Regression

PHYS591000 Spring 2021

Reference: https://indico.cern.ch/event/619370/ Lecture by Michael Kagan

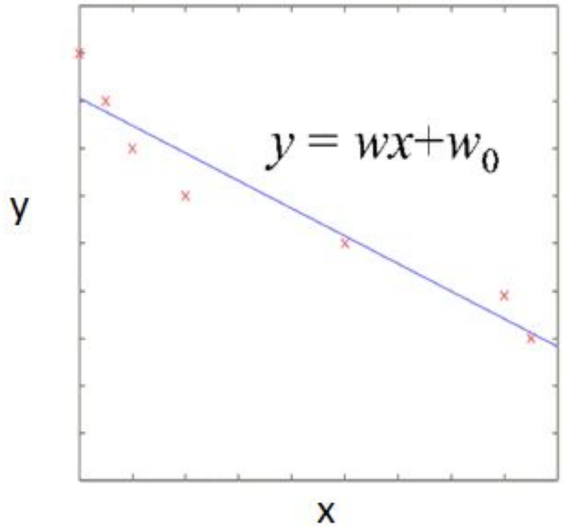
Outline

- Linear regression: Simply fit a line!
- Linear regression with multivariable and polynomials
- Problem of overfitting and regularization

Regression

- Start with the simplest case: Fit a line!
  → Linear regression

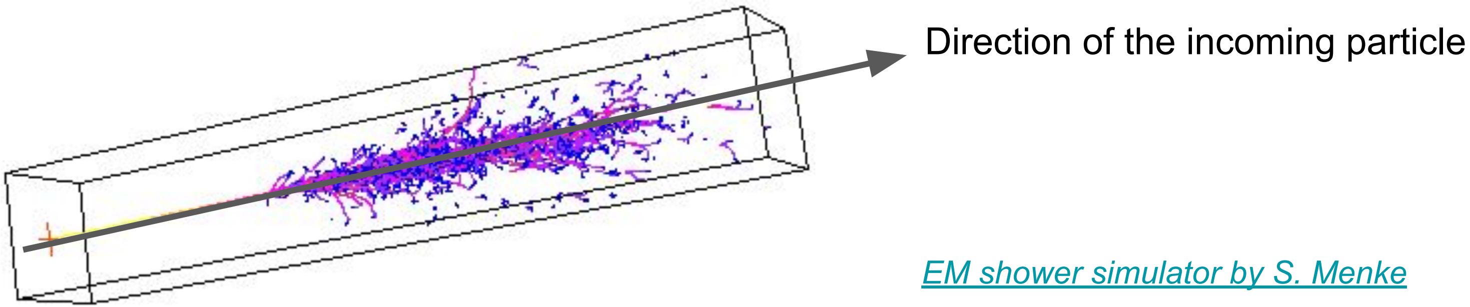
- Physics example: Predict energy of a particle using information of a calorimeter



Linear Regression

- Calorimeters are used to measure the energy of a particle

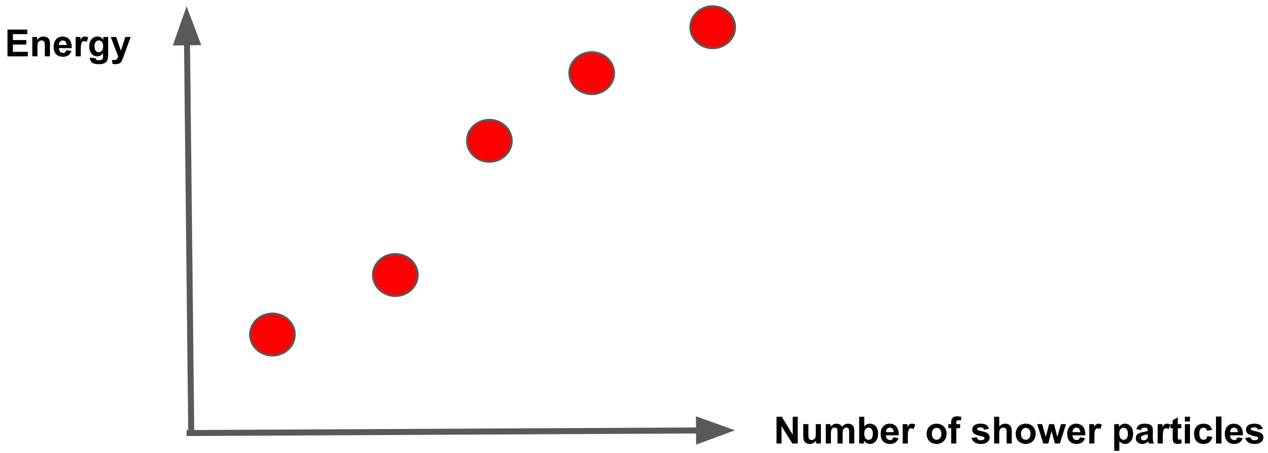
- The incoming particle interacts with materials in the calorimeter and produce a bunch of other particles. → “Shower”



EM shower simulator by S. Menke

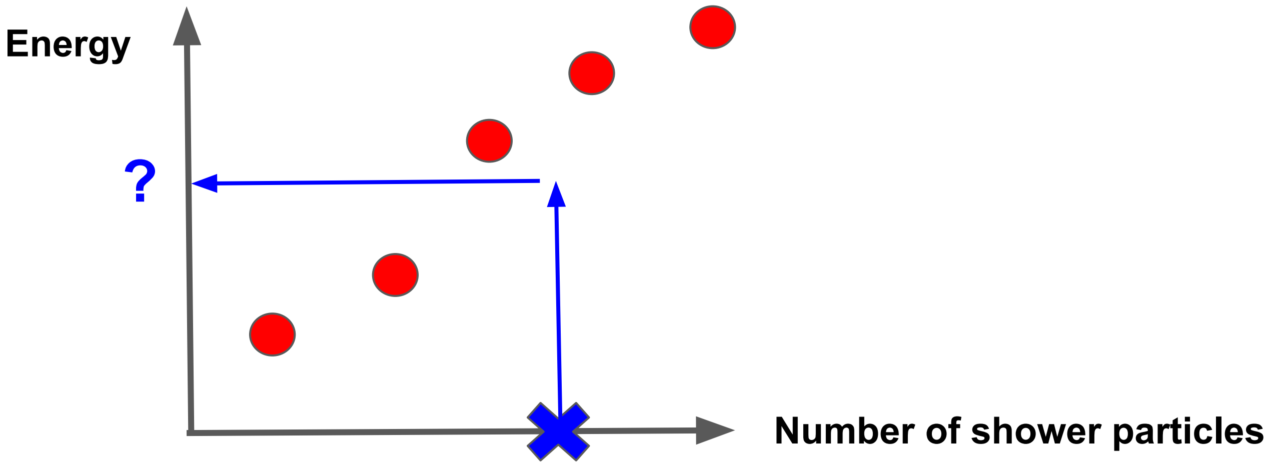
Linear Regression

- Expect the energy of the particle depends on the properties of the shower it creates, e.g. number of particles in the shower



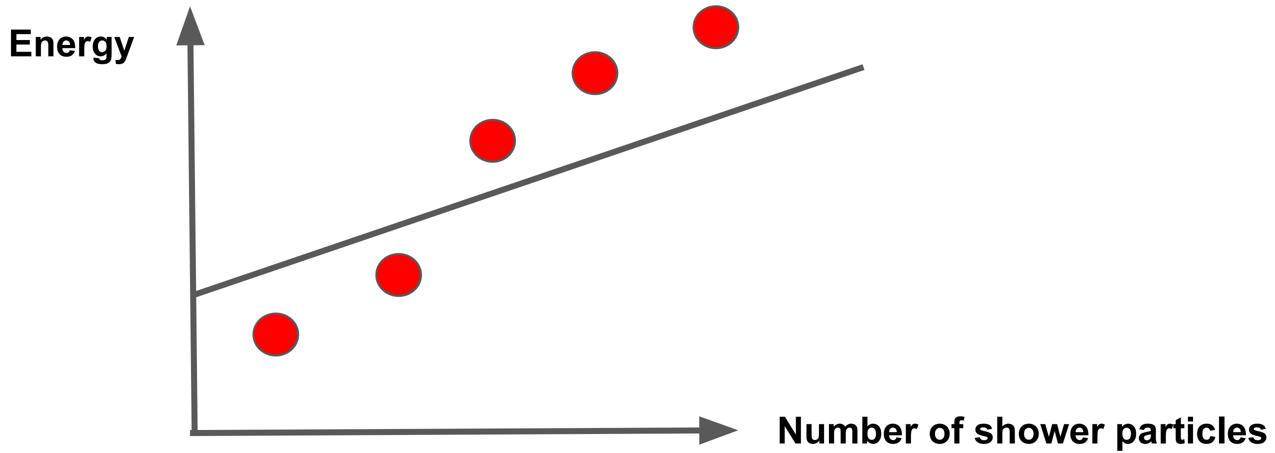
Linear Regression

- Goal: Predict the energy of a particle given the number of shower particles it creates.



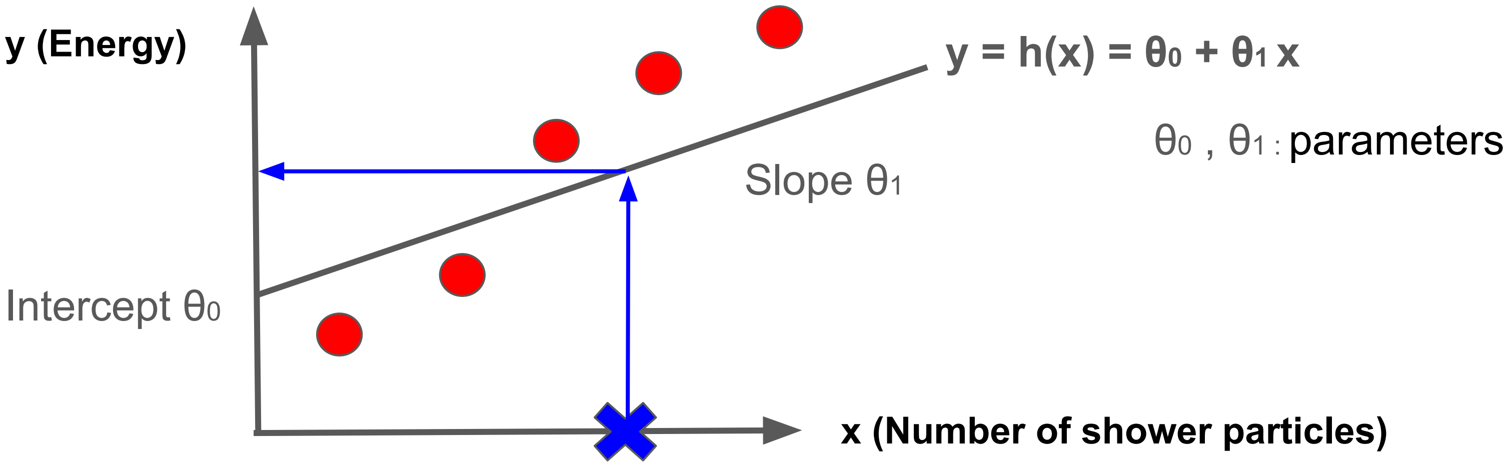
Linear Regression

- Fit a line!



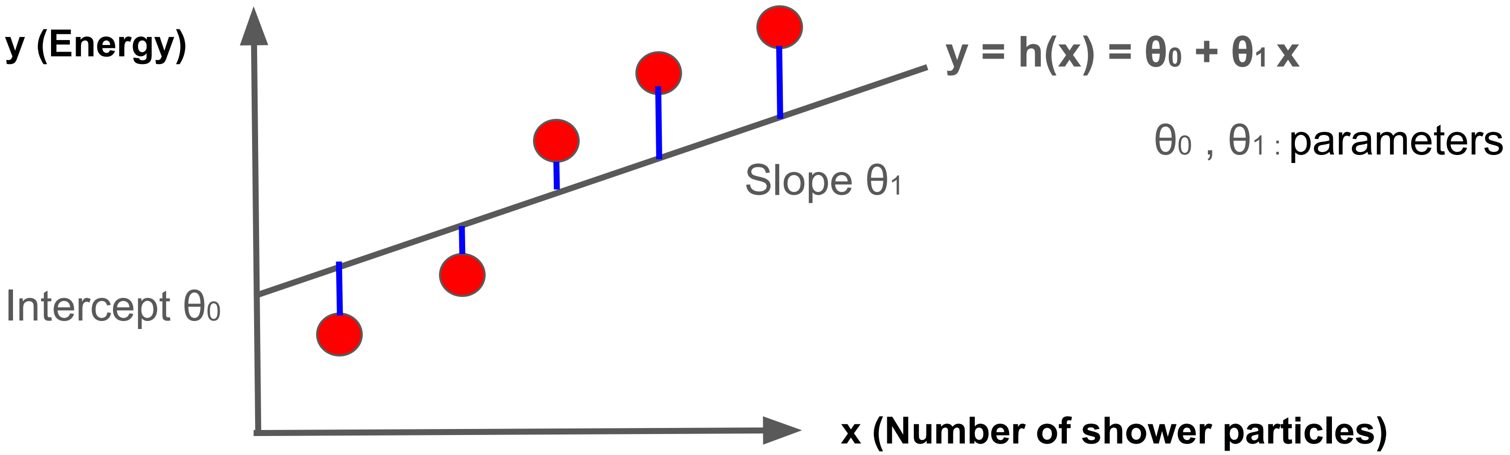
Linear Regression

- Make a prediction of y (“target”) given the value of x (“feature”) using a hypothesis $y = h(x) = \theta_0 + \theta_1 x$



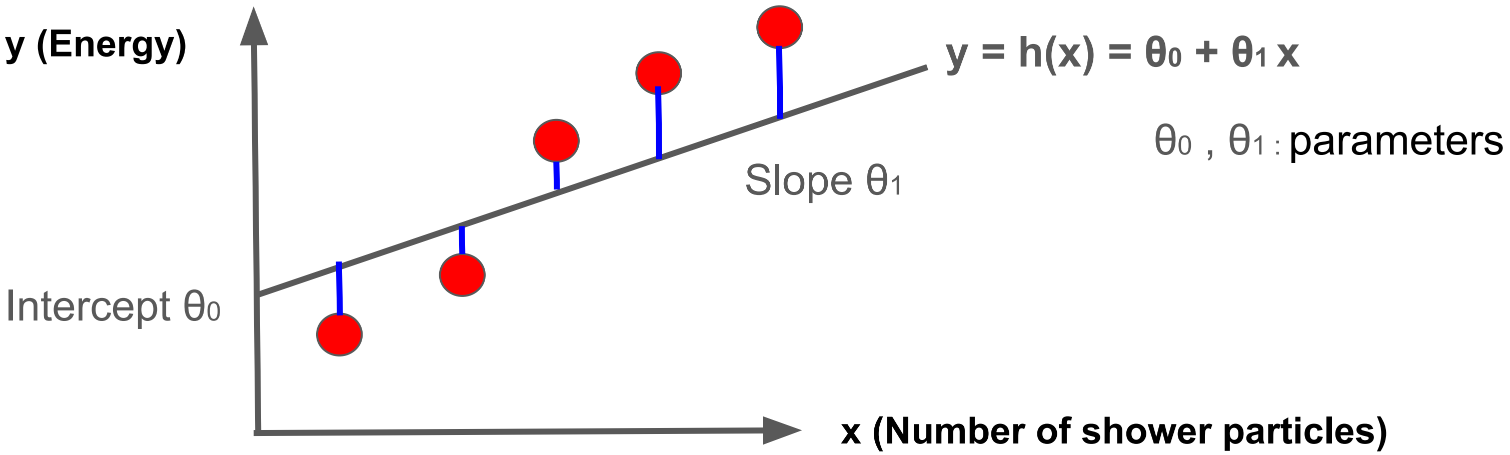
Linear Regression

- The line $y = h(x) = \theta_0 + \theta_1 x$ is the line that gives the *closest* predictions to all the points from the training sample



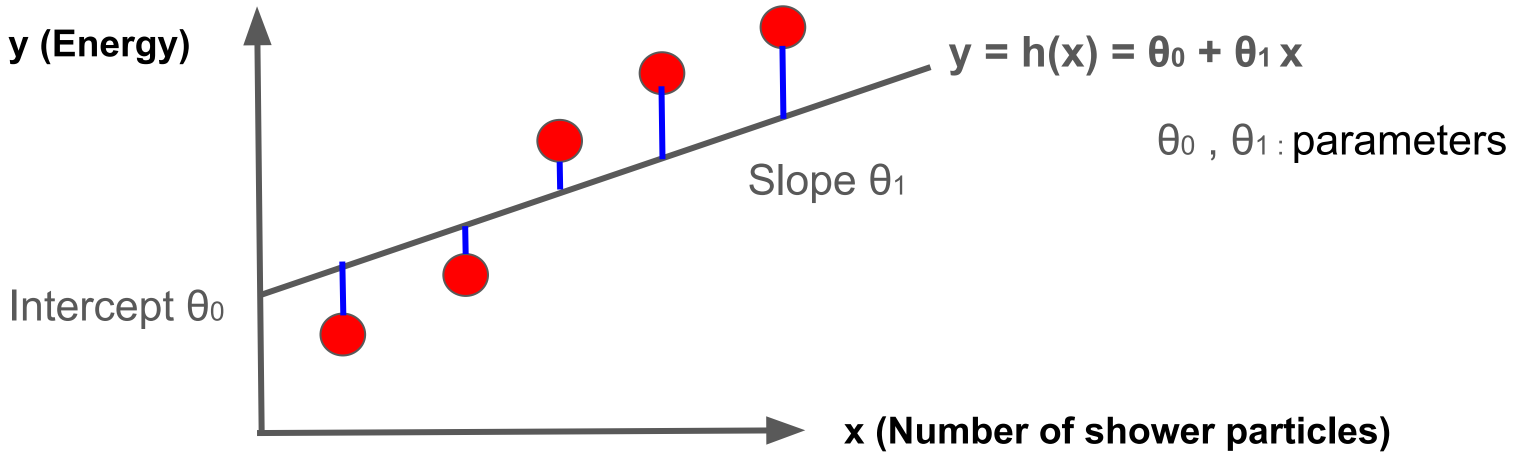
Linear Regression

- Find θ_0 and θ_1 that minimizes the sum of the squares $\Sigma_i(h(x_i)-y_i)^2$ from each (x_i, y_i) of the training sample → method of least squares



Linear Regression

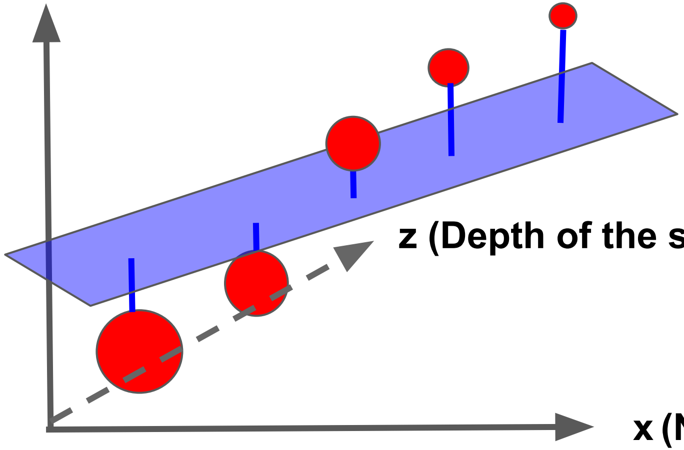
- $\Sigma_i(h(x_i)-y_i)^2$ is called the objective function (cost function, loss function)



Linear Regression

- Can extend to multivariable (more features) regression

→ Minimize $\Sigma_i(h(x_i,z_i)-y_i)^2$



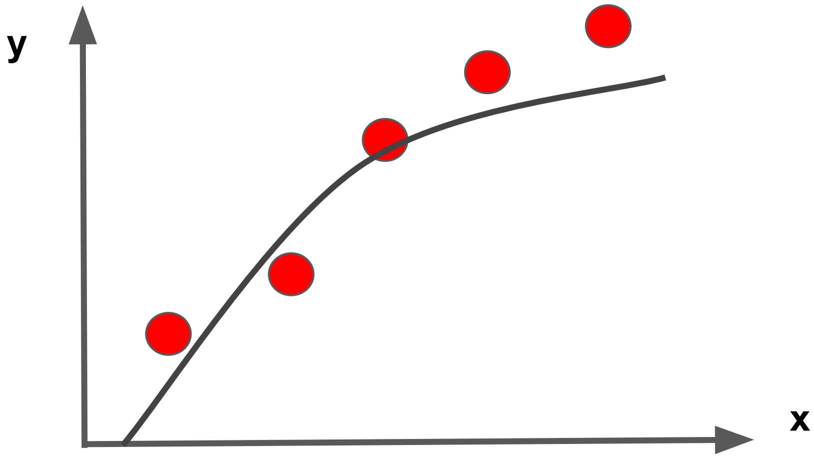
$$\mathbf{y = h(x, z) = \theta_0 + \theta_1 x + \theta_2 z}$$

θ_0 , θ_1 , θ_2: parameters

Linear Regression

- What if we want to fit with a polynomial function?

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

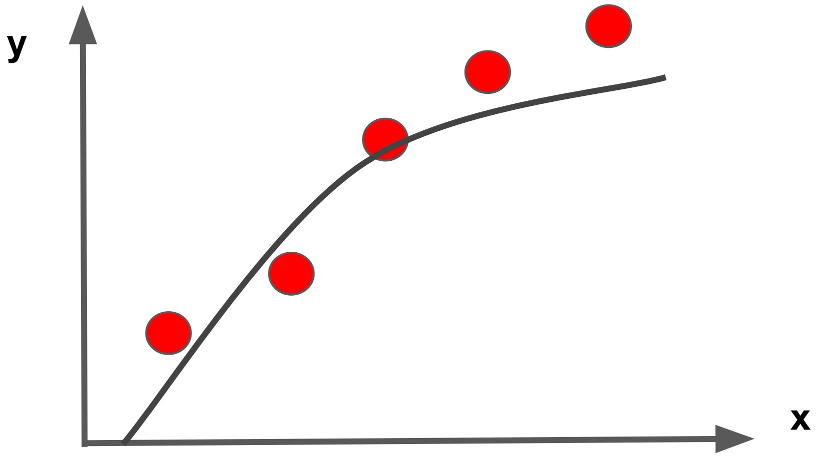


Linear Regression

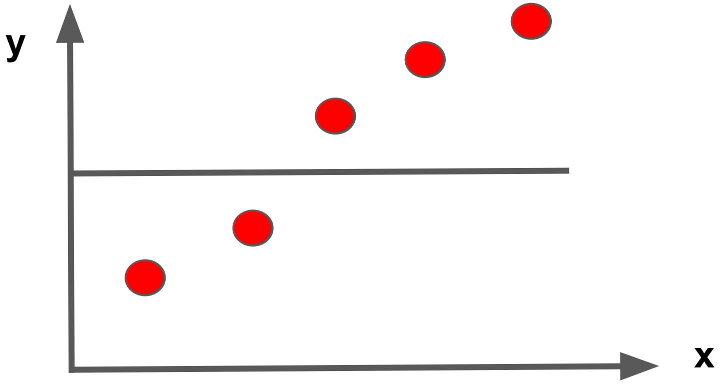
- What if we want to fit with a polynomial function?
→ Same as fitting with multiple features!

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

$$y = h(x, z) = \theta_0 + \theta_1 x + \theta_2 z$$

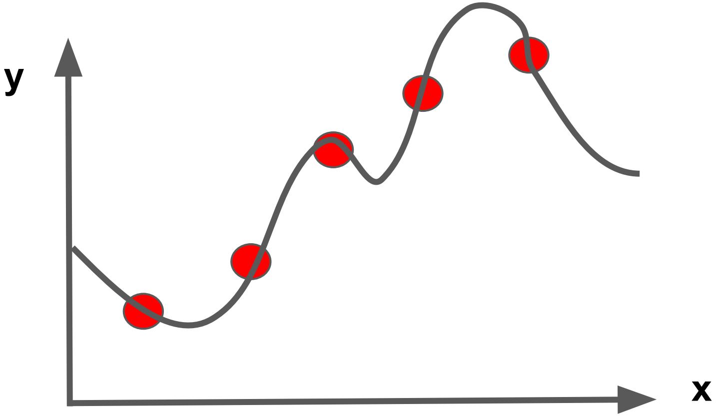


Linear Regression



$$y = h(x) = \theta_0$$

Underfitting



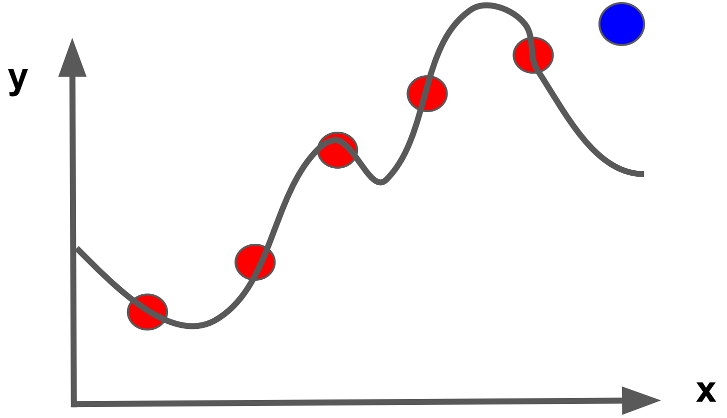
$$y = h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Overfitting

Linear Regression

Training

Testing



$$y = h(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

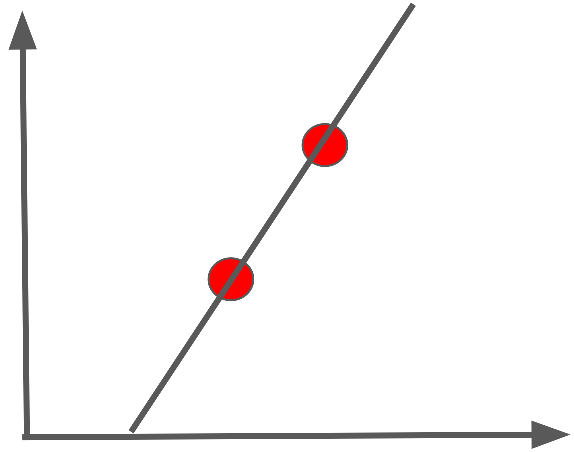
Overfitting: When there are too many parameters/features, the learned hypothesis may fit the training set ‘too well’ (sum of least square → 0) but fail to generalize to new samples (fail to make good predictions for test samples).

Overfitting

- Options to avoid/fix overfitting:
  - Manually reduce the number of features/parameters
  - Regularization: Add a ‘penalty’ for the sizes of parameters

Regularization

- Example: overfitting from two points → Sum of squares $\Sigma_i(h(x_i)-y_i)^2 = 0$

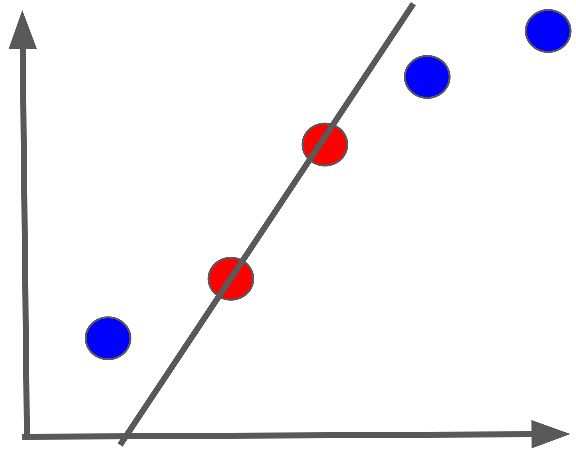


$$y = h(x) = \theta_0 + \theta_1 x$$

Regularization

- The line fits training (red) samples perfectly but fails to generalize to test samples (blue)

y (Energy)



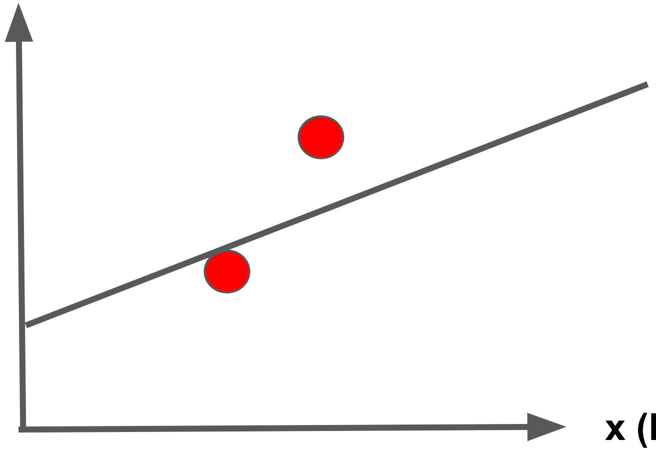
$$y = h(x) = \theta_0 + \theta_1 x$$

x (Number of shower particles)

Regularization

- Regularization: Minimize $\Sigma_i(h(x_i)-y_i)^2$ + $\boldsymbol{\lambda\theta_1^2}$ **λ:**hyperparameter

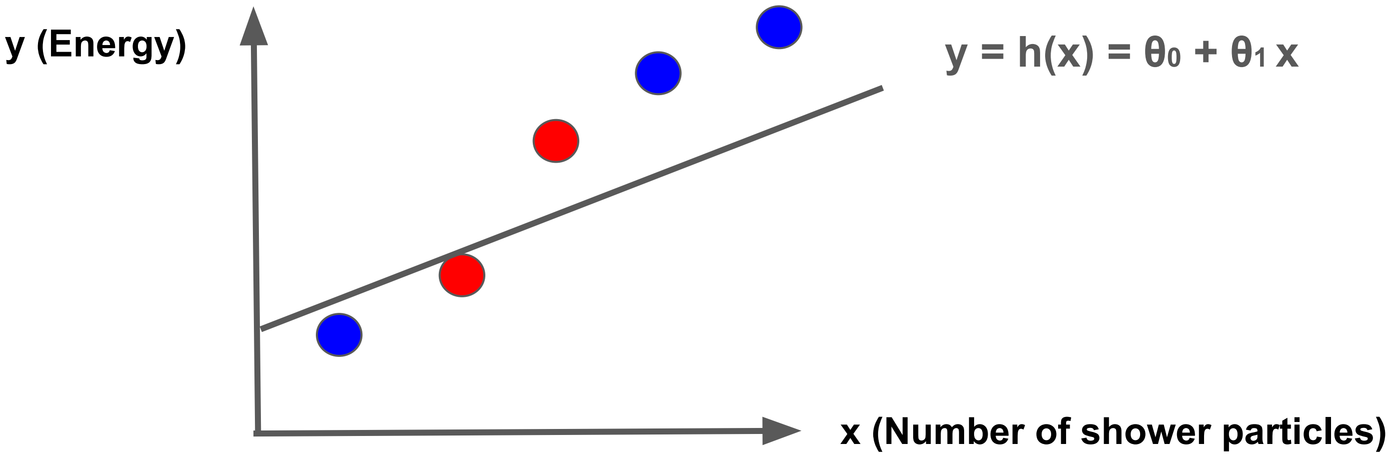
Slope θ_1



Regularization

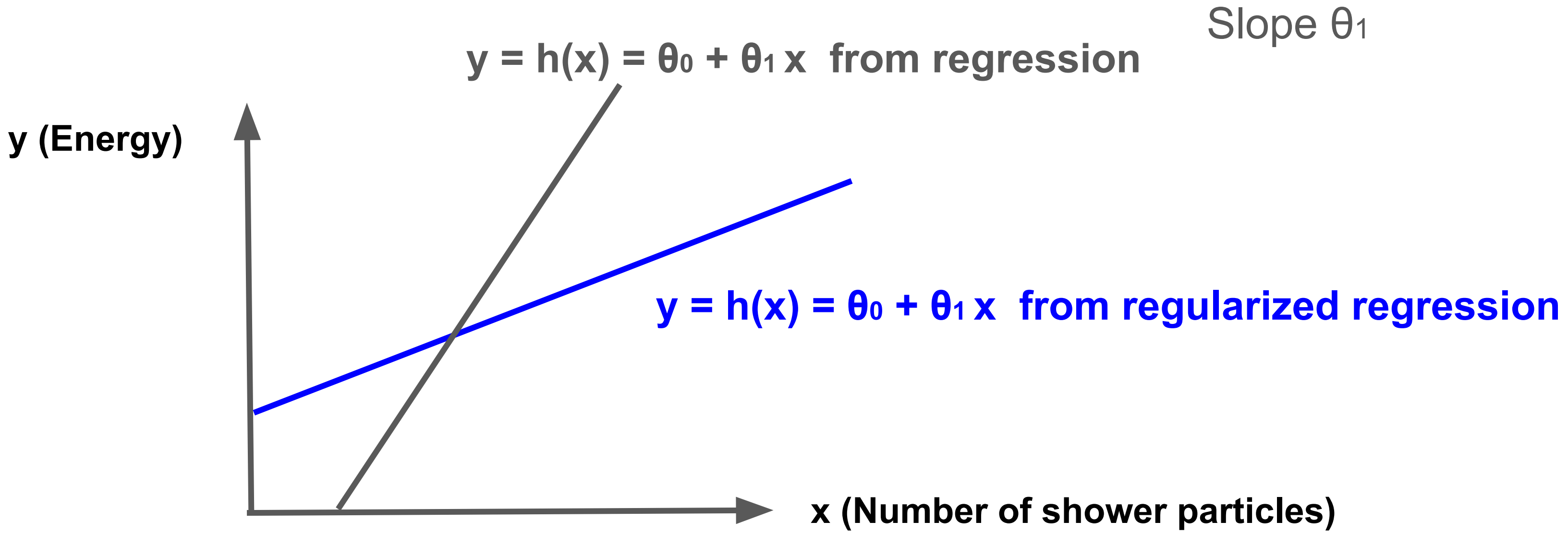
- Regularization: Minimize $\Sigma_i(h(x_i)-y_i)^2$ + $\boldsymbol{\lambda\theta_1^2}$ **λ:**hyperparameter

Slope θ_1



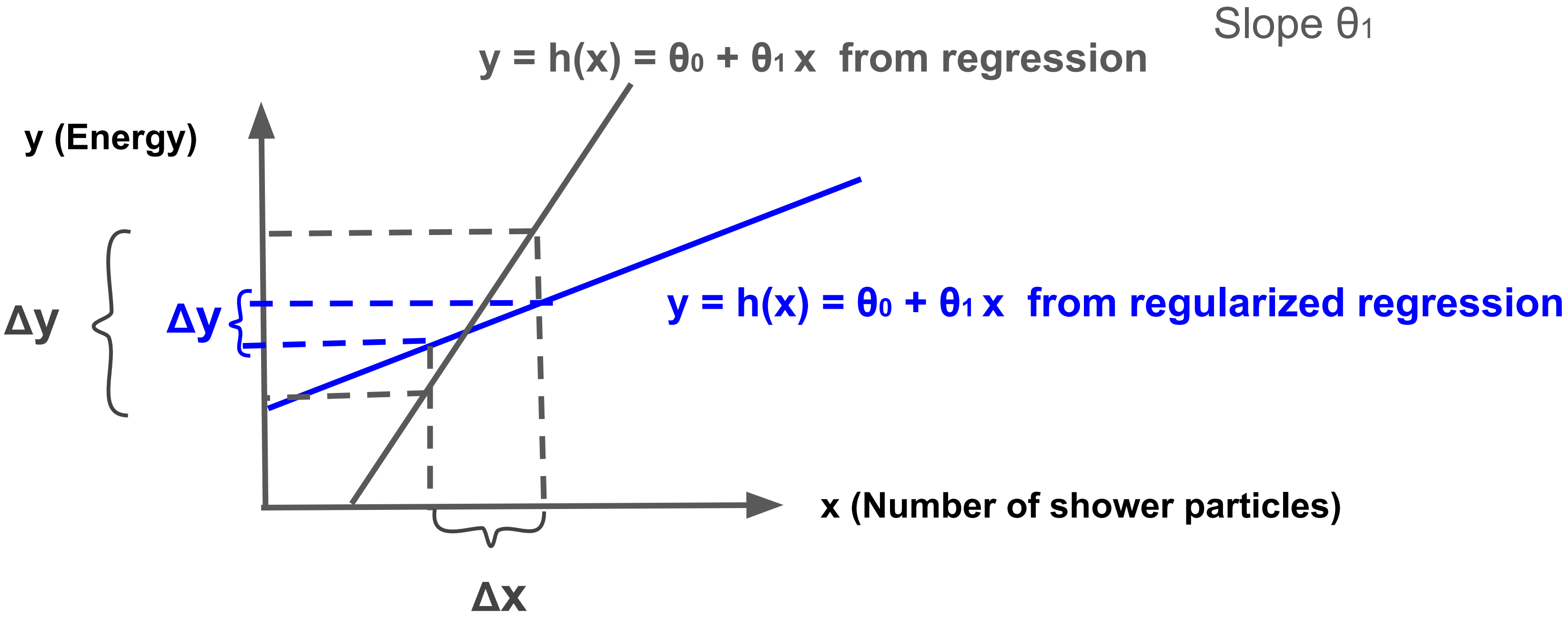
Regularization

- Regularization: Minimize $\Sigma_i(h(x_i)-y_i)^2 + \boldsymbol{\lambda\theta_1^2}$ **λ:**hyperparameter



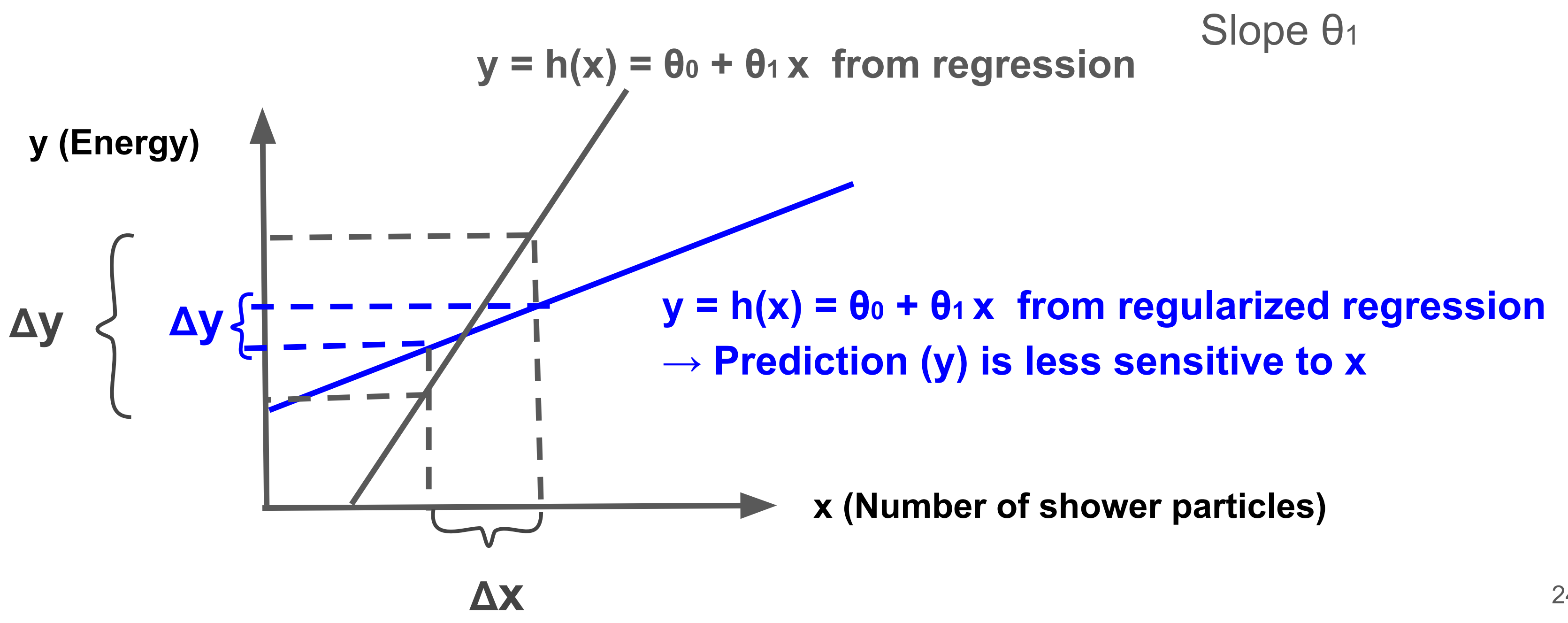
Regularization

- Regularization: Minimize $\Sigma_i(h(x_i)-y_i)^2$ + $\boldsymbol{\lambda\theta_1^2}$ **λ:**hyperparameter



Regularization

- Regularization: Minimize $\Sigma_i(h(x_i)-y_i)^2$ + $\boldsymbol{\lambda\theta_1^2}$ **λ:**hyperparameter

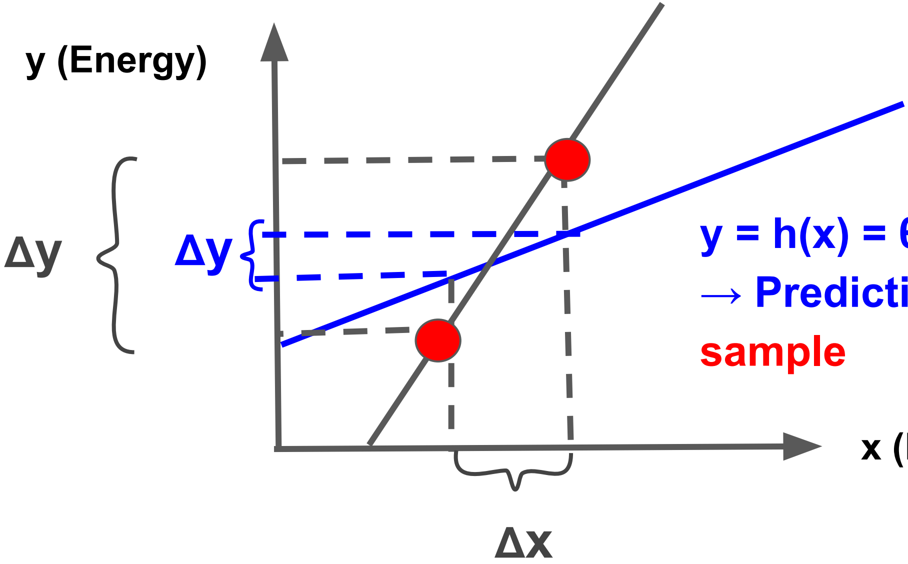


Regularization

- Regularization: Minimize $\Sigma_i(h(x_i)-y_i)^2$ + $\boldsymbol{\lambda\theta_1^2}$ **λ:**hyperparameter

Slope θ_1

$y = h(x) = \theta_0 + \theta_1 x$ from regression



x (Number of shower particles)

$y = h(x) = \theta_0 + \theta_1 x$ from regularized regression
→ Prediction (y) is less sensitive to the training sample

Regularization

- Regularization:
  Minimize $\Sigma_i(h(x_i)-y_i)^2$ (with θ_0)+ $\boldsymbol{\lambda}\Sigma_i\boldsymbol{\theta_i}^{\mathbf{2}}$ (from i=1, i.e. without θ_0)
  (Do not penalize the overall constant θ_0)

- **λ:hyperparameter** chosen with an independent **validation** sample
  → Then apply on another independent testing sample to evaluate the performance

- Ridge regression: $\boldsymbol{\lambda}\Sigma_i\boldsymbol{\theta_i}^{\mathbf{2}}$ (from i=1)
  Lasso regression: $\boldsymbol{\lambda}\Sigma_i|\boldsymbol{\theta_i}|$ (from i=1)

Summary-I

- Linear regression: Fit the training sample with a line by minimizing the sum of squares $\Sigma_i(h(x_i)-y_i)^2$
- Linear regression can be generalized to cases with multivariable (more than one feature) or with polynomials.

Summary-II

- When there are too many features one may overfit the training sample
    - Consequence: Cannot generalize to test samples

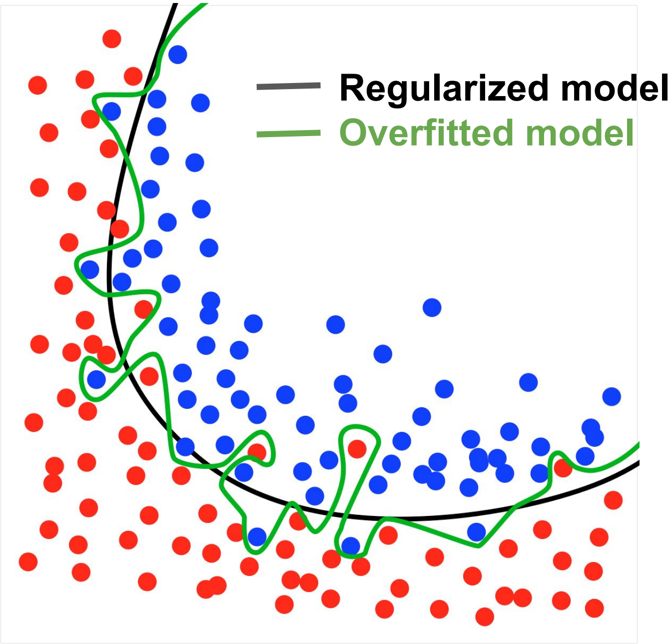
- Regularization: Add a penalty for the size of the parameters to make the prediction less sensitive to the training sample

- E.g. Ridge regression: minimize $\Sigma_i(h(x_i)-y_i)^2 + \lambda\Sigma_i\theta_i^2$ (from i=1)
    - The hyperparameter λ can be tuned with validation samples

Addendum

- Overfitting may happen for classification too!
  → Sometimes called “overtraining”

- Similarly one can ‘regularize’ the classification algorithm. We’ll touch upon the relevant ideas later.



Source: Wikipedia