Linear Regression

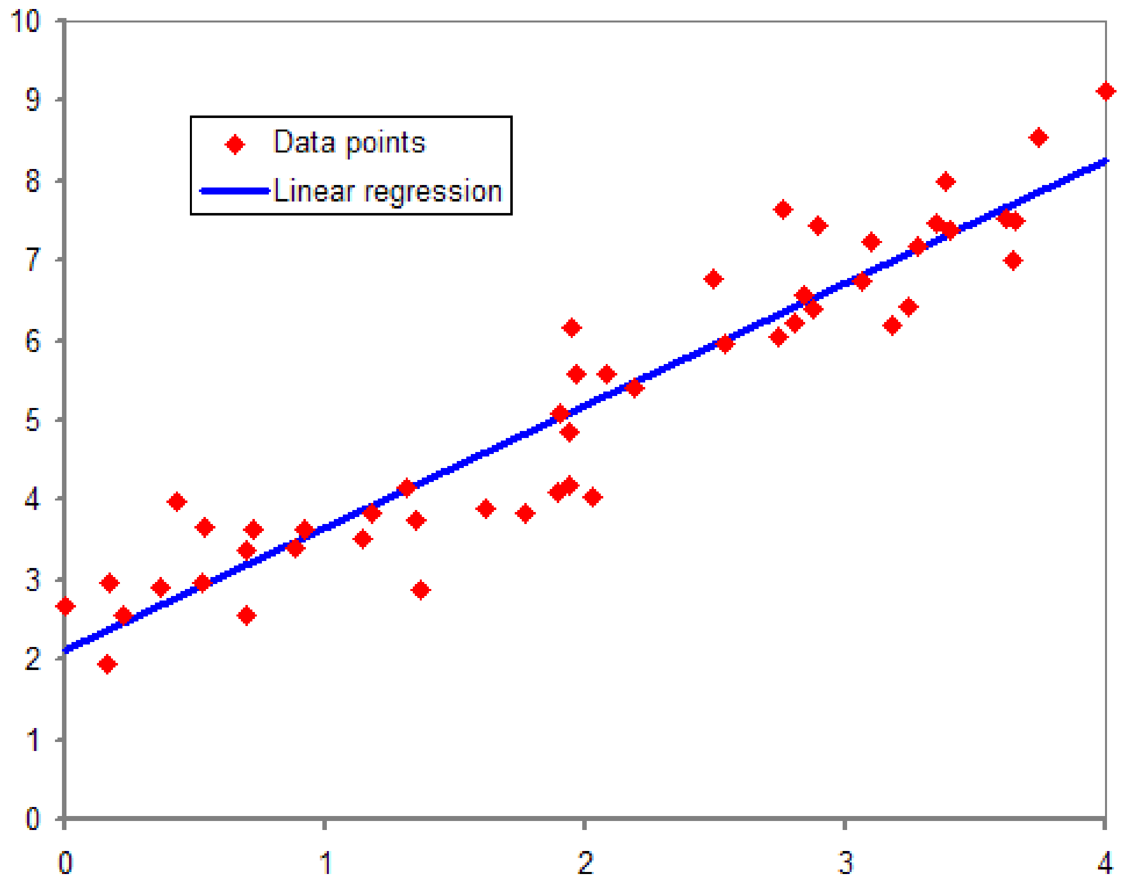
AI/ML Teaching

Goals

- Basic concept of linear regression
- Estimating parameters with linear models
- Linear regression in PyTorch

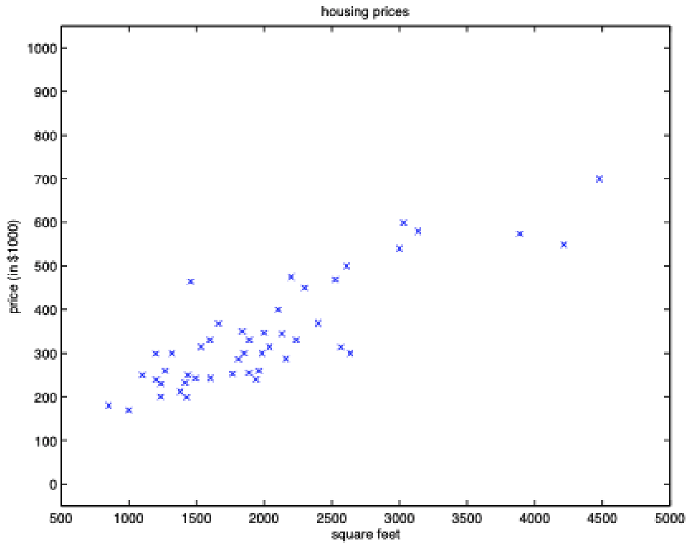
Regression

In statistical modeling, **regression analysis** is a set of statistical processes for estimating the relationships between a dependent variable (often called the *outcome* or *response* variable, or a *label* in machine learning parlance) and one or more error-free independent variables (often called *regressors*, *predictors*, *covariates*, *explanatory variables* or *features*).

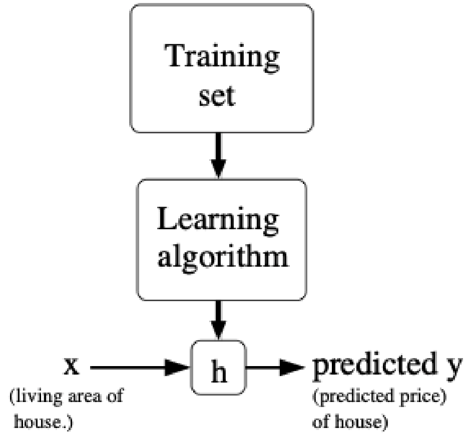


Supervised Learning

Living area ($feet^2$)	Price (1000\$s)
2104	400
1600	330
2400	369
1416	232
3000	540
⋮	⋮



- Input: $x^{(i)}$
- Output: $y^{(i)}$
  - Continuous → regression
  - Discrete → classification
- Training example $\left(x^{(i)}, y^{(i)}\right)$
- Training set $\{x^{(i)}, y^{(i)}\}; i = 1, \dots m$
- Given a training set, to learn a function $h: \mathcal{X} \rightarrow \mathcal{Y}$

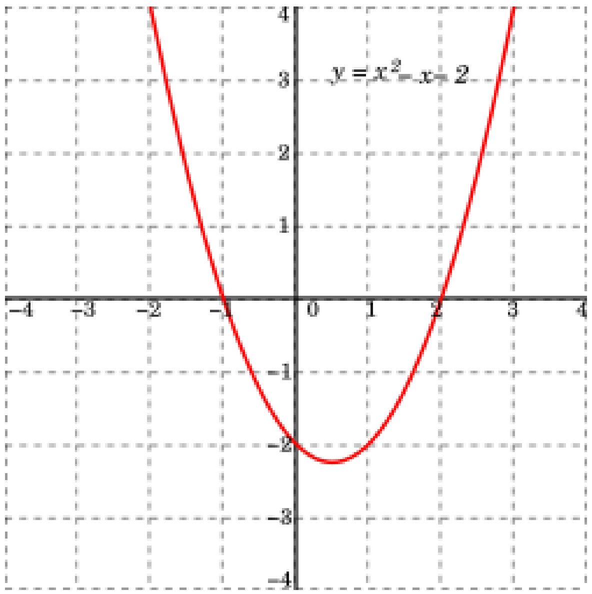


Supervised Learning

- $h(x) = \theta x$
- Cost function: $J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h\left(x^{(i)}\right) - y^{(i)}\right)^2$ → Least-square!
- $\min_{\theta} J(\theta)$: minimum of quadratic function
  - $\frac{\partial J(\theta)}{\partial \theta} = 0$
- Learning algorithm

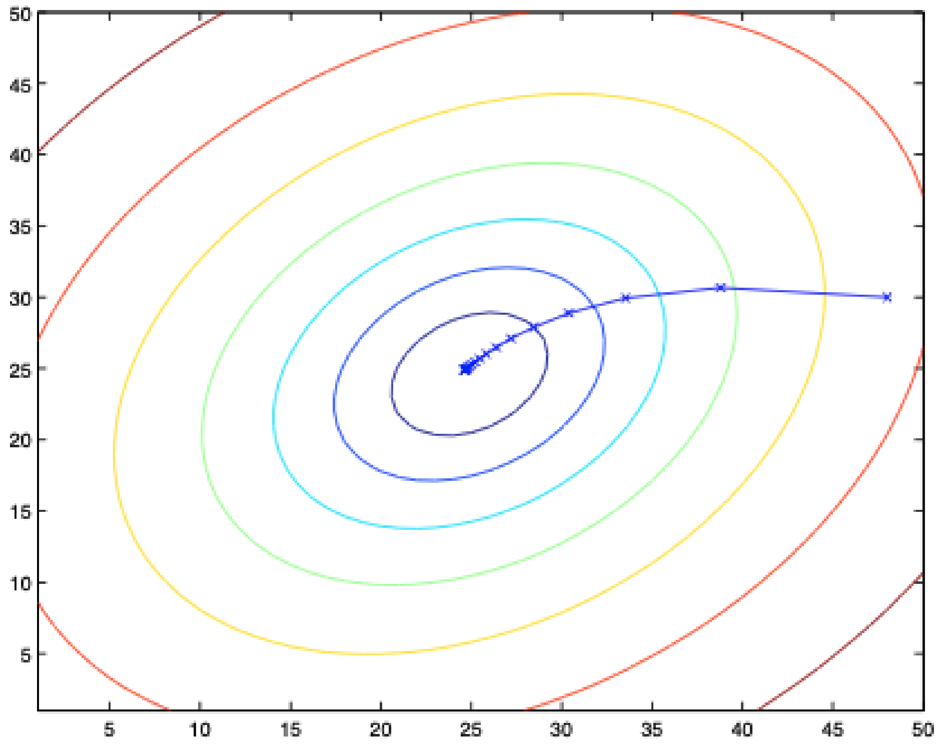
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

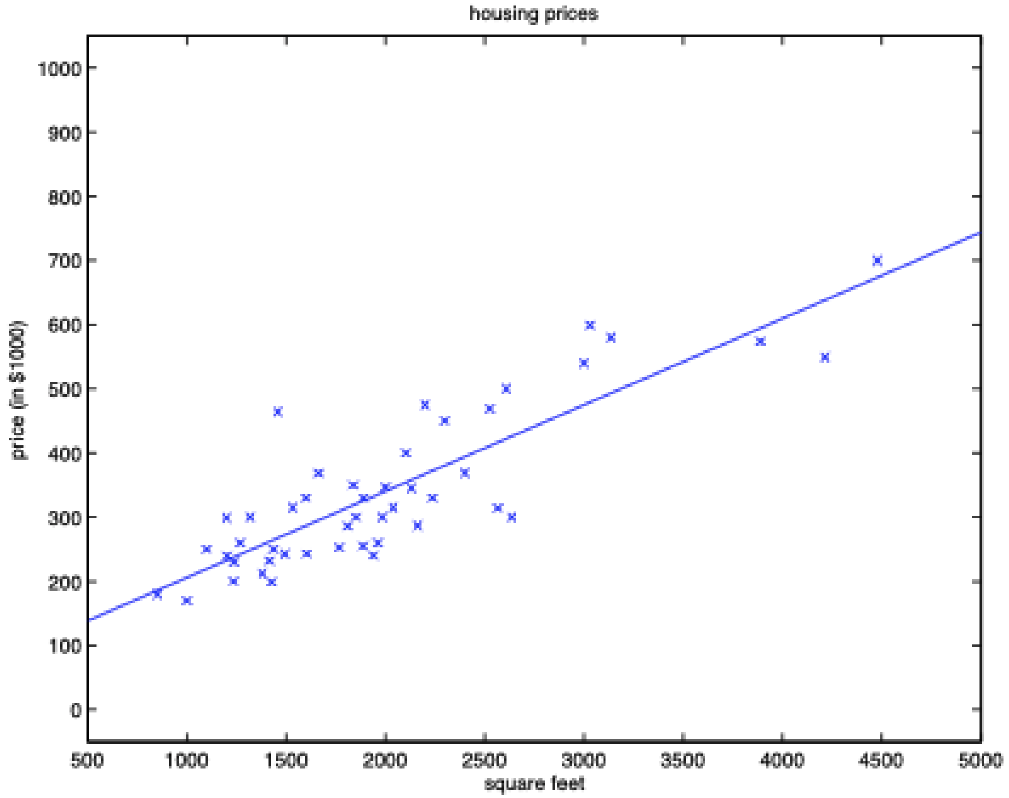
$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2}\left(h_\theta(x) - y\right)^2 \\
&= 2 \cdot \frac{1}{2}\left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j}\left(h_\theta(x) - y\right) \\
&= \left(h_\theta(x) - y\right) \cdot \frac{\partial}{\partial \theta_j}\left(\sum_{i=0}^{n} \theta_i x_i - y\right) \\
&= \left(h_\theta(x) - y\right) x_j
\end{aligned}$$



Multi-variable: $h(x) = \theta_1 x + \theta_0$

- (Stochastic) gradient descent





Closed-form solution

$$X = \begin{bmatrix} — (x^{(1)})^T — \\ — (x^{(2)})^T — \\ \vdots \\ — (x^{(m)})^T — \end{bmatrix}. \qquad \vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}.$$

$$\boxed{X\theta - \vec{y}} = \begin{bmatrix} (x^{(1)})^T\theta \\ \vdots \\ (x^{(m)})^T\theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} = \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{bmatrix}.$$

$$h_\theta(x^{(i)}) = (x^{(i)})^T\theta$$

$$\begin{aligned} \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) &= \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \\ &= J(\theta) \end{aligned}$$

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta \frac{1}{2}(X\theta - \vec{y})^T(X\theta - \vec{y}) \\ &= X^T X\theta - X^T\vec{y} \end{aligned}$$

$$\theta = (X^T X)^{-1} X^T \vec{y}.$$

Closed-form solution is possible because it is “linear” regression

Is this linear regression?

$$h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

Reference

- Andrew Ng, CS229 Lecture Notes