矩阵求导术(上)



长躯鬼侠 · 9 个月前

矩阵求导的技术, 在统计学、控制论、机器学习等领域有广泛的应用。鉴于我看过的一些资料 或言之不详、或繁乱无绪,本文来做个科普,分作两篇,上篇讲标量对矩阵的求导术,下篇讲 矩阵对矩阵的求导术。本文使用小写字母x表示标量, 粗体小写字母 **2**表示向量, 大写字母X 表示矩阵。

首先来琢磨一下定义,标量时矩阵X的导数,定义为 $\dfrac{\partial f}{\partial X}:=\left\lceil \dfrac{\partial f}{\partial X_{i:i}} \right
ceil$,即时X逐元素求导

排成与X尺寸相同的矩阵。然而,这个定义在计算中并不好用,实用上的原因是在对较复杂的 函数难以逐元素求导;哲理上的原因是逐元素求导破坏了整体性。试想,为何要将1看做矩阵X 而不是各元素 X_{ij} 的函数呢?答案是用矩阵运算更整洁。所以在求导时不宜拆开矩阵,而是 要找一个从整体出发的算法。为此,我们来回顾,一元微积分中的导数(标量对标量的导数) 与微分有联系: df=f'(x)dx ;多元微积分中的梯度(标量对向量的导数)也与微分有联

系:
$$df=\sum_i \frac{\partial f}{\partial x_i} dx_i = \frac{\partial f}{\partial x}^T dx$$
,这里第一个等号是全微分公式,第二个等号表达了梯度 $\frac{\partial f}{\partial x}$ 与微分的联系;受此启发,我们将矩阵导数与微分建立联系:

知

|三| 写文章

$$df = \sum_{i,j} rac{\partial f}{\partial X_{ij}} dX_{ij} = ext{tr}\left(rac{\partial f}{\partial X}^T dX
ight)$$
,这里tr代表迹(trace)是方阵对角线元素之

和,满足性质:对尺寸相同的矩阵A,B, $\mathbf{tr}(A^TB)=\sum_{i,j}A_{ij}B_{ij}$,这用泛函分析的语言来说 $\mathbf{tr}(A^TB)$ 是矩阵A,B的**内积**,因此上式与原定义相容。

然后来建立运算法则。回想遇到较复杂的一元函数如 $f = \log(2 + \sin x)e^{\sqrt{x}}$,我们是如何求导的呢?通常不是从定义开始求极限,而是先建立了初等函数求导和四则运算、复合等法则,再来运用这些法则。故而,我们来创立常用的矩阵微分的运算法则:

- 1. 加减法: $d(X\pm Y)=dX\pm dY$;矩阵乘法:d(XY)=dXY+XdY ;转置: $d(X^T)=(dX)^T$;迹: $d\mathrm{tr}(X)=\mathrm{tr}(dX)$ 。
- 2. 逆: $dX^{-1} = -X^{-1}dXX^{-1}$ 。此式可在 $XX^{-1} = I$ 两侧求微分来证明。
- 3. 行列式: $d|X|=\mathrm{tr}(X^\#dX)$,其中 $X^\#$ 表示X的伴随矩阵,在X可逆时又可以写作 $d|X|=|X|\mathrm{tr}(X^{-1}dX)$ 。此式可用Laplace展开来证明,详见张贤达《矩阵分析与应用》第279页。
- 4. 逐元素乘法: $d(X\odot Y)=dX\odot Y+X\odot dY$, \odot 表示尺寸相同的矩阵X,Y逐元素相乘。
- 5. 逐元素函数: $d\sigma(X)=\sigma'(X)\odot dX$, $\sigma(X)=[\sigma(X_{ij})]$ 是逐元素运算的标量函数。

知

矩阵求导术(上)-知乎专栏 2017/9/29

我们试图利用矩阵导数与微分的联系 $df=\mathrm{tr}\left(rac{\partial f}{\partial X}^TdX
ight)$,在求出左侧的微分 df后,

该如何写成右侧的形式并得到导数呢?这需要一些迹技巧(trace trick):

- 1. 标量套上迹: $a = \operatorname{tr}(a)$ 。
- 2. 转置: $\operatorname{tr}(A^T) = \operatorname{tr}(A)$ 。
- 3. 线性: $\operatorname{tr}(A \pm B) = \operatorname{tr}(A) \pm \operatorname{tr}(B)$ 。
- 4. 矩阵乘法交换: $\mathrm{tr}(AB)=\mathrm{tr}(BA)$ 。两侧都等于 $\sum_{i,j}A_{ij}B_{ji}$ 。
- 5. 矩阵乘法/逐元素乘法交换: $\operatorname{tr}(A^T(B\odot C)) = \operatorname{tr}((A\odot B)^TC)$ 。 两侧都等于 $\sum_i A_{ij} B_{ij} C_{ij}$.

观察一下可以断言,**若标量函数f是矩阵X经加减乘法、行列式、逆、逐元素函数等运算构成**, 则使用相应的运算法则对f求微分,再使用迹技巧给df套上迹并将其它项交换至dX左侧,即能 得到导数。

在建立法则的最后,来谈一谈复合:假设已求得 $\frac{\partial f}{\partial Y}$,而Y是X的函数,如何求 $\frac{\partial f}{\partial X}$ 呢?在

微积分中有标量求导的链式法则 $\frac{\partial f}{\partial x}=\frac{\partial f}{\partial y}\frac{\partial y}{\partial x}$,但这里我们不能沿用链式法则,因为矩阵对矩阵的导数 $\frac{\partial Y}{\partial X}$ 截至目前仍是未定义的。于是我们继续追本溯源,链式法则是从何而来?

知

|三| 写文章 000 源头仍然是微分。我们直接从微分入手建立复合法则:先写出 $df=\mathrm{tr}\left(rac{\partial f}{\partial Y}^TdY
ight)$,再将dY用dX表示出来代入,并使用迹技巧将其他项交换至dX左侧,即可得到 $rac{\partial f}{\partial X}$ 。

接下来演示一些算例。特别提醒要依据已经建立的运算法则来计算,不能随意套用微积分中标量导数的结论,比如认为AX对X的导数为A,这是没有根据、意义不明的。

例1:
$$oldsymbol{f} = oldsymbol{a}^T X oldsymbol{b}$$
,求 $rac{\partial f}{\partial X}$ 。

解:先使用矩阵乘法法则求微分: $df = \boldsymbol{a}^T dX \boldsymbol{b}$,再套上迹并做交换:

$$df=\mathrm{tr}(oldsymbol{a}^TdXoldsymbol{b})=\mathrm{tr}(oldsymbol{b}oldsymbol{a}^TdX)$$
,对照导数与微分的联系,得到 $rac{\partial f}{\partial X}=oldsymbol{a}oldsymbol{b}^T$ 。

注意:这里不能用 $\dfrac{\partial f}{\partial X}=m{a}^T\dfrac{\partial X}{\partial X}m{b}=$?,导数与乘常数矩阵的交换是不合法则的运算(而

微分是合法的)。有些资料在计算矩阵导数时,会略过求微分这一步,这是逻辑上解释不通的。

例2【线性回归】:
$$oldsymbol{l} = \|oldsymbol{X}oldsymbol{w} - oldsymbol{y}\|^2$$
,求 $rac{\partial l}{\partial oldsymbol{w}}$ 。

解:严格来说这是标量对向量的导数,不过可以把向量看做矩阵的特例。将向量范数写成 $m{l} = (m{X}m{w} - m{y})^T (m{X}m{w} - m{y})$,求微分,使用矩阵乘法、转置等法则:

知

$$dl=(Xdm{w})^T(Xm{w}-m{y})+(Xm{w}-m{y})^T(Xdm{w})=2(Xm{w}-m{y})^TXdm{w}$$
。 对照导数与微分的联系,得到 $rac{\partial l}{\partial m{w}}=2X^T(Xm{w}-m{y})$ 。

例3【多元logistic回归】: $m{l} = -m{y}^T \log \operatorname{softmax}(W m{x})$,求 $\frac{\partial l}{\partial W}$ 。其中 $m{y}$ 是除一个元素为1外其它元素为0的向量; $\operatorname{softmax}(m{a}) = \frac{\exp(m{a})}{\mathbf{1}^T \exp(m{a})}$,其中 $\exp(m{a})$ 表示逐元素求指数, $\mathbf{1}$ 代表全1向量。

解:首先将softmax函数代入并写成

$$m{l} = -m{y}^T \left(\log(\exp(Wm{x})) - \mathbf{1} \log(\mathbf{1}^T \exp(Wm{x})) \right) = -m{y}^T Wm{x} + \log(\mathbf{1}^T \exp(Wm{x}))$$
,这里要注意向量除标量求逐元素log满足 $\log(m{b}/c) = \log(m{b}) - \mathbf{1} \log(c)$,以及 $m{y}$ 满足 $m{y}^T \mathbf{1} = \mathbf{1}$ 。求微分,使用矩阵乘法、逐元素函数等法则:

$$dl = -m{y}^T dWm{x} + rac{m{1}^T \left(\exp(Wm{x}) \odot (dWm{x})
ight)}{m{1}^T \exp(Wm{x})}$$
。 再套上迹并做交换,其中第二项的

分子是

$$\operatorname{tr}\left(\mathbf{1}^T\left(\exp(Woldsymbol{x})\odot(dWoldsymbol{x})
ight)
ight)=\operatorname{tr}\left((\mathbf{1}\odot\exp(Woldsymbol{x}))^TdWoldsymbol{x}
ight)=\operatorname{tr}(\exp(Woldsymbol{x})^TdWoldsymbol{x}
ight)$$
,故

$$dl = ext{tr}\left(-oldsymbol{y}^T dWoldsymbol{x} + rac{\exp(Woldsymbol{x})^T dWoldsymbol{x}}{oldsymbol{1}^T \exp(Woldsymbol{x})}
ight) = ext{tr}(oldsymbol{x}(ext{softmax}(Woldsymbol{x}) - oldsymbol{y})^T dW)$$

。对照导数与微分的联系,得到
$$rac{\partial l}{\partial W} = (ext{softmax}(Wm{x}) - m{y})m{x}^T$$
。

知

2017/9/29 矩阵求导术(上)- 知乎专栏

另解:定义 $oldsymbol{a} = Woldsymbol{x}$,则 $oldsymbol{l} = -oldsymbol{y}^T \log \operatorname{softmax}(oldsymbol{a})$,先如上求出

$$rac{\partial l}{\partial m{a}} = \operatorname{softmax}(m{a}) - m{y}$$
,再利用复合法则:

$$doldsymbol{l} = ext{tr}\left(rac{\partial l}{\partial oldsymbol{a}}^T doldsymbol{a}
ight) = ext{tr}\left(rac{\partial l}{\partial oldsymbol{a}}^T dWoldsymbol{x}
ight) = ext{tr}\left(oldsymbol{x}rac{\partial l}{\partial oldsymbol{a}}^T dW
ight)$$
,得到 $rac{\partial l}{\partial W} = rac{\partial l}{\partial oldsymbol{a}}oldsymbol{x}^T$ 。

例4【方差的最大似然估计】:样本 $m{x}_1,\ldots,m{x}_n\sim N(m{\mu},\Sigma)$,其中 $m{\Sigma}$ 是对称正定矩阵,求方差 $m{\Sigma}$ 的最大似然估计。写成数学式是:

$$l=\log |\Sigma|+rac{1}{n}\sum_{i=1}^n(m{x}_i-ar{m{x}})^T\Sigma^{-1}(m{x}_i-ar{m{x}})$$
,求 $rac{\partial l}{\partial \Sigma}$ 的零点。

解:首先求微分,使用矩阵乘法、行列式、逆等运算法则,第一项是

$$d\log |\Sigma| = |\Sigma|^{-1} d|\Sigma| = \operatorname{tr}(\Sigma^{-1} d\Sigma)$$
,第二项是

$$rac{1}{n}\sum_{i=1}^n(oldsymbol{x}_i-ar{oldsymbol{x}})^Td\Sigma^{-1}(oldsymbol{x}_i-ar{oldsymbol{x}})=-rac{1}{n}\sum_{i=1}^n(oldsymbol{x}_i-ar{oldsymbol{x}})^T\Sigma^{-1}d\Sigma\Sigma^{-1}(oldsymbol{x}_i-ar{oldsymbol{x}})\,.$$

再给第二项套上迹做交换: $dl=\mathrm{tr}\left(\left(\Sigma^{-1}-\Sigma^{-1}S_n\Sigma^{-1}\right)d\Sigma
ight)$,其中

$$S_n:=rac{1}{n}\sum_{i=1}^n(oldsymbol{x}_i-ar{oldsymbol{x}})(oldsymbol{x}_i-ar{oldsymbol{x}})^T$$
 定义为样本方差。对照导数与微分的联系,有

$$rac{\partial l}{\partial \Sigma} = (\Sigma^{-1} - \Sigma^{-1} S_n \Sigma^{-1})^T$$
,其零点即 Σ 的最大似然估计为 $\Sigma = S_n$ 。

最后一例留给经典的神经网络。神经网络的求导术是学术史上的重要成果,还有个专门的名字

知

导术来推导并不复杂。为简化起见,我们推导二层神经网络的BP算法。

例5【二层神经网络】:
$$l = - m{y}^T \log \operatorname{softmax}(W_2 \sigma(W_1 m{x}))$$
,求 $\frac{\partial l}{\partial W_1}$ 和 $\frac{\partial l}{\partial W_2}$ 。

其中 $m{y}$ 是除一个元素为1外其它元素为0的向量, $ext{softmax}(m{a}) = rac{\exp(m{a})}{m{1}^T \exp(m{a})}$ 同例3,

$$\sigma(\cdot)$$
是逐元素sigmoid函数 $\sigma(a)=rac{1}{1+\exp(-a)}$ 。

解:定义
$$oldsymbol{a}_1 = W_1oldsymbol{x}$$
, $oldsymbol{h}_1 = \sigma(oldsymbol{a}_1)$, $oldsymbol{a}_2 = W_2oldsymbol{h}_1$,则

$$l=-m{y}^T\log \operatorname{softmax}(m{a_2})$$
。在例3中已求出 $rac{\partial l}{\partial m{a_2}}=\operatorname{softmax}(m{a_2})-m{y}$ 。使用复合

法则,注意此处 h_1,W_2 都是变量:

$$dl = ext{tr}\left(rac{\partial l}{\partial m{a}_2}^T dm{a}_2
ight) = ext{tr}\left(rac{\partial l}{\partial m{a}_2}^T dW_2m{h}_1
ight) + ext{tr}\left(rac{\partial l}{\partial m{a}_2}^T W_2 dm{h}_1
ight)$$
,使用矩

阵乘法交换的迹技巧从第一项得到
$$\dfrac{\partial l}{\partial W_2}=\dfrac{\partial l}{\partial m{a_2}}m{h}_1^T$$
 ,从第二项得到 $\dfrac{\partial l}{\partial m{h}_1}=W_2^T\dfrac{\partial l}{\partial m{a_2}}$

。接下来求 $\frac{\partial l}{\partial a_1}$,继续使用复合法则,并利用矩阵乘法和逐元素乘法交换的迹技巧:

$$\operatorname{tr}\left(rac{\partial l}{\partial oldsymbol{h}_{1}}^{T}doldsymbol{h}_{1}
ight)=\operatorname{tr}\left(rac{\partial l}{\partial oldsymbol{h}_{1}}^{T}(\sigma'(oldsymbol{a}_{1})\odot doldsymbol{a}_{1})
ight)=\operatorname{tr}\left(\left(rac{\partial l}{\partial oldsymbol{h}_{1}}\odot\sigma'(oldsymbol{a}_{1})
ight)^{T}doldsymbol{a}_{1}
ight)$$

,得到
$$\dfrac{\partial l}{\partial m{a}_1} = \dfrac{\partial l}{\partial m{h}_1} \odot \sigma'(m{a}_1)$$
。为求 $\dfrac{\partial l}{\partial W_1}$,再用一次复合法则:

$$\mathrm{tr}\left(rac{\partial l}{\partial oldsymbol{a}_1}^T doldsymbol{a}_1
ight) = \mathrm{tr}\left(rac{\partial l}{\partial oldsymbol{a}_1}^T dW_1oldsymbol{x}
ight) = \mathrm{tr}\left(oldsymbol{x}rac{\partial l}{\partial oldsymbol{a}_1}^T dW_1
ight)$$
,得到 $rac{\partial l}{\partial W_1} = rac{\partial l}{\partial oldsymbol{a}_1}oldsymbol{x}^T$ 。

下篇见zhuanlan.zhihu.com/p/24...。

机器学习 矩阵分析 优化



☆ 收藏 ○ 分享 ○ 坐报



58 条评论



写下你的评论...



方不觉

nbnb,一直对矩阵求导感觉无从下手,这几条法则比背公式好记多了!



resurrectcore

网上有个pdf叫做 The Matrix Cookbook.

9 个月前

6 赞



賈含章

写的很清楚,很实用!

9 个月前

1 赞



Liwei Cai 回复 resurrectcore

② 查看对话

我看过,相信写这篇文章的人也看过。感觉这里的方法比硬背公式简单,而且对于搞机器学习的这里的够用了

9个月前

3 赞



紫杉

期待下集。。我都忘了很多这部分内容了

9个月前

2 赞



ThinkingCat

写的真好,醍醐灌顶

9个月前

1 赞

9 个月前



王赟 Maigo

赞啊!终于找到系统的方法了!

9 个月前 4 赞



独孤阿毛

你好~请问一下,这是对矩阵函数求导,还是对函数矩阵求导?

9 个月前



渣渣

行列式微分的那个可以把行列式放进tr里变成伴随矩阵。行列式对其中元素的偏导显然就是 其代数余子式,应该不用行列式不为0。这公式叫jacobi's formula

9 个月前 1 赞

1 2 3 4 ... 6 下一页

https://zhuanlan.zhihu.com/p/24709748 10/10