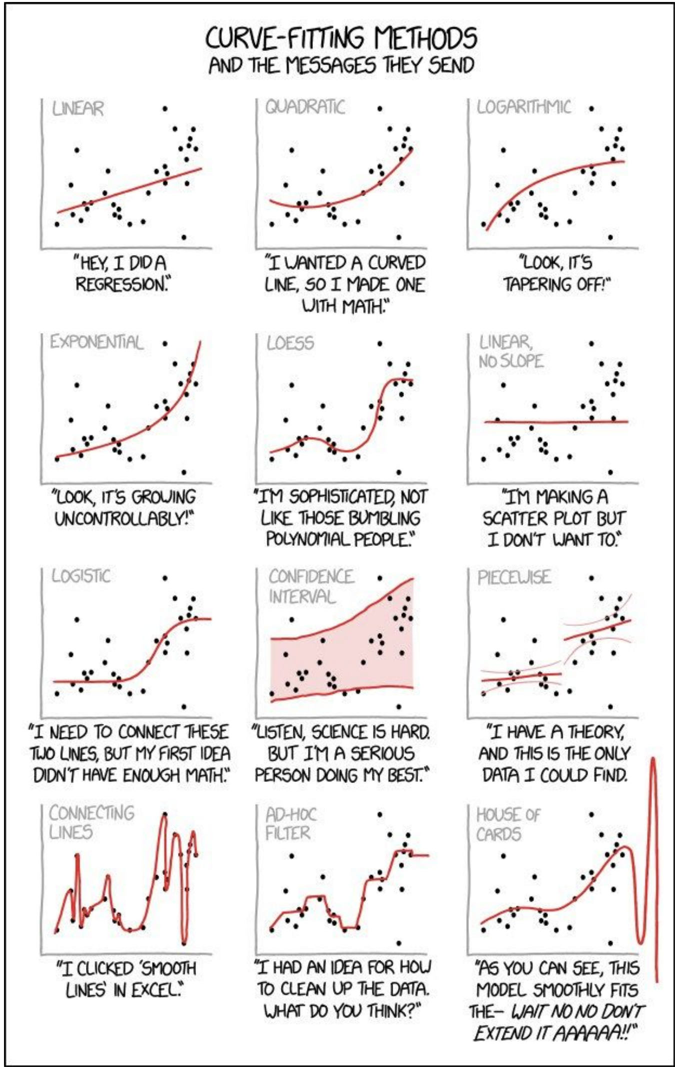
Machine Learning Course

Lecture 2: Linear models

MIPT, 2019

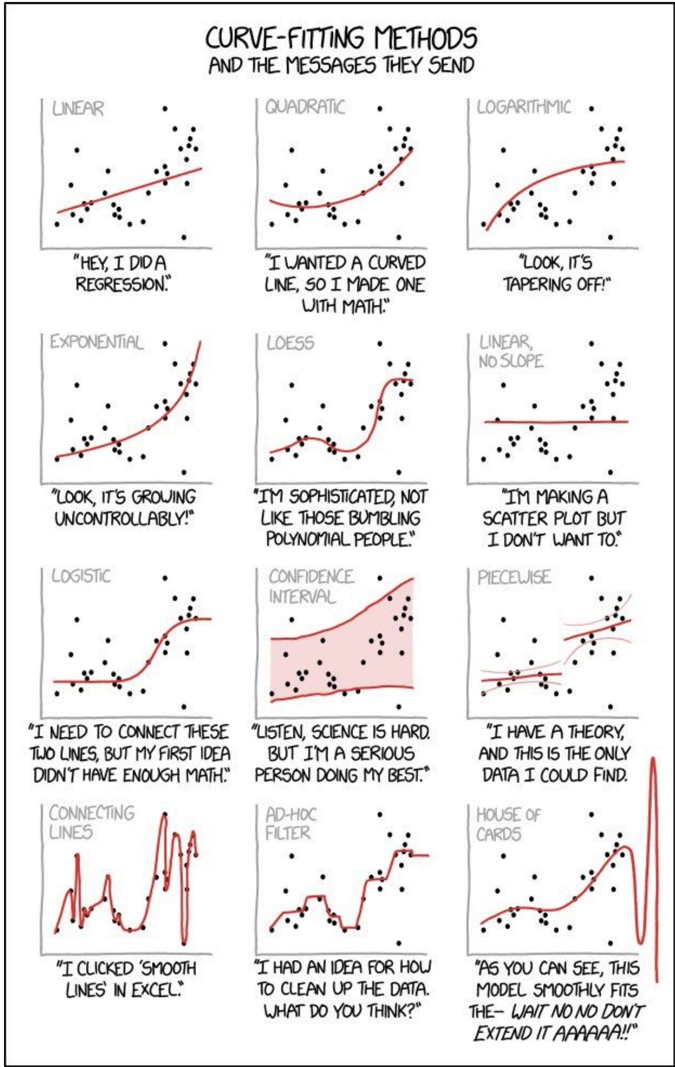
Outline

1. Overview of linear models.
2. Linear regression.
3. Analytical solution.
4. Regularization.
5. Gauss-Markov theorem.
6. Probabilistic interpretation and intuition.


CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
LINEAR, NO SLOPE
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO."
LOGISTIC
"I NEED TO CONNECT THESE TWO LINES, BUT MY FIRST IDEA DIDN'T HAVE ENOUGH MATH."
CONFIDENCE INTERVAL
"LISTEN, SCIENCE IS HARD. BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"I HAVE A THEORY, AND THIS IS THE ONLY DATA I COULD FIND."
CONNECTING LINES
"I CLICKED 'SMOOTH LINES' IN EXCEL."
AD-HOC FILTER
"I HAD AN IDEA FOR HOW TO CLEAN UP THE DATA. WHAT DO YOU THINK?"
HOUSE OF CARDS
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE- WAIT NO NO DON'T EXTEND IT AAAAAA!!"

Example questions linear regression can solve (up right picture):

- What will be my monthly spending for the next year?
- Which factor is more important in deciding my monthly spending?
- How monthly income and trips per month are correlated with monthly spending?


CURVE-FITTING METHODS
AND THE MESSAGES THEY SEND
LINEAR
"HEY, I DID A REGRESSION."
QUADRATIC
"I WANTED A CURVED LINE, SO I MADE ONE WITH MATH."
LOGARITHMIC
"LOOK, IT'S TAPERING OFF!"
EXPONENTIAL
"LOOK, IT'S GROWING UNCONTROLLABLY!"
LOESS
"I'M SOPHISTICATED, NOT LIKE THOSE BUMBLING POLYNOMIAL PEOPLE."
LINEAR, NO SLOPE
"I'M MAKING A SCATTER PLOT BUT I DON'T WANT TO."
LOGISTIC
"I NEED TO CONNECT THESE TWO LINES, BUT MY FIRST IDEA DIDN'T HAVE ENOUGH MATH."
CONFIDENCE INTERVAL
"LISTEN, SCIENCE IS HARD. BUT I'M A SERIOUS PERSON DOING MY BEST."
PIECEWISE
"I HAVE A THEORY, AND THIS IS THE ONLY DATA I COULD FIND."
CONNECTING LINES
"I CLICKED 'SMOOTH LINES' IN EXCEL."
AD-HOC FILTER
"I HAD AN IDEA FOR HOW TO CLEAN UP THE DATA. WHAT DO YOU THINK?"
HOUSE OF CARDS
"AS YOU CAN SEE, THIS MODEL SMOOTHLY FITS THE- WAIT NO NO DON'T EXTEND IT AAAAAA!!"

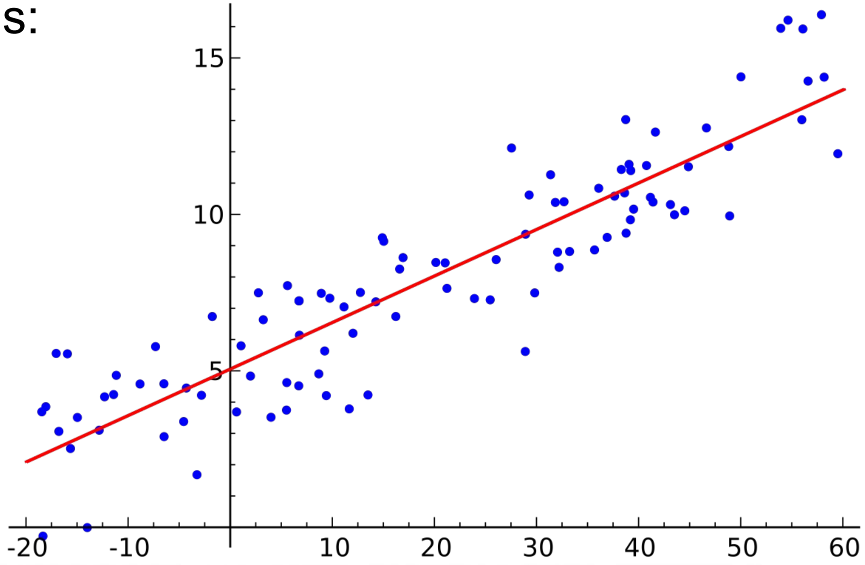
Linear models

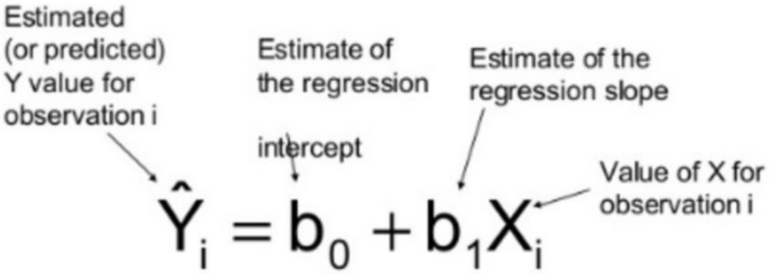
$$\underset{\downarrow}{Y} = \underbrace{X_1 + X_2 + X_3}_{\downarrow}$$

Dependent Variable	Independent Variable
Outcome Variable	Predictor Variable
Response Variable	Explanatory Variable

Linear models

- Predictive models:





Estimated (or predicted) Y value for observation i

Estimate of the regression intercept

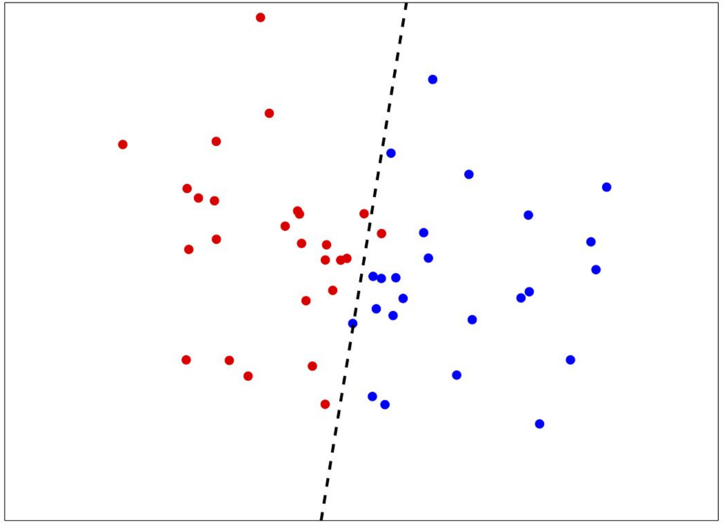
Estimate of the regression slope

Value of X for observation i

$$\hat{Y}_i = b_0 + b_1 X_i$$

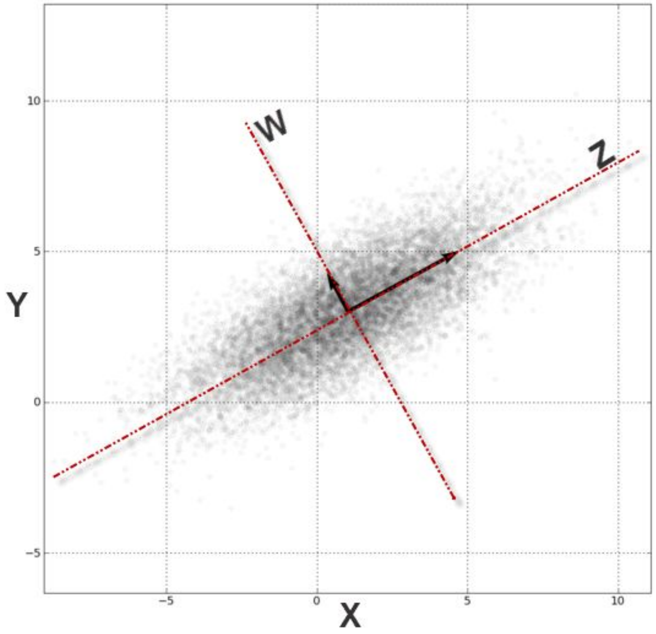
Linear models

- Predictive models:
- Classification models:



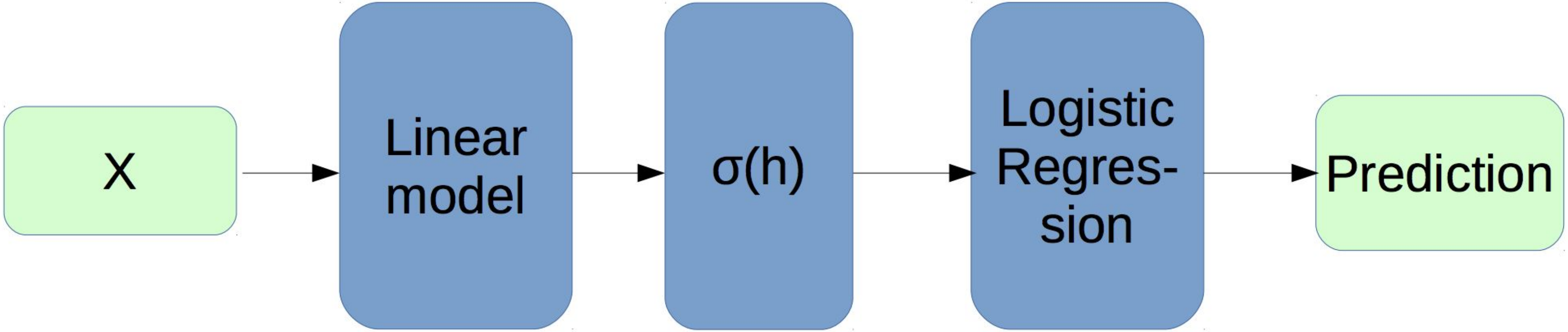
Linear models

- Predictive models:
- Classification models:
- Unsupervised models (e.g. PCA analysis)



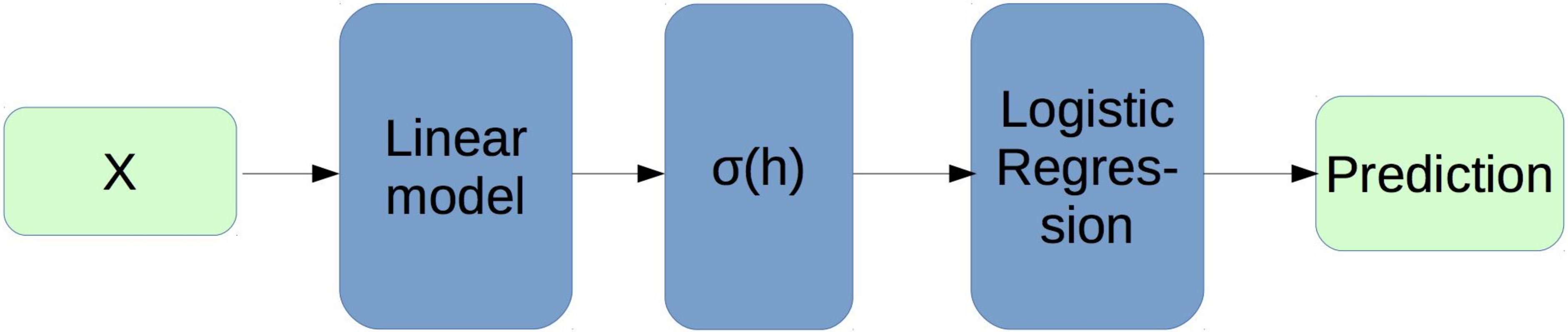
Linear models

- Predictive models:
- Classification models:
- Unsupervised models (e.g. PCA analysis)
- Building block of other models (ensembles, NNs, etc.)



Linear models

- Predictive models:
- Classification models:
- Unsupervised models (e.g. PCA analysis)
- Building block of other models (ensembles, NNs, etc.)



Actually, it’s a neural network. We will meet it later.

Linear regression

Linear regression problem statement:

- Training set $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, where ($\mathbf{x} \in \mathbb{R}^p$, $y \in \mathbb{R}$) .

Linear regression

Linear regression problem statement:

- Training set $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, where $(\mathbf{x} \in \mathbb{R}^p,\ y \in \mathbb{R}\ )$.

- Prediction model is linear: $\hat{y}_i = a(w_0, \mathbf{w}, \mathbf{x}_i) = w_0 + w_1 x_{i1} + \ldots w_p x_{ip}$,

  Where $\mathbf{w} = (w_1, \ldots w_p)$ is weights vector, w_0 is bias term.

Linear regression

Linear regression problem statement:

- Training set $\mathcal{L} = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, where $(\mathbf{x} \in \mathbb{R}^p,\ y \in \mathbb{R}\ )$.

- Prediction model is linear: $\hat{y}_i = a(w_0, \mathbf{w}, \mathbf{x}_i) = w_0 + w_1 x_{i1} + \dots w_p x_{ip}$,

  Where $\mathbf{w} = (w_1, \dots w_p)$ is weights vector, w_0 is bias term.

- Least squares method provides a solution:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{w}_0 + (\mathbf{x}_1 \dots \mathbf{x}_n)^T \mathbf{w} - (y_1, \dots, y_n)\|_2^2$$

Analytical solution

Denote quadratic loss function: $Q(\mathbf{w}) = (Y - X\mathbf{w})^T(Y - X\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2$,

where $X = [\mathbf{x}_1, \dots, \mathbf{x}_n], \quad \mathbf{x}_i \in \mathbb{R}^p$, $Y = [y_1, \dots, y_n], \quad y_i \in \mathbb{R}$.

Analytical solution

Denote quadratic loss function: $Q(\mathbf{w}) = (Y - X\mathbf{w})^T(Y - X\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2$,

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathbf{x}_i \in \mathbb{R}^p$, $Y = [y_1, \ldots, y_n], \quad y_i \in \mathbb{R}$.

To find optimal solution let's equal to zero the derivative of the equation above:

Analytical solution

Denote quadratic loss function: $Q(\mathbf{w}) = (Y - X\mathbf{w})^T(Y - X\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2$,

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathbf{x}_i \in \mathbb{R}^p$, $Y = [y_1, \ldots, y_n], \quad y_i \in \mathbb{R}$.

To find optimal solution let's equal to zero the derivative of the equation above:

$$\nabla_{\mathbf{w}} Q(\mathbf{w}) = \nabla_{\mathbf{w}}[Y^T Y - Y^T X\mathbf{w} - \mathbf{w}^T X^T Y + \mathbf{w}^T X^T X\mathbf{w}] =$$

Analytical solution

Denote quadratic loss function: $Q(\mathbf{w}) = (Y - X\mathbf{w})^T(Y - X\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2$,

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathbf{x}_i \in \mathbb{R}^p$, $Y = [y_1, \ldots, y_n], \quad y_i \in \mathbb{R}$.

To find optimal solution let’s equal to zero the derivative of the equation above:

$$\nabla_{\mathbf{w}} Q(\mathbf{w}) = \nabla_{\mathbf{w}}[Y^T Y - Y^T X\mathbf{w} - \mathbf{w}^T X^T Y + \mathbf{w}^T X^T X\mathbf{w}] =$$

$$= 0 - X^T Y - X^T Y + (X^T X + X^T X)\mathbf{w} = 0$$

Analytical solution

Denote quadratic loss function: $Q(\mathbf{w}) = (Y - X\mathbf{w})^T(Y - X\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2$,

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathbf{x}_i \in \mathbb{R}^p$, $Y = [y_1, \ldots, y_n], \quad y_i \in \mathbb{R}$.

To find optimal solution let’s equal to zero the derivative of the equation above:

$$\nabla_{\mathbf{w}} Q(\mathbf{w}) = \nabla_{\mathbf{w}}[Y^T Y - Y^T X\mathbf{w} - \mathbf{w}^T X^T Y + \mathbf{w}^T X^T X\mathbf{w}] =$$
$$= 0 - X^T Y - X^T Y + (X^T X + X^T X)\mathbf{w} = 0$$

So the target vector **w** should be

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T Y$$

Analytical solution

Denote quadratic loss function: $Q(\mathbf{w}) = (Y - X\mathbf{w})^T(Y - X\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2$,

where $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n], \quad \mathbf{x}_i \in \mathbb{R}^p, \; Y = [y_1, \ldots, y_n], \quad y_i \in \mathbb{R}.$

To find optimal solution let's equal to zero the derivative of the equation above:

$$\nabla_{\mathbf{w}} Q(\mathbf{w}) = \nabla_{\mathbf{w}} [Y^T Y - Y^T X\mathbf{w} - \mathbf{w}^T X^T Y + \mathbf{w}^T X^T X\mathbf{w}] =$$
$$= 0 - X^T Y - X^T Y + (X^T X + X^T X)\mathbf{w} = 0$$

So the target vector **w** should be

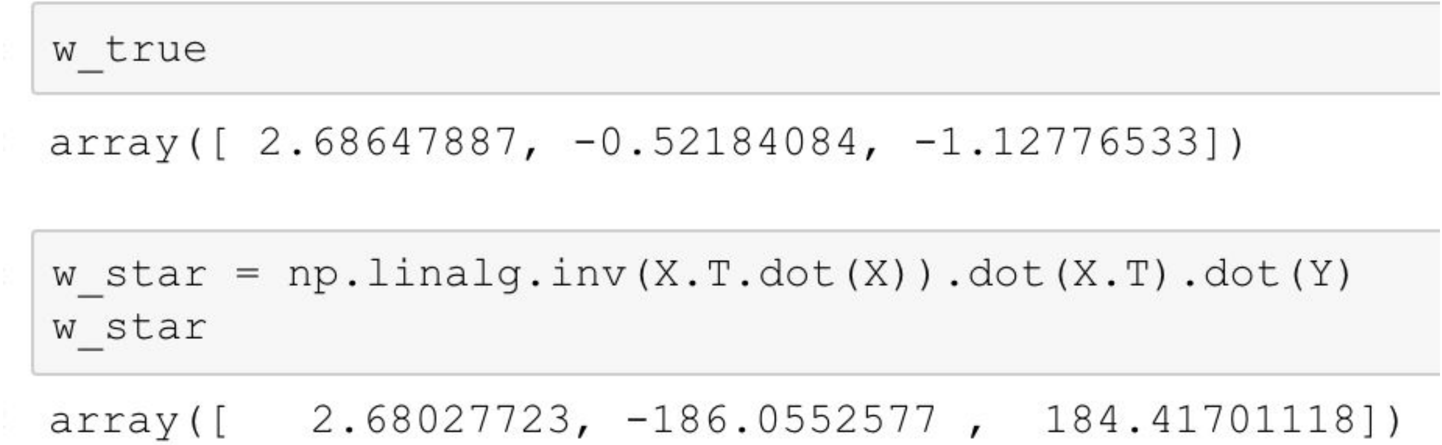
$$\hat{\mathbf{w}} = \boxed{(X^T X)}^{-1} X^T Y$$

what if this matrix is *singular?*

Unstable solution

In case of multicollinear features the matrix X^TX is almost singular .

It leads to unstable solution:



```
w_true
```

```
array([ 2.68647887, -0.52184084, -1.12776533])
```

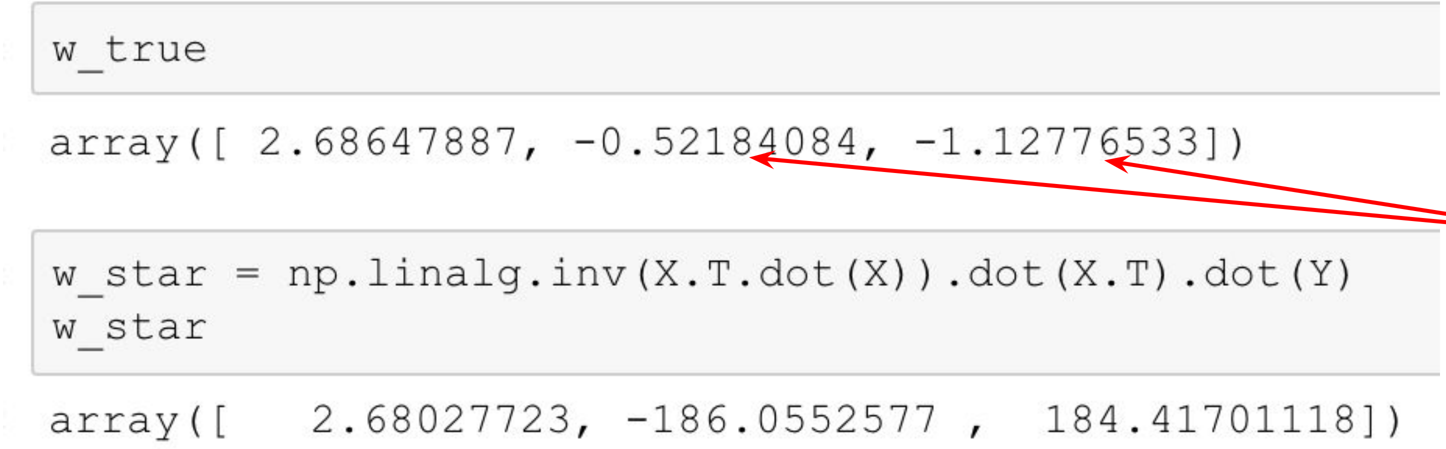
```
w_star = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(Y)
w_star
```

```
array([   2.68027723, -186.0552577 ,  184.41701118])
```

Unstable solution

In case of multicollinear features the matrix $X^T X$ is almost singular .

It leads to unstable solution:



```
w_true
```

```
array([ 2.68647887, -0.52184084, -1.12776533])
```

```
w_star = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(Y)
w_star
```

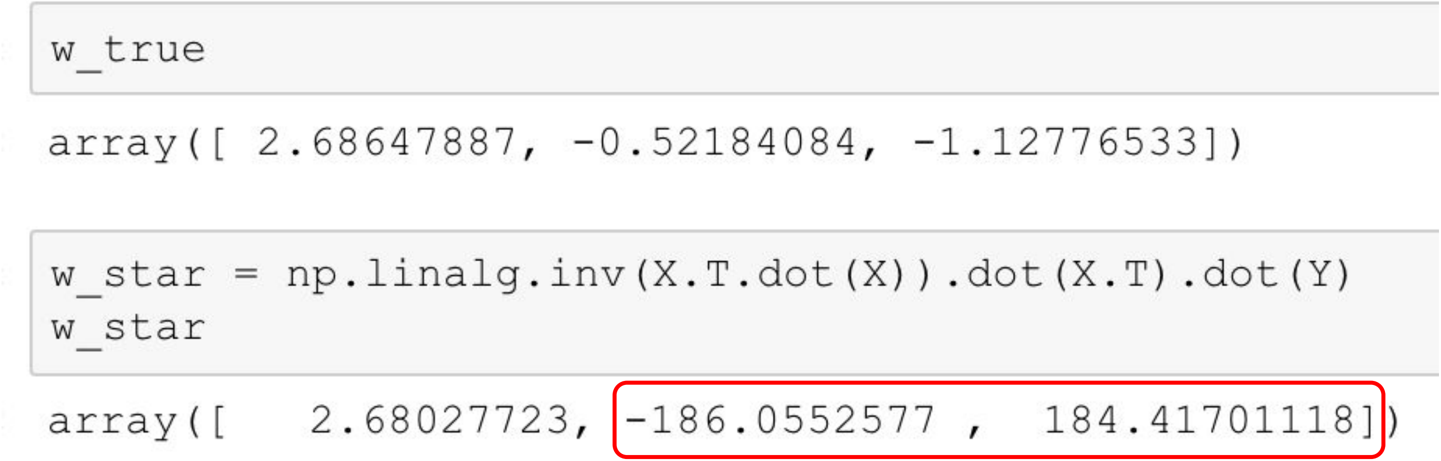
```
array([   2.68027723, -186.0552577 ,  184.41701118])
```

corresponding features are almost collinear

Unstable solution

In case of multicollinear features the matrix $X^T X$ is almost singular .

It leads to unstable solution:



```
w_true
```

```
array([ 2.68647887, -0.52184084, -1.12776533])
```

```
w_star = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(Y)
w_star
```

```
array([   2.68027723, -186.0552577 ,  184.41701118])
```

the coefficients are huge and sum up to almost 0

Regularization

To make the matrix nonsingular, we can add a diagonal matrix:

$$\hat{\mathbf{w}} = (X^T X + \lambda I)^{-1} X^T Y \,,$$

where $I = \mathrm{diag}[1_1, \ldots, 1_p]$.

Actually, it’s a solution for the case $Q(\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2 + \lambda^2 \|\mathbf{w}\|_2^2$.

Regularization

To make the matrix nonsingular, we can add a diagonal matrix:

$$\hat{\mathbf{w}} = (X^T X + \lambda I)^{-1} X^T Y \,,$$

where $I = \mathrm{diag}[1_1, \ldots, 1_p]$.

Actually, it’s a solution for the case $Q(\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2 + \lambda^2 \|\mathbf{w}\|_2^2$.

exercise: check it by yourself

Regularization

To make the matrix nonsingular, we can add a diagonal matrix:

$$\hat{\mathbf{w}} = (X^T X + \lambda I)^{-1} X^T Y \,,$$

where $I = \mathrm{diag}[1_1, \ldots, 1_p]$.

Actually, it’s a solution for the case $Q(\mathbf{w}) = \|Y - X\mathbf{w}\|_2^2 + \lambda^2 \|\mathbf{w}\|_2^2$.

Gauss-Markov theorem

Let $Y = X\mathbf{w} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots \varepsilon_n]$ is vector of error random variables.

Gauss-Markov theorem

Let $Y = X\mathbf{w} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots \varepsilon_n]$ is vector of error random variables.

The Gauss–Markov assumptions concern the set of error random variables:

Gauss-Markov theorem

Let $Y = X\mathbf{w} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots \varepsilon_n]$ is vector of error random variables.

The Gauss–Markov assumptions concern the set of error random variables:

- They have zero mean: $\mathbb{E}[\varepsilon_i] = 0 \; \forall i$

Gauss-Markov theorem

Let $Y = X\mathbf{w} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} = \left[\varepsilon_1, \ldots \varepsilon_n\right]$ is vector of error random variables.

The Gauss–Markov assumptions concern the set of error random variables:

- They have zero mean: $\mathbb{E}[\varepsilon_i] = 0 \; \forall i$
- They are homoscedastic, that is all have the same finite variance:

$$\mathrm{Var}(\varepsilon_i) = \sigma^2 < \inf \;\; \forall i$$

Gauss-Markov theorem

Let $Y = X\mathbf{w} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} = \left[\varepsilon_1, \ldots \varepsilon_n\right]$ is vector of error random variables.

The Gauss–Markov assumptions concern the set of error random variables:

- They have zero mean: $\mathbb{E}\left[\varepsilon_i\right] = 0 \; \forall i$
- They are homoscedastic, that is all have the same finite variance:

$$\mathrm{Var}(\varepsilon_i) = \sigma^2 < \inf \;\; \forall i$$

- Distinct error terms are uncorrelated:

$$\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0 \; \forall I \neq j.$$

Gauss-Markov theorem

Let $Y = X\mathbf{w} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} = \left[\varepsilon_1, \ldots \varepsilon_n\right]$ is vector of error random variables.

The Gauss–Markov assumptions concern the set of error random variables:

- They have zero mean: $\mathbb{E}[\varepsilon_i] = 0 \; \forall i$
- They are homoscedastic, that is all have the same finite variance:

$$\mathrm{Var}(\varepsilon_i) = \sigma^2 < \inf \;\; \forall i$$

- Distinct error terms are uncorrelated:

$$\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0 \; \forall I \neq j.$$

Then the solution $\hat{\mathbf{w}} = (X^T X)^{-1} X^T Y$ delivers BLEU: **B**est **L**inear **U**nbiased **E**stimator.

Different norms

Once more: loss functions:

- $MSE = \frac{1}{n}\|\mathbf{x}^T\mathbf{w} - \mathbf{y}\|_2^2$

Regularization terms:

- L_2: $\|\mathbf{w}\|_2^2$

Different norms

Once more: loss functions:

- $MSE = \frac{1}{n}\|\mathbf{x}^T\mathbf{w} - \mathbf{y}\|_2^2$

- $MAE = \frac{1}{n}\|\mathbf{x}^T\mathbf{w} - \mathbf{y}\|_1$

Regularization terms:

- L_2: $\|\mathbf{w}\|_2^2$

- L_1: $\|\mathbf{w}\|_1$

Different norms

Once more: loss functions:

- $\boxed{MSE = \frac{1}{n}\|\mathbf{x}^T\mathbf{w} - \mathbf{y}\|_2^2}$

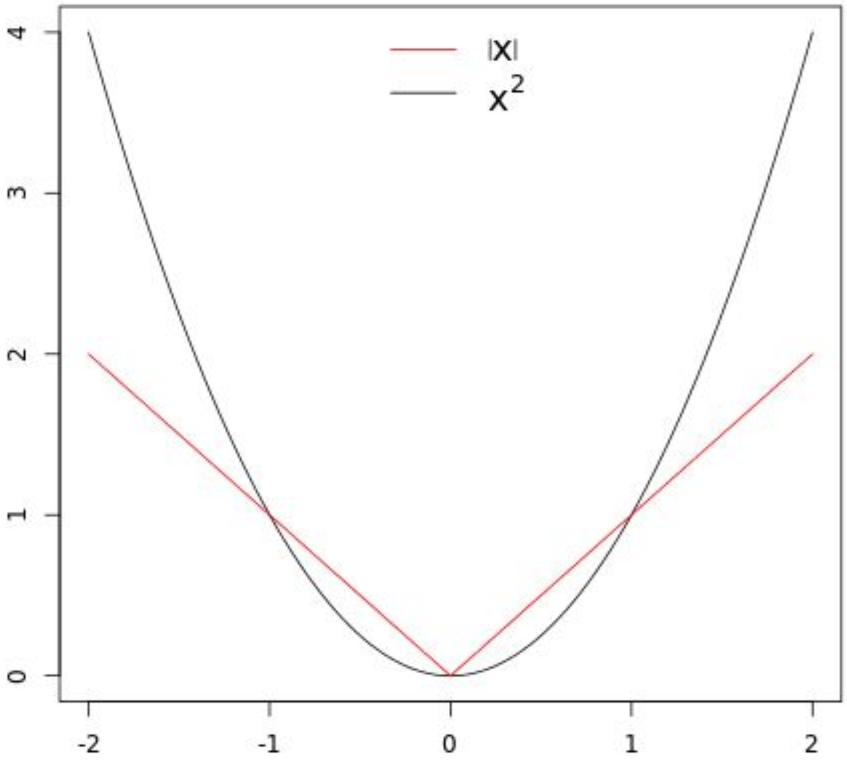
only works for Gauss-Markov theorem

- $MAE = \frac{1}{n}\|\mathbf{x}^T\mathbf{w} - \mathbf{y}\|_1$

Regularization terms:

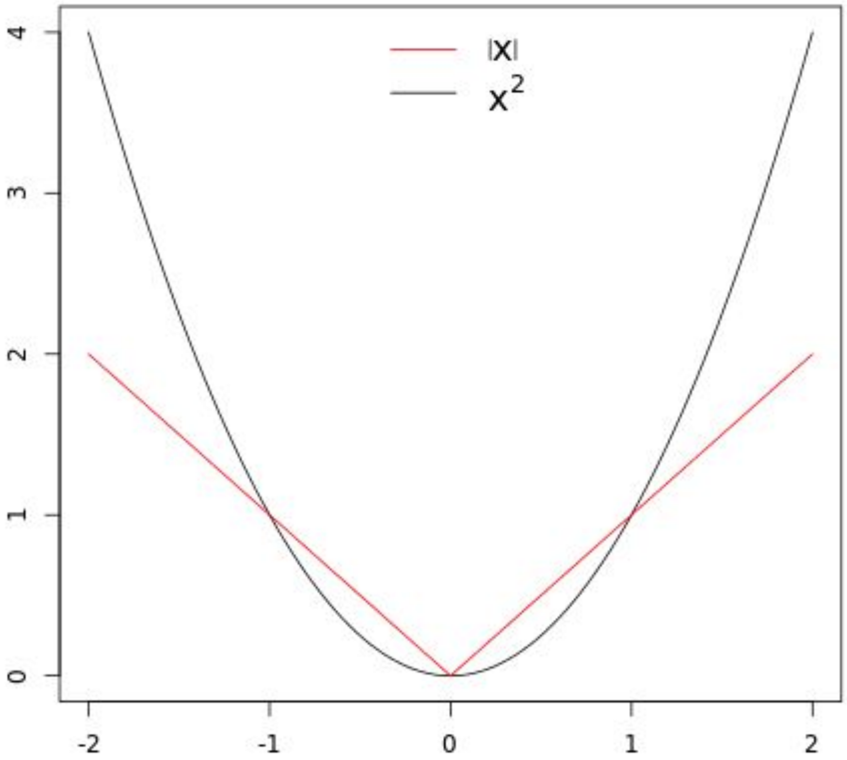
- $L_2 \colon \|\mathbf{w}\|_2^2$
- $L_1 \colon \|\mathbf{w}\|_1$

What’s the difference?



What’s the difference?

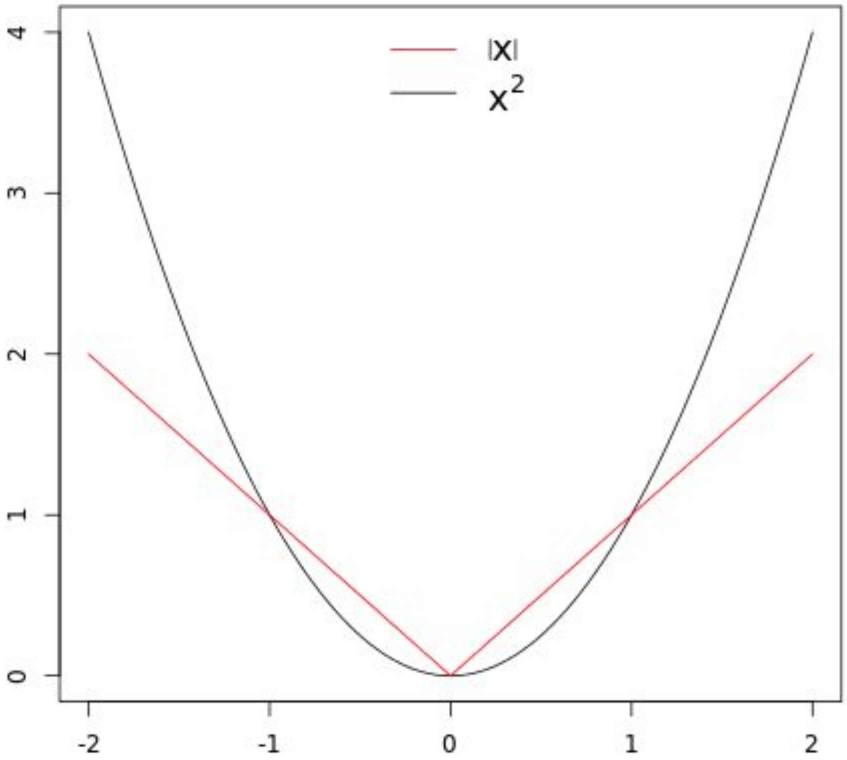
- MSE
  - delivers BLUE according to Gauss-Markov theorem
  - differentiable
  - sensitive to noise



What’s the difference?

- MSE
  - delivers BLUE according to Gauss-Markov theorem
  - differentiable
  - sensitive to noise

- MAE
  - non-differentiable
    - (actually, it is differentiable)
  - much more prone to noise

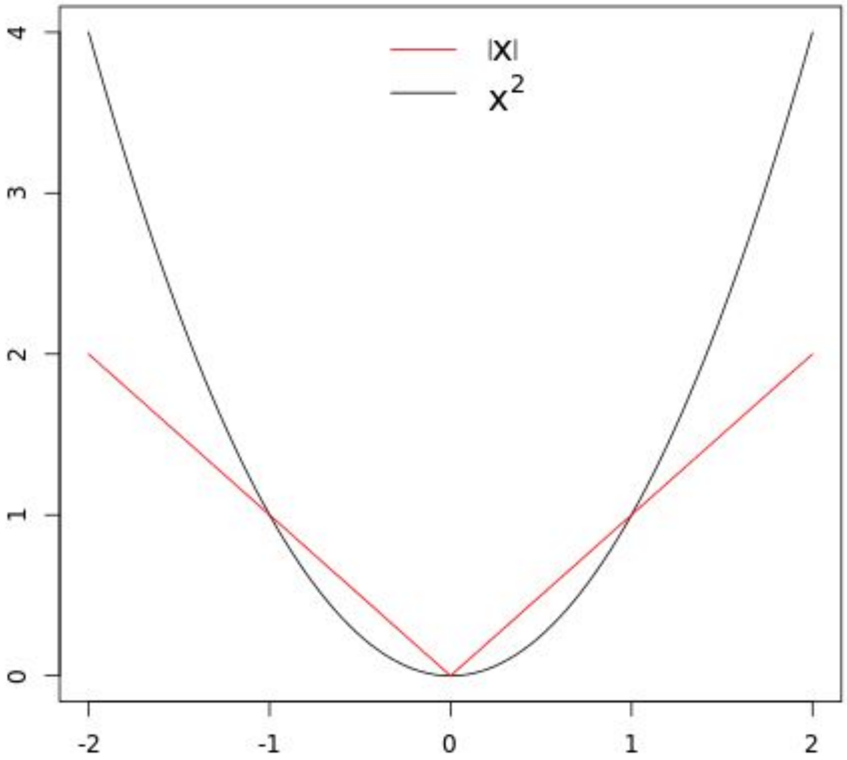


What’s the difference?

- MSE
  - delivers BLUE according to Gauss-Markov theorem
  - differentiable
  - sensitive to noise

- MAE
  - non-differentiable
    - (actually, it is differentiable)
  - much more prone to noise

- L_2 regularization
  - constraints weights
  - delivers more stable solution
  - Differentiable



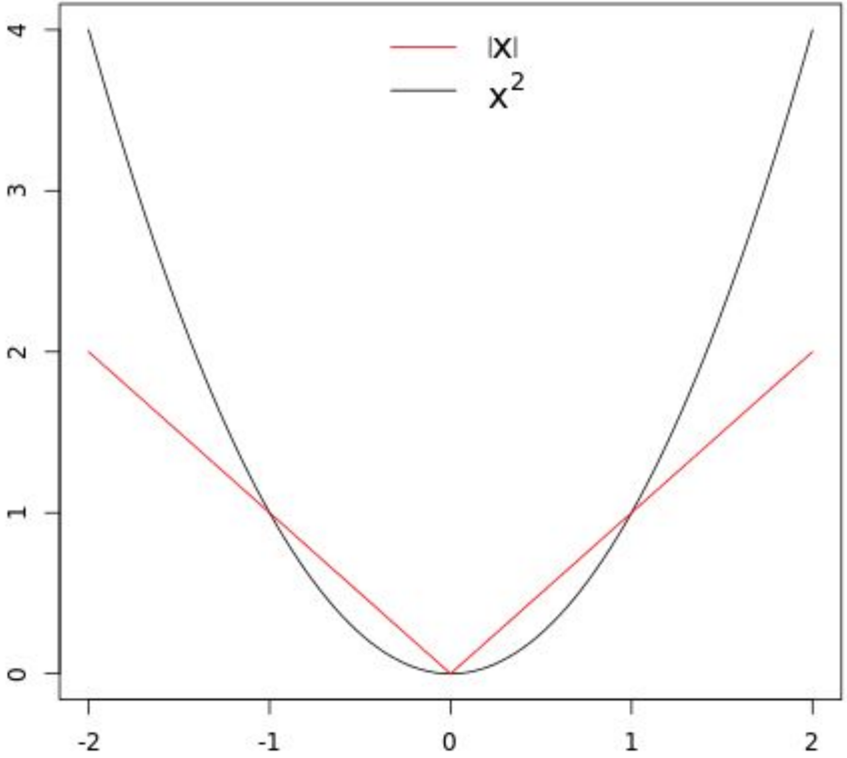
What’s the difference?

- MSE
  - delivers BLUE according to Gauss-Markov theorem
  - differentiable
  - sensitive to noise

- MAE
  - non-differentiable
    - (actually, it is differentiable)
  - much more prone to noise

- L_2 regularization
  - constraints weights
  - delivers more stable solution
  - Differentiable

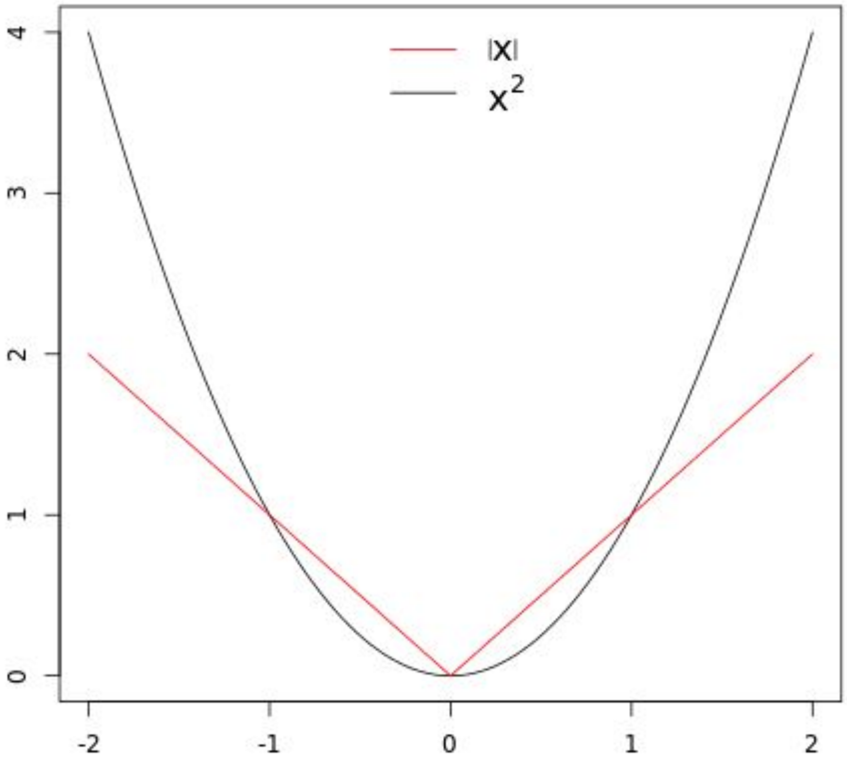
- L_1 regularization
  - non-differentiable
    - (actually, the same as MAE ;)
  - selects features



What’s the difference?

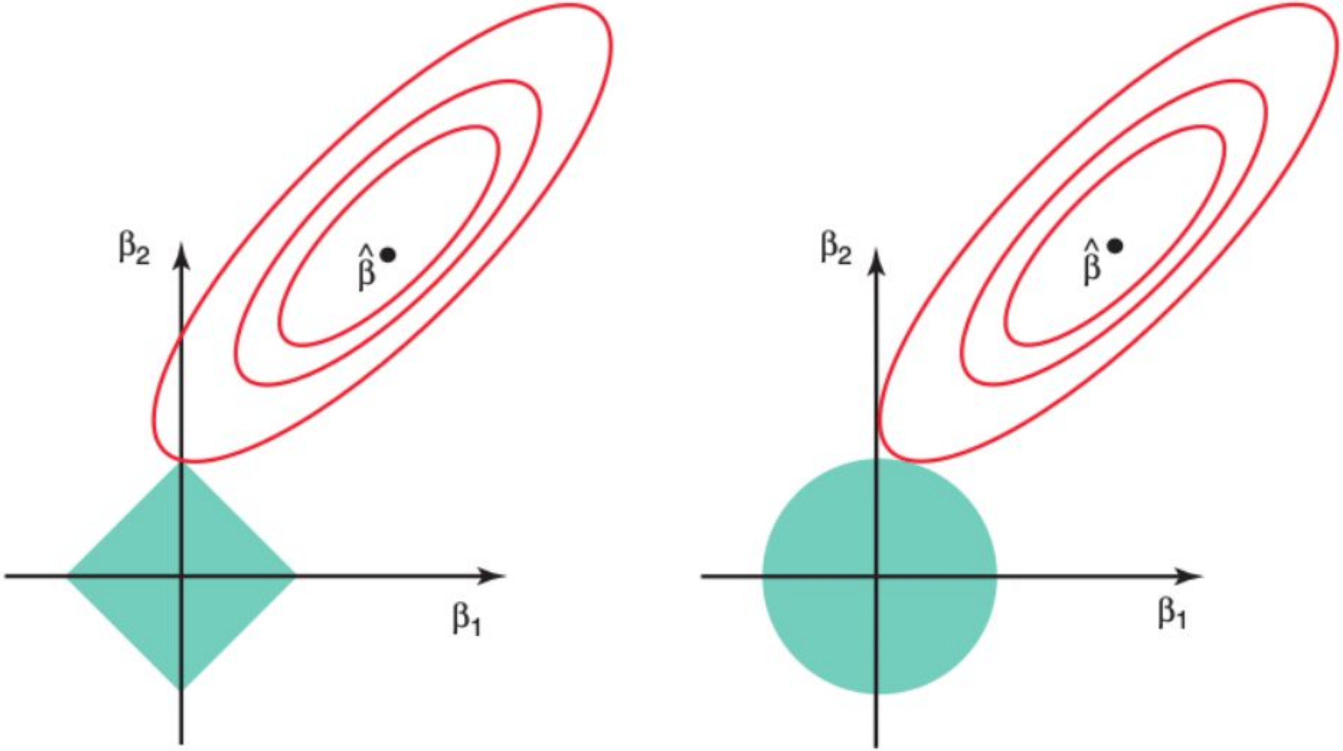
- MSE
    - delivers BLUE according to Gauss-Markov theorem
    - differentiable
    - sensitive to noise
- MAE
    - non-differentiable
        - (actually, it is differentiable)
    - much more prone to noise
- L_2 regularization
    - constraints weights
    - delivers more stable solution
    - Differentiable
- L_1 regularization
    - non-differentiable
        - (actually, the same as MAE ;)
    - selects features

Does what?



Regularization: illustration

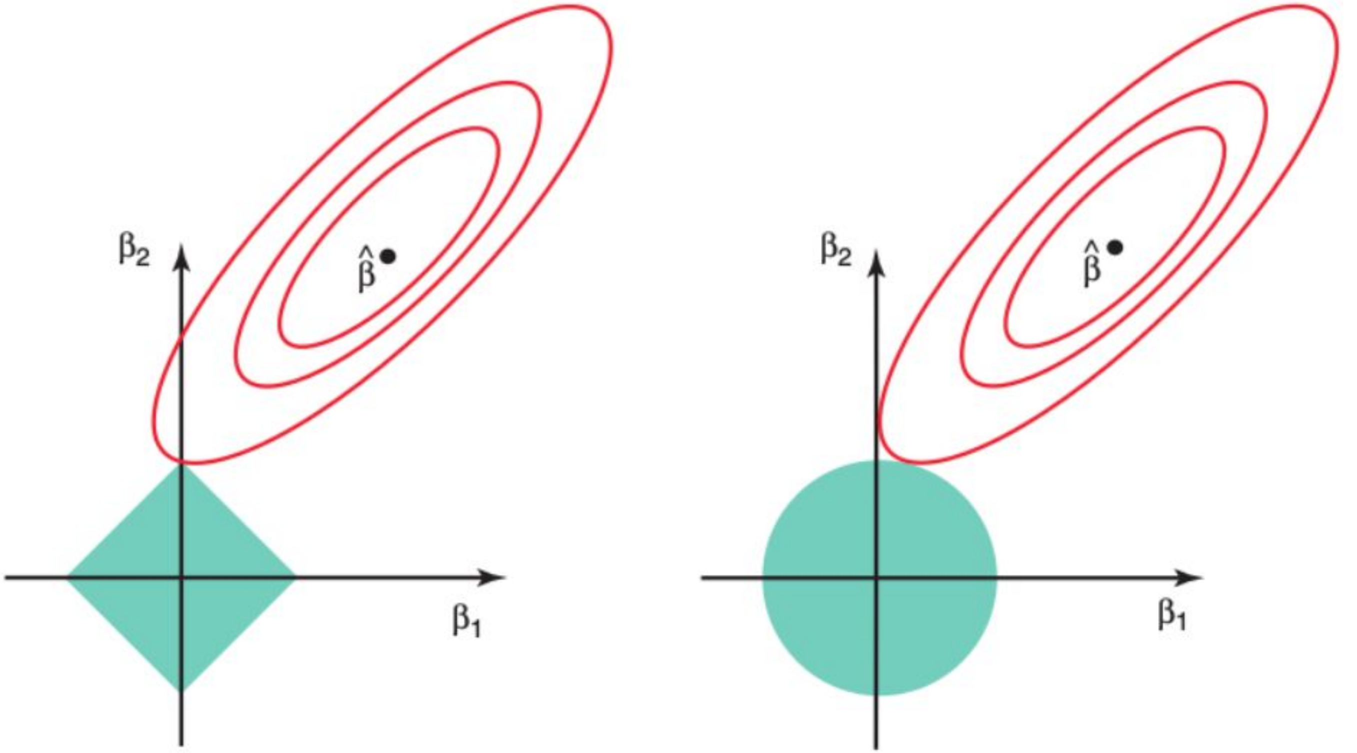
$$\mathbf{w} = [\beta_1, \beta_2]$$



source: Modern Multivariate Statistical Techniques

Regularization: illustration

$$\mathbf{w} = [\beta_1, \beta_2]$$



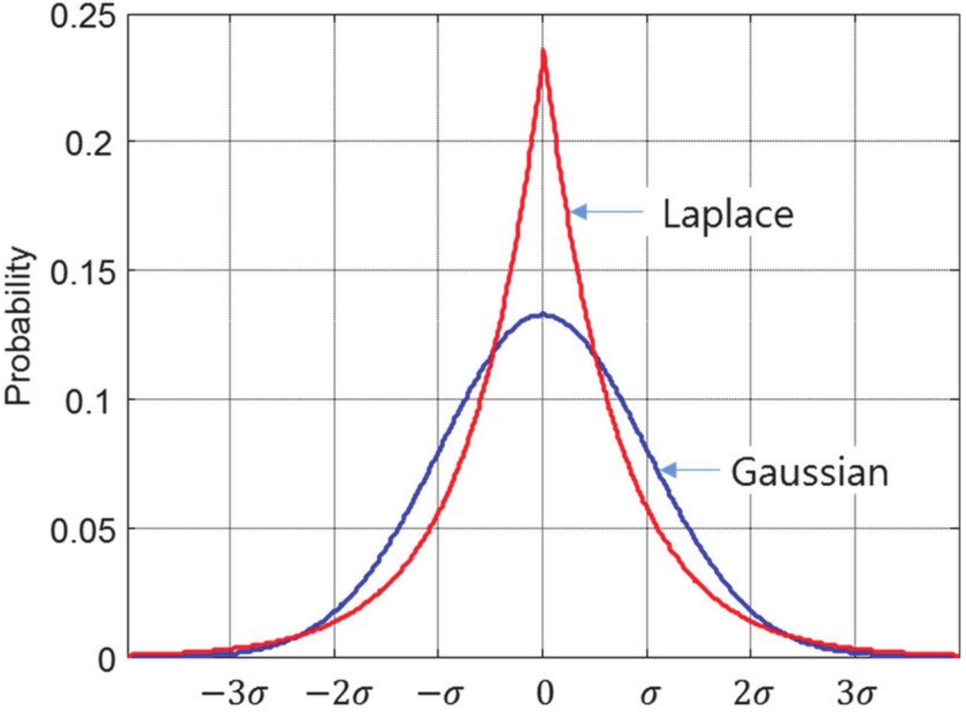
L_1 regularization

L_2 regularization

source: Modern Multivariate Statistical Techniques

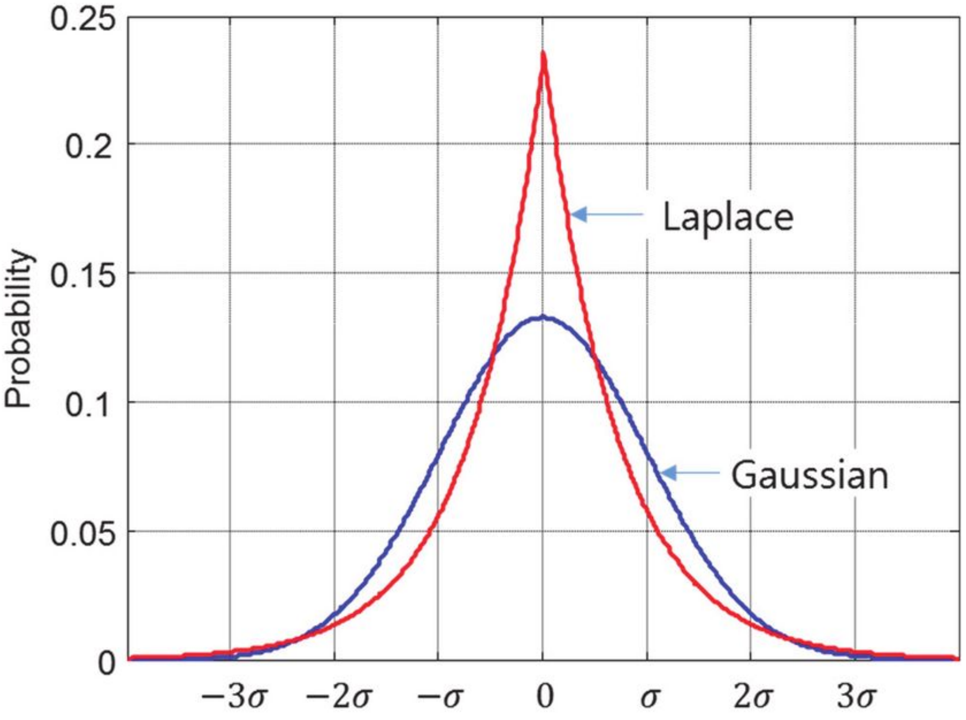
Regularization: probability interpretation

assume **w** elements are sampled from some *specific* distribution (prior distribution for the weights vector)



Regularization: probability interpretation

assume **w** elements are sampled from some *specific* distribution (prior distribution for the weights vector)

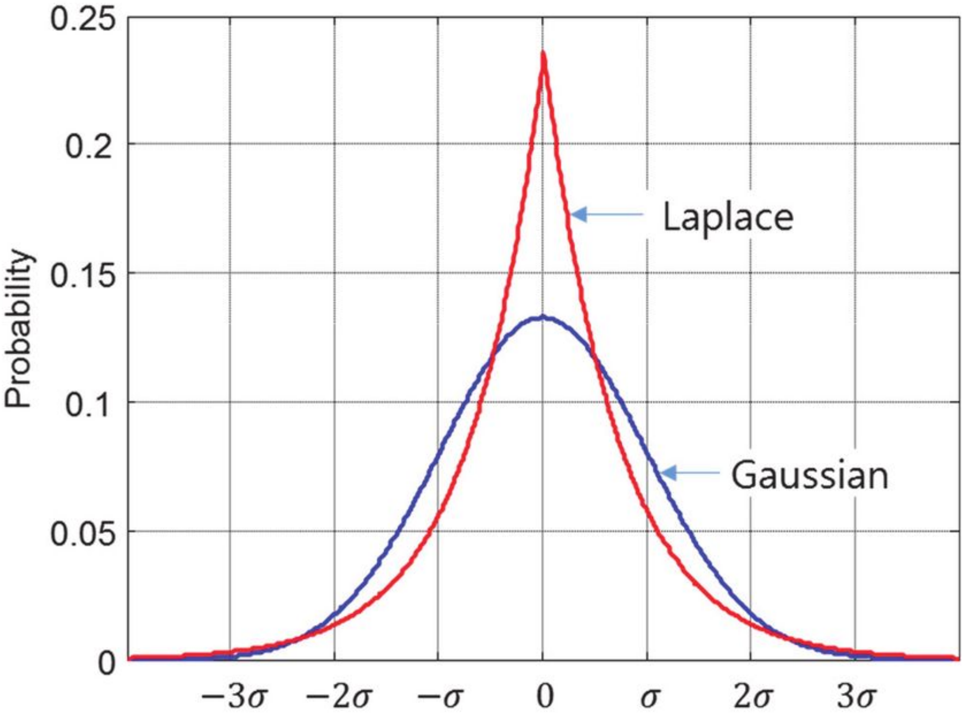


$= L_1$ regularization

$= L_2$ regularization

Regularization: probability interpretation

assume **w** elements are sampled from some *specific* distribution (prior distribution for the weights vector)



$= L_1$ regularization

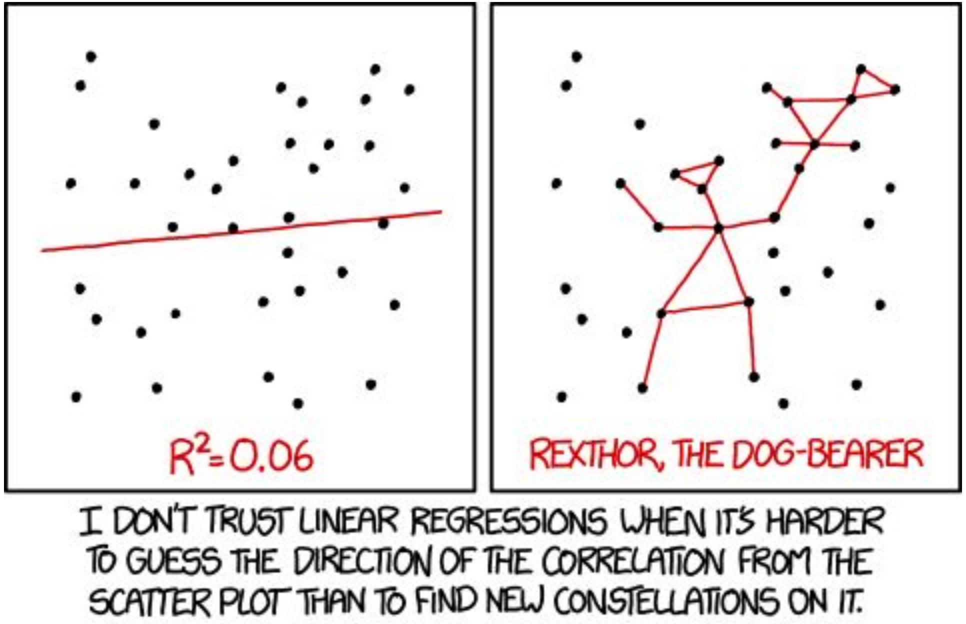
$= L_2$ regularization

see seminar extra materials for more

Welcome to the church of Bayes

Дмитрий Ветров - заведующий Международной лабораторией глубинного обучения и байесовских методов

That’s all. Practice coming next.


R^2=0.06
REXTHOR, THE DOG-BEARER
I DON'T TRUST LINEAR REGRESSIONS WHEN IT'S HARDER TO GUESS THE DIRECTION OF THE CORRELATION FROM THE SCATTER PLOT THAN TO FIND NEW CONSTELLATIONS ON IT.