Chapter 2 Linear Algebra

2.1 Scalars, Vectors, Matrices and Tensors

- Tensors : In some cases we will need an array with more than two axes. In the general case, an array of numbers arranged on a regular grid with a variable number of axes is known as a tensor.

  张量所描述的物理量是不随观察者或者说参照系而变化的，当参照系变化时（其实就是基向量变化），其分量也会相应变化，最后结果就是基向量与分量的组合（也就是张量）保持不变。

2.2 Multiplying Matrices and Vector

- 矩阵乘积以及向量乘积

2.3 Identity and Inverse Matrices

- 单位矩阵及逆矩阵

2.4 Linear Dependence and Span

- 线性相关及共轭矩阵

2.5 Norms(范数)

- we usually measure the size of vectors using a function called a norm.Formally, the L^P norm is given by

  $||x||_p = (\sum_i |x_i|^p)^{1/p}$
- One other norm that commonly arises in machine learning is the L^∞ norm, also known as the max norm.

  $||x||_\infty = \mathbf{max}_i\,|x_i|$
- 当要计算矩阵的模时,通常采用Frobenius norm

  $||A||_F = \sqrt{(}\sum_{i,j} A_{i,j}^2)$

2.6 Special Kinds of Matrices and Vectors

- Diagonal matrices(对角线矩阵):We write **diag(v)** to denote a square diagonal matrix whose diagonal entries are given by the entries of the vector **v**.即,v向量中的各维数字代表着对角矩阵中的对角元.
  - Not all diagonal matrices need be square. It is possible to construct a rectangular diagonal matrix.除了正方对角矩阵之外,也可以有长方对角矩阵.
- A symmetric matrix is any matrix that is equal to its own transpose(对称矩阵)
- A unit vector is a vector with unit norm.
- An orthogonal matrix is a square matrix whose rows are mutually orthonormal and whose columns are mutually orthonormal.正交矩阵

2.7 Eigendecomposition

$Av = \lambda v$

The eigendecomposition of A is then given by:

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

- Not every matrix can be decomposed into eigenvalues and eigenvectors. In some cases, the decomposition exists but involves complex rather than real numbers.有的时候,分解是个很复杂的.
- Specifically, every real symmetric matrix can be decomposed into an expression using only real-valued eigenvectors and eigenvalues.特别地,正对称矩阵经过特征值分解后,可分解为正交矩阵.
- positive definite:正定矩阵,特征值全为正值

  positive semidefinite:半正定矩阵,特征值全为正值和零

  negative definite:负定矩阵,,特征值全为正值

  negative semidefinite:半负定矩阵,,特征值全为正值和零

2.8 Singular Value Decomposition

- 奇异值分解与特征值分解的最大不同在于,特征值分解只能用于正方矩阵,而特征值分解能用于任意大小的矩阵.

  $A = UDV^T$
- 其中,U和V全为orthogonal matrices(正交矩阵),这一点也和特征值分解相同
- 在这里,D->被称为A的sigular value(奇异值).U的列向量被称为A的左奇异向量.V的列向量被称为A的右奇异向量.
- 这里还有一个深层解析,那就是
  - $AA^T = Udiag(\lambda)U^{-1}$:The left-singular vectors of A are the eigenvectors of AA^T
  - $A^TA = Vdiag(\lambda)V^{-1}$:The right-singular vectors of A are the eigenvectors of A^TA
  - $D = \sqrt(diag(\lambda))$:The nonzero singular values of A are the square roots of the eigenvalues of A^TA and AA^T

:

2.9 The Moore-Penrose Pseudoinverse(摩尔 - 彭罗斯伪逆)

- 矩阵求逆的操作只局限于正方矩阵,但是如果我一定要他的逆矩阵呢?
- 逆矩阵的定义就是,对于A,其逆矩阵应满足下面的方程组:

  $\begin{cases} Ax = y \\ x = A^{-1}y \end{cases}$
- 虽然我们无法向正方矩阵那样获得完美的逆矩阵,但是我们可以尽可能地逼近这个解,摩尔 - 彭罗斯伪逆应运而生.我们规定A的摩尔 - 彭罗斯伪逆为A^+.其理论解为:

  $A^+ = lim_{\alpha\to\infty}(A^TA + \alpha I)^{-1}A^T$

  但是真实计算的时候使用的是,

  $A+ = VD^+U^T$

  并且这里的D^+的算法很简单,就是把其对角线上非零元素取倒数后,再取其转置矩阵.
- 上面的这个方法,只对于m<n,即A has more columns than rows即行数大于列数.对于m>n的情况,只能说有所进展,这里的A^+效果的评价标准是,可以让

  $x = A^+y$

取得最接近原先x的值.

2.10 The Trace Operator(迹算子)

- The trace operator gives the sum of all the diagonal entries of a matrix: $Tr(A) = \sum_i A_{i,j}$
- 一些操作在不使用求和符号的情况下很难实现，但可以使用矩阵的乘积和迹算子指定。例如，迹算子提供了另外一种定义矩阵Frobenius范数的方式.

  $||A||_F = \sqrt(AA^T)$
- 范数:范数是一种强化了的距离概念，它在定义上比距离多了一条数乘的运算法则。有时候为了便于理解，我们可以把范数当作距离来理解。

  frobenius范数:计算矩阵的范数.这里之所以说,这是另外一种计算范数的方法是因为,一般的定义的范数是:

  $||A||_F = \sqrt(\sum_i \sum_j a_{i,j}^2)$

  即,每个元素的平方和的平方根(这里的范数都是L2范数)
- 迹算子还可以发现一些很有用的恒等式(identity)。例如:
  - 迹算子对转置算子是不变的： $Tr(A) = Tr(A^T)$
  - 方阵的乘积顺序对于迹算子没有影响:$Tr(ABC)=Tr(BCA)=Tr(CAB)

    更一般地: $Tr(\prod_{i=1}^{n} F^{(i)}) = Tr(F^{(n)} \prod_{i=1}^{n-1} F^{(i)})$
  - 更甚至,即使结果的尺寸也发生变化的时候,例如A,B分别是m*n和n*m,这种情况下依然:

    $Tr(AB) = Tr(BA)$

2.11 The Determinant

- 行列式的值
  - $det(A)$
- **行列式的值的意义**:The absolute value of the determinant can be thought of as a measure of how much multiplication by the matrix expands or contracts space. 即,把行列式作为一个转换器来看,它是对乘以它的变量的一个线性转换器,其中,行列式的值代表了其体量(体积)的变化,之前的变量在线性变化后是整体规模变大了,还是变小了,都要看这个行列式的值,若行列式的值为0,那么被乘的那个变量将会失去其体量,若行列式的值为1,则体量不变.

  这里的体量怎么理解呢?

  行列式等于它的各个列对应的向量张成的平行2n面体的体积，这是因为**行列式是一个交替多重线性形式，而我们通常理解的欧式空间中的体积也是这样一个函数**,（单位立方体体积为1，沿某条边扩大c倍体积就扩大c倍，交换两条边以后体积反号——这一条是补充定义的，我们认为体积是有向体积，其数值表示体积大小，正负号表示各条边的排列顺序或坐标轴手性）,而满足归一性、多线性、反对称性的函数是唯一的，所以行列式的直观理解就是欧式空间中的有向体积。(引用自知乎)见[链接][https://www.zhihu.com/question/26294660]

2.12 Example: Principal Components Analysis(主成分分析)

- 背景设计:对[one-hot向量][https://en.wikipedia.org/wiki/One-hot],进行有损压缩.Lossy compression means storing the points in a way that requires less memory but may lose some precision.
-
- 目的的公式化表达:
  - $x \in \mathbb{R}^n, c \in \mathbb{R}^l, n > c$,均为列向量
  - encoder:$f(x) = c$,
  - decoder:$g(c) = x$
  - 并且对于$f()$和$g()$的要求是,$x \approx f(g(x))$

- PCA is defined by our choice of the decoding function.为了使解码器简单化,我们使用矩阵相乘的方法,把c映射回n维空间,即

  $g(c) = Dc = x, D \in \mathbb{R}^{n*l}$,

  这里对于D有两个要求:

  - 由于有 $g(c) = Dc = x$,那么在计算decoder的时候,就要求$f(x) = D^{-1}x = c$,又因为,我们知道,正交矩阵的转置矩阵是其逆矩阵,因此,将 D 限定(constrain)为正交矩阵可以简化运算.(注意,实际的话,除非$l = n$,否则无法实现完全意义上的正交化,因为不是正方矩阵)
  - 在上面讲,The determinant 的时候,讲到过,**[行列式等于它的各个列对应的向量张成的平行2n面体的体积]**,对其一个维度的向量的**volume**进行扩大之后,其整个的**volume**也会变大,因此,根据

    $g(c) = Dc = x$

    这里的 x 是不变的,若是 D 的第 i 列的向量变大, c_i 的值应该是变小的,这样便对 c 进行了不平衡的转换,因此,也应该规定, D 的每一列都是单位向量.

- 下一步就是如何把**求** D 的算法实现.现在,我们看看我们要求的东西

  - 未知,D, c
  - 已知 x,以及公式 $x = Dc$

  就是说,一个公式里有两个未知数,因此我们必须假设已知一个,这里假设 D 已知.求 c 的最佳值 c*,然后通过转换把 c 用 D 表达.

  我们利用,范数计算,即经过D 转换的 c 得到的 x' 应该与原来的 x 在距离上是最近的,因此得到如下公式

  $c^* = argmin_c||x - g(c)||_2^2$

  这个式子右边的部分可以转化为:

  $(x - g(c))^T(x - g(c))$

  再根据分配法则(distributive property):

  $x^Tx - x^Tg(c) - g(c)^Tx + g(c)^Tg(c) = x^Tx - 2x^Tg(c) + g(c)^Tg(c)$

  由于 x 是已知的,因此,x^Tx这一项省略,变成了

  $c^* = argmin_c\{-2x^Tg(c) + g(c)^Tg(c)\}$

  再根据上边提到的 D 的正交性以及单

  位向量前提(注意,下面的小标 $_l$ 是c的维度,I_l就是指边长为 l 的单位矩阵):

  $c^* = argmin_c\{-2x^TDc + c^TD^TDc\} = argmin_c\{-2x^TDc + c^TI_lc\}$

  注意不要忘了,我们是在假设 D 已知的前提下,解决这个最优问题就用到矢量微积分(vector calculus),就是把矩阵当做变量,求函数的最小值.就是找到,函数曲线的凹点.

  $\nabla_c(-2x^TDc + c^Tc) = 0$

  $-2D^Tx + 2c = 0$

  $c = D^Tx$

  注意上面的最后一条,是可以取得最小误差的公式,得到了未知数c的表达式.

  这样就可以得到,相当于消去公式中的 c :

  $r(x) = g(f(x)) = DD^TX$

- 下一步就是如何选择编码器encoder的变量 D ,求其公式为:

  $D^* = argmax_D\sqrt{(}(x_j^{(i)} - r(x^{(i)})_j)$ **subject to** $D^TD = I_l$

  这里做了一个简化的假设,即假设 $l = 1$, 因此 $d \in \mathbb{R}^{n*1}$,$d^T * x$为标量,那上面那个公式变成了:

  $d^* = argmax_d||(x^{(i)} - dd^Tx^{(i)}||^2$ **subject to** $||d||_2 = 1$

$d^* = argmax_d||(x^{(i)} - d^T x^{(i)} d||^2$ **subject to** $||d||_2 = 1$

$d^* = argmax_d||(x^{(i)} - x^{(i)T} dd||^2$ **subject to** $||d||_2 = 1$

由于这里都是线性操作,将所有的$x^{(I)}$结合起来,使$X \in \mathbb{R}^{m*n}$,也就是说将原来的单个的列向量,横着放之后,又把m个点装在一个矩阵里面,,回到X的问题上,

$d^* = argmax_d||(X - Xdd^T)||_F^2$

从上上面那个等式到上面这个等式,一打眼看确实有些云里雾里,但是仔细推敲后发现没问题,需要倒过来看 再根据迹算子的性质:

$= argmax_d Tr((X - Xdd^T)^T(X - Xdd^T))$

$= argmax_d Tr(X^T X - X^T X dd^T - dd^T X^T X + dd^T X^T X dd^T)$

由于这是自变量为d的公式,去掉其中不含d的项

$d^* = argmax_d(-2Tr(X^T X dd^T) + Tr(dd^T X^T X dd^T))$

走到这里不要忘了,约束条件

$d^* = argmax_d(-2Tr(X^T X dd^T) + Tr(X^T X dd^T dd^T))$, subject to $d^T d = 1$

因此根据$d^T d = 1$

$d^* = argmax_d(-2Tr(X^T X dd^T) + Tr(X^T X dd^T))$

$= argmax_d(-Tr(X^T X dd^T))$

再根据迹算子的对称法则:

$= argmax_d(-Tr(d^T X^T X d))$

那么,下面这个问题就可以用特征值分解去解决,这里的d应该是,$X^T X$的最大的特征值对应的特征向量.(这里有些不太明白)

- 将 l的维度扩大后类别便可以得到答案.