

Lecture 2: Towards Learning in Linear Models

Lecturer: Matthew Wicker

1 Learning Objectives

At the end of last lecture, we took a first step towards learning by defining a linear model. In this lecture, we will start by revisiting this model and will revisit vector calculus to observe how to fit this model to data.

2 Recall our Linear Models

Supervised learning Supervised learning settings assume that we have access input-output pairs, (x, y) , where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$. It is typically assumed that we have many such pairs comprising a dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^d$. When d (the number of data-points) is very small, we call the task *few-shot* learning. In lecture we will cover a variety of different advanced problem settings; however, for the mathematics in this course we will typically assume we have an abundance of data, and while we will see some unsupervised learning, we will focus on supervised regression in this lecture.

Linear Regression Model Linear regression is exactly what the name suggests: performing regression with a linear model. We assume a supervised setting where we have access to a dataset \mathcal{D} , as detailed above. More than just fitting the labels y , we assume some Gaussian noise. Assuming domain of \mathbb{R}^n and a one dimensional co-domain we can write our model as $f(x) = x^\top \theta$. Thus our model can be written as:

$$\hat{y}^{(i)} = \theta^\top x^{(i)}$$

The key idea behind learning is to find θ such that $\hat{y}^{(i)} \approx y^{(i)}$. We also note that you may have previously seen linear models written as affine transformation: $y = mx + b$ where the b is the bias or constant term. In this case, we simply can add a 1 entry to the end of each feature vector and then the corresponding weight for that feature vector becomes the bias. This can be viewed as the most simplistic form of basis expansion!

2.1 Basis Expansion

Though our linear model is, as we have formulated it, restricted to lines through the origin, we now present basis expansion as a way of modelling non-linear functions. To begin, let's start with

the simplest example, a one dimensional domain and a one dimensional co-domain. The key idea of basis expansion is to (1) expand our one-dimensional feature into many dimensions and (2) use non-linear functions to increase the expressiveness of our previously linear model. We start by considering a basis function $\phi(x) : \mathbb{R} \mapsto \mathbb{R}^3$, specifically we will first consider polynomial basis of the form $\phi(x) = \langle 1, x, x^2 \rangle$. We can then assign weights $\theta \in \mathbb{R}^3$ to each of our expanded features. This makes our fully expanded function:

$$\hat{y} = \theta_0 \phi(x)_0 + \theta^1 \phi(x)_1 + \theta^1 \phi(x)_1 \quad (1)$$

$$= \theta_0 + \theta^1 x + \theta^1 x^2 \quad (2)$$

We can then see that the output of the function is linearly related to the expanded features. Of course this basis expansion does not just work in one dimension. For example, if our unexpanded feature vector is two dimensional (as opposed to the one dimensional example we just gave), then we can have a basis expansion $\phi : \mathbb{R}^2 \mapsto \mathbb{R}^6$, and following the specific example from before this expansion would return the vector $\langle 1, x_1, x_2, x_1^2, x_2^2, x_1 x_2 \rangle$. As a minor exercise, you can expand this function as we have done above in the one dimensional case.

Radial Basis Function Kernel Importantly, polynomial basis expansion is just a single flavor of basis expansion. Another popular and widely used form of basis expansion is kernel basis expansion. Though we do not fully cover the extent of kernel basis expansion models here, they are good to work with to practice key skills (e.g., the vector calculus we will practice at the end of this lecture). A kernel is a function that takes in two points, x and x' , and returns a non-negative distance between those points. The most common kernel is the *radial basis function kernel* (a.k.a. RBF kernel) which takes a fixed parameter γ and is defined as $\kappa(x', x) = \exp(-\gamma \|x - x'\|^2)$. Typically, one picks fixed centers x and then the basis expansion proceeds by computing the expanded feature set with respect to the distance to these centers.

3 (Brief) Linear Algebra Review

In the lecture slides we do a far more detailed linear algebra review. For an expansion of this section, please see those slides.¹ Linear algebra concerns itself with different linear and affine transformations. The objects that describe such transformations are vectors, $x = \langle x_1, \dots, x_n \rangle$ which, having n real-valued entries we say “lives” in \mathbb{R}^n . Matrices, can be viewed as 2D arrays:

$$A = \begin{bmatrix} A_{0,0} & A_{1,0} \\ A_{0,1} & A_{1,1} \end{bmatrix}$$

It is important that we quickly review a few facts that you should always keep in mind when doing algebra in this course:

¹Future versions of these lecture notes will contain all of the material from the slides.

1. Matrix multiplication is **not** commutative: $AB \neq BA$
2. Matrix multiplication is associative: $A(BC) = (AB)C$
3. Matrix multiplication is distributive: $A(B + C) = AB + AC$
4. $(AB)^\top = A^\top B^\top$
5. If A is invertible then $A^{-1}A = I$ (it is useful to recall that a matrix is invertible when it is square and has non-negative eigen values).

Index Notation As reasoning about matrix and vector products can sometimes be a bit cumbersome, it is often useful to write out the operations we perform in index notation. The matrix product $C = AB$ can be written as: $C_{i,j} = \sum_K A_{i,k} B_{k,j}$ it can also be useful to adopt a more “pythonic” index notation where we consider the system $Ax = b$ and write the first entry of the vector as: $A_{1,:}x = b_1$ which indicates that the first value of the result is simply the dot product of the first row of matrix A with the vector x .

Matrix Decomposition As we have mentioned, matrix multiplication can sometimes be difficult to manipulate algebraically and it is useful to use linear algebra rules beyond the associativity and commutativity of matrices. The most common such rule is eigen decomposition. An eigenvector v and eigenvalue λ of a matrix A are a vector and real value satisfying:

$$Av = \lambda v$$

We will assume that you have solid working knowledge of linear algebra, but as a refresher take a moment to write out why it is the case (intuitively or algebraically) that an invertible matrix ($\in \mathbb{R}^{n \times n}$) has n linearly independent eigenvectors. The eigen decomposition of a matrix A is given by collecting its eigenvectors v_1, \dots, v_n into a matrix V and their corresponding eigenvalues into a vector $\lambda = \langle \lambda_1, \dots, \lambda_n \rangle$. We then write out the eigen decomposition as:

$$A = V \text{diag}(\lambda) V^{-1}$$

Another very useful component of eigen decomposition is the ability to bound certain matrix products:

$$\lambda_{\min}(A) \|x\|_2^2 \leq x^\top A x \leq \lambda_{\max}(A) \|x\|_2^2$$

Hopefully, this section has helped refresh your memory on linear algebra, but as with the rest of these lecture notes it is not an exhaustive list of concepts to remember. We refer those needed to brush up on their linear algebra skills to the linear algebra section of Ian Goodfellow’s deep learning book.

3.1 Index Notation

Example is given in the lecture notes where we prove the claim about the loss and the ℓ_2 norm

- Free indices appear only once in an expression and thus are not summed over. Dummy indices appear twice, and are implicitly summed over.
- To help avoid confusion, one tip is to use roman letters (i, j, k) for free indices, and greek letters (λ, μ, ρ) for dummy indices.
- Dummy indices should never appear in the final answer.
- The free indices should always be the same in every term in an expression.
- An index should never appear more than twice in a single term.

4 A Minimal Optimization Formulation

Given the above discussion of basis expansion, we can now write down a more general version of linear regression (including an expanded basis) which takes the form:

$$\hat{y}^{(i)} = \theta^\top \phi(x^{(i)})$$

The critical learning question in this model is then:

How do we pick the best θ ?

Unfortunately, this question does not have a straight-forward answer and depends on several critical modelling decisions that we will unravel throughout this course. However, for now, we will take the most basic approach which is the standard, frequentist learning approach. This is where we model the “best” θ as the one that minimizes a loss or error function, \mathcal{L} . One such loss is:

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \|y^{(i)} - \hat{y}^{(i)}\|_2^2$$

This is known as the ℓ_2 or mean squared error loss. Later in the course, we will pick apart exactly when and why someone might make this modelling choice. But, for now, as in previous courses, we will take this as a given. Thus, we can write down *learning* in the linear regression model as the following optimization problem:

$$\operatorname{argmin}_{\theta} [\mathbb{E}_{(x,y) \sim \mathcal{D}} \mathcal{L}(y, \hat{y})]$$

That is, the best θ is the value of θ that minimizes the expected loss. Notice that though it does not appear in the above equation, \hat{y} depends directly on θ . To solve this optimization problem in general, i.e., for general losses and basis expansions, we must turn to vector calculus as the foundation of general optimization techniques.

5 Calculus Revisited

Recall from your previous education in calculus that one can think about or define the derivative of a real-valued function, $f(x)$, as being the limit of the difference quotient:

$$f'(x) = \frac{df}{dx} = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (3)$$

We highlight that this also prescribes one way of approximating a derivative when differentiation is computationally difficult. For interested readers, you may look up zeroth-order optimization to learn more. Of course, in general, the entire field of calculus prescribes exactly how functions change locally around a given parameter value. For example, several very useful derivatives of common functions include:

$$\begin{array}{ll} f(x) = x^n & f'(x) = nx^{n-1} \\ f(x) = \sin(x) & f'(x) = \cos(x) \\ f(x) = \tanh(x) & f'(x) = 1 - \tanh^2(x) \\ f(x) = \exp(x) & f'(x) = \exp(x) \\ f(x) = \log(x) & f'(x) = \frac{1}{x} \end{array}$$

Moreover, it is good to refresh yourself on the key rules of differentiation that allow us to easily differentiate compositions of functions. You will find practice exercises at the bottom of these lecture notes. I highly recommend working on these as this skill will prove critical to practical problems in class and on your exam.

- Sum Rule

$$(f(x) + g(x))' = f'(x) + g'(x) = \frac{df(x)}{dx} + \frac{dg(x)}{dx}$$

- Product Rule

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x) = \frac{df(x)}{dx}g(x) + f(x)\frac{dg(x)}{dx}$$

- Chain Rule

$$(g \circ f)'(x) = (g(f(x)))' = g'(f(x))f'(x) = \frac{dg(f(x))}{df} \frac{df(x)}{dx}$$

- Quotient Rule

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{(g(x))^2} = \frac{\frac{df}{dx}g(x) - f(x)\frac{dg}{dx}}{(g(x))^2}$$

5.1 Gradients and Partial Derivatives

The above rules that we have recalled from your early calculus education can be generalized to real-valued functions f that take vectors as inputs, i.e., $f : \mathbb{R}^n \mapsto \mathbb{R}$. Notice first that if we simply look at the i^{th} index of our input vector and hold all other indices constant, then, as before, we can compute the derivative which we now denote $\partial f / \partial x_i$ which is known as the partial derivative:

Partial Derivative For a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $x \mapsto f(x)$, $x \in \mathbb{R}^n$ of n variables x_1, \dots, x_n , we define the partial derivatives as follows:

$$\begin{aligned}\frac{\partial f}{\partial x_1} &= \lim_{h \rightarrow 0} \frac{f(x_1 + h, x_2, \dots, x_n) - f(x)}{h}, \\ \frac{\partial f}{\partial x_2} &= \lim_{h \rightarrow 0} \frac{f(x_1, x_2 + h, \dots, x_n) - f(x)}{h}, \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= \lim_{h \rightarrow 0} \frac{f(x_1, x_2, \dots, x_n + h) - f(x)}{h},\end{aligned}$$

and collect them in the row vector ∇f (gradient of f):

$$\nabla f = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_n} \right].$$

Now that we have seen how the gradient of a function is the collection of its partial derivatives, let's look at a vector valued function and consider its derivative: $f : \mathbb{R}^n \mapsto \mathbb{R}^m$. In this case, we can again think about isolating the i^{th} output of the function and know that gradient with respect to the i^{th} output is simply a row vector of partial derivatives. So, it follows that the gradient of our function with respect to the input $\partial f / \partial x$ would be a matrix $\in \mathbb{R}^{m \times n}$ because we have a row vector for each of the outputs so we have m rows and n columns. This Jacobian matrix can be expressed:

$$\nabla f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad \frac{\partial f_i}{\partial x_j}$$

Where, f_i represents the i^{th} component of the vector-valued function f , and x_j represents the j -th component of the input vector. Finally, let's write out what the shapes of a matrix value function with respect to a matrix of outputs. It is clear that for every input and output we need to have a partial derivative so if we have a function $f : \mathbb{R}^{n \times m} \mapsto \mathbb{R}^{k \times l}$ then we will have a gradient which is a tensor of shape $(n \times m) \times (k \times l)$.

5.2 Vector Calculus Identities to Remember

Before moving on to seeing how vector calculus can help us in our goal of learning, let's revisit a few helpful identities:

$$\begin{aligned}
\frac{\partial x^\top a}{\partial x} &= a^\top \\
\frac{\partial}{\partial X} f(X)^\top &= \left(\frac{\partial f(X)}{\partial X} \right)^\top \\
\frac{\partial}{\partial X} \text{tr}(f(X)) &= \text{tr} \left(\frac{\partial f(X)}{\partial X} \right) \\
\frac{\partial}{\partial X} f(X)^{-1} &= -f(X)^{-1} \frac{\partial f(X)}{\partial X} f(X)^{-1} \\
\frac{\partial a^\top X b}{\partial X} &= ab^\top \\
\frac{\partial}{\partial s} (x - As)^\top W (x - As) &= -2(x - As)^\top W A
\end{aligned}$$

where the last identity assumes W is symmetric.

5.3 Using Linear Algebra & Calculus to Learn Parameters

Now that we have reviewed both calculus and linear algebra, we are ready to “learn” in our linear model. Recall that we are interested in a linear regression model in a supervised learning scenario. This is characterized by having a labelled dataset $\{(x^{(i)}, y^{(i)})\}_{i=0}^N$ with $x \in \mathbb{R}^k$ and for now we will assume $y \in \mathbb{R}$. The form of our linear-in-the-parameters model is:

$$y = \theta^\top \phi(x) + \epsilon$$

where $\phi : \mathbb{R}^k \mapsto \mathbb{R}^n$ and $\theta \in \mathbb{R}^n$ is our learnable parameters. We established in our last lecture that “learning” in this model equates to finding the best θ that fits our data. We also wrote out a loss function as:

$$\mathcal{L}(\hat{y}_i, y_i) = (\hat{y}_i - y_i)^2$$

or the squared ℓ_2 norm distance. Of course, this is formulated for a single prediction \hat{y}_i and label y_i , however, when we consider the vector $y = \langle y_1, \dots, y_N \rangle$ and the input matrix:

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

then, we can express the mean squared loss of a given parameter θ as: $\|\theta^\top \phi(X) - y\|_2^2$. Now, we can expand the loss of our model as:

$$(\theta^\top \phi(X) - y)^\top (\theta^\top \phi(X) - y)$$

Now, we will be slightly hand-wavy here and will come back to the intuition for this in Lecture 4, but this function is clearly quadratic in θ and so we can solve for its minimum by taking the derivative of the function and setting it to zero. We recommend that you do this step-by-step as an exercise, but we will write here that:

$$\nabla_{\theta} ||\theta^{\top} \phi(X) - y||_2^2 = -2\phi(X)^{\top} (y - \theta^{\top} \phi(X)) \quad (4)$$

Now, setting this equal to zero and solving we can find the optimal selection of θ :

$$\begin{aligned} -2\phi(X)^{\top} (y - \theta^{\top} \phi(X)) &= 0 \\ \phi(X)^{\top} y - \phi(X)^{\top} \phi(X) \theta^{\top} &= 0 \\ \phi(X)^{\top} y &= \phi(X)^{\top} \phi(X) \theta^{\top} \\ (\phi(X)^{\top} \phi(X))^{-1} \phi(X)^{\top} y &= \theta^{\top} \end{aligned}$$

To clean this up slightly, let's assume $\phi(X)$ is the identity function and therefore we have arrived at the fact that our least-squares estimate is: $(X^{\top} X)^{-1} X^{\top} y$.

6 Higher-Order Derivatives

While first-order gradients are fundamental to optimization techniques in machine learning, higher-order derivatives also play a key role in both optimization and analysis of models.

6.1 Second Derivative w.r.t. Scalar

Consider a function $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ of two variables x, y . We use the following notation for higher-order partial derivatives (and for gradients):

- $\frac{\partial^2 f}{\partial x^2}$ is the second partial derivative of f with respect to x .
- $\frac{\partial^n f}{\partial x^n}$ is the n th partial derivative of f with respect to x .
- For other variables x_i , we apply the chain rule: $\frac{\partial^2 f}{\partial x \partial y}$ is the partial derivative obtained by first partially differentiating with respect to x and then with respect to y .
- $\frac{\partial f}{\partial y \partial x}$ is the partial derivative obtained by first partially differentiating with respect to y and then with respect to x .

The Hessian is the collection of all second-order partial derivatives. If $f(x, y)$ is a twice (continuously) differentiable function, then: $\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x}$. We note that the order that you differentiate does not matter, you will end up with a symmetric matrix:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

That we call the Hessian. The Hessian is denoted as $\nabla_{x,y}^2 f(x, y)$. Generally, for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the Hessian is an $n \times n$ matrix. The Hessian measures the curvature of the function locally around (x, y) .

6.2 Using Second Derivative in Optimization

The Hessian matrix $\nabla_{x,y}^2 f(x, y)$ plays a crucial role in characterizing the local behavior of a function. In particular, it helps identify important properties related to function minima. If the Hessian matrix is *positive definite*, denoted as $\nabla_{x,y}^2 f(x, y) \succ 0$, it signifies that the function has a local minimum at the point (x, y) . This is a significant result because it implies that the function curvature is positive in all directions at that point. In simple terms, if the Hessian matrix is positive definite at a particular point, it indicates that the function is "bowl-shaped" around that point. This property is valuable in optimization problems as it helps in locating optima. However, it's important to note that a positive definite Hessian is a sufficient condition for a local minimum, but not a necessary one. There could be cases where the Hessian is positive definite, but the point is not a local minimum, for example, a saddle point. Therefore, while a positive definite Hessian is a strong indicator of a minimum, it should be used in conjunction with other optimization techniques and criteria to confirm the presence of a minimum. We will discuss these criteria in detail in lecture 4. We will additionally see how second order information like the Hessian can be used in approximating posterior distributions and in accelerating our optimization approaches in lecture 13.

A Lecture 2: Vector Differentiation

Question 1 (Circle). Consider a vector function $\mathbf{x}(t) = [\cos t \quad \sin t]^\top$.

- Draw the set of points that this function passes through.
- To build intuition, draw the velocity vector at a few points by considering the direction that the point moves in.
- Find the derivative $d\mathbf{x}/dt$. Draw this vector for some point t .

Question 2 (Index notation). Turn the following matrix-vector expressions into index notation:

- | | |
|--------------------|--|
| a. $ABC\mathbf{x}$ | c. $\text{Tr}(AB)$ |
| b. $\text{Tr}(A)$ | d. $\mathbf{y}^\top A^\top \mathbf{x}$ |

Turn the following index expressions back to matrix-vector notation:

- | | |
|--------------------------------------|-----------------------------|
| a. $\sum_{ijk} A_{ij} B_{jk} C_{ki}$ | c. $x_i x_j$ |
| b. $b_i + \sum_j A_{ij} b_j$ | d. $\sum_j \delta_{ij} a_j$ |

Question 3 (Index notation proofs). Using index notation, show that

- $\mathbf{x}^\top A \mathbf{y} = \mathbf{y}^\top A \mathbf{x}$ if A is symmetric, i.e. $A = A^\top$.
- $\mathbf{x}^\top \mathbf{y} = \text{Tr}(\mathbf{x}^\top \mathbf{y}) = \text{Tr}(\mathbf{y}^\top \mathbf{x})$, for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$.
- $\text{Tr}(ABC) = \text{Tr}(CAB)$.

Question 4 (MML 5.5-5.6). First find the dimensions, then the Jacobian. It's probably easiest here to use index notation.

- $f(\mathbf{x}) = \sin(x_1) \cos(x_2)$, find $df/d\mathbf{x}$.
- $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{y}$, find $df/d\mathbf{x}$.
- $f(\mathbf{x}) = \mathbf{x} \mathbf{x}^\top$, find $df/d\mathbf{x}$.
- $f(\mathbf{t}) = \sin(\log(\mathbf{t}^\top \mathbf{t}))$, find $df/d\mathbf{t}$.
- $f(X) = \text{Tr}(AXB)$ for $A \in \mathbb{R}^{D \times E}$, $X \in \mathbb{R}^{E \times F}$, $B \in \mathbb{R}^{F \times D}$, find df/dX .

Question 5 (MML 5.7-5.8: Chain rule). Compute the derivatives $df/d\mathbf{x}$ of the following functions.

- First, write out the chain rule for the given decomposition.

- Give the shapes of intermediate results, and make clear which dimension(s) will be summed over.
 - Provide expressions for the derivatives, and describe your steps in detail. Providing an expression means specifying everything up to the point where you could implement it.
 - Give the results in vector notation if you can.
- a. $f(z) = \log(1 + z)$, $z = \mathbf{x}^\top \mathbf{x}$, $\mathbf{x} \in \mathbb{R}^D$.
 - b. $f(\mathbf{z}) = \sin(\mathbf{z})$, $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{E \times D}$. What sizes are \mathbf{x} and \mathbf{b} ?
 - c. $f(z) = \exp(-\frac{1}{2}z)$, $z = \mathbf{y}^\top \mathbf{S}^{-1} \mathbf{y}$, $\mathbf{y} = \mathbf{x} - \boldsymbol{\mu}$.
 - d. $f(\mathbf{A}) = \text{Tr}(\mathbf{A})$, $\mathbf{A} = \mathbf{x}\mathbf{x}^\top + \sigma^2 \mathbf{I}$.
 - e. $f(\mathbf{z}) = \tanh(\mathbf{z})$, $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{M \times N}$.
 - f. $f(\mathbf{A}) = \mathbf{x}^\top \mathbf{A} \mathbf{x}$, $\mathbf{A} = \mathbf{x}\mathbf{x}^\top$.

Remember: Generally, scalar functions are applied elementwise to vectors/matrices.

Question 6 (Hessian of Linear Regression). For the stationary point of linear regression, find the Hessian, and prove that it is positive definite, perhaps by making some assumptions. Discuss your assumptions.

Question 7. Prove the following theorem:

Theorem A.1. *Given a positive semi-definite matrix $S \in \mathbb{R}^{m \times m}$, a feature vector $x' \in \mathbb{R}^m$, and a similarity threshold δ , all vectors $x'' \in \mathbb{R}^n$ satisfying $d_S(x', x'') \leq \delta$ are contained within the axis-aligned orthotope:*

$$\left[x' - \delta \sqrt{d}, x' + \delta \sqrt{d} \right]$$

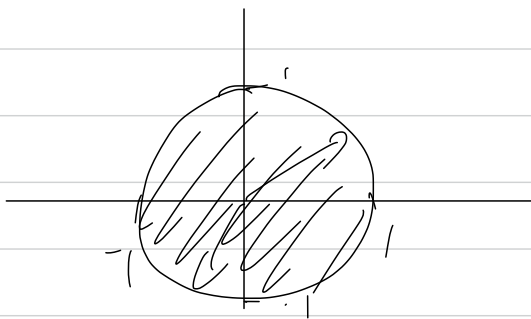
where $d = \text{diag}(S)$, the vector containing the elements along the diagonal of S .

Question 8. A colleague of yours suggests to study the transfer learning set up we discussed in class in a purely linear setting using linear models with basis expansion. Does this idea make sense? Formulate the problem and write down if and why you think transfer learning can be studied using linear models.

Question 1 (Circle). Consider a vector function $\mathbf{x}(t) = [\cos t \ \sin t]^T$.

- Draw the set of points that this function passes through.
- To build intuition, draw the velocity vector at a few points by considering the direction that the point moves in.
- Find the derivative $d\mathbf{x}/dt$. Draw this vector for some point t .

a, b



c

$$\mathbf{x} = \begin{pmatrix} \cos t \\ \sin t \end{pmatrix}$$

$$\frac{d\mathbf{x}}{dt} = \begin{pmatrix} -\sin t \\ \cos t \end{pmatrix}$$

Question 2 (Index notation). Turn the following matrix-vector expressions into index notation:

- \mathbf{ABCx}
- $\text{Tr}(\mathbf{A})$
- $\text{Tr}(\mathbf{AB})$
- $\mathbf{y}^T \mathbf{A}^T \mathbf{x}$

Turn the following index expressions back to matrix-vector notation:

- $\sum_{ijk} A_{ij} B_{jk} C_{ki}$
- $b_i + \sum_j A_{ij} b_j$
- $x_i x_j$
- $\sum_j \delta_{ij} a_j$

a.

$$\begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

$$M_{ij} = i^{\text{th}} \text{ row}, j^{\text{th}} \text{ col}$$

$$\sum_j \sum_k \sum_l A_{ij} B_{jk} C_{kl} x_l \quad ?$$

b. $\sum_i A_{ii}$

c. $\sum_i \sum_j A_{ij} b_j$

$$\begin{bmatrix} - & \cdot & \cdot \\ - & \cdot & \cdot \\ - & \cdot & \cdot \end{bmatrix} \begin{bmatrix} - & \cdot & \cdot \\ - & \cdot & \cdot \\ - & \cdot & \cdot \end{bmatrix}$$

d. $\begin{pmatrix} 1 \times n \end{pmatrix} \begin{pmatrix} n \times m \end{pmatrix} x$

$$\sum_{j,i} y_j A_{ij} x_i$$

Turn the following index expressions back to matrix-vector notation:

a. $\sum_{ijk} A_{ij} B_{jk} C_{ki}$

c. $x_i x_j$

b. $b_i + \sum_j A_{ij} b_j$

d. $\sum_j \delta_{ij} a_j$

a.
$$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

 $\text{Tr}(ABC)$

b.
$$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} + \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} A \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} b$$

c.
$$x x^T$$

d.
$$a$$

Question 3 (Index notation proofs). Using index notation, show that

1. $\mathbf{x}^T A \mathbf{y} = \mathbf{y}^T A \mathbf{x}$ if A is symmetric, i.e. $A = A^T$.

2. $\mathbf{x}^T \mathbf{y} = \text{Tr}(\mathbf{x}^T \mathbf{y}) = \text{Tr}(\mathbf{y}^T \mathbf{x})$, for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^D$.

3. $\text{Tr}(ABC) = \text{Tr}(CAB)$.

$$\begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}^T \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = \sum_i \sum_j x_i A_{ji} y_j$$

$$y^T A x = \sum_i y_i A_{ji} x_j$$

$$\sum_i \sum_j x_i A_{ji} y_j = y_j A_{ji} x_i = y_j A_{ij} x_i \quad \text{if } A = A^T$$

$$= y^T A x$$

2. $x^T y: \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix} = \sum_i x_i y_i = \text{scalar} \quad \therefore \text{Tr}(x^T y) = \text{scalar}$

$$y^T x: \sum_i y_i x_i = \sum_i x_i y_i \quad \therefore \text{Tr}(y^T x) = \text{Tr}(x^T y) = x^T y$$

$$3. \quad \text{Tr}(ABC) = \sum_i (ABC)_{ii} = \sum_i \sum_j (AB)_{ij} C_{ji} = \sum_i \sum_j \sum_k a_{ik} b_{kj} C_{ji}$$

$$\sum_i \sum_j \sum_k a_{ij} b_{jk} C_{ki} = \sum_{k,i,j} C_{ki} a_{ij} b_{jk} = \text{Tr}(CAB)$$

Question 4 (MML 5.5-5.6). First find the dimensions, then the Jacobian. It's probably easiest here to use index notation.

a. $f(\mathbf{x}) = \sin(x_1) \cos(x_2)$, find $df/d\mathbf{x}$.

b. $f(\mathbf{x}) = \mathbf{x}^T \mathbf{y}$, find $df/d\mathbf{x}$.

c. $f(\mathbf{x}) = \mathbf{x} \mathbf{x}^T$, find $df/d\mathbf{x}$.

d. $f(\mathbf{t}) = \sin(\log(\mathbf{t}^T \mathbf{t}))$, find $df/d\mathbf{t}$.

e. $f(\mathbf{X}) = \text{Tr}(\mathbf{A}\mathbf{X}\mathbf{B})$ for $\mathbf{A} \in \mathbb{R}^{D \times E}$, $\mathbf{X} \in \mathbb{R}^{E \times F}$, $\mathbf{B} \in \mathbb{R}^{F \times D}$, find $df/d\mathbf{X}$.

a.
$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \cos(x_1) \cos(x_2) \\ -\sin(x_1) \sin(x_2) \end{bmatrix}$$

b. $\dim(\mathbf{x}^T \mathbf{y})$

$$f = \mathbf{x}^T \mathbf{y} = \sum_i x_i y_i \quad \frac{\partial f}{\partial x_j} = \sum_i \frac{\partial x_i}{\partial x_j} y_i = \sum_i \delta_{ij} y_i = y_j \quad \text{if } i=j$$

$$\therefore \frac{\partial f}{\partial \mathbf{x}} = \mathbf{y}^T$$

c. $\begin{matrix} n \times 1 & 1 \times n \\ \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} & \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \end{matrix} \rightarrow \dim: n \times n$

$$\mathbf{d} = \mathbf{x} \mathbf{x}^T$$

$$\frac{\partial \mathbf{d}}{\partial \mathbf{x}} = \mathcal{L} \in \mathbb{R}^3$$

Let

$$\mathbf{x} \mathbf{x}^T = \begin{bmatrix} x_1^2 & x_1 x_2 & \dots & x_1 x_n \\ x_2 x_1 & x_2^2 & & x_2 x_n \\ \vdots & \vdots & \ddots & \vdots \\ x_n x_1 & x_n x_2 & \dots & x_n^2 \end{bmatrix}$$

$$\frac{\partial (x_i x_j)}{\partial x_k} = \frac{\partial x_i}{\partial x_k} x_j + \frac{\partial x_j}{\partial x_k} x_i$$

$$= \delta_{ik} x_j + \delta_{jk} x_i$$

$$M_{ij} = x_i x_j$$

$$= \begin{cases} x_j + x_i & \text{if } i=j=k \\ x_j & \text{if } k=i \neq j \\ x_i & \text{if } k=j \neq i \\ 0 & \text{if } k \neq i, j \end{cases}$$

d. $f = \sin(\log(\mathbf{t}^T \mathbf{t}))$

$$\mathbf{t}^T \mathbf{t} = \sum_i t_i t_i = \sum_i t_i^2 \rightarrow \frac{\partial \mathbf{t}^T \mathbf{t}}{\partial t_j} = \sum_i \delta_{ij} 2 t_i = 2 t_j = 2 \mathbf{t}^T$$

$$\frac{\partial f}{\partial \mathbf{t}} = \cos(\log(\mathbf{t}^T \mathbf{t})) \times \frac{\partial \log(\mathbf{t}^T \mathbf{t})}{\partial \mathbf{t}}$$

$$= \frac{1}{\mathbf{t}^T \mathbf{t}} \times 2 \mathbf{t}^T \rightarrow \cos(\log(\mathbf{t}^T \mathbf{t})) \times \frac{2 \mathbf{t}^T}{\mathbf{t}^T \mathbf{t}}$$

$$e, \text{Tr}(A^T B), \quad A: D \times F, \quad x: F \times F, \quad B: F \times D \quad \therefore A^T B = D \times D$$

$$= \sum_i^D (A^T B)_{ii} = \sum_i^D \sum_{k=1}^F (A^T)_{ik} b_{ki} = \sum_i \sum_k \sum_j a_{ij} x_{jk} b_{ki}$$

$$\frac{\partial}{\partial x_{nm}} = \sum_{ij} a_{ij} \frac{\partial x_{jk}}{\partial x_{nm}} b_{ki} = a_{ij} \delta_{jn} \delta_{km} b_{ki} = a_{ij} b_{ki} \text{ if } j=n \text{ \& } k=m$$

$$\rightarrow \sum_i a_{in} b_{mi} = \sum_i b_{mi} a_{in} = (B^T A)_{mn}$$

Question 5 (MML 5.7-5.8: Chain rule). Compute the derivatives df/dx of the following functions.

- First, write out the chain rule for the given decomposition.

- Give the shapes of intermediate results, and make clear which dimension(s) will be summed over.
- Provide expressions for the derivatives, and describe your steps in detail. Providing an expression means specifying everything up to the point where you could implement it.
- Give the results in vector notation if you can.

- $f(z) = \log(1+z)$, $z = \mathbf{x}^T \mathbf{x}$, $\mathbf{x} \in \mathbb{R}^D$.
- $f(\mathbf{z}) = \sin(\mathbf{z})$, $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{E \times D}$. What sizes are \mathbf{x} and \mathbf{b} ?
- $f(z) = \exp(-\frac{1}{2}z)$, $z = \mathbf{y}^T \mathbf{S}^{-1} \mathbf{y}$, $\mathbf{y} = \mathbf{x} - \boldsymbol{\mu}$.
- $f(\mathbf{A}) = \text{Tr}(\mathbf{A})$, $\mathbf{A} = \mathbf{x}\mathbf{x}^T + \sigma^2 \mathbf{I}$.
- $\mathbf{f}(\mathbf{z}) = \tanh(\mathbf{z})$, $\mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{M \times N}$.
- $f(\mathbf{A}) = \mathbf{x}^T \mathbf{A} \mathbf{x}$, $\mathbf{A} = \mathbf{x}\mathbf{x}^T$.

$$a. \log(1+z), \quad z = \mathbf{x}^T \mathbf{x} \quad \begin{bmatrix} 1 \times 1 \end{bmatrix} \begin{bmatrix} D \times 1 \end{bmatrix} = 1 \times 1$$

$$z = \sum_i x_i x_i = \sum_i x_i^2 \rightarrow \frac{\partial z}{\partial x_i} = 2x_i = 2\mathbf{x}^T$$

$$\frac{1}{1+z} \leftarrow z\mathbf{x}^T: \quad \frac{z\mathbf{x}^T}{1+x^T \mathbf{x}}$$

$$b. \sin(\mathbf{z}) \quad , \quad \mathbf{z} = \mathbf{A}\mathbf{x} + \mathbf{b} \quad \mathbf{A} = D \times D, \quad \mathbf{z} = D \times 1 \quad \mathbf{b} = E \times 1 \quad \mathbf{z}: E \times 1$$

$$z_i = \sum_j a_{ij} x_j + b_i \quad \frac{\partial z_i}{\partial x_j} = a_{ij} \rightarrow \frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \mathbf{A}$$

$$\frac{\partial}{\partial z_j} (\sin(z_i)): \quad \delta_{ij} \cos(z_i) \leftarrow \dots = \cos(z_i) \text{ if } i=j \rightarrow \begin{bmatrix} \cos(z_1) & 0 & 0 \\ 0 & \cos(z_2) & 0 \\ 0 & 0 & \cos(z_3) \end{bmatrix} = \text{diag}(\cos(z_i))$$

$$\underline{\text{diag}[\cos(\mathbf{z})]} \mathbf{A}$$

$$c. \quad e^{-\frac{1}{2}z} \quad , z: y^T S^{-1} y \quad , y = x - u$$

$$\text{chain rule: } e^{fem}: e^{g(x)} \times g'(x)$$

$$g(z): -\frac{1}{2}z$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y} \times \frac{\partial y}{\partial x}$$

$$\frac{\partial z}{\partial y}: \frac{\partial (y^T S^{-1} y)}{\partial y} \xrightarrow{\text{use ident.}} \frac{\partial x^T B x}{\partial x} = x^T (B + B^T) \rightarrow y^T (S^{-1} + S^{-T})$$

$$\frac{\partial y}{\partial x} = I$$

$$\rightarrow e^{-\frac{1}{2}((x-u)^T S^{-1} (x-u))} (x-u)^T (S^{-1} + S^{-T})$$

$$d \quad \text{Tr}(A), A: xx^T + \sigma^2 I, \quad x = p \times 1$$

$$xx^T = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

$$\text{Tr}(A): \sum_{i=1}^p [(xx^T)_{ii} + \sigma^2] = \sum_{i=1}^p (xx^T)_{ii} + p\sigma^2$$

$$xx^T = x_i^2 \rightarrow \sum_{i=1}^p (x_i^2 + \sigma^2)$$

$$\frac{\partial (x_i)^2}{\partial x_i} = 2x_i = 2x^T$$

$$\therefore = 2x^T$$

$$e \quad \tanh(z), \quad z: Ax+b \quad A: m \times n \quad x: n \times 1 \quad b: m \times 1$$

$$\tanh(\cdot) \rightarrow (1 - \tanh^2(\cdot)) \times d(\cdot)$$

$$Ax \rightarrow A_i = \sum_j a_{ij} x_j \xrightarrow{\text{d.w.r.t. } x_j} \frac{\partial}{\partial x_j}: a_{ij} = A$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial z} \frac{\partial z}{\partial x}$$

$$\text{diag} [1 - \tanh^2(Ax+b)] A$$

$$f(A): x^T A x, \quad A = x x^T$$

$$f(x): x^T x x^T x :$$

