# IS-ZC444: ARTIFICIAL INTELLIGENCE

## Lecture-14: Bayesian Network + Regression


**Dr. Kamlesh Tiwari**
Assistant Professor
Department of Computer Science and Information Systems,
BITS Pilani, Pilani, Jhunjhunu-333031, Rajasthan, INDIA

October 31, 2018 (WILP @ BITS-Pilani Jul-Nov 2018)

# Probability

- $P(x, y) = P(x) \times P(y|x)$

# Probability

- $P(x, y) = P(x) \times P(y|x)$
- **Independence** of $x$ and $y$ implies $P(y|x) = P(y)$

# Probability

- $P(x, y) = P(x) \times P(y|x)$
- **Independence** of $x$ and $y$ implies $P(y|x) = P(y)$
- Then $P(x, y) = P(x) \times P(y)$

# Probability

- $P(x, y) = P(x) \times P(y|x)$
- **Independence** of $x$ and $y$ implies $P(y|x) = P(y)$
- Then $P(x, y) = P(x) \times P(y)$
- Bayes Rule

$$P(x|y) = \frac{P(x, y)}{P(y)} = \frac{P(y|x) \times P(x)}{P(y)}$$

# Probability

- $P(x, y) = P(x) \times P(y|x)$
- **Independence** of $x$ and $y$ implies $P(y|x) = P(y)$
- Then $P(x, y) = P(x) \times P(y)$
- Bayes Rule

$$P(x|y) = \frac{P(x, y)}{P(y)} = \frac{P(y|x) \times P(x)}{P(y)}$$

- **Marginal:** distribution of a single variable $x$ can be obtained from a given joint distribution $p(x, y)$ by

$$p(x) = \sum_{y} p(x, y)$$

# Probability

- $P(x, y) = P(x) \times P(y|x)$
- **Independence** of $x$ and $y$ implies $P(y|x) = P(y)$
- Then $P(x, y) = P(x) \times P(y)$
- Bayes Rule

$$P(x|y) = \frac{P(x, y)}{P(y)} = \frac{P(y|x) \times P(x)}{P(y)}$$

- **Marginal:** distribution of a single variable $x$ can be obtained from a given joint distribution $p(x, y)$ by

$$p(x) = \sum_{y} p(x, y)$$

- The process of computing a marginal from a joint distribution is called marginalisation.

$$p(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = \sum_{x_i} p(x_1, x_2, ..., x_n)$$

# Let's see this

$$\sum_{J} \left( p(J|R) \times f(R) \right) =$$

# Let's see this

$$\sum_J \left(p(J|R) \times f(R)\right) = f(R)$$

**Proof:**

$$\begin{aligned}
\sum_J \left(p(J|R) \times f(R)\right) &= \sum_J \left(\frac{p(J,R)}{p(R)} \times f(R)\right) \\
&= \frac{\sum_J \left(p(J,R) \times f(R)\right)}{p(R)} \\
&= \frac{f(R) \times \sum_J p(J,R)}{p(R)} \\
&= \frac{f(R) \times p(R)}{p(R)} \\
&= f(R)
\end{aligned}$$

# Belief network (a graphical model)

- A **belief network** introduces structure into a probabilistic model by using graphs to represent independence assumptions among the variables

# Belief network (a graphical model)

- A **belief network** introduces structure into a probabilistic model by using graphs to represent independence assumptions among the variables
- Independently specifying all the attributed is overkill

# Belief network (a graphical model)

- A **belief network** introduces structure into a probabilistic model by using graphs to represent independence assumptions among the variables
- Independently specifying all the attributed is overkill
- With distribution of $n$ attributes, marginal for one takes $O(2^{n-1})$

# Belief network (a graphical model)

- A **belief network** introduces structure into a probabilistic model by using graphs to represent independence assumptions among the variables
- Independently specifying all the attributed is overkill
- With distribution of $n$ attributes, marginal for one takes $O(2^{n-1})$
- By constraining variable interaction (specifying independence) one can get the form like

$$p(x_1, x_2, ..., x_{100}) = \Pi_{i=1}^{99} \phi(x_i, x_{i+1})$$

# Belief network (a graphical model)

- A **belief network** introduces structure into a probabilistic model by using graphs to represent independence assumptions among the variables
- Independently specifying all the attributed is overkill
- With distribution of $n$ attributes, marginal for one takes $O(2^{n-1})$
- By constraining variable interaction (specifying independence) one can get the form like

$$p(x_1, x_2, ..., x_{100}) = \Pi_{i=1}^{99} \phi(x_i, x_{i+1})$$

- Belief networks are a convenient framework for representing such independence assumptions

# Belief network (a graphical model)

- A **belief network** introduces structure into a probabilistic model by using graphs to represent independence assumptions among the variables
- Independently specifying all the attributed is overkill
- With distribution of $n$ attributes, marginal for one takes $O(2^{n-1})$
- By constraining variable interaction (specifying independence) one can get the form like

$$p(x_1, x_2, ..., x_{100}) = \Pi_{i=1}^{99} \phi(x_i, x_{i+1})$$

- Belief networks are a convenient framework for representing such independence assumptions
- Belief networks are also called as *Bayes' Networks* or *Bayesian Belief Networks*

# Modeling Independencies

One morning Tracey leaves her house and realities that her grass is wet. Is it due to overnight rain or did she forget to turn off the sprinkler last night? Next she notices that the grass of her neighbor, Jack, is also wet.

# Modeling Independencies

One morning Tracey leaves her house and realities that her grass is wet. Is it due to overnight rain or did she forget to turn off the sprinkler last night? Next she notices that the grass of her neighbor, Jack, is also wet.

($R$=1) $\rightarrow$ rain last night,
($S$=1) $\rightarrow$ sprinkler on last night,
($J$=1) $\rightarrow$ Jack's grass is wet,
($T$=1) $\rightarrow$ Traceya's Grass is wet

# Modeling Independencies

One morning Tracey leaves her house and realities that her grass is wet. Is it due to overnight rain or did she forget to turn off the sprinkler last night? Next she notices that the grass of her neighbor, Jack, is also wet.

($R$=1) $\rightarrow$ rain last night,
($S$=1) $\rightarrow$ sprinkler on last night,
($J$=1) $\rightarrow$ Jack's grass is wet,
($T$=1) $\rightarrow$ Traceya's Grass is wet

- Model of Traceya's world involves probability distribution on $T, J, R, S$ that has $2^4 = 16$ states

# Modeling Independencies (contd..)

- However we know

$$p(T, J, R, S) = p(T|J, R, S)p(J, R, S)$$

# Modeling Independencies (contd..)

- However we know

$$\begin{aligned} p(T,J,R,S) &= p(T|J,R,S)p(J,R,S) \\ &= p(T|J,R,S)p(J|R,S)p(R,S) \end{aligned}$$

# Modeling Independencies (contd..)

- However we know

$$
\begin{aligned}
p(T,J,R,S) &= p(T|J,R,S)p(J,R,S) \\
&= p(T|J,R,S)p(J|R,S)p(R,S) \\
&= p(T|J,R,S)p(J|R,S)p(R|S)p(S)
\end{aligned}
$$

# Modeling Independencies (contd..)

- However we know

$$\begin{aligned} p(T,J,R,S) &= p(T|J,R,S)p(J,R,S) \\ &= p(T|J,R,S)p(J|R,S)p(R,S) \\ &= p(T|J,R,S)p(J|R,S)p(R|S)p(S) \end{aligned}$$

- Computation of $p(T|J,R,S)$ requires us to specify $2^3 = 8$ values

# Modeling Independencies (contd..)

- However we know

$$\begin{aligned} p(T,J,R,S) &= p(T|J,R,S)p(J,R,S) \\ &= p(T|J,R,S)p(J|R,S)p(R,S) \\ &= p(T|J,R,S)p(J|R,S)p(R|S)p(S) \end{aligned}$$

- Computation of $p(T|J,R,S)$ requires us to specify $2^3 = 8$ values
- With $p(T=1|J,R,S)$, one can use normalization to compute $p(T=0|J,R,S)$ as $1 - p(T=1|J,R,S)$

# Modeling Independencies (contd..)

- However we know

$$\begin{aligned} p(T,J,R,S) &= p(T|J,R,S)p(J,R,S) \\ &= p(T|J,R,S)p(J|R,S)p(R,S) \\ &= p(T|J,R,S)p(J|R,S)p(R|S)p(S) \end{aligned}$$

- Computation of $p(T|J,R,S)$ requires us to specify $2^3 = 8$ values
- With $p(T=1|J,R,S)$, one can use normalization to compute $p(T=0|J,R,S)$ as $1 - p(T=1|J,R,S)$
- Computation of other factors would also need 4+2+1 values
- Total we need 8+4+2+1=15 values

# Conditional Independence

- We may assume that Traceya's grass is wet depends only directly on whether or not it has been raining and whether or not her sprinkler was on so $p(T|J,R,S) = p(T|R,S)$

# Conditional Independence

- We may assume that Traceya's grass is wet depends only directly on whether or not it has been raining and whether or not her sprinkler was on so $p(T|J,R,S) = p(T|R,S)$
- Assume that Jack's grass is wet is influenced only directly by whether or not it has been raining $p(J|R,S) = p(J|R)$

# Conditional Independence

- We may assume that Traceya's grass is wet depends only directly on whether or not it has been raining and whether or not her sprinkler was on so $p(T|J,R,S) = p(T|R,S)$
- Assume that Jack's grass is wet is influenced only directly by whether or not it has been raining $p(J|R,S) = p(J|R)$
- Furthermore, we assume the rain is not directly influenced by the sprinkler $p(R|S) = p(R)$

# Conditional Independence

- We may assume that Traceya's grass is wet depends only directly on whether or not it has been raining and whether or not her sprinkler was on so $p(T|J,R,S) = p(T|R,S)$
- Assume that Jack's grass is wet is influenced only directly by whether or not it has been raining $p(J|R,S) = p(J|R)$
- Furthermore, we assume the rain is not directly influenced by the sprinkler $p(R|S) = p(R)$
- Therefore, our model becomes

$$\begin{aligned} p(T,J,R,S) &= p(T|J,R,S)p(J|R,S)p(R|S)p(S) \\ &= p(T|R,S)p(J|R)p(R)p(S) \end{aligned}$$

# Conditional Independence

- We may assume that Traceya's grass is wet depends only directly on whether or not it has been raining and whether or not her sprinkler was on so $p(T|J,R,S) = p(T|R,S)$
- Assume that Jack's grass is wet is influenced only directly by whether or not it has been raining $p(J|R,S) = p(J|R)$
- Furthermore, we assume the rain is not directly influenced by the sprinkler $p(R|S) = p(R)$
- Therefore, our model becomes

$$\begin{aligned} p(T,J,R,S) &= p(T|J,R,S)p(J|R,S)p(R|S)p(S) \\ &= p(T|R,S)p(J|R)p(R)p(S) \end{aligned}$$

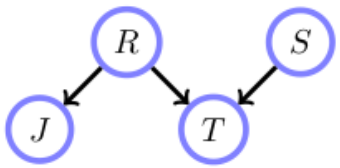
- Number of values we need to specify is 4+2+1+1=8

# Conditional Independence

- We may assume that Traceya's grass is wet depends only directly on whether or not it has been raining and whether or not her sprinkler was on so $p(T|J,R,S) = p(T|R,S)$
- Assume that Jack's grass is wet is influenced only directly by whether or not it has been raining $p(J|R,S) = p(J|R)$
- Furthermore, we assume the rain is not directly influenced by the sprinkler $p(R|S) = p(R)$
- Therefore, our model becomes

$$\begin{aligned} p(T,J,R,S) &= p(T|J,R,S)p(J|R,S)p(R|S)p(S) \\ &= p(T|R,S)p(J|R)p(R)p(S) \end{aligned}$$

- Number of values we need to specify is 4+2+1+1=8
- We can represent these conditional independencies as



# Belief network

How to represent these conditional independencies?

# Belief network

How to represent these conditional independencies?

- **Belief network** is a distribution of the form

$$p(x_1, x_2, ..., x_n) = \Pi_{i=1}^{n} p(x_i | pa(x_i))$$

where $pa(x_i)$ represent the parental variables of variable $x_i$

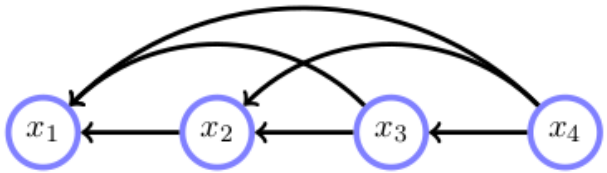
# Belief network

How to represent these conditional independencies?

- **Belief network** is a distribution of the form

$$p(x_1, x_2, ..., x_n) = \Pi_{i=1}^{n} p(x_i | pa(x_i))$$

  where $pa(x_i)$ represent the parental variables of variable $x_i$
- Represented as a directed graph, with an arrow pointing from a parent variable to child variable, a belief network corresponds to a Directed Acyclic Graph (DAG)



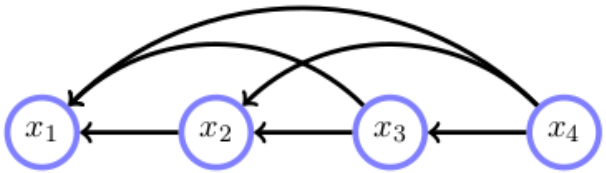
# Belief network

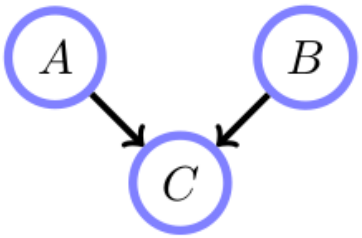
How to represent these conditional independencies?

- **Belief network** is a distribution of the form

$$p(x_1, x_2, ..., x_n) = \Pi_{i=1}^{n} p(x_i|pa(x_i))$$

  where $pa(x_i)$ represent the parental variables of variable $x_i$
- Represented as a directed graph, with an arrow pointing from a parent variable to child variable, a belief network corresponds to a Directed Acyclic Graph (DAG)




$$p(A, B, C) = p(C|A, B)p(A)p(B)$$

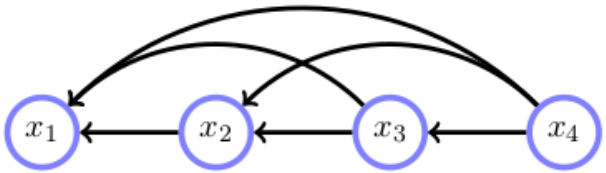
# Belief network

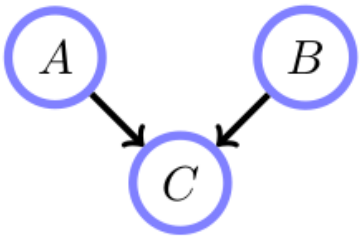
How to represent these conditional independencies?

- **Belief network** is a distribution of the form

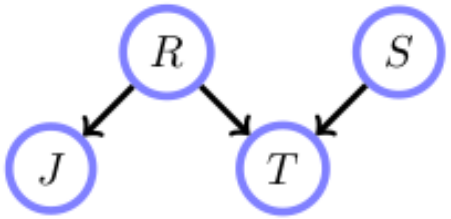
$$p(x_1, x_2, ..., x_n) = \Pi_{i=1}^{n} p(x_i | pa(x_i))$$

  where $pa(x_i)$ represent the parental variables of variable $x_i$
- Represented as a directed graph, with an arrow pointing from a parent variable to child variable, a belief network corresponds to a Directed Acyclic Graph (DAG)




$$p(A, B, C) = p(C|A, B)p(A)p(B)$$



# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$.

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$,

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$,

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$,

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$, $p(T{=}1|R{=}1, S{=}1) = 1$,

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$, $p(T{=}1|R{=}1, S{=}1) = 1$, $p(T{=}1|R{=}0, S{=}1) = 0.9$,

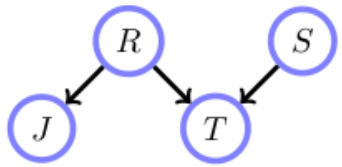
# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$, $p(T{=}1|R{=}1, S{=}1) = 1$, $p(T{=}1|R{=}0, S{=}1) = 0.9$, $p(T{=}1|R{=}0, S{=}0) = 0$

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$, $p(T{=}1|R{=}1, S{=}1) = 1$, $p(T{=}1|R{=}0, S{=}1) = 0.9$, $p(T{=}1|R{=}0, S{=}0) = 0$

Using following Beleif Network; calculate

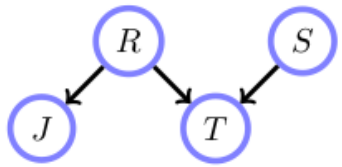



1. Probability that the sprinkler was *on* overnight, given that Traceya's grass is wet.

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$, $p(T{=}1|R{=}1, S{=}1) = 1$, $p(T{=}1|R{=}0, S{=}1) = 0.9$, $p(T{=}1|R{=}0, S{=}0) = 0$

**Using following Beleif Network; calculate**

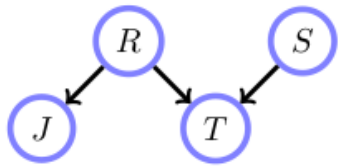



1. Probability that the sprinkler was *on* overnight, given that Traceya's grass is wet. $p(S{=}1|T{=}1) = 0.3382$

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$, $p(T{=}1|R{=}1, S{=}1) = 1$, $p(T{=}1|R{=}0, S{=}1) = 0.9$, $p(T{=}1|R{=}0, S{=}0) = 0$

**Using following Beleif Network; calculate**

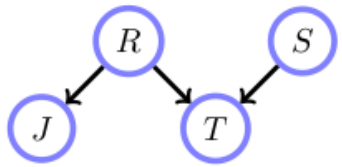



1. Probability that the sprinkler was *on* overnight, given that Traceya's grass is wet. $p(S{=}1|T{=}1) = 0.3382$ How? on next slide
2. Probability that Traceya's sprinkler was *on* overnight, given that her grass is wet and that Jack's grass is also wet.

# Example

One morning Tracey realises that her grass is wet and the grass of her neighbour, Jack, is also wet. Let the prior probabilities be $p(R{=}1) = 0.2$ and $p(S{=}1) = 0.1$. We set $p(J{=}1|R{=}1) = 1$, $p(J{=}1|R{=}0) = 0.2$, $p(T{=}1|R{=}1, S{=}0) = 1$, $p(T{=}1|R{=}1, S{=}1) = 1$, $p(T{=}1|R{=}0, S{=}1) = 0.9$, $p(T{=}1|R{=}0, S{=}0) = 0$

Using following Beleif Network; calculate




1. Probability that the sprinkler was *on* overnight, given that Traceya's grass is wet. $p(S{=}1|T{=}1) = 0.3382$ How? on next slide
2. Probability that Traceya's sprinkler was *on* overnight, given that her grass is wet and that Jack's grass is also wet. $p(S{=}1|T{=}1, J{=}1) = 0.1604$

# Example: Probability of $p(S{=}1|T{=}1)$

$$
\begin{aligned}
p(S=1|T=1) &= \frac{p(S=1,T=1)}{p(T=1)} \\
&= \frac{\sum_{J,R} p(S=1,J,R,T=1)}{\sum_{J,R,S} p(T=1,J,R,S)} \qquad (1) \\
&= \frac{\sum_{J,R} p(J|R)p(T=1|R,S=1)p(R)p(S=1)}{\sum_{J,R,S} p(J|R)p(T=1|R,S)p(R)p(S)} \\
&= \frac{\sum_{J,R} p(T=1|R,S=1)p(R)p(S=1)}{\sum_{J,R,S} p(T=1|R,S)p(R)p(S)} \qquad (2) \\
&= \frac{0.9\times 0.8\times 0.1 + 1\times 0.2\times 0.1}{.9\times .8\times .1 + 1\times .2\times .1 + 0\times .8\times .9 + 1\times .2\times .9} \\
&= \boxed{0.3382}
\end{aligned}
$$

Uses given belief network in (1) and proof in (2)

# Regression

**Regression** predicts value of continuous a target variable

| $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|
| 10 | 50 | 20 | 10 |
| 11 | 31 | 22 | 12 |
| 11 | 12 | 15 | 4 |
| 20 | 55 | 20 | 22 |
| 23 | 41 | 27 | 1 |
| 31 | 12 | 35 | 9 |
| 13 | 18 | 12 | 23 |
| 21 | 55 | 16 | 16 |
| 32 | 56 | 27 | 22 |
| 8 | 22 | 35 | ?? |

What should come at the place of ??

# Regression

**Regression** predicts value of continuous a target variable

# Regression

**Regression** predicts value of continuous a target variable

- A simplest model for regression can be a linear combination of the input variables

$$y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n$$

where $x$ is a $n$ dimensional vector $(x_1, x_2, ..., x_n)$ representing some feature

# Regression

**Regression** predicts value of continuous a target variable

- A simplest model for regression can be a linear combination of the input variables

$$y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n$$

where $x$ is a $n$ dimensional vector $(x_1, x_2, ..., x_n)$ representing some feature

- It can be extended by considering linear combination of fixed nonlinear functions $\phi$ (called basis functions)

$$y(x, w) = w_0 + \sum_{i=1}^{n} w_i \phi_i(x)$$

# Regression

**Regression** predicts value of continuous a target variable

- A simplest model for regression can be a linear combination of the input variables

$$y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n$$

where $x$ is a $n$ dimensional vector $(x_1, x_2, ..., x_n)$ representing some feature

- It can be extended by considering linear combination of fixed nonlinear functions $\phi$ (called basis functions)

$$y(x, w) = w_0 + \sum_{i=1}^{n} w_i \phi_i(x)$$

- In short $y(x, w) = w^T \phi(x)$

# Regression

**Regression** predicts value of continuous a target variable

- A simplest model for regression can be a linear combination of the input variables

$$y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n$$

where $x$ is a $n$ dimensional vector $(x_1, x_2, ..., x_n)$ representing some feature

- It can be extended by considering linear combination of fixed nonlinear functions $\phi$ (called basis functions)

$$y(x, w) = w_0 + \sum_{i=1}^{n} w_i \phi_i(x)$$

- In short $y(x, w) = w^T \phi(x)$
- Objective is to choose $w$ such that it makes $y(x^{(i)}, w)$ as close to $y^{(i)}$ as possible

# Our Regression Example

- If we could correct estimate the values of $w$'s we could determine $y(x^{(i)}, w)$ for all values

| $x_1$ | $x_2$ | $x_3$ | $y$ | $y(x^{(i)}, w)$ |
|---|---|---|---|---|
| 10 | 50 | 20 | 10 | 8 |
| 11 | 31 | 22 | 12 | 9 |
| 11 | 12 | 15 | 4 | 3 |
| 20 | 55 | 20 | 22 | 26 |
| 23 | 41 | 27 | 1 | 1 |
| 31 | 12 | 35 | 9 | 4 |
| 13 | 18 | 12 | 23 | 30 |
| 21 | 55 | 16 | 16 | 13 |
| 32 | 56 | 27 | 22 | 21 |
| 8 | 22 | 35 | ?? | 6 |

Now the question is that how good this $w$ is?

# Regression

- Determining $w$, is similar to solving a minimization problem. Let us define a **squared error cost function** as

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2$$

where $m$ is number of training examples

# Regression

- Determining $w$, is similar to solving a minimization problem. Let us define a **squared error cost function** as

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2$$

  where $m$ is number of training examples
- Then one have to minimize the value of $J(w)$

$$\underset{w}{\operatorname{argmin}}\, J(w)$$

# Regression

- Determining $w$, is similar to solving a minimization problem. Let us define a **squared error cost function** as

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2$$

  where $m$ is number of training examples
- Then one have to minimize the value of $J(w)$

$$\underset{w}{\operatorname{argmin}}\, J(w)$$

- Basic idea: Push $w_i$ a bit against the direction of its gradient

# Linear Regression

| $x_1$ | $x_2$ | $x_3$ | $y$ | $y(x^{(i)}, w)$ | $(y(x^{(i)}, w) - y)^2$ |
|---|---|---|---|---|---|
| 10 | 50 | 20 | 10 | 8 | 4 |
| 11 | 31 | 22 | 12 | 9 | 9 |
| 11 | 12 | 15 | 4 | 3 | 1 |
| 20 | 55 | 20 | 22 | 26 | 16 |
| 23 | 41 | 27 | 1 | 1 | 0 |
| 31 | 12 | 35 | 9 | 4 | 25 |
| 13 | 18 | 12 | 23 | 30 | 49 |
| 21 | 55 | 16 | 16 | 13 | 9 |
| 32 | 56 | 27 | 22 | 21 | 1 |

Assume for some $w$ we computed $y(x^{(i)}, w)$ then

$$\begin{aligned} J(w) &= \frac{1}{2 \times 9} \times 114 \\ &= 6.33 \end{aligned}$$

# Gradient Descent

**Algorithm 1:** Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{\partial}{\partial w_j} J(w)$
4 **until** *converge*;
5 **return** $w$

# Gradient Descent

**Algorithm 2:** Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{\partial}{\partial w_j} J(w)$
4 **until** *converge*;
5 **return** $w$

- Here $\alpha$ is a learning rate. If $\alpha$ is small enough then $J(w)$ would decrease in every iteration

# Gradient Descent

**Algorithm 3:** Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{\partial}{\partial w_j} J(w)$
4 **until** *converge*;
5 **return** $w$

- Here $\alpha$ is a learning rate. If $\alpha$ is small enough then $J(w)$ would decrease in every iteration
(large $\alpha$ can overshoot the minimum and may fail to converge)

# Gradient Descent

**Algorithm 4:** Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{\partial}{\partial w_j} J(w)$
4 **until** *converge*;
5 **return** $w$

- Here $\alpha$ is a learning rate. If $\alpha$ is small enough then $J(w)$ would decrease in every iteration
  (large $\alpha$ can overshoot the minimum and may fail to converge)
- Susceptible to local minimum

# Gradient Descent

**Algorithm 5:** Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{\partial}{\partial w_j} J(w)$
4 **until** *converge*;
5 **return** $w$

- Here $\alpha$ is a learning rate. If $\alpha$ is small enough then $J(w)$ would decrease in every iteration
  (large $\alpha$ can overshoot the minimum and may fail to converge)
- Susceptible to local minimum
- As it moves closer to local minimum, it automatically takes smaller steps as gradient decreases

# Partial Derivative term

$$\frac{\partial}{\partial w_j} J(w) = \frac{\partial}{\partial w_j} \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2$$

# Partial Derivative term

$$\begin{aligned}\frac{\partial}{\partial w_j} J(w) &= \frac{\partial}{\partial w_j} \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2 \\ &= \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)})^2\end{aligned}$$

# Partial Derivative term

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{\partial}{\partial w_j} \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} 2(y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)})
\end{aligned}
$$

# Partial Derivative term

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{\partial}{\partial w_j} \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} 2(y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)}) \\
&= \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} y(x^{(i)}, w)
\end{aligned}
$$

# Partial Derivative term

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{\partial}{\partial w_j} \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} 2(y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)}) \\
&= \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} y(x^{(i)}, w)
\end{aligned}
$$

For $y(x^{(i)}, w) = w_0 + w_1 x_1^{(i)} + ... + w_n x_n^{(i)}$ we have

# Partial Derivative term

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{\partial}{\partial w_j} \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} 2(y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)}) \\
&= \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} y(x^{(i)}, w)
\end{aligned}
$$

For $y(x^{(i)}, w) = w_0 + w_1 x_1^{(i)} + ... + w_n x_n^{(i)}$ we have $\boxed{\frac{\partial}{\partial w_j} y(x^{(i)}, w) = x_j^{(i)}}$

# Partial Derivative term

$$\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{\partial}{\partial w_j} \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)})^2 \\
&= \frac{1}{2m} \sum_{i=1}^{m} 2(y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} (y(x^{(i)}, w) - y^{(i)}) \\
&= \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) \frac{\partial}{\partial w_j} y(x^{(i)}, w)
\end{aligned}$$

For $y(x^{(i)}, w) = w_0 + w_1 x_1^{(i)} + ... + w_n x_n^{(i)}$ we have $\boxed{\frac{\partial}{\partial w_j} y(x^{(i)}, w) = x_j^{(i)}}$

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$$

# Batch-Gradient Descent

**Algorithm 6:** Batch-Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$
4 **until** *converge*;
5 **return** $w$

# Batch-Gradient Descent

**Algorithm 7:** Batch-Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$
4 **until** *converge*;
5 **return** $w$

- At every step it evaluate all training examples

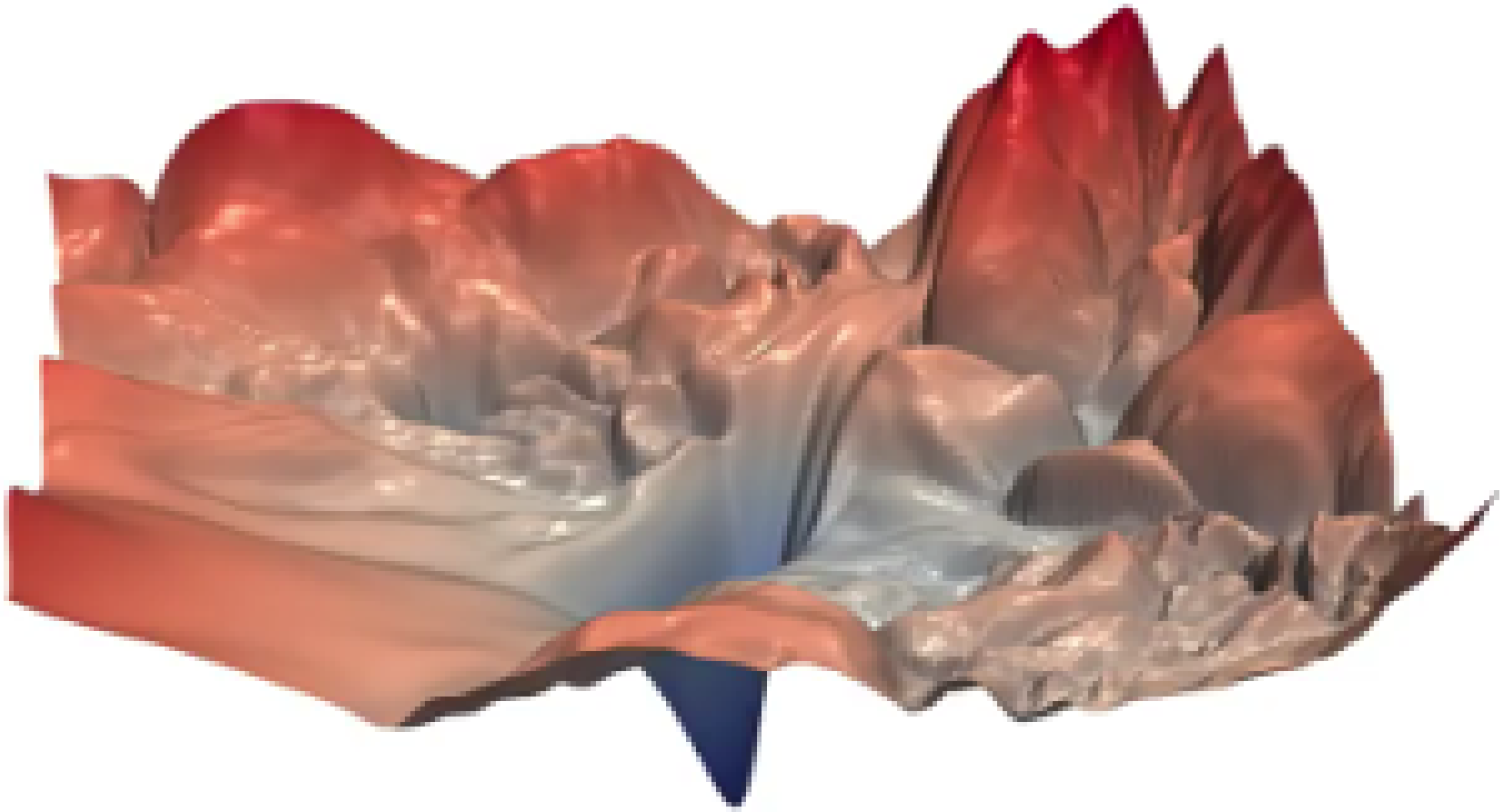
# Batch-Gradient Descent

**Algorithm 8:** Batch-Gradient Descent

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$
4 **until** *converge*;
5 **return** $w$

- At every step it evaluate all training examples
- Some time it is also called multi-variate linear regression

# Real Landscape



# Example: Gradient Descent (learning rate $\alpha$)

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 10 | 50 | 20 | 10 |
| 2 | 11 | 31 | 22 | 12 |
| 3 | 11 | 12 | 15 | 4 |
| 4 | 20 | 55 | 20 | 22 |
| 5 | 23 | 41 | 27 | 1 |
| 6 | 31 | 12 | 35 | 9 |
| 7 | 13 | 18 | 12 | 23 |
| 8 | 21 | 55 | 16 | 16 |
| 9 | 32 | 56 | 27 | 22 |
| 10 | 8 | 22 | 35 | 11 |

# Example: Gradient Descent (learning rate $\alpha$)

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 10 | 50 | 20 | 10 |
| 2 | 11 | 31 | 22 | 12 |
| 3 | 11 | 12 | 15 | 4 |
| 4 | 20 | 55 | 20 | 22 |
| 5 | 23 | 41 | 27 | 1 |
| 6 | 31 | 12 | 35 | 9 |
| 7 | 13 | 18 | 12 | 23 |
| 8 | 21 | 55 | 16 | 16 |
| 9 | 32 | 56 | 27 | 22 |
| 10 | 8 | 22 | 35 | 11 |

**Learning rate** $\alpha = 0.1$

# Example: Gradient Descent (learning rate $\alpha$)

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 10 | 50 | 20 | 10 |
| 2 | 11 | 31 | 22 | 12 |
| 3 | 11 | 12 | 15 | 4 |
| 4 | 20 | 55 | 20 | 22 |
| 5 | 23 | 41 | 27 | 1 |
| 6 | 31 | 12 | 35 | 9 |
| 7 | 13 | 18 | 12 | 23 |
| 8 | 21 | 55 | 16 | 16 |
| 9 | 32 | 56 | 27 | 22 |
| 10 | 8 | 22 | 35 | 11 |

**Learning rate** $\alpha = 0.1$

J=396.662506
w=( 0.500 0.500 0.500 0.500)

J=19454472.000000
w=( -2.055 -51.070 -100.970 -62.640)

J=1036526813184.000000
w=( 590.236 11518.771 23902.906 13778.349)

J=55230041021218816.000000
w=( -135891.922 -2653678.250 -5525792.000 -3170425.000)

J=2942865354556228763648.000000
w=( 31365378.000 612476928.000 1275658624.000 731686912.000)

J=156806972273681738831495168.000000
w=( -7240111104.000 -141378551808.000 -294465732608.000
-168895037440.000)

J=835526654652697102726982742835 2.000000
w=( 1671254376448.000 32634791002112.000 67972370530304.000
38986479304704.000)

J=44520007922259187977070606888730624 0.000000

w=( -385780270759936.000 -7533178826784768.000

-15690251045437440.000 -8999357718200320.000)

# Example: Gradient Descent (learning rate $\alpha$)

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 10 | 50 | 20 | 10 |
| 2 | 11 | 31 | 22 | 12 |
| 3 | 11 | 12 | 15 | 4 |
| 4 | 20 | 55 | 20 | 22 |
| 5 | 23 | 41 | 27 | 1 |
| 6 | 31 | 12 | 35 | 9 |
| 7 | 13 | 18 | 12 | 23 |
| 8 | 21 | 55 | 16 | 16 |
| 9 | 32 | 56 | 27 | 22 |
| 10 | 8 | 22 | 35 | 11 |

**Learning rate** $\alpha = 0.001$

# Example: Gradient Descent (learning rate $\alpha$)

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 10 | 50 | 20 | 10 |
| 2 | 11 | 31 | 22 | 12 |
| 3 | 11 | 12 | 15 | 4 |
| 4 | 20 | 55 | 20 | 22 |
| 5 | 23 | 41 | 27 | 1 |
| 6 | 31 | 12 | 35 | 9 |
| 7 | 13 | 18 | 12 | 23 |
| 8 | 21 | 55 | 16 | 16 |
| 9 | 32 | 56 | 27 | 22 |
| 10 | 8 | 22 | 35 | 11 |

**Learning rate** $\alpha = 0.001$

| J | w |
|---|---|
| 396.663 | ( 0.500 0.500 0.500 0.500) |
| 664.137 | ( 0.474 -0.016 -0.515 -0.131) |
| 1131.021 | ( 0.508 0.631 0.881 0.628) |
| 1943.882 | ( 0.464 -0.249 -0.910 -0.435) |
| 3357.625 | ( 0.523 0.888 1.492 0.914) |
| 5815.401 | ( 0.446 -0.630 -1.641 -0.908) |
| 10087.491 | ( 0.549 1.356 2.518 1.456) |
| 17512.684 | ( 0.415 -1.274 -2.941 -1.693) |
| 30417.834 | ( 0.592 2.183 4.276 2.432) |
| 52847.020 | ( 0.359 -2.383 -5.221 -3.028) |
| 91828.805 | ( 0.668 3.630 7.314 4.151) |
| 159578.781 | ( 0.263 -4.302 -9.200 -5.330) |
| 277327.562 | ( 0.799 6.152 12.580 7.155) |
| 481973.594 | ( 0.093 -7.633 -16.125 -9.316) |
| 837646.250 | ( 1.025 10.537 21.725 12.387) |
| 1455801.375 | ( -0.201 -13.418 -28.168 -16.234) |
| 2530147.500 | ( 1.417 18.162 37.611 21.491) |
| 4397349.000 | ( -0.715 -23.472 -49.103 -28.249) |
| 7642525.500 | ( 2.097 31.415 65.218 37.319) |
| 13282603.000 | ( -1.608 -40.944 -85.492 -49.126) |
| 23084998.000 | ( 3.278 54.449 113.196 64.832) |
| 40121436.000 | ( -3.162 -71.310 -148.738 -85.405) |
| 69730584.000 | ( 5.329 94.483 196.578 112.653) |
| 121190936.000 | ( -5.863 -124.085 -258.660 -148.456) |
| 210628448.000 | ( 8.894 164.060 341.494 195.769) |
| 366069856.000 | ( -10.559 -215.809 -449.705 -258.035) |
| 636225152.000 | ( 15.088 284.983 593.355 340.226) |
| 1105751936.000 | ( -18.721 -375.224 -781.739 -448.479) |
| 1921783808.000 | ( 25.852 495.147 1031.086 591.291) |
| 3340036608.000 | ( -32.908 -652.287 -1358.811 -779.468) |

# Example: Gradient Descent (learning rate $\alpha$)

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 10 | 50 | 20 | 10 |
| 2 | 11 | 31 | 22 | 12 |
| 3 | 11 | 12 | 15 | 4 |
| 4 | 20 | 55 | 20 | 22 |
| 5 | 23 | 41 | 27 | 1 |
| 6 | 31 | 12 | 35 | 9 |
| 7 | 13 | 18 | 12 | 23 |
| 8 | 21 | 55 | 16 | 16 |
| 9 | 32 | 56 | 27 | 22 |
| 10 | 8 | 22 | 35 | 11 |

**Learning rate** $\alpha = 0.0001$

# Example: Gradient Descent (learning rate $\alpha$)

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 10 | 50 | 20 | 10 |
| 2 | 11 | 31 | 22 | 12 |
| 3 | 11 | 12 | 15 | 4 |
| 4 | 20 | 55 | 20 | 22 |
| 5 | 23 | 41 | 27 | 1 |
| 6 | 31 | 12 | 35 | 9 |
| 7 | 13 | 18 | 12 | 23 |
| 8 | 21 | 55 | 16 | 16 |
| 9 | 32 | 56 | 27 | 22 |
| 10 | 8 | 22 | 35 | 11 |

**Learning rate** $\alpha = 0.0001$

| J | w | |
|---|---|---|
| 396.663 | ( 0.500 0.500 0.500 0.500) | |
| 246.798 | ( 0.497 0.448 0.399 0.437) | |
| 158.286 | ( 0.495 0.408 0.321 0.388) | |
| 105.980 | ( 0.494 0.377 0.262 0.349) | |
| 75.041 | ( 0.493 0.353 0.218 0.319) | |
| 56.711 | ( 0.492 0.334 0.184 0.295) | |
| 45.826 | ( 0.491 0.320 0.159 0.276) | |
| 39.335 | ( 0.491 0.308 0.140 0.260) | |
| 35.439 | ( 0.490 0.299 0.126 0.248) | |
| 33.077 | ( 0.490 0.291 0.115 0.238) | |
| 31.621 | ( 0.490 0.285 0.108 0.229) | |
| 30.703 | ( 0.490 0.280 0.103 0.222) | |
| 30.104 | ( 0.490 0.276 0.099 0.216) | |
| 29.694 | ( 0.489 0.273 0.097 0.210) | |
| 29.399 | ( 0.489 0.270 0.096 0.206) | |
| 29.172 | ( 0.489 0.268 0.095 0.202) | |
| 28.987 | ( 0.489 0.266 0.096 0.198) | |
| 28.830 | ( 0.489 0.264 0.096 0.194) | |
| 28.689 | ( 0.489 0.262 0.097 0.191) | |
| 28.560 | ( 0.489 0.260 0.098 0.188) | |
| 28.439 | ( 0.489 0.259 0.099 0.185) | |
| 28.325 | ( 0.489 0.258 0.101 0.182) | |
| 28.216 | ( 0.489 0.256 0.102 0.179) | |
| 28.111 | ( 0.489 0.255 0.104 0.177) | |
| 28.011 | ( 0.489 0.254 0.105 0.174) | |
| 27.913 | ( 0.489 0.253 0.107 0.172) | |
| 27.819 | ( 0.489 0.252 0.109 0.170) | |
| 27.728 | ( 0.489 0.251 0.110 0.167) | |
| 27.555 | ( 0.490 0.249 0.114 0.163) | |
| 24.926 | ( 0.507 0.207 0.215 0.020) | **Iteration 300** |
| 24.768 | ( 0.710 0.219 0.213 0.005) | **Iteration 3000** |

# Example: Gradient Descent (Feature scaling)

**Feature scaling**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 0.08 | 0.86 | 0.35 | 10 |
| 2 | 0.12 | 0.43 | 0.43 | 12 |
| 3 | 0.12 | 0.00 | 0.13 | 4 |
| 4 | 0.50 | 0.98 | 0.35 | 22 |
| 5 | 0.62 | 0.66 | 0.65 | 1 |
| 6 | 0.96 | 0.00 | 1.00 | 9 |
| 7 | 0.21 | 0.14 | 0.00 | 23 |
| 8 | 0.54 | 0.98 | 0.17 | 16 |
| 9 | 1.00 | 1.00 | 0.65 | 22 |
| 10 | 0.00 | 0.23 | 1.00 | 11 |

**Learning rate** $\alpha = 0.1$

# Example: Gradient Descent (Feature scaling)

**Feature scaling**

| | $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|---|
| 1 | 0.08 | 0.86 | 0.35 | 10 |
| 2 | 0.12 | 0.43 | 0.43 | 12 |
| 3 | 0.12 | 0.00 | 0.13 | 4 |
| 4 | 0.50 | 0.98 | 0.35 | 22 |
| 5 | 0.62 | 0.66 | 0.65 | 1 |
| 6 | 0.96 | 0.00 | 1.00 | 9 |
| 7 | 0.21 | 0.14 | 0.00 | 23 |
| 8 | 0.54 | 0.98 | 0.17 | 16 |
| 9 | 1.00 | 1.00 | 0.65 | 22 |
| 10 | 0.00 | 0.23 | 1.00 | 11 |

**Learning rate** $\alpha = 0.1$

| J | w | |
|---|---|---|
| 95.472 | ( 0.500 0.500 0.500 0.500) | |
| 73.399 | ( 1.679 1.025 1.220 0.983) | |
| 58.326 | ( 2.658 1.455 1.822 1.364) | |
| 48.020 | ( 3.470 1.808 2.326 1.663) | |
| 40.961 | ( 4.147 2.096 2.749 1.893) | |
| 36.116 | ( 4.710 2.331 3.106 2.066) | |
| 32.778 | ( 5.180 2.522 3.407 2.193) | |
| 30.468 | ( 5.574 2.677 3.662 2.283) | |
| 28.859 | ( 5.903 2.803 3.880 2.341) | |
| 27.729 | ( 6.181 2.904 4.066 2.373) | |
| 26.925 | ( 6.415 2.985 4.226 2.385) | |
| 26.344 | ( 6.613 3.049 4.364 2.379) | |
| 25.916 | ( 6.782 3.100 4.485 2.360) | |
| 25.593 | ( 6.926 3.140 4.590 2.329) | |
| 25.342 | ( 7.050 3.170 4.683 2.289) | |
| 25.141 | ( 7.158 3.193 4.766 2.241) | |
| 24.974 | ( 7.252 3.210 4.839 2.188) | |
| 24.833 | ( 7.334 3.222 4.906 2.129) | |
| **24.708** | ( 7.407 3.230 4.966 2.067) | **Iteration 18** |
| 24.596 | ( 7.472 3.234 5.021 2.003) | |
| 24.493 | ( 7.530 3.236 5.071 1.935) | |
| 24.397 | ( 7.583 3.235 5.118 1.866) | |
| 24.306 | ( 7.632 3.233 5.161 1.796) | |
| 24.219 | ( 7.677 3.229 5.202 1.725) | |
| 24.136 | ( 7.718 3.225 5.241 1.653) | |
| 24.056 | ( 7.757 3.219 5.277 1.581) | |
| 23.979 | ( 7.794 3.213 5.311 1.509) | |
| 23.903 | ( 7.830 3.206 5.344 1.436) | |
| 23.830 | ( 7.863 3.198 5.375 1.364) | |
| 23.759 | ( 7.896 3.191 5.405 1.292) | **Iteration 30** |
| 20.174 | ( 12.021 4.618 4.794 -7.329) | **Iteration 3000** |

# Similar Mechanism for Classification

Classification have predefined fixed number of labels (0 and 1 in this case)

| $x_1$ | $x_2$ | $x_3$ | *Class* |
|---|---|---|---|
| 10 | 50 | 20 | 1 |
| 11 | 31 | 22 | 1 |
| 11 | 12 | 15 | 0 |
| 20 | 55 | 20 | 0 |
| 23 | 41 | 27 | 0 |
| 31 | 12 | 35 | 1 |
| 13 | 18 | 12 | 0 |
| 21 | 55 | 16 | 1 |
| 32 | 56 | 27 | 0 |
| 8 | 22 | 35 | ?? |

What should come at the place of ??

# Logistic Regression

Moving from linear regression $y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n$ to **logistic regression**

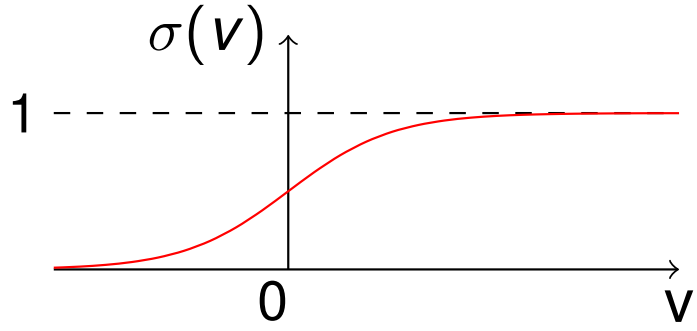
$$y(x, w) = \sigma(w_0 + w_1 x_1 + ... + w_n x_n)$$

# Logistic Regression

Moving from linear regression $y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n$ to **logistic regression**

$$y(x, w) = \sigma(w_0 + w_1 x_1 + ... + w_n x_n)$$

- Enables "classification" apart from the regression. Where $\sigma$ is called as sigmoid function that produces values in range $[0, 1]$ and is defined as $\sigma(v) = \frac{1}{1+e^{-v}}$




## Decision on classification

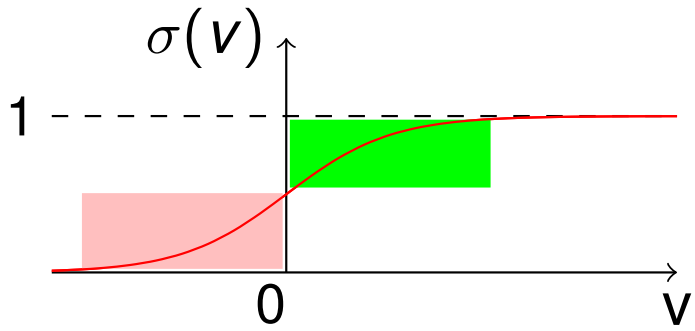
$$classification = \begin{cases} 1 & \text{if } y(x, w) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

# Logistic Regression

Moving from linear regression $y(x, w) = w_0 + w_1 x_1 + ... + w_n x_n$ to **logistic regression**

$$y(x, w) = \sigma(w_0 + w_1 x_1 + ... + w_n x_n)$$

- Enables "classification" apart from the regression. Where $\sigma$ is called as sigmoid function that produces values in range $[0, 1]$ and is defined as $\sigma(v) = \frac{1}{1+e^{-v}}$




## Decision on classification

$$classification = \begin{cases} 1 & \text{if } y(x, w) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

# Decision Boundary in Logistic Regression

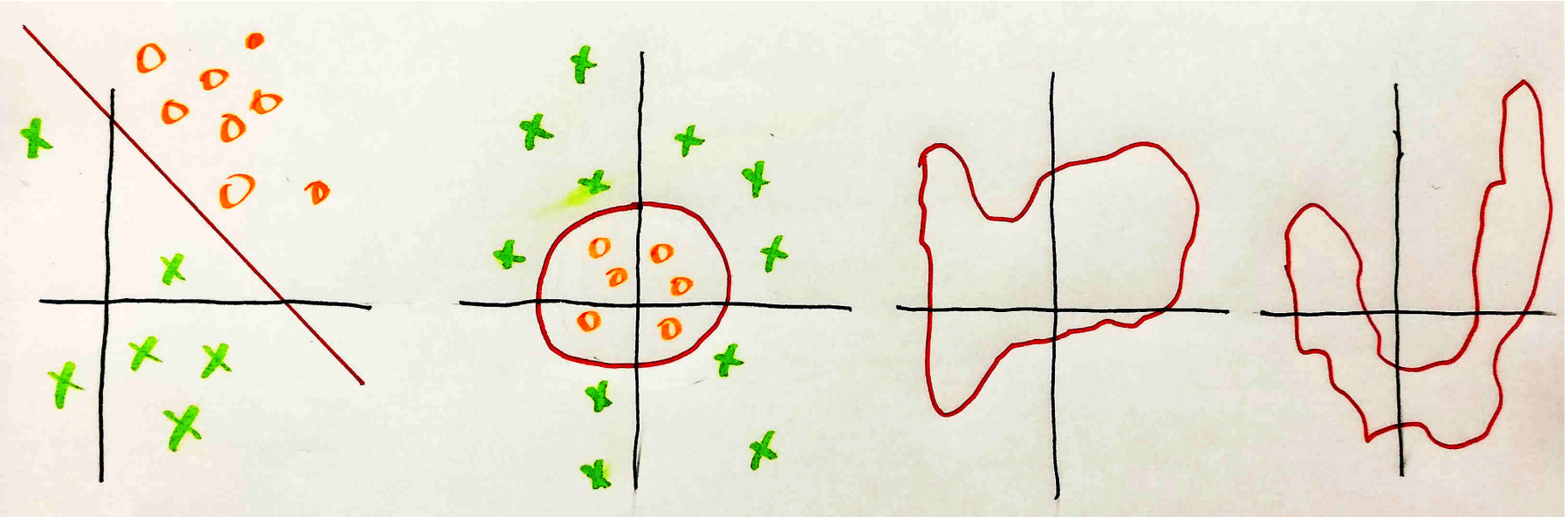
$$classification = \begin{cases} 1 & \text{if } y(x, w) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- This choice of $w$ partitions the space into two sections and the hyper-plane separating them is called **decision boundary**

# Decision Boundary in Logistic Regression

$$classification = \begin{cases} 1 & \text{if } y(x, w) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

- This choice of $w$ partitions the space into two sections and the hyper-plane separating them is called **decision boundary**
- By adding more complex or polynomial terms one can get more complex decision boundary



# Cost Function

- Cost function used for the liner regression

$$J(w) = \frac{1}{2m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)})^2$$

becomes a non convex function in case of logistic regression

# Cost Function

- Cost function used for the liner regression

$$J(w) = \frac{1}{2m}\sum_{i=1}^{m}(y(x^{(i)}, w) - y^{(i)})^2$$

becomes a non convex function in case of logistic regression

## Therefore, a different cost function is chosen

$$J(w) = \frac{1}{m}\sum_{i=1}^{m} Cost(y(x^{(i)}, w), y^{(i)})$$

where

$$Cost(y(x^{(i)}, w), y^{(i)}) = \begin{cases} -\log(y(x^{(i)}, w)) & \text{if } y^{(i)} = 1 \\ -\log(1 - y(x^{(i)}, w)) & \text{otherwise} \end{cases}$$

A simplified version of this cost function is

$$Cost(y(x^{(i)}, w), y^{(i)}) = -y^{(i)}\log(y(x^{(i)}, w)) - (1 - y^{(i)})\log(1 - y(x^{(i)}, w))$$

# Learning With This Cost Function

- Learning corresponds to the minimization of $J(w)$ by changing $w$

$$\underset{w}{\operatorname{argmin}}\, J(w) = \frac{1}{m} \sum_{i=1}^{m} Cost(y(x^{(i)}, w), y^{(i)})$$

# Learning With This Cost Function

- Learning corresponds to the minimization of $J(w)$ by changing $w$

$$\underset{w}{\operatorname{argmin}}\, J(w) = \frac{1}{m}\sum_{i=1}^{m} Cost(y(x^{(i)}, w), y^{(i)})$$

$$\underset{w}{\operatorname{argmin}}\, J(w) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(y(x^{(i)}, w)) - (1 - y^{(i)})\log(1 - y(x^{(i)}, w))]$$

# Learning With This Cost Function

- Learning corresponds to the minimization of $J(w)$ by changing $w$

$$\underset{w}{\text{argmin}}\, J(w) = \frac{1}{m}\sum_{i=1}^{m} Cost(y(x^{(i)}, w), y^{(i)})$$

$$\underset{w}{\text{argmin}}\, J(w) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(y(x^{(i)}, w)) - (1 - y^{(i)})\log(1 - y(x^{(i)}, w))]$$

- Gradient Descent can be used for this purpose

**Algorithm 11:** Logistic Regression

1 Initialize $w$ randomly
2 **repeat**
3 | Simultaneously update all $w_j$ with $w_j - \alpha\frac{\partial}{\partial w_j}J(w)$
4 **until** *converge*;
5 **return** $w$

# The Partial Derivative Term

Recall differentiation

$$\frac{d}{dx} x^{-1} = \frac{-1}{x^2}$$

$$\frac{d}{dx} \log x = \frac{1}{x}$$

$$\frac{d}{dx} \log \sin x = \frac{1}{\sin x} \frac{d}{dx} \sin x = \frac{1}{\sin x} \cos x$$

# The Partial Derivative Term

Recall differentiation

$$\frac{d}{dx}x^{-1} = \frac{-1}{x^2} \qquad \frac{d}{dx}\log x = \frac{1}{x} \qquad \frac{d}{dx}\log\sin x = \frac{1}{\sin x}\frac{d}{dx}\sin x = \frac{1}{\sin x}\cos x$$

## Let $v = w_0x_0 + w_1x_1 + \ldots + w_nx_n$ Then

$$\frac{\partial}{\partial w_j}v =$$

# The Partial Derivative Term

Recall differentiation

$$\frac{d}{dx}x^{-1} = \frac{-1}{x^2} \qquad \frac{d}{dx}\log x = \frac{1}{x} \qquad \frac{d}{dx}\log\sin x = \frac{1}{\sin x}\frac{d}{dx}\sin x = \frac{1}{\sin x}\cos x$$

## Let $v = w_0x_0 + w_1x_1 + \ldots + w_nx_n$ Then

$$\frac{\partial}{\partial w_j}v = \frac{\partial}{\partial w_j}(w_0x_0 + w_1x_1 + \ldots + w_nx_n) =$$

# The Partial Derivative Term

Recall differentiation

$$\frac{d}{dx}x^{-1} = \frac{-1}{x^2} \qquad \frac{d}{dx}\log x = \frac{1}{x} \qquad \frac{d}{dx}\log\sin x = \frac{1}{\sin x}\frac{d}{dx}\sin x = \frac{1}{\sin x}\cos x$$

## Let $v = w_0x_0 + w_1x_1 + \ldots + w_nx_n$ Then

$$\frac{\partial}{\partial w_j}v = \frac{\partial}{\partial w_j}(w_0x_0 + w_1x_1 + \ldots + w_nx_n) = x_j$$

# The Partial Derivative Term

Recall differentiation

$$\frac{d}{dx}x^{-1} = \frac{-1}{x^2} \qquad \frac{d}{dx}\log x = \frac{1}{x} \qquad \frac{d}{dx}\log\sin x = \frac{1}{\sin x}\frac{d}{dx}\sin x = \frac{1}{\sin x}\cos x$$

## Let $v = w_0x_0 + w_1x_1 + \ldots + w_nx_n$ Then

$$\frac{\partial}{\partial w_j}v = \frac{\partial}{\partial w_j}(w_0x_0 + w_1x_1 + \ldots + w_nx_n) = x_j$$

$$J(w) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(y(x^{(i)}, w)) - (1 - y^{(i)})\log(1 - y(x^{(i)}, w))]$$

# The Partial Derivative Term

Recall differentiation

$$\frac{d}{dx}x^{-1} = \frac{-1}{x^2} \qquad \frac{d}{dx}\log x = \frac{1}{x} \qquad \frac{d}{dx}\log\sin x = \frac{1}{\sin x}\frac{d}{dx}\sin x = \frac{1}{\sin x}\cos x$$

## Let $v = w_0x_0 + w_1x_1 + \ldots + w_nx_n$ Then

$$\frac{\partial}{\partial w_j}v = \frac{\partial}{\partial w_j}(w_0x_0 + w_1x_1 + \ldots + w_nx_n) = x_j$$

$$\begin{aligned} J(w) &= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(y(x^{(i)}, w)) - (1 - y^{(i)})\log(1 - y(x^{(i)}, w))] \\ \frac{\partial}{\partial w_j}J(w) &= \frac{1}{m}\sum_{i=1}^{m}[-\frac{\partial}{\partial w_j}y^{(i)}\log(y(x^{(i)}, w)) - \frac{\partial}{\partial w_j}(1 - y^{(i)})\log(1 - y(x^{(i)}, w))] \end{aligned}$$

# The Partial Derivative Term

Recall differentiation

$$\frac{d}{dx}x^{-1} = \frac{-1}{x^2} \qquad \frac{d}{dx}\log x = \frac{1}{x} \qquad \frac{d}{dx}\log\sin x = \frac{1}{\sin x}\frac{d}{dx}\sin x = \frac{1}{\sin x}\cos x$$

## Let $v = w_0x_0 + w_1x_1 + \ldots + w_nx_n$ Then

$$\frac{\partial}{\partial w_j}v = \frac{\partial}{\partial w_j}(w_0x_0 + w_1x_1 + \ldots + w_nx_n) = x_j$$

$$\begin{aligned}
J(w) &= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(y(x^{(i)}, w)) - (1 - y^{(i)})\log(1 - y(x^{(i)}, w))] \\
\frac{\partial}{\partial w_j}J(w) &= \frac{1}{m}\sum_{i=1}^{m}[-\frac{\partial}{\partial w_j}y^{(i)}\log(y(x^{(i)}, w)) - \frac{\partial}{\partial w_j}(1 - y^{(i)})\log(1 - y(x^{(i)}, w))] \\
&= \frac{1}{m}\sum_{i=1}^{m}[-A - B] \qquad (3)
\end{aligned}$$

# The Partial Derivative

$$A = \frac{\partial}{\partial w_j} y^{(i)} \log(y(x^{(i)}, w))$$

# The Partial Derivative

$$
\begin{aligned}
A &= \frac{\partial}{\partial w_j} y^{(i)} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{\partial}{\partial w_j} \log(y(x^{(i)}, w))
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
A &= \frac{\partial}{\partial w_j} y^{(i)} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{\partial}{\partial w_j} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{1}{y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w)
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
A &= \frac{\partial}{\partial w_j} y^{(i)} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{\partial}{\partial w_j} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{1}{y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= y^{(i)} \times \frac{1}{\frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} \frac{1}{1+e^{-v}}
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
A &= \frac{\partial}{\partial w_j} y^{(i)} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{\partial}{\partial w_j} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{1}{y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= y^{(i)} \times \frac{1}{\frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} \frac{1}{1+e^{-v}} \\
&= y^{(i)} \times (1+e^{-v}) \times \frac{-1}{(1+e^{-v})^2} \times \frac{\partial}{\partial w_j} (1+e^{-v})
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
A &= \frac{\partial}{\partial w_j} y^{(i)} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{\partial}{\partial w_j} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{1}{y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= y^{(i)} \times \frac{1}{\frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} \frac{1}{1+e^{-v}} \\
&= y^{(i)} \times (1+e^{-v}) \times \frac{-1}{(1+e^{-v})^2} \times \frac{\partial}{\partial w_j} (1+e^{-v}) \\
&= \frac{-y^{(i)}}{1+e^{-v}} \times (0 + e^{-v} \times (-1) \frac{\partial}{\partial w_j} v)
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
A &= \frac{\partial}{\partial w_j} y^{(i)} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{\partial}{\partial w_j} \log(y(x^{(i)}, w)) \\
&= y^{(i)} \times \frac{1}{y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= y^{(i)} \times \frac{1}{\frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} \frac{1}{1+e^{-v}} \\
&= y^{(i)} \times (1+e^{-v}) \times \frac{-1}{(1+e^{-v})^2} \times \frac{\partial}{\partial w_j} (1+e^{-v}) \\
&= \frac{-y^{(i)}}{1+e^{-v}} \times (0 + e^{-v} \times (-1) \frac{\partial}{\partial w_j} v) \\
&= y^{(i)} \times \frac{e^{-v}}{1+e^{-v}} \times x_j \qquad (4)
\end{aligned}
$$

# The Partial Derivative

$$B = \frac{\partial}{\partial w_j}(1 - y^{(i)}) \log(1 - y(x^{(i)}, w))$$

# The Partial Derivative

$$
\begin{aligned}
B &= \frac{\partial}{\partial w_j}(1 - y^{(i)}) \log(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{1}{1 - y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j}(1 - y(x^{(i)}, w))
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
B &= \frac{\partial}{\partial w_j}(1 - y^{(i)}) \log(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{1}{1 - y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j}(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{-1}{1 - \frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w)
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
B &= \frac{\partial}{\partial w_j}(1 - y^{(i)}) \log(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{1}{1 - y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j}(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{-1}{1 - \frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{\partial}{\partial w_j} \frac{1}{1 + e^{-v}}
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
B &= \frac{\partial}{\partial w_j}(1 - y^{(i)}) \log(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{1}{1 - y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j}(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{-1}{1 - \frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{\partial}{\partial w_j} \frac{1}{1 + e^{-v}} \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{-1}{(1 + e^{-v})^2} \times \frac{\partial}{\partial w_j}(1 + e^{-v})
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
B &= \frac{\partial}{\partial w_j}(1 - y^{(i)}) \log(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{1}{1 - y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j}(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{-1}{1 - \frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{\partial}{\partial w_j} \frac{1}{1 + e^{-v}} \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{-1}{(1 + e^{-v})^2} \times \frac{\partial}{\partial w_j}(1 + e^{-v}) \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{-1}{(1 + e^{-v})^2} \times (0 + e^{-v} \frac{\partial}{\partial w_j}(-v))
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
B &= \frac{\partial}{\partial w_j}(1 - y^{(i)}) \log(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{1}{1 - y(x^{(i)}, w)} \times \frac{\partial}{\partial w_j}(1 - y(x^{(i)}, w)) \\
&= (1 - y^{(i)}) \times \frac{-1}{1 - \frac{1}{1+e^{-v}}} \times \frac{\partial}{\partial w_j} y(x^{(i)}, w) \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{\partial}{\partial w_j} \frac{1}{1 + e^{-v}} \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{-1}{(1 + e^{-v})^2} \times \frac{\partial}{\partial w_j}(1 + e^{-v}) \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{-1}{(1 + e^{-v})^2} \times (0 + e^{-v} \frac{\partial}{\partial w_j}(-v)) \\
&= (1 - y^{(i)}) \times \frac{(-1)(1 + e^{-v})}{e^{-v}} \times \frac{e^{-v}}{(1 + e^{-v})^2} \times \frac{\partial}{\partial w_j} v
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
B &= \frac{\partial}{\partial w_j}(1-y^{(i)})\log(1-y(x^{(i)},w)) \\
&= (1-y^{(i)})\times\frac{1}{1-y(x^{(i)},w)}\times\frac{\partial}{\partial w_j}(1-y(x^{(i)},w)) \\
&= (1-y^{(i)})\times\frac{-1}{1-\frac{1}{1+e^{-v}}}\times\frac{\partial}{\partial w_j}y(x^{(i)},w) \\
&= (1-y^{(i)})\times\frac{(-1)(1+e^{-v})}{e^{-v}}\times\frac{\partial}{\partial w_j}\frac{1}{1+e^{-v}} \\
&= (1-y^{(i)})\times\frac{(-1)(1+e^{-v})}{e^{-v}}\times\frac{-1}{(1+e^{-v})^2}\times\frac{\partial}{\partial w_j}(1+e^{-v}) \\
&= (1-y^{(i)})\times\frac{(-1)(1+e^{-v})}{e^{-v}}\times\frac{-1}{(1+e^{-v})^2}\times(0+e^{-v}\frac{\partial}{\partial w_j}(-v)) \\
&= (1-y^{(i)})\times\frac{(-1)(1+e^{-v})}{e^{-v}}\times\frac{e^{-v}}{(1+e^{-v})^2}\times\frac{\partial}{\partial w_j}v \\
&= (1-y^{(i)})\times\frac{-1}{1+e^{-v}}\times x_j
\end{aligned}
\tag{5}
$$

# The Partial Derivative

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} [-A - B]$$

# The Partial Derivative

$$\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{1}{m} \sum_{i=1}^{m} [-A - B] \\
&= \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \times \frac{e^{-v}}{1 + e^{-v}} \times x_j - (1 - y^{(i)}) \times \frac{-1}{1 + e^{-v}} \times x_j]
\end{aligned}$$

# The Partial Derivative

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{1}{m} \sum_{i=1}^{m} [-A - B] \\
&= \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \times \frac{e^{-v}}{1 + e^{-v}} \times x_j - (1 - y^{(i)}) \times \frac{-1}{1 + e^{-v}} \times x_j] \\
&= \frac{1}{m} \sum_{i=1}^{m} [(1 - y^{(i)}) - y^{(i)} \times e^{-v}] \times \frac{x_j}{1 + e^{-v}}
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{1}{m} \sum_{i=1}^{m} [-A - B] \\
&= \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \times \frac{e^{-v}}{1 + e^{-v}} \times x_j - (1 - y^{(i)}) \times \frac{-1}{1 + e^{-v}} \times x_j] \\
&= \frac{1}{m} \sum_{i=1}^{m} [(1 - y^{(i)}) - y^{(i)} \times e^{-v}] \times \frac{x_j}{1 + e^{-v}} \\
&= \frac{1}{m} \sum_{i=1}^{m} [1 - y^{(i)} \times (1 + e^{-v})] \times \frac{x_j}{1 + e^{-v}}
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{1}{m}\sum_{i=1}^{m}[-A - B] \\
&= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)} \times \frac{e^{-v}}{1+e^{-v}} \times x_j - (1-y^{(i)}) \times \frac{-1}{1+e^{-v}} \times x_j] \\
&= \frac{1}{m}\sum_{i=1}^{m}[(1-y^{(i)}) - y^{(i)} \times e^{-v}] \times \frac{x_j}{1+e^{-v}} \\
&= \frac{1}{m}\sum_{i=1}^{m}[1 - y^{(i)} \times (1+e^{-v})] \times \frac{x_j}{1+e^{-v}} \\
&= \frac{1}{m}\sum_{i=1}^{m}[\frac{1}{1+e^{-v}} - y^{(i)}] \times x_j
\end{aligned}
$$

# The Partial Derivative

$$
\begin{aligned}
\frac{\partial}{\partial w_j} J(w) &= \frac{1}{m}\sum_{i=1}^{m}[-A - B] \\
&= \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)} \times \frac{e^{-v}}{1+e^{-v}} \times x_j - (1-y^{(i)}) \times \frac{-1}{1+e^{-v}} \times x_j] \\
&= \frac{1}{m}\sum_{i=1}^{m}[(1-y^{(i)}) - y^{(i)} \times e^{-v}] \times \frac{x_j}{1+e^{-v}} \\
&= \frac{1}{m}\sum_{i=1}^{m}[1 - y^{(i)} \times (1+e^{-v})] \times \frac{x_j}{1+e^{-v}} \\
&= \frac{1}{m}\sum_{i=1}^{m}[\frac{1}{1+e^{-v}} - y^{(i)}] \times x_j \\
&= \frac{1}{m}\sum_{i=1}^{m}[y(x^{(i)}, w) - y^{(i)}] \times x_j \qquad (6)
\end{aligned}
$$

# The Partial Derivative

Partial derivative term

$$\frac{\partial}{\partial w_j} J(w) = \frac{\partial}{\partial w_j} \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \log(y(x^{(i)}, w)) - (1 - y^{(i)}) \log(1 - y(x^{(i)}, w))]$$

# The Partial Derivative

Partial derivative term

$$\frac{\partial}{\partial w_j} J(w) = \frac{\partial}{\partial w_j} \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \log(y(x^{(i)}, w)) - (1 - y^{(i)}) \log(1 - y(x^{(i)}, w))]$$

comes out to be

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$$

# The Partial Derivative

Partial derivative term

$$\frac{\partial}{\partial w_j} J(w) = \frac{\partial}{\partial w_j} \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \log(y(x^{(i)}, w)) - (1 - y^{(i)}) \log(1 - y(x^{(i)}, w))]$$

comes out to be

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$$

**Algorithm 14:** Logistic Regression

1. Initialize $w$ randomly
2. **repeat**
3. Simultaneously update all $w_j$ with $w_j - \alpha \times \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$
4. **until** *converge*;
5. **return** $w$

# The Partial Derivative

Partial derivative term

$$\frac{\partial}{\partial w_j} J(w) = \frac{\partial}{\partial w_j} \frac{1}{m} \sum_{i=1}^{m} [-y^{(i)} \log(y(x^{(i)}, w)) - (1 - y^{(i)}) \log(1 - y(x^{(i)}, w))]$$

comes out to be

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$$

**Algorithm 15:** Logistic Regression

1. Initialize $w$ randomly
2. **repeat**
3. Simultaneously update all $w_j$ with $w_j - \alpha \times \frac{1}{m} \sum_{i=1}^{m} (y(x^{(i)}, w) - y^{(i)}) x_j^{(i)}$
4. **until** *converge*;
5. **return** $w$

It looks identical to liner regression but, $y(x^{(i)}, w)$ is different here $y(x^{(i)}, w) = \frac{1}{1 + e^{-(w_0 + w_1 x_1^{(i)} + \ldots + w_n x_n^{(i)})}}$

# Example: Logistic Regression

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | *Class* |
|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 |
| 2 | 3 | 2 | 2 | 1 |
| 3 | 2 | 3 | 2 | 1 |
| 4 | 2 | 2 | 3 | 1 |
| 5 | 7 | 6 | 9 | 0 |
| 6 | 9 | 7 | 6 | 0 |
| 7 | 9 | 6 | 7 | 0 |
| 8 | 6 | 8 | 9 | 0 |
| 9 | 8 | 9 | 6 | 0 |
| 10 | 8 | 9 | 9 | 0 |

**Learning rate** $\alpha = 0.01$

# Example: Logistic Regression

**Consider following data**

| | $x_1$ | $x_2$ | $x_3$ | *Class* |
|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 |
| 2 | 3 | 2 | 2 | 1 |
| 3 | 2 | 3 | 2 | 1 |
| 4 | 2 | 2 | 3 | 1 |
| 5 | 7 | 6 | 9 | 0 |
| 6 | 9 | 7 | 6 | 0 |
| 7 | 9 | 6 | 7 | 0 |
| 8 | 6 | 8 | 9 | 0 |
| 9 | 8 | 9 | 6 | 0 |
| 10 | 8 | 9 | 9 | 0 |

**Learning rate** $\alpha = 0.01$

| J | w |
|---|---|
| 6.912 | ( 0.500 0.500 0.500 0.500) |
| 6.496 | ( 0.494 0.453 0.455 0.454) |
| 5.944 | ( 0.488 0.406 0.410 0.408) |
| 5.316 | ( 0.482 0.360 0.366 0.363) |
| 4.692 | ( 0.477 0.313 0.321 0.317) |
| 4.072 | ( 0.471 0.267 0.277 0.272) |
| 3.460 | ( 0.465 0.221 0.233 0.227) |
| 2.860 | ( 0.460 0.175 0.189 0.182) |
| 2.279 | ( 0.454 0.130 0.146 0.138) |
| 1.735 | ( 0.449 0.086 0.104 0.095) |
| 1.262 | ( 0.445 0.044 0.064 0.054) |
| 0.906 | ( 0.441 0.008 0.029 0.018) |
| 0.685 | ( 0.438 -0.022 0.000 -0.011) |
| 0.566 | ( 0.437 -0.044 -0.020 -0.032) |
| 0.504 | ( 0.436 -0.060 -0.035 -0.048) |
| 0.470 | ( 0.436 -0.072 -0.046 -0.059) |
| 0.451 | ( 0.436 -0.081 -0.055 -0.068) |
| 0.438 | ( 0.436 -0.088 -0.061 -0.074) |
| 0.431 | ( 0.437 -0.093 -0.066 -0.080) |
| 0.425 | ( 0.438 -0.098 -0.070 -0.084) |
| 0.422 | ( 0.439 -0.101 -0.074 -0.088) |
| 0.419 | ( 0.440 -0.105 -0.077 -0.091) |
| 0.417 | ( 0.441 -0.107 -0.079 -0.093) |
| 0.416 | ( 0.443 -0.110 -0.081 -0.095) |
| 0.415 | ( 0.444 -0.112 -0.082 -0.097) **Iteration 25** |
| 0.412 | ( 0.451 -0.119 -0.088 -0.103) **Iteration 30** |
| 0.348 | ( 0.857 -0.179 -0.084 -0.132) **Iteration 300** |
| 0.116 | ( 3.256 -0.409 -0.135 -0.291) **Iteration 3000** |
| 0.012 | ( 7.596 -0.748 -0.361 -0.588) **Iteration 30000** |
| 0.001 | ( 11.975 -1.091 -0.599 -0.896) **Iteration 300000** |

# Example: Find J

As $(w_0, w_1, w_2, w_3)$=$(0.5, 0.5, 0.5, 0.5)$, $v = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$
$y(x^{(i)}, w) = \sigma(v)$
And log term is $-y^{(i)} \log(y(x^{(i)}, w)) - (1 - y^{(i)}) \log(1 - y(x^{(i)}, w))$

| $i$ | $x_1$ | $x_2$ | $x_3$ | $y^{(i)}$ | v | $y(x^{(i)}, w)$ | log term |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 | 3.5 | 0.970 | 0.029 |
| 2 | 3 | 2 | 2 | 1 | 4.0 | 0.982 | 0.018 |
| 3 | 2 | 3 | 2 | 1 | 4.0 | 0.982 | 0.018 |
| 4 | 2 | 2 | 3 | 1 | 4.0 | 0.982 | 0.018 |
| 5 | 7 | 6 | 9 | 0 | 11.5 | 0.999 | 11.49 |
| 6 | 9 | 7 | 6 | 0 | 11.5 | 0.999 | 11.49 |
| 7 | 9 | 6 | 7 | 0 | 11.5 | 0.999 | 11.49 |
| 8 | 6 | 8 | 9 | 0 | 12 | 0.999 | 11.51 |
| 9 | 8 | 9 | 6 | 0 | 12 | 0.999 | 11.51 |
| 10 | 8 | 9 | 9 | 0 | 13 | 0.999 | 11.51 |
| | | | | | | Total/10: | 6.9118 |

# Example: Find next W

Let $(w_0, w_1, w_2, w_3)$=(0.5, 0.5, 0.5, 0.5) and $t_i = (y(x^{(i)}, w) - y^{(i)})x_j^{(i)}$

Then $\frac{1}{m}\sum_{i=1}^{m}(y(x^{(i)}, w) - y^{(i)})x_j^{(i)} = \frac{1}{m}\sum_{i=1}^{m} t_i$ let $\hat{y}^{(i)} = y(x^{(i)}, w)$

Then update $w_j$ with $w_j - \alpha \times \frac{1}{m}\sum_{i=1}^{m} t_i$ we have set $\alpha = 0.01$

| $i$ | $x_0$ | $x_1$ | $x_2$ | $x_3$ | $y^{(i)}$ | $\hat{y}^{(i)}$ | $t_0$ | $t_1$ | $t_2$ | $t_3$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 2 | 1 | 0.970 | -0.029 | -0.058 | -0.058 | -0.058 |
| 2 | 1 | 3 | 2 | 2 | 1 | 0.982 | -0.017 | -0.053 | -0.035 | -0.035 |
| 3 | 1 | 2 | 3 | 2 | 1 | 0.982 | -0.017 | -0.035 | -0.053 | -0.035 |
| 4 | 1 | 2 | 2 | 3 | 1 | 0.982 | -0.017 | -0.035 | -0.035 | -0.053 |
| 5 | 1 | 7 | 6 | 9 | 0 | 0.999 | 0.999 | 6.999 | 5.999 | 8.999 |
| 6 | 1 | 9 | 7 | 6 | 0 | 0.999 | 0.999 | 8.999 | 6.999 | 5.999 |
| 7 | 1 | 9 | 6 | 7 | 0 | 0.999 | 0.999 | 8.999 | 5.999 | 6.999 |
| 8 | 1 | 6 | 8 | 9 | 0 | 0.999 | 0.999 | 5.999 | 7.999 | 8.999 |
| 9 | 1 | 8 | 9 | 6 | 0 | 0.999 | 0.999 | 7.999 | 8.999 | 5.999 |
| 10 | 1 | 8 | 9 | 9 | 0 | 0.999 | 0.999 | 7.999 | 8.999 | 8.999 |
| | | | | | | Total | 5.916 | 46.815 | 44.815 | 45.815 |
| | | | | | | $w_i - \alpha \times (total/m)$ | 0.494 | 0.453 | 0.455 | 0.454 |

# Example: Classification across Iterations

Following table shows classification as the weights get modified along $1^{st}$, $100^{th}$, $300^{th}$ and $500^{th}$ iteration

| $i$ | $x_1$ | $x_2$ | $x_3$ | $y^{(i)}$ | 1 | 100 | 300 | 500 |
|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 1 | 1 | 0 | 1 | 1 |
| 2 | 3 | 2 | 2 | 1 | 1 | 0 | 0 | 1 |
| 3 | 2 | 3 | 2 | 1 | 1 | 0 | 1 | 1 |
| 4 | 2 | 2 | 3 | 1 | 1 | 0 | 1 | 1 |
| 5 | 7 | 6 | 9 | 0 | 1 | 0 | 0 | 0 |
| 6 | 9 | 7 | 6 | 0 | 1 | 0 | 0 | 0 |
| 7 | 9 | 6 | 7 | 0 | 1 | 0 | 0 | 0 |
| 8 | 6 | 8 | 9 | 0 | 1 | 0 | 0 | 0 |
| 9 | 8 | 9 | 6 | 0 | 1 | 0 | 0 | 0 |
| 10 | 8 | 9 | 9 | 0 | 1 | 0 | 0 | 0 |

# Thank You!

**Thank you very much for your attention!**

**Queries ?**

(Reference[1])

[1] 1) Book - *AIMA*, ch-14, Russell and Norvig. 2) Book - *Bayeian Reasoning and Machine Learning*, ch-04, David Barber.