try_frobeniusAndMatrixArgDiff

September 24, 2020

[1]:
```
# SOURCE = https://www.kannon.link/free/2019/10/30/
↪symbolic-matrix-differentiation-with-sympy/
from sympy import diff, symbols, MatrixSymbol, Transpose, Trace, Matrix, Function


def squared_frobenius_norm(expr):
    return Trace(expr * Transpose(expr))

k, m, n = symbols('k m n')

X = MatrixSymbol('X', m, k)
W = MatrixSymbol('W', k, n)
Y = MatrixSymbol('Y', m, n)

# Matrix(X)
A = MatrixSymbol('A', 3, 4)
B = MatrixSymbol('B', 4, 2)
C = MatrixSymbol('C', 3, 2)
Matrix(A)
```

[1]:

$$\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix}$$

[2]:
```
diff(squared_frobenius_norm(X*W - Y), W)
```

[2]:

$$2X^T(XW - Y)$$

[3]:
```
sq = squared_frobenius_norm(A*B - C); sq
```

[3]:

$$\operatorname{tr}\left((AB - C)\left(B^T A^T - C^T\right)\right)$$

[4]:
```
diff(squared_frobenius_norm(A*B - C), B)
```

[4]:

$$2A^T(AB - C)$$

[5]:
```
sq.args[0]
```

[5]:

$$(AB - C)\left(B^T A^T - C^T\right)$$

[6]:
```python
from sympy import srepr, expand, simplify, collect, factor, cancel, apart

#srepr(sq.args[0])
expand(sq.args[0])
```

[6]:

$$ABB^T A^T - ABC^T - CB^T A^T + CC^T$$

[7]:
```python
#diff(sq.args[0], B)
#diff(expand(sq.args[0]), B).doit()
from sympy import Symbol
Xm = Matrix(3,3, lambda i,j : Symbol("x_{}{}".format(i+1,j+1), commutative=True))
Wm = Matrix(3,2, lambda i,j : Symbol("w_{}{}".format(i+1,j+1), commutative=True))

X = MatrixSymbol('X',3,3)
W = MatrixSymbol('W', 3,2);
```

[8]:
```python
diff(X*W, X)
```

[8]:

$$\frac{\partial}{\partial X} XW$$

[9]:
```python
diff(X*W, X).subs({X:Xm})
```

[9]:

$$\frac{\partial}{\partial \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}} \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} W$$

[10]:
```python
diff(X*W, X).subs({X:Xm}).doit()
```

[10]:

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} W & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} W & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} W \\ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} W & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} W & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} W \\ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} W & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} W & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} W \end{bmatrix}$$

[11]:
```python
diff(X*W, X).subs({X:Xm}).doit().subs({W:Wm})
```

[11]:

$$\begin{bmatrix} \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} & \begin{bmatrix}0&1&0\\0&0&0\\0&0&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} & \begin{bmatrix}0&0&1\\0&0&0\\0&0&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} \\ \begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} & \begin{bmatrix}0&0&0\\0&1&0\\0&0&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&1\\0&0&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} \\ \begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&1&0\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} & \begin{bmatrix}0&0&0\\0&0&0\\0&0&1\end{bmatrix}\begin{bmatrix}w_{11}&w_{12}\\w_{21}&w_{22}\\w_{31}&w_{32}\end{bmatrix} \end{bmatrix}$$

[12]:
```
# expand(diff(X*W, X).subs({X:Xm}).doit().subs({W:Wm}))# STUCK doesn't work to␣
↪expand out from here
#diff(X*W, X).replace(X,Xm)# ERROR so I must use subs instead (noncommutatitve␣
↪scalars in matrix multiplication not supported)
diff(X*W, X).subs({X:Xm, W:Wm}).doit()
```

[12]:

$$\begin{bmatrix} \begin{bmatrix}w_{11}&w_{12}\\0&0\\0&0\end{bmatrix} & \begin{bmatrix}w_{21}&w_{22}\\0&0\\0&0\end{bmatrix} & \begin{bmatrix}w_{31}&w_{32}\\0&0\\0&0\end{bmatrix} \\ \begin{bmatrix}0&0\\w_{11}&w_{12}\\0&0\end{bmatrix} & \begin{bmatrix}0&0\\w_{21}&w_{22}\\0&0\end{bmatrix} & \begin{bmatrix}0&0\\w_{31}&w_{32}\\0&0\end{bmatrix} \\ \begin{bmatrix}0&0\\0&0\\w_{11}&w_{12}\end{bmatrix} & \begin{bmatrix}0&0\\0&0\\w_{21}&w_{22}\end{bmatrix} & \begin{bmatrix}0&0\\0&0\\w_{31}&w_{32}\end{bmatrix} \end{bmatrix}$$

[13]:
```
g,f = symbols('g f', cls = Function)
f(X).replace(X, X.T).diff(X).replace(X.T, X)
```

[13]:

$$\left.\frac{d}{d\xi_1} f(\xi_1)\right|_{\xi_1=X} \frac{d}{dX} X$$

[14]:
```
g(f(X)).replace(X, X.T).diff(X).replace(X.T, X)
```

[14]:

$$\left.\frac{d}{d\xi_1} f(\xi_1)\right|_{\xi_1=X} \frac{d}{df(X)} g(f(X)) \frac{d}{dX} X$$

[15]:
```
# f(X,W).replace(X,X.T).diff(X)### CRASHES
```

[16]:
```
```

[16]:
```
```

[16]:
```
```

[16]:
```
```

[16]:
```
```

[16]:
```
type(sq.args[0])
```

[16]: sympy.matrices.expressions.matmul.MatMul

[17]:
```
from sympy import symbols, Function

#h,g,f = symbols('h g f', cls=Function)
f = Function('f')
g = Function('g')
h = g(f(sq.args[0]))
h
```

[17]:
$$g\left(f\left((AB - C)\left(B^T A^T - C^T\right)\right)\right)$$

[18]:
```
diff(h, B)
```

[18]:
$$\frac{d}{d\xi_1} f(\xi_1)\Big|_{\xi_1=(AB-C)\left(B^T A^T - C^T\right)} \frac{\partial}{\partial f((AB - C)\,(B^T A^T - C^T))} g\left(f\left((AB - C)\left(B^T A^T - C^T\right)\right)\right) \frac{\partial}{\partial B}(AB - C)\left(B^T A^T -\right.$$

[19]:
```
from sympy import Derivative

#h.replace(f, Trace)
```

[20]:
```
diff(sq.args[0], B)
```

[20]:
$$\frac{\partial}{\partial B}(AB - C)\left(B^T A^T - C^T\right)$$

[21]:
```
from sympy import Trace


h = f(Trace(sq.args[0]))

diff(h, B)
```

[21]:
$$2\frac{d}{d\xi_1} f(\xi_1)\Big|_{\xi_1=\mathrm{tr}\left((AB-C)\left(B^T A^T - C^T\right)\right)} A^T (AB - C)$$

[22]:
```
h = g(f(A*B))
h
```

[22]:
$$g(f(AB))$$

[23]:
```
diff(h, A)
```

[23]:
$$\frac{d}{d\xi_1} f(\xi_1)\Big|_{\xi_1=AB} \frac{\partial}{\partial f(AB)} g(f(AB)) \frac{\partial}{\partial A} AB$$

[24]:
```python
from sympy import ZeroMatrix
Z = ZeroMatrix(3,4); Z
Matrix(Z)
```

[24]:
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[25]:
```python
type(A.T)
```

[25]: sympy.matrices.expressions.transpose.Transpose

[26]:
```python
type(Z + A)
```

[26]: sympy.matrices.expressions.matexpr.MatrixSymbol

[27]:
```python
type(A*1)
```

[27]: sympy.matrices.expressions.matexpr.MatrixSymbol

[28]:
```python
type(A)
```

[28]: sympy.matrices.expressions.matexpr.MatrixSymbol

[29]:
```python
type(A*B)
```

[29]: sympy.matrices.expressions.matmul.MatMul

[30]:
```python
from sympy.matrices.expressions.matexpr import MatrixExpr

#Matrix(MatrixExpr(A)) # ERROR
```

[31]:
```python

```

[31]:
```python
# diff(h, A) # WHAT THIS IS STILL BAD

# This is why:
assert type(A.T) != type(A.T.T)
#h = g(f(Z + A))
#D = MatrixSymbol('D', 3,4)

#ad = A+D
from sympy.abc import i,j,x,a,b,c

h = g(f(A.T))

h
```

[31]: $g\left(f\left(A^T\right)\right)$

[32]:
```
diff(h, A).replace(A.T,A)
```

[32]: $\left.\frac{d}{d\xi_1} f(\xi_1)\right|_{\xi_1=A} \frac{d}{df(A)} g(f(A)) \frac{d}{dA} A$

[33]:
```
diff(A.T, A).replace(A.T, A)
```

[33]: $\frac{d}{dA} A$

[34]:
```
diff(A.T, A).replace(A, Matrix(A))#.doit()
```

[34]: $$\frac{d}{d\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix}} \left(\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix}\right)^T$$

[35]:
```
diff(A.T, A).replace(A, Matrix(A)).doit()
```

[35]: $$\begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{bmatrix}$$

[36]:
```
from sympy import Symbol
from sympy.abc import b

#A = MatrixSymbol('A', 3,4)
M = Matrix(3,4, lambda i,j : Symbol('x_{}{}'.format(i+1,j+1)))
Matrix(M)
```

[36]: $$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}$$

[37]:
```
Matrix(A)
```

[37]:

$$\begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \end{bmatrix}$$

[38]:

```
g, f = symbols('g f', cls = Function)

#__ = lambda mat: mat.T # transposes matrix symbol

diff( g(f(M,b)), b)
```

[38]:

$$\frac{\partial}{\partial f\left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}, b\right)} g\left(f\left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}, b\right)\right) \left(\frac{\partial}{\partial b} f\left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix}, b\right) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right.$$

[39]:

```
diff( g(f(M,b)), b).replace(M, A)
```

[39]:

$$\frac{\partial}{\partial f(A,b)} g(f(A,b)) \left(\frac{\partial}{\partial b} f(A,b) + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\right)$$

[40]:

```
Ms = MatrixSymbol('M',2,2)
Ds = MatrixSymbol('D',2,2)
M = Matrix(2,2, lambda i,j: Symbol("m_{}{}".format(i+1,j+1)))
D = Matrix(2,2, lambda i,j: Symbol("d_{}{}".format(i+1,j+1)))

diff( g(f(M, D)), D )
```

[40]:

$$\left[\begin{matrix} \begin{bmatrix} \frac{\partial}{\partial \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}} f\left(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}\right) \frac{\partial}{\partial f\left(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}\right)} g\left(f\left(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}\right)\right) & 0 \\ 0 & 0 \end{bmatrix} & \left[\right. \\ \begin{bmatrix} 0 & 0 \\ \frac{\partial}{\partial \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}} f\left(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}\right) \frac{\partial}{\partial f\left(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}\right)} g\left(f\left(\begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}, \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}\right)\right) & 0 \end{bmatrix} & \left[\right. \end{matrix}\right.$$

[41]:

```
diff( g(f(M, D)), D ).replace(D, Ds).replace(M, Ms)
```

[41]:

$$\left[\begin{matrix} \begin{bmatrix} \frac{\partial}{\partial D} f(M,D) \frac{\partial}{\partial f(M,D)} g(f(M,D)) & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & \frac{\partial}{\partial D} f(M,D) \frac{\partial}{\partial f(M,D)} g(f(M,D)) \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ \frac{\partial}{\partial D} f(M,D) \frac{\partial}{\partial f(M,D)} g(f(M,D)) & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & \frac{\partial}{\partial D} f(M,D) \frac{\partial}{\partial f(M,D)} g(f(M,D)) \end{bmatrix} \end{matrix}\right]$$

[42]:

```
diff(Ds,Ds).replace(Ds,D).doit()
```

[42]:

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix}$$

[43]:

```
#diff( g(f(Ms, Ds.T)), Ds )#.replace(Ds.T, Ds)
```

[44]:

[44]:

[44]: