1MAE004 - OPTIMIZATION: Recipes for Quadratic form

Prof. Joseph Morlier

March 5, 2025

1 Scalar product?

Let $\boldsymbol{x}$ defined as $\begin{bmatrix} x_1 & x_2 & \ldots & x_m \end{bmatrix}^T$ and $\boldsymbol{y}$ as $\begin{bmatrix} y_1 & y_2 & \ldots & y_m \end{bmatrix}^T$. The scalar product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ of dimensions $(m \times 1)$ is the scalar that is obtained by summing the products of the respective components in a given basis:

$$< \boldsymbol{x}, \boldsymbol{y} >= x_1 y_1 + x_2 y_2 + \ldots + x_m y_m = \boldsymbol{x}^T \boldsymbol{y} = \boldsymbol{y}^T \boldsymbol{x}$$

The norm of a vector can be defined as $\|x\| = \sqrt{x^T x}$.

It is possible to show the Schwarz' inequality hold. $\left| x^T y \right| \leq \|x\| \|y\|$

2 Revisiting multivariable calculus

Let $f(x) = Ax$ (mapping $\mathbf{R}^n \rightarrow \mathbf{R}^m$) where A is a constant $m \times n$ matrix. Then,

$$df = d(Ax) = dAx + Adx = Adx \Longrightarrow f'(x) = A$$

We have $dA = 0$ here because A does not change when we change x. Consider the function $f(x) = Ax$ where A is a constant $m \times n$ matrix. Then, by applying the distributive law for matrix-vector products, we have

$$\begin{aligned} df &= f(x + dx) - f(x) = A(x + dx) - Ax \\ &= Ax + Adx - Ax = Adx = f'(x)dx \end{aligned}$$

Therefore, $f'(x) = A$.

Let $f(x) = Ax$ (mapping $\mathbf{R}^n \rightarrow \mathbf{R}^m$) where A is a constant $m \times n$ matrix. Then,

$$df = d(Ax) = d\widetilde{A}x + Adx = Adx \Longrightarrow f'(x) = A$$

We have $dA = 0$ here because A does not change when we change x.

Let $f(x) = x^T Ax$ (mapping $\mathbf{R}^n \rightarrow \mathbf{R}$). Then,

$$df = dx^T(Ax) + x^T d(Ax) = \underbrace{dx^T Ax}_{= x^T A^T dx} + x^T Adx = x^T \left(A + A^T\right) dx = (\nabla f)^T dx$$

and hence $f'(x) = x^T \left(A + A^T\right)$. (In the common case where A is symmetric, this simplifies to $f'(x) = 2x^T A$.) Note that here we have applied in the previous example in computing $d(Ax) = Adx$. Furthermore, f is a scalar valued function, so we may also obtain the gradient $\nabla f = \left(A + A^T\right) x$ as before (which simplifies to $2Ax$ if A is symmetric).

3 Bilinear form?

A bilinear form in the variables x_i and y_j is the scalar

$$B = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_i y_j$$

which can be written in matrix form

$$B(x, y) = x^T \mathbf{A} y = y^T \mathbf{A}^T x$$

where $x = \left[\begin{array}{cccc} x_1 & x_2 & \dots & x_m \end{array}\right]^T$, $y = \left[\begin{array}{cccc} y_1 & y_2 & \dots & y_n \end{array}\right]^T$, and $\boldsymbol{A}$ is the $(m \times n)$ matrix of the coefficients a_{ij} representing the core of the form.

What is the size of B(x, y)?

> 1xm x mxn x nx1 = 1x1; i.e. a scalar

How can you write this $x^T Ay = y^T A^T x$?

> The Bilinear form is a scalar (1x1). So I can write $scalar = scalar^T$
> HINTS: use $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$
> $x^T \mathbf{A} y = (x^T \mathbf{A} y)^T = (\mathbf{A} y)^T (x^T)^T = y^T \mathbf{A}^T x$

Given the bilinear form, the gradient of the form with respect to x is given by

$$\operatorname{grad}_x B(x, y) = \left(\frac{\partial B(\boldsymbol{x}, y)}{\partial x}\right)^T = \mathbf{A} y$$

But How?

> $\operatorname{grad}_x B(x, y) = \left(\frac{\partial B(x,y)}{\partial x}\right)^T = \left(\frac{\partial x^T \mathbf{A} y}{\partial x}\right)^T = \left(\frac{\partial y^T \mathbf{A}^T x}{\partial x}\right)^T = (y^T \mathbf{A}^T)^T = \mathbf{A} y$

whereas the gradient of B with respect to y is given (same demonstration) by

$$\operatorname{grad}_y B(x, y) = \left(\frac{\partial B(x, y)}{\partial y}\right)^T = \mathbf{A}^T x$$

4 Quadratic form?

A special case of bilinear form is the quadratic form

$$Q(x) = x^T \mathbf{A} x$$

Given the quadratic form with $\mathbf{A}$ symmetric $(A = A^T)$, the gradient of the form with respect to x is given by

$$\operatorname{grad}_x Q(x) = \left(\frac{\partial Q(x)}{\partial x}\right)^T = 2\mathbf{A} x$$

But How?

We therefore see that we must derive with respect to x an expression where x appears twice: once on the right, and once on the left. To visualize this, let's denote these two vectors in blue and red (this is just a play of color: whether it is written in red or blue, the vector x always represents the same thing!):

$$\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)=\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)+\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)$$

The derivative with respect to the red x does not pose a problem: you just need to apply the little math reminder framed above:

$$\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)=\mathbf{A} \mathrm{x}$$

For the derivative with respect to the blue x we must manage to bring the blue x to the left. To do this, we use the fact that Quadratic form is... a scalar! We can therefore happily transpose a scalar: this does not change it!

$$\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)=\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)^T$$

HINTS: use $(\mathbf{AB})^T=\mathbf{B}^T\mathbf{A}^T$ and $A=A^T$ i.e. A is symmetric.
It leads to:

$$\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)^T=\frac{\partial}{\partial \mathrm{x}}(\mathbf{A} \mathrm{x})^T\left(\mathrm{x}^T\right)^T=\frac{\partial}{\partial \mathrm{x}}(\mathrm{x})^T \mathbf{A}^T \mathrm{x}=\mathbf{A} \mathrm{x}$$

This leads to:

$$\frac{\partial}{\partial \mathrm{x}}\left(\mathrm{x}^T \mathbf{A} \mathrm{x}\right)=\mathbf{A} \mathrm{x}+\mathbf{A} \mathrm{x}=2 \mathbf{A} \mathrm{x}$$

5 Check with Python?

Consider a matrix of quadratic form given hereafter

$$A=\left[\begin{array}{ccc}3 & 2 & 0 \\ 2 & -1 & 4 \\ 0 & 4 & -2\end{array}\right]$$

construct the quadratic form $\mathbf{x}^T A \mathbf{x}$.

$$\begin{aligned}
\mathbf{x}^T A \mathbf{x} & =\left[\begin{array}{lll}x_1 & x_2 & x_3\end{array}\right]\left[\begin{array}{ccc}3 & 2 & 0 \\ 2 & -1 & 4 \\ 0 & 4 & -2\end{array}\right]\left[\begin{array}{l}x_1 \\ x_2 \\ x_3\end{array}\right] \\
& =\left[\begin{array}{lll}x_1 & x_2 & x_3\end{array}\right]\left[\begin{array}{c}3 x_1+2 x_2 \\ 2 x_1-x_2+4 x_3 \\ 4 x_2-2 x_3\end{array}\right] \\
& =x_1\left(3 x_1+2 x_2\right)+x_2\left(2 x_1-x_2+4 x_3\right)+x_3\left(4 x_2-2 x_3\right) \\
& =3 x_1^2+4 x_1 x_2-x_2^2+8 x_2 x_3-2 x_3^2
\end{aligned}$$

Fortunately, there is an easier way to calculate quadratic form. Notice that coefficients of x_i^2 is on the principal diagonal and coefficients of $x_i x_j$ are be split evenly been $(i,j)-$ and $(j,i)-$ entries in A.

Consider another example,

$$A=\left[\begin{array}{cccc}3 & 2 & 0 & 5 \\ 2 & -1 & 4 & -3 \\ 0 & 4 & -2 & -4 \\ 5 & -3 & -4 & 7\end{array}\right]$$

All x_i^2 's terms are

$$3 x_1^2-x_2^2-2 x_3^2+7 x_4^2$$

whose coefficients are from principal diagonal.

All $x_i x_j$'s terms are

$$4x_1x_2 + 0x_1x_3 + 10x_1x_4 + 8x_2x_3 - 6x_2x_4 - 8x_3x_4$$

Add up together then quadratic form is

$$3x_1^2 - x_2^2 - 2x_3^2 + 7x_4^2 + 4x_1x_2 + 0x_1x_3 + 10x_1x_4 + 8x_2x_3 - 6x_2x_4 - 8x_3x_4$$

Let's verify in SymPy.

```python
import simpy as sy

x1, x2, x3, x4 = sy.symbols('x_1 x_2 x_3 x_4')
A = sy.Matrix([[3,2,0,5],[2,-1,4,-3],[0,4,-2,-4],[5,-3,-4,7]])
x = sy.Matrix([x1, x2, x3, x4])
sy.expand(x.T*A*x)
```

Listing 1: Python example

The result is exactly the same as we derived

$$3x_1^2 - x_2^2 - 2x_3^2 + 7x_4^2 + 4x_1x_2 + 0x_1x_3 + 10x_1x_4 + 8x_2x_3 - 6x_2x_4 - 8x_3x_4$$

6 go deeper

Consider $f(x) = x^T Ax$ where $x \in \mathbf{R}^n$ and A is a square $n \times n$ matrix, and thus $f(x) \in \mathbf{R}$. Compute $df, f'(x)$, and ∇f.

We can do this directly from the definition.

$$\begin{aligned}
df &= f(x+dx) - f(x) \\
&= (x+dx)^T A(x+dx) - x^T Ax \\
&= x^T Ax + dx^T Ax + x^T Adx + dx^T Adx - x^T Ax \\
&= \underbrace{x^T \left(A + A^T\right)}_{f'(x)=(\nabla f)^T} dx \Longrightarrow \nabla f = \left(A + A^T\right) x.
\end{aligned}$$

Here, we dropped terms with more than one dx factor as these are asymptotically negligible. Another trick was to combine $dx^T Ax$ and $x^T Adx$ by realizing that these are scalars and hence equal to their own transpose: $dx^T Ax = \left(dx^T Ax\right)^T = x^T A^T dx$. Hence, we have found that $f'(x) = x^T \left(A + A^T\right) = (\nabla f)^T$, or equivalently $\nabla f = \left[x^T \left(A + A^T\right)\right]^T = \left(A + A^T\right) x$.

It is, of course, also possible to compute the same gradient component-by-component, the way you probably learned to do in multivariable calculus. First, you would need to write $f(x)$ explicitly in terms of the components of x, as $f(x) = x^T Ax = \sum_{i,j} x_i A_{i,j} x_j$. Then, you would compute $\partial f / \partial x_k$ for each k, taking care that x appears twice in the f summation. However, this approach is awkward, error-prone, labor-intensive, and quickly becomes worse as we move on to more complicated functions. It is much better, we feel, to get used to treating vectors and matrices as a whole, not as mere collections of numbers.

7 References

- Lectures of Linear Algebra https://github.com/weijie-chen/Linear-Algebra-With-Python/
- Matrix Calculus for Machine Learning and Beyond by Bright et al
- Linear Algebra and Its Applications by Gilbert Strang
- Linear Algebra and Its Applications by David Lay
- Introduction to Linear Algebra With Applications by DeFranza and Gagliardi
- Linear Algebra With Applications by Gareth Williams