Quadratic Approximation

**Review:**

**Linear approximation:**

\[ f(x,y) \approx f(x^*, y^*) + f_x(x^*, y^*)(x-x^*) \]
\[ + f_y(x^*, y^*)(y-y^*) \]

For a general \( f(\bar{x}) \), \( \bar{x} = [x_1, x_2, \ldots, x_n]^T \)

\[ f(\bar{x}) \approx f(\bar{x}^*) + \left[ D_{\bar{x}}f(\bar{x}^*) \right] (\bar{x} - \bar{x}^*) \]

\[
D_{\bar{x}}f = \begin{bmatrix}
\frac{\partial f(\bar{x})}{\partial x_1} & \cdots & \frac{\partial f(\bar{x})}{\partial x_n}
\end{bmatrix}
\]

**Taylor series for scalar valued \( f(x) \):**

\[ f(x) = \sum_{k=0}^{\infty} f^{(k)}(x^*) \frac{(x-x^*)^k}{k!} \]

**Jacobian \( D_{\bar{x}}f \) for \( \bar{f} = \begin{bmatrix} f_1(x_1, \ldots, x_n) \\
\vdots \\
f_m(x_1, \ldots, x_n) \end{bmatrix} \):**

\[
D_{\bar{x}}\bar{f} = \begin{bmatrix}
D_{\bar{x}}f_1 \\
\vdots \\
D_{\bar{x}}f_m
\end{bmatrix} = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n}
\end{bmatrix}
\]
1. Quadratic Approximation and Vector Differentiation

As shown in the previous discussion, a common way to approximate a non-linear high-dimensional functions is to perform linearization near a point. In the case of a two-dimensional function \( f(x, y) \) with scalar output, the linear approximation of \( f(x, y) \) at a point \((x_*, y_*)\) is given by

\[
f(x, y) \approx f(x_*, y_*) + f_x(x_*, y_*)(x - x_*) + f_y(x_*, y_*)(y - y_*)
\]

where as in the previous section,

\[
f_x(x_*, y_*) = \frac{\partial f(x, y)}{\partial x}\bigg|_{(x_*, y_*)} \quad \text{and} \quad f_y(x_*, y_*) = \frac{\partial f(x, y)}{\partial y}\bigg|_{(x_*, y_*)}.
\]

In vector form, this can be written as:

\[
f(\vec{x}) \approx f(\vec{x}_*) + \left[ D_{\vec{x}} f \big|_{\vec{x}_*} \right] (\vec{x} - \vec{x}_*). \tag{3}
\]

Recall from the previous discussion that \(D_{\vec{x}} f\) is a row-vector filled with the partial derivatives \(\frac{\partial f(\vec{x})}{\partial x_i}\):

\[
D_{\vec{x}} f = \begin{bmatrix}
\frac{\partial f(\vec{x})}{\partial x_1} & \cdots & \frac{\partial f(\vec{x})}{\partial x_n}
\end{bmatrix} = \begin{bmatrix}
f_{x_1}(\vec{x}) & \cdots & f_{x_n}(\vec{x})
\end{bmatrix}. \tag{4}
\]

Our goal is to extend this idea to a quadratic approximation. To do this, we need some notion of a second derivative.

For this discussion, we will only be considering these types of functions from \(\mathbb{R}^n \to \mathbb{R}\), since that is the typical form for a cost function used during optimization.

(a) Given the function \(f(\vec{x}) = e^{-2\vec{x}}\), find the first and second derivatives, and write out its quadratic approximation at \(\vec{x} = \vec{x}_*\).

*Hint: Use Taylor’s theorem.*

\[
f(\vec{x}) = e^{-2\vec{x}}
\]

\[
f'(\vec{x}) = -2e^{-2\vec{x}}
\]

\[
f''(\vec{x}) = 4e^{-2\vec{x}}
\]

**Quadratic Approx:** keep 2\textsuperscript{nd} order term in Taylor expansion

**Taylor expand** \(f(\vec{x})\) around \(\vec{x}_*\):

\[
f(\vec{x}) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\vec{x}_*)}{k!}(\vec{x} - \vec{x}_*)^k
\]

\[
= f(\vec{x}_*) + f'(\vec{x}_*)(\vec{x} - \vec{x}_*) + \frac{1}{2}f''(\vec{x}_*)(\vec{x} - \vec{x}_*)^2 + \cdots
\]
\[
\begin{aligned}
\hat{f}(x) & \approx f(X^*) + f'(X^*)(x-x^*) + \frac{1}{2} f''(X^*)(x-x^*)^2 \\
& = e^{-2x^*} - 2e^{-2x^*}(x-x^*) + 2e^{-2x^*}(x-x^*)^2
\end{aligned}
\]

(b) To write second partial derivatives compactly, we will introduce a new notation that builds off the notation \(f_x\) and \(f_y\) introduced previously. To compute \(f_{xy}\), we first take the derivative in \(x\), then in \(y\):

\[
f_{xy}(x_*, y_*) = \frac{\partial f_x(x, y)}{\partial y} \bigg|_{(x_*, y_*)} = \frac{\partial^2 f(x, y)}{\partial y \partial x} \bigg|_{(x_*, y_*)}.
\] (5)

Given the function \(f(x, y) = x^2y^2\), find all of the first and second partial derivatives.

\[
\begin{align*}
\text{1st order derivatives: } & f_x, \ f_y \\
\text{2nd order derivatives: } & f_{xx}, \ f_{xy}, \ f_{yx}, \ f_{yy}
\end{align*}
\]

\[
\begin{aligned}
f_x(x, y) &= \frac{\partial f(x, y)}{\partial x} = 2xy^2 \\
f_y(x, y) &= \frac{\partial f(x, y)}{\partial y} = 2x^2y
\end{aligned}
\]

\[
\begin{aligned}
f_{xx}(x, y) &= \frac{\partial f_x(x, y)}{\partial x} = \frac{\partial}{\partial x}(2xy^2) = 2y \\
f_{xy}(x, y) &= \frac{\partial f_x(x, y)}{\partial y} = \frac{\partial}{\partial y}(2xy^2) = 4xy \\
f_{yx}(x, y) &= \frac{\partial f_y(x, y)}{\partial x} = \frac{\partial}{\partial x}(2x^2y) = 4xy \\
f_{yy}(x, y) &= \frac{\partial f_y(x, y)}{\partial y} = \frac{\partial}{\partial y}(2x^2y) = 2x^2
\end{aligned}
\]
(c) To find the quadratic approximation of $f(x, y)$ near $(x_*, y_*)$, we plug in $f(x_* + \Delta x, y_* + \Delta y)$ and drop the terms that are higher order than quadratic:

$$f(x_* + \Delta x, y_* + \Delta y) = (x_* + \Delta x)^2(y_* + \Delta y)^2$$

$$= (x_*^2 + 2x_*\Delta x + (\Delta x)^2)(y_*^2 + 2y_*\Delta y + (\Delta y)^2)$$

$$\approx x_*^2y_*^2 + 2x_*y_*\Delta x + 2x_*^2y_*\Delta y + y_*^2(\Delta x)^2 + 4x_*y_*(\Delta x)(\Delta y) + x_*^2(\Delta y)^2$$

$$= f(x_*, y_*) + f_x(x_*, y_*)\Delta x + f_y(x_*, y_*)\Delta y + \frac{1}{2}f_{xx}(x_*, y_*)(\Delta x)^2 + \frac{1}{2}f_{yy}(x_*, y_*)(\Delta y)^2$$

$$+ \frac{1}{2}f_{xy}(x_*, y_*)(\Delta x)(\Delta y)$$

This is slightly different from the expression we get via the Taylor series expansion. How would we rewrite this expression, so that all second derivatives are involved, each with a coefficient of $\frac{1}{2}$?

**Note:** $f_{xy} = f_{yx}$

$$f_{xy}(x, y) = \frac{1}{2}f_{xy}(x, y) + \frac{1}{2}f_{yx}(x, y)$$

**Plug back in:**

$$f(x + \Delta x, y + \Delta y) = f(x_*, y_*)$$

$$+ f_x(x_*, y_*)\Delta x + f_y(x_*, y_*)\Delta y$$

$$+ \frac{1}{2}f_{xx}(x_*, y_*)(\Delta x)^2 + \frac{1}{2}f_{yy}(x_*, y_*)(\Delta y)^2$$

$$+ \frac{1}{2}f_{xy}(x_*, y_*)(\Delta x)(\Delta y)$$
(d) Just as we created the derivative row vector to hold all the first partial derivatives to help in writing linearization in matrix/vector form:

\[
D_{\bar{x}} f = \begin{bmatrix}
\frac{\partial f(\bar{x})}{\partial x_1} & \ldots & \frac{\partial f(\bar{x})}{\partial x_n}
\end{bmatrix}
\begin{bmatrix}
f_{x_1}(\bar{x}) \\
\vdots \\
f_{x_n}(\bar{x})
\end{bmatrix}
\]

(13)

we would like to create a matrix to hold all the second partial derivatives to help in writing quadratic approximation in matrix/vector form:

\[
H_{\bar{x}} f = \begin{bmatrix}
\frac{\partial^2 f(\bar{x})}{\partial x_1^2} & \ldots & \frac{\partial^2 f(\bar{x})}{\partial x_1 \partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial^2 f(\bar{x})}{\partial x_n \partial x_1} & \ldots & \frac{\partial^2 f(\bar{x})}{\partial x_n^2}
\end{bmatrix}
\begin{bmatrix}
f_{x_1 x_1}(\bar{x}) \\
\vdots \\
f_{x_n x_n}(\bar{x})
\end{bmatrix}
\]

(14)

This matrix is the Hessian of \( f \). Note that this quantity is different from the Jacobian matrix that was covered in the previous discussion. In contrast to the Hessian, which is the matrix of second partial derivatives of a scalar-valued vector-input function \( f: \mathbb{R}^n \to \mathbb{R} \), the Jacobian is the matrix of first partial derivatives of a vector-valued vector-input function \( \tilde{f}: \mathbb{R}^n \to \mathbb{R}^k \).

In fact, the Hessian is the (Jacobian) derivative of the derivative; if we let \( \tilde{g}(\bar{x}) = (D_{\bar{x}} f)^\top \) (so that it’s a column vector and the dimensions work out), then \( H_{\bar{x}} f = D_{\bar{x}} \tilde{g} \).

To get a feel for the Hessian of \( f \), find \( H_{(x,y)} f \) for the \( f \) above, that is, \( f(x,y) = x^2 y^3 \).

\[
H_{(x,y)} f = \begin{bmatrix}
\frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\
\frac{\partial^2 f(x,y)}{\partial y \partial x} & \frac{\partial^2 f(x,y)}{\partial y^2}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
f_{xx} & f_{xy} \\
f_{yx} & f_{yy}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
2y^2 & 4xy \\
4xy & 2x^2
\end{bmatrix}
\]

(e) Using the Hessian, write out the general formula for the quadratic approximation of a scalar-valued function \( f \) of a vector \( \bar{x} \) in vector/matrix form.
1st order approximation:

\[ f(\bar{x}_* + \Delta \bar{x}) \approx f(\bar{x}_*) + \left[ D_{\bar{x}} f \bigg|_{\bar{x}_*} \right] (\Delta \bar{x}) \]

2nd order approximation:

\[ f(\bar{x}_* + \Delta \bar{x}) \approx f(\bar{x}_*) + \left[ D_{\bar{x}} f \bigg|_{\bar{x}_*} \right] (\Delta \bar{x}) + \frac{1}{2} (\Delta \bar{x})^T \left[ H_{\bar{x}} f \bigg|_{\bar{x}_*} \right] (\Delta \bar{x}) \]

(f) Show that the quadratic approximation for the scalar-valued function \( f(\bar{w}) = \bar{x}^T \bar{w} \) around \( \bar{w} = \bar{w}_* \) is

\[ f(\bar{w}_* + \Delta \bar{w}) \approx \bar{x}^T \bar{w}_* \left( 1 + \bar{x}^T (\Delta \bar{w}) + \frac{1}{2} \left( \bar{x}^T (\Delta \bar{w}) \right)^2 \right). \]  

Here, assume that \( \bar{x} \) is just some given vector — a constant vector.

*Hint:* You can compute the following partial derivatives:

\[ f_{\bar{w}_i}(\bar{w}) = x_i f(\bar{w}) \]  

\[ f_{\bar{w}_i \bar{w}_j}(\bar{w}) = x_i x_j f(\bar{w}). \]

Now compute \( D_{\bar{w}} f \) and \( H_{\bar{w}} f \), and plug it into the quadratic approximation formula.
\[ f_{w_i}(\bar{w}) = \frac{\partial}{\partial w_i} e^{\bar{x}^T \bar{w}} = x_i e^{\bar{x}^T \bar{w}} = x_i f(\bar{w}) \]

\[ f_{w_i} f_{w_j} = \frac{\partial f_{w_i}(\bar{w})}{\partial w_j} = \frac{\partial}{\partial w_j} (x_i f(\bar{w})) = x_i x_j f(\bar{w}) \]

\[ Df = \begin{bmatrix} f_{w_1}(\bar{w}) & \cdots & f_{w_n}(\bar{w}) \end{bmatrix} \]

\[ = f(\bar{w}) \bar{X}^T \]

\[ H_{w_i} f = \begin{bmatrix} f_{w_1}(\bar{w}) & \cdots & f_{w_n}(\bar{w}) \\ \vdots \\ f_{w_m}(\bar{w}) & f_{w_n}(\bar{w}) \end{bmatrix} \]

\[ = f(\bar{w}) \bar{X} \bar{X}^T \]

\[ f(\bar{w} + \Delta \bar{w}) \approx f(\bar{w}) + Df|_{\bar{w}}(\Delta \bar{w}) + \frac{1}{2} (\Delta \bar{w})^T [H_{w_i} f|_{\bar{w}}] (\Delta \bar{w}) \]

\[ = e^{\bar{x}^T \bar{w}} \left( 1 + \bar{x}^T (\Delta \bar{w}) + \frac{1}{2} (\bar{x}^T (\Delta \bar{w}))^2 \right) \]
(g) Use linearity to give the quadratic approximation for the function \( \sum_{i=1}^{n} e^{x_i^T \Delta \vec{w}} \) around \( \vec{w} = \vec{w}_* \). Here, assume that the \( \vec{x}_i \) are just some given vectors.

\[
f(\vec{w}_* + \Delta \vec{w}) \approx \sum_{i=1}^{n} e^{x_i^T \Delta \vec{w}} \left( 1 + x_i (\Delta \vec{w}) + \frac{1}{2} (x_i^T (\Delta \vec{w}))^2 \right)
\]

(h) **Practice.** The second derivative also has an interpretation as the derivative of the derivative. However, we saw that the derivative of a scalar-valued function with respect to a vector is naturally a row. If you wanted to approximate how much the derivative changed by moving a small amount \( \Delta \vec{w} \), how would you get such an estimate using your expression for the second derivative?

Linearly approximate scalar-valued \( f \):

\[
f(\vec{w}_* + \Delta \vec{w}) \approx f(\vec{w}_*) + f'(\vec{w}_*) \Delta \vec{w}
\]

Let \( f'(\vec{w}_* + \Delta \vec{w}) \) to be "f":

\[
f'(\vec{w}_* + \Delta \vec{w}) \approx f'(\vec{w}_*) + f''(\vec{w}_*) \Delta \vec{w}
\]

but now "f" is \( \Delta \vec{w} f \)

\[
\Delta \vec{w} f \approx \Delta \vec{w} f |_{\vec{w}_*} + ?
\]

\[
\sim \frac{1}{x_i} + \frac{1}{x_i}
\]
\[ Df = D_\omega f|_{\omega} + (\omega)^T [H_\omega f|_{\omega}] \]