunate clash in terminology: a "bias weight" does not set the statistical "bias" of the model (its expected test error, averaged over different training sets). In other sources this constant could have various other symbols, including β_0, w_0, or c.
- θ like $\mathbf{w}$ often used for parameters. Sometimes used to mean a collection of all the parameters in the model. Used for the 'hyperparameters' in Gaussian processes.
- $\boldsymbol{\phi}(\mathbf{x})$ vector-valued function (usually non-linear 'basis functions') used to replace features with a new representation early in the course. In a neural network (later in the course) the equivalent is a *hidden layer*, which could be called $\mathbf{h}$.
- Φ (a capital ϕ), like X a matrix of inputs, but now containing $N \times K$ transformed inputs made with the $\boldsymbol{\phi}$ function.
- σ either a standard deviation, *or* the logistic sigmoid function $\sigma(a) = 1/(1 + e^{-a})$.