Faculty of Science

L2 – Linear Regression II

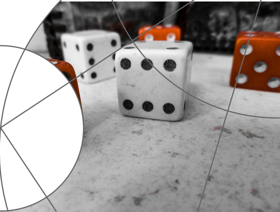
Modelling and Analysis of Data

Fabian Gieseke

Machine Learning Section
Department of Computer Science
University of Copenhagen

Universitetsparken 1, 1-1-N110
fabian.gieseke@di.ku.dk

21 November 2019



Outline

Outline

Quiz Time!

Proof II

- The partial derivatives are given by:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_0} &= 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) \\
\frac{\partial \mathcal{L}}{\partial w_1} &= 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} {x_n}^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)
\end{aligned}
$$

Proof II

- The partial derivatives are given by:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial w_0} &= 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) \\
\frac{\partial \mathcal{L}}{\partial w_1} &= 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} {x_n}^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)
\end{aligned}
$$

- Task: Derive the Hessian matrix!

$$
\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} & \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \\ \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_0} & \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} \end{bmatrix}
$$

Proof II

- The partial derivatives are given by:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_0} &= 2w_0 + 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} x_n\right) - \frac{2}{N}\left(\sum_{n=1}^{N} t_n\right) \\ \frac{\partial \mathcal{L}}{\partial w_1} &= 2w_1 \frac{1}{N}\left(\sum_{n=1}^{N} {x_n}^2\right) + \frac{2}{N}\left(\sum_{n=1}^{N} x_n(w_0 - t_n)\right)\end{aligned}$$

- Task: Derive the Hessian matrix!

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} & \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \\ \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_0} & \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} \end{bmatrix}$$

- The Hessian is given by

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} & \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \\ \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_0} & \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} \end{bmatrix} = \begin{bmatrix} 2 & \frac{2}{N}\left(\sum_{n=1}^{N} x_n\right) \\ \frac{2}{N}\left(\sum_{n=1}^{N} x_n\right) & \frac{2}{N}\left(\sum_{n=1}^{N} {x_n}^2\right) \end{bmatrix}$$

Proof II

- We have

$$
\begin{aligned}
D &= \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} - \left(\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \right)^2 \\
&= 4 \cdot \left(\frac{1}{N} \sum_{n=1}^{N} {x_n}^2 \right) - 4 \left(\frac{1}{N} \sum_{n=1}^{N} x_n \right)^2 \\
&= 4 \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^2 > 0
\end{aligned}
$$

where we assumed that not all the x_n are the same.

Proof II

- We have

$$
\begin{aligned}
D &= \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} - \left(\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \right)^2 \\
&= 4 \cdot \left(\frac{1}{N} \sum_{n=1}^{N} {x_n}^2 \right) - 4 \left(\frac{1}{N} \sum_{n=1}^{N} x_n \right)^2 \\
&= 4 \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^2 > 0
\end{aligned}
$$

  where we assumed that not all the x_n are the same.

- We also have $\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} > 0$

Proof II

- We have

$$
\begin{aligned}
D &= \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} - \left(\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \right)^2 \\
&= 4 \cdot \left(\frac{1}{N} \sum_{n=1}^{N} {x_n}^2 \right) - 4 \left(\frac{1}{N} \sum_{n=1}^{N} x_n \right)^2 \\
&= 4 \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^2 > 0
\end{aligned}
$$

  where we assumed that not all the x_n are the same.
- We also have $\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} > 0$
- Thus, according to the second derivative test, we have a local minimum at the critical point $(\hat{w}_0, \hat{w}_1)$!

Proof II

- We have

$$\begin{aligned} D &= \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} - \left(\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \right)^2 \\ &= 4 \cdot \left(\frac{1}{N} \sum_{n=1}^{N} {x_n}^2 \right) - 4 \left(\frac{1}{N} \sum_{n=1}^{N} x_n \right)^2 \\ &= 4 \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^2 > 0 \end{aligned}$$

  where we assumed that not all the x_n are the same.
- We also have $\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} > 0$
- Thus, according to the second derivative test, we have a local minimum at the critical point $(\hat{w}_0, \hat{w}_1)$!
- Question: Are we done?

Proof II

- We have

$$\begin{aligned} D &= \frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} \frac{\partial^2 \mathcal{L}}{\partial w_1 \partial w_1} - \left(\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_1} \right)^2 \\ &= 4 \cdot \left(\frac{1}{N} \sum_{n=1}^{N} {x_n}^2 \right) - 4 \left(\frac{1}{N} \sum_{n=1}^{N} x_n \right)^2 \\ &= 4 \frac{1}{N} \sum_{n=1}^{N} (x_n - \bar{x})^2 > 0 \end{aligned}$$

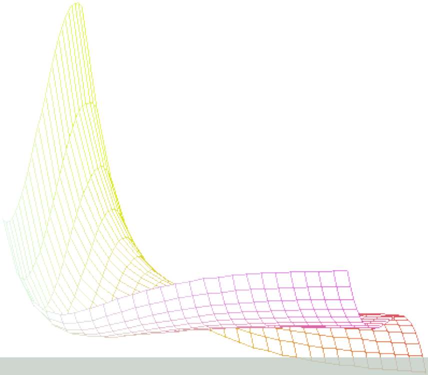
  where we assumed that not all the x_n are the same.
- We also have $\frac{\partial^2 \mathcal{L}}{\partial w_0 \partial w_0} > 0$
- Thus, according to the second derivative test, we have a local minimum at the critical point $(\hat{w}_0, \hat{w}_1)$!
- Question: Are we done?

No! We have not yet shown that it is also a global minimum!

Proof II

Question: Local minimum and only one critical point.
Is the local minimum also a global minimum?

Proof II



Example

Consider the function $f(x,y) = e^{3x} + y^3 - 3ye^x$. We have

1. $f_x = 3e^{3x} - 3ye^x$ and
2. $f_y = 3y^2 - 3e^x$.

Setting the derivatives to zero yields $(0,1)$ as only critical point. The second derivatives test shows that this is a local minimum. However, $f(0,-3) = -17 < f(0,1) = -1$.

For details, see `http://www.math.tamu.edu/~tvogel/gallery/node16.html`.

Outline

Linear Regression Using Matrix Notation

- We have: $f(x; w_0, w_1) = f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{x}^T \boldsymbol{w}$ with $\boldsymbol{x} = [1, x]^T$ and $\boldsymbol{w} = [w_0, w_1]^T$

Linear Regression Using Matrix Notation

- We have: $f(x; w_0, w_1) = f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{x}^T \boldsymbol{w}$ with $\boldsymbol{x} = [1, x]^T$ and $\boldsymbol{w} = [w_0, w_1]^T$
- Let's "augment" all data points $x_1, x_2, \ldots, x_N$. This yields an augmented data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times 2}$ and an associated target vector $\boldsymbol{t} \in \mathbb{R}^N$:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{bmatrix} \quad \text{and} \quad \boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

Linear Regression Using Matrix Notation

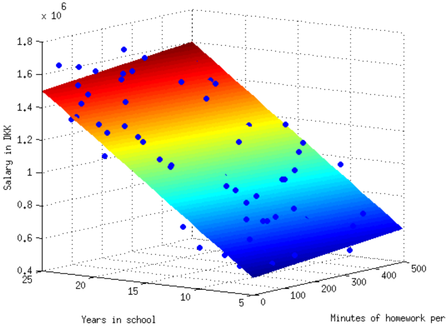
- We have: $f(x; w_0, w_1) = f(\boldsymbol{x}; \boldsymbol{w}) = \boldsymbol{x}^T \boldsymbol{w}$ with $\boldsymbol{x} = [1, x]^T$ and $\boldsymbol{w} = [w_0, w_1]^T$
- Let's "augment" all data points $x_1, x_2, \ldots, x_N$. This yields an augmented data matrix $\boldsymbol{X} \in \mathbb{R}^{N\times 2}$ and an associated target vector $\boldsymbol{t} \in \mathbb{R}^N$:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{bmatrix} \quad \text{and} \quad \boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

- Then, we can write the overall loss as:

$$\mathcal{L}(w_0, w_1) = \frac{1}{N} \sum_{n=1}^{N} \left((w_0 + x_n w_1) - t_n\right)^2 = \frac{1}{N} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

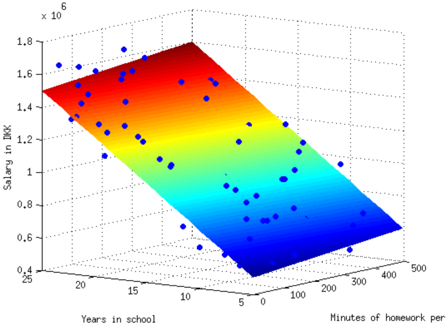
Multivariate Linear Regression



General Form

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Goal: Linear model $f(\boldsymbol{x}; \boldsymbol{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$

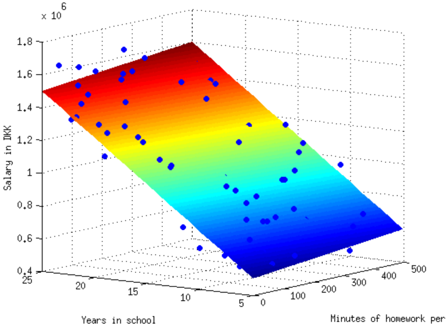
Multivariate Linear Regression



General Form

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Goal: Linear model $f(\boldsymbol{x}; \boldsymbol{w}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_D x_D$
  - Don't get confused :-): $\mathbf{x}_i \in \mathbb{R}^D$ and $x_i \in \mathbb{R}$

Multivariate Linear Regression



General Form

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Goal: Linear model $f(\boldsymbol{z}; \boldsymbol{w}) = w_0 + w_1 z_1 + w_2 z_2 + \ldots + w_D z_D$
  - Don't get confused :-): $\mathbf{x}_n \in \mathbb{R}^D$ and $x_n \in \mathbb{R}$
  - If you like: Replace x_n by z_n ...

Multivariate Linear Regression

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Let's "augment" all data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. This yields an augmented data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times (D+1)}$ and an associated target vector $\boldsymbol{t} \in \mathbb{R}^N$:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,D} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,D} \\ \vdots & & & & \\ 1 & x_{N,1} & x_{N,2} & \ldots & x_{N,D} \end{bmatrix} \quad \text{and} \quad \boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

Multivariate Linear Regression

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Let's "augment" all data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. This yields an augmented data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times (D+1)}$ and an associated target vector $\boldsymbol{t} \in \mathbb{R}^N$:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,D} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,D} \\ \vdots & & & & \\ 1 & x_{N,1} & x_{N,2} & \ldots & x_{N,D} \end{bmatrix} \quad \text{and} \quad \boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

- As before, we can write the overall loss in the following form:

Overall Loss

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} (f(\mathbf{x}_n; \mathbf{w}) - t_n)^2 = \frac{1}{N} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

Multivariate Linear Regression

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Let's "augment" all data points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. This yields an augmented data matrix $\boldsymbol{X} \in \mathbb{R}^{N \times (D+1)}$ and an associated target vector $\boldsymbol{t} \in \mathbb{R}^N$:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \ldots & x_{1,D} \\ 1 & x_{2,1} & x_{2,2} & \ldots & x_{2,D} \\ \vdots & & & & \\ 1 & x_{N,1} & x_{N,2} & \ldots & x_{N,D} \end{bmatrix} \quad \text{and} \quad \boldsymbol{t} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix}$$

- As before, we can write the overall loss in the following form:

Overall Loss

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N} \sum_{n=1}^{N} \left(f(\mathbf{x}_n; \mathbf{w}) - t_n \right)^2 = \frac{1}{N} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

How can we obtain the optimal $(D+1)$-dimensional coefficient vector $\boldsymbol{w}$?

Simplifying the Objective

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(f(x_n;\boldsymbol{w}) - t_n)^2 = \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

$$\mathcal{L}(\boldsymbol{w}) \quad = \quad \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

Simplifying the Objective

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(f(x_n;\boldsymbol{w}) - t_n)^2 = \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}) &= \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}) \\ &= \frac{1}{N}((\boldsymbol{X}\boldsymbol{w})^T - \boldsymbol{t}^T)(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})\end{aligned}$$

Simplifying the Objective

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(f(x_n;\boldsymbol{w}) - t_n)^2 = \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}) &= \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}) \\
&= \frac{1}{N}((\boldsymbol{X}\boldsymbol{w})^T - \boldsymbol{t}^T)(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}) \\
&= \frac{1}{N}(\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{X}\boldsymbol{w} - \frac{1}{N}\boldsymbol{t}^T\boldsymbol{X}\boldsymbol{w} - \frac{1}{N}(\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}
\end{aligned}$$

Simplifying the Objective

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(f(x_n;\boldsymbol{w}) - t_n)^2 = \frac{1}{N}(\boldsymbol{Xw} - \boldsymbol{t})^T(\boldsymbol{Xw} - \boldsymbol{t})$$

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}) &= \frac{1}{N}(\boldsymbol{Xw} - \boldsymbol{t})^T(\boldsymbol{Xw} - \boldsymbol{t}) \\
&= \frac{1}{N}((\boldsymbol{Xw})^T - \boldsymbol{t}^T)(\boldsymbol{Xw} - \boldsymbol{t}) \\
&= \frac{1}{N}(\boldsymbol{Xw})^T\boldsymbol{Xw} - \frac{1}{N}\boldsymbol{t}^T\boldsymbol{Xw} - \frac{1}{N}(\boldsymbol{Xw})^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t} \\
&= \frac{1}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{Xw} - \frac{2}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}
\end{aligned}$$

Simplifying the Objective

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}(f(x_n;\boldsymbol{w}) - t_n)^2 = \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$$

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}) &= \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}) \\
&= \frac{1}{N}((\boldsymbol{X}\boldsymbol{w})^T - \boldsymbol{t}^T)(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t}) \\
&= \frac{1}{N}(\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{X}\boldsymbol{w} - \frac{1}{N}\boldsymbol{t}^T\boldsymbol{X}\boldsymbol{w} - \frac{1}{N}(\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t} \\
&= \frac{1}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}
\end{aligned}$$

with $\boldsymbol{t}^T\boldsymbol{X}\boldsymbol{w} = ((\boldsymbol{X}\boldsymbol{w})^T\boldsymbol{t})^T \in \mathbb{R}$.

Gradient & Stationary Point

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}$$

Gradient & Stationary Point

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}$$

Toolbox (Table 1.4 in Rogers & Girolami)

1. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{x} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
2. $f(\mathbf{w}) = \boldsymbol{x}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
3. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{w}$
4. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{C}\boldsymbol{w}$ (if C is symmetric)

Gradient & Stationary Point

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}$$

Toolbox (Table 1.4 in Rogers & Girolami)

1. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{x} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
2. $f(\mathbf{w}) = \boldsymbol{x}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
3. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{w}$
4. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{C}\boldsymbol{w}$ (if C is symmetric)

Task: Derive the gradient for $\mathcal{L}(\boldsymbol{w})$

Gradient & Stationary Point

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}$$

Toolbox (Table 1.4 in Rogers & Girolami)

1. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{x} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
2. $f(\mathbf{w}) = \boldsymbol{x}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
3. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{w}$
4. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{C}\boldsymbol{w}$ (if C is symmetric)

The gradient is given by $\nabla\mathcal{L}(\mathbf{w}) = \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t}$.

Gradient & Stationary Point

$$\mathcal{L}(\mathbf{w}) = \frac{1}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{t} + \frac{1}{N}\boldsymbol{t}^T\boldsymbol{t}$$

Toolbox (Table 1.4 in Rogers & Girolami)

1. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{x} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
2. $f(\mathbf{w}) = \boldsymbol{x}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = \boldsymbol{x}$
3. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{w}$
4. $f(\mathbf{w}) = \boldsymbol{w}^T\boldsymbol{C}\boldsymbol{w} \Rightarrow \nabla f(\boldsymbol{w}) = 2\boldsymbol{C}\boldsymbol{w}$ (if C is symmetric)

The gradient is given by $\nabla\mathcal{L}(\mathbf{w}) = \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t}$. Therefore:

$$\begin{aligned} & \nabla\mathcal{L}(\mathbf{w}) &&= \mathbf{0} \\ \Leftrightarrow\quad & \tfrac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \tfrac{2}{N}\boldsymbol{X}^T\boldsymbol{t} &&= \mathbf{0} \\ \Leftrightarrow\quad & \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} &&= \boldsymbol{X}^T\boldsymbol{t} \end{aligned}$$

Gradient & Stationary Point

The gradient is given by $\nabla \mathcal{L}(\mathbf{w}) = \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t}$. Therefore:

$$
\begin{aligned}
& \nabla \mathcal{L}(\mathbf{w}) && = \mathbf{0} \\
\Leftrightarrow \quad & \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t} && = \mathbf{0} \\
\Leftrightarrow \quad & \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} && = \boldsymbol{X}^T\boldsymbol{t}
\end{aligned}
$$

Gradient & Stationary Point

The gradient is given by $\nabla \mathcal{L}(\mathbf{w}) = \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t}$. Therefore:

$$\begin{aligned} & \nabla \mathcal{L}(\mathbf{w}) && = \mathbf{0} \\ \Leftrightarrow \quad & \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t} && = \mathbf{0} \\ \Leftrightarrow \quad & \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} && = \boldsymbol{X}^T\boldsymbol{t} \end{aligned}$$

Finally, let's multiply both sides (from left) with $\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}$. This yields

$$\boldsymbol{I}\mathbf{w} \quad = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{t}$$

where $\boldsymbol{I} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{X}\right)$ is the identity matrix.

Gradient & Stationary Point

The gradient is given by $\nabla\mathcal{L}(\mathbf{w}) = \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t}$. Therefore:

$$\begin{aligned} & \nabla\mathcal{L}(\mathbf{w}) && = \mathbf{0} \\ \Leftrightarrow \quad & \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t} && = \mathbf{0} \\ \Leftrightarrow \quad & \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} && = \boldsymbol{X}^T\boldsymbol{t} \end{aligned}$$

Finally, let's multiply both sides (from left) with $\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}$. This yields

$$\boldsymbol{I}\mathbf{w} \quad = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{t}$$

where $\boldsymbol{I} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{X}\right)$ is the identity matrix. This yields:

Minimizer for Linear Regression

$$\hat{\mathbf{w}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{t}$$

Gradient & Stationary Point

The gradient is given by $\nabla\mathcal{L}(\mathbf{w}) = \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t}$. Therefore:

$$\begin{aligned} & \nabla\mathcal{L}(\mathbf{w}) && = \mathbf{0} \\ \Leftrightarrow \quad & \frac{2}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \frac{2}{N}\boldsymbol{X}^T\boldsymbol{t} && = \mathbf{0} \\ \Leftrightarrow \quad & \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} && = \boldsymbol{X}^T\boldsymbol{t} \end{aligned}$$

Finally, let's multiply both sides (from left) with $\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}$. This yields

$$\boldsymbol{I}\mathbf{w} \quad = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{t}$$

where $\boldsymbol{I} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{X}\right)$ is the identity matrix. This yields:

Minimizer for Linear Regression

$$\hat{\mathbf{w}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{t}$$

Note: There is no guarantee that the matrix $\left(\boldsymbol{X}^T\boldsymbol{X}\right)$ is invertible.

Predictions?

Let $\mathbf{x}_{new} \in \mathbb{R}^D$ be a new point. How can we compute the predicted value?

1. Prepend a one: $[1, \mathbf{x}_{new}^T]$
2. Compute $t_{new} = [1, \mathbf{x}_{new}^T]\hat{\mathbf{w}}$

Quiz Time!

Summary: Multivariate Linear Regression

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Goal: Find $(D+1)$-dimensional weight vector $\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_D]^T$ that minimizes $\mathcal{L}(\mathbf{w}) = \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$, i.e., which is a solution for

$$
\begin{aligned}
\nabla \mathcal{L}(\mathbf{w}) &= \mathbf{0} \\
\Leftrightarrow \quad \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} &= \boldsymbol{X}^T\boldsymbol{t}
\end{aligned} \tag{1}
$$

Summary: Multivariate Linear Regression

- Given: Pairs of the form $(\mathbf{x}_1, t_1), \ldots, (\mathbf{x}_N, t_N) \in \mathbb{R}^D \times \mathbb{R}$.
- Goal: Find $(D+1)$-dimensional weight vector $\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_D]^T$ that minimizes $\mathcal{L}(\mathbf{w}) = \frac{1}{N}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})^T(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{t})$, i.e., which is a solution for

$$\begin{aligned} \nabla\mathcal{L}(\mathbf{w}) &= \mathbf{0} \\ \Leftrightarrow \quad \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} &= \boldsymbol{X}^T\boldsymbol{t} \end{aligned} \tag{1}$$

Computation in Practice

1. Definition of data matrix $\mathbf{X} \in \mathbb{R}^{N\times(D+1)}$
   (make use of Numpy arrays and functions!)
2. There are different ways to compute an optimal weight vector $\hat{\mathbf{w}}$:
   1. Compute $\hat{\mathbf{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{t}$ (e.g., via `numpy.linalg.inv`)
   2. Directly solve the system of equations (1) (e.g., via `numpy.linalg.solve`)
   3. ...
3. For new point $\mathbf{x}_{new} \in \mathbb{R}^D$: Compute $t_{new} = [1, \mathbf{x}_{new}^T]\hat{\mathbf{w}}$

Outline

1. Recap + Proof II

2. Multivariate Case

3. Implementation

4. Summary & Outlook

Coding Time!

jupyter Multivariate Linear Regression Last Checkpoint: a few seconds ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

copy selected cells

In [ ]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy
```

We shall work with the dataset found in the file 'murderdata.txt', which is a 20 x 5 data matrix where the columns correspond to

Index (not for use in analysis)

Number of inhabitants

Percent with incomes below $5000

Percent unemployed

Murders per annum per 1,000,000 inhabitants

Reference:

Helmut Spaeth, Mathematical Algorithms for Linear Regression, Academic Press, 1991, ISBN 0-12-656460-4.

D G Kleinbaum and L L Kupper, Applied Regression Analysis and Other Multivariable Methods, Duxbury Press, 1978, page 150.

http://people.sc.fsu.edu/~jburkardt/datasets/regression

What to do?

We start by loading the data; today we will study how the number of murders relates to the percentage of unemployment.

In [ ]:

```
data = numpy.loadtxt('murderdata.txt')
N, d = data.shape
```

We consider all both features simulaneously.

In [ ]:

```
t = data[:,4]
X = data[:,2:4]
print("Number of training instances: %i" % X.shape[0])
print("Number of features: %i" % X.shape[1])
```

In [ ]:

```
# NOTE: This template makes use of Python classes. If
# you are not yet familiar with this concept, you can
# find a short introduction here:
# http://introtopython.org/classes.html

class LinearRegression():
    """
    Linear regression implementation.
    """

    def __init__(self):
```

Coding Time!

Computation in Practice

1. Definition of data matrix $\mathbf{X} \in \mathbb{R}^{N\times(D+1)}$
   (make use of Numpy arrays and functions!)
2. There are different ways to compute an optimal weight vector $\hat{\mathbf{w}}$:
   1. Compute $\hat{\mathbf{w}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{t}$ (e.g., via `numpy.linalg.inv`)
   2. Directly solve the system of equations (e.g., via `numpy.linalg.solve`):

$$\begin{aligned} \nabla\mathcal{L}(\mathbf{w}) &= \mathbf{0} \\ \Leftrightarrow \quad \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} &= \boldsymbol{X}^T\boldsymbol{t} \end{aligned}$$

   3. ...
3. For new point $\mathbf{x}_{new} \in \mathbb{R}^D$: Compute $t_{new} = [1, \mathbf{x}_{new}^T]\hat{\mathbf{w}}$

```
In [ ]: t = data[:,4]
        X = data[:,2:4]
        print("Number of training instances: %i" % X.shape[0])
        print("Number of features: %i" % X.shape[1])
```

```
In [ ]: # NOTE: This template makes use of Python classes. If
        # you are not yet familiar with this concept, you can
        # find a short introduction here:
        # http://introtopython.org/classes.html

        class LinearRegression():
            """
            Linear regression implementation.
            """

            def __init__(self):
```

Outline

1. Recap + Proof II

2. Multivariate Case

3. Implementation

4. Summary & Outlook

Summary & Outlook

Today

- Seen how to derive the multi-dimensional linear regression model.
- Seen how to implement these things in Numpy!

Outlook

- Learn about “nonlinear” regression using linear models
  (next Tuesday)
- Learn about a probabilistic interpretation of the least squares loss
  (later, Kim)
- Learn about a Bayesian approach to regression, resulting in a regression model with uncertainty
  (later, Kim)