ECE5268 – Theory of Neural Networks
Lecture Notes

Instructor: Dr. Georgios C. Anagnostopoulos

Spring 2020

Abstract

This is a collection of lecture notes pertaining to ECE5268 *Theory of Neural Networks*, which was taught by Georgios C. Anagnostopoulos (GCA) in Spring 2020. They were prepared/compiled by the following participant students:

- Luis A. Pantin (LAP), graduate student of Computer Engineering.
- Jose A. Della Sala Pereira (JADSP), graduate student of Computer Engineering.
- Micah L. Billouin (MLB), graduate student of Computer Engineering.
- Allen M. Shultz (AMS), graduate student of Computer Engineering.
- *(see source code to see how to add your name & initials here.)*

Please note that all these notes have not been subjected to sufficient scrutiny, so they may, potentially, contain mistakes and typos. Nevertheless, they are hopefully useful in summarizing what material was covered in order to be used for review/study purposes.

Contents

ECE5268: Theory of Neural Networks **Spring 2020**

Lecture 1: Linear Algebra Preliminaries

Thu, Jan 16, 2020

Lecturer: GCA *Scribe(s): GCA*

Note: *This header style is courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

This lecture provides a quick review and overview of useful linear algebra-related concepts that will come in handy in the future. Notice that our discussion will limit itself to finite-dimensional (ordinary) vectors and operators (matrices).

1.1 Vectors

We will consider all vectors as column vectors, *i.e.*, the D-dimensional vector $\mathbf{v} \in \mathbb{R}^D$ will have its elements v_d arranged in a single column. Occasionally, it helps to regard such a vector as a matrix of D rows and 1 column, *i.e.*$\mathbf{v} \in \mathbb{R}^{D\times 1}$. The vector $\mathbf{v}^T \in \mathbb{R}^{1\times D}$ will then be a row vector. Also occasionally, we will use "vector" and "point" (in the D-dimensional space interchangeably). Special vectors we are going to use are the zero vector $\mathbf{0}$ and the all-ones vector $\mathbf{1}$.

The *dot product* of two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ is the scalar

$$\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u} = \sum_{i=1}^{D} u_i v_i \tag{1.1}$$

Occasionally, it is also referred to as an *inner product*. This is not to be confused with the *outer product* $\mathbf{u}\mathbf{v}^T \in \mathbb{R}^{D\times D}$, which is a matrix. Also, notice that the outer product is non-commutative, *i.e.*, $\mathbf{u}\mathbf{v}^T \neq \mathbf{v}\mathbf{u}^T$. If $D = 3$, we have

$$\mathbf{u}\mathbf{v}^T = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} = \begin{bmatrix} u_1v_1 & u_1v_2 & u_1v_3 \\ u_2v_1 & u_2v_2 & u_2v_3 \\ u_3v_1 & u_3v_2 & u_3v_3 \end{bmatrix} \tag{1.2}$$

The product is referred to as "outer," since it yields a more complicated object, a matrix. This is in contrast to the dot (inner) product, which yields something simpler, a scalar.

Based on the dot product, we can define a pair of vectors $\mathbf{u}$ and $\mathbf{v}$ to be *orthogonal*, if and only if $\mathbf{u}^T\mathbf{v} = 0$.

A vector norm is a function $\|\cdot\|$ that measures the "length" $\|\mathbf{u}\|$ of a vector $\mathbf{u}$. We are already familiar with such a function, namely the Euclidean or L_2 norm $\|\cdot\|_2$. For such a function to be eligible to be called a norm, it has to satisfy certain (common-sense) axioms; please refer to [10] for details. It is worth mentioning the L_p norm for $p \geq 1$, which is defined as

$$\|\mathbf{u}\|_p \triangleq \left[\sum_{i=1}^{D} |u_i|^p\right]^{\frac{1}{p}} \qquad p \geq 1 \tag{1.3}$$

For $p = 2$, we get the Euclidean norm, for which we have

$$\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^T\mathbf{u}} \tag{1.4}$$

For $p \to +\infty$, we get the L_∞ or *max* norm, which is given as

$$\|\mathbf{u}\|_\infty \triangleq \max_i |u_i| \tag{1.5}$$

If $\mathbf{u} \in \mathbb{R}^D$, an important relationship between norms is the following

$$\|\mathbf{u}\|_p \leq \|\mathbf{u}\|_r \leq D^{(1/r-1/p)} \|\mathbf{u}\|_p \qquad 1 \leq p \leq r \tag{1.6}$$

which allows us to bound one L_p norm in terms of another. A couple of other important inequalities involving norms are

$$|\mathbf{u}^T\mathbf{v}| \leq \|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \qquad \text{Cauchy's Inequality} \tag{1.7}$$

and the more general inequality

$$|\mathbf{u}^T\mathbf{v}| \leq \|\mathbf{u}\|_p \|\mathbf{v}\|_q \qquad \text{Hölder's Inequality} \tag{1.8}$$

where

$$q \geq 1 : \frac{1}{p} + \frac{1}{q} = 1 \tag{1.9}$$

Notice that Cauchy's Inequality is a special case of Hölder's Inequality (obtained for $p = 2$). Norms with p and q satisfying Eq. (1.9) are called *(Hölder) conjugate*. For example, L_∞ and L_1 are such conjugates and the L_2 norm is self-conjugate.

Based on the notion of a vector norm one can define the notion of a *metric* (distance) $d(\mathbf{u}, \mathbf{v}) \triangleq \|\mathbf{u} - \mathbf{v}\|$ between two points/vectors[1]. More details on metrics can be found online at [9].

[1]The opposite is also true; given a metric d, one can define/recover a norm as $\|\mathbf{u}\| \triangleq d(\mathbf{u}, \mathbf{0})$.

1.2 Matrices

$\mathbf{A} \in \mathbb{R}^{M \times N}$ will denote a matrix of M rows and N columns and $\mathbf{A}^T \in \mathbb{R}^{N \times M}$ will stand for its transpose. Obviously, $\left(\mathbf{A}^T\right)^T = \mathbf{A}$. If $\mathbf{A}^T = \mathbf{A}$, we call $\mathbf{A}$ *symmetric*. If $M = N$, then $\mathbf{A}$ is referred to as a *square* matrix, otherwise *rectangular*; to be precise, if $M > N$, we will call it a "tall" matrix and, if $M < N$, a "wide" matrix. Special matrices we are going to encounter are the (necessarily square) identity matrix $\mathbf{I}$ and the zero matrix $\mathbf{O}$.

Regarding transposition, for conformable matrices (having the right dimensions, so that they can be multiplied) it holds that

$$(\mathbf{ABC})^T = \mathbf{C}^T\mathbf{B}^T\mathbf{A}^T \tag{1.10}$$

Apart from the traditional matrix product, there is also the element-wise, Hadamard product $\odot$ of matrices of the same dimensions. The $(m,n)^{\text{th}}$ element of $\mathbf{A} \odot \mathbf{B}$ is simply given as $a_{m,n}b_{m,n}$.

Now, here are some reminders regarding square matrices. If $\mathbf{A} \in \mathbb{R}^{N \times N}$, then the matrix $\mathbf{A}^{-1} \in \mathbb{R}^{N \times N}$ that is such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$ is called $\mathbf{A}$'s inverse. This inverse exists, if and only if $\det(\mathbf{A}) \neq 0$; in that case, $\mathbf{A}$ is called *invertible*. It can be shown that $\left(\mathbf{A}^{-1}\right)^{-1} = \mathbf{A}$. Regarding determinants, it holds that

$$\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B}) \tag{1.11}$$

Hence, the product $\mathbf{AB}$ is invertible, if and only if each one of these matrices is invertible. As a matter of fact, one can easily show that

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \tag{1.12}$$

A square matrix $\mathbf{U}$ is called *orthogonal*, if and only if $\mathbf{U}^{-1} = \mathbf{U}^T \quad \Leftrightarrow \quad \mathbf{U}\mathbf{U}^{-1} = \mathbf{U}^{-1}\mathbf{U} = \mathbf{I}$. It has columns and rows that are mutually orthogonal and are of unit length. Such a matrix has the following interesting, easily-verifiable property:

$$\|\mathbf{Uv}\|_2 = \|\mathbf{v}\|_2 \tag{1.13}$$

In other words, when $\mathbf{U}$ is applied onto $\mathbf{v}$, it does not change $\mathbf{v}$'s Euclidean length. It turns out that an orthogonal matrix will rotate and/or reflect a vector. A transformation that preserves lengths is called an *isometry*. Furthermore, it is easy to see that $\det(\mathbf{U}) = \pm 1$.

The *trace* of a square matrix is defined as the sum of its diagonal elements, *i.e.*,

$$\text{trace}\{\mathbf{A}\} \triangleq \sum_{n=0}^{N} a_{n,n} \tag{1.14}$$

Apart from the obvious fact that $\text{trace}\{\mathbf{A}+\mathbf{B}\} = \text{trace}\{\mathbf{A}\} + \text{trace}\{\mathbf{B}\}$, the trace has some other interesting properties:

$$\text{trace}\Big\{\mathbf{A}^T\Big\} = \text{trace}\{\mathbf{A}\} \tag{1.15}$$

$$\text{trace}\{\mathbf{ABC}\} = \text{trace}\{\mathbf{BCA}\} = \text{trace}\{\mathbf{CAB}\} \qquad \text{Circular Shift Invariance} \tag{1.16}$$

$$\mathbf{u}^T\mathbf{v} = \text{trace}\Big\{\mathbf{u}^T\mathbf{v}\Big\} = \text{trace}\Big\{\mathbf{u}\mathbf{v}^T\Big\} = \text{trace}\Big\{\mathbf{v}\mathbf{u}^T\Big\} \qquad \text{Trace Trick} \tag{1.17}$$

Note that, in Eq. (1.16), the three matrices involved do not have to be square, as long as their products shown result in a square matrix, so that its trace can be taken.

When manipulating traces, primarily based on the first two properties, we will use $\overset{\text{i.t.}}{=}$ ("equal when inside traces") as convenient type of equality to declare that its Left Hand Side (LHS) equals its Right Hand Side (RHS), when both quantities are inside traces. For example, when we'll write $\mathbf{ABC} \overset{\text{i.t.}}{=} \mathbf{BCA}$ we will mean that $\text{trace}\{\mathbf{ABC}\} = \text{trace}\{\mathbf{BCA}\}$.

At this point, let us mention the $\text{diag}(\cdot)$ operator. For a vector $\mathbf{v} \in \mathbb{R}^D$, $\text{diag}(\mathbf{v})$ will stand for a diagonal $D \times D$ matrix, whose diagonal elements are given by $\mathbf{v}$. Furthermore, for a matrix $\mathbf{A} \in \mathbb{R}^{D\times D}$, $\text{diag}(\mathbf{A})$ will stand for a D-dimensional vector consisting of $\mathbf{A}$'s diagonal elements[2]. For example, it holds that $\text{diag}(\text{diag}(\mathbf{v})) = \mathbf{v}$.

Finally, let us touch up on weighted dot products. If $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$ and $\mathbf{A} \in \mathbb{R}^{D\times D}$, then we call the scalar quantity $\mathbf{u}^T\mathbf{A}\mathbf{v}$ the *weighted dot product* of $\mathbf{u}$ and $\mathbf{v}$ with weight matrix $\mathbf{A}$. By taking the transpose of this scalar quantity, we can easily show that

$$\mathbf{u}^T\mathbf{A}\mathbf{v} \overset{.^T}{=} \mathbf{v}^T\mathbf{A}^T\mathbf{u} = \mathbf{u}^T\left(\frac{\mathbf{A}+\mathbf{A}^T}{2}\right)\mathbf{v} \tag{1.18}$$

which renders the weight matrix symmetric. The quantity $(\mathbf{A}+\mathbf{A}^T)/2$ is called the *symmetric part* of $\mathbf{A}$ and, for a symmetric matrix, equals the matrix itself.

1.3 Partitioned Matrices

A *partitioned* or *block matrix* is a matrix that has been partitioned into sub-matrices, each of which is called a *block* of the matrix. The blocks might be square or rectangular matrices, row or column

[2]This operator precisely coincides with the `diag` function of MATLAB®.

vectors or, even, scalars, *i.e.*, of any dimensions, so that, when arranged together, yield the original matrix.

A few important identities follow right next; it is assumed that all matrix multiplications depicted are between conformable matrices.

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}^T = \begin{bmatrix}\mathbf{A}^T & \mathbf{B}^T\end{bmatrix} \tag{1.19}$$

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\mathbf{C} = \begin{bmatrix}\mathbf{AC}\\ \mathbf{BC}\end{bmatrix} \tag{1.20}$$

$$\mathbf{A}\begin{bmatrix}\mathbf{B} & \mathbf{C}\end{bmatrix} = \begin{bmatrix}\mathbf{AB} & \mathbf{AC}\end{bmatrix} \tag{1.21}$$

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{C}\\ \mathbf{D}\end{bmatrix} = \mathbf{AC}+\mathbf{BD} \tag{1.22}$$

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{C} & \mathbf{D}\end{bmatrix} = \begin{bmatrix}\mathbf{AC} & \mathbf{AD}\\ \mathbf{BC} & \mathbf{BD}\end{bmatrix} \tag{1.23}$$

$$\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\begin{bmatrix}\mathbf{E} & \mathbf{F}\\ \mathbf{G} & \mathbf{H}\end{bmatrix} = \begin{bmatrix}\mathbf{AE}+\mathbf{BG} & \mathbf{AF}+\mathbf{BH}\\ \mathbf{CE}+\mathbf{DG} & \mathbf{CF}+\mathbf{DH}\end{bmatrix} \tag{1.24}$$

As one can witness from the previous identities, these properties generalize the corresponding ones that pertain to matrices with scalar blocks. They can be very useful, as the following example shows. However, when applying them, ensure that matrices are comformable in the resulting products.

Example 1. Let's assume that, based on a set of vectors $\{\mathbf{x}_n\}_{n=1}^N$, which are organized into rows of a matrix $\mathbf{X}$, we construct a new vectors $\mathbf{y}_n = \mathbf{W}\mathbf{x}_n$. If $\mathbf{Y}$ is the matrix that contains the $\mathbf{y}_n$'s as its rows, let us assume we would like to find a relationship between the matrices $\mathbf{X}, \mathbf{W}$ and $\mathbf{Y}$.

Proof. We proceed as follows:

$$\mathbf{Y} = \begin{bmatrix}\mathbf{y}_1^T\\ \vdots\\ \mathbf{y}_N^T\end{bmatrix} = \begin{bmatrix}\mathbf{x}_1^T\mathbf{W}^T\\ \vdots\\ \mathbf{x}_N^T\mathbf{W}^T\end{bmatrix} \overset{(1.20)}{=} \underbrace{\begin{bmatrix}\mathbf{x}_1^T\\ \vdots\\ \mathbf{x}_N^T\end{bmatrix}}_{=\mathbf{X}}\mathbf{W}^T = \mathbf{X}\mathbf{W}^T \quad \Leftrightarrow \quad \mathbf{Y} = \mathbf{X}\mathbf{W}^T$$

Such a matrix-only expression comes in very handy, when working in MATLAB® as it replaces a `for`-loop ($n = 1, \ldots, N$) with a matrix multiplication.

□

If $\mathbf{A}^T = [\mathbf{a}_1 \dots \mathbf{a}_N]$ and $\mathbf{B}^T = [\mathbf{b}_1 \dots \mathbf{b}_N]$, then

$$(1.22) \quad \Rightarrow \quad \mathbf{A}^T\mathbf{B} = [\mathbf{a}_1 \dots \mathbf{a}_N] \begin{bmatrix} \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_N^T \end{bmatrix} = \sum_{n=1}^{N} \mathbf{a}_n \mathbf{b}_n^T \tag{1.25}$$

and

$$(1.23) \quad \Rightarrow \quad \mathbf{A}\mathbf{B}^T = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix} [\mathbf{b}_1 \dots \mathbf{b}_N] = \begin{bmatrix} \mathbf{a}_1^T\mathbf{b}_1 & \cdots & \mathbf{a}_1^T\mathbf{b}_N \\ \vdots & \ddots & \vdots \\ \mathbf{a}_N^T\mathbf{b}_1 & \cdots & \mathbf{a}_N^T\mathbf{b}_N \end{bmatrix} \tag{1.26}$$

As a matter of fact, Eq. (1.25) can be generalized even further to

$$\mathbf{A}^T \operatorname{diag}(\mathbf{c})\, \mathbf{B} = \sum_{n=1}^{N} c_n \mathbf{a}_n \mathbf{b}_n^T \tag{1.27}$$

Eq. (1.27) is useful in converting a weighted sum of outer products into a product of matrices. Another such useful identity is

$$(1.25) \quad \Rightarrow \quad \sum_{n=1}^{N} c_n \mathbf{a}_n = [\mathbf{a}_1 \dots \mathbf{a}_N]\, \mathbf{c} \tag{1.28}$$

1.4 Matrix Vectorization

Here, we'll introduce the vectorization operator vec, which stacks the N columns $\mathbf{a}_n$ of a matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$, from first to last, in order to produce a single, long vector as

$$\operatorname{vec} \mathbf{A} = \operatorname{vec} [\mathbf{a}_1 \dots \mathbf{a}_N] = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_N \end{bmatrix} \in \mathbb{R}^{NM} \tag{1.29}$$

It turns out that, via the use of vec one can express any linear mapping from the set of $\mathbb{R}^{M\times N}$ matrices to itself via the mapping $\mathbf{B} \operatorname{vec}(\cdot)$, where $\mathbf{B} \in \mathbb{R}^{MN\times MN}$ is an arbitrary matrix. This is what determines vec's importance.

Now, notice that $\operatorname{vec} \mathbf{A} \neq \operatorname{vec} \mathbf{A}^T$, unless this matrix is symmetric. In particular, if $\mathbf{A} \in \mathbb{R}^{M\times N}$ their (linear) relationship is given as

$$\operatorname{vec} \mathbf{A} = \mathbf{K}_{N,M} \operatorname{vec} \mathbf{A}^T \tag{1.30}$$

and

$$\operatorname{vec} \mathbf{A}^T = \mathbf{K}_{M,N} \operatorname{vec} \mathbf{A} \tag{1.31}$$

where $\mathbf{K}_{M,N} \in \mathbb{R}^{MN \times MN}$ is referred to as the (M, N) *commutation matrix*. A quick example is shown below

$$\mathbf{A} \triangleq \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \Rightarrow \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{=\mathbf{K}_{2,2}} \underbrace{\begin{bmatrix} a \\ c \\ b \\ d \end{bmatrix}}_{=\operatorname{vec} \mathbf{A}} = \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}}_{=\operatorname{vec} \mathbf{A}^T} \tag{1.32}$$

Interchanging the roles of $\mathbf{A}$ and $\mathbf{A}^T$ in Eq. (1.30) and Eq. (1.31) implies that

$$\mathbf{K}_{N,M} = \mathbf{K}_{M,N}^T \tag{1.33}$$

From Eq. (1.31) we see that the role of $\mathbf{K}_{M,N}$ is to permute (rearrange) the elements of $\operatorname{vec} \mathbf{A}$ into the order of elements as they appear in $\operatorname{vec} \mathbf{A}^T$. Commutation matrices are *permutation matrices*, *i.e.*, their entries consist of 0's and 1's and are orthogonal, *i.e.*,

$$\mathbf{K}_{M,N}^{-1} = \mathbf{K}_{M,N}^T \tag{1.34}$$

With respect to traces, this operator has the following interesting property:

$$\operatorname{trace}\Big\{\mathbf{A}^T \mathbf{B}\Big\} = (\operatorname{vec} \mathbf{A})^T \operatorname{vec} \mathbf{B} \tag{1.35}$$

Based on vec and the previous identity, one can define the *Frobenius* dot product for matrices

$$\langle \mathbf{A}, \mathbf{B} \rangle_F \triangleq (\operatorname{vec} \mathbf{A})^T \operatorname{vec} \mathbf{B} \overset{(1.35)}{=} \operatorname{trace}\Big\{\mathbf{A}^T \mathbf{B}\Big\} \tag{1.36}$$

Note that, in order to simplify notation, we may sometimes write $\operatorname{vec}^T \cdot$ instead of $(\operatorname{vec} \cdot)^T$. For example, $\operatorname{vec}^T \mathbf{A}$ will stand for $(\operatorname{vec} \mathbf{A})^T$.

Finally, commutation matrices are useful constructs in manipulating the interplay between vectorizations and Kronecker products, which are discussed further below. Perhaps, it may be surprising that we rarely have to explicitly form them, especially when writing code. For example, when trying to obtain $\text{vec}\,\mathbf{A}^T$ from $\text{vec}\,\mathbf{A}$ via Eq. (1.31), it is faster and more straightforward to perform this conversion via code that doesn't rely on first forming $\mathbf{K}_{M,N}$ and then multiplying it with $\text{vec}\,\mathbf{A}$ as Eq. (1.31) dictates. Nevertheless, for the curious among us, the $\mathbf{K}_{M,N}$ commutation matrix is explicitly given as

$$\mathbf{K}_{M,N} = \sum_{m=1}^{M}\sum_{n=1}^{N}\left(\mathbf{e}_{M,m}\mathbf{e}_{N,n}^T\right)\otimes\left(\mathbf{e}_{N,n}\mathbf{e}_{M,m}^T\right)$$

where $\mathbf{e}_{M,m}\in\mathbb{R}^M$ is the m^{th} column or row of the identity matrix $\mathbf{I}_M\in\mathbb{R}^{M\times M}$, etc.

1.5 Kronecker Product

Yet another matrix product that we will encounter is the *Kronecker* product $\otimes$ and it involves block matrices. Let $\mathbf{A}\in\mathbb{R}^{M\times N}$ and $\mathbf{B}\in\mathbb{R}^{P\times Q}$. Then, we define $\mathbf{A}\otimes\mathbf{B}$ as the block-partitioned matrix

$$\mathbf{A}\otimes\mathbf{B}\triangleq\begin{bmatrix}a_{1,1}\mathbf{B} & \cdots & a_{1,N}\mathbf{B}\\ \vdots & \ddots & \vdots\\ a_{M,1}\mathbf{B} & \cdots & a_{M,N}\mathbf{B}\end{bmatrix}\in\mathbb{R}^{MP\times NQ} \tag{1.37}$$

A trivial, but useful, Kronecker product between two vectors is

$$\mathbf{a}\otimes\mathbf{b}^T=\mathbf{a}\mathbf{b}^T \tag{1.38}$$

It is easy to verify that the Kronecker product is not commutative; in general, $\mathbf{A}\otimes\mathbf{B}\neq\mathbf{B}\otimes\mathbf{A}$. However, since both these Kronecker products contain the same element-wise products of $\mathbf{A}$ and $\mathbf{B}$, a rearrangement of their entries can convert one Kronecker product to the other as shown below:

$$\mathbf{A}\otimes\mathbf{B}=\mathbf{K}_{M,P}(\mathbf{B}\otimes\mathbf{A})\mathbf{K}_{Q,N} \tag{1.39}$$

and

$$\mathbf{B}\otimes\mathbf{A}=\mathbf{K}_{P,M}(\mathbf{B}\otimes\mathbf{A})\mathbf{K}_{N,Q} \tag{1.40}$$

The Kronecker product has several important and easily verifiable properties:

$$(a\mathbf{A}) \otimes \mathbf{B} = \mathbf{A} \otimes (a\mathbf{B}) = a(\mathbf{A} \otimes \mathbf{B}) \qquad \text{Homogeneity} \qquad (1.41)$$

$$\mathbf{A} \otimes (\mathbf{B} + \mathbf{C}) = \mathbf{A} \otimes \mathbf{B} + \mathbf{A} \otimes \mathbf{C} \qquad \text{Distributivity} \qquad (1.42)$$

$$(\mathbf{A} + \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes \mathbf{C} + \mathbf{B} \otimes \mathbf{C} \qquad (1.43)$$

$$(\mathbf{A} \otimes \mathbf{B}) \otimes \mathbf{C} = \mathbf{A} \otimes (\mathbf{B} \otimes \mathbf{C}) \qquad \text{Associativity} \qquad (1.44)$$

$$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T \qquad \text{Transposition} \qquad (1.45)$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}) \qquad \text{Ordinary/Kronecker Product Association} \qquad (1.46)$$

Note that Eq. (1.46) holds only if the products $\mathbf{AC}$ and $\mathbf{BD}$ can be formed, *i.e.*, the matrices involved are comformable.

Furthermore, for square matrices $\mathbf{A} \in \mathbb{R}^{M \times M}$ and $\mathbf{B} \in \mathbb{R}^{N \times N}$ we have that

$$\operatorname{trace}\{\mathbf{A} \otimes \mathbf{B}\} = \operatorname{trace}\{\mathbf{A}\} \operatorname{trace}\{\mathbf{B}\} \qquad \text{Trace} \qquad (1.47)$$

$$\det(\mathbf{A} \otimes \mathbf{B}) = \det(\mathbf{A})^N \det(\mathbf{B})^M \qquad \text{Determinant} \qquad (1.48)$$

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1} \qquad \text{Inversion} \qquad (1.49)$$

In Eq. (1.49), both matrices $\mathbf{A}$ and $\mathbf{B}$ are assumed to be invertible.

1.6 Traces, Vecs & Kronecker Products

There a few important identities that implicate traces, vec's and Kronecker products. As they will come in handy, they are provided below.

First we show certain relationships between vec and $\otimes$. Let us assume that $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{B} \in \mathbb{R}^{N \times P}$ and $\mathbf{C} \in \mathbb{R}^{P \times Q}$. Then,

$$\operatorname{vec}(\mathbf{ABC}) = \left[(\mathbf{C}^T\mathbf{B}^T) \otimes \mathbf{I}_M\right] \operatorname{vec} \mathbf{A} \qquad (1.50a)$$

$$\operatorname{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \operatorname{vec} \mathbf{B} \qquad (1.50b)$$

$$\operatorname{vec}(\mathbf{ABC}) = [\mathbf{I}_Q \otimes (\mathbf{AB})] \operatorname{vec} \mathbf{C} \qquad (1.50c)$$

We've already seen an identity that related trace$\{\cdot\}$ and vec, namely Eq. (1.35). Based on this identity and Eq. (1.50b) we can show that, if $\mathbf{X} \in \mathbb{R}^{M \times N}$

$$\text{trace}\Big\{\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}\Big\} \overset{(1.35)}{=} \text{vec}^T\,\mathbf{X}\,\text{vec}(\mathbf{A}\mathbf{X}\mathbf{B}) \overset{(1.50b)}{=} \text{vec}^T\,\mathbf{X}(\mathbf{B}^T\otimes\mathbf{A})\,\text{vec}\,\mathbf{X} \overset{\substack{(1.18),\\(1.45)\\\Rightarrow}}{}$$

$$\Rightarrow \quad \text{trace}\Big\{\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}\Big\} = \text{vec}^T\,\mathbf{X}\left(\frac{\mathbf{B}^T\otimes\mathbf{A}+\mathbf{B}\otimes\mathbf{A}^T}{2}\right)\text{vec}\,\mathbf{X} \tag{1.51}$$

$$\text{trace}\{\mathbf{X}\mathbf{A}\mathbf{X}\mathbf{B}\} \overset{(1.35)}{=} \underbrace{\text{vec}^T\,\mathbf{X}^T}_{\substack{\overset{(1.31)}{=}(\text{vec}^T\,\mathbf{X})\mathbf{K}_{M,N}^T \overset{(1.33)}{=}(\text{vec}^T\,\mathbf{X})\mathbf{K}_{N,M}}}\text{vec}(\mathbf{A}\mathbf{X}\mathbf{B}) \overset{(1.50b)}{=} (\text{vec}^T\,\mathbf{X})\mathbf{K}_{N,M}(\mathbf{B}^T\otimes\mathbf{A})\,\text{vec}\,\mathbf{X} \overset{\substack{(1.18),\\(1.45),\\(1.33)\\\Rightarrow}}{}$$

$$\Rightarrow \quad \text{trace}\{\mathbf{X}\mathbf{A}\mathbf{X}\mathbf{B}\} = \text{vec}^T\,\mathbf{X}\left[\frac{\mathbf{K}_{N,M}(\mathbf{B}^T\otimes\mathbf{A})+(\mathbf{B}\otimes\mathbf{A}^T)\mathbf{K}_{M,N}}{2}\right]\text{vec}\,\mathbf{X} \tag{1.52}$$

where Eq. (1.18) was used to symmetrize the weight matrix of the previous weighted dot products. In other words, the quantities $\text{trace}\Big\{\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}\Big\}$ and $\text{trace}\{\mathbf{X}\mathbf{A}\mathbf{X}\mathbf{B}\}$, which are called *quadratic terms* in $\mathbf{X}$ are nothing else but weighted dot products of $\text{vec}\,\mathbf{X}$ with itself.

1.7 Frobenius Matrix Norm

The *Frobenius matrix norm* induced by the dot product of Eq. (1.36) is defined as

$$|||\mathbf{A}|||_F \triangleq \sqrt{\langle\mathbf{A},\mathbf{A}\rangle_F} = \sqrt{(\text{vec}\,\mathbf{A})^T\,\text{vec}\,\mathbf{A}} = \|\text{vec}\,\mathbf{A}\|_2 = \sqrt{\text{trace}\{\mathbf{A}^T\mathbf{A}\}} \tag{1.53}$$

This norm has the following noteworthy properties:

$$(1.16, 1.53) \quad \Rightarrow \quad \Big|\Big|\Big|\mathbf{A}^T\Big|\Big|\Big|_F = |||\mathbf{A}|||_F \tag{1.54}$$

which implies that

$$|||[\mathbf{a}_1\cdots\mathbf{a}_N]|||_F = \Big|\Big|\Big|\begin{bmatrix}\mathbf{a}_1^T\\ \vdots\\ \mathbf{a}_N^T\end{bmatrix}\Big|\Big|\Big|_F \overset{\substack{(1.53),\\(1.25),\\(1.17)}}{=} \sqrt{\sum_{n=1}^{N}\|\mathbf{a}_n\|_2^2} \tag{1.55}$$

In other words, the Frobenius norm of a matrix is the L_2 norm of the L_2 norms of the columns or rows of that matrix.

1.8 The diag() Operator

A matrix operator we will find useful from time to time is the $\text{diag}(\cdot)$ operator. It is defined as follows:

- For a matrix argument $A \in \mathbb{R}^{D\times D}$, $\text{diag}(\mathbf{A}) \in \mathbb{R}^D$ returns $\mathbf{A}$'s diagonal elements as a vector.
- For a vector argument $\mathbf{v} \in \mathbb{R}^D$, $\text{diag}(\mathbf{v}) \in \mathbb{R}^{D\times D}$ returns a diagonal matrix, whose diagonal elements are given by $\mathbf{v}$.

This operator often pops up when trying to manipulate Hadamard products and obeys the following identities:

$$\text{diag}(\text{diag}(\mathbf{A})) = \mathbf{A} \odot \mathbf{I} \tag{1.56}$$

$$\text{diag}(\mathbf{a})\,\mathbf{b} = \mathbf{a} \odot \mathbf{b} \tag{1.57}$$

$$\mathbf{x}^T (\mathbf{A} \odot \mathbf{B})\,\mathbf{y} = \text{trace}\Big\{\mathbf{A}^T \,\text{diag}(\mathbf{x})\,\mathbf{B}\,\text{diag}(\mathbf{y})\Big\} \tag{1.58}$$

1.9 Spectral Decomposition of Symmetric Matrices

Definition 1 (Eigen-pair of Square Matrix)**.** Let $\mathbf{A} \in \mathbb{R}^{D\times D}$ and assume that there is a (in general, complex-valued) vector $\mathbf{v} \neq \mathbf{0}$, such that $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ for some complex scalar λ. Then, $(\lambda, \mathbf{v})$ is called an eigen-pair of $\mathbf{A}$, $\mathbf{v}$ is called an eigenvector and λ is called its associated eigenvalue.

From the definition, one can see that, if $\mathbf{v}$ is an eigenvector of $\mathbf{A}$ with associated eigenvalue λ, then so are $-\mathbf{v}$, $\mathbf{v}/\left\|\mathbf{v}\right\|_2$ and, in general, $a\mathbf{v}$ for any constant a associated with the same eigenvalue. Hence, we can always choose eigenvectors of unit Euclidean length, but, still, there will be a sign ambiguity.

For $D \times D$ symmetric matrices, one can show that any of their eigen-pairs is real-valued and that they have at most D distinct eigenvalues. Furthermore, one can show the following Eigen-Value Decomposition (EVD) theorem.

Theorem 1 (Eigenvalue Decomposition)**.** *Let $\mathbf{A} \in \mathbb{R}^{D\times D}$, such that $\mathbf{A}^T = \mathbf{A}$. Then, $\mathbf{A}$ can be factored as*

$$\mathbf{A} = \mathbf{V}\underbrace{\text{diag}(\boldsymbol{\lambda})}_{\boldsymbol{\Lambda}\triangleq}\mathbf{V}^T = \sum_{i=1}^{D}\lambda_i\mathbf{v}_i\mathbf{v}_i^T \tag{1.59}$$

where $\mathbf{V} \triangleq [\mathbf{v}_1 \cdots \mathbf{v}_D]$ is an orthogonal matrix that consists of unit-length eigen-vectors of $\mathbf{A}$ associated with the corresponding eigenvalues in $\boldsymbol{\lambda} \triangleq [\lambda_1 \cdots \lambda_D]^T$. If they eigenvalues are distinct, then the factorization is unique up to a permutation of the eigenpairs and up to the signs of the individual eigenvectors.

Based on Theorem 1 and using properties of the determinant and trace, one can immediately show that

$$\det(\mathbf{A}) = \prod_{i=1}^{D} \lambda_i \tag{1.60}$$

and

$$\text{trace}\{\mathbf{A}\} = \sum_{i=1}^{D} \lambda_i \tag{1.61}$$

and

$$\mathbf{A}^{-1} = \mathbf{V}\,\text{diag}(\boldsymbol{\lambda})^{-1}\,\mathbf{V}^T \tag{1.62}$$

provided that no eigenvalue is 0.

EVD is a versatile tool that allows us to show additional interesting results. An example follows.

Example 2. Let $\mathbf{A} = \mathbf{V}\,\text{diag}(\boldsymbol{\lambda})\,\mathbf{V}^T$ be the EVD of $\mathbf{A}$ and assume that $\boldsymbol{\lambda} \succcurlyeq \mathbf{0}$. Then, there is a real-valued matrix $\mathbf{B}$, s.t. $\mathbf{A} = \mathbf{B}^T\mathbf{B}$.

Proof. Since $\boldsymbol{\lambda} \succcurlyeq \mathbf{0}$, $\mathbf{A}$'s eigenvalues are all non-negative, so their square roots are real-valued and we can rewrite the EVD as

$$\mathbf{A} = \underbrace{\mathbf{V}\,\text{diag}(\boldsymbol{\lambda})^{\frac{1}{2}}}_{\mathbf{B}^T \triangleq}\underbrace{\text{diag}(\boldsymbol{\lambda})^{\frac{1}{2}}\,\mathbf{V}^T}_{=\mathbf{B}} = \mathbf{B}^T\mathbf{B}$$

Notice that $\mathbf{B}^T\mathbf{B}$ is symmetric.

□

Matrix $\mathbf{B}$ can be thought of as the "square root" of the symmetric matrix $\mathbf{A}$.

A result concerning the EVD of Kronecker products follows.

Proposition 1 (Eigen-Pairs of Kronecker Product)**.** *Let* $\mathbf{A}, \in \mathbb{R}^{M\times M}$ *and* $\mathbf{B}, \in \mathbb{R}^{N\times N}$ *be two square matrices with eigen-pairs* $\{(\lambda_m, \mathbf{u}_m)\}_{m=1}^{M}$ *and* $\{(\mu_n, \mathbf{v}_n)\}_{n=1}^{N}$ *respectively. Then,* $\mathbf{A}\otimes\mathbf{B}$ *has eigen-pairs* $\{(\lambda_m\mu_n, \mathbf{u}_m \otimes \mathbf{v}_n)\}_{m=1,n=1}^{M,N}$.

Let us note that, given this result, the validity of Eq. (1.47) and Eq. (1.48) should be more obvious now.

1.10 Definiteness of Symmetric Matrices

The definiteness of a symmetric matrix, in some sense, generalizes the sign of scalar quantities. This fact will become more apparent in the sequel.

Definition 2 (Definiteness)**.** Let $\mathbf{A} \in \mathbb{R}^{D\times D}$, such that $\mathbf{A}^T = \mathbf{A}$. Then,

1. $\mathbf{A}$ is called *positive definite*, denoted $\mathbf{A} \succ 0$, iff $\mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$.
2. $\mathbf{A}$ is called *positive semi-definite*, denoted $\mathbf{A} \succcurlyeq 0$, iff $\mathbf{x}^T\mathbf{A}\mathbf{x} \geq 0$ for all $\mathbf{x} \neq \mathbf{0}$.
3. $\mathbf{A}$ is called *negative definite*, denoted $\mathbf{A} \prec 0$, iff $\mathbf{x}^T\mathbf{A}\mathbf{x} < 0$ for all $\mathbf{x} \neq \mathbf{0}$.
4. $\mathbf{A}$ is called *negative semi-definite*, denoted $\mathbf{A} \preccurlyeq 0$, iff $\mathbf{x}^T\mathbf{A}\mathbf{x} \leq 0$ for all $\mathbf{x} \neq \mathbf{0}$.
5. $\mathbf{A}$ is called *indefinite*, iff $\mathbf{x}^T\mathbf{A}\mathbf{x}$ can take on any sign for all $\mathbf{x} \neq \mathbf{0}$.

One can easily prove statements like:

$$\mathbf{A} \succ 0 \quad \Leftrightarrow \quad \mathbf{A}^k \succ 0, \quad k \in \mathbb{Z} \tag{1.63}$$

$$\mathbf{A} \succ 0, \mathbf{B} \succ 0, a > 0, b > 0 \quad \Rightarrow \quad a\mathbf{A} + b\mathbf{B} \succ 0 \tag{1.64}$$

$$\mathbf{B} \in \mathbb{R}^{M\times N} \quad \Rightarrow \quad \mathbf{B}^T\mathbf{B} \succcurlyeq 0 \tag{1.65}$$

as well as the next proposition. Once again, the EVD usually plays a key role in showing these results, except in the case of Eq. (1.65), which can be shown in a simpler manner.

Proposition 2. *Let* $\mathbf{A} \in \mathbb{R}^{D\times D}$ *be a symmetric matrix with eigenvalues* $\boldsymbol{\lambda}$*. Then,* $\mathbf{A} \succ 0$*, iff* $\boldsymbol{\lambda} \succ \mathbf{0}$*.*

Proof. ($\Rightarrow$) By Definition 2, $\mathbf{A} \succ 0 \quad \Leftrightarrow \quad \mathbf{x}^T\mathbf{A}\mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$. Choose $\mathbf{x} = \mathbf{v}_i$, where $\mathbf{v}_i$ is the unit-length eigenvector of $\mathbf{A}$ associated with λ_i. Notice that $\mathbf{v}_i \neq 0$ by an eigenvector's definition; see Definition 1. Then, $\mathbf{v}_i^T \underbrace{\mathbf{A}\mathbf{v}_i}_{=\lambda_i\mathbf{v_i}} = \lambda_i \underbrace{\mathbf{v}_i^T\mathbf{v}_i}_{=\|\mathbf{v}_i\|_2^2=1} = \lambda_i > 0$ for any $i = 1, \ldots, D$.

($\Leftarrow$) Assume $\boldsymbol{\lambda} \succ \mathbf{0}$, *i.e.*, $\lambda_i > 0$. Then, $\mathbf{x}^T\mathbf{A}\mathbf{x} = \underbrace{\mathbf{x}^T\mathbf{V}}_{\mathbf{y}^T\triangleq}\operatorname{diag}(\boldsymbol{\lambda})\underbrace{\mathbf{V}^T\mathbf{x}}_{=\mathbf{y}} = \mathbf{y}^T\operatorname{diag}(\boldsymbol{\lambda})\,\mathbf{y} = \sum_{i=1}^D \lambda_i y_i^2 > 0$, because all $\lambda_i > 0$ and, if $\mathbf{x} \neq 0$, then $\mathbf{y} \neq 0$, since $\mathbf{V}$, being orthogonal, is invertible.

□

As a side note, it should be apparent that, if $\mathbf{A} \succ 0$, then $\mathbf{A}$ is invertible.

At this point it probably has become apparent that the eigenvalues of a symmetric matrix may play a significant role in determining its definiteness. As a matter of fact, the next result completely determines this role. Its proof is omitted here, because it requires notions and skills related to constrained optimization.

Theorem 2. *Let* $\mathbf{A} \in \mathbb{R}^{D\times D}$*, such that* $\mathbf{A}^T = \mathbf{A}$*. Also, let* λ_1 *and* λ_D *be respectively its largest and smallest eigenvalues. Then, for any* $\mathbf{x} \in \mathbb{R}^D$

$$\lambda_D \|\mathbf{x}\|_2^2 \leq \mathbf{x}^T\mathbf{A}\mathbf{x} \leq \lambda_1 \|\mathbf{x}\|_2^2 \tag{1.66}$$

Theorem 2 makes it clear, for example, that, if all eigenvalues of the matrix are non-negative (non-positive), then the matrix is positive (negative) semi-definite. Also, if it has mixed-sign eigenvalues, it is indefinite. In conclusion, the definiteness of a symmetric matrix is established by the signs of its eigenvalues.

Next, we show a result regarding the definiteness of a Kronecker product, that follows immediately from Proposition 1.

Corollary 1 (Definiteness of Kronecker Product)**.** *Consider the Kronecker product* $\mathbf{A} \otimes \mathbf{B}$*. Then,*

- *If* $\mathbf{A} \succ 0$ *and* $\mathbf{B} \succ 0$*, then* $\mathbf{A} \otimes \mathbf{B} \succ 0$
- *If* $\mathbf{A} \succ 0$ *or* $\mathbf{A} \succcurlyeq 0$ *and* $\mathbf{B} \succ 0$ *or* $\mathbf{B} \succcurlyeq 0$*, then* $\mathbf{A} \otimes \mathbf{B} \succcurlyeq 0$

Finally, we conclude this section with a similar result pertaining to the definiteness of Hadamard products.

Proposition 3 (Definiteness of Hadamard Product)**.** *Consider the Hadamard product* $\mathbf{A} \odot \mathbf{B}$*. Then,*

- *If* $\mathbf{A} \succ 0$ *and* $\mathbf{B} \succ 0$*, then* $\mathbf{A} \odot \mathbf{B} \succ 0$
- *If* $\mathbf{A} \succ 0$ *or* $\mathbf{A} \succcurlyeq 0$ *and* $\mathbf{B} \succ 0$ *or* $\mathbf{B} \succcurlyeq 0$*, then* $\mathbf{A} \odot \mathbf{B} \succcurlyeq 0$

The last proposition can be shown by using the definitions of definiteness, the EVDs of the matrices involved and Eq. (1.57).

1.11 Some Additional Material

This section provides some additional materials that were discussed in class.

GCA's Comment: *Joseph, please add your lecture notes here.*

1.12 Questions To Consider

1. Show Eq. (1.60) through Eq. (1.62).
2. Show Eq. (1.63) through Eq. (1.65).

3. If $\mathbf{X} \triangleq [\mathbf{x}_1 \cdots \mathbf{x}_N]^T \in \mathbb{R}^{N \times D}$ and $\mathbf{A} \in \mathbb{R}^{D \times D}$ is symmetric, show that

$$\sum_{n=1}^{N} \mathbf{x}_n^T \mathbf{A} \mathbf{x}_n = \text{trace}\Big\{\mathbf{X}\mathbf{A}\mathbf{X}^T\Big\} \tag{1.67}$$

1.13 Extra Material: Inner Products, Vector Norms & Metrics

The notion of an inner product generalizes the concept of the vector dot product. In some respect, inner products can be thought of as quantifying the degree of similarity between a pair of vectors.

Definition 3 (Inner Product)**.** A function $\langle \cdot, \cdot \rangle : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is called a (real-valued) *inner product* of the vector space $\mathbb{R}^D$ if it obeys the following axioms for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^D$ and $a \in \mathbb{R}$:

$$\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle \qquad \text{Symmetry} \tag{1.68}$$
$$\langle a\mathbf{x}, \mathbf{y} \rangle = a \langle \mathbf{x}, \mathbf{y} \rangle \qquad \text{Homogeneity} \tag{1.69}$$
$$\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{z}, \mathbf{y} \rangle \qquad \text{Superposition} \tag{1.70}$$
$$\langle \mathbf{x}, \mathbf{x} \rangle \geq 0 \qquad \text{Positive-definiteness} \tag{1.71}$$
$$\langle \mathbf{x}, \mathbf{x} \rangle = 0 \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{0} \tag{1.72}$$

Based on this definition, the ordinary dot product is an inner product, but it is not the only function that satisfies the aforementioned axioms. The next proposition identifies another inner product, often referred to as a *weighted dot product*.

Proposition 4. *For a positive definite symmetric matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$, the bivariate function*

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{A}} \triangleq \mathbf{x}^T \mathbf{A} \mathbf{y} \tag{1.73}$$

is an inner product on $\mathbb{R}^D$.

The proof of this proposition is trivial and, therefore, omitted. Notice that this inner product yields the dot product for $\mathbf{A} = \mathbf{I}_D$.

On the other hand, vector norms quantify the "size" or "length" of vectors in some sense. It turns out that, apart from some universally agreed-upon properties, a vector norm can be otherwise quite arbitrary.

Definition 4 (Norm)**.** A function $\|\cdot\| : \mathbb{R}^D \to \mathbb{R}$ is called a *vector norm* of the vector space $\mathbb{R}^D$ if it obeys the following axioms for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ and $a \in \mathbb{R}$:

$$\|\mathbf{x}\| \geq 0 \qquad \text{Non-Negativity} \tag{1.74}$$

and

$$\|\mathbf{x}\| = 0 \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{0} \tag{1.75}$$

$$\|a\mathbf{x}\| = |a| \, \|\mathbf{x}\| \qquad \text{Absolute Homogeneity} \tag{1.76}$$

$$\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \qquad \text{Triangle Inequality/Sub-additivity} \tag{1.77}$$

Functions that obey all the above axioms except Eq. (1.75) are called *vector semi-norms*. The L_p vector norms we encountered earlier satisfy all these axioms (hence, their name). Vector norms can be constructed from inner products.

Proposition 5 (Induced Vector Norm)**.** *If $\langle \cdot, \cdot \rangle$ is an inner product on $\mathbb{R}^D$, then the function evaluated on $\mathbf{x} \in \mathbb{R}^D$*

$$\|\mathbf{x}\| \triangleq \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} \tag{1.78}$$

is a vector norm on $\mathbb{R}^D$. This norm is referred to as the norm induced by the given inner product.

The proof, again, is straightforward is omitted. A useful induced norm is the following.

Definition 5 (Weighted Euclidean Norm)**.** For positive-definite symmetric matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$, the norm induced by the weighted dot product

$$\|\mathbf{x}\|_{\mathbf{A}} \triangleq \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}} \tag{1.79}$$

is called the *weighted Euclidean norm* of $\mathbf{x} \in \mathbb{R}^D$ with weight matrix $\mathbf{A}$.

It can be shown that, if $\mathbf{A} \succcurlyeq 0$, then $\|\mathbf{x}\|_{\mathbf{A}}$ is only a semi-norm.

Next, we'll generalize the notion of a distance.

Definition 6 (Metric)**.** A bi-variate function $d(\cdot, \cdot) : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is called a *metric* or *distance* on the vector space $\mathbb{R}^D$, if it obeys the following axioms for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^D$ and $a \in \mathbb{R}$:

$$d(\mathbf{x}, \mathbf{y}) \geq 0 \qquad \text{Non-Negativity} \tag{1.80}$$

and

$$d(\mathbf{x}, \mathbf{y}) = 0 \quad \Leftrightarrow \quad \mathbf{x} = \mathbf{y} \qquad \text{Identity Of Indiscernibles} \tag{1.81}$$

$$d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x}) \qquad \text{Symmetry} \tag{1.82}$$

$$d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z}) \qquad \text{Triangle Inequality} \tag{1.83}$$

If a function satisfies all aforementioned axioms except Eq. (1.81), it is called a *pseudo-metric*.

Metrics can be naturally defined based on norms and vice versa, as the following, easy-to-prove result states.

Proposition 6 (Norm-Induced Metric and Metric-Induced Norm)**.** *Given a (semi-)norm* $\|\cdot\|$ *on* $\mathbb{R}^D$*, then*

$$d(\mathbf{x}, \mathbf{y}) \triangleq \|\mathbf{x} - \mathbf{y}\| \tag{1.84}$$

is a (pseudo-)metric on $\mathbb{R}^D$*. Conversely, given a (pseudo-)metric* $d(\cdot, \cdot)$ *on* $\mathbb{R}^D$*, then*

$$\|\mathbf{x}\| \triangleq d(\mathbf{x}, \mathbf{0}) \tag{1.85}$$

is a (semi-)norm on $\mathbb{R}^D$*.*

In plain English, given a distance (metric), the length (norm) of a vector can be defined as its distance from the origin. Conversely, given a length, a distance between two vectors can be defined based on the length of their difference. A particular metric that we will encounter is the one defined next.

Definition 7 (Mahalanobis Distance)**.** For positive-definite symmetric matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$, the metric induced by the weighted Euclidean norm

$$d_{\mathbf{A}}(\mathbf{x} - \mathbf{y}) \triangleq \|\mathbf{x} - \mathbf{y}\|_{\mathbf{A}} \tag{1.86}$$

is called the *Mahalnobis distance* between $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$ with weight matrix $\mathbf{A}$.

Notice that, if $\mathbf{A} \succcurlyeq 0$, one can easily show that $d_{\mathbf{A}}(\cdot, \cdot)$ is just a pseudo-metric.

1.14 Extra Material: Graphing the Locus of Constant Mahalanobis Distance from a Given Point

For a given positive-definite symmetric matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$, a fixed point $\mathbf{x}_o \in \mathbb{R}^D$ and a given radius $r \geq 0$, $\mathcal{S} \triangleq \left\{\mathbf{x} \in \mathbb{R}^D : \|\mathbf{x} - \mathbf{x}_o\|_{\mathbf{A}} = r\right\}$ is the set of all points in $\mathbb{R}^D$, whose Mahalanobis distance with weight matrix $\mathbf{A}$ from $\mathbf{x}_o$ equals r. It turns out that, for $D = 2$ this set is an ellipse on the plane with center $\mathbf{x}_o$, while for general D it describes the surface of a hyper-ellipsoid embedded in $\mathbb{R}^D$ and centered at the same point. Its orientation is determined by the eigenvectors of $\mathbf{A}$ and its eccentricity (the various ratios between the lengths of its axes) is determined by $\mathbf{A}$'s eigenvalues. It is useful to be able to graph this ellipse on the plane ($D = 2$). This subsection discusses a method to do so.

If the EVD of $\mathbf{A}$ is given as $\mathbf{A} \overset{\text{EVD}}{=} \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, define the matrix $\mathbf{B} \triangleq \mathbf{V}\mathbf{\Lambda}^{-\frac{1}{2}}$, such that $\mathbf{B}^T\mathbf{A}\mathbf{B} = \mathbf{I}$. Notice that $\mathbf{\Lambda}^{-\frac{1}{2}}$ (hence, $\mathbf{B}$) exists (*i.e.*, has real-valued, finite entries), since, by hypothesis, $\mathbf{A} \succ 0$ and, therefore, all its eigenvalues are strictly positive. Then, $\mathcal{S}$ for $D = 2$ is equivalently given by

$$\mathbf{x} = r\mathbf{B}\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix} + \mathbf{x}_o \qquad \theta \in [0, 2\pi) \tag{1.87}$$

As θ goes from 0 to 2π, $\mathbf{x}$ will trace out the ellipse in question.

To convince ourselves that this is true, we verify this as follows: for some θ, let $\mathbf{z} \triangleq [\cos(\theta) \quad \sin(\theta)]^T$ and notice that $\|\mathbf{z}\|_2 = 1$. Then, based on Eq. (1.87), $\mathbf{x} - \mathbf{x}_o = r\mathbf{B}\mathbf{z}$ and

$$\|\mathbf{x} - \mathbf{x}_o\|_{\mathbf{A}} = \sqrt{(\mathbf{x} - \mathbf{x}_o)^T\mathbf{A}(\mathbf{x} - \mathbf{x}_o)} = \sqrt{r^2(\mathbf{B}\mathbf{z})^T\mathbf{A}(\mathbf{B}\mathbf{z})} = r\sqrt{\mathbf{z}^T\underbrace{\mathbf{B}^T\mathbf{A}\mathbf{B}}_{=\mathbf{I}}\mathbf{z}} =$$

$$= r\sqrt{\mathbf{z}^T\mathbf{z}} = r\underbrace{\|\mathbf{z}\|_2}_{=1} = r$$

In order to come up with different types of such ellipses (orientation and eccentricity), it useful to know that the eigenvector matrix (the matrix that contains the eigenvectors as columns) of a 2×2 symmetric matrix $\mathbf{A}$ can always be taken as

$$\mathbf{V} = \begin{bmatrix}\cos(\phi) & -\sin(\phi)\\ \sin(\phi) & \cos(\phi)\end{bmatrix} \tag{1.88}$$

for some $\phi \in [0, 2\pi)$. Using trigonometric identities, it is easy to check that $\mathbf{V}^T\mathbf{V} = \mathbf{V}\mathbf{V}^T = \mathbf{I}$. $\mathbf{A}$'s second eigenvector (second column of $\mathbf{V}$; call it $\mathbf{v}_2$) is always $\pi/2$ radians counter-clockwise ahead of its first eigenvector (first column of $\mathbf{V}$; call it $\mathbf{v}_1$). If λ_1 is $\mathbf{A}$'s largest eigenvalue, then the major axis of the ellipse will be along the $\mathbf{v}_2$ direction. Additionally, the semi-major axis' length will be r/λ_2, while the semi-minor axis' length will be r/λ_1.

ECE5268: Theory of Neural Networks **Spring 2020**

Lecture 2: Elements of Multivariate Derivatives & Optimization

Tue, Jan 21, 2020

Lecturer: GCA *Scribe(s): GCA*

Note: *This header style is courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

This lecture provides a quick (over/re)view of the most essential notions regarding multi-variate derivatives. Next, we discuss how some of these play an important role in optimization of scalar functions of many variables.

2.1 Multi-Variate Derivatives

The notions of the first- and higher-order derivatives of a a uni-variate function (function of a single independent variable) can be generalized to scalar multi-variate functions $f : \mathbb{R}^D \to \mathbb{R}$; in the context of Machine Learning (ML), the role of such functions is going to be fulfilled by loss/cost functions to be minimized. Derivatives of vector- or matrix-valued functions are beyond our scope, with the exception of the Jacobian and Hessian; an excellent reference that covers multi-variate derivatives in all their generality and is freely-accessible online is [3]. Note that, throughout the material of this lecture, we will assume that all partial partial derivatives involved exist, *i.e.*, the functions involved are (once and, potentially, twice) differentiable.

Notation-wise, when a function f depends on many variables that can be (or are) arranged into a vector, *e.g.*, $x_1, \ldots, x_D$, then its value will be denoted as in more compact form $f(\mathbf{x})$, where $\mathbf{x} \triangleq [x_1 \cdots x_D]^T$, instead of $f(x_1, \ldots, x_D)$. Similarly, when f depends on many variables than can be (or are) arranged as a $M \times N$ matrix $\mathbf{X}$, we will denote its values as $f(\mathbf{X})$.

Caution: With the exception of Section 4.4, all matrices of independent variables (*e.g.*, $\mathbf{X}$ in the previous paragraph) w.r.t. which we will compute a variety of derivatives will be assumed to lack any special structure, meaning that (i) all their elements must be variables (*e.g.*, none of the elements are constants, such as 0) and (ii) their elements must be mutually independent. This precludes, for example, symmetric, (tri-)diagonal, Toeplitz, etc. matrices. Section 4.4 very briefly discusses the case, where the independent variables of scalar functions are arranged into symmetric matrices.

2.1.1 Derivative Operators

Since there are many variables involved, it shouldn't be surprising that multivariate derivatives of f involve partial derivatives w.r.t. each one of the variables. These quantities, hence, can be

arranged as vectors or matrices in some way that is notationaly meaningful, convenient and even practical. For example, if the variables are arranged as a vector (matrix) inside f, it makes sense to arrange them in a single vector (matrix). We start off with some definitions.

Definition 8 (Gradient & Derivative Operators)**.** If $\mathbf{x} \in \mathbb{R}^D$ is a vector of variables, then the *gradient operator* w.r.t. $\mathbf{x}$ is defined as the D-element column-vector operator

$$\frac{d}{d\mathbf{x}} \triangleq \begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \vdots \\ \dfrac{\partial}{\partial x_D} \end{bmatrix} \tag{2.1}$$

and the *derivative operator* w.r.t. $\mathbf{x}$ is defined as the D-element row-vector operator

$$\frac{d}{d\mathbf{x}^T} \triangleq \left(\frac{d}{d\mathbf{x}}\right)^T = \begin{bmatrix} \dfrac{\partial}{\partial x_1} & \cdots & \dfrac{\partial}{\partial x_D} \end{bmatrix} \tag{2.2}$$

Definition 9 (Gradient Matrix & Derivative Operators)**.** If $\mathbf{X} \in \mathbb{R}^{M \times N}$ is a matrix of variables, then the *matrix gradient operator* w.r.t. $\mathbf{X}$ is defined as the $M \times N$ matrix operator

$$\frac{d}{d\mathbf{X}} \triangleq \begin{bmatrix} \dfrac{\partial}{\partial x_{1,1}} & \cdots & \dfrac{\partial}{\partial x_{1,N}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial}{\partial x_{M,1}} & \cdots & \dfrac{\partial}{\partial x_{M,N}} \end{bmatrix} \tag{2.3}$$

and the *matrix derivative operator* w.r.t. $\mathbf{X}$ is defined as the $N \times M$ matrix operator

$$\frac{d}{d\mathbf{X}^T} \triangleq \left(\frac{d}{d\mathbf{X}}\right)^T = \begin{bmatrix} \dfrac{\partial}{\partial x_{1,1}} & \cdots & \dfrac{\partial}{\partial x_{1,M}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial}{\partial x_{N,1}} & \cdots & \dfrac{\partial}{\partial x_{N,M}} \end{bmatrix} \tag{2.4}$$

One can see that Eq. (2.1) and Eq. (2.2) are generalizations of Eq. (2.3) and Eq. (2.4) respectively, since the column vector $\mathbf{x}$ can be thought o as a single column of $\mathbf{X}$.

Alternatively, the matrix gradient operator can be expressed as a gradient vector via vec as follows:

$$\frac{d}{d\,\mathrm{vec}\,\mathbf{X}} \triangleq \begin{bmatrix} \dfrac{\partial}{\partial x_{1,1}} \\ \dfrac{\partial}{\partial x_{2,1}} \\ \vdots \\ \dfrac{\partial}{\partial x_{M-1,N}} \\ \dfrac{\partial}{\partial x_{M,N}} \end{bmatrix} = \mathrm{vec}\left(\frac{d}{d\mathbf{X}}\right) \tag{2.5}$$

Note that we will often denote gradient operators simply as ∇ or ∇^T for their transpose form. When necessary to avoid ambiguity, we will subscript them with the variable, *e.g.*, $\nabla_{\mathbf{X}}$.

We define as the *second-order derivative operator* or *Hessian operator* as

$$\frac{d^2}{d\mathbf{x}d\mathbf{x}^T} \triangleq \begin{bmatrix} \dfrac{\partial^2}{\partial x_1^2} & \cdots & \dfrac{\partial^2}{\partial x_1 \partial x_D} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2}{\partial x_D \partial x_1} & \cdots & \dfrac{\partial^2}{\partial x_D^2} \end{bmatrix} = \frac{d}{d\mathbf{x}}\frac{d}{d\mathbf{x}^T} = \begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \vdots \\ \dfrac{\partial}{\partial x_D} \end{bmatrix} \begin{bmatrix} \dfrac{\partial}{\partial x_1} & \cdots & \dfrac{\partial}{\partial x_D} \end{bmatrix} \tag{2.6}$$

The previous definition implies that the Hessian operator can be thought of as the outer product of the gradient operator with itself. Since this operator is applied to twice-differentiable functions, the order in which the mixed partial derivatives appear in its elements is irrelevant, we see that the operator is symmetric. We will occasionally use ∇^2 to denote the Hessian operator, or $\nabla^2_{\mathbf{x}}$ to indicate the independent variable.

2.1.2 Derivatives of Scalar Functions

Based on these definitions, we can define the *gradient vector* of $f : \mathbf{x} \mapsto f(\mathbf{x}) \in \mathbb{R}$ as

$$\frac{df(\mathbf{x})}{d\mathbf{x}} \triangleq \begin{bmatrix} \dfrac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \dfrac{\partial f(\mathbf{x})}{\partial x_D} \end{bmatrix} \tag{2.7}$$

and the transpose of the previous quantity as its derivative. Furthermore, the *gradient matrix* of $f : \mathbf{X} \in \mathbb{R}^{M\times N} \mapsto f(\mathbf{X}) \in \mathbb{R}$ will be defined as

$$\frac{df(\mathbf{X})}{d\mathbf{X}} \triangleq \begin{bmatrix} \frac{\partial f(\mathbf{X})}{\partial x_{1,1}} & \cdots & \frac{\partial f(\mathbf{X})}{\partial x_{1,N}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(\mathbf{X})}{\partial x_{M,1}} & \cdots & \frac{\partial f(\mathbf{X})}{\partial x_{M,N}} \end{bmatrix} \tag{2.8}$$

where, again, the transpose of the previous quantity denotes the derivative matrix of f.

Both the gradient vector and matrices of a scalar function are considered first-order derivatives. They only differ in how their first-order partial derivatives of f are organized. In the case where $N = 1$ ($\mathbf{X}$ has only one column), then the gradient matrix of Eq. (2.8) coincides with the gradient vector of Eq. (2.7).

When it comes to second-order derivatives of f, we realize that many mixed second-order partial derivatives are involved. Let's assume that f is a scalar function of $\mathbf{x} \in \mathbb{R}^D$.

Based on the Hessian operator's definition in Eq. (2.6), the *Hessian matrix* of f is defined as

$$\frac{d^2 f(\mathbf{x})}{d\mathbf{x} d\mathbf{x}^T} \triangleq \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_D} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_D \partial x_1} & \cdots & \frac{\partial^2 f(\mathbf{x})}{\partial x_D^2} \end{bmatrix} \tag{2.9}$$

Notice that a Hessian matrix is always symmetric for a twice-differentiable scalar function.

For functions that depend on variables that, originally, are arranged as matrices (*i.e.*, $f : \mathbf{X} \mapsto f(\mathbf{X})$), their Hessian matrix is composed by first vectorizing their variables, then computing $\frac{d^2 f(\mathbf{X})}{d \operatorname{vec} \mathbf{X} d \operatorname{vec}^T \mathbf{X}}$ and, finally, bring this expression into a concise form, if possible. To clarify, if $\mathbf{X} \triangleq [\mathbf{x}_1 \cdots \mathbf{x}_N]$, then the last quantity takes the block matrix form

$$\frac{d^2 f(\mathbf{X})}{d \operatorname{vec} \mathbf{X} d \operatorname{vec}^T \mathbf{X}} = \begin{bmatrix} \frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{x}_1 \partial \mathbf{x}_1^T} & \cdots & \frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{x}_1 \partial \mathbf{x}_N^T} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{x}_N \partial \mathbf{x}_1^T} & \cdots & \frac{\partial^2 f(\mathbf{X})}{\partial \mathbf{x}_N \partial \mathbf{x}_N^T} \end{bmatrix} \tag{2.10}$$

The gradient and the Hessian generalize the notion of the first- and second-order derivatives of a function to scalar multi-variate functions. For example, for an analytic multi-variate function f, its Taylor series expansion around $\mathbf{x}_o$ is given as

$$f(\mathbf{x}) = f(\mathbf{x}_o) + \mathbf{g}(\mathbf{x}_o)^T(\mathbf{x} - \mathbf{x}_o) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_o)^T\mathbf{H}(\mathbf{x}_o)(\mathbf{x} - \mathbf{x}_o) + \text{H.O.T.} \tag{2.11}$$

where $\mathbf{g}(\mathbf{x}_o)$ and $\mathbf{H}(\mathbf{x}_o)$ are respectively f's gradient and Hessian matrix evaluated at $\mathbf{x} = \mathbf{x}_o$.

2.1.3 Derivatives of Vector-Valued Functions

When it comes to vector-valued functions, an important matrix that comes into play is the *Jacobian* matrix. Let $\mathbf{f} : \mathbb{R}^M \to \mathbb{R}^N$ be differentiable in the sense that all its first-order partial derivatives exist everywhere. Then, the Jacobian of $\mathbf{f}$ is given as

$$\frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}^T} \triangleq \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_N(\mathbf{x})}{\partial x_M} \end{bmatrix} \in \mathbb{R}^{N \times M} \tag{2.12}$$

The Jacobian matrix generalizes the notion of a derivative to vector-valued functions. When $M = N = D$ and its determinant is non-zero everywhere, then, it turns out that $\mathbf{f}$ is an invertible mapping. Furthermore, the Taylor expansion of f at $\mathbf{x}_o$, if it is analytic, is of the form

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_o) + \mathbf{J}(\mathbf{x}_o)(\mathbf{x} - \mathbf{x}_o) + \text{H.O.T.} \tag{2.13}$$

2.2 Computation of Multi-Variate Derivatives

One can use the definitions of gradient vector/matrix, as well as of the Hessian and Jacobian matrices to compute these quantities of given functions in a component-wise/element-wise fashion, *i.e.*, by computing the elements of these quantities one by one.

This "element-wise" approach, however, presents disadvantages, when these multi-variate derivatives can be neatly expressed as products of matrices and/or vectors; and they often do. The disadvantages are two-fold:

- First, in such a scenario, element-wise computations often yield multiple, elaborate sums for each element. These expressions may obscure the fact that the multi-variate derivative takes on a much more concise form as a product of matrices/vectors. Moreover, re-writing the multi-variate derivative as a product of matrices/vectors based on the particular form that their elements take, can be a tedious, if not difficult, task.

- Having the expressions for each element of a multi-variate derivative is perhaps ideal, when implementing code via `for`-loops in a low-level programming language. But, when using higher level programming languages, that offer optimized/parallelized implementation of linear algebra operations, a matrix/vector product form of a multi-variate derivative is more preferable, as it allows for authoring more readable code.

In the next lecture, we will discuss differentials of functions, which, along with some useful results pertaining to them, allow us often enough to compute and express multi-variate derivatives as matrix/vector products.

2.3 Elements of Optimization

In this section we will show the role that the gradient vector $\mathbf{g}$ and the Hessian matrix $\mathbf{H}$ of a scalar function f that depends on $\mathbf{x} \in \mathbb{R}^D$ play in identifying minimizers and maximizers of f.

GCA's Comment: *Micah, please add your lecture notes here about stationary points of f, first- and second-order conditions for minimizers and maximizers.*

If we take the first derivative, the gradient vector ($\mathbf{g}$), of the function f w.r.t. its parameters, $\mathbf{x}$ and equate it to zero, the values of $\mathbf{x}$ obtained, defined as $\mathbf{x}^*$, are the *stationary points* of f.

$$\mathbf{g} = \frac{df(\mathbf{x})}{d\mathbf{x}}\bigg|_{\mathbf{x}^*} = \underline{0} \tag{2.14}$$

These are the points where the slope of the tangent line on the function is zero. The gradient vector indicates the direction and magnitude of slope of steepest ascent. We can also look at the curvature of f by taking the second order derivative of f (assuming that f is twice differentiable). If

$$\frac{d^2 f(\mathbf{x})}{d\mathbf{x}^2}\bigg|_{\mathbf{x}=\mathbf{x}^*} > 0 \tag{2.15}$$

then f in the neighborhood of $\mathbf{x}^*$ is convex and $\mathbf{x}^*$ is a *local minimizer*. If

$$\frac{d^2 f(\mathbf{x})}{d\mathbf{x}^2}\bigg|_{\mathbf{x}=\mathbf{x}^*} < 0 \tag{2.16}$$

then f in the neighborhood of $\mathbf{x}^*$ is concave and $\mathbf{x}^*$ is a *local maximizer*. If $\mathbf{x}^*$ are neither local minimizers or local maximizers, then they are points of inflexion or *saddle points*.

ECE5268: Theory of Neural Networks **Spring 2020**

Lecture 3: Multi-Variate Differentials

Thu, Jan 23, 2020

Lecturer: GCA *Scribe(s): GCA*

Note: *This header style is courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

This lecture provides additional details on some useful properties of differentials that apply to both scalar-valued and vector-valued functions.

3.1 Differentials

Given, say, a scalar function, computing its multi-variate derivatives by computing their partial derivatives element-wise, one by one, can be a very daunting task and very much prone to errors except, perhaps, for some very simple functions. Fortunately, though, there is a simpler way to compute them via the notion of differentials, which is examined here.

A rigorous exposition of differentials will be avoided in these notes; the interested reader can resort to [3] for such thing. Instead, we will approach this topic in a very casual and practical manner. Recall that, for a scalar function f of a single variable, you can think of its differential $df(x) = f'(x)dx$ at x as the infinitesimal change in f, when x is perturbed by an infinitesimal change dx. Previously, f' denotes f's first-order derivative. In what follows, this notion is generalized to multiple variables.

Differentials can be applied to many independent variables as well as functions of them that can be vector- as well as matrix-valued. Note that, for $\mathbf{A} \in \mathbb{R}^{M\times N}$, whose entries are functions of some independent variables, we define

$$d\mathbf{A} \triangleq \begin{bmatrix} da_{1,1} & \cdots & da_{1,N} \\ \vdots & \ddots & \vdots \\ da_{M,1} & \cdots & da_{M,N} \end{bmatrix} \tag{3.1}$$

It is instructional to think of $da_{m,n}$ as the derivative of $a_{m,n}$ w.r.t. an unspecified independent variable times its differential.

In what follows, the matrices $\mathbf{A}$ and $\mathbf{B}$ are assumed to be dependent on a (unspecified) number of independent variables, unless stated otherwise, and that they have the right dimensions, so the following statements are meaningful. Matrices are used here to render the results as general as

possible. However, they apply also to the case of scalars and row/column vector quantities, since the latter can be viewed as special kinds of matrices.

$$d(\mathbf{A}^T) = (d\mathbf{A})^T \tag{3.2}$$
$$d\,\mathrm{vec}\,\mathbf{A} = \mathrm{vec}\,d\mathbf{A} \tag{3.3}$$
$$d\,\mathrm{trace}\{\mathbf{A}\} = \mathrm{trace}\{d\mathbf{A}\} \tag{3.4}$$
$$\mathbf{A}\text{ constant} \quad \Leftrightarrow \quad d\mathbf{A} = \mathbf{O} \tag{3.5}$$
$$d(\mathbf{A}+\mathbf{B}) = d\mathbf{A} + d\mathbf{B} \tag{3.6}$$
$$d(\mathbf{AB}) = (d\mathbf{A})\mathbf{B} + \mathbf{A}(d\mathbf{B}) \tag{3.7}$$
$$d(\mathbf{A}\otimes\mathbf{B}) = (d\mathbf{A})\otimes\mathbf{B} + \mathbf{A}\otimes(d\mathbf{B}) \tag{3.8}$$
$$d(\mathbf{A}\odot\mathbf{B}) = (d\mathbf{A})\odot\mathbf{B} + \mathbf{A}\odot(d\mathbf{B}) \tag{3.9}$$

Additionally, from Eq. (3.5) and Eq. (3.7), if $\mathbf{A}$ is a constant matrix (or, even, scalar), then $d(\mathbf{AB}) = \mathbf{A}(d\mathbf{B})$. Hence, overall, we observe that the differential is a linear operator that commutes with transposition, vectorization and trace operators. Furthermore, with respect to products (even of Kronecker or Hadamard kind) it behaves like Leibniz's product rule for the ordinary, scalar derivative that we are familiar with.

A few important differentials are listed below. The conditions accompanying each differential need to hold for every set of values of the independent variables on which the square matrix $\mathbf{A}$ depends.

$$d\mathbf{A}^{-1} = -\mathbf{A}^{-1}(d\mathbf{A})\mathbf{A}^{-1} \quad \text{if } \exists\ \mathbf{A}^{-1} \tag{3.10}$$
$$d\det(\mathbf{A}) = \det(\mathbf{A})\,\mathrm{trace}\Big\{(d\mathbf{A})^T\mathbf{A}^{-T}\Big\} \quad \text{if } \exists\ \mathbf{A}^{-1} \tag{3.11}$$

where $\mathbf{A}^{-T}$ stands for $(\mathbf{A}^T)^{-1}$. A derivation of Eq. (3.11) can be found in [3] and is rather involved. On the other hand, Eq. (3.10) is easily derived as follows:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \quad \Rightarrow \quad d\left(\mathbf{A}\mathbf{A}^{-1}\right) = \underbrace{d\mathbf{I}}_{=\mathbf{O}} \quad \overset{(3.7)}{\Leftrightarrow} \quad (d\mathbf{A})\,\mathbf{A}^{-1} + \mathbf{A}d\mathbf{A}^{-1} = \mathbf{O} \quad \overset{\mathbf{A}^{-1}\cdot}{\Rightarrow}$$
$$\Rightarrow \quad \mathbf{A}^{-1}(d\mathbf{A})\mathbf{A}^{-1} + \underbrace{\mathbf{A}^{-1}\mathbf{A}}_{=\mathbf{I}}d\mathbf{A}^{-1} = \mathbf{O} \quad \Leftrightarrow \quad d\mathbf{A}^{-1} = -\mathbf{A}^{-1}(d\mathbf{A})\mathbf{A}^{-1}$$

Pay special attention to the deliberate use of parentheses to specify which quantity the differential operator d is applied to.

ECE5268: Theory of Neural Networks **Spring 2020**

Lecture 4: Multi-Variate Differentials (Part 2)

Tue, Jan 28, 2020

Lecturer: GCA *Scribe(s): GCA*

Note: *This header style is courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

In this lecture we discuss the use of differentials in computing gradient vectors and matrices and Jacobian matrices.

Section 4.4 discusses gradient matrices w.r.t. symmetric matrices; such gradient matrices require special care.

Finally, Section 4.5 and Section 4.6 present some advanced materials pertaining to differentials that allow one to identify the Hessian matrix (or some of its blocks) of a scalar function. Apart from the notion of *partial differential* discussed in the latter section, we won't make use of the remaining material covered in these two sections.

4.1 Gradient Identification Rules

Differentials play an instrumental role in identifying the gradient vector or gradient matrix of a scalar function thanks to the following results, most of which are proved and discussed in [3] in much more detail. In summary, the general idea is that, when one requires to compute a gradient quantity of a scalar function w.r.t. a parameter, one computes the differential of this function in terms of differentials of the paramater; after bringing this expression in a suitable form dictated by a gradient identification rule, one can immediately read off the gradient. In what follows, we'll use $\equiv$ as a special case of equality, which stands for "coincides with."

Theorem 3 (Gradient Identification Rule)**.** *Let $f : \mathbb{R}^{M\times N} \to \mathbb{R}$ be differentiable. Then,*

$$df(\mathbf{X}) = \mathrm{trace}\Big\{(d\mathbf{X}^T)\mathbf{G}(\mathbf{X})\Big\} \quad \Leftrightarrow \quad \frac{df(\mathbf{X})}{d\mathbf{X}} \equiv \mathbf{G}(\mathbf{X}) \tag{4.1}$$

If $f : \mathbb{R}^D \to \mathbb{R}$ instead, the previous expression specializes to

$$df(\mathbf{x}) = (d\mathbf{x}^T)\mathbf{g}(\mathbf{x}) \quad \Leftrightarrow \quad \frac{df(\mathbf{x})}{d\mathbf{x}} \equiv \mathbf{g}(\mathbf{x}) \tag{4.2}$$

The second result coincides with what we have been exposed to in multi-variate calculus; it is the fact that the total differential of f is given as $df(\mathbf{x}) = \sum_{i=1}^{D} \frac{\partial f(\mathbf{x})}{\partial x_i} dx_i$. Using this theorem, one can show, for example, the following list of a few simple, yet important, gradients:

$$\frac{d\mathbf{x}^T\mathbf{b}}{d\mathbf{x}} = \mathbf{b} \tag{4.3}$$

$$\frac{d\mathbf{x}^T\mathbf{A}\mathbf{x}}{d\mathbf{x}} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x} \tag{4.4}$$

$$\frac{d\mathbf{x}^T\mathbf{A}\mathbf{x}}{d\mathbf{A}} = \mathbf{x}\mathbf{x}^T \tag{4.5}$$

$$\frac{d\,\mathrm{trace}\Big\{\mathbf{X}^T\mathbf{A}\Big\}}{d\mathbf{X}} = \mathbf{A} \tag{4.6}$$

$$\frac{d\,\mathrm{trace}\Big\{\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}\Big\}}{d\mathbf{X}} = \mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{A}^T\mathbf{X}\mathbf{B}^T \tag{4.7}$$

$$\frac{d\,\mathrm{trace}\{\mathbf{X}^{-1}\mathbf{A}\}}{d\mathbf{X}} \overset{(3.10)}{=} -\Big(\mathbf{X}^{-1}\mathbf{A}\mathbf{X}^{-1}\Big)^T = -\mathbf{X}^{-T}\mathbf{A}^T\mathbf{X}^{-T} \tag{4.8}$$

$$\frac{d\,\det(\mathbf{X})}{d\mathbf{X}} \overset{(3.11)}{=} \det(\mathbf{X})\,\mathbf{X}^{-T} \tag{4.9}$$

while other such gradients can be found in [4], which is a freely available online textbook. The next example illustrates this very method on a more elaborate function.

Example 3. Find the differential of $f(\mathbf{W}) \triangleq |||\mathbf{Y} - \mathbf{X}\mathbf{W}|||_F^2$.

Proof. Notice that our independent variable here is $\mathbf{W}$; all other matrices are considered constant. We start off as follows

$$df(\mathbf{W}) = d\,\mathrm{trace}\Big\{(\mathbf{Y} - \mathbf{X}\mathbf{W})^T(\mathbf{Y} - \mathbf{X}\mathbf{W})\Big\} \overset{(3.4)}{=} \mathrm{trace}\Big\{\underbrace{d(\mathbf{Y}^T\mathbf{Y})}_{=\mathbf{O}}\Big\} - \mathrm{trace}\Big\{(d\mathbf{W}^T)\mathbf{X}^T\mathbf{Y}\Big\} -$$
$$- \mathrm{trace}\Big\{\underbrace{\mathbf{Y}^T\mathbf{X}d\mathbf{W}}_{\overset{\text{i.t.}}{=}(d\mathbf{W}^T)\mathbf{X}^T\mathbf{Y}}\Big\} + \mathrm{trace}\Big\{(d\mathbf{W}^T)\mathbf{X}^T\mathbf{X}\mathbf{W} + \underbrace{\mathbf{W}^T\mathbf{X}^T\mathbf{X}d\mathbf{W}}_{\overset{\text{i.t.}}{=}(d\mathbf{W}^T)\mathbf{X}^T\mathbf{Y}}\Big\} =$$
$$= \mathrm{trace}\Big\{(d\mathbf{W}^T)\underbrace{2\mathbf{X}^T(\mathbf{X}\mathbf{W} - \mathbf{Y})}_{\mathbf{G}(\mathbf{W})\overset{(4.1)}{\equiv}}\Big\}$$

Therefore, the required gradient matrix is

$$\frac{d\|\mathbf{Y}-\mathbf{X}\mathbf{W}\|_F^2}{d\mathbf{W}} = 2\mathbf{X}^T(\mathbf{X}\mathbf{W}-\mathbf{Y}) \tag{4.10}$$

In the previous derivation, the main idea was to use the differential rules of Eq. (3.2) to Eq. (3.9) as well as all available trace properties of Eq. (1.15) through Eq. (1.17), to bring the expression inside the trace to the suitable form, which will allow us to identify the gradient matrix.

□

4.2 Jacobian Matrix Identification Rule

Now, we turn to the first-order derivative of vector-valued functions, namely Jacobian matrices.

Theorem 4 (Jacobian Matrix Identification Rule)**.** *Let* $\mathbf{f}:\mathbb{R}^N \to \mathbb{R}^M$ *be differentiable everywhere. Then,*

$$d\mathbf{f}(\mathbf{x}) = \mathbf{J}(\mathbf{x})d\mathbf{x} \quad \Leftrightarrow \quad \frac{d\mathbf{f}(\mathbf{x})}{d\mathbf{x}^T} \equiv \mathbf{J}(\mathbf{x}) \tag{4.11}$$

What follows are a couple of simple examples that apply the previous result to find Jacobian matrices.

Example 4 (Jacobian Matrix of Linear Affine Functions)**.** Let $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}+\mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are constant quantities. We want to compute the Jacobian matrix of $\mathbf{f}$ (w.r.t. $\mathbf{x}$).

Proof. By taking the differential of $\mathbf{f}$ we get

$$d\mathbf{f}(\mathbf{x}) = d(\mathbf{A}\mathbf{x}+\mathbf{b}) = d(\mathbf{A}\mathbf{x}) + d\mathbf{b} = \mathbf{A}d\mathbf{x}.$$

Thus, by Theorem 4, we get that

$$\frac{d}{d\mathbf{x}^T}(\mathbf{A}\mathbf{x}+\mathbf{b}) = \mathbf{A} \tag{4.12}$$

□

Example 5 (Jacobian Matrix of Component-Wise Applied Functions)**.** Compute the Jacobian matrix of the *component-wise* mapping $\mathbf{g}:\mathbf{a}\in\mathbb{R}^D \mapsto \mathbf{g}(\mathbf{a})\in\mathbb{R}^D$, where $g_i(\mathbf{a}) \triangleq g(a_i)$ for all $i=1,\ldots,D$ and where g is a scalar function with derivative g'.

Proof. For all $i = 1, \ldots, D$, we have that $dg_i(a_i) = g'(a_i)da_i$. Arranging these values in a vector, we obtain

$$d\mathbf{g}(\mathbf{a}) = \begin{bmatrix} g'(a_1)da_1 \\ g'(a_2)da_2 \\ \vdots \end{bmatrix} = \mathbf{g}'(\mathbf{a}) \odot d\mathbf{a} \overset{(1.57)}{=} \operatorname{diag}(\mathbf{g}'(\mathbf{a}))\, d\mathbf{a}$$

and, hence, due to Theorem 4,

$$\frac{d\mathbf{g}(\mathbf{a})}{d\mathbf{a}^T} \equiv \operatorname{diag}(\mathbf{g}'(\mathbf{a})) \tag{4.13}$$

where $\mathbf{g}'$ is the function that applies g' component-wise on its argument.

□

Moreover, with the help of Theorem 4, one can also compute the differential of a gradient vector of a scalar function f of a vector $\mathbf{x}$ as follows:

$$\begin{aligned} & d\left(\frac{df(\mathbf{x})}{d\mathbf{x}}\right) \overset{(4.11)}{=} \left[\frac{d}{d\mathbf{x}^T}\left(\frac{df(\mathbf{x})}{d\mathbf{x}}\right)\right] d\mathbf{x} = \frac{d^2 f(\mathbf{x})}{d\mathbf{x} d\mathbf{x}^T} d\mathbf{x} = \nabla^2 f(\mathbf{x})\; d\mathbf{x} \quad \Leftrightarrow \\ \Leftrightarrow \quad & d\left(\frac{df(\mathbf{x})}{d\mathbf{x}}\right) = \nabla^2 f(\mathbf{x})\; d\mathbf{x} \end{aligned} \tag{4.14}$$

4.3 Chain Rules

The response of many ML models due to a given input is computed as a composition of functions and some of their parameters may be nested deep inside these compositions. Prime examples of such circumstances are Multi-Layer Perceptrons (MLPs). In order to train these models, one has to compute gradient vectors or matrices of function compositions. This can be a daunting task without some sort of chain rules that are applicable to the multi-variate case. This section presents such rules. The starting point of all these rules are the following chain rules for differentials, which are presented next.

Theorem 5 (Chain Rules for Differentials)**.** *Let f be a scalar-valued function and G a scalar-, vector- or matrix-valued function of a matrix $\mathbf{X}$. Then,*

$$df(G(\mathbf{X})) = \operatorname{trace}\left\{ dG^T \frac{df(G)}{dG} \right\}\Big|_{G=G(\mathbf{X})} \tag{4.15}$$

which, for $G = g$ a scalar function, specializes to

$$df(g(\mathbf{X})) = f'(g(\mathbf{X}))dg(\mathbf{X}) \tag{4.16}$$

and where f' denotes the derivative of f. Moreover, let $\mathbf{f}$ and $\mathbf{g}$ be vector-valued functions of vector-valued arguments. Then,

$$d\mathbf{f}(\mathbf{g}(\mathbf{x})) = \frac{d\mathbf{f}(\mathbf{g})}{d\mathbf{g}^T} d\mathbf{g}\Big|_{\mathbf{g}=\mathbf{g}(\mathbf{x})} \tag{4.17}$$

Direct, simple consequences of Theorem 5 and, in particular, of Eq. (4.16) are the following identities:

$$d\left(\frac{\mathbf{f}}{g}\right) = \frac{(d\mathbf{f})g - \mathbf{f}dg}{g^2} \tag{4.18}$$

$$df^a = af^{a-1}df \tag{4.19}$$

$$de^f = e^f df \tag{4.20}$$

$$d\ln f = \frac{df}{f} \tag{4.21}$$

where we assume that f and g depend on some independent parameters $\mathbf{X}$. Of particular usefulness are the next two multi-variate derivative chain rules, which stem from Theorem 5.

Proposition 7 (Gradient Chain Rules)**.**

$$\frac{df(g(\mathbf{X}))}{d\mathbf{X}} = \frac{df(g)}{dg}\frac{dg}{d\mathbf{X}}\Big|_{g=g(\mathbf{X})} = f'(g(\mathbf{X}))\frac{dg(\mathbf{X})}{d\mathbf{X}} \tag{4.22}$$

$$\frac{df(\mathbf{g}(\mathbf{x}))}{d\mathbf{x}} = \left(\frac{d\mathbf{g}}{d\mathbf{x}^T}\right)^T \frac{df(\mathbf{g})}{d\mathbf{g}}\Big|_{\mathbf{g}=\mathbf{g}(\mathbf{x})} = \left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}^T}\right)^T \frac{df(\mathbf{g})}{d\mathbf{g}}\Big|_{\mathbf{g}=\mathbf{g}(\mathbf{x})} \tag{4.23}$$

Proof. We are only going to prove Eq. (4.22). The proof of Eq. (4.23) follows more or less similar steps, but, in addition, also relies on the Jacobian matrix identification rule of Theorem 4.

By the chain rule of differentials in Theorem 5 and the gradient identification rule of Theorem 3, we have:

$$df(g(\mathbf{X})) \overset{(4.16)}{=} f'(g(\mathbf{X}))\underbrace{dg(\mathbf{X})}_{} = \operatorname{trace}\left\{(d\mathbf{X})^T\left[f'(g(\mathbf{X}))\frac{dg(\mathbf{X})}{d\mathbf{X}}\right]\right\} \overset{(4.1)}{\Rightarrow}$$

$$\overset{(4.1)}{=} \operatorname{trace}\left\{(d\mathbf{X})^T\frac{dg(\mathbf{X})}{d\mathbf{X}}\right\}$$

$$\Rightarrow \quad \frac{df(g(\mathbf{X}))}{d\mathbf{X}} = f'(g(\mathbf{X}))\frac{dg(\mathbf{X})}{d\mathbf{X}}$$

□

The aforementioned rules are relatively easy to remember, as they seem to obey the following pseudo-behavior:

$$\frac{df(g)}{\cancel{dg}}\frac{\cancel{dg}}{d\mathbf{X}} = \frac{df(g)}{d\mathbf{X}} \quad \text{and} \quad \left(\frac{d\mathbf{g}}{d\mathbf{x}^T}\right)^T \frac{df(\mathbf{g})}{d\mathbf{g}} = \frac{\cancel{d\mathbf{g}^T}}{d\mathbf{x}}\frac{df(\mathbf{g})}{\cancel{d\mathbf{g}}} = \frac{df(\mathbf{g})}{d\mathbf{x}}$$

Additionally, notice that the first matrix in Eq. (4.23) is the transposed Jacobian matrix of $\mathbf{g}$. Next, we present a simple example of how such rules may be applied.

Example 6 (Applying a Gradient Chain Rule)**.** Let $f(g) \triangleq e^g$ and $g(\mathbf{x}) \triangleq \mathbf{x}^T\mathbf{a}$ for some constant vector $\mathbf{a}$. We want to compute the gradient of $f(g(\mathbf{x})) = e^{\mathbf{x}^T\mathbf{a}}$ w.r.t. $\mathbf{x}$.

Proof. We will apply Eq. (4.22) for $\mathbf{X} = \mathbf{x}$. Since $f'(g) = e^g$ and $\frac{dg(\mathbf{x})}{d\mathbf{x}} = \frac{d\mathbf{x}^T\mathbf{a}}{d\mathbf{x}} \overset{(4.3)}{=} \mathbf{a}$, by Eq. (4.22), the desired gradient vector is given as

$$\frac{df(g(\mathbf{x}))}{d\mathbf{x}} = \frac{de^{\mathbf{x}^T\mathbf{a}}}{d\mathbf{x}} = e^{\mathbf{x}^T\mathbf{a}}\,\mathbf{a}$$

□

Another quick example of applying Eq. (4.22) is

$$\frac{d\ln\det(\mathbf{X})}{d\mathbf{X}} = \mathbf{X}^{-T} \tag{4.24}$$

via Eq. (4.9).

The next result is based on Eq. (4.17) of Theorem 5.

Proposition 8 (Jacobian Matrix Chain Rules)**.**

$$\frac{d\mathbf{f}(\mathbf{g}(\mathbf{x}))}{d\mathbf{x}^T} = \frac{d\mathbf{f}(\mathbf{g})}{d\mathbf{g}^T}\frac{d\mathbf{g}}{d\mathbf{x}^T}\bigg|_{\mathbf{g}=\mathbf{g}(\mathbf{x})} = \frac{d\mathbf{f}(\mathbf{g})}{d\mathbf{g}^T}\bigg|_{\mathbf{g}=\mathbf{g}(\mathbf{x})}\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}^T} \tag{4.25}$$

Next, we'll illustrate an example that makes use of the previous proposition.

Example 7 (Jacobian Matrix of Softmax)**.** The *softmax* function $\mathbf{s}$ of a vector $\mathbf{x} \in \mathbb{R}^C$ is defined as $\mathbf{s}(\mathbf{x}) \triangleq \frac{e^{\mathbf{x}}}{\mathbf{1}_C^T e^{\mathbf{x}}}$, where exponentiation is applied component-wise to its argument and $\mathbf{1}_C \in \mathbb{R}^C$ is the all-ones vector. We seek the function's Jacobian matrix. Note that the softmax function and its Jacobian matrix play an important role in training classification models.

Proof. Notice that $\mathbf{s}$ can be written as the composition $\mathbf{s}(\mathbf{x}) = \mathbf{f}(\mathbf{g}(\mathbf{x}))$ of $\mathbf{f}(\mathbf{g}) \triangleq \frac{\mathbf{g}}{\mathbf{1}_C^T\mathbf{g}}$ and $\mathbf{g}(\mathbf{x}) = e^{\mathbf{x}}$. We start off by computing the Jacobian matrix of $\mathbf{f}$ as follows:

$$d\mathbf{f}(\mathbf{g}) = d\left(\frac{\mathbf{g}}{\mathbf{1}_C^T\mathbf{g}}\right) \overset{(4.18)}{=} \frac{1}{\mathbf{1}_C^T\mathbf{g}}\left[\mathbf{I}_C - \underbrace{\frac{\mathbf{g}}{\mathbf{1}_C^T\mathbf{g}}}_{=\mathbf{s}}\mathbf{1}_C^T\right] d\mathbf{g} \quad \overset{(4.11)}{\Rightarrow} \quad \frac{d\mathbf{f}(\mathbf{g})}{d\mathbf{g}^T} = \frac{1}{\mathbf{1}_C^T\mathbf{g}}\left[\mathbf{I}_C - \mathbf{s}\mathbf{1}_C^T\right]$$

where $\mathbf{I}_C$ is the $C \times C$ identity matrix. From Example 5, we readily obtain

$$\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} = \operatorname{diag}(e^{\mathbf{x}}) = \operatorname{diag}(\mathbf{g})$$

where we have suppressed the dependence on $\mathbf{x}$ in our previous notation. Combining these two results with Eq. (4.25), we get

$$\begin{aligned}\frac{d\mathbf{s}(\mathbf{x})}{d\mathbf{x}^T} &= \frac{1}{\mathbf{1}_C^T\mathbf{g}}\left[\mathbf{I}_C - \mathbf{s}\mathbf{1}_C^T\right]\operatorname{diag}(\mathbf{g}) = \frac{\operatorname{diag}(\mathbf{g})}{\mathbf{1}_C^T\mathbf{g}} - \frac{\mathbf{s}}{\mathbf{1}_C^T\mathbf{g}}\mathbf{1}_C^T\operatorname{diag}(\mathbf{g}) = \\ &= \operatorname{diag}\left(\frac{\mathbf{g}}{\mathbf{1}_C^T\mathbf{g}}\right) - \mathbf{s}\underbrace{\mathbf{1}_C^T\operatorname{diag}\left(\frac{\mathbf{g}}{\mathbf{1}_C^T\mathbf{g}}\right)}_{\overset{(1.57)}{=}\left(\frac{\mathbf{g}}{\mathbf{1}_C^T\mathbf{g}}\right)^T} = \operatorname{diag}(\mathbf{s}) - \mathbf{s}\mathbf{s}^T\end{aligned}$$

and, therefore, explicitly showing the dependence on $\mathbf{x}$ again, we have

$$\frac{d\mathbf{s}(\mathbf{x})}{d\mathbf{x}^T} = \operatorname{diag}(\mathbf{s}(\mathbf{x})) - \mathbf{s}(\mathbf{x})\mathbf{s}^T(\mathbf{x})$$

□

4.3.1 At the Heart of Backprop

Assume a scalar-valued, composite function $q(\mathbf{X}) \triangleq \ell(\mathbf{f}_L(\mathbf{f}_{L-1}(\cdots\mathbf{f}_1(\mathbf{X})\cdots)))$, whose gradient matrix w.r.t. $\mathbf{X}$ we want to compute.

With what we have seen so far in this section about chain rules of differentials, we proceed as follows:

$$dq(\mathbf{X}) = d\ell(\mathbf{f}_L) \overset{(4.15)}{=} (d\mathbf{f}_L)^T \frac{d\ell(\mathbf{f}_L)}{d\mathbf{f}_L} \overset{(4.17)}{=} (d\mathbf{f}_{L-1})^T \left(\frac{d\mathbf{f}_L}{d\mathbf{f}_{L-1}^T}\right)^T \frac{d\ell(\mathbf{f}_L)}{d\mathbf{f}_L} \overset{(4.17)}{=} \cdots$$
$$= (d\mathbf{f}_1)^T \underbrace{\left[\prod_{k=2}^{L} \left(\frac{d\mathbf{f}_k}{d\mathbf{f}_{k-1}^T}\right)^T\right] \frac{d\ell(\mathbf{f}_L)}{d\mathbf{f}_L}}_{\equiv \frac{d\ell}{d\mathbf{f}_1}}$$

Eventually, by computing the differential $d\mathbf{f}_1$, we will be able to compute the needed gradient matrix. In general, this may be a messy task. However, if we assume that the differential $d\mathbf{f}_1$ takes the form $d\mathbf{f}_1 = (d\mathbf{X})\mathbf{y}$ for some vector $\mathbf{y}$, which may be a function of $\mathbf{X}$, substitution in the previous expression yields

$$dq(\mathbf{X}) = \mathbf{y}^T (d\mathbf{X})^T \frac{d\ell}{d\mathbf{f}_1} = \text{trace}\left\{(d\mathbf{X})^T \frac{d\ell}{d\mathbf{f}_1}\mathbf{y}^T\right\} \quad \overset{(4.1)}{\Rightarrow} \quad \frac{q(\mathbf{X})}{d\mathbf{X}} = \frac{d\ell}{d\mathbf{f}_1}\mathbf{y}^T$$

This ability of chaining (transposed) Jacobian matrices to compute gradient matrices lies at the core of Error Back-Propagation (EBP), which is used for training MLPs.

4.4 Special Topic: Gradients w.r.t. Symmetric Matrices

As mentioned at the very beginning of this document, the first- and second-order multi-variate derivatives of functions discussed so far are w.r.t. general, unstructured matrices of variables $\mathbf{X}$. It turns out that if $\mathbf{X}$ is symmetric, then the gradient and Hessian matrix identification results we have seen so far do not hold in their stated form and need to be modified. The following example will convince you about this particular need.

Example 8. Compute the gradient matrix of $f(\mathbf{C}) \triangleq \mathbf{a}^T\mathbf{C}\mathbf{b}$, where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^D$ and $\mathbf{C} \in \mathbb{R}^{D\times D}$ is symmetric.

Proof. Using the techniques we have been exposed to so far, as well as trace properties and the fact that $\mathbf{C}$ is symmetric we obtain

$$df(\mathbf{C}) = \text{trace}\left\{(d\mathbf{C})^T\mathbf{b}\mathbf{a}^T\right\} = \tag{4.26}$$
$$= \text{trace}\left\{(d\mathbf{C})^T\mathbf{a}\mathbf{b}^T\right\} \tag{4.27}$$

Then according to Theorem 3 we should have either

$$\frac{df(\mathbf{C})}{d\mathbf{C}} = \mathbf{b}\mathbf{a}^T \tag{4.28}$$

if we base our result on Eq. (4.26), or

$$\frac{df(\mathbf{C})}{d\mathbf{C}} = \mathbf{a}\mathbf{b}^T \tag{4.29}$$

if we base our result on Eq. (4.27) instead. Obviously, if $\mathbf{a} = \mathbf{b}$, then both answers coincide and equal $\mathbf{a}\mathbf{a}^T$, which is a symmetric matrix as expected. Note that, due to $\mathbf{C}$'s symmetry, we expect the gradient matrix to also be symmetric.

However, when $\mathbf{a} \neq \mathbf{b}$, then both results are wrong because neither the first of Eq. (4.28) nor the second one of Eq. (4.29) are symmetric matrices.

□

In this subsection, we'll present an alternative gradient matrix identification theorem that applies to functions of symmetric matrices, since such functions often appear in ML contexts. Whatever follows is based on a discussion about gradient matrices w.r.t. structured matrices that appears in [4, Sec. 2.8]. We'll start off with a definition.

Definition 10 (Special Symmetrization Operator)**.** We define the special symmetrization operator $\text{Ssymm}\{\cdot\}$ from the set of $D \times D$ matrices onto itself as

$$\text{Ssymm}\{\mathbf{A}\} \triangleq \mathbf{A} + \mathbf{A}^T - \mathbf{A} \odot \mathbf{I}_D \tag{4.30}$$

for any square matrix $\mathbf{A} \in \mathbb{R}^{D \times D}$.

Based on this definition, a gradient matrix formula is presented next, that is valid for symmetric matrices of variables.

Theorem 6 (Gradient Matrix of a Scalar Function w.r.t. a Symmetric Matrix)**.** *Let $f : \mathbb{R}^{D \times D} \to \mathbb{R}$ be differentiable and a function of a symmetric argument $\mathbf{C} \in \mathbb{R}^{D \times D}$. Then,*

$$\frac{df(\mathbf{C})}{d\mathbf{C}} = \text{Ssymm}\left\{\frac{df(\mathbf{X})}{d\mathbf{X}}\right\}\Bigg|_{\mathbf{X}=\mathbf{C}} \tag{4.31}$$

where $\mathbf{X} \in \mathbb{R}^{D \times D}$ unstructured matrix.

In other words, if we are interested in the gradient matrix of $f(\mathbf{C})$ w.r.t. $\mathbf{C}$, we first replace $\mathbf{C}$ in f with a general, unstructured matrix $\mathbf{X}$ and then we compute the gradient matrix of $f(\mathbf{X})$ w.r.t.

$\mathbf{X}$. Next, we compute the special symmetrization of this gradient matrix and, finally, we substitute $\mathbf{X}$ with $\mathbf{C}$. For our previous example, this would yield

$$\frac{d\mathbf{a}^T\mathbf{C}\mathbf{b}}{d\mathbf{C}} = \text{Ssymm}\left\{\frac{d\mathbf{a}^T\mathbf{X}\mathbf{b}}{d\mathbf{X}}\right\}\Big|_{\mathbf{X}=\mathbf{C}} = \text{Ssymm}\left\{\mathbf{a}\mathbf{b}^T\right\}\Big|_{\mathbf{X}=\mathbf{C}} = \text{Ssymm}\left\{\mathbf{a}\mathbf{b}^T\right\} = \\ = \mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T - (\mathbf{a}\mathbf{b}^T)\odot\mathbf{I}_D \tag{4.32}$$

Often, we would like to find the stationary points of a function f of a symmetric matrix $\mathbf{A}$. According to what we have discussed so far, we would first replace $\mathbf{A}$ with an unstructured matrix $\mathbf{X}$ in f, then compute its gradient matrix $\dot{\mathbf{F}} \triangleq \frac{df(\mathbf{X})}{d\mathbf{X}}\Big|_{\mathbf{X}=\mathbf{A}}$, and, finally , try to solve the equation $\text{Ssymm}\left\{\dot{\mathbf{F}}\right\} = \mathbf{O}$ for $\mathbf{A}$. A key observation, that can be easily verified, regarding the $\text{Ssymm}\{\cdot\}$ operator is that:

$$\text{Ssymm}\left\{\dot{\mathbf{F}}\right\} = \mathbf{O} \quad \Leftrightarrow \quad \dot{\mathbf{F}}^T = -\dot{\mathbf{F}}^T \tag{4.33}$$

i.e., $\dot{\mathbf{F}}$ must be skew-symmetric. If $\dot{\mathbf{F}}$ is already symmetric, the last statement simplifies to $\dot{\mathbf{F}} = \mathbf{O}$. All this implies the following approach to identifying the stationary points of f:

1. Compute the gradient matrix $\dot{\mathbf{F}} \triangleq \frac{df(\mathbf{A})}{d\mathbf{A}}$ as if $\mathbf{A}$ were an unstructured matrix.
2. Whenever possible, simplify the obtained expression of $\dot{\mathbf{F}}$ using the fact that $\mathbf{A}$ is symmetric.
3. If $\dot{\mathbf{F}}$ is a symmetric gradient matrix, then solve $\dot{\mathbf{F}} = \mathbf{O}$ for $\mathbf{A}$.
4. If $\dot{\mathbf{F}}$ is not a symmetric gradient matrix, then solve $\dot{\mathbf{F}} + \dot{\mathbf{F}}^T = \mathbf{O}$ for $\mathbf{A}$.

The following example will illustrate these points.

Example 9. Let $f(\mathbf{A}) \triangleq \ln\det(\mathbf{A}) - \text{trace}\left\{\mathbf{A}\mathbf{S}^T\right\}$, where $\mathbf{A}^T = \mathbf{A}$ and $\mathbf{S}$ is an invertible square matrix. Find the stationary points of f.

Proof. First, we are going to compute $\dot{\mathbf{F}} \triangleq \frac{df(\mathbf{A})}{d\mathbf{A}}$ as if $\mathbf{A}$ has no special structure. We obtain from Eq. (4.24) and Eq. (4.6) that

$$\dot{\mathbf{F}} = \mathbf{A}^{-T} - \mathbf{S} \quad \overset{\mathbf{A}^T=\mathbf{A}}{=} \quad \mathbf{A}^{-1} - \mathbf{S} \tag{4.34}$$

If $\mathbf{S}$ is symmetric, then $\dot{\mathbf{F}}$ is symmetric as well and the stationary point is given as

$$\dot{\mathbf{F}} = \mathbf{O} \quad \overset{(4.34)}{\Rightarrow} \quad \mathbf{A} = \mathbf{S}^{-1} \tag{4.35}$$

If $\mathbf{S}$ is not symmetric, neither is $\dot{\mathbf{F}}$ and the stationary point is given as

$$\dot{\mathbf{F}} + \dot{\mathbf{F}}^T = \mathbf{O} \quad \overset{(4.34)}{\Rightarrow} \quad \mathbf{A} = \left(\frac{\mathbf{S} + \mathbf{S}^T}{2} \right)^{-1} \tag{4.36}$$

It is worth pointing out that the latter result simplifies to the former one, when $\mathbf{S}$ is symmetric.

□

4.5 Advanced Material: Second-Order Differential & Hessian Identification Rules

The second-order differential $d^2 f$ of a scalar-, vector- or matrix-valued function f is defined as $d^2 f = d(df)$. The following simple rule applies: if $\mathbf{X}$ is the matrix containing independent parameters, then

$$d(d\mathbf{X}) = \mathbf{O} \tag{4.37}$$

i.e., the differential of a differential of independent parameters equals zero. When computing second-order differentials, we simply follow all rules about differentials delineated in Section 3.1. Next, a simple example is provided.

Example 10. Compute the second-order differential of $f(\mathbf{x}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$.

Proof. We begin by computing f's (first-order) differential.

$$df(\mathbf{x}) = d\left(\mathbf{x}^T \mathbf{A} \mathbf{x}\right) = (d\mathbf{x}^T)\mathbf{A}\mathbf{x} + \underbrace{\mathbf{x}^T (d\mathbf{A})\mathbf{x}}_{=0} + \underbrace{\mathbf{x}^T \mathbf{A} d\mathbf{x}}_{\overset{\cdot^T}{=} (d\mathbf{x}^T)\mathbf{A}^T\mathbf{x}} = (d\mathbf{x}^T) \underbrace{\left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{x}}_{\frac{df(\mathbf{x})}{d\mathbf{x}} \overset{(4.2)}{\equiv}} \tag{4.38}$$

where, by the way, we point out the gradient vector of f. Now, taking once again the differential of Eq. (4.38) yields

$$\begin{aligned} (4.38) \overset{d\cdot}{\Rightarrow} \quad d^2 f(\mathbf{x}) &= \underbrace{(d^2\mathbf{x}^T)}_{\overset{(4.37)}{=}\mathbf{0}} \left(\mathbf{A} + \mathbf{A}^T\right)\mathbf{x} + (d\mathbf{x}^T) \underbrace{d\left(\mathbf{A} + \mathbf{A}^T\right)}_{=\mathbf{O}} \mathbf{x} + (d\mathbf{x}^T)\left(\mathbf{A} + \mathbf{A}^T\right) d\mathbf{x} = \\ &= (d\mathbf{x}^T)\left(\mathbf{A} + \mathbf{A}^T\right) d\mathbf{x} \end{aligned} \tag{4.39}$$

□

By doing so and with the help of identification rules described below, one can compute the Hessian matrix of a scalar function.

Theorem 7 (Hessian Identification Rule I for Scalar Functions of a Vector Argument)**.** *Let f be a scalar function of a vector-valued argument that is twice-differentiable. Then,*

$$d^2 f(\mathbf{x}) = d\mathbf{x}^T \mathbf{A}(\mathbf{x})\, d\mathbf{x} \quad \Leftrightarrow \quad \frac{d^2 f(\mathbf{x})}{d\mathbf{x} d\mathbf{x}^T} \equiv \mathbf{A}(\mathbf{x}) + \mathbf{A}^T(\mathbf{x}) \tag{4.40}$$

Notice that the result of the previous theorem yields twice the symmetric part of $\mathbf{A}$, which is a symmetric matrix, and, hence, is consistent with the requirement that the resulting Hessian matrix be symmetric. Based on the previous example and using Theorem 7 we can easily identify the Hessian of f of Example 10 as

$$\nabla^2 f(\mathbf{x}) = \nabla^2 \left(\mathbf{x}^T \mathbf{A} \mathbf{x} \right) = \mathbf{A} + \mathbf{A}^T \tag{4.41}$$

When it comes to scalar functions of a matrix argument, the next theorem helps us identify Hessian matrices.

Theorem 8 (Hessian Identification Rule II for Scalar Functions of a Matrix Argument)**.** *Let f be a scalar function of a matrix-valued argument that is twice-differentiable. If the independent variable is denoted by $\mathbf{X} \in \mathbb{R}^{M \times N}$ and $\nabla^2 \equiv \dfrac{d^2}{d\,\mathrm{vec}\,\mathbf{X} d\,\mathrm{vec}^T \mathbf{X}}$, then,*

$$d^2 f(\mathbf{X}) = \mathrm{trace}\Big\{ (d\mathbf{X}^T) \mathbf{A} (d\mathbf{X}) \mathbf{B} \Big\} \quad \Leftrightarrow \quad \nabla^2 f(\mathbf{X}) \equiv \mathbf{B}^T \otimes \mathbf{A} + \mathbf{B} \otimes \mathbf{A}^T \tag{4.42}$$

$$d^2 f(\mathbf{X}) = \mathrm{trace}\{ (d\mathbf{X}) \mathbf{A} (d\mathbf{X}) \mathbf{B} \} \quad \Leftrightarrow \quad \nabla^2 f(\mathbf{X}) \equiv \mathbf{K}_{N,M} (\mathbf{B}^T \otimes \mathbf{A}) + (\mathbf{B} \otimes \mathbf{A}^T) \mathbf{K}_{M,N} \tag{4.43}$$

where $\mathbf{K}_{M,N}$ is the (M, N) commutation matrix and the dependence of $\mathbf{A}$ and $\mathbf{B}$ on $\mathbf{X}$ has been notationally suppressed.

The result of Theorem 8 simplifies to the one of Theorem 7, when $\mathbf{X}$ becomes a single column vector $\mathbf{X}$. While we are not going to provide a complete proof, we are only going to make a few remarks. Given the veracity of Theorem 7, Eq. (4.42) can be shown by vectorizing $\mathbf{X}$ and then using Eq. (1.51) to finally apply Eq. (4.40), in order to identify the Hessian matrix. Finally, Eq. (4.43) can be obtained by following a similar approach, but using Eq. (1.52) instead of Eq. (1.51).

An immediate result of Eq. (4.42) is that

$$\nabla^2 \,\mathrm{trace}\Big\{ \mathbf{X}^T \mathbf{A} \mathbf{X} \mathbf{B} \Big\} = \mathbf{B}^T \otimes \mathbf{A} + \mathbf{B} \otimes \mathbf{A}^T \tag{4.44}$$

4.6 Advanced Material: Mixed Second-Order Derivatives

In many cases the parameters a multi-variate scalar function depends on are naturally grouped into more than one vectors or matrices, which are referred to as *parameter blocks*. This dependence may take the form of $f(\mathbf{X}_1, \mathbf{X}_2)$, for example, for two blocks $\mathbf{X}_1$ and $\mathbf{X}_2$. When trying to compute the Hessian matrix of f, while in theory we can lump these blocks into a single vector and attempt to use the machinery of the previous section to identify the Hessian matrix, doing so my turn out overly complex and cumbersome. In such cases, computing all mixed second-order derivatives to be discussed in this section and forming the Hessian matrix, block by block, often is a far more easier task. While our discussion below considers functions that depend only on two matrix parameter blocks, the notions and techniques presented can be easily extended to the case of multiple blocks.

First, we define the mixed second-order derivative operator as

$$\frac{d^2}{d\,\mathrm{vec}\,\mathbf{X}_1 d\,\mathrm{vec}^T\,\mathbf{X}_2} \triangleq \frac{d}{d\,\mathrm{vec}\,\mathbf{X}_1}\left(\frac{d}{d\,\mathrm{vec}\,\mathbf{X}_2}\right)^T \tag{4.45}$$

that has the property:

$$\left(\frac{d^2}{d\,\mathrm{vec}\,\mathbf{X}_1 d\,\mathrm{vec}^T\,\mathbf{X}_2}\right)^T = \frac{d^2}{d\,\mathrm{vec}\,\mathbf{X}_2 d\,\mathrm{vec}^T\,\mathbf{X}_1} \tag{4.46}$$

For notational simplicity, we will often denote the operator of Eq. (4.45) simply as $\nabla_{\mathbf{X}_1}\nabla^T_{\mathbf{X}_2}$. Then, Eq. (4.46) implies that $(\nabla_{\mathbf{X}_1}\nabla^T_{\mathbf{X}_2})^T = \nabla_{\mathbf{X}_2}\nabla^T_{\mathbf{X}_1}$.

Note that, if $\mathbf{X}_1$ coincides with $\mathbf{X}_2$, *i.e.*, $\mathbf{X}_1 = \mathbf{X}_2 = \mathbf{X}$ then this operator just becomes the Hessian operator w.r.t. $\mathbf{X}$.

Based on these definitions, if the vector $\mathbf{x} \triangleq [\mathrm{vec}^T\,\mathbf{X}_1 \quad \mathrm{vec}^T\,\mathbf{X}_2]^T$ lumps all parameters into a single column vector, then the Hessian operator w.r.t. to $\mathbf{x}$ is given as

$$\nabla^2_{\mathbf{x}} = \begin{bmatrix} \nabla^2_{\mathbf{X}_1} & \nabla_{\mathbf{X}_1}\nabla^T_{\mathbf{X}_2} \\ \nabla_{\mathbf{X}_2}\nabla^T_{\mathbf{X}_1} & \nabla^2_{\mathbf{X}_2} \end{bmatrix} \tag{4.47}$$

and the Hessian matrix of $f(\mathbf{x}) = f(\mathbf{X}_1, \mathbf{X}_2)$ is given as

$$\nabla^2_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \nabla^2_{\mathbf{X}_1} f(\mathbf{X}_1, \mathbf{X}_2) & \nabla_{\mathbf{X}_1}\nabla^T_{\mathbf{X}_2} f(\mathbf{X}_1, \mathbf{X}_2) \\ \nabla_{\mathbf{X}_2}\nabla^T_{\mathbf{X}_1} f(\mathbf{X}_1, \mathbf{X}_2) & \nabla^2_{\mathbf{X}_2} f(\mathbf{X}_1, \mathbf{X}_2) \end{bmatrix} \tag{4.48}$$

Since this Hessian matrix must be symmetric, we conclude that the Hessian blocks on the diagonal, $\nabla^2_{\mathbf{X}_1} f(\mathbf{X}_1, \mathbf{X}_2)$ and $\nabla^2_{\mathbf{X}_2} f(\mathbf{X}_1, \mathbf{X}_2)$ must be symmetric matrices, and the off-diagonal blocks must be the transposes of each other, *i.e.*, $\left[\nabla_{\mathbf{X}_2}\nabla^T_{\mathbf{X}_1} f(\mathbf{X}_1, \mathbf{X}_2)\right]^T = \nabla_{\mathbf{X}_1}\nabla^T_{\mathbf{X}_2} f(\mathbf{X}_1, \mathbf{X}_2)$.

As mentioned earlier, computing $\nabla^2_{\mathbf{x}} f(\mathbf{x})$ as a wholesome quantity might be much more difficult than computing each one of the Hessian blocks separately (note that, due to the aforementioned symmetry, we only need to compute 3 distinct blocks for Eq. (4.48)) and then, finally, constructing the Hessian matrix.

Given all this, the question becomes now how to compute these blocks. In order to address it, we introduce the notion of *partial differential* of a (in general, matrix-valued) function.

Definition 11 (Partial Differential)**.** Assume a function f that depends on L block parameters $\{\mathbf{X}_\ell\}_{\ell=1}^L$. Then, the partial differential $d_k f(\mathbf{X}_1, \ldots, \mathbf{X}_L)$ of f w.r.t. the k^{th} block $\mathbf{X}_k$ is computed as the differential of f by regarding all other blocks as constant quantities, *i.e.*, independent of $\mathbf{X}_k$.

Using the concept of a partial differential, the following identification rule can be shown.

Theorem 9 (Mixed Derivative Identification Rule for Scalar Functions)**.** *Let a scalar function f depend on L block parameters $\{\mathbf{X}_i\}_{i=1}^L$ and let $\mathbf{x} \triangleq [\text{vec}^T \mathbf{X}_1 \cdots \text{vec}^T \mathbf{X}_L]^T$, so that $f(\mathbf{x})$ stands for $f(\mathbf{X}_1, \ldots \mathbf{X}_L)$. If d_k and d_ℓ are the partial differential operators w.r.t. $\mathbf{X}_k$ and $\mathbf{X}_\ell$ respectively, then*

$$\begin{aligned} & d_\ell d_k f(\mathbf{x}) = d_k d_\ell f(\mathbf{x}) = \text{trace}\Big\{(d_k \mathbf{X}_k)^T \mathbf{B}\ d_\ell \mathbf{X}_\ell\Big\} \quad \Leftrightarrow \\ \Leftrightarrow \quad & \nabla_{\mathbf{X}_k} \nabla^T_{\mathbf{X}_\ell} f(\mathbf{x}) \equiv \begin{cases} \mathbf{I}_{N_\ell} \otimes \mathbf{B} & k \neq \ell \\ \mathbf{I}_{N_\ell} \otimes (\mathbf{B} + \mathbf{B}^T) & k = l \end{cases} \end{aligned} \tag{4.49}$$

where $\mathbf{B}$ is the (k, ℓ) block of the Hessian matrix $\nabla^2_{\mathbf{x}} f(\mathbf{x})$. If each block is a column vector, i.e., $\mathbf{X}_i = \mathbf{x}_i$, then Eq. (4.49) simplifies to

$$d_\ell d_k f(\mathbf{x}) = d_k d_\ell f(\mathbf{x}) = (d_k \mathbf{x}_k)^T \mathbf{B}\ d_\ell \mathbf{x}_\ell \quad \Leftrightarrow \quad \nabla_{\mathbf{x}_k} \nabla^T_{\mathbf{x}_\ell} f(\mathbf{x}) \equiv \begin{cases} \mathbf{B} & k \neq \ell \\ \mathbf{B} + \mathbf{B}^T & k = l \end{cases} \tag{4.50}$$

Notice that, when f depends only one block $L = 1$, then the results of Theorem 9 become identical to the corresponding results of Theorem 7 and Theorem 8. Also notice that, when $k = l$, we obtain the Hessian block matrix $\nabla^2_k f(\mathbf{x})$ by taking twice the symmetric parts of the results for $k \neq \ell$ in Eq. (4.49) and Eq. (4.50).

A simple example will illustrate the use of the previous theorem.

Example 11. Consider the function $f(\mathbf{x}) \triangleq \frac{1}{2}\mathbf{x}^T \mathbf{A} \mathbf{x}$, where $\mathbf{A}^T = \mathbf{A}$. Let's say that $\mathbf{x}$ is split (artificially, we have to admit) into two blocks as $\mathbf{x} = [\mathbf{x}_1^T \quad \mathbf{x}_2^T]^T$. Derive f's Hessian matrix by computations of block mixed-derivatives.

Proof. Let us first express f in terms of the block variables.

$$f(\mathbf{x}) = f(\mathbf{x}_1, \mathbf{x}_2) = \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}^T}_{\mathbf{x}^T \equiv} \underbrace{\begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}}_{\mathbf{A} \equiv} \underbrace{\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}}_{\mathbf{x} \equiv} \tag{4.51}$$

Notice that, since $\mathbf{A}$ is symmetric, we necessarily have that $\mathbf{A}_{1,1}$ and $\mathbf{A}_{2,2}$ are symmetric and $\mathbf{A}_{2,1}^T = \mathbf{A}_{1,2}$. Performing the block operations shown in Eq. (4.51) we get

$$\begin{aligned} f(\mathbf{x}_1, \mathbf{x}_2) &= \frac{1}{2}\mathbf{x}_1^T\mathbf{A}_{1,1}\mathbf{x} + \frac{1}{2}\mathbf{x}_1^T\mathbf{A}_{1,2}\mathbf{x}_2 + \frac{1}{2}\underbrace{\mathbf{x}_2^T\mathbf{A}_{2,1}\mathbf{x}_1}_{\overset{T}{=}\mathbf{x}_1^T\mathbf{A}_{2,1}^T\mathbf{x}_2 = \mathbf{x}_1^T\mathbf{A}_{1,2}\mathbf{x}_2} + \frac{1}{2}\mathbf{x}_2^T\mathbf{A}_{2,2}\mathbf{x} = \\ &= \frac{1}{2}\mathbf{x}_1^T\mathbf{A}_{1,1}\mathbf{x}_2 + \mathbf{x}_1^T\mathbf{A}_{1,2}\mathbf{x}_2 + \frac{1}{2}\mathbf{x}_2^T\mathbf{A}_{2,2}\mathbf{x}_2 \end{aligned} \tag{4.52}$$

Taking first the partial differential d_1 of f in Eq. (4.52) yields

$$\begin{aligned} (4.52) \overset{d_1}{\Rightarrow} \quad d_1 f(\mathbf{x}_1, \mathbf{x}_2) &= (d_1\mathbf{x}_1)^T\mathbf{A}_{1,1}\mathbf{x}_1 + (d_1\mathbf{x}_1)^T\mathbf{A}_{1,2}\mathbf{x}_2 + \frac{1}{2}\underbrace{d_1\left(\mathbf{x}_2^T\mathbf{A}_{2,2}\mathbf{x}_2\right)}_{=0} = \\ &= (d_1\mathbf{x}_1)^T\mathbf{A}_{1,1}\mathbf{x}_1 + (d_1\mathbf{x}_1)^T\mathbf{A}_{1,2}\mathbf{x}_2 \end{aligned} \tag{4.53}$$

Now, taking the partial differential d_2 of $d_1 f$ in Eq. (4.53) gives

$$(4.53) \overset{d_2}{\Rightarrow} \quad d_2 d_1 f(\mathbf{x}_1, \mathbf{x}_2) = \underbrace{d_2\left[(d_1\mathbf{x}_1)^T\mathbf{A}_{1,1}\mathbf{x}_1\right]}_{=0} + (d_1\mathbf{x}_1)^T\underbrace{\mathbf{A}_{1,2}}_{\nabla_{\mathbf{x}_1}\nabla_{\mathbf{x}_2}^T f(\mathbf{x}_1,\mathbf{x}_2) \overset{(4.50)}{=}} d_2\mathbf{x}_2 \tag{4.54}$$

Notice that, had we first taken the partial differential d_2 of f and then the partial differential d_1 of $d_2 f$ (in other words, if we had switched the order of the steps we took), we would have obtained the very same result as what we got in Eq. (4.54).

In a similar fashion, by considering the differentials $d_1 d_1 \equiv d_1^2$ and $d_2 d_2 \equiv d_2^2$ of f, we can compute the blocks on the Hessian's diagonal $\nabla_{\mathbf{x}_1}\nabla_{\mathbf{x}_1}^T f(\mathbf{x}_1, \mathbf{x}_2)$ and $\nabla_{\mathbf{x}_1}\nabla_{\mathbf{x}_1}^T f(\mathbf{x}_1, \mathbf{x}_2)$, which equal $\mathbf{A}_{1,1}$ and $\mathbf{A}_{2,2}$ respectively. As a reminder, $\nabla_{\mathbf{x}_2}\nabla_{\mathbf{x}_1}^T f(\mathbf{x}_1, \mathbf{x}_2) = \left[\nabla_{\mathbf{x}_1}\nabla_{\mathbf{x}_2}^T f(\mathbf{x}_1, \mathbf{x}_2)\right]^T = \mathbf{A}_{1,2}^T = \mathbf{A}_{2,1}$. Hence, the Hessian matrix $\nabla_{\mathbf{x}}^2 f(\mathbf{x})$ is put together as

$$\nabla_{\mathbf{x}}^2 f(\mathbf{x}) = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{1,2}^T & \mathbf{A}_{2,2} \end{bmatrix} = \mathbf{A} \tag{4.55}$$

which is the same result we would have obtained, if we computed f's Hessian w.r.t. $\mathbf{x}$ directly.

□

Finally, for completeness, based on Theorem 9 and using Eq. (1.31), Eq. (1.33) and Eq. (1.50b), we can show a more general identification theorem for these mixed derivatives.

Theorem 10 (Mixed Derivative Identification Rule for Scalar Functions – General Case)**.** *Let a scalar function f depend on L block parameters $\{\mathbf{X}_i\}_{i=1}^{L}$ and let $\mathbf{x} \triangleq [\mathrm{vec}^T \mathbf{X}_1 \cdots \mathrm{vec}^T \mathbf{X}_L]^T$, so that $f(\mathbf{x})$ stands for $f(\mathbf{X}_1, \ldots \mathbf{X}_L)$. If d_k and d_ℓ are the partial differential operators w.r.t. $\mathbf{X}_k \in \mathbb{R}^{M_k \times N_k}$ and $\mathbf{X}_\ell$ respectively, then*

$$\begin{aligned} & d_\ell d_k f(\mathbf{x}) = d_k d_\ell f(\mathbf{x}) = \mathrm{trace}\Big\{(d_k \mathbf{X}_k)^T \mathbf{A}\ (d_\ell \mathbf{X}_\ell)\mathbf{B}\Big\} \quad \Leftrightarrow \\ \Leftrightarrow \quad & \nabla_{\mathbf{X}_k} \nabla^T_{\mathbf{X}_\ell} f(\mathbf{x}) \equiv \begin{cases} \mathbf{B}^T \otimes \mathbf{A} & k \neq \ell \\ \mathbf{B}^T \otimes \mathbf{A} + \mathbf{B} \otimes \mathbf{A}^T & k = \ell \end{cases} \end{aligned} \tag{4.56}$$

and

$$\begin{aligned} & d_\ell d_k f(\mathbf{x}) = d_k d_\ell f(\mathbf{x}) = \mathrm{trace}\{(d_k \mathbf{X}_k)\mathbf{A}\ (d_\ell \mathbf{X}_\ell)\mathbf{B}\} \quad \Leftrightarrow \\ \Leftrightarrow \quad & \nabla_{\mathbf{X}_k} \nabla^T_{\mathbf{X}_\ell} f(\mathbf{x}) \equiv \begin{cases} \mathbf{K}_{N_k,M_k}(\mathbf{B}^T \otimes \mathbf{A}) & k \neq \ell \\ \mathbf{K}_{N_k,M_k}(\mathbf{B}^T \otimes \mathbf{A}) + (\mathbf{B} \otimes \mathbf{A}^T)\mathbf{K}_{M_k,N_k} & k = \ell \end{cases} \end{aligned} \tag{4.57}$$

where $\mathbf{K}_{N_k,M_k}$ is the (N_k, M_k) commutation matrix and $\nabla_{\mathbf{X}_k} \nabla^T_{\mathbf{X}_\ell} f(\mathbf{x})$ is the (k, ℓ) block of the Hessian matrix $\nabla^2_{\mathbf{x}} f(\mathbf{x})$.

Once again, notice that, when $k = l$, we obtain the Hessian block matrix $\nabla^2_k f(\mathbf{x})$ by taking twice the symmetric parts of the results for $k \neq \ell$ in Eq. (4.56) and Eq. (4.57).

4.7 Questions To Consider

1. If $\mathbf{a}, \mathbf{b} \in \mathbb{R}^D$ and $\mathbf{C} \in \mathbb{R}^{D \times D}$ are constants, while λ is an independent parameter, compute the differential of $\mathbf{a}^T(\mathbf{C} + \lambda \mathbf{I}_D)^{-1}\mathbf{b}$.
2. Show that $\frac{d \ln \det(\mathbf{X})}{d\mathbf{X}} = \mathbf{X}^{-1}$
3. Show Eq. (4.3) through Eq. (4.8).
4. Derive the Hessian of $f(\mathbf{x}) \triangleq \mathrm{trace}\Big\{\mathbf{X}^T \mathbf{Y}^{-1} \mathbf{X}\Big\}$, where $\mathbf{x} \triangleq [\mathrm{vec}^T \mathbf{X} \quad \mathrm{vec}^T \mathbf{Y}]^T$ and $\mathbf{Y}^T = \mathbf{Y}$.

ECE5268: Theory of Neural Networks **Spring 2020**

Lecture 5: Linear Regression

Thu, Jan 30, 2020

Lecturer: GCA *Scribe(s): AMS*

Note: *This header style is courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

This lecture discusses an overview of linear regression. Important concepts in linear regression are covered such as an introduction to regression, loss functions, training a linear regression model and gradient vectors.

5.1 Introduction to Regression

The regression task: given a training set of i.i.d. input/output (I/O) samples $\{(\mathbf{x}_n \in \mathbb{R}^D, y_n)\}_{M=1}^N$, we want to learn a function/map.

$$f : \mathbb{R}^D \Rightarrow \mathbb{R} \quad s.t. \quad \hat{y}_n \triangleq f(\mathbf{x}_n) \approx y_n \tag{5.1}$$

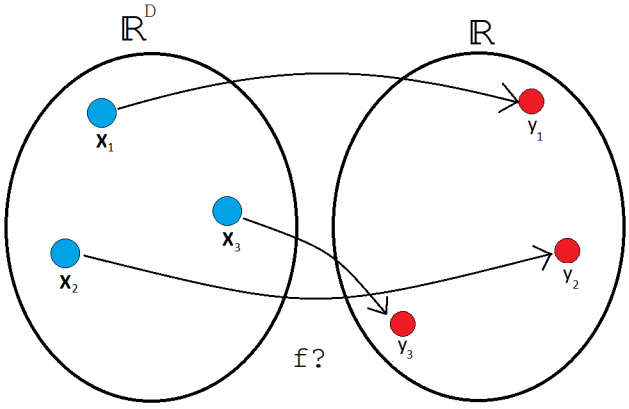

Figure 1: Regression is used to obtain a map between f($\mathbf{x_n}$) and y_n

5.2 Linear Regression

Suppose that the true output is equal to the predicted output/target for input $\mathbf{x}_n$ plus some error. Mathematically, hypothesize that $y_n = \hat{y}_n + \epsilon_n$ and $\hat{y} = \mathbf{W}^T\mathbf{x} + b$ where $\hat{y}$ is a model's I/O equation, and $\mathbf{W} \in \mathbb{R}^D$, $b \in \mathbb{R}$.

For $D-1$, $\hat{y} = wx + b$ For $D \geq 2$, the graph $\hat{y}$ as a function of $\mathbf{x}$ is a hyper-plane.

Remark:

$$\hat{y} = \mathbf{W}^T\mathbf{x} + b = \underbrace{\begin{bmatrix}\mathbf{W}\\ b\end{bmatrix}^T}_{\tilde{\mathbf{W}}^{\triangleq}} \underbrace{\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix}}_{\tilde{\mathbf{x}}^{\triangleq}} = \underbrace{\begin{bmatrix}\mathbf{W}^T & b\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix}}_{\tilde{\mathbf{W}}^T\tilde{\mathbf{x}}} \tag{5.2}$$

Note:

$$\begin{bmatrix}\mathbf{x}\\ 1\end{bmatrix} \tag{5.3}$$

is known as the augmented input vector.

In order to appraise the *closeness* of $\hat{y}$ to y_n, we will use a loss function. The most popular for regression is the squared loss function:

$$\ell(y, \hat{y}) = (y - \hat{y})^2 \tag{5.4}$$

Since we have N training samples, we will to use the squared loss averaged over the entire training set as a measure of how well the $\hat{y}(\mathbf{x})$ fits the given y values.

Mean Squared Error (MSE as computed on the training set) is defined as

$$MSE(\hat{y}(\tilde{\mathbf{x}})) = \underbrace{\frac{1}{N}\Sigma\ell(y_n, \hat{y}_n)}_{\text{where } \hat{y}_n \triangleq \hat{y}(\tilde{\mathbf{x}}_n)} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \tilde{\mathbf{w}}^T\tilde{\mathbf{x}}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\epsilon_n^2 \tag{5.5}$$

In our case finding the best $\hat{y}(\tilde{\mathbf{x}})$ results in finding the "best" $\tilde{\mathbf{w}}$, i.e. $\tilde{\mathbf{w}}^* \triangleq \arg\min MSE_{train}(\tilde{\mathbf{w}})$. Training our linear regression model amounts to finding $\tilde{\mathbf{w}}^*$.

$$MSE_{train}(\tilde{\mathbf{w}}) = \frac{1}{N}\left\| \underbrace{\begin{bmatrix}y_1\\ \vdots\\ y_N\end{bmatrix}^T}_{\mathbf{y}^{\triangleq}} - \begin{bmatrix}\tilde{\mathbf{x}}_\mathbf{1}^\mathbf{T}\tilde{\mathbf{w}}\\ \vdots\\ \tilde{\mathbf{x}}_\mathbf{N}^\mathbf{T}\tilde{\mathbf{w}}\end{bmatrix} \right\|_2^2 \tag{5.6}$$

$$\begin{bmatrix}\tilde{\mathbf{x}}_\mathbf{1}^\mathbf{T}\tilde{\mathbf{w}}\\ \vdots\\ \tilde{\mathbf{x}}_\mathbf{N}^\mathbf{T}\tilde{\mathbf{w}}\end{bmatrix} = \underbrace{\begin{bmatrix}\tilde{\mathbf{x}}_\mathbf{1}^\mathbf{T}\\ \vdots\\ \tilde{\mathbf{x}}_\mathbf{N}^\mathbf{T}\end{bmatrix}}_{X^{\triangleq}} \tilde{\mathbf{w}} \tag{5.7}$$

Remark: The "design matrix" is $Nx(D+1)$ where the last column is 1_N.

Hence, $MSE_{train}(\tilde{\mathbf{w}}) = \frac{1}{N}\|\mathbf{y} - X\tilde{\mathbf{w}}\|_2^2 = \frac{1}{N}(y - X\tilde{\mathbf{w}}) = \frac{1}{N}(\tilde{\mathbf{w}}^T X^T X\tilde{\mathbf{w}} - 2\tilde{\mathbf{w}}^T X^T\mathbf{y} + \mathbf{y}^T\mathbf{y})$.

5.2.1 Gradient Vectors

The gradient vector, sometimes denoted as $\nabla\mathbf{f}(x_0, y_0)$, is perpendicular to the level curve $f(x, y) = k$ at the point (x_0, y_0). This projects similarly as more dimensions are added.

$$\frac{\partial MSE(\tilde{\mathbf{w}})}{\partial\tilde{\mathbf{w}}} = \frac{1}{N}(\underbrace{\frac{\partial\tilde{\mathbf{w}}^T X^T X\tilde{\mathbf{w}}}{\partial\tilde{\mathbf{w}}}}_{-2X^T X\tilde{\mathbf{w}}} - 2\underbrace{\frac{\partial\tilde{\mathbf{w}}^T X^T\mathbf{y}}{\partial\tilde{\mathbf{w}}}}_{X^T\mathbf{y}} + \cancel{\frac{\partial\|\mathbf{y}\|_2^2}{\partial\tilde{\mathbf{w}}}}) \tag{5.8}$$

and

$$\frac{\partial MSE(\mathbf{w})}{\partial\mathbf{w}} = \frac{2}{N}X^T(X\tilde{\mathbf{w}} - y) \tag{5.9}$$

where finding $\tilde{\mathbf{w}}$ that results in 0 produces the stationary points of a gradient vector. Equation Eq. (5.9) can be simplified to more easily solve for stationary points:

$$X^T X\tilde{\mathbf{w}} = X^T\mathbf{y} \quad \Rightarrow \quad \begin{cases} \exists(X^T X)^{-1}, & \tilde{\mathbf{w}}^* = (X^T X)^{-1}X^T\mathbf{y} \\ o/w, & \text{the solution is much more complicated..} \end{cases} \tag{5.10}$$

Note:

$$\frac{\partial^2 MSE(\tilde{\mathbf{w}})}{\partial\tilde{\mathbf{w}}\partial\tilde{\mathbf{w}}^T} = 2X^X \succcurlyeq 0 \tag{5.11}$$

thus, the MSE is convex in $\tilde{\mathbf{w}}$, since $\text{MSE}(\tilde{\mathbf{w}}) \geq 0 \quad \Rightarrow \quad$ any stationary point of $\text{MSE}(\tilde{\mathbf{w}})$ is a minimum.

Remark: The matrix $(X^T X)^{-1}X^T$, when the inverse exists, is called the Moore-Penrose pseudo-inverse $X^\dagger$ of X.

The minimum $\tilde{\mathbf{w}}$ for $\text{MSE}_{\text{train}}(\mathbf{w}) = \text{MSE}_{\text{train}}(\hat{\mathbf{w}})$ and can be solved for using the following

$$
\begin{aligned}
min(\tilde{\mathbf{w}}) \text{ for } MSE_{train}(\tilde{\mathbf{w}}) &= MSE_{train}(\tilde{\mathbf{w}}^*) \\
&= \frac{1}{N} \left\| \mathbf{y} - X(X^T X)^{-1} X^T \mathbf{y} \right\|_2^2 \\
&= \frac{1}{N} \left\| (I_n - XX^\dagger)\mathbf{y} \right\|_2^2 \\
&= \frac{1}{N} \mathbf{y}^T (I_N - XX^\dagger)\mathbf{y} \\
&= \frac{1}{N} \mathbf{y}^T \underbrace{(I_N - XX^\dagger)^T (I_N - XX^\dagger)}_{I_N - XX^\dagger = ?} \mathbf{y} \\
&= \frac{1}{N} \|\mathbf{y}\|_{I_N - XX^\dagger}^2
\end{aligned}
\tag{5.12}
$$

ECE5268: Theory of Neural Networks **Spring 2020**

Lecture 6: Linear Regression, Ridge Regression and Regularization

Thu, Feb 06, 2020

Lecturer: GCA *Scribe(s): LAP*

Note: *This header style is courtesy of UC Berkeley EECS dept.*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

This lecture continues discussing Linear Regression from Lecture 5, and introduces Ridge Regression and Regularization methods.

6.1 Cont'd. With Linear Regression

Point 1: $\tilde{\mathbf{w}}^* = \tilde{\mathbf{X}}^{\mathbf{T}}\mathbf{y}$ (6.1)

where $\tilde{\mathbf{X}}^T = (\tilde{\mathbf{X}}^T\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^T$. The term $\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$ is not invertible if $N \leq D$ (number of samples is less or equal to the number of weights, except for the intercept). In this case, there are more parameters $(D+1)$ than training samples (N).

Thus, $\tilde{\mathbf{w}}^*$ is not unique and $MSE_{training}(\tilde{\mathbf{w}}^*) = 0$. This occurs when the training data are not enough in comparison to the model's complexity.

A clear indication of over-training is when the $MSE_{training}$ is very low and the MSE_{ho} is very large.

MSE_{ho} is a measure of how well a linear regression model generalizes (how accurate its predictions are for samples not used in training).

Non-Linear Feature Transformation:

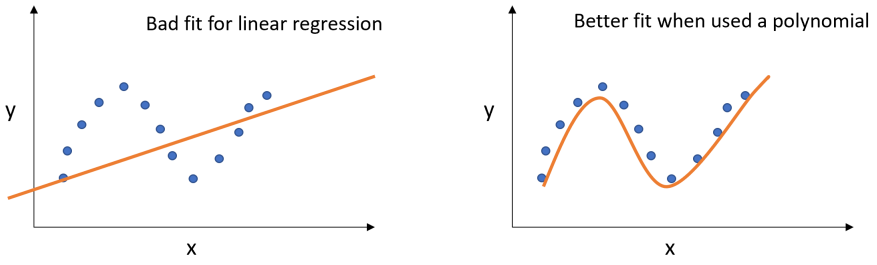

Figure 2: Model fit with a linear and polynomial

We can non-linearly transform features and then use a Linear Model to fit them. The model's response surface can be a hyper-surface other than a hyper-plane.

Example: Polynomial Regression

$$\widehat{y}(x) = w_1 x^2 + w_2 x + b = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix}^T \underbrace{\begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix}}_{\tilde{\varphi}(x)} \tag{6.2}$$

$$\varphi : x \longrightarrow \begin{bmatrix} x^2 \\ x \end{bmatrix} \qquad \text{and} \qquad \tilde{\varphi} : x \longrightarrow \begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix} \tag{6.3}$$

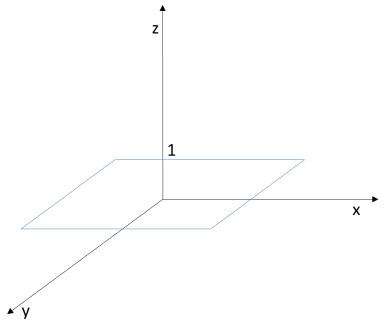

Figure 3: IMAGE NOT COMPLETED

This way we perform Linear Regression in our transformed input space mapped by $\tilde{\varphi}$, the effect in the original feature space is to fit a polynomial curve.

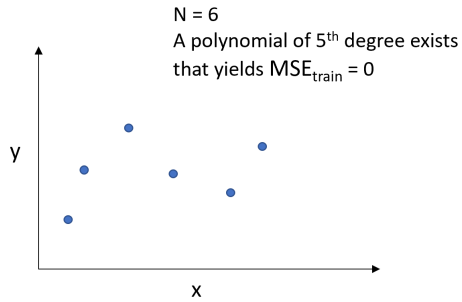

Figure 4: Polynomial of 5^{th} degree that yields $MSE_{train} = 0$

6.2 Ridge Regression

$$|\,\widehat{\mathbf{y}}(x) - \widehat{\mathbf{y}}(x')\,| = |\,\tilde{\mathbf{w}}^T\tilde{\varphi}(x) - \tilde{\mathbf{w}}^T\tilde{\varphi}(x')\,| \leq \|\tilde{\mathbf{w}}\|_2\,\|\tilde{\varphi}(x) - \tilde{\varphi}(x')\|_2 \tag{6.4}$$

In this equation, the greater $\|\tilde{\mathbf{w}}\|_2$, the greater $|\widehat{\mathbf{y}}(x) - \widehat{\mathbf{y}}(x')|$ can be. Thus, by controlling the "size" of $\tilde{\mathbf{w}}$ (as measured by some vector norm of our choosing) we can control $\widehat{y}(x)$'s smoothness.

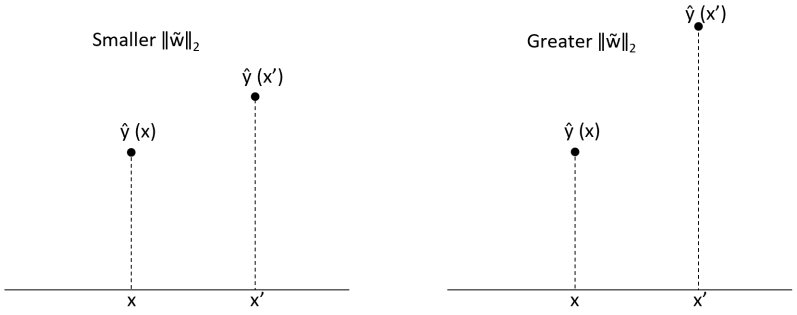

Figure 5: Difference between $\widehat{y}(x)$ and $\widehat{y}(x')$

Notice that $\frac{d\widehat{y}(x)}{d\tilde{\varphi}} = \tilde{\mathbf{w}}$, hence by controlling $\|\tilde{\mathbf{w}}\|$, we can control $\left\|\frac{d\widehat{y}(x)}{d\tilde{\varphi}(x)}\right\|$.

Ridge Regression MSE:

$$MSE_{train}(\tilde{\mathbf{w}}) = \frac{1}{N}\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{w}}\|_2^2 \tag{6.5}$$

Ivanov Regularization:

$$\min\ MSE_{train}(\tilde{\mathbf{w}}) \tag{6.6}$$

where $\tilde{\mathbf{w}} \triangleq \begin{bmatrix}\mathbf{w}\\ b\end{bmatrix}$, and $\mathbf{w} : \|\mathbf{w}\|_2 \leq R$ (radius), for some $R > 0$ (as seen in Figure 6).

Regularization

Definition: Regularization approaches are heuristic methods to prevent a model from over-fitting or over-training. For example, the Ivanov Regularization is a constrained-based method.

It turns out that there is a $\lambda \geq 0$ for every R such that the Ivanov constraint minimization problem can be recast into the following penalized unconstrained minimization problem (Tikhonor regularization)

$$\min_{\tilde{\mathbf{w}}} \left[MSE_{train}(\tilde{\mathbf{w}}) + \lambda \|\mathbf{w}\|_2^2 \right] \tag{6.7}$$

The optimal value of λ (or R, previously) cannot be determined by minimizing MSE_{train}. Hence, λ is referred to as hyper-parameter. Optimal values of hyper parameters are identified through a validation procedure.

Ivanov Regularization Geometric representation:

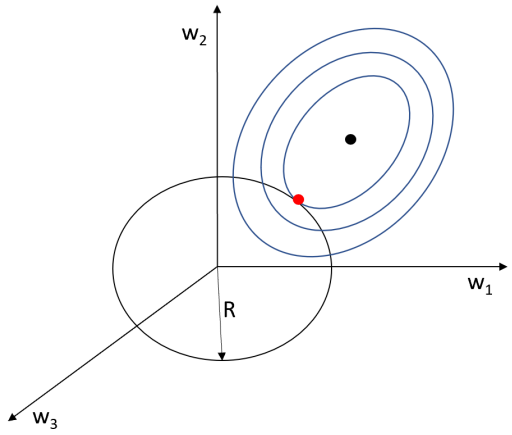

Figure 6: IMAGE NOT FINISHED

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, October 2007.

[3] Jan R. Magnus and Heinz Neudecker. *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 3rd edition, 2007. URL: http://www.janmagnus.nl/misc/mdc2007-3rdedition.

[4] K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. Version 20121115. URL: http://www2.imm.dtu.dk/pubdb/p.php?3274.

[5] Silmaril. An introduction to typesetting with LaTeX. Accessed: 2017-01-10. URL: http://latex.silmaril.ie/formattinginformation/credits.html.

[6] StackExchange. TeX. Accessed: 2017-01-10. URL: http://tex.stackexchange.com/.

[7] Wikibook. Latex. Accessed: 2017-01-10. URL: https://en.wikibooks.org/wiki/LaTeX.

[8] Wikipedia. LaTeX. Accessed: 2017-01-10. URL: https://en.wikipedia.org/wiki/LaTeX.

[9] Wikipedia. Metric (mathematics), Wikipedia article. Accessed: 2017-01-10. URL: https://en.wikipedia.org/wiki/Metric_(mathematics).

[10] Wikipedia. Norm (mathematics), Wikipedia article. Accessed: 2017-01-10. URL: https://en.wikipedia.org/wiki/Norm_(mathematics).

7 Instructions

We will be using LaTeX [8] to record notes for this class. For those of you that are unfamiliar with it, LaTeX is a digital typesetting system that puts an emphasis on aesthetically pleasing visual results and was specifically developed to typeset mathematical expressions in such a way. It was also design to allow an author to mostly focus on the content of a document rather than its particular formatting and typesetting.

A short list of readily-accessible LaTeX resources is given below:

- The LaTeX Wikibook [7].
  This is an excellent reference guide for beginners.
- An introduction to typesetting with LaTeX [5].
- TeX on StackExchange [6].
  Very often, StackExchange is the most reliable source for LaTeX-related solutions.
- Online LaTeXEquation Editor.
  If you are unfamiliar in typesetting mathematical expressions in LaTeX, you will find this editor indispensable. Put together your expression in the editor and, once happy with it, copy and paste it into the lecture notes.

Additionally, a list of books on LaTeX is provided here.

We will be using Overleaf to compile and maintain these lecture notes. Overleaf provides a free (with some restrictions) online environment for collaboratively authoring documents in LaTeX. In specific, you can access and modify the project pertaining to these notes via a access link that I will provide you on Canvas. Please do not share this link with people outside the class, as access to this link will enable them to modify the LaTeX project and even erase it. If you want somebody else to have read-only access to it, you can share with them a read-only link, that Overleaf can generate for you.

If you are an absolute LaTeX beginner, I strongly recommend you to initially work on your edits outside of this OverLeaf project. For example, you can download a copy of this project and upload it to your own OverLeaf account. Once done with your edits, you can copy and paste them into this project.

Alternatively, if you want to have your own LaTeX system installed on your computer to work on LaTeX projects (like this one) offline, you will first need to install a LaTeX distribution, such as MikTeX for PCs. Next, you probably would want to install a development environment / GUI, such as TeXnic Center or TeX Studio, again, for PCs. For managing bibliographies, I highly recommend JabRef, which is cross-platform. In case you have more suggestions for other platforms, such as MacOS, please share them in this paragraph.

7.1 Getting Started

First, add your three-letter name acronym in `sty/acronyms.sty`. Next, add your name (via the acronym) and brief information about yourself in the abstract section of the main file named `ECE5268-Sp20-LectureNotes.tex`.

If you are a scribe responsible for, say, lecture XX, create a file `LectureXX.tex` in the root directory and record the lecture notes in it.

Next, in `ECE5268-Sp20-LectureNotes.tex`, uncomment the appropriate `\include` line, so the contents of `LectureXX.tex` become part of the document.

Finally, add an item entry in `Updates.tex` letting everybody know about the changes you made to this document.

When authoring the lecture notes, keep in mind the following important points:

1. Avoid colloquialisms and ensure technical accuracy.
   Avoid the use of lay man's terms and use technical terms instead. Also, avoid using colloquialisms in your narrative as much as possible. Finally, ensure that, whatever you are presenting, is technically sound. Feel free to consult with me on these issues.

2. Keep your LaTeX markup clean and organized.
   This will greatly help the rest of us, when trying to add materials and/or make corrections. Feel free to add comments (any line that starts with a %) in the source code, like notes and reminders that might be helpful in one capacity or another.

3. Never leave the project in a "bad" state.
   This LaTeX project should always be maintained in a state of no errors and no warnings. If you encounter an error or warning, please ensure that it is fixed/addressed before you finish your edits. Do not rely on others to fix issues that you have introduced into the document. A good practice in order to avoid such problems is to make changes in this document in a very incremental fashion.

Happy TeX-ing!

7.2 Useful LaTeX Rudiments

This section provides some LaTeX basics to get you started along with some conventions that we are going to be using for putting together this document. Despite being terse, I think they will turn out to be useful.

Before starting though, let me emphasize that, especially if you are a LaTeX beginner, it is important for you to inspect the source code already available to you, especially in `Instructions.tex`, in order to learn how to accomplish things in LaTeX.

For your reference, this LaTeX project is organized as follows:

- root (in Overleaf: `files`) directory
  It contains the main TEX file `ECE5268-Sp20-LectureNotes.tex`, which gathers all other source files to produce the document. You will be touching this file only in minor ways as explained in Section 7.1. Also, it includes the individual `LectureXX.tex` files that will be containing the material that you will be authoring. Finally, `References.bib` contains the accompanying bibliography of the document. You will be editing this latter file to include additional references.

- `sty` directory
  First, it contains `packages.sty`, which loads all necessary packages, and `macros.sty`, which defines a variety of macros; you are very unlikely to have to touch these files. Finally, it contains `acronyms.sty`, in which you can define acronyms to be used in the text.

- `figures` directory
  This directory should contain all `JPG`, `PNG`, `EPS` or `PDF` figures used in the manuscript.

You are strongly encouraged to explore all these files and familiarize yourselves with their contents.

7.2.1 General Stuff

For rookies: Unlike MS Word and other similar text editors, LATEX was designed to liberate the author from having to arrange document elements, such as paragraphs, figures, tables, etc., on a page. LATEX takes into consideration of century-old typesetting rules (*e.g.*, how letters should be spaced out on a line of text, where is the best location on a page to place an image, etc.) to do that for you[3]. This means that you should allow LATEX to decide how to arrange a page. Do not sweat it, if, say, a figure does not appear on the precise page that you want.

To make a new paragraph, leave at least one empty line between the preceding text and your new paragraph. Use the `\textbf{}` macro to make text appear in **boldface** and `\textit{}` to *italisize* it. Finally, use `\uline{}` to underline text. Use this last 3 features judiciously[4]. Super- and subscripts in text mode can be accomplished via the `\textsuperscript{}` and `\textsubscript{}` macros. For example, `2\textsuperscript{nd}` would render as 2^{nd}. Finally, if you would like to use an underscore (_), dollar sign ($), percent sign (%) or ampersand (&) within the text, you have to escape the corresponding character with a backslash, *e.g.*, `\$`.

Finally, if you want to hyper-link some text, you can use the `\href{<url>}{<text>}` macro. For example, `\href{https://www.fit.edu}{FIT}` will render the clickable[5] hyperlink FIT. If you need to start a new page (*e.g.*, a paragraph is split between two pages in an "ugly" manner), you can use `\newpage` (in our example, prior to that paragraph) to force LATEX to perform a page break.

While authoring, you may find the following macros (defined in `macros.sty`) useful: `\ie` (rendered as *i.e.*, "id est"), `\eg` (rendered as *e.g.*, "for example"), `\vs` (rendered as *vs.*, "versus"), `\wrt` (rendered as w.r.t., "with respect to"), `\st` (rendered as s.t., "such that") and `\iid` (rendered as

[3] It uses constrained optimization algorithms to do so!
[4] In other words, do not overdo it. By the way, footnotes can be inserted via the `footnote` macro.
[5] I guess, outside of OverLeaf, in the PDF version of the document

i.i.d., "independent and identically distributed"). Finally, notice that, if needed, single quotes in LaTeX are accomplished with the pair '..' (left single quote, right single quote), while double quotes via "..." (twice left single quote, twice right single quote).

With regards to (cross-)referencing within the document, LaTeX offers labeling capabilities via the `\label{}` macro. Any document element, such a section, a figure or a theorem can be labeled and referenced. For example, this section was labeled via `\label{sec:GeneralStuff}` (*i.e.*, its label is `sec:GeneralStuff`) and can be referenced using `\secref{sec:GeneralStuff}`, which renders as Section 7.2.1. While any contiguous string (with no spaces in between) can be used as a label (*e.g.*, we could have named/labeled this section simply as `GeneralStuff`), we will use labels starting with a few characters followed by a colon (:), which indicate what type of element is labeled/referenced. For example, the `sec:` part in `sec:GeneralStuff` tells us that the label is naming a section (or subsection, or sub-subsection, like in our case here). The macro that we will be using to reference an element, will also be dependent on the type element. In specific, we will adopt the following conventions:

- Lectures
  `lec:` labels referenced by `\lecref{}`.
- Sections, subsections, sub-subsections
  `sec:` labels referenced by `\secref{}`.
- Figures
  `fig:` labels referenced by `\figref{}`.
- Tables
  `tab:` labels referenced by `\tabref{}`.
- Algorithms
  `alg:` labels referenced by `\algref{}`.
- Equations
  `eq:` labels referenced by `\eqref{}`.
- Probflems
  `prob:` labels referenced by `\probref{}`.
  This will be used to refer to mathematical expressions that depict an optimization problem or an equation to be solved.
- Definitions
  `def:` labels referenced by `\defref{}`.
- Propositions
  `prop:` labels referenced by `\propref{}`.
- Theorems
  `theo:` labels referenced by `\theoref{}`.

- Examples
  `ex:` labels referenced by `\exref{}`.

All the aforementioned macros have been (re)defined in `macros.sty` and are provided (i) to customize the referencing style of elements depending on their type and (ii) to keep uniformity of referencing throughout this document.

With regards to choosing label names, let us abide by the following conventions: (i) for (sub)sections, let's concatenate their title in CamelCase, (ii) for equations, lets use labels of the form "lecXX_Y", where XX (*e.g.*, 01, 02, ..., 10, ...) is the number of the lecture and Y is an increasing number, (iii) for figures, let us use the filename in CamelCase form as a label.

7.2.2 Math in LaTeX

Typesetting math in LaTeX can be accomplished in two different ways: (i) inline, *i.e.*, within the flowing text. This can be done in several ways, but the preferred way would be by using the dollar sign pair `$ ... $` as in `$\alpha = 1$`, which renders as $\alpha = 1$. (ii) standalone, *i.e.*, as a single equation or a set of them. In LaTeX there are several ways of going about it, but the preferred way would be by using an `align` environment, which uses the & character to align expressions. For example,

```
\begin{align}
x & = 1 \nonumber % \nonumber prevents this expression to be numbered
\\ % starts a new line
y & = 2 \nonumber
\end{align}
```

would render as

$$x = 1$$
$$y = 2$$

We will be heavily using this environment to depict individual equations, as well as derivations. Look at examples in this document for its effective use.

To depict math symbols in boldface (*e.g.*, to denote vectors and matrices), use the `\mathbf{}`. For example, `\mathbf{v}` will render as $\mathbf{v}$. However, this won't work for Greek characters, which will require you to use the `\boldsymbol{}` macro; `\boldsymbol{\theta}` will correctly show as $\boldsymbol{\theta}$ instead of θ.

Make sure you examine and familiarize yourselves with additional "convenience" math macros that have already been defined in `sty/macros.sty`. For example, `\det{}` can be used to denote determinants of matrices as in `\det{\mathbf{A}}`, which renders $\det(\mathbf{A})$.

Occasionally, we will be using "theorem"-like environments for definitions, theorems, examples, etc. In `packages.sty`, I have already defined several of them in `sty/packages.sty`, but the most useful of them will be `definition` (for formal definitions), `theorem` (for important results), `proposition` (for less important results) and `example` (for examples). Below you will find an example of a definition and a theorem. Once again, please consult the pertinent source code to understand how to use these environments.

Definition 12 (Even Number)**.** An integer n is called even, if it is divisible by 2.

Theorem 11 (Fundamental Theorem of Algebra)**.** *An N-th degree polynomial with complex-valued coefficients has exactly N roots.*

Proof. pro It is obvious, isn't it? :-) □

7.2.3 Figures, Tables & Algorithms

Figures should be in `JPG` (JPEG), `PNG` (Portable Network Graphics), `EPS` (Encapsulated PostScript) or `PDF` (Portable Document Format) format and stored in `sty/figures`. These figures could be either the product of a software-generated graph (*e.g.*, via MATLAB®), if precision is required (consult with me), or can be a scanned version of a (nicely) hand-drawn depiction. However, please avoid digitizing such depiction using a cell phone camera; make use of a scanner (*e.g.*, in the library) to do so. Also, please make sure that you do not use a resolution more than 300 DPIs, when scanning, to avoid large files; Overleaf provides only a limited amount of space for projects. Figures should always feature a caption that concisely explains what is depicted and what it means; the same thing applies to table captions. Very Important: If you use an image from another source, in the figure's caption, either include a reference to it (*e.g.*, if it is a paper or book you got it from) or a link (*e.g.*, if you got it from a Wikipedia page).

An example of a figure is shown in Figure 7; make sure you also consult the corresponding source code.

Tables in LaTeX have quite a bit of learning curve; you may want to refer to the LaTeX Wikibook mentioned earlier for more details and "recipes" for creating tables. The good news are that I do not anticipate that we will be using a lot of tables in these lecture notes. A simple example of a table is given in Table 7.2.3.

Dr. A's Comment(s): *Apparently, there is a problem with correctly referencing this table; I will investigate this.*

A Bunch of Nonsense	
ABC	AAAAAAAAAAAAAAAAAAAAAAA
DEF	BBBBBBBBBBBBBBBBBBBBBBB
GHI	CCCCCCCCCCCCCCCCCCCCCCC

Table 1: Some random table caption here.

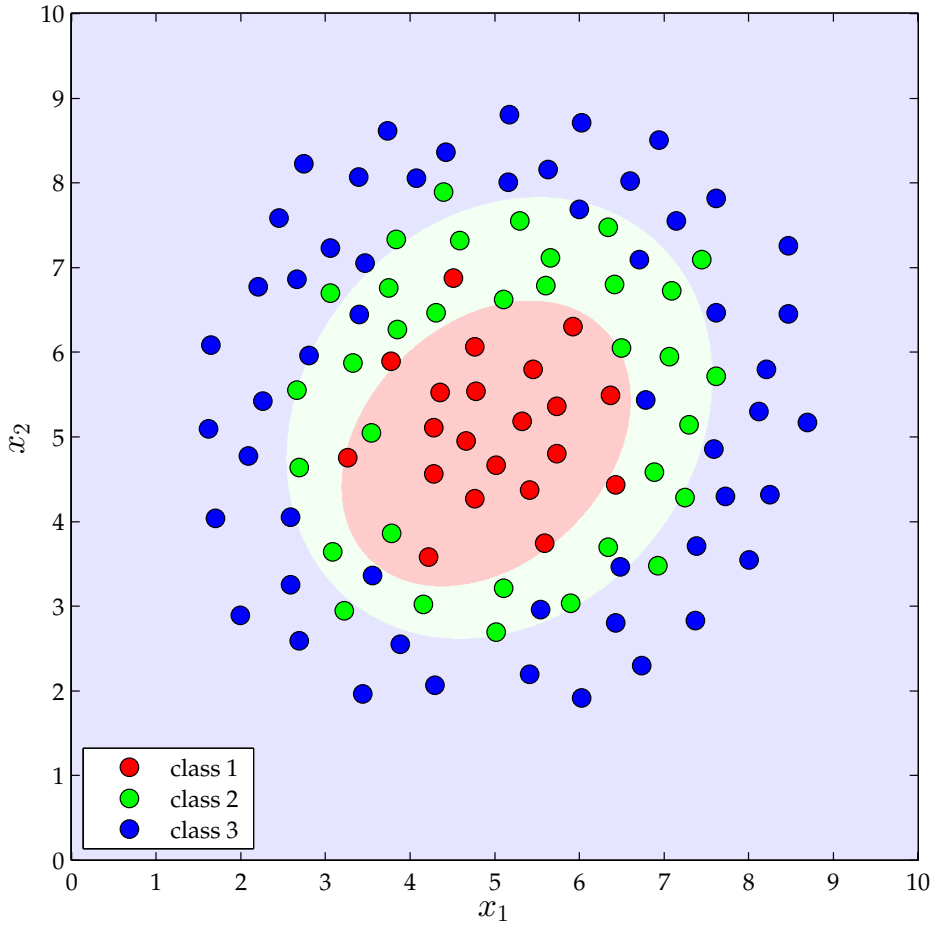

Figure 7: Classification results of a multi-nomial regression model applied to a 3-class toy problem with artificially-generated data. What is shown are the training samples (depicted as points) and the 3 decision regions obtained after training. This illustrates how a non-linear data transformation allows the model to distinguish between these 3 non-linearly separable data distributions. Source: Figure 10 of [2] (for example).

Algorithms in LaTeX also have some learning curve, albeit a less steep one. Please consult the Algorithms Package Manual for all the details. An example is given in Algorithm 1; again, please consult the source code.

Algorithm 1 PCA training based on the sample covariance matrix of the data.

Input: Training samples $\mathbf{X} \in \mathbb{R}^{N \times D}$, fraction $0 \leq p \leq 1$ of total variance explained.
Output: Principal components $\mathbf{V}_H \in \mathbb{R}^{D \times H}$, sample mean $\hat{\mu}_x \in \mathbb{R}^D$.

// Compute sample mean $\hat{\mu}_x$, e.g.
$\hat{\mu}_x \leftarrow \frac{1}{N}\mathbf{X}^T\mathbf{1}_N$
// Compute sample covariance matrix $\hat{\mathbf{C}}_x$ from $\mathbf{X}$, e.g.
$\widetilde{\mathbf{X}} \leftarrow \mathbf{X} - \mathbf{1}_N\hat{\mu}_x^T = \mathbf{P}_N\mathbf{X}$
$\hat{\mathbf{C}}_x \leftarrow \frac{1}{N}\widetilde{\mathbf{X}}^T\widetilde{\mathbf{X}}$
// Perform Eigen-Value Decomposition (EVD) on $\hat{\mathbf{C}}_x$. It is assumed that the eigen-pairs are sorted in descending order of eigenvalues and that the eigenvectors are all normalized to unit Euclidean length.
$[\mathbf{V}, \lambda] \overset{\text{EVD}}{\leftarrow} \hat{\mathbf{C}}_x$
// Determine the number H of principal components that need to be retained.
$H \leftarrow \arg\min\left\{k \in \{1, \ldots, D\} : \sum_{i=1}^{k} \lambda_i \geq p\,\text{trace}\left\{\hat{\mathbf{C}}_x\right\}\right\}$
// Return the first H columns of $\mathbf{V}$.
return $\mathbf{V}_H, \hat{\mu}_x$.

7.2.4 Bibliographic References

Occasionally, we will be in need of citing bibliographic references, which may include books, papers and even external URLs to outside resources, like software or datasets. The first order of business is to add that reference into the project's bibliography file (or bib file, for short), namely `References.bib`. For LaTeX beginners, I highly recommend you to first use JabRef (mentioned earlier) to create such an entry and then copy-and-paste it into `References.bib` at the top of this file. Make sure each bibliographic entry (bibentry, for short) has a key (preferably, auto-generated by JabRef). Next, you can use the `\cite{}` macro to cite this reference inside the text. For example, `\cite{Bishop1995}` will render as [1] and an entry for that item will be automatically included in the References section of this document.Here's an example of how you would reference certain pages of a book: `\cite[p. 1--10]{Bishop2007}` would render as [2, p. 1–10].

I quick tip: sometimes we want to preserve the capitalization of words in a bibitem's title, when it is displayed in the References section. In order to achieve this, enclose the title word(s) of the bibentry (residing in the bib file) in curly brackets `{...}`.

7.3 Notational Conventions

$\mathbb{R}$ will denote the set of real numbers, while $\mathbf{Z}$ will denote the set of all integers. Other arbitrary sets will mostly depicted in calligraphic form, *e.g.*, $\mathcal{X}$. Finite-dimensional vectors will be denoted by a lower-case letter in boldface, *e.g.*, $\mathbf{x} \in \mathbb{R}^D$, and will always be considered as column-vectors, while

ordinary matrices will be denoted by a upper-case boldface letter, *e.g.*, $\mathbf{A} \in \mathbb{R}^{M \times N}$. Furthermore, $\mathbf{A}^T \in \mathbb{R}^{N \times M}$ will denote $\mathbf{A}$'s transpose. If $\mathbf{A}$ is a square matrix (*i.e.*, $M = N$), then $\det(\mathbf{A})$ and $\text{trace}\{\mathbf{A}\}$ will denote its determinant and trace (sum of diagonal elements) respectively.

"iff" will stand for "if and only if." $\triangleq$ will stand for "defined as" or "let the Left Hand Side (LHS) be defined as the Right Hand Side (RHS)." Moreover, $\equiv$ will stand for "coincides with."

It is often convenient to use the *Iverson bracket*, which is defined as

$$[\text{predicate}] \triangleq \begin{cases} 1 & \text{if the predicate is true} \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

For example, $[a > 0] = 1$, if $a > 0$ is true.

For vectors, $\succ$, $\succcurlyeq$, $\prec$, $\preccurlyeq$ will stand for the component-wise version of the ordinary $>$, $\geq$, $<$ and $\leq$ relations; *e.g.*, $\mathbf{v} \succcurlyeq \mathbf{0}$ means that the components of $\mathbf{v}$ are all non-negative. The same relationships for symmetric matrices will denote their definiteness, *e.g.*, $\mathbf{A} \succcurlyeq 0$ conveys that $\mathbf{A}$ is positive semi-definite.

This subsection needs to be updated in the future as needed.

Updates

- Georgios C. Anagnostopoulos (GCA) (2020-01-24): Made OverLeaf project available.
- GCA (2020-01-24): Added draft versions of Lecture 1 and Lecture 2.
- GCA (2020-01-24): Added Lecture 3.
- GCA (2020-01-26): Added Lecture 4.
- Allen M. Shultz (AMS) (2020-02-03): Added Lecture 5.
- Luis A. Pantin (LAP) (2020-02-03): Added Lecture 6.