EECS 395/495 Machine Learning

Aggelos K. Katsaggelos

Joseph Cummings Professor
Northwestern University
Department of EECS
Department of Linguistics
Argonne National Laboratory
NorthShore University Health System
Evanston, IL 60208
http://ivpl.eecs.northwestern.edu

iVPL
image & video processing lab

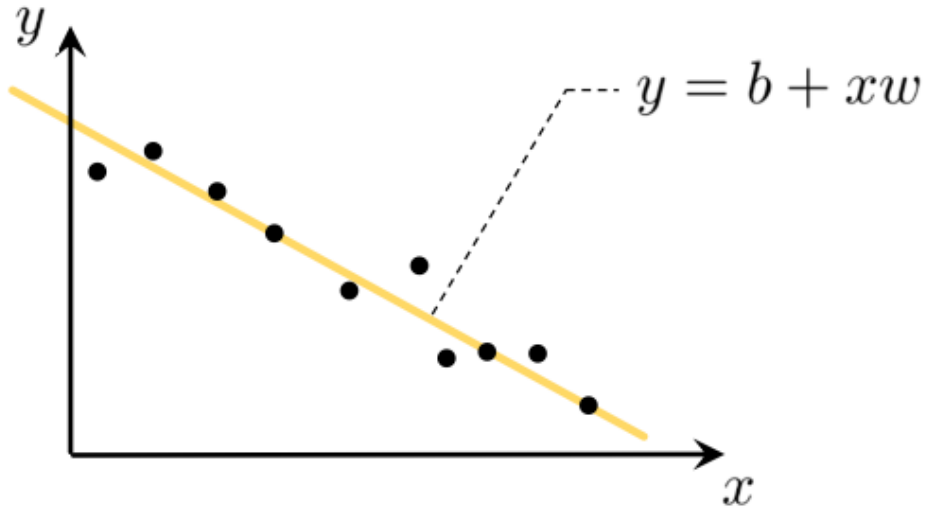
Linear Regression

Linear regression

Data: $\{(\mathbf{x}_p, y_p)\}_{p=1}^{P}$

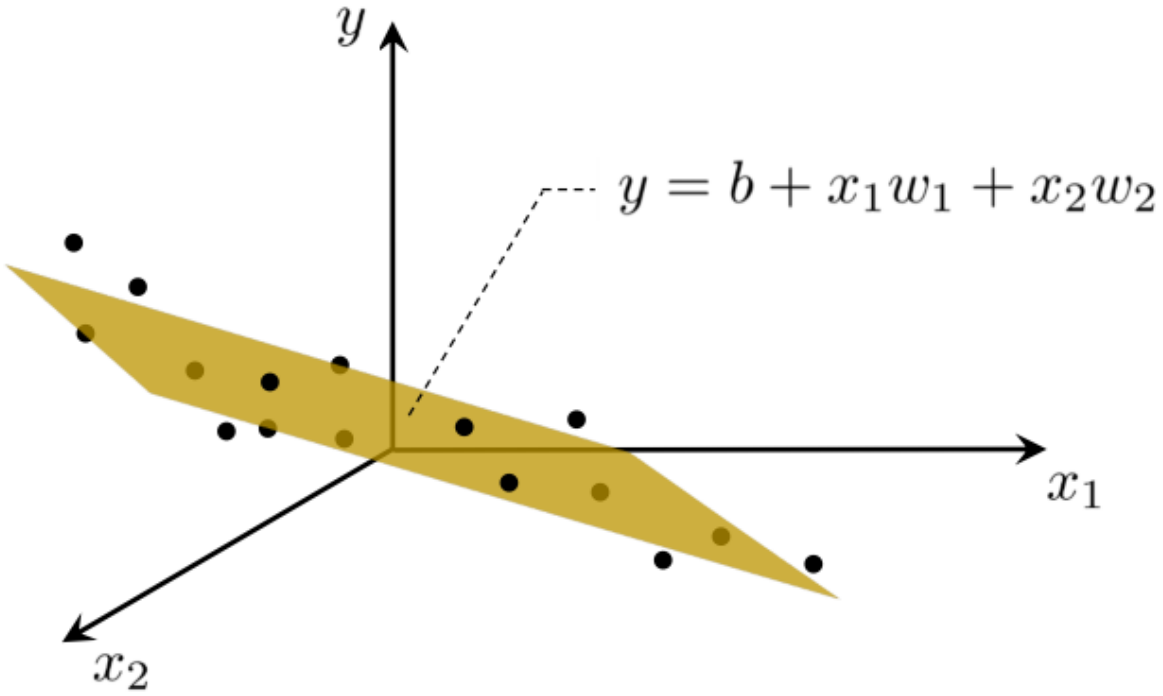
Output: continuous scalar

Input: N-dimensional (here $N = 1, 2$)



Want to find b and $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}$ such that

$$b + \mathbf{x}_p^T \mathbf{w} \approx y_p, \quad p = 1, ..., P$$



The Least Squares solution

$$b + \mathbf{x}_p^T \mathbf{w} \approx y_p, \quad p = 1, ..., P$$

$$\Downarrow$$

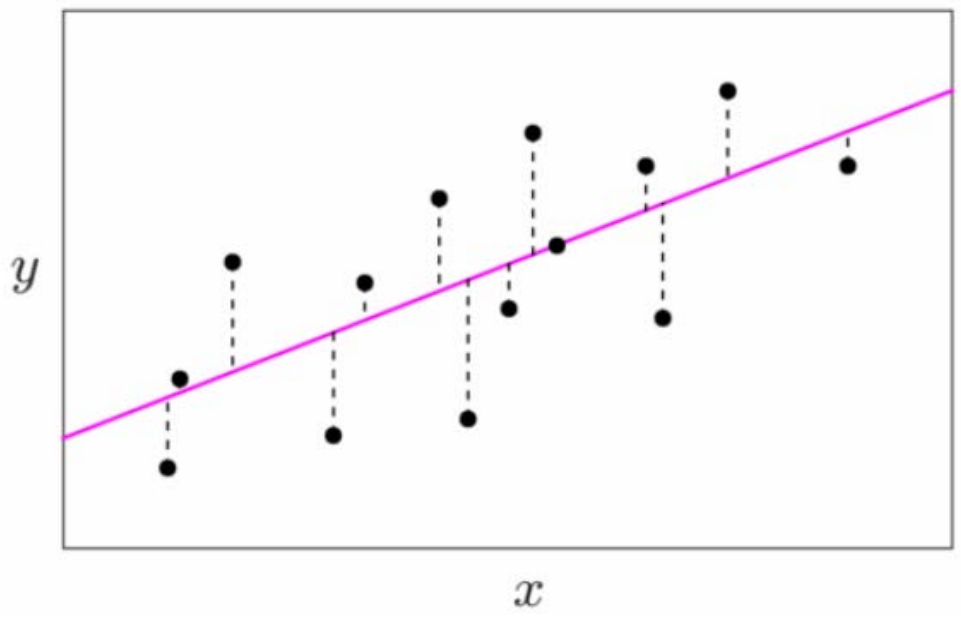
$$\left(b + \mathbf{x}_p^T \mathbf{w} - y_p\right)^2$$

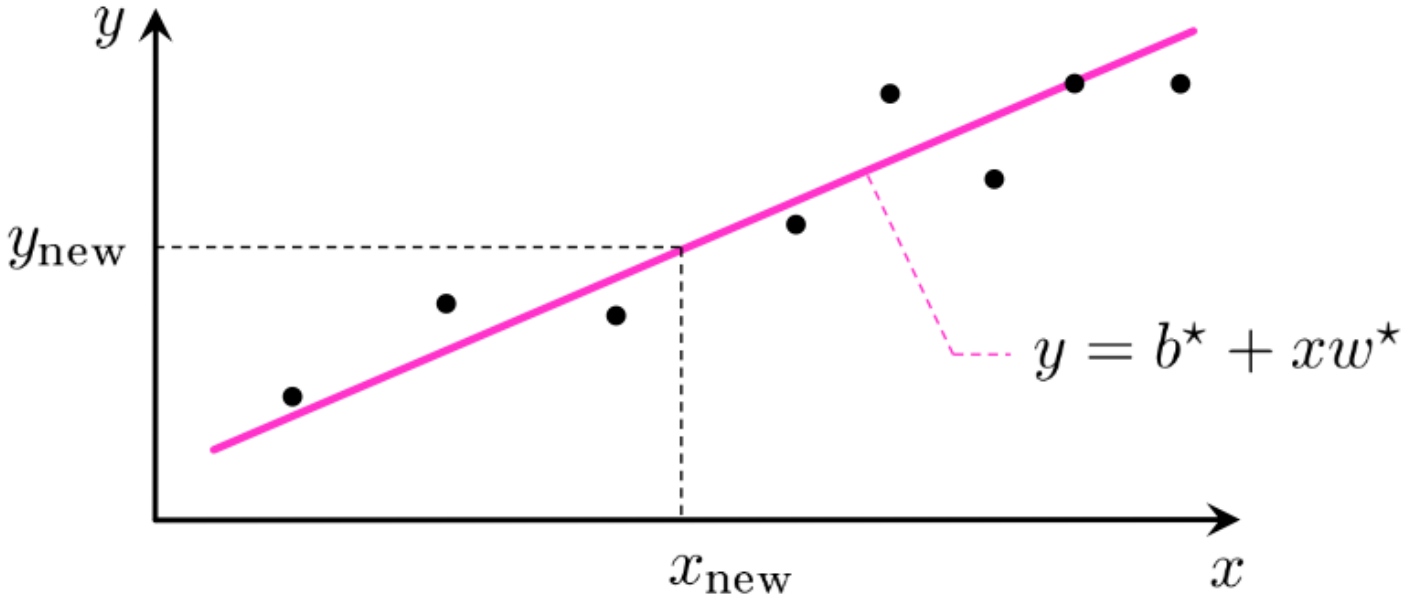
$$\Downarrow$$

$$(b^\star, \mathbf{w}^\star)$$

$$\Downarrow$$

$$y_{\text{new}} = b^\star + \mathbf{x}_{\text{new}}^T \mathbf{w}^\star$$





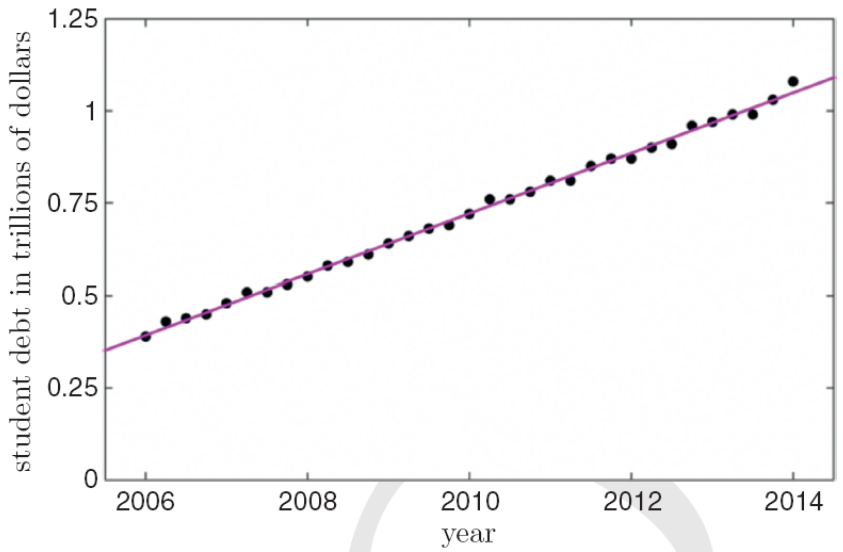
Minimization of the LS Cost Function

$$\tilde{\mathbf{x}}_p = \begin{bmatrix} 1 \\ \mathbf{x}_p \end{bmatrix} \qquad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \quad \Longrightarrow \quad g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} - y_p\right)^2$$

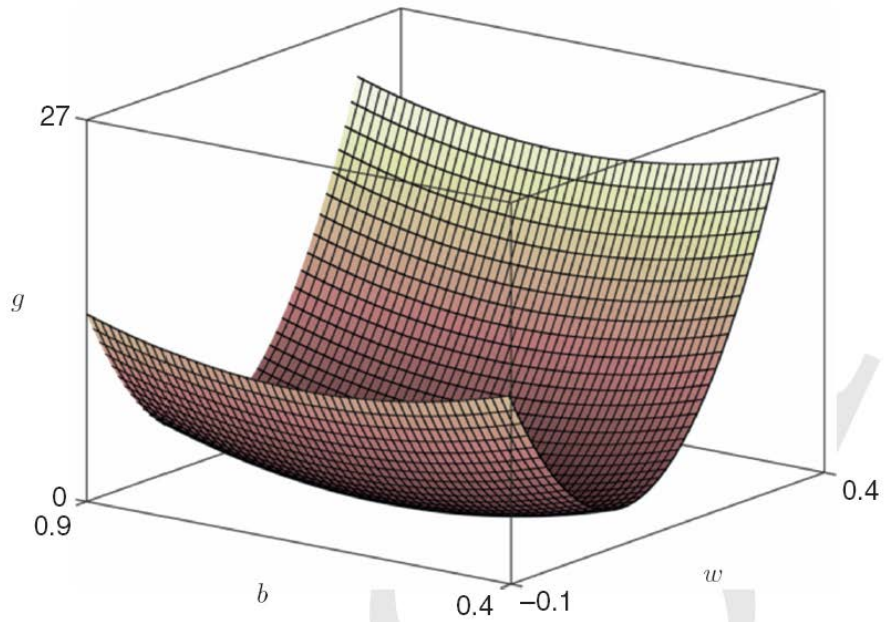
$$\nabla g\left(\tilde{\mathbf{w}}\right) = 2\sum_{p=1}^{P} \tilde{\mathbf{x}}_p \left(\tilde{\mathbf{x}}_p^T \tilde{\mathbf{w}} - y_p\right) = 2\left(\sum_{p=1}^{P} \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T\right) \tilde{\mathbf{w}} - 2\sum_{p=1}^{P} \tilde{\mathbf{x}}_p y_p$$

$$\left(\sum_{p=1}^{P} \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T\right) \tilde{\mathbf{w}} = \sum_{p=1}^{P} \tilde{\mathbf{x}}_p y_p \quad \Longrightarrow \quad \tilde{\mathbf{w}}^\star = \left(\sum_{p=1}^{P} \tilde{\mathbf{x}}_p \tilde{\mathbf{x}}_p^T\right)^{-1} \sum_{p=1}^{P} \tilde{\mathbf{x}}_p y_p$$

Student Loan Debt



Total student loan debt in the United States measured quarterly from 2006 to 2014. The rapid increase of the debt, measured by the slope of the trend line fit to the data, confirms the concerning claim that student debt is growing (dangerously) fast. The debt data shown in this figure was taken from [46].



The surface generated by the Least Squares cost function using the student loan debt data shown in Fig. 1.8, is clearly convex. However, regardless of the dataset, the Least Squares cost for linear regression is always convex.

Weighted feature sum

$$b + \mathbf{x}_p^T \mathbf{w} \approx y_p \quad \text{for all } p = 1 \ldots P.$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_P \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,P} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,P} \end{bmatrix}$$

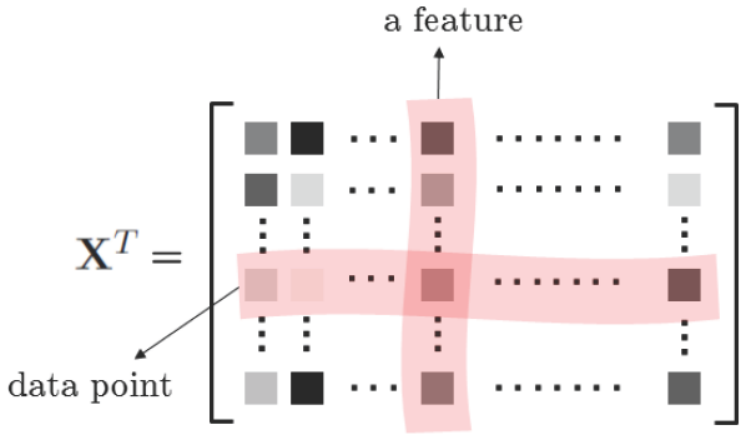
$N \times P$

$$\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_P \end{bmatrix}^T$$

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_N \end{bmatrix}^T$$

$$b\mathbf{1}_{P\times 1} + \mathbf{X}^T \mathbf{w} \approx \mathbf{y}$$

Weighted sum of features



$$\begin{bmatrix} b \\ \vdots \\ b \end{bmatrix} + \overset{\mathbf{X}^T}{\begin{bmatrix} \blacksquare & \blacksquare & \cdots & \blacksquare \end{bmatrix}} \overset{\mathbf{w}}{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}} \approx \overset{\mathbf{y}}{\begin{bmatrix} \blacksquare \end{bmatrix}} \longleftrightarrow \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} b + \begin{bmatrix} \blacksquare \end{bmatrix} w_1 + \begin{bmatrix} \blacksquare \end{bmatrix} w_2 + \cdots + \begin{bmatrix} \blacksquare \end{bmatrix} w_N \approx \begin{bmatrix} \blacksquare \end{bmatrix}$$

What are “features” ?

Features: summary

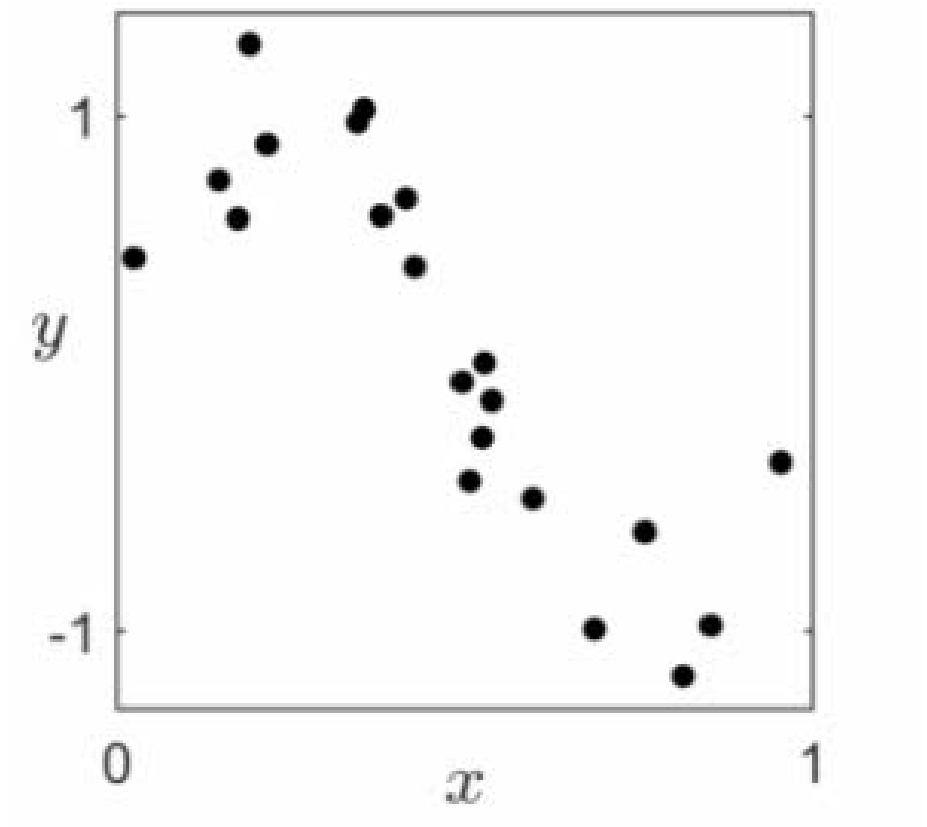
What are features?

- **Geometrically:** features are transformations of the input that provoke a good *nonlinear* fit/separation

- **Philosophically:** features are those defining characteristics of the phenomenon underlying a dataset that allow for optimal learning

Knowledge-driven feature design for regression

Feature design: regression



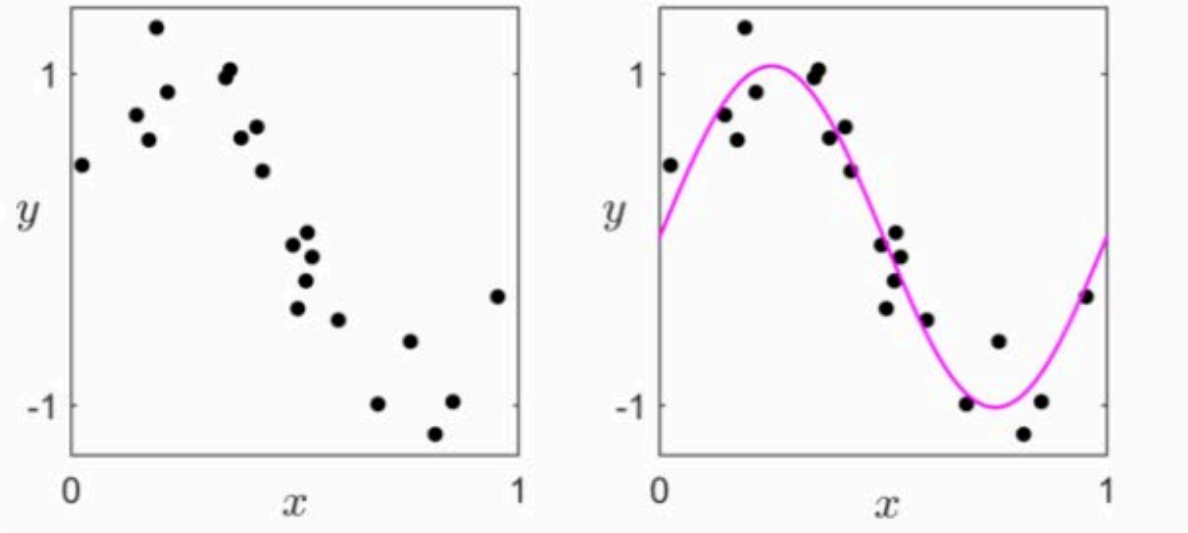
Feature design: regression

- Data appears distributed periodically, i.e., for each p it looks like

$$b + f(x_p)\, w = b + \sin(2\pi x_p)\, w \approx y_p$$

  where w needs to be tuned to make $\approx$ as tight as possible

- In ML terms: $f(x_p) = \sin(2\pi x_p)$ called a *feature transformation*
- Note the feature and output are *linearly* related in w



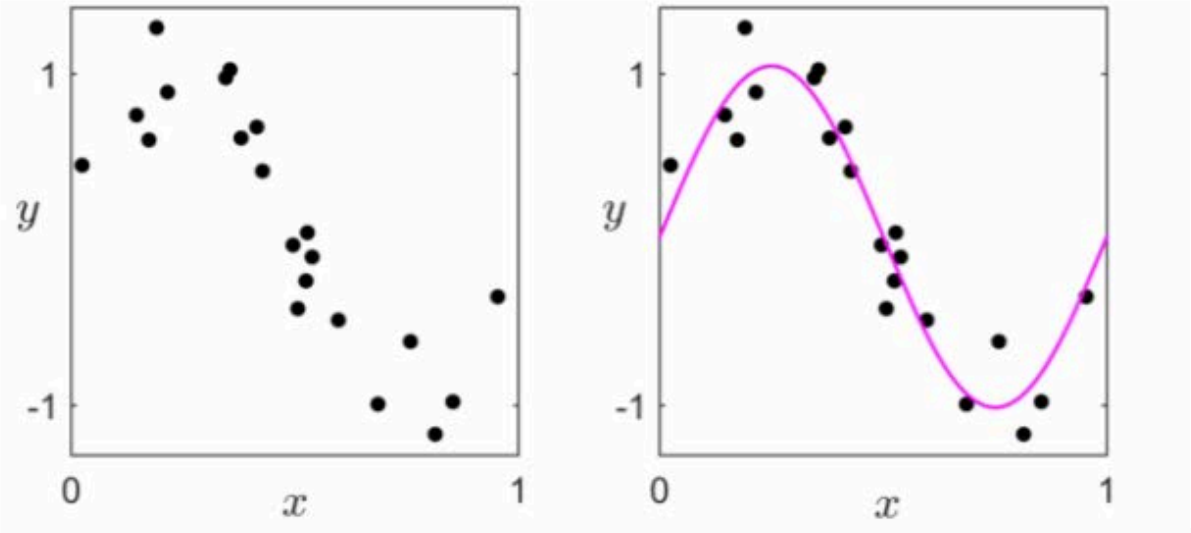
Feature design: regression

- Data appears distributed periodically, i.e., for each p it looks like

$$b + f\left(x_p\right) w = b + \sin\left(2\pi x_p\right) w \approx y_p$$

where w needs to be tuned to make $\approx$ as tight as possible

- Tune w so that it minimizes the squared between each feature and its corresponding output



$$\underset{w}{\text{minimize}} \sum_{p=1}^{P} \left(f\left(x_p\right) w - y_p\right)^2$$

can solve for optimal w in closed form or use iterative procedure like e.g., gradient descent

Minimization of the LS Cost Function

$$\underset{b,w}{\text{minimize}} \sum_{p=1}^{P} \left(b + f_p w - y_p\right)^2$$

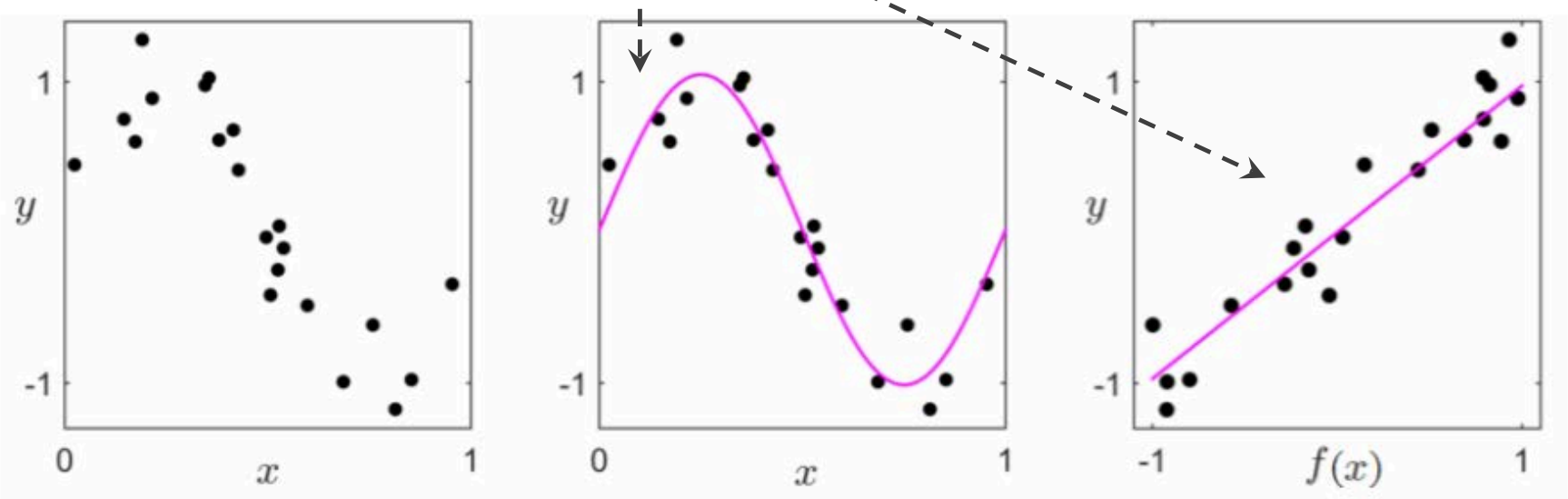
$$\tilde{\mathbf{f}}_p = \begin{bmatrix} 1 \\ f_p \end{bmatrix}, \qquad \tilde{\mathbf{w}} = \begin{bmatrix} b \\ w \end{bmatrix} \longrightarrow \quad g\left(\tilde{\mathbf{w}}\right) = \sum_{p=1}^{P} \left(\tilde{\mathbf{f}}_p^T \tilde{\mathbf{w}} - y_p\right)^2$$

$$\left(\sum_{p=1}^{P} \tilde{\mathbf{f}}_p \tilde{\mathbf{f}}_p^T\right) \tilde{\mathbf{w}} = \sum_{p=1}^{P} \tilde{\mathbf{f}}_p y_p.$$

Feature design: regression

A properly designed feature (or set of features) for linear regression provides a good *nonlinear* fit in the original feature space and, simultaneously, a good *linear* fit in the transformed feature space.

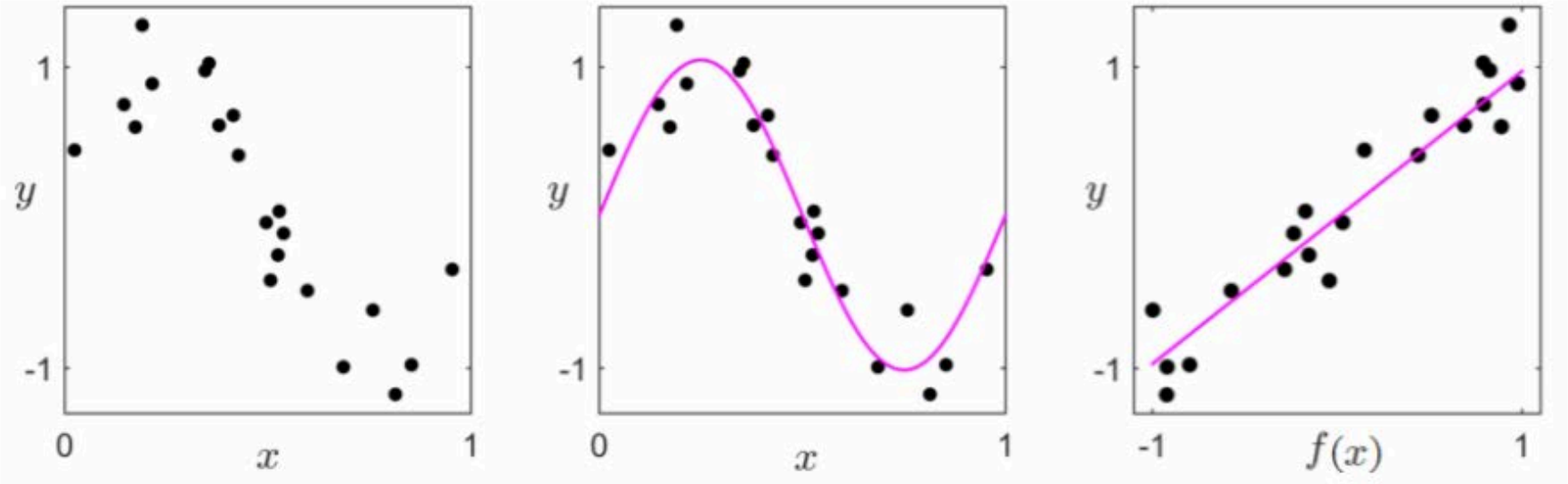
$$b + f(x)\,w = b + \sin(x)\,w \approx y$$



Feature design: regression

A properly designed feature (or set of features) for linear regression provides a good *nonlinear* fit in the original feature space and, simultaneously, a good *linear* fit in the transformed feature space.

$$b + \sum_{m=1}^{M} f_m(x_p)\, w_m \approx y_p$$



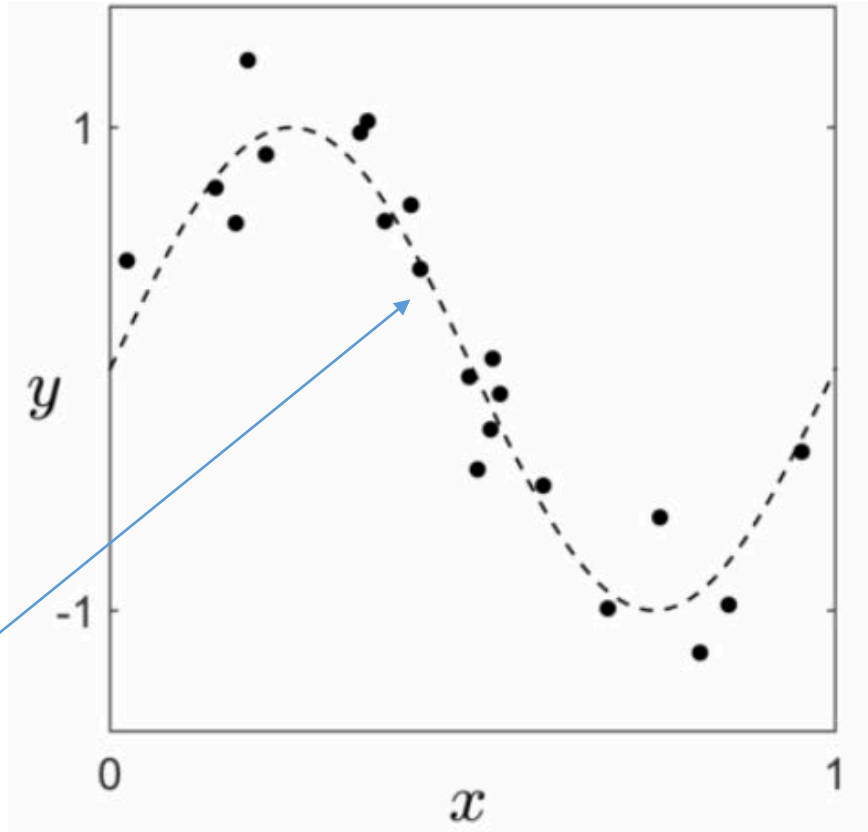
Feature design: regression

What are we aiming to do with features?

- The data we receive are noisy samples from an underlying function, the phenomenon (e.g., gravity) we wish to understand/predict

- In determining correct features we aim to approximate this function from the data as well as possible

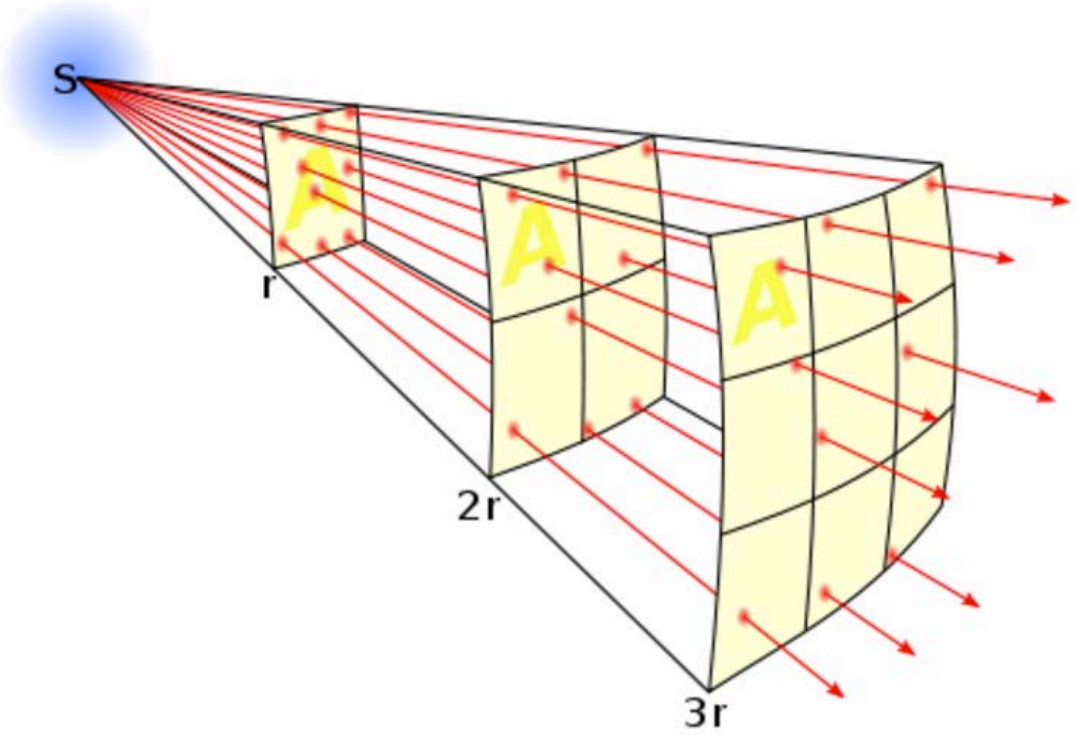
- Estimate this as $b + \sum_{m=1}^{M} f_m(\mathbf{x})\, w_m$



Galileo and feature design

- In 1638 Galileo creates a wonderful experiment to quantify the pull of gravity on objects

- Intuition at the time: gravity works like light, its pull dissipates as $1/(\text{distance})^2$



number of rays of light passing through a patch dissipates like $1/r^2$ where r is the distance from the source

Galileo and feature design

- No precise clock existed at the time – so he couldn't just drop an object and time how long it took to reach the ground

- So did something similar using a ramp

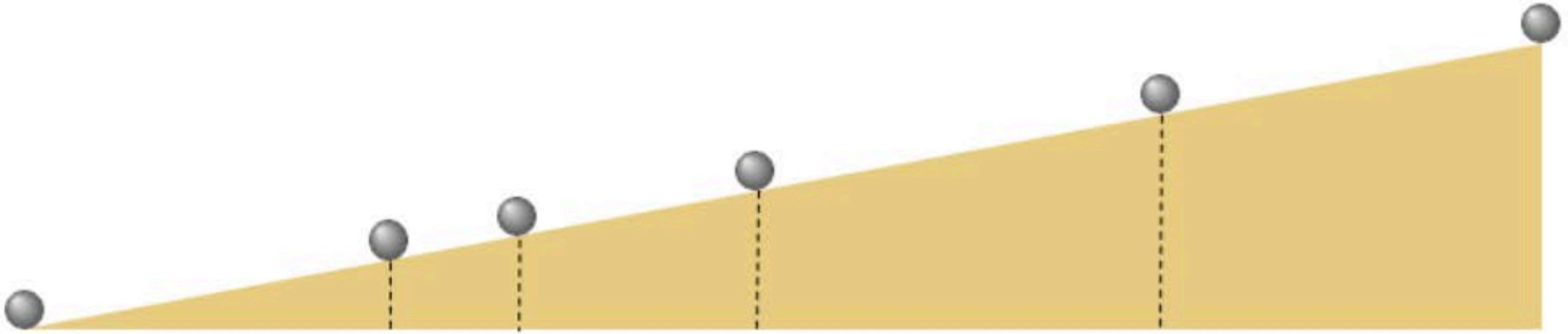


Image taken from [29]

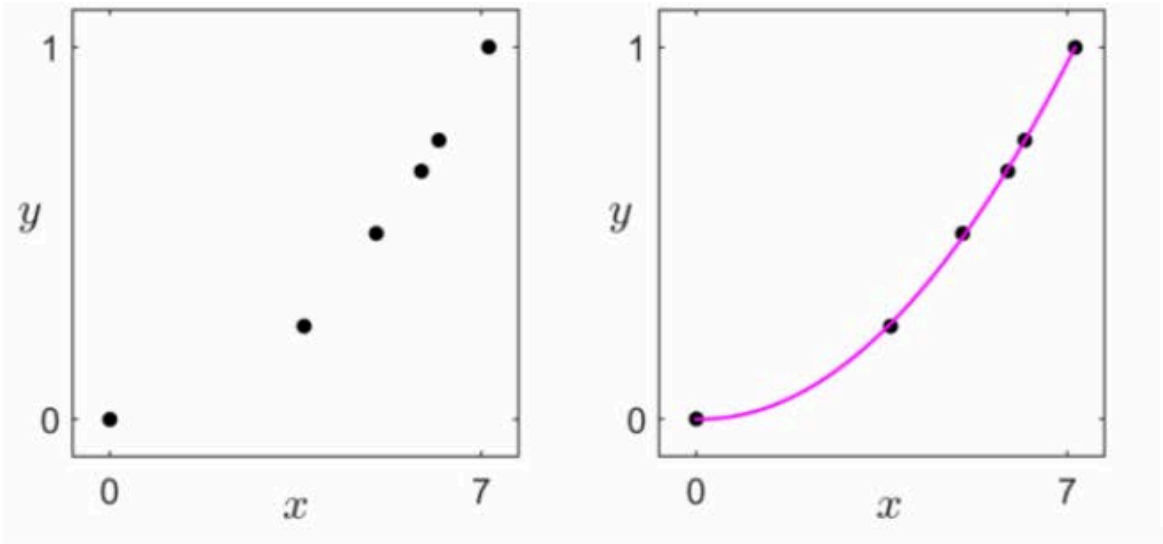
Galileo and feature design

- Data appears distributed quadratically, i.e., for each p it looks like

$$f\left(x_p\right) w = x_p^2 w \approx y_p$$

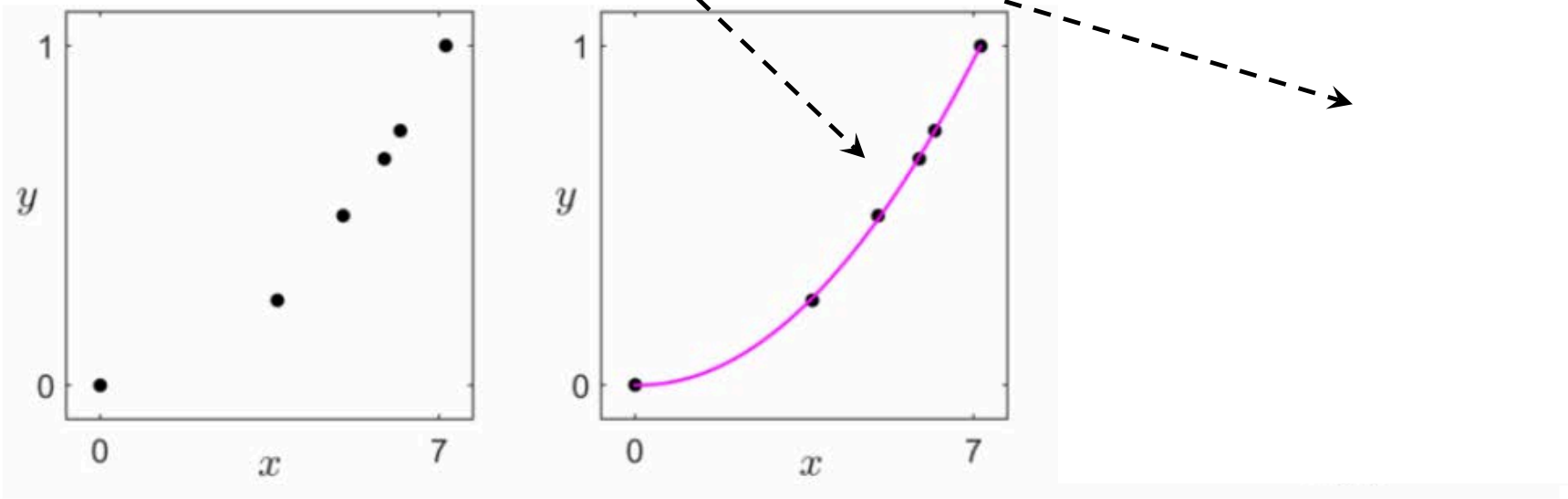
  where w needs to be tuned to make $\approx$ as tight as possible

- In ML terms: $f\left(x_p\right) = x_p^2$ called a *feature transformation* or simply a *feature*
- Note the feature and output are *linearly* related in w



Galileo and feature design

$$f(x)\,w = x^2 w \approx y$$



Moore's law

$$\exp\left(b + x_p w\right) \approx y_p$$

$$b + x_p w \approx \log\left(y_p\right)$$

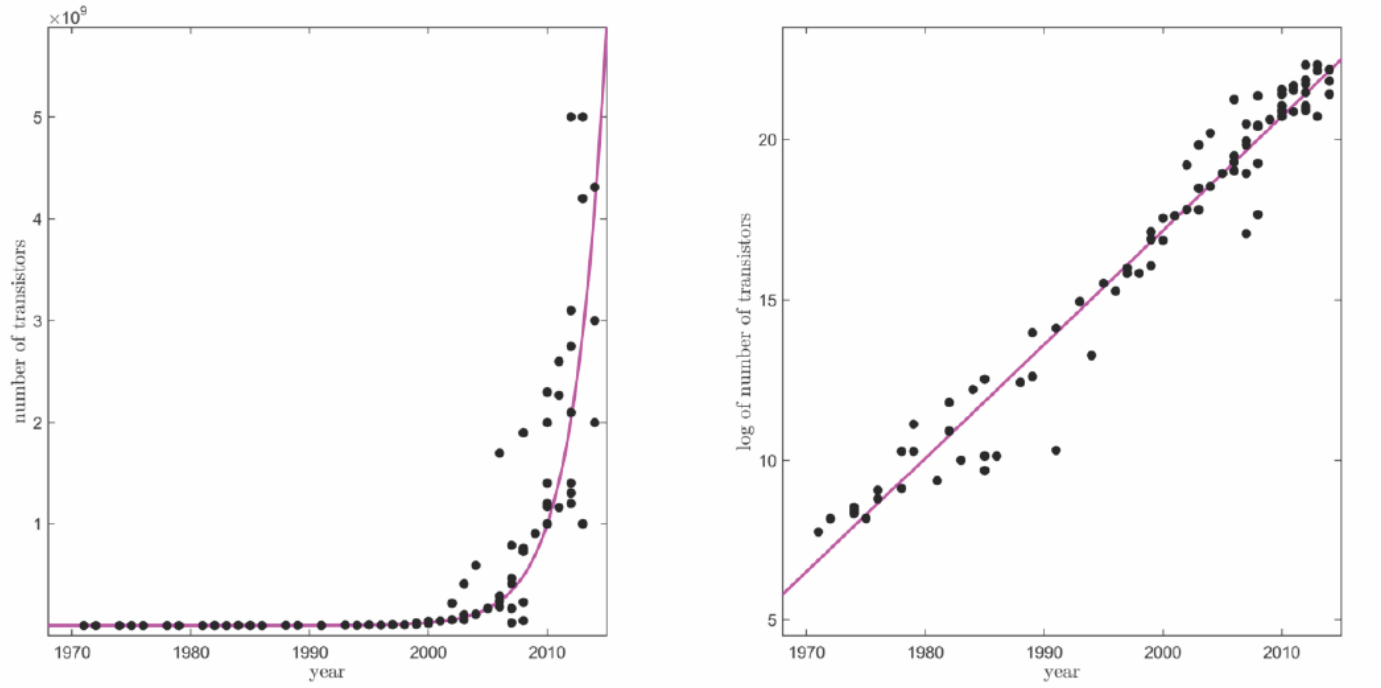


Figure 3.9. *(left) As Moore proposed, number of transistors in microprocessors versus the year they were invented, follows an exponential pattern. (right) Transforming the data by taking the log of the output reveals the linear relationship in the new space. A linear fit in this space corresponds directly to the exponential one in the original space.*

Gauss and the orbit of celestial bodies

$$\left(\frac{x_{1,p}}{\nu_1}\right)^2 + \left(\frac{x_{2,p}}{\nu_2}\right)^2 \approx 1$$

$$x_{1,p}^2 w_1 + x_{2,p}^2 w_2 = 1$$

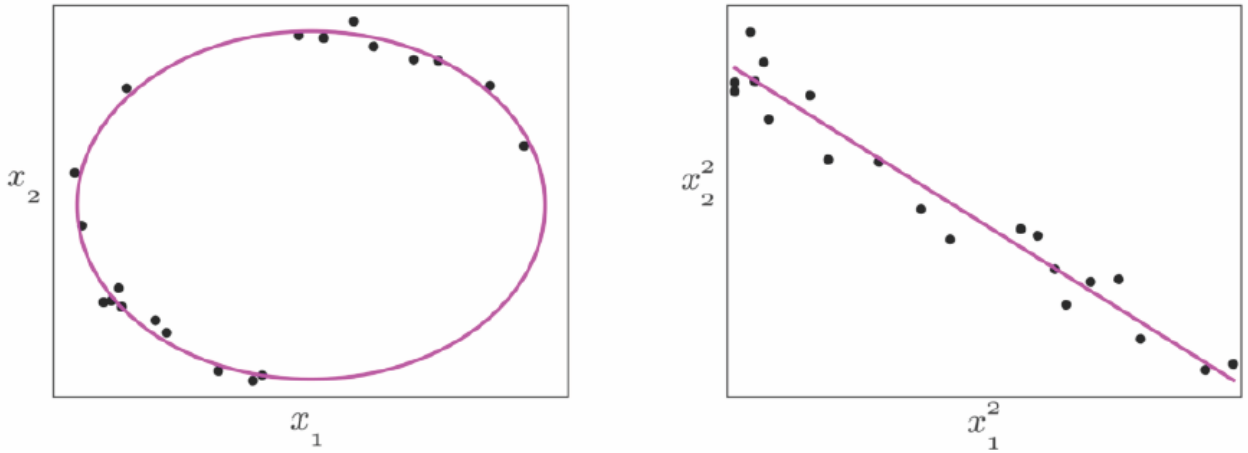
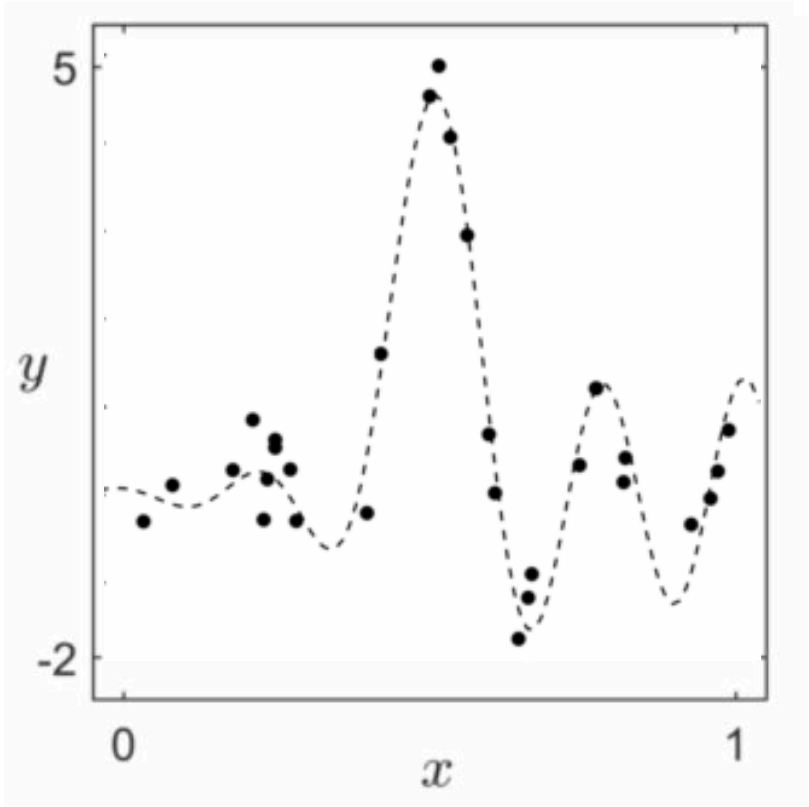


Figure 3.10. *(left) Simulated observation data for the location of the asteroid Pallas on its orbital plane. The ellipsoidal curve fit to the data approximates the true orbit of Pallas. (right) Fitting an ellipsoid to the data in the ambient feature space is equivalent to fitting a line to the data in a new space where both dimensions are squared.*

Feature design: regression (re-visited)

- intuition can be hard to come by
- inputs are rarely one/two dimensional, can be hard to visualize data
- even so it can be very hard to determine proper feature transformation 'by eye'
- if we have enough (relatively clean) data we can *learn* the appropriate features automatically



$$y(x) = e^{3x}\frac{\sin\left(3\pi^2(x-0.5)\right)}{3\pi^2(x-0.5)}.$$