Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression, Overfitting, Factorization Machines

Krisztian Buza

Department of Artificial Intelligence
Eötvös Loránd University
Budapest, Hungary

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Revison: Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Revison: Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$
$\boldsymbol{v}_2 = (5, 0, 4, 8)$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Revison: Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \dots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \dots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$$\boldsymbol{v}_1 = (\boxed{3}, 4, 2, 0)$$
$$\boldsymbol{v}_2 = (\boxed{5}, 0, 4, 8)$$

$$\boldsymbol{v}_1\ \boldsymbol{v}_2 = 3\text{x}5$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Revison: Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{wx} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$$\boldsymbol{v}_1 = (3, \boxed{4}, 2, 0)$$

$$\boldsymbol{v}_2 = (5, \boxed{0}, 4, 8)$$

$$\boldsymbol{v}_1\ \boldsymbol{v}_2 = 3\text{x}5 + 4\text{x}0$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Revison: Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{wx} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$
$\boldsymbol{v}_2 = (5, 0, 4, 8)$

$\boldsymbol{v}_1\ \boldsymbol{v}_2 = 3x5 + 4x0 + 2x4$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Revison: Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{wx} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$$\boldsymbol{v}_1 = (3, 4, 2, \boxed{0})$$
$$\boldsymbol{v}_2 = (5, 0, 4, \boxed{8})$$

$$\boldsymbol{v}_1\ \boldsymbol{v}_2 = 3x5 + 4x0 + 2x4 + 0x8$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Revison: Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$
$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{wx} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$$\boldsymbol{v}_1 = (3, 4, 2, \boxed{0})$$
$$\boldsymbol{v}_2 = (5, 0, 4, \boxed{8})$$

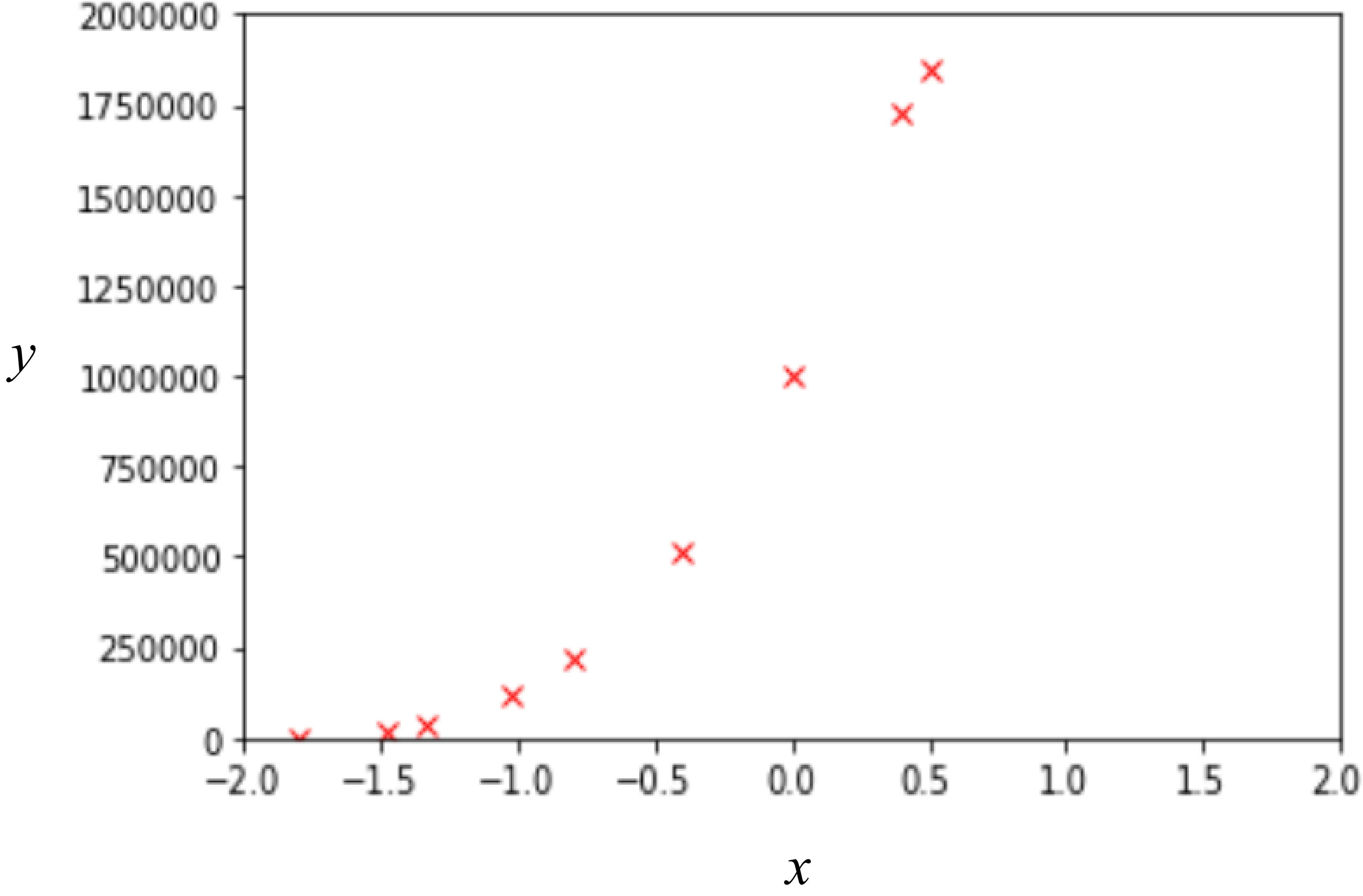
$$\boldsymbol{v}_1\ \boldsymbol{v}_2 = 3\text{x}5 + 4\text{x}0 + 2\text{x}4 + 0\text{x}8 = 15 + 0 + 8 + 0 = 23$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression

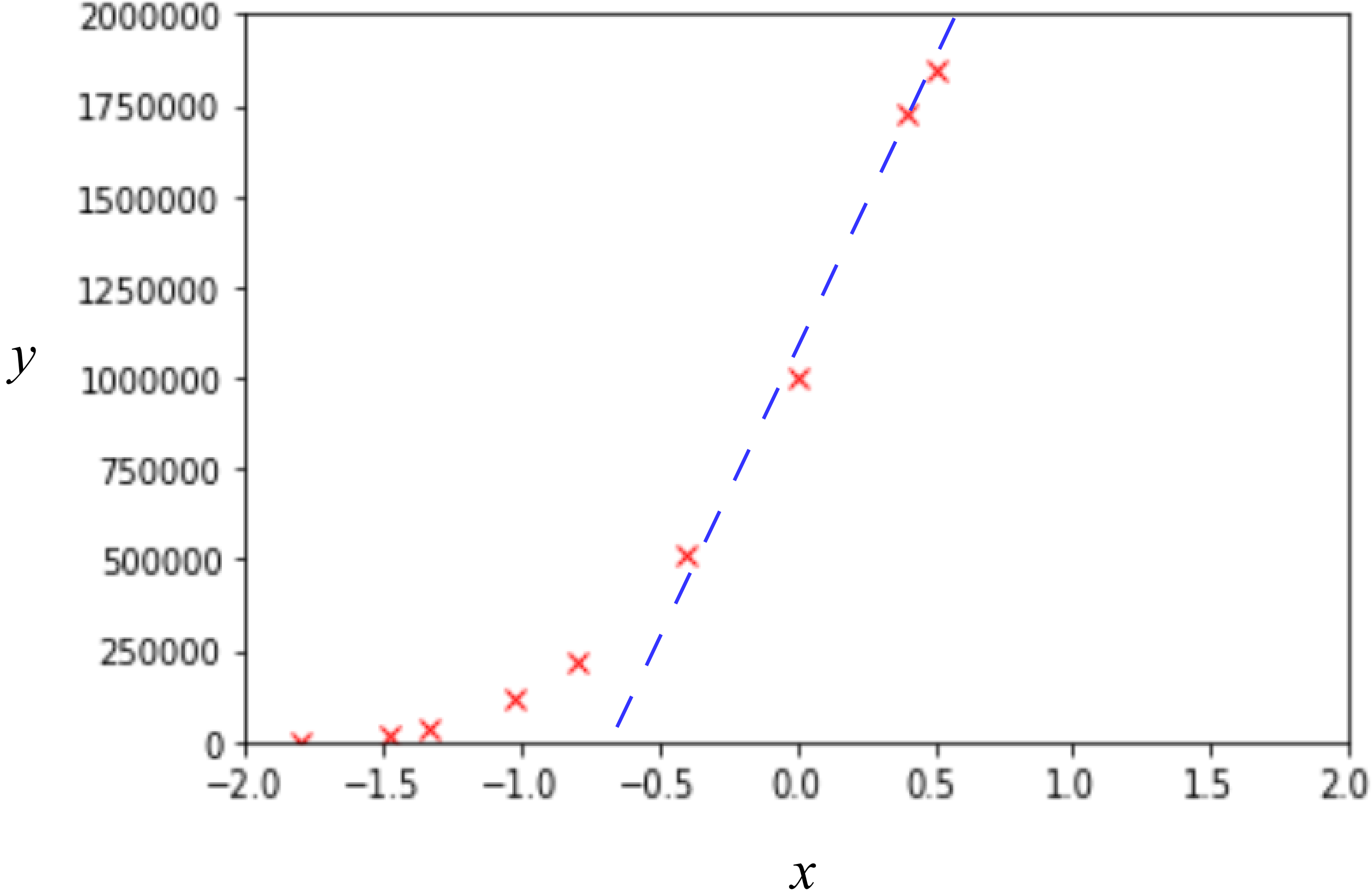
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Motivating Example



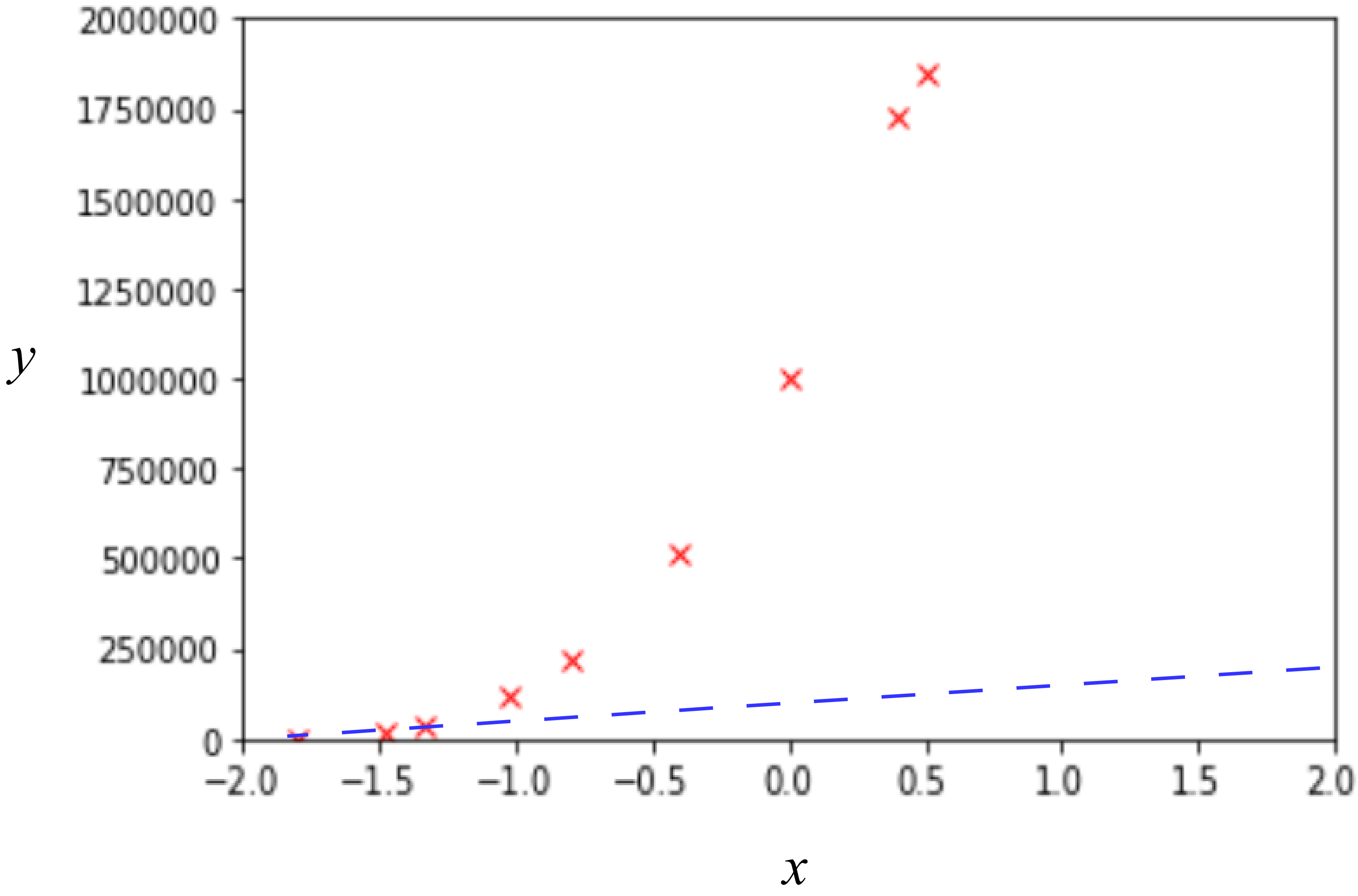
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Motivating Example



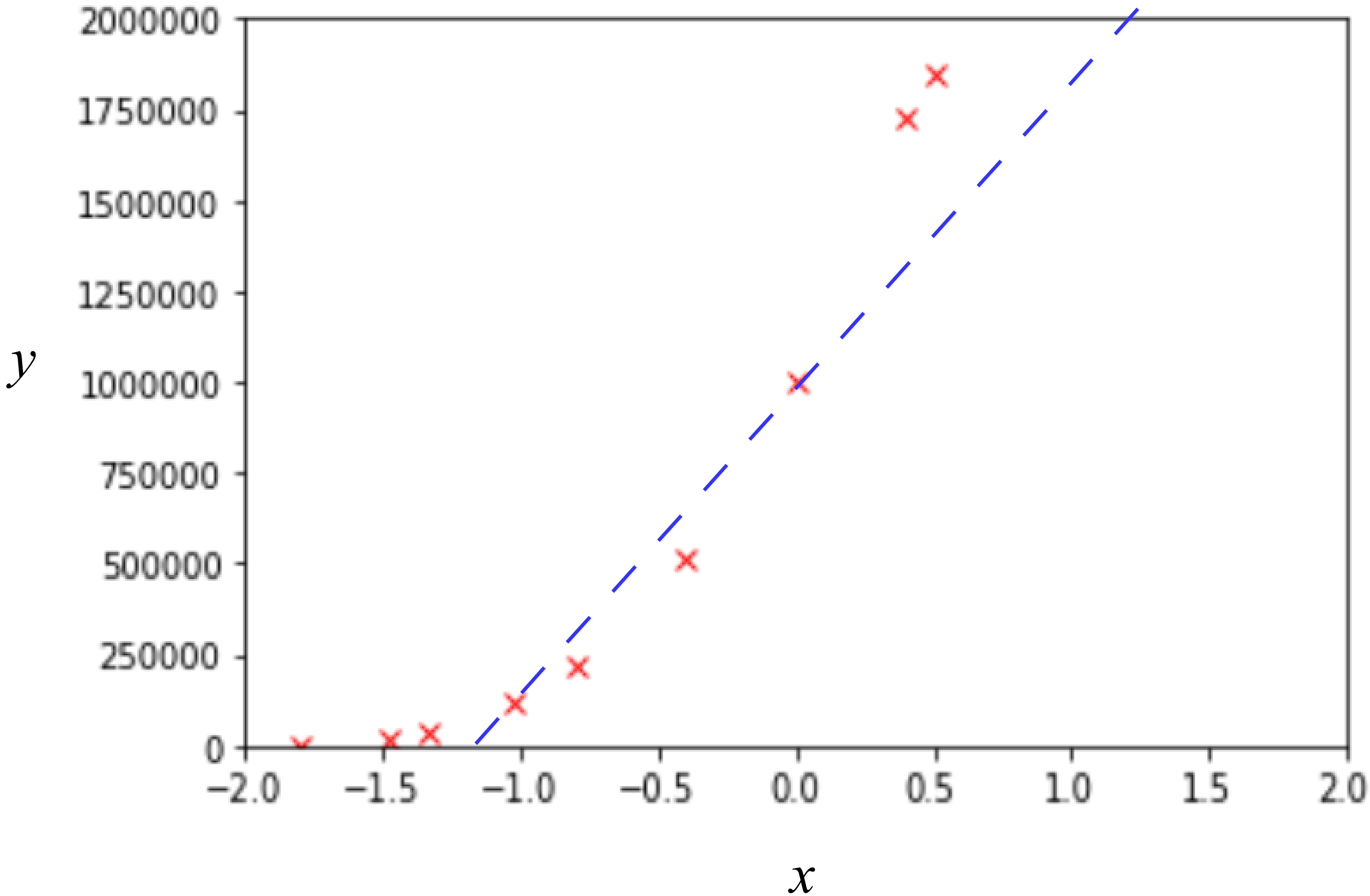
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Motivating Example



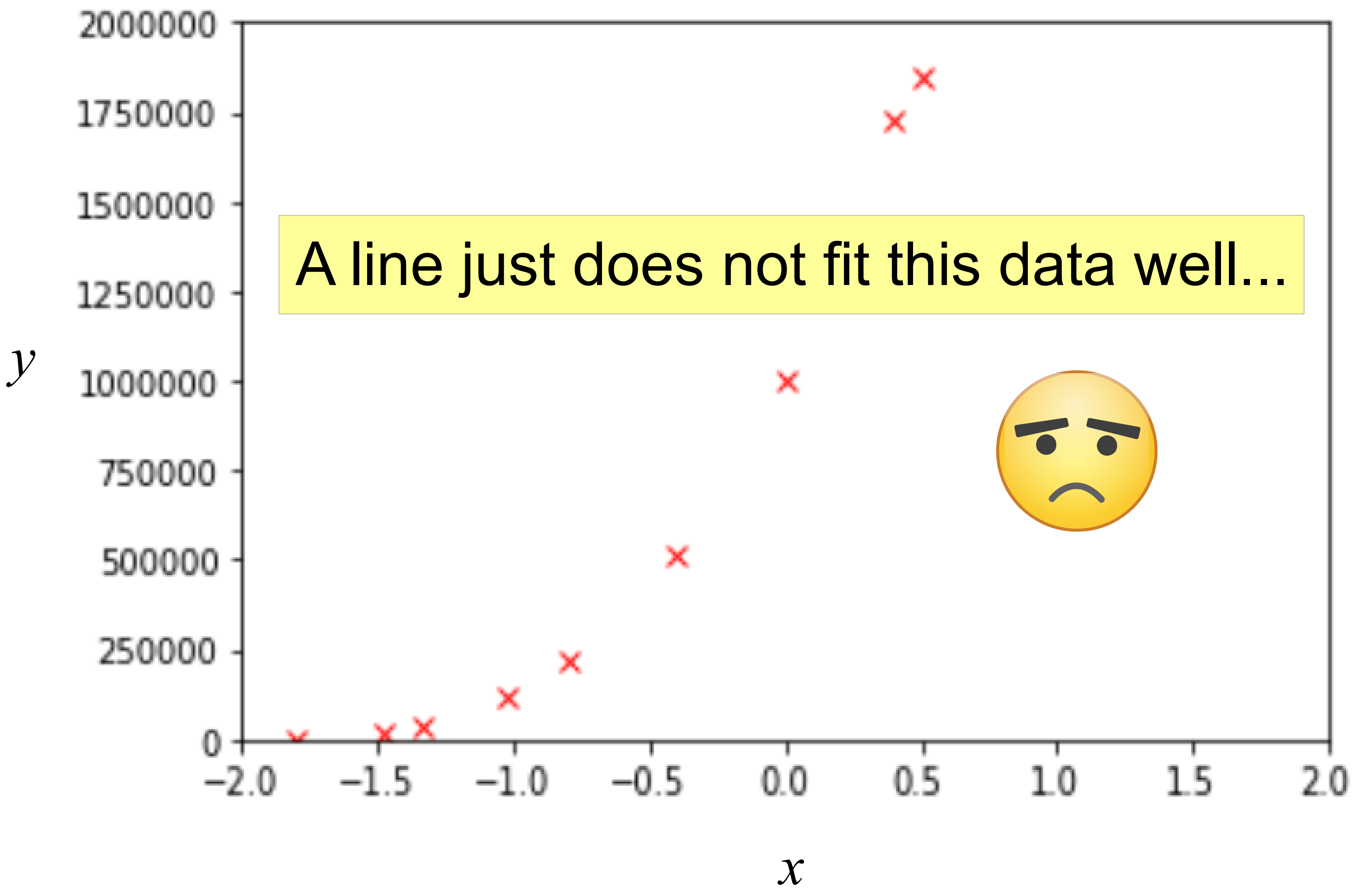
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Motivating Example



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Motivating Example



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with a Single Variable

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p$$

- Data

x	y
x_1	y_1
x_2	y_2
...	...
x_i	y_i
...	...
x_n	y_n

i-th instance

n

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with a Single Variable

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p =$$

$$= w^{(0)}x^0 + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p$$

- Data

x	y
x_1	y_1
x_2	y_2
...	...
x_i	y_i
...	...
x_n	y_n

i-th instance

n

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with a Single Variable

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p =$$

$$= w^{(0)}x^0 + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p = \sum_{j=0}^{p} w^{(j)}x^p$$

where p is the degree of the polynomial

- Data

x	y
x_1	y_1
x_2	y_2
…	…
x_i	y_i
…	…
x_n	y_n

n

i-th instance

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with a Single Variable

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p =$$

$$= w^{(0)}x^0 + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p = \sum_{j=0}^{p} w^{(j)}x^p$$

where p is the degree of the polynomial

- Data

x	y
x_1	y_1
x_2	y_2
…	…
x_i	y_i
…	…
x_n	y_n

n

i-th instance

How to learn the parameters of the model ?

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with a Single Variable

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p =$$

$$= w^{(0)}x^0 + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p = \sum_{j=0}^{p} w^{(j)}x^p$$

where p is the degree of the polynomial

- Data

x	y
x_1	y_1
x_2	y_2
...	...
x_i	y_i
...	...
x_n	y_n

n

i-th instance

Introduce new features and use linear regression

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with a Single Variable

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p =$$

$$= w^{(0)}x^0 + w^{(1)}x + w^{(2)}x^2 + \ldots + w^{(p)}x^p = \sum_{j=0}^{p} w^{(j)}x^p$$

- Data

i-th instance → x_i

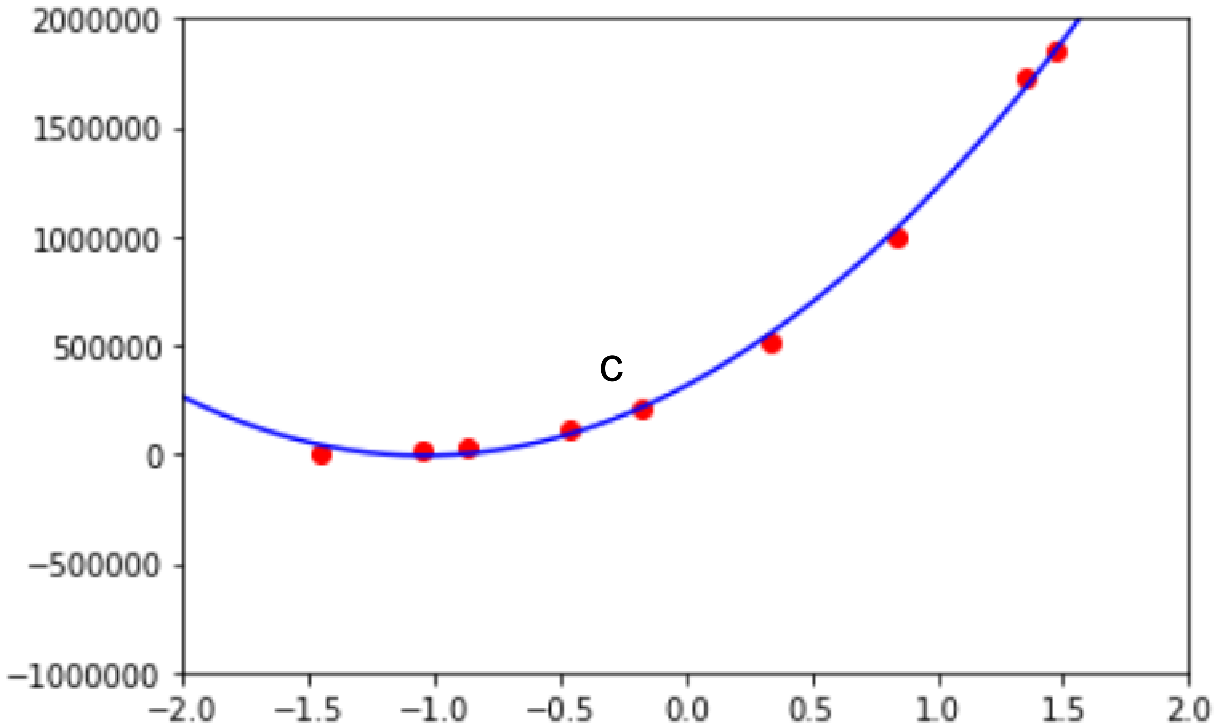
x	y
x_1	y_1
x_2	y_2
...	...
x_i	y_i
...	...
x_n	y_n

→

x	x^2	...	x^p	y
x_1	x_1^2	...	x_1^p	y_1
x_2	x_2^2	...	x_2^p	y_2
...	...	...	...	...
x_i	x_i^2	...	x_i^p	y_i
...	...	...	...	...
x_n	x_n^2	...	x_n^p	y_n

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

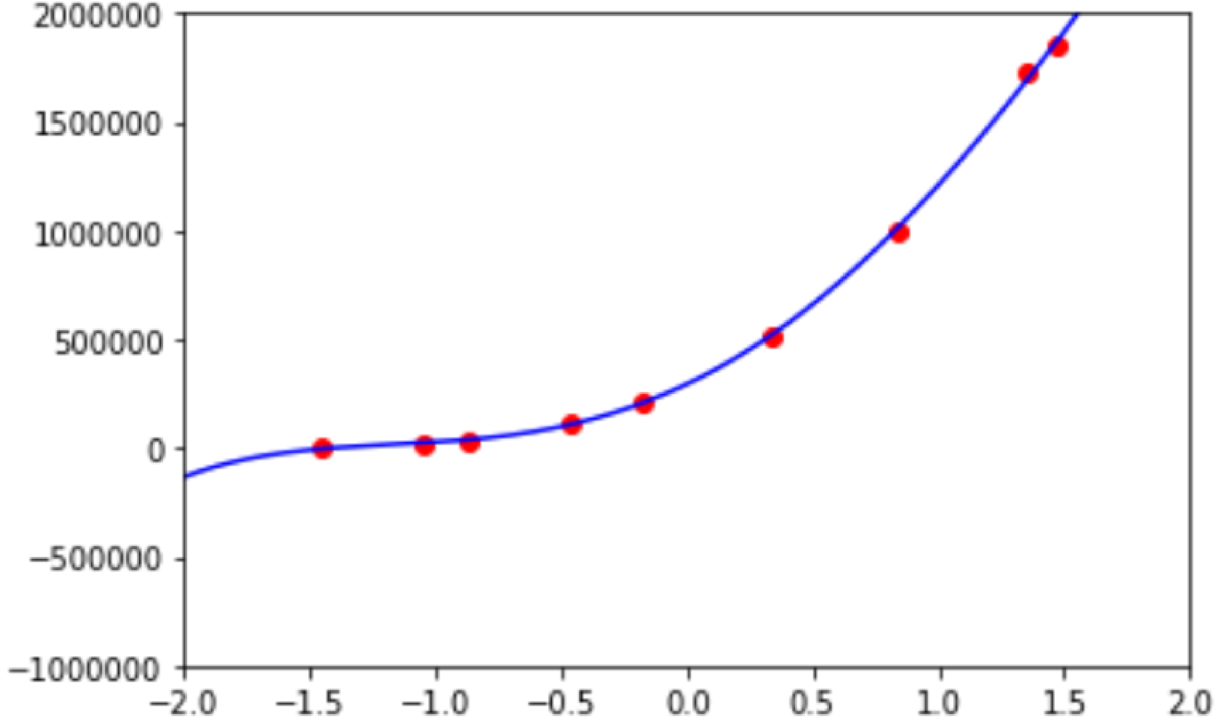
Polynomial of 2nd degree



RMSE = 32227

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

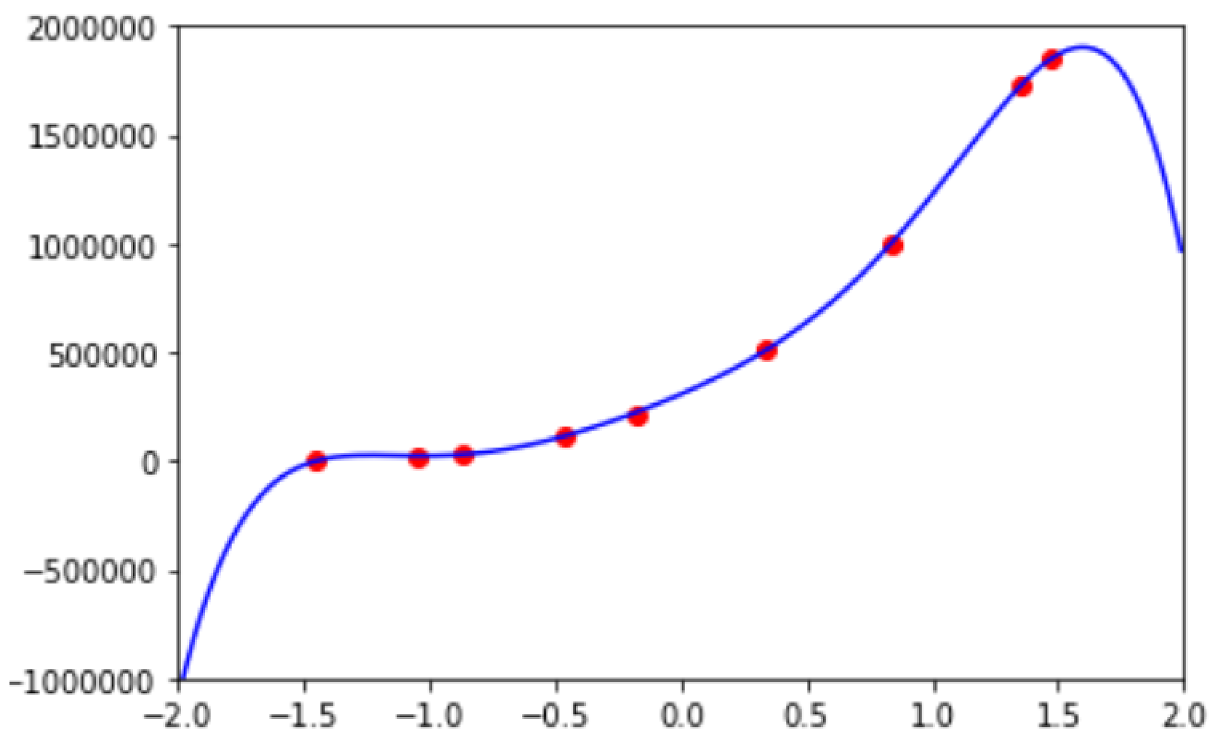
Polynomial of 4th degree



RMSE = 17566

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

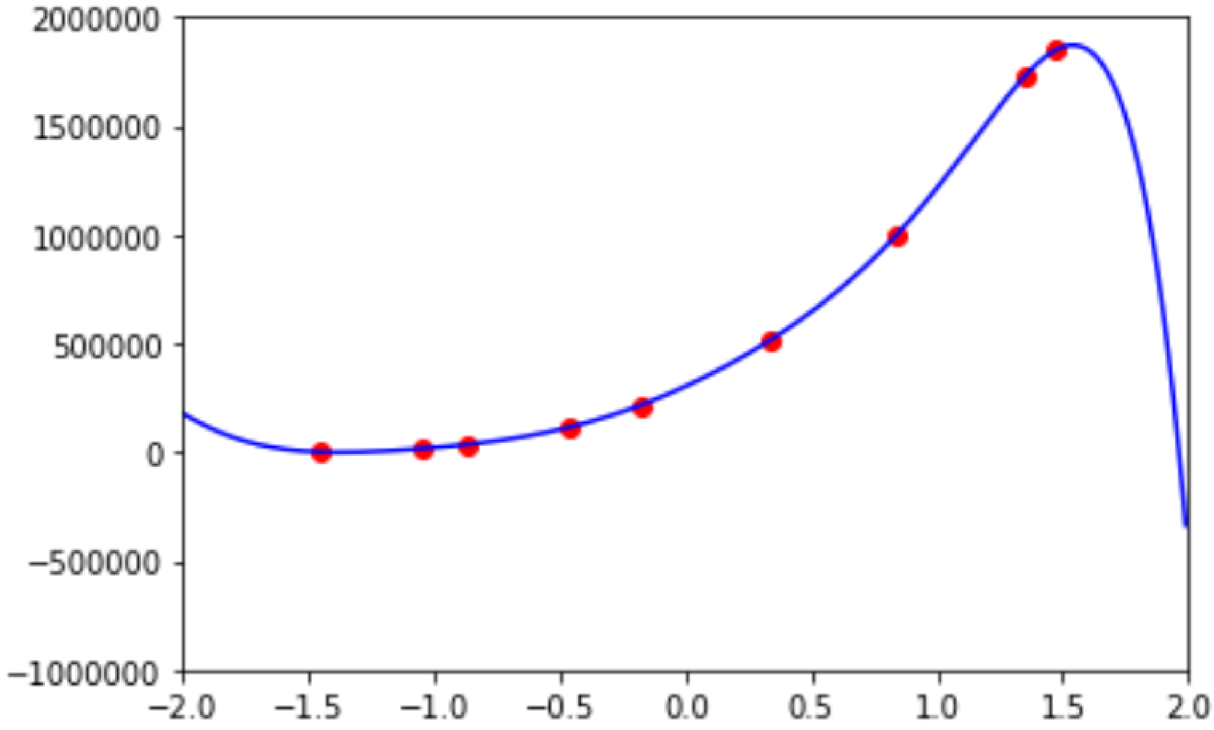
Polynomial of 6th degree



RMSE = 5206

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial of 8th degree

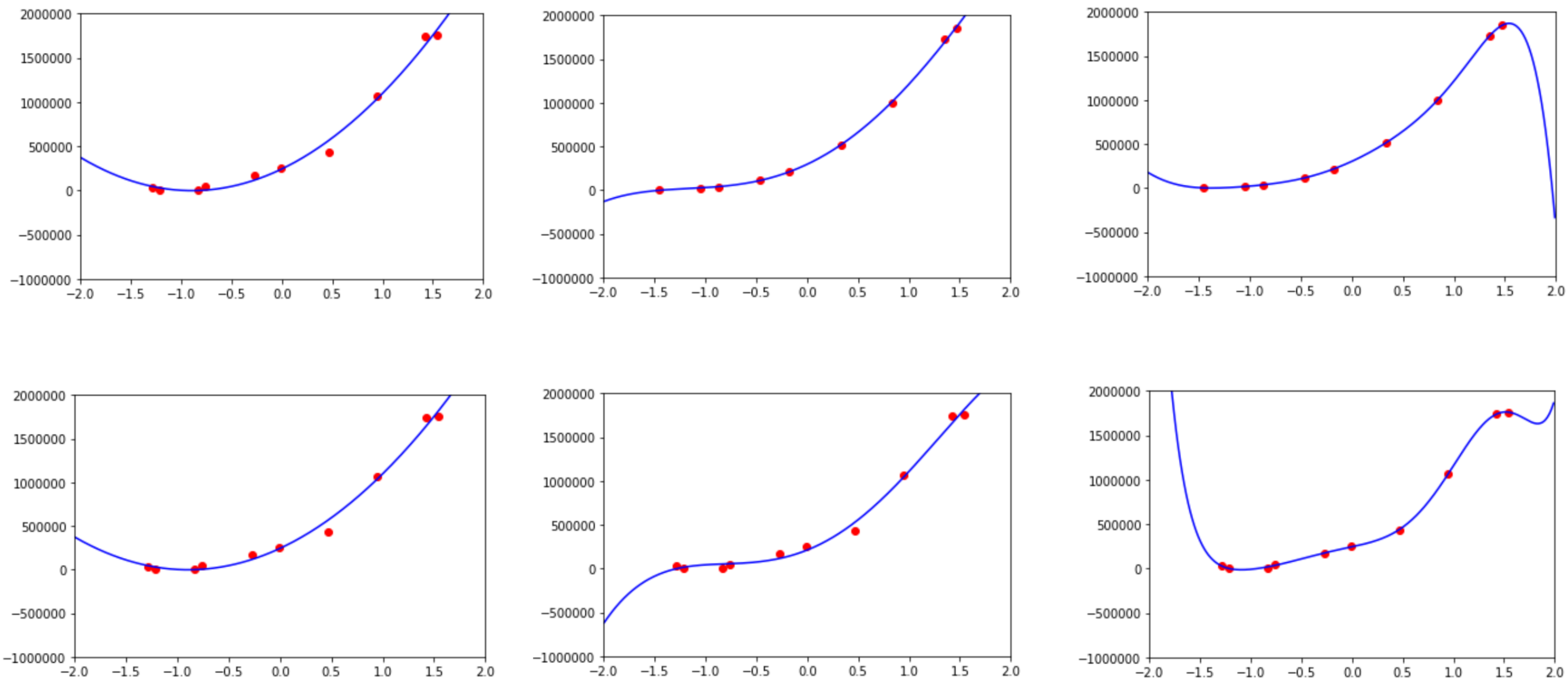


RMSE ≈ 0

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Let us Consider a Slightly Different Dataset...

$d = 2$ $\qquad$ $d = 4$ $\qquad$ $d = 8$



Top row: models fitted on the data we used on the previous slides
Bottom row: models fitted on slightly different data

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

What is the Problem with Polynomials of high degree (in the previous example) ?

1. For some new observation (e.g. $x = 1.9$ or $x = -1.8$), the error of the model may be **very high**.
2. Slightly different data may lead to a substantially different model.

- What happens? What is behind these problems?

The model approximates the training data extremely closely, in other words: the model overfits the data.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Is This a Real-World Problem?

„The Stop sign is misclassified into our target class of Speed Limit 45 in 100% of the images taken according to our evaluation methodology.“

Please see:
https://spectrum.ieee.org/cars-that-think/transportation/sensors/slight-street-sign-modifications-can-fool-machine-learning-algorithms

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Intermediate Summary

- **Overfitting**: the model fits the training data very well, but it does not generalize well to unseen instances
- Lessons learned:
  - In order to have a fair (unbiased) estimate of the error of the model, you must use **new*** data when you evaluate the model
    (*new data = data that has not been used while the model was trained)
  - The learning process must be modified in order to avoid overfitting

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Intermediate Summary

- **<u>Overfitting</u>**: the model fits the training data very well, but it does not generalize well to unseen instances
- Lessons learned:
  - In order to have a fair (unbiased) estimate of the error of the model, you must use **<u>new</u>*** data when you evaluate the model
    (*new data = data that has not been used while the model was trained)
    test data
  - The learning process must be modified in order to avoid overfitting

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Intermediate Summary

- **<u>Overfitting</u>**: the model fits the training data very well, but it does not generalize well to unseen instances
- Lessons learned:
  - In order to have a fair (unbiased) estimate of the error of the model, you must use **<u>new</u>*** data when you evaluate the model
    (*new data = data that has not been used while the model was trained)

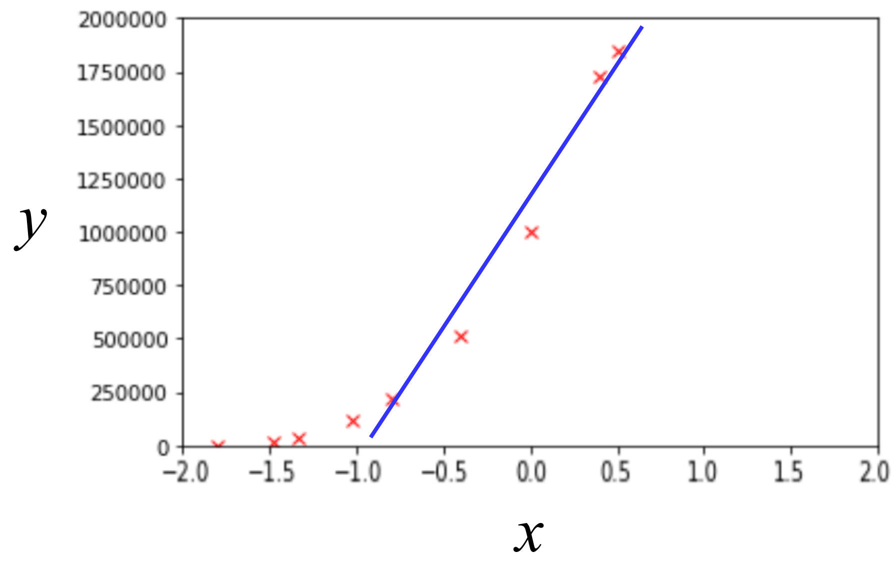
    test data
  - The learning process must be modified in order to avoid overfitting

<u>training data</u> (a.k.a. „train data“) - the data that has been used to determine the appropriate values of model parameters

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Underfitting

- The „opposite“ of overfitting
  - the model is too simple,
  - therefore, it is not able to appropriately capture the regularities.



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Regularization

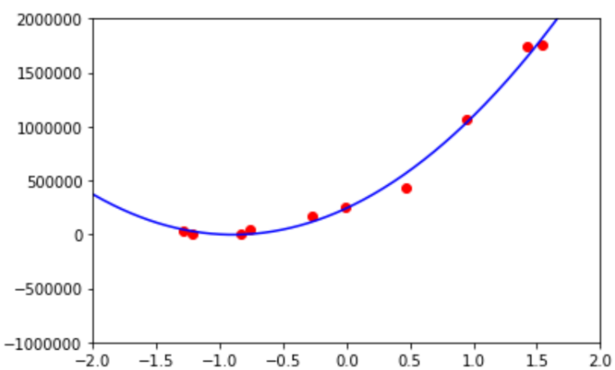
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How to Avoid Overfitting?

- Limit the degree of the polynomial
- In general: limit the complexity (capacity, expressive power) of the model
- Penalize models if the absolute value of the parameters is high

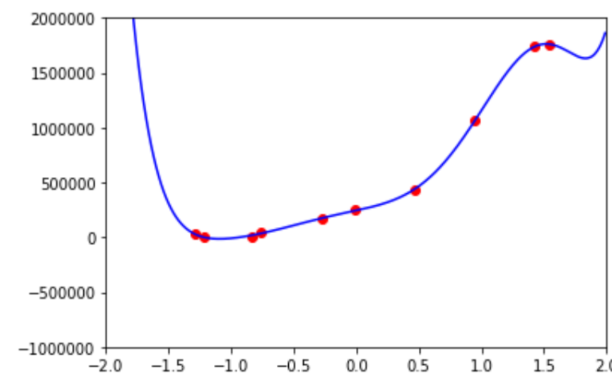
„Good“ and „Bad“ Models

$d = 2$



$$\hat{y} = 243232 + 548482x + \\ + 306067x^2$$

$d = 8$



$$\hat{y} = 244650 + 247270x + \\ + 61758x^2 + 468874x^3 + \\ + 501191x^4 + (-111649)x^5 + \\ + (-289177)x^6 + (-22128)x^7 + \\ + 54911x^8$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Regression with Regularization

- Objective function (a.k.a. cost function) without regularization (this is what we considered so far):

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

- Objective function with L_1 regularization:

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{i=1}^{p} |w^{(i)}|$$

- Objective function with L_2 regularization:

$$E = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{i=1}^{p} (w^{(i)})^2$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with L_2 Regularization

- Find appropriate parameter values with gradient descent
  - **Step 1** Set $w^{(0)}, w^{(1)}, ..., w^{(p)}$ to some random values
  - **Step 2** $w^{(0)} \leftarrow w^{(0)} - \epsilon \frac{1}{n} \sum_{i=1}^{n} 2(\hat{y}_i - y_i)$

    for j in $1...m$

    $$w^{(j)} \leftarrow w^{(j)} - \epsilon \Big(\frac{1}{n} \sum_{i=1}^{n} 2x_i^p (\hat{y}_i - y_i) + \lambda \, 2w^{(j)} \Big)$$

    where

    $$\hat{y}_i = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

  - **Step 3** Repeat Step 2 as long as you can non-negligibly decrease the error

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with L_1 Regularization

- Find appropriate parameter values with gradient descent
  - **Step 1** Set $w^{(0)}, w^{(1)}, \ldots, w^{(p)}$ to some random values
  - **Step 2** $w^{(0)} \leftarrow w^{(0)} - \epsilon \frac{1}{n} \sum_{i=1}^{n} 2(\hat{y}_i - y_i)$

    for j in $1...m$

    $$w^{(j)} \leftarrow w^{(j)} - \epsilon \Big(\frac{1}{n} \sum_{i=1}^{n} 2x_i^p (\hat{y}_i - y_i) + \lambda \, sgn(w^{(j)}) \Big)$$

    where $\hat{y}_i = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$

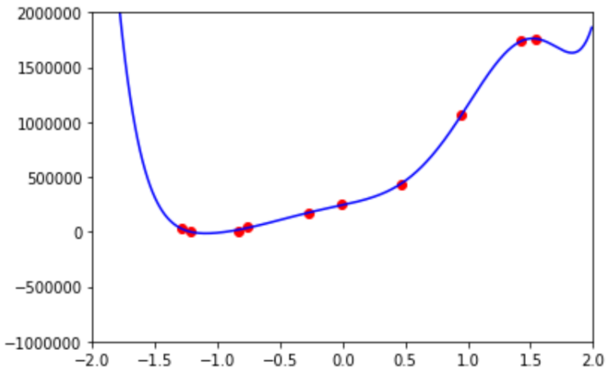
    $sgn(w^{(j)}) = 1$ if $w^{(j)} > 0$

    $sgn(w^{(j)}) = -1$ otherwise
  - **Step 3** Repeat Step 2 as long as you can non-negligibly decrease the error

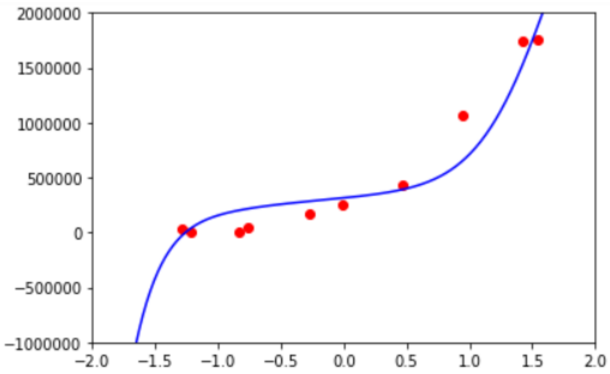
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Models without and with Regularization ($d = 8$)

without regularization



with L_2 regularization $\lambda = 7$



$$\hat{y} = 244650 + 247270x + 61758x^2 + 468874x^3 + 501191x^4 + (-111649)x^5 + (-289177)x^6 + (-22128)x^7 + 54911x^8$$

$$\hat{y} = 314642 + 115814x + 40039x^2 + 90224x^3 + 52954x^4 + 65828x^5 + 47012x^6 + 5949x^7 + (-21071)x^8$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Further Remarks on Regression

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Regularized Linear Regression

- Regularization can be applied to (multivariate) linear regression as well:
  - L_1 regularization → LASSO
  - L_2 regularization → Ridge Regression
- L_1 regularization encourages sparsity

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with Multiple Variables

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1,1)}x^{(1)} + w^{(1,2)}x^{(2)} + \ldots +$$
$$+ w^{(2,1)}(x^{(1)})^2 + w^{(2,2)}(x^{(2)})^2 + w^{(2,3)}(x^{(1)}x^{(2)})$$

- Data

m

$x^{(1)}$	$x^{(2)}$	…	$x^{(m)}$	y
$x_1^{(1)}$	$x_1^{(2)}$	…	$x_1^{(m)}$	y_1
$x_2^{(1)}$	$x_2^{(2)}$	…	$x_2^{(m)}$	y_2
…	…	…	…	…
$x_i^{(1)}$	$x_i^{(2)}$	…	$x_i^{(m)}$	y_i
…	…	…	…	…
$x_n^{(1)}$	$x_n^{(2)}$	…	$x_n^{(m)}$	y_n

i-th instance

n

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Polynomial Regression with Multiple Variables

- We consider models of the form

$$\hat{y} = w^{(0)} + w^{(1,1)}x^{(1)} + w^{(1,2)}x^{(2)} + \ldots + \\ + w^{(2,1)}(x^{(1)})^2 + w^{(2,2)}(x^{(2)})^2 + w^{(2,3)}(x^{(1)}x^{(2)})$$

$x^{(1)}$	$x^{(2)}$	…	$x^{(m)}$	y
$x_1^{(1)}$	$x_1^{(2)}$	…	$x_1^{(m)}$	y_1
$x_2^{(1)}$	$x_2^{(2)}$	…	$x_2^{(m)}$	y_2
…	…	…	…	…
$x_i^{(1)}$	$x_i^{(2)}$	…	$x_i^{(m)}$	y_i
…	…	…	…	…
$x_n^{(1)}$	$x_n^{(2)}$	…	$x_n^{(m)}$	y_n

new features corresponding to the model

$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	…	$x^{(m)}$	$(x^{(1)})^2$	$(x^{(2)})^2$	$x^{(1)}x^{(2)}$	…	y
1	$x_1^{(1)}$	$x_1^{(2)}$	…	$x_1^{(m)}$	$(x_1^{(1)})^2$	$(x_1^{(2)})^2$	$x_1^{(1)}x_1^{(2)}$	…	y_1
1	$x_2^{(1)}$	$x_2^{(2)}$	…	$x_2^{(m)}$	$(x_2^{(1)})^2$	$(x_2^{(2)})^2$	$x_2^{(1)}x_2^{(2)}$	…	y_2
…	…	…	…	…	…	…	…	…	…
1	$x_i^{(1)}$	$x_i^{(2)}$	…	$x_i^{(m)}$	$(x_i^{(1)})^2$	$(x_i^{(2)})^2$	$x_i^{(1)}x_i^{(2)}$	…	y_i
…	…	…	…	…	…	…	…	…	…
1	$x_n^{(1)}$	$x_n^{(2)}$	…	$x_n^{(m)}$	$(x_n^{(1)})^2$	$(x_n^{(2)})^2$	$x_n^{(1)}x_n^{(2)}$		y_n

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Factorization Machines (optional)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Factorization Machines

- We want to consider all pairwise interactions

$$\hat{y} = w^{(0)} + w^{(1)}x^{(1)} + w^{(2)}x^{(2)} + w^{(3)}x^{(3)} + \ldots +$$
$$+ w^{(1,2)}(x^{(1)}x^{(2)}) + w^{(1,3)}(x^{(1)}x^{(3)}) + \ldots$$

but: we do not want to introduce quadratically many parameters (and features)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Factorization Machines

- For each of the variables, we introduce a vector
$$\mathbf{v}^{(i)} = (v^{(i,1)}, \ldots, v^{(i,f)})$$
so that the scalar product of two vectors will be characteristic to the interaction between the two variables.
- Thus, the model equation is:
$$\hat{y}(\mathbf{x}) = w^{(0)} + \sum_{i=1}^{n} w^{(i)} x^{(i)} + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \Big(\sum_{k=1}^{f} v^{(i,k)} v^{(j,k)} \Big) x^{(i)} x^{(j)}$$

S. Rendle (2010): Factorization machines. 10th IEEE International Conference on Data Mining (ICDM), pp. 995–1000.

Algorithm Training the Factorization Machine*

Require: Training data D, number of epochs e, learning rate ϵ, standard deviation σ

Ensure: Weights $w^{(0)}, w^{(1)}, \dots w^{(m)}$ and $v^{(1,1)}, \dots, v^{(n,f)}$

Initialize $w^{(0)}, w^{(1)}, \dots w^{(m)}$ and $v^{(1,1)}, \dots, v^{(n,f)}$ from standard normal distribution with zero mean and standard deviation σ

for epoch in $1 \dots e$ **do**

  for each $(x, y) \in D$ in random order **do**

$$\hat{y} \leftarrow w^{(0)} + \sum_{i=1}^{m} w^{(i)} x^{(i)} + \sum_{i=1}^{m} \sum_{j=i+1}^{m} \left(\sum_{k=1}^{f} v^{(i,k)} v^{(j,k)} \right) x^{(i)} x^{(j)}$$

$$w^{(0)} \leftarrow w^{(0)} - \epsilon \; 2(\hat{y} - y)$$

    for i in $1 \dots m$ **do**

$$w^{(i)} \leftarrow w^{(i)} - \epsilon \; 2(\hat{y} - y) x^{(i)}$$

    for i in $1 \dots m$ **do**

      for j in $1 \dots f$ **do**

$$v^{(i,j)} \leftarrow v^{(i,j)} - \epsilon \; 2(\hat{y} - y)(x^{(i)} \sum_{k=1}^{m} v^{(k,j)} x^{(k)} - v^{(i,j)} (x^{(i)})^2)$$

return $w^{(0)}, w^{(1)}, \dots, w^{(m)}$ and $v^{(1,1)}, \dots, v^{(n,f)}$

*** Stochastic Gradient Descent**

K. Buza, T. Horváth (2019): Factorization Machines for Blog Feedback Prediction, 11th International Conference on Computer Recognition Systems (CORES)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Summary

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Summary

- Polynomial Regression
- Overfitting
  - It is important in general!
  - **In order to have an unbiased estimate of the accuracy of the model, a new dataset (test data) should be used when the model is evaluated**
- Regularization
- Factorization Machines

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Essential Concepts

- Overfitting
- Underfitting
- Degree of a Polynomial
- Regularization
- Training data
- Test data
- Lasso
- Ridge Regression
- (Factorization machine)
- (Stochastic Gradient Descent)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Presentations' about Social, Economical, Technological or Moral Aspects of AI

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Students' Presentations' about Social, Economical, Technological or Moral Aspects of AI

- Form groups of 2-3 students
- Each group
  - should select a topic → e-mail to buza@inf.elte.hu
    deadline: 9th October 2019

    you are welcome to propose **new topics**, the ones on the next slides are possible topics, but you do not need to select from that list

    (the same topic will only be assigned to one group, „First Come – First Served“)
  - is expected to deliver a short presentation on the 20th or 27th November (≈10 minutes, 4 – 6 slides)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Possible topics

- Roles in a machine learning team, job titles
- Steps of a machine learning and data science project
- AI transformation playbook
  https://landing.ai/ai-transformation-playbook/
- AI pitfalls
- Short survey of AI applications (application domains)
- Short survey of AI techniques (methods)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Possible topics

- General ethical guidelines
- Ethical issues and guidelines related to a particular application (e.g. self-driving car)
- AI in the light of a selected religion
- legal framework of AI (most essential legal principles)
- AI vs. privacy (Siri)

- Sofia (robot, citizen of Saudi Arabia)
- IBM Watson (and „his“ medical studies)

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Possible topics

- AI applications using
  - Amazon Web Services
  - Google Collab
  - Hungarian HPC
- What is GPU and TPU?
- Cython
- Clean code guidelines for machine learning
- Presentation of a Python library
- Gartners curve – overoptimism?
- Agile Software Development for machine learning
- Gradient Descent (GD) variants: Stochastic GD and Batch GD – advantages and disadvantages compared with „standard“ GD