Linear Methods 2

Shivaram Kalyanakrishnan

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

February 2023

This Lecture

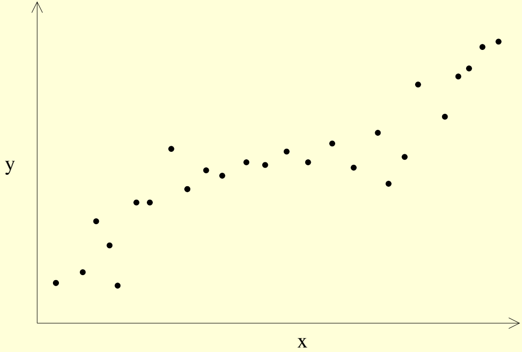
- Convergence of Perceptron Learning Algorithm
- Linear regression

This Lecture

- Convergence of Perceptron Learning Algorithm
- Linear regression

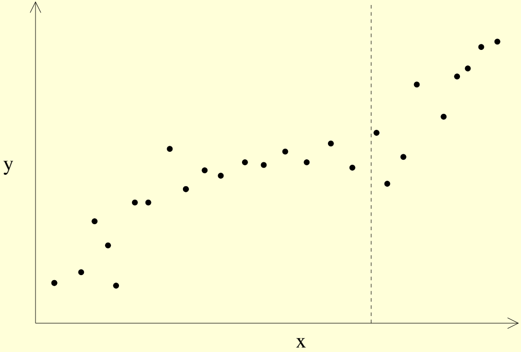
Regression problem

- Illustration with $d = 1$.



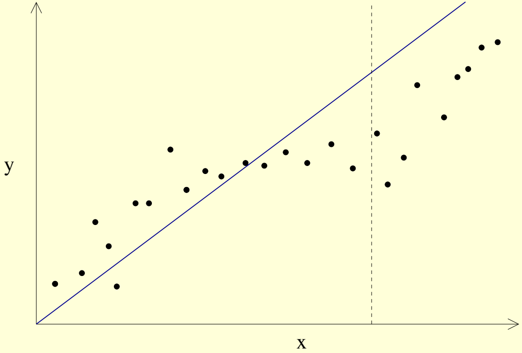
Regression problem

- Illustration with $d = 1$.
- Given arbitrary x, predict its y-value.



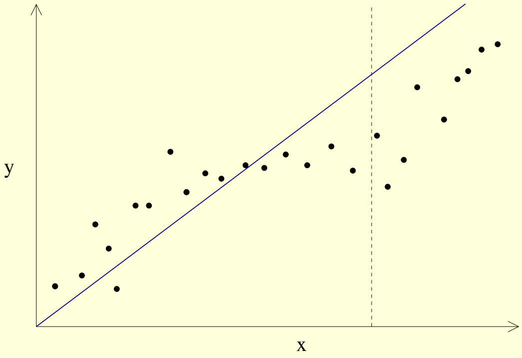
Regression problem

- Illustration with $d = 1$.
- Given arbitrary x, predict its y-value.
- Assume $y = wx$ (linear model); w is the parameter to learn.



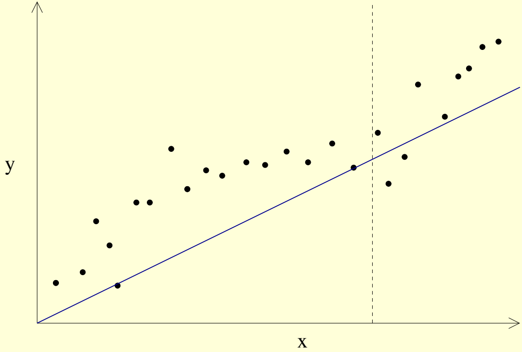
Regression problem

- Illustration with $d = 1$.
- Given arbitrary x, predict its y-value.
- Assume $y = wx$ (linear model); w is the parameter to learn.
- What is the optimal choice of w?



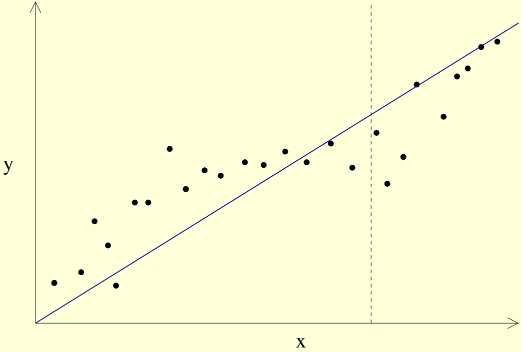
Regression problem

- Illustration with $d = 1$.
- Given arbitrary x, predict its y-value.
- Assume $y = wx$ (linear model); w is the parameter to learn.
- What is the optimal choice of w?



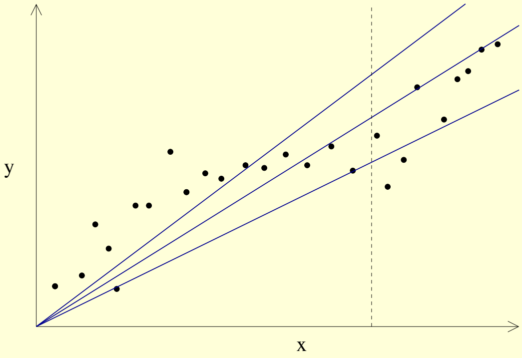
Regression problem

- Illustration with $d = 1$.
- Given arbitrary x, predict its y-value.
- Assume $y = wx$ (linear model); w is the parameter to learn.
- What is the optimal choice of w?



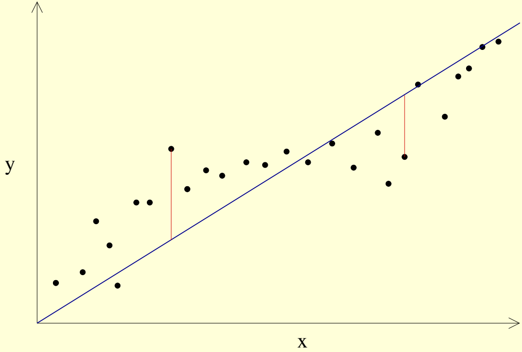
Regression problem

- Illustration with $d = 1$.
- Given arbitrary x, predict its y-value.
- Assume $y = wx$ (linear model); w is the parameter to learn.
- What is the optimal choice of w?



Regression problem

- Illustration with $d = 1$.
- Given arbitrary x, predict its y-value.
- Assume $y = wx$ (linear model); w is the parameter to learn.
- What is the optimal choice of w?



- **Idea**: the optimal w (call it w_{opt}) must give a line from which deviations are small.

Formally...

- In general w is d-dimensional.

Formally...

- In general w is d-dimensional.
- Define

$$E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2,$$

Formally...

- In general w is d-dimensional.
- Define

$$E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2,$$

$$w_{\text{opt}} = \underset{w}{\operatorname{argmin}}\, E(w).$$

Formally...

- In general w is d-dimensional.
- Define

$$E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2,$$

$$w_{\text{opt}} = \underset{w}{\operatorname{argmin}}\, E(w).$$

- $E(w)$ is a “sum of squared errors” (SSE).

Formally...

- In general w is d-dimensional.
- Define

$$E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2,$$

$$w_{\text{opt}} = \underset{w}{\text{argmin}}\, E(w).$$

- $E(w)$ is a "sum of squared errors" (SSE).

- How to find w_{opt}?

Formally...

- In general w is d-dimensional.
- Define

$$E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2,$$

$$w_{\mathrm{opt}} = \underset{w}{\operatorname{argmin}}\, E(w).$$

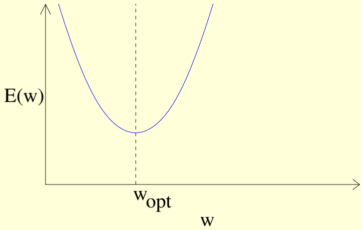
- $E(w)$ is a "sum of squared errors" (SSE).

- How to find w_{opt}?

- We give three methods!

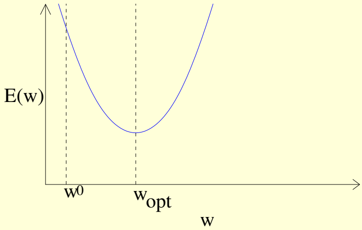
Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).



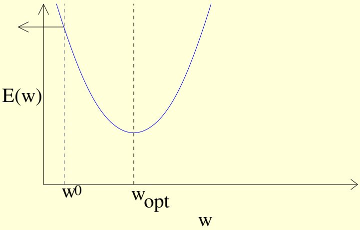
Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).
- Start with an initial guess w^0.



Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).
- Start with an initial guess w^0.
- Calculate $\frac{dE(w)}{dw}$ at w_0. It conveys how $E(w)$ varies with w at w_0.

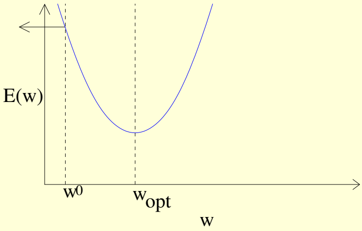


Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).
- Start with an initial guess w^0.
- Calculate $\frac{dE(w)}{dw}$ at w_0. It conveys how $E(w)$ varies with w at w_0.
- Move in the direction that $E(w)$ is decreasing:

$$w^1 \leftarrow w^0 - \alpha \left(\frac{dE(w)}{dw} \right)_{w=w^0},$$

where α is a learning rate—say 10^{-4}.

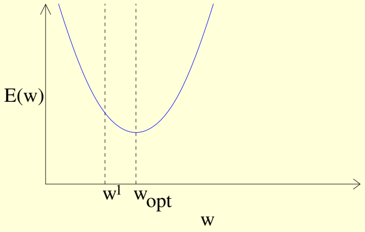


Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).
- Start with an initial guess w^0.
- Calculate $\frac{dE(w)}{dw}$ at w_0. It conveys how $E(w)$ varies with w at w_0.
- Move in the direction that $E(w)$ is decreasing:

$$w^1 \leftarrow w^0 - \alpha \left(\frac{dE(w)}{dw} \right)_{w=w^0},$$

where α is a learning rate—say 10^{-4}.

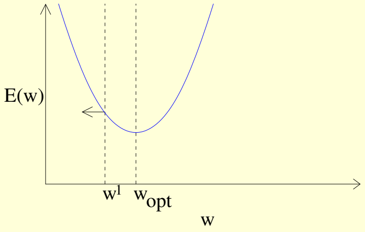


Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).
- Start with an initial guess w^0.
- Calculate $\frac{dE(w)}{dw}$ at w_0. It conveys how $E(w)$ varies with w at w_0.
- Move in the direction that $E(w)$ is decreasing:

$$w^1 \leftarrow w^0 - \alpha \left(\frac{dE(w)}{dw} \right)_{w=w^0},$$

where α is a learning rate—say 10^{-4}.
- Continue in the same way from w^1!

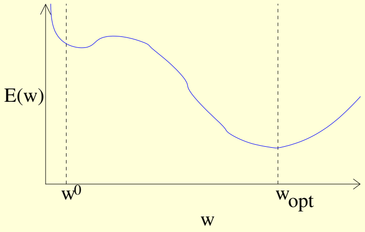


Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).
- Start with an initial guess w^0.
- Calculate $\frac{dE(w)}{dw}$ at w_0. It conveys how $E(w)$ varies with w at w_0.
- Move in the direction that $E(w)$ is decreasing:

$$w^1 \leftarrow w^0 - \alpha \left(\frac{dE(w)}{dw} \right)_{w=w^0},$$

  where α is a learning rate—say 10^{-4}.
- Continue in the same way from w^1!
- But what if $E(w)$ looks like this?!



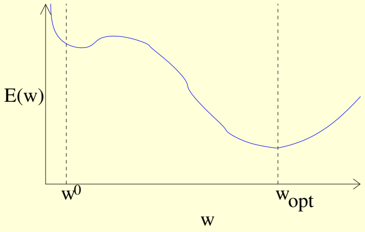
Method 1: Gradient Descent

- Observe that $E(w)$ is a differentiable function of w.
- Illustration below uses $d = 1$ (so w is a scalar).
- Start with an initial guess w^0.
- Calculate $\frac{dE(w)}{dw}$ at w_0. It conveys how $E(w)$ varies with w at w_0.
- Move in the direction that $E(w)$ is decreasing:

$$w^1 \leftarrow w^0 - \alpha \left(\frac{dE(w)}{dw} \right)_{w=w^0},$$

where α is a learning rate—say 10^{-4}.

- Continue in the same way from w^1!
- But what if $E(w)$ looks like this?!
- Can't happen. With a linear model and SSE, $E(w)$ is guaranteed to be convex, with unique minimum.



In Higher Dimension d

- For $d = 1$, we used $\frac{dE(w)}{dw}$, which is a scalar, to improve w.

In Higher Dimension d

- For $d = 1$, we used $\frac{dE(w)}{dw}$, which is a scalar, to improve w.
- For $d > 1$, this quantity generalises to the d-dimensional gradient vector $\nabla_w E(w)$. If we take $w = (w_1, w_2, \ldots, w_d)$, then

$$\nabla_w E(w) = \left(\frac{\partial E(w)}{\partial w_1}, \frac{\partial E(w)}{\partial w_2}, \ldots, \frac{\partial E(w)}{\partial w_d} \right).$$

In Higher Dimension d

- For $d = 1$, we used $\frac{dE(w)}{dw}$, which is a scalar, to improve w.
- For $d > 1$, this quantity generalises to the d-dimensional gradient vector $\nabla_w E(w)$. If we take $w = (w_1, w_2, \ldots, w_d)$, then

$$\nabla_w E(w) = \left(\frac{\partial E(w)}{\partial w_1}, \frac{\partial E(w)}{\partial w_2}, \ldots, \frac{\partial E(w)}{\partial w_d} \right).$$

- Since $E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2$, we get $\nabla_w E(w) = -2 \sum_{i=1}^{n} (y^i - w \cdot x^i) x^i$.

In Higher Dimension d

- For $d = 1$, we used $\frac{dE(w)}{dw}$, which is a scalar, to improve w.
- For $d > 1$, this quantity generalises to the d-dimensional gradient vector $\nabla_w E(w)$. If we take $w = (w_1, w_2, \ldots, w_d)$, then

$$\nabla_w E(w) = \left(\frac{\partial E(w)}{\partial w_1}, \frac{\partial E(w)}{\partial w_2}, \ldots, \frac{\partial E(w)}{\partial w_d} \right).$$

- Since $E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2$, we get $\nabla_w E(w) = -2 \sum_{i=1}^{n} (y^i - w \cdot x^i) x^i$.
- We perform the update $w^{i+1} \leftarrow w^i - \alpha (\nabla_w E(w))_{w = w^i}$ using this formula.

In Higher Dimension d

- For $d = 1$, we used $\frac{dE(w)}{dw}$, which is a scalar, to improve w.

- For $d > 1$, this quantity generalises to the d-dimensional gradient vector $\nabla_w E(w)$. If we take $w = (w_1, w_2, \ldots, w_d)$, then

$$\nabla_w E(w) = \left(\frac{\partial E(w)}{\partial w_1}, \frac{\partial E(w)}{\partial w_2}, \ldots, \frac{\partial E(w)}{\partial w_d} \right).$$

- Since $E(w) = \sum_{i=1}^{n} (y^i - w \cdot x^i)^2$, we get $\nabla_w E(w) = -2 \sum_{i=1}^{n} (y^i - w \cdot x^i) x^i$.

- We perform the update $w^{i+1} \leftarrow w^i - \alpha (\nabla_w E(w))_{w=w^i}$ using this formula.

- This process will eventually reach w_{opt}.

Method 2: Direct Formula

- Create a “data matrix” X with n rows and d columns:
  X_{ij} is the j-th feature of the i-th data point (x^i).

Method 2: Direct Formula

- Create a "data matrix" X with n rows and d columns:
  X_{ij} is the j-th feature of the i-th data point (x^i).

- Put the labels in an $n \times 1$ matrix (a.k.a. n-vector) Y: that is, $Y_i = y^i$.

Method 2: Direct Formula

- Create a "data matrix" X with n rows and d columns:
  X_{ij} is the j-th feature of the i-th data point (x^i).
- Put the labels in an $n \times 1$ matrix (a.k.a. n-vector) Y: that is, $Y_i = y^i$.
- We need d weights for w_{opt};

Method 2: Direct Formula

- Create a "data matrix" X with n rows and d columns:
  X_{ij} is the j-th feature of the i-th data point (x^i).

- Put the labels in an $n \times 1$ matrix (a.k.a. n-vector) Y: that is, $Y_i = y^i$.

- We need d weights for w_{opt}; obtain them as a vector

$$w_{\text{opt}} = (X^\top X)^{-1} X^\top Y.$$

Method 3: Use a library!

```
#scikit-learn code looks something like this.
lm = Ridge()
lm.fit(X, Y)
Ynew = lm.predict(Xnew)
```

Questions

- Why did we define $E(w)$ as a sum of squared errors? Are there alternatives?

Questions

- Why did we define $E(w)$ as a sum of squared errors? Are there alternatives?

  Yes. For example, we could have defined

  $$E(w) = \sum_{i=1}^{n} |y^i - w \cdot x^i|.$$

  But this formulation is not as easy to solve.

Questions

- Why did we define $E(w)$ as a sum of squared errors? Are there alternatives?
  Yes. For example, we could have defined
  $$E(w) = \sum_{i=1}^{n} |y^i - w \cdot x^i|.$$
  But this formulation is not as easy to solve.
- Is linear regression used commonly in practice?

Questions

- Why did we define $E(w)$ as a sum of squared errors? Are there alternatives?

  Yes. For example, we could have defined

  $$E(w) = \sum_{i=1}^{n} |y^i - w \cdot x^i|.$$

  But this formulation is not as easy to solve.

- Is linear regression used commonly in practice?

  Yes! And it also forms the basis for several other methods in ML.

References

- Note on Perceptron Learning Algorithm (see course page).
- Chapter 7, **A Course in Machine Learning**, Hal Daumé III. Available on-line at `http://ciml.info/`.