Introduction to Machine Learning
Instructor: Prof. Abir De
Supervised vs. Unsupervised Learning and Method of Least Squares

Supervised vs Unsupervised

Task: Suppose you had a basket and it is fulled with some fresh fruits your task is to arrange the same type fruits at one place. Suppose the fruits are apple, banana, cherry, grape

Case: 1

- You already know: Shape (parametrize shape?), Color
- **Train data**: Pre-classified data
- Goal: Learn from the pre-classified data and predict on new unclassified fruits.
- This type of learning is called as **supervised learning**.

Case 2:

- In this case, you know nothing about the fruits, you are seeing them for the first time!
- How will you arrange fruits of the same type together?
- One approach is to consider various characteristics of a fruit and divide them on the basis of that.
- Suppose you divide the fruits on the basis of *color* first.
  - ...
  - ...
- Now you take another physical characteristic, size. The grouping will then be:
  - ...
  - ...
  - ..
  - ...
- ..

Case 2:

- In this case, you know nothing about the fruits, you are seeing them for the first time!
- How will you arrange fruits of the same type together?
- One approach is to consider various characteristics of a fruit and divide them on the basis of that.
- Suppose you divide the fruits on the basis of *color* first.
  - **Red Color Group**: Apples and cheery
  - **Green Color Group**: Bananas and grapes
- Now you take another physical characteristic, size. The grouping will then be:
  - **Red color and big size**: Apple
  - **Red color and small size**: Cheery
  - **Green color and big Size**: Banana
  - **Green color and small Size**: Grapes
- This type of learning is **unsupervised learning**

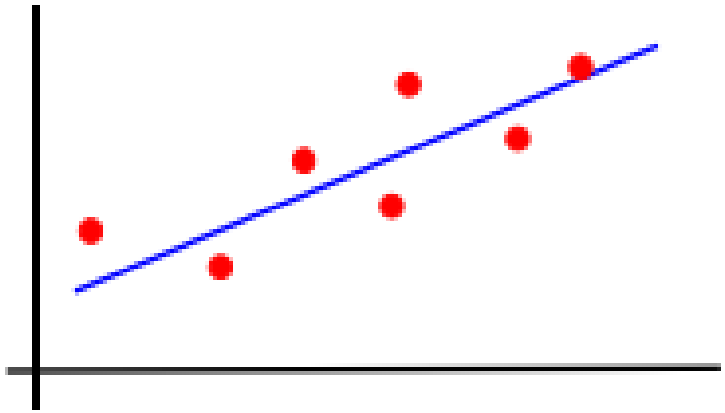
Supervised Learning

Unsupervised Learning

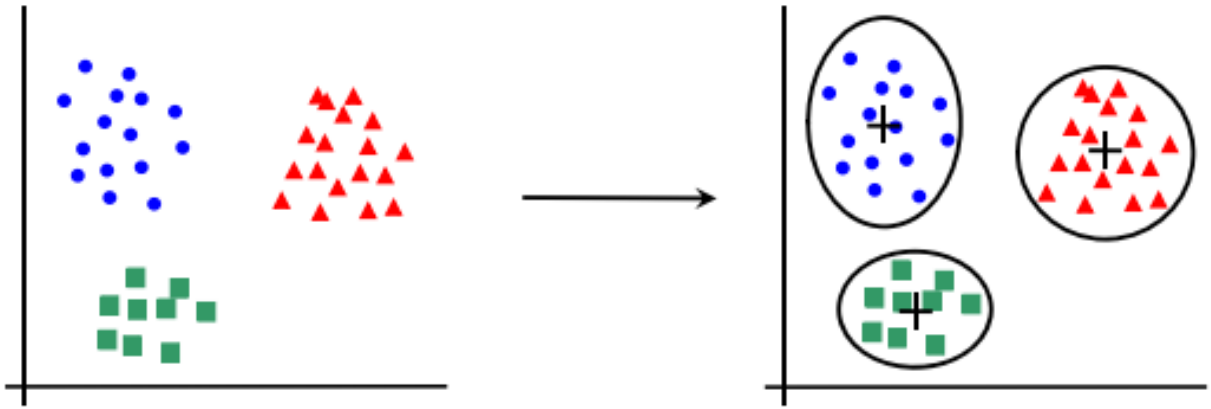
- In supervised learning, the desired outputs are provided which are used to train the machine whereas in unsupervised learning no desired outputs are provided, instead the data is analysed and studied through clustering, mining associations, reduce dimensionality, *etc.* into different classes

Three Canonical Learning Problems

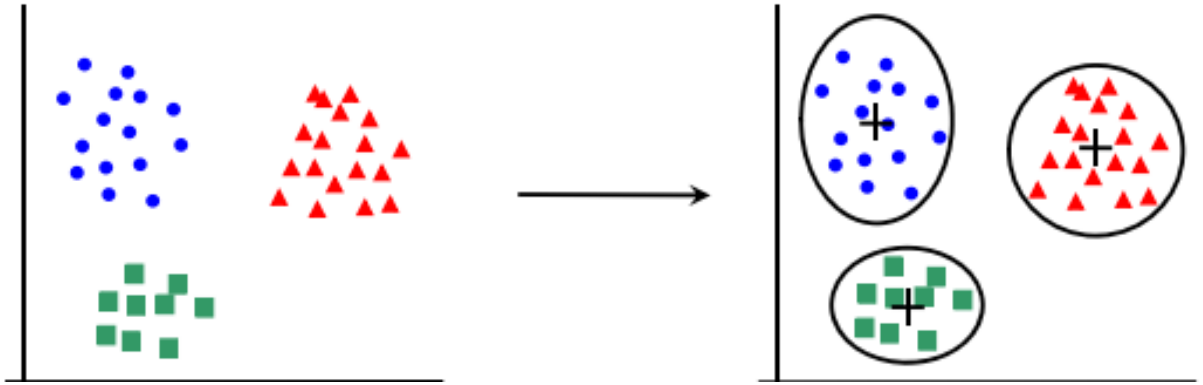
1. Regression - Supervised
   - Estimate parameters, e.g. least square fit



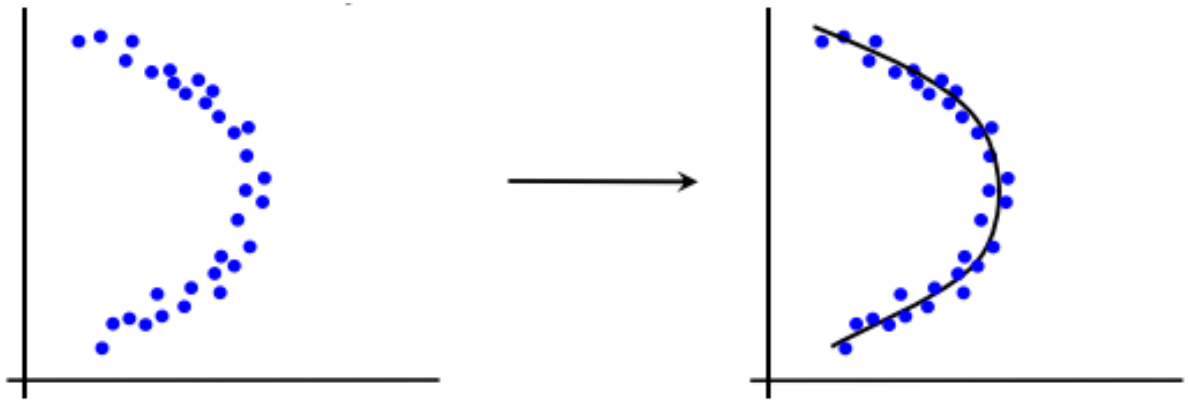
2. Classification - Supervised
   - estimate class, eq handwritten digit classification



3. Unsupervised Learning - model the data
   - clustering



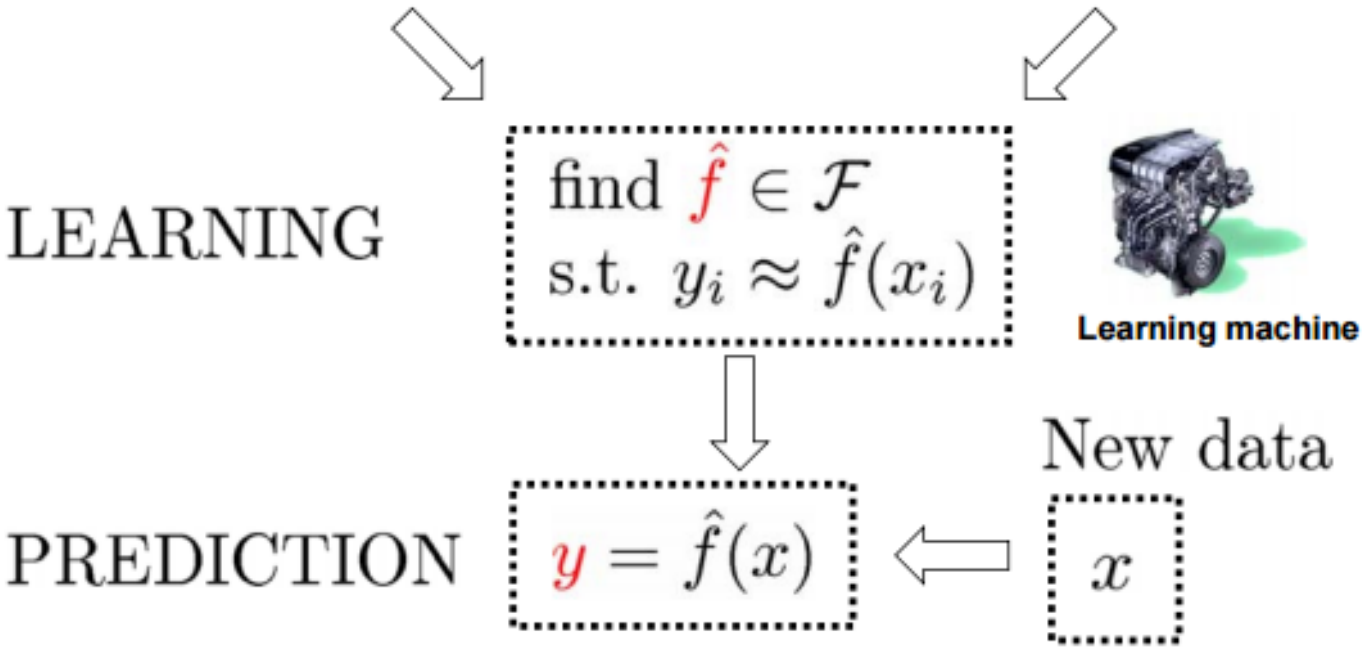
   - dimentionality reduction



Supervised Learning

Functions F Training Data

$f : X \to Y$ $\{ (x^i, y^i) \in X * Y \}$



Next ....

We will start with linear regression and least square method to calculate parameters for linear regression problems.

Recap

- **Machine Learning in general**
  - Supervised Learning
  - Unsupervised Learning
  - Applications and examples
- **Canonical Learning Problems**
  - Regression Supervised
  - Classification Supervised
  - Unsupervised modeling of data

Agenda

- What is data?
  - Noise in data
- How to predict?
  - Fitting a curve
  - Error measurement
  - Minimizing Error
- Method of Least Squares

What is data?

- For us, data is the information about the problem, you are solving using ML, in quantized form
- This data can be from any source, some examples are
  - Prices of stock and stock indexes such as BSE or Nifty
  - Prices of house, area and size of the house
  - Temperature of a place, latitude, longitude and time of year
- The objective of ML is to predict or classify something using the given data
- Hence, one or more than one parameters of the data must also represent the output of our program

Noise in Data

- Data in real life problems are generally collected through surveys
- And surveys may have random human errors
- Hence most methods we will be using deals with expectations as they minimize the effect of error in our predictions
- It is better to find outliers and clean data in the first step. This is known as data cleansing

Example dataset for this lecture

- For this lecture we will consider variation of cost of the house with the area of the house
- In this example we want to find a pattern or curve which this dataset follows, hence predict the price for any value of area

Figure: House purchase data - for illustration purpose only

How to predict?

- Curve fitting is the process of constructing a curve, or mathematical function, that has the best fit to a series of data points, possibly subject to constraints. - Wikipedia
- Thus we need a critera to compare two curves on a dataset
- We describe an error function F(f, D) which takes a curve f and dataset D as input and returns a real number
- Error function must be such that it can capture how worse is our

Example

- Consider the example below where we have two curves on our dataset defined by blue(f_b) and red(f_r) line respectively. We want to find which is the better fit.

Figure: House purchase data curve fit

Question

What are some options for F(f,D)?
Hint: Measurement of difference from original value.

Examples of F

- $\sum_D f(x_i) - y_i$
- $\sum_D |f(x_i) - y_i|$
- $\sum_D (f(x_i) - y_i)^2$
- $\sum_D (f(x_i) - y_i)^3$
- and many more

Question

What F do you think can give us best fit curve and why?
Hint: Intuition of distances.

Squared Error

$$\sum_{D}(f(x_i) - y_i)^2$$

- To find the best fit curve we try to minimize the above function
- It is continuous and differentiable
- It can ve visualized as square of Euclidean distance between predicted points and actual points
- How we can perform mathematical treatment over this function will be covered in further lectures.
- This mathematical treatment is known as method of least squares. Can you find the reason why it is known as "Method of Least Squares"?
  Hint: Unit square is the basic unit in a graph.

Regression, More Formally

- Formal Definition
- Types of Regression
- Geometric Interpretation of least square solution

Linear Regression as a canonical example

- **Optimization** (Formally deriving least Square Solution)
- **Regularization** (Ridge Regression, Lasso), **Bayesian Interpretation** (Bayesian Linear Regression)
- **Non-parametric estimation** (Local linear regression),
- **Non-linearity through Kernels** (Support Vector Regression)

Linear Regression with Illustration

- Regression is about learning to predict a set of output variables (*dependent variables*) as a function of a set of input variables (*independent variables*)
- Example
  - A company wants to determine how much it should spend on T.V commercials to increase sales to a desired level y^*
  - **Basis?**

Linear Regression with Illustration

- Regression is about learning to predict a set of output variables (*dependent variables*) as a function of a set of input variables (*independent variables*)
- Example
  - A company wants to determine how much it should spend on T.V commercials to increase sales to a desired level y^*
  - **Basis?** It has previous observations of the form ¡x_i,y_i¿,
    - x_i is an instance of money spent on advertisements and y_i was the corresponding observed sale figure

Linear Regression with Illustration

- Regression is about learning to predict a set of output variables (*dependent variables*) as a function of a set of input variables (*independent variables*)
- Example
  - A company wants to determine how much it should spend on T.V commercials to increase sales to a desired level y^*
  - **Basis?** It has previous observations of the form ¡x_i,y_i¿,
    - x_i is an instance of money spent on advertisements and y_i was the corresponding observed sale figure
  - Suppose the observations support the following linear approximation

$$y = \beta_0 + \beta_1 * x \tag{1}$$

    Then $x^* = \frac{y^* - \beta_0}{\beta_1}$ can be used to determine the money to be spent
- **Estimation** for Regression: Determine appropriate value for β_0 and β_1 from the past observations

Linear Regression with Illustration

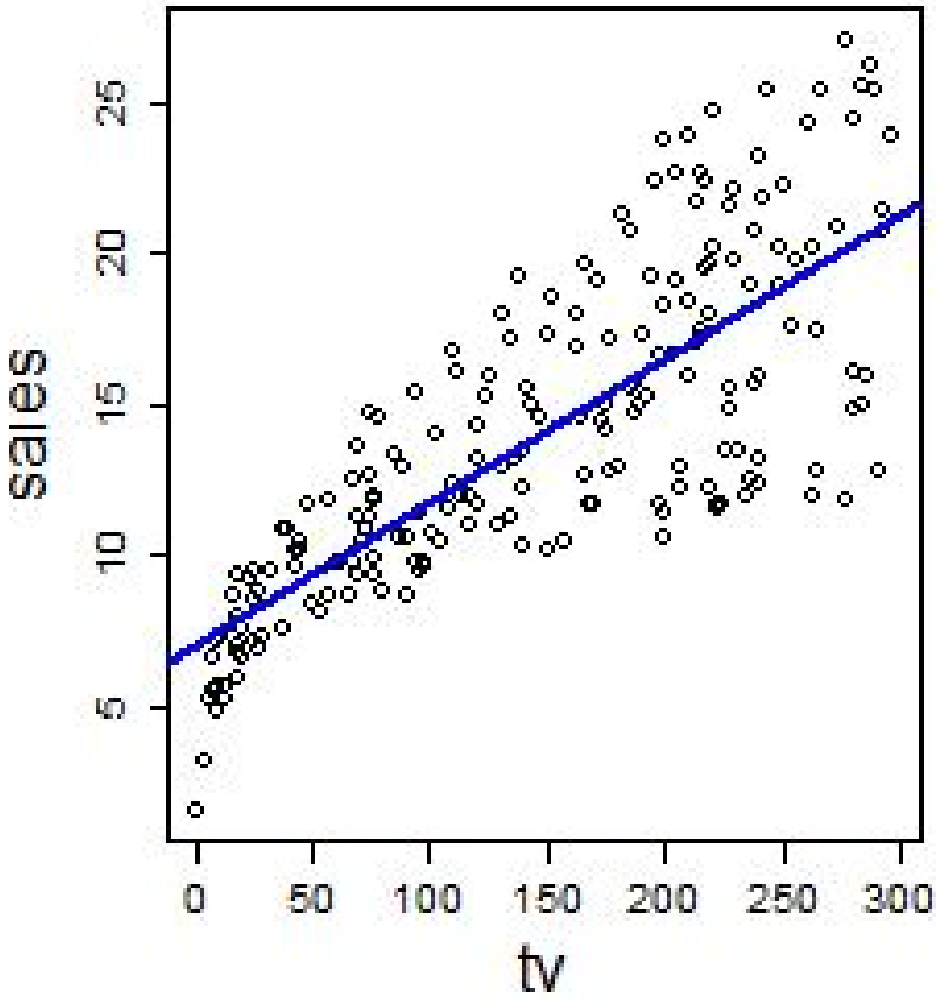


Figure: Linear regression on T.V advertising vs sales figure

What will it mean to have sales as a non-linear function of investment in advertising?

Basic Notation

- Data set: $\mathcal{D} =< x_1, y_1 >, .., < x_m, y_m >$
  - Notation (used throughout the course)
  - m = number of training examples
  - $x's$ = input/independent variables
  - $y's$ = output/dependent/'target' variables
  - (x, y) - a single training example
  - (x_j, y_j) - specific example (j^{th} training example)
  - j is an index into the training set
- ϕ_i's are the attribute/basis functions, and let

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & ..... & \phi_p(x_1) \\ . & & & \\ . & & & \\ \phi_1(x_m) & \phi_2(x_m) & ..... & \phi_p(x_m) \end{bmatrix} \tag{2}$$

-

$$y = \begin{bmatrix} y_1 \\ . \\ y_m \end{bmatrix} \tag{3}$$

Formal Definition

- **General Regression problem**: Determine a function f^* such that $f^*(x)$ is the best predictor for y, with respect to $\mathcal{D}$:

$$f^* = \underset{f \in F}{\operatorname{argmin}}\ E(f, \mathcal{D})$$

  Here, F denotes the class of functions over which the error minimization is performed

- **Parametrized Regression problem**: Need to determine parameters w for the function $f\left(\phi(\mathrm{x}), \mathrm{w}\right)$ which minimize our error function $E\left(f\left(\phi(\mathrm{x}), \mathrm{w}\right), \mathcal{D}\right)$

$$\mathrm{w}^* = \underset{\mathrm{w}}{\operatorname{argmin}} \left\langle E\left(f\left(\phi(\mathrm{x}), \mathrm{w}\right), \mathcal{D}\right)\right\rangle$$

$$\mathrm{w}^* = \underset{\mathrm{w}}{\operatorname{argmin}} \left\{ \sum_{j=1}^{m} \left(f\left(\mathrm{x_j}, \mathrm{w}\right) - y_j\right)^2 \right\}$$

Types of Regression

- Classified based on the function class and error function
- F is space of linear functions $f(\phi(\mathrm{x}), \mathrm{w}) = \mathrm{w}^T \phi(\mathrm{x}) + b \Longrightarrow$ Linear Regression
  - Problem is then to determine w^* such that,

$$\mathrm{w}^* = \underset{\mathrm{w}}{\operatorname{argmin}}\ E(\mathrm{w}, \mathcal{D}) \tag{4}$$

Types of Regression (contd.)

- **Ridge Regression:** A shrinkage parameter (regularization parameter) is added in the error function to reduce discrepancies due to variance
- **Logistic Regression:** Models conditional probability of dependent variable given independent variables and is extensively used in classification tasks

$$f(\phi(\mathrm{x}), \mathrm{w}) = \log \frac{\Pr(\mathrm{y}|\mathrm{x})}{1 - \Pr(\mathrm{y}|\mathrm{x})} = b + \mathrm{w}^T * \phi(\mathrm{x}) \qquad (5)$$

- Lasso regression, Stepwise regression and several others

Least Square Solution

- Form of $E()$ should lead to accuracy and tractability
- The squared loss is a commonly used error/loss function. It is the sum of squares of the differences between the actual value and the predicted value

$$E(f, \mathcal{D}) = \sum_{j=1}^{m} (f(x_j) - y_j)^2 \tag{6}$$

- The least square solution for linear regression is obtained as

$$\mathrm{w}^* = \underset{\mathrm{w}}{\operatorname{argmin}} \sum_{j=1}^{m} \left(\sum_{i=1}^{p} (w_i \phi_i(x_j) - y_j)^2 \right) \tag{7}$$

- The minimum value of the squared loss is zero
- If zero were attained at w^*, we would have .....

- The minimum value of the squared loss is zero
- If zero were attained at w^*, we would have $\forall u, \phi^T(x_u)\mathrm{w}^* = y_u$, or equivalently $\Phi\mathrm{w}^* = \mathrm{y}$, where

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \dots & \phi_p(x_1) \\ \dots & \dots & \dots \\ \phi_1(x_m) & \dots & \phi_p(x_m) \end{bmatrix}$$

and

$$\mathrm{y} = \begin{bmatrix} y_1 \\ \dots \\ y_m \end{bmatrix}$$

- It has a solution if y is in the column space (the subspace of R^m formed by the column vectors) of Φ

- The minimum value of the squared loss is zero
- If zero were NOT attainable at w^*, what can be done?

Geometric Interpretation of Least Square Solution

- Let y^* be a solution in the column space of Φ
- The least squares solution is such that the distance between y^* and y is minimized
- Therefore.....

Geometric Interpretation of Least Square Solution

- Let y^* be a solution in the column space of Φ
- The least squares solution is such that the distance between y^* and y is minimized
- Therefore, the line joining y^* to y should be orthogonal to the column space

$$\Phi w = y^* \tag{8}$$

$$(y - y^*)^T \Phi = 0 \tag{9}$$

$$(y^*)^T \Phi = (y)^T \Phi \tag{10}$$

$$(\Phi w)^T \Phi = y^T \Phi \tag{11}$$

$$w^T \Phi^T \Phi = y^T \Phi \tag{12}$$

$$\Phi^T \Phi w = \Phi^T y \tag{13}$$

$$w = (\Phi^T \Phi)^{-1} \Phi^T y \tag{14}$$

- Here $\Phi^T \Phi$ is invertible if and only if Φ has full column rank

Proof?

Theorem : $\Phi^T\Phi$ is invertible if and only if Φ is full column rank
Proof :
Given that Φ has full column rank and hence columns are linearly independent, we have that $\Phi x = 0 \Rightarrow x = 0$
Assume on the contrary that $\Phi^T\Phi$ is non invertible. Then $\exists x \neq 0$ such that $\Phi^T\Phi x = 0$

$$\begin{aligned} &\Rightarrow x^T\Phi^T\Phi x = 0 \\ &\Rightarrow (\Phi x)^T\Phi x = 0 \\ &\quad\Rightarrow \Phi x = 0 \end{aligned}$$

This is a contradiction. Hence $\Phi^T\Phi$ is invertible if Φ is full column rank
If $\Phi^T\Phi$ is invertible then $\Phi x = 0$ implies $(\Phi^T\Phi x) = 0$, which in turn implies $x = 0$, This implies Φ has full column rank if $\Phi^T\Phi$ is invertible. The converse can also be proved similarly.

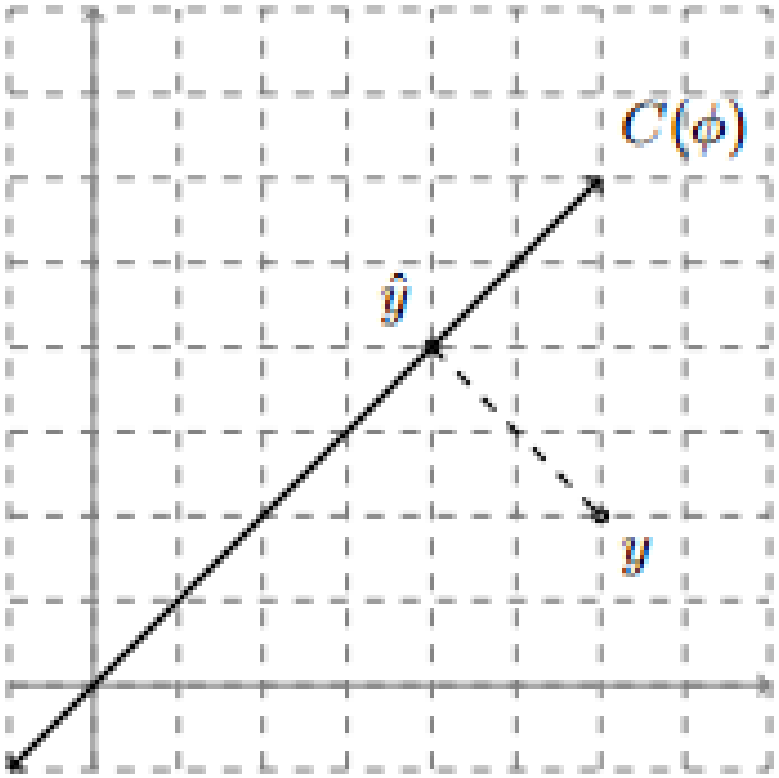


Figure: Least square solution y^* is the orthogonal projection of y onto column space of Φ