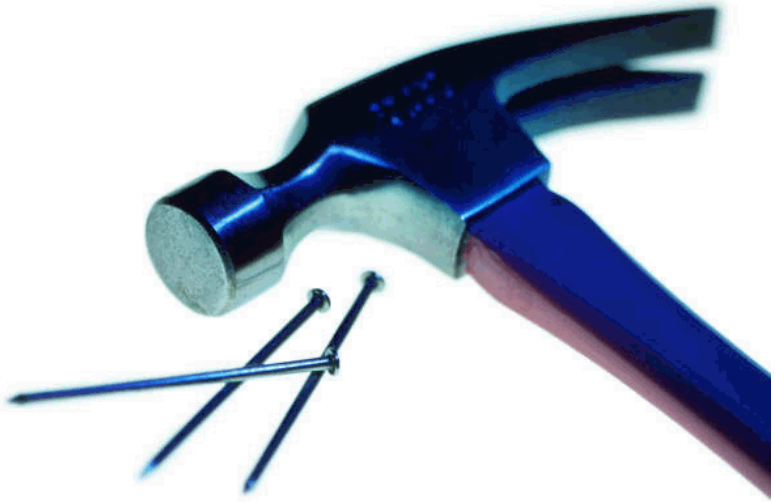


Machine Learning: The Optimization Perspective

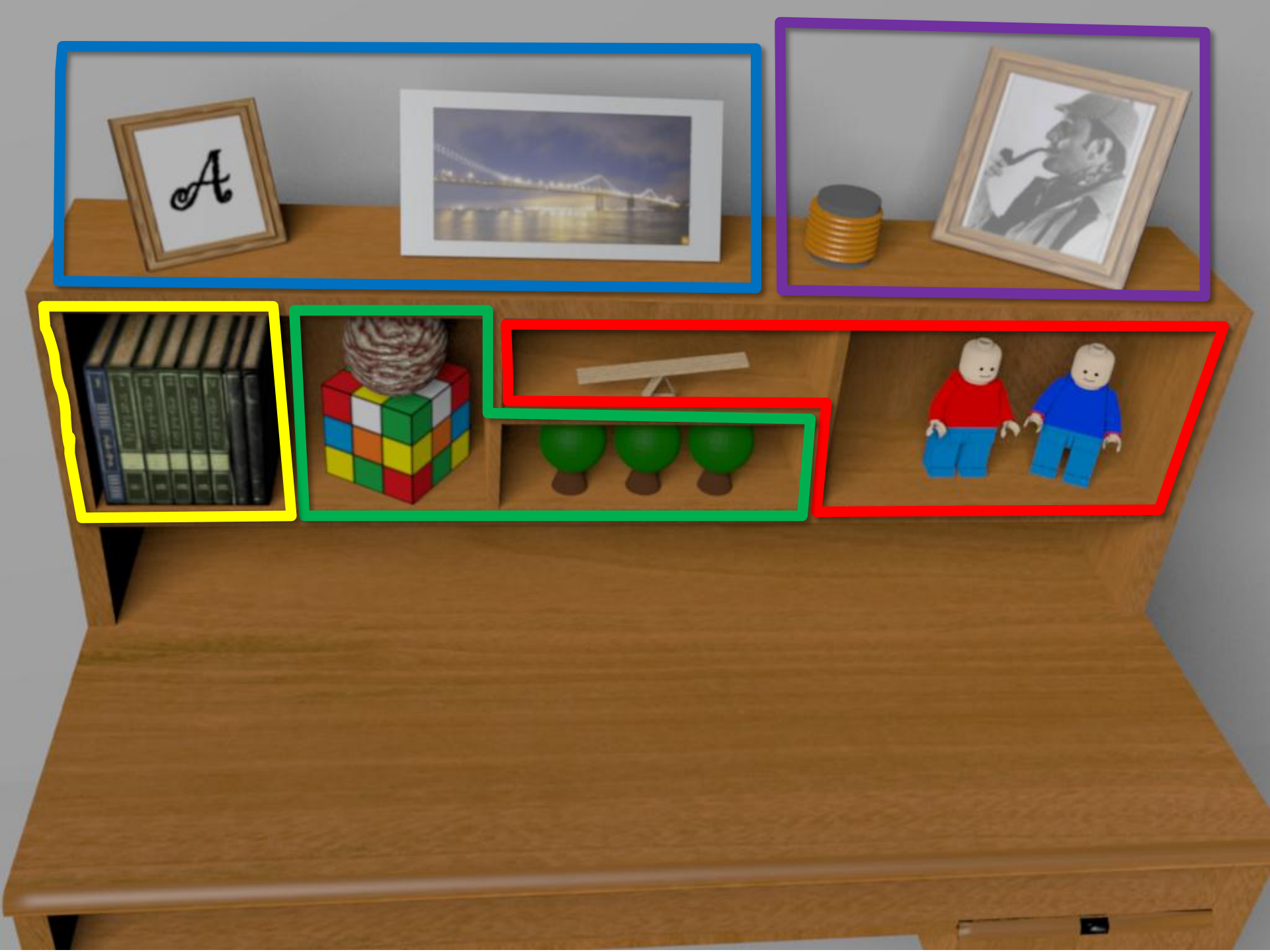
Konstantin Tretyakov
http://kt.era.ee

AACIMP 2012
August 2012

STACC Software Technology and Applications Competence Center

BIT

TARTU ÜLIKOOL · UNIVERSITAS TARTUENSIS · 1632



So far…

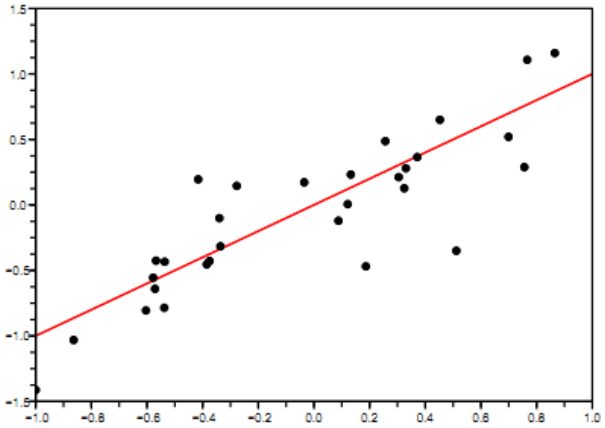
- Machine learning is important and interesting
- The general concept:

Fitting models to data

So far…

- Machine learning is important and interesting
- The general concept:

Searching for the best model fitting to data

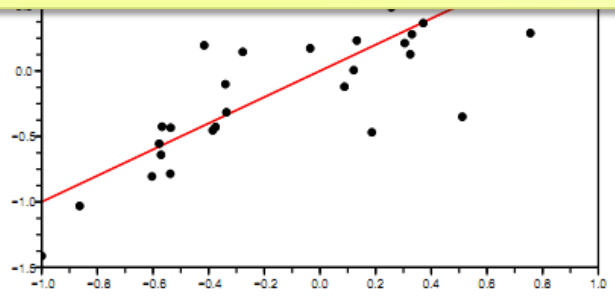


So far…

- Machine learning is important and interesting
- The g

Optimization

Probability Theory



Today

1. Optimization is **important**
2. Optimization is **possible**

Today

1. Optimization is **important**
2. Optimization is **possible***

* Basic **techniques**

- Constrained / Unconstrained
- Analytic / Iterative
- Continuous / Discrete

Special cases of optimization

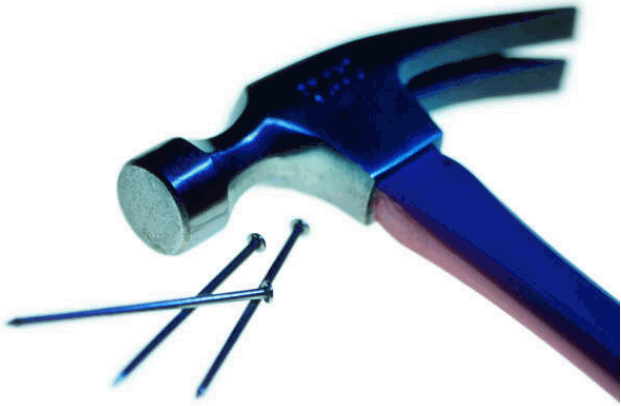
- Machine learning
- …

Special cases of optimization

- Machine learning
- Algorithms and data structures
- General problem-solving
- Management and decision-making

Special cases of optimization

- Machine learning
- Algorithms and data structures
- General problem-solving
- Management and decision-making
- Evolution
- *The Meaning of Life?*



Optimization task

Given **a function**

$$f(\mathbf{x}) : \mathbf{x} \rightarrow \mathbb{R}$$

find the argument x resulting in the **optimal** value.

Constrained optimization task

Given **a function**

$$f(\mathbf{x}) : \mathbf{x} \rightarrow \mathbb{R}$$

find the argument x resulting in the **optimal** value, subject to

$$\mathbf{x} \in \mathcal{C}$$

Optimization methods

In principle, **x** can be anything:

- **Discrete**
  - Value (e.g. a name)
  - Structure (e.g. a graph, plaintext)
  - Finite / infinite
- **Continuous***
  - Real-number, vector, matrix, …
  - Complex-number, function, …

Optimization methods

In principle, **f** can be anything:

- **Random oracle**
- **Structured**
- **Continuous**
- **Differentiable**
- **Convex**
- **…**

Optimization methods

		Knowledge about f	
		Not much	A lot
Type of X	Discrete		
	Continuous		

Optimization methods

		Knowledge about f	
		Not much	A lot
Type of X	Discrete	**Combinatorial search:** Brute-force, Stepwise, MCMC, Population-based, …	Algorithmic
	Continuous	**Numeric methods:** Gradient-based, Newton-like, MCMC, Population-based, …	Analytic

Optimization methods

Finding a **weight-vector** w, that minimizes the model error

Type of X	Knowledge about f: Not much	Knowledge about f: A lot
	Combinatorial search: Brute-force, Stepwise, MCMC, Population-based, …	**Algorithmic**
Continuous	**Numeric methods:** Gradient-based, Newton-like, MCMC, Population-based, …	**Analytic**

Optimization methods

Finding a **weight-vector** w, that minimizes the model error, in a **fairly general case**

		Knowledge about **f**	
		Not much	**A lot**
	Continuous	**Combinatorial search:** Brute-force, Stepwise, MCMC, Population-based, …	**Algorithmic**
		Numeric methods: Gradient-based, Newton-like, MCMC, Population-based, …	**Analytic**

Optimization methods

Finding a **weight-vector** w, that minimizes the model error, in a **very general case**

		Knowledge about **f**	
		Not much	**A lot**
	Continuous	**Combinatorial search:** Brute-force, Stepwise, MCMC, Population-based, …	**Algorithmic**
		Numeric methods: Gradient-based, Newton-like, MCMC, Population-based, …	**Analytic**

Optimization methods

Finding a **weight-vector** w, that minimizes the model error, in **many practical cases**

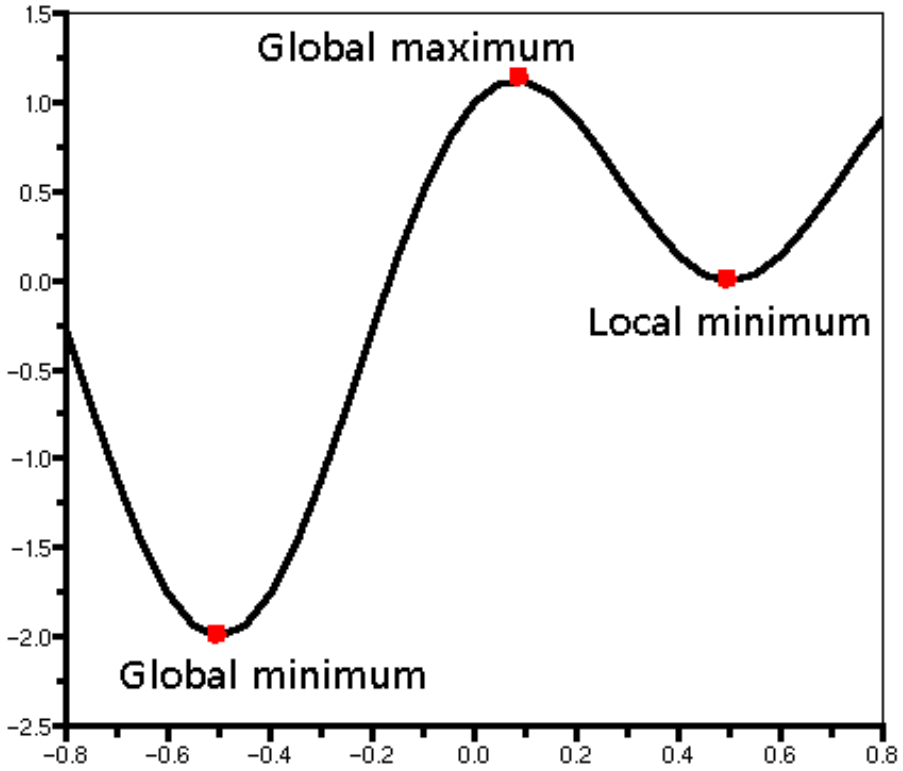
		Knowledge about f	
		Not much	**A lot**
		Combinatorial search: Brute-force, Stepwise, MCMC, Population-based, …	**Algorithmic**
	Continuous	**Numeric methods:** Gradient-based, Newton-like, MCMC, Population-based, …	**Analytic**

Optimization methods

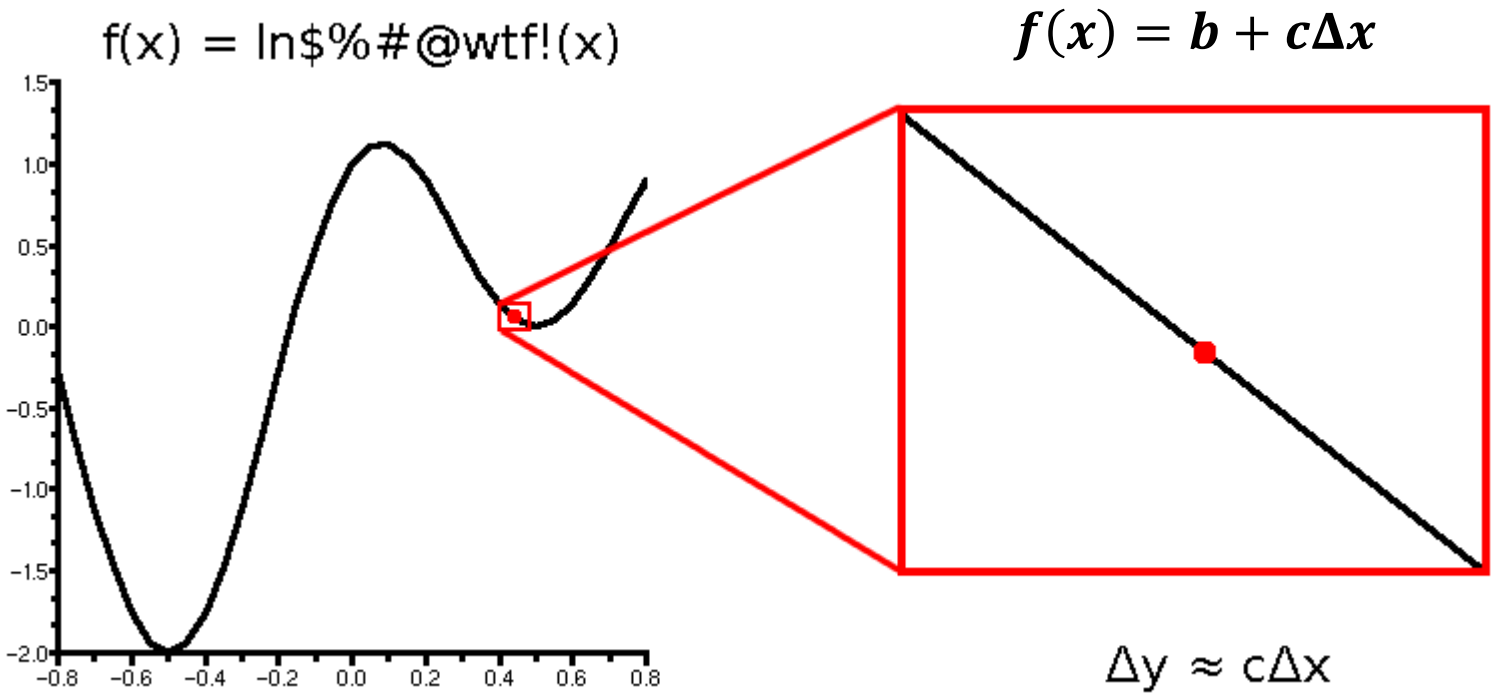
This lecture

		Knowledge about f	
		Not much	A lot
		Combinatorial search: Brute-force, Stepwise, MCMC, Population-based, …	Algorithmic
	Continuous	Numeric methods: Gradient-based, Newton-like, MCMC, Population-based, …	Analytic

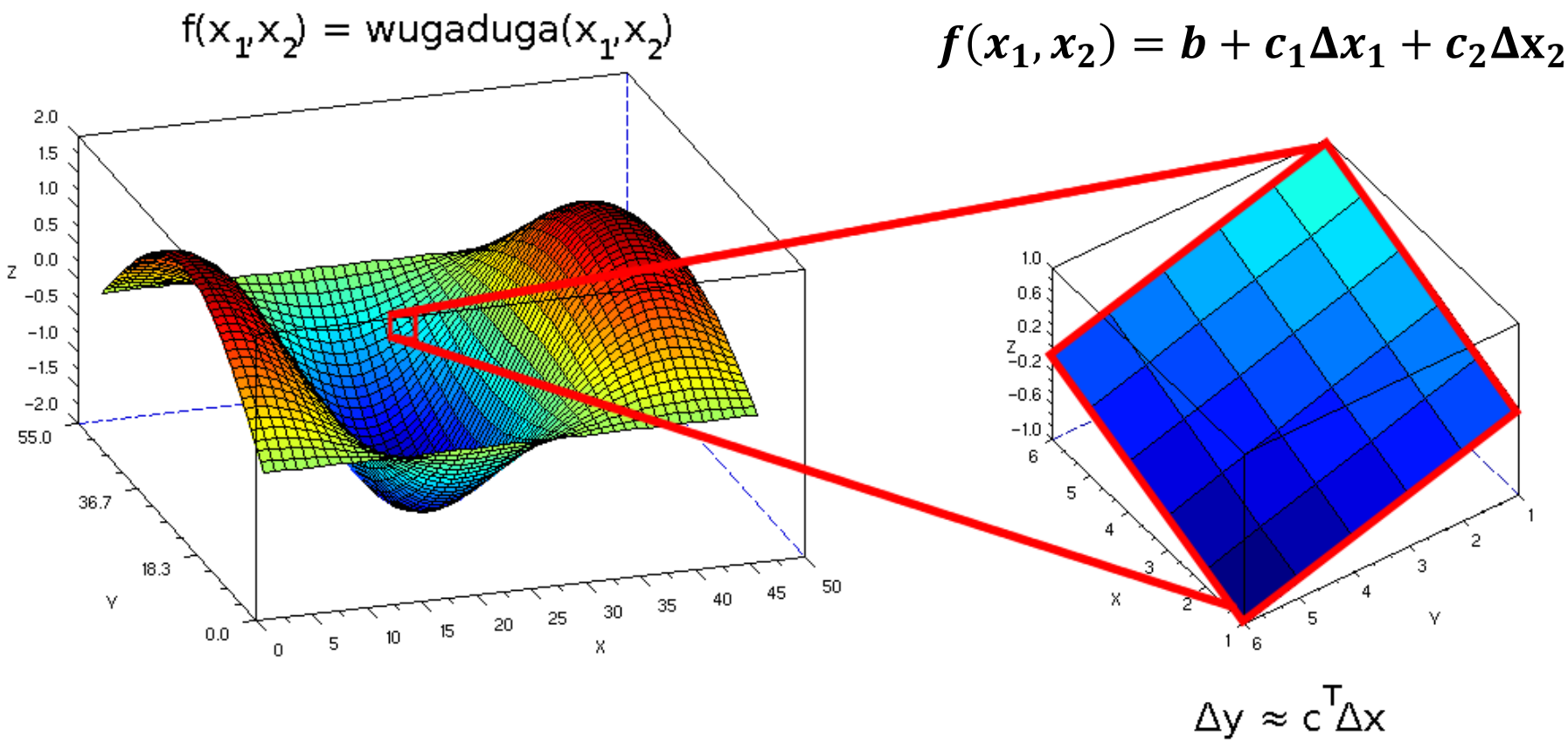
Minima and maxima



Differentiability



Differentiability



Differentiability

Definition. We call a function $f : \mathbb{R}^m \to \mathbb{R}$ differentiable at point $\mathbf{x}_0$ if there exists $\mathbf{c}(\mathbf{x}_0) \in \mathbb{R}^m$ such that:

$$\Delta f(\mathbf{x}_0) = f(\mathbf{x}_0 + \Delta\mathbf{x}) - f(\mathbf{x}_0) = \mathbf{c}(\mathbf{x}_0)^T \Delta\mathbf{x} + o(\Delta\mathbf{x})$$

We call $\mathbf{c}(\mathbf{x}_0)$ the gradient or derivative* of f (at point $\mathbf{x}_0$) and denote it by:

$$\frac{\partial f(\mathbf{x}_0)}{\partial \mathbf{x}} \quad \text{or} \quad f'(\mathbf{x}_0) \quad \text{or} \quad \nabla f(\mathbf{x}_0)$$

The Most Important Observation

Let f be differentiable and let $\nabla f(\mathbf{x}_0) = \mathbf{c} \neq \mathbf{0}$.
Take $\mathbf{\Delta x} = \ldots$. Then:

$$f(\mathbf{x}_0 + \mathbf{\Delta x}) \approx f(\mathbf{x}_0) + \mathbf{c}^T \ldots < f(\mathbf{x}_0).$$

therefore $\mathbf{x}_0$ can't be a minimum of f.

The Most Important Observation

Let f be differentiable and let $\nabla f(\mathbf{x}_0) = \mathbf{c} \neq \mathbf{0}$.
Take $\mathbf{\Delta x} = -\mu\mathbf{c}$. Then:

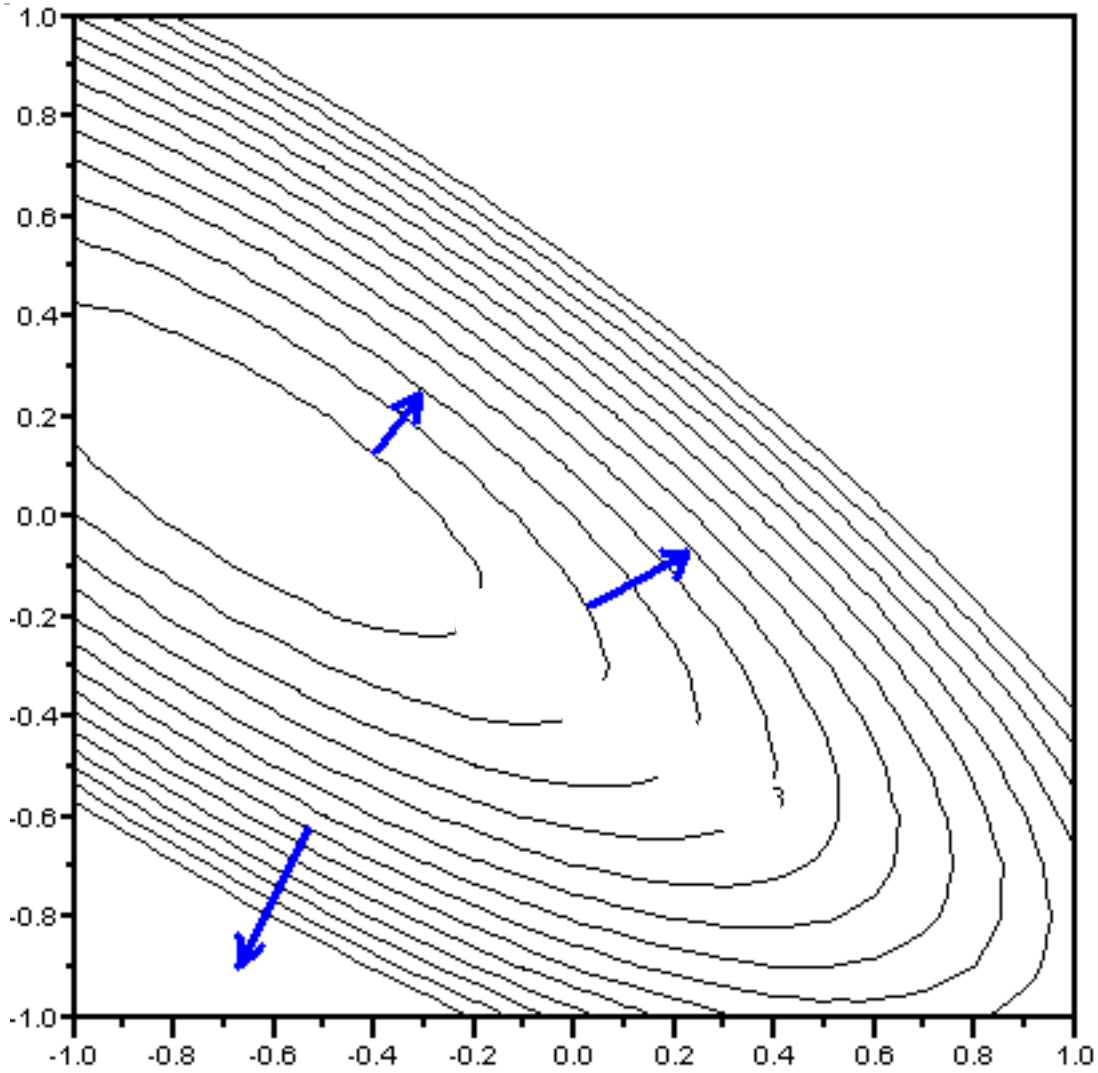
$$f(\mathbf{x}_0 + \mathbf{\Delta x}) \approx f(\mathbf{x}_0) + \mathbf{c}^T(-\mu\mathbf{c}) < f(\mathbf{x}_0).$$

therefore $\mathbf{x}_0$ can't be a minimum of f.

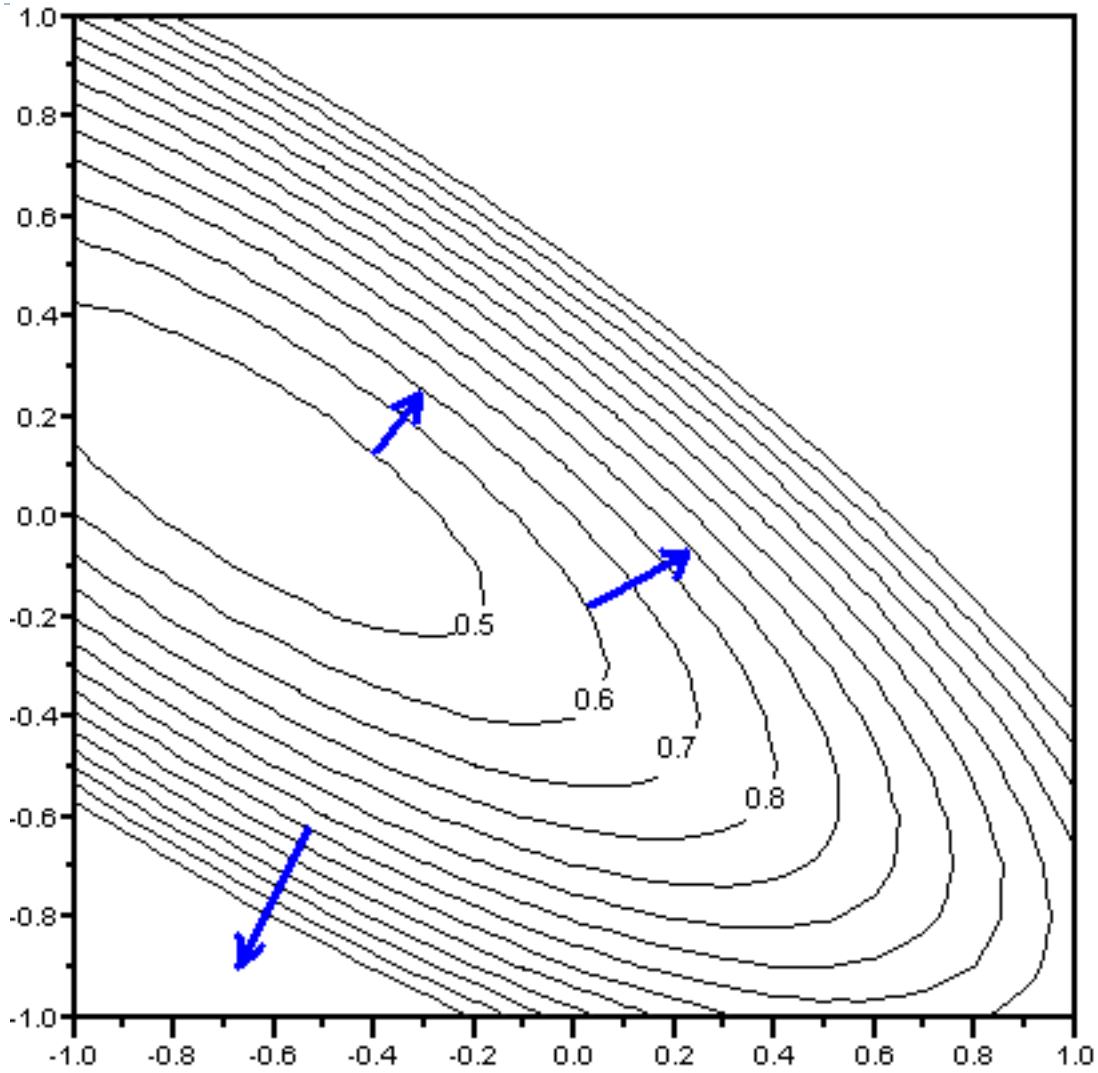
The Most Important Observation

- This small observation gives us everything we need for now
  - A nice **interpretation of the gradient**
  - An **extremality criterion**
  - An **iterative algorithm** for function minimization

Interpretation of the gradient



Interpretation of the gradient



Extremality criterion

Theorem (Fermat). Let f be differentiable. Then

$$\mathbf{x}_0 \text{ is an extremum } \Rightarrow \nabla f(\mathbf{x}_0) = \mathbf{0}.$$

The converse does not hold in general.

Iterative algorithm

1. Pick random point $\boldsymbol{x}_0$
2. If $\nabla f(\boldsymbol{x}_0) = \mathbf{0}$, then we've found an extremum.
3. Otherwise,

Iterative algorithm

1. Pick random point $\boldsymbol{x}_0$
2. If $\nabla f(\boldsymbol{x}_0) = \mathbf{0}$, then we've found an extremum.
3. Otherwise, make a small step downhill:

$$\boldsymbol{x}_1 \leftarrow \boldsymbol{x}_0 - \mu_0 \nabla f(\boldsymbol{x}_0)$$

Iterative algorithm

1. Pick random point $\boldsymbol{x}_0$
2. If $\nabla f(\boldsymbol{x}_0) = \mathbf{0}$, then we've found an extremum.
3. Otherwise, make a small step downhill:
$$\boldsymbol{x}_1 \leftarrow \boldsymbol{x}_0 - \mu_0 \nabla f(\boldsymbol{x}_0)$$
4. … and then another step
$$\boldsymbol{x}_2 \leftarrow \boldsymbol{x}_1 - \mu_1 \nabla f(\boldsymbol{x}_1)$$
5. … and so on until

Gradient descent

1. Pick random point $\boldsymbol{x}_0$
2. If $\nabla f(\boldsymbol{x}_0) = \mathbf{0}$, then we've found an extremum.
3. Otherwise, make a small step downhill:
$$\boldsymbol{x}_1 \leftarrow \boldsymbol{x}_0 - \mu_0 \nabla f(\boldsymbol{x}_0)$$
4. … and then another step
$$\boldsymbol{x}_2 \leftarrow \boldsymbol{x}_1 - \mu_1 \nabla f(\boldsymbol{x}_1)$$
5. … and so on until $\nabla f(\boldsymbol{x}_n) \approx \mathbf{0}$ or we're tired.

With a smart choice of μ_i we'll converge to a minimum

Gradient descent

1.
2.
3.

$$\boldsymbol{x}_1 \leftarrow \boldsymbol{x}_0 - \mu_0 \nabla f(\boldsymbol{x}_0)$$

4.

$$\boldsymbol{x}_2 \leftarrow \boldsymbol{x}_1 - \mu_1 \nabla f(\boldsymbol{x}_1)$$

Gradient descent

$$\boldsymbol{x}_{i+1} \leftarrow \boldsymbol{x}_i - \mu_i \nabla f(\boldsymbol{x}_i)$$

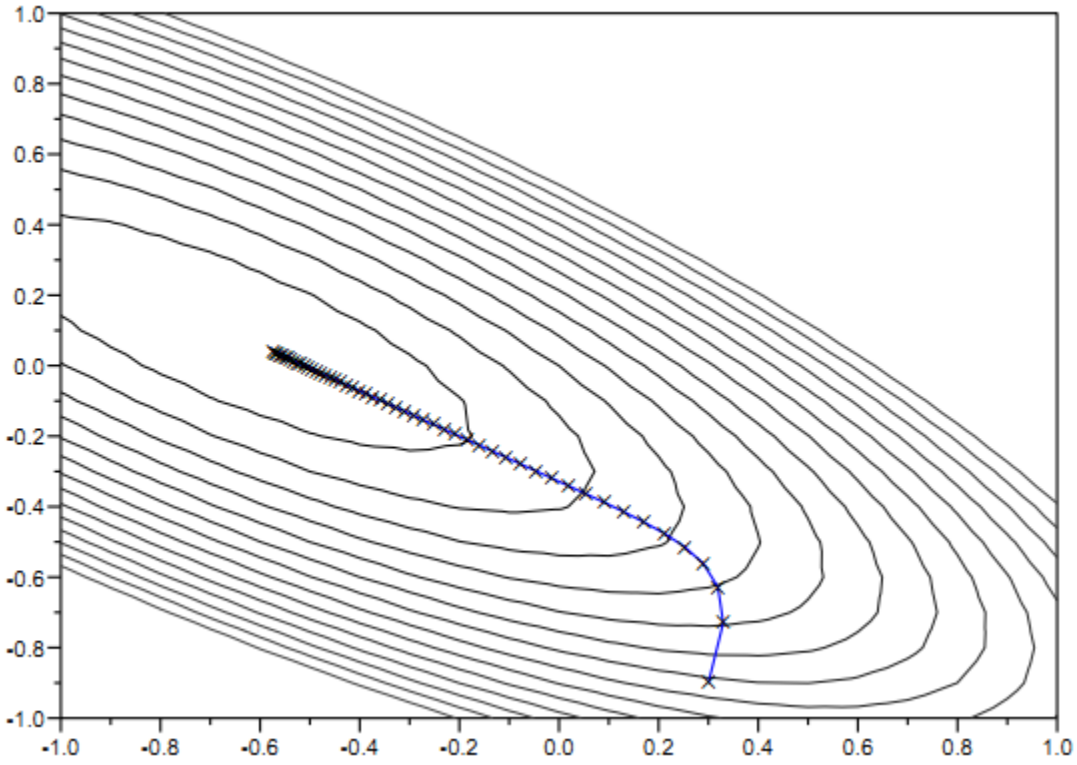
Gradient descent

$$\Delta \boldsymbol{x}_i = -\mu_i \nabla f(\boldsymbol{x}_i)$$

Gradient descent

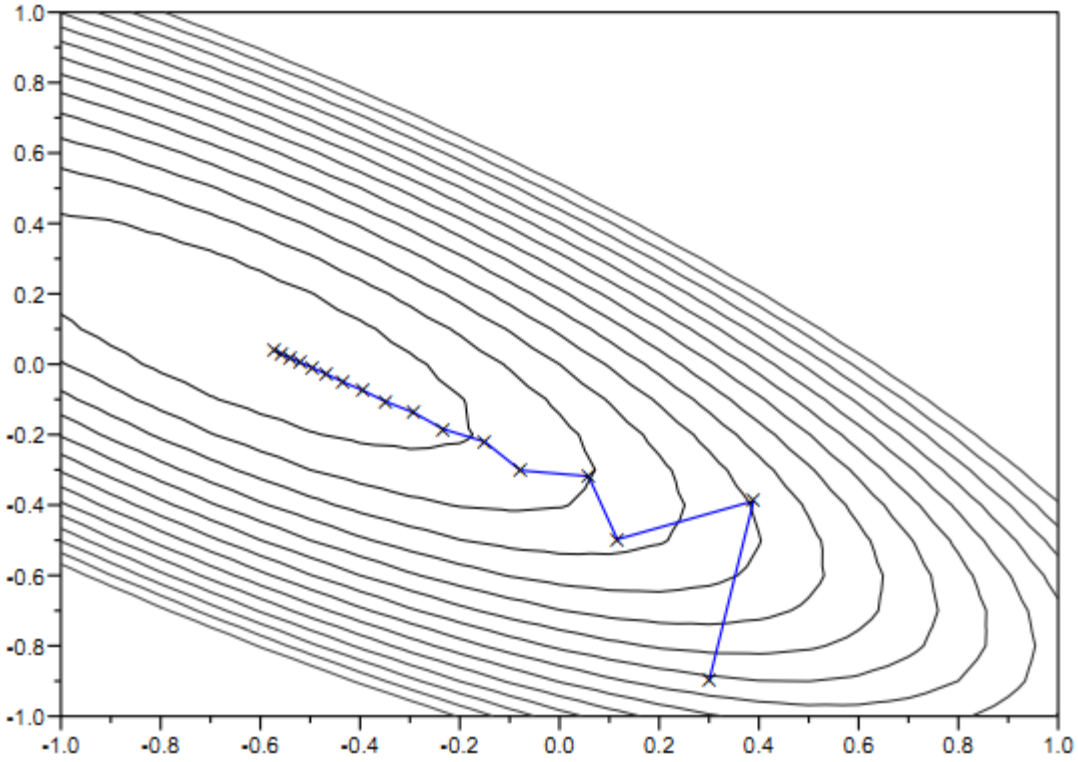
$$\Delta \boldsymbol{x}_i = -\mu \boldsymbol{c}$$

Gradient descent (fixed step)



$$\Delta \boldsymbol{x}_i = -\mu \nabla f(\boldsymbol{x}_i)$$

Gradient descent (fixed step)



$$\Delta \boldsymbol{x}_i = -\mu \nabla f(\boldsymbol{x}_i)$$

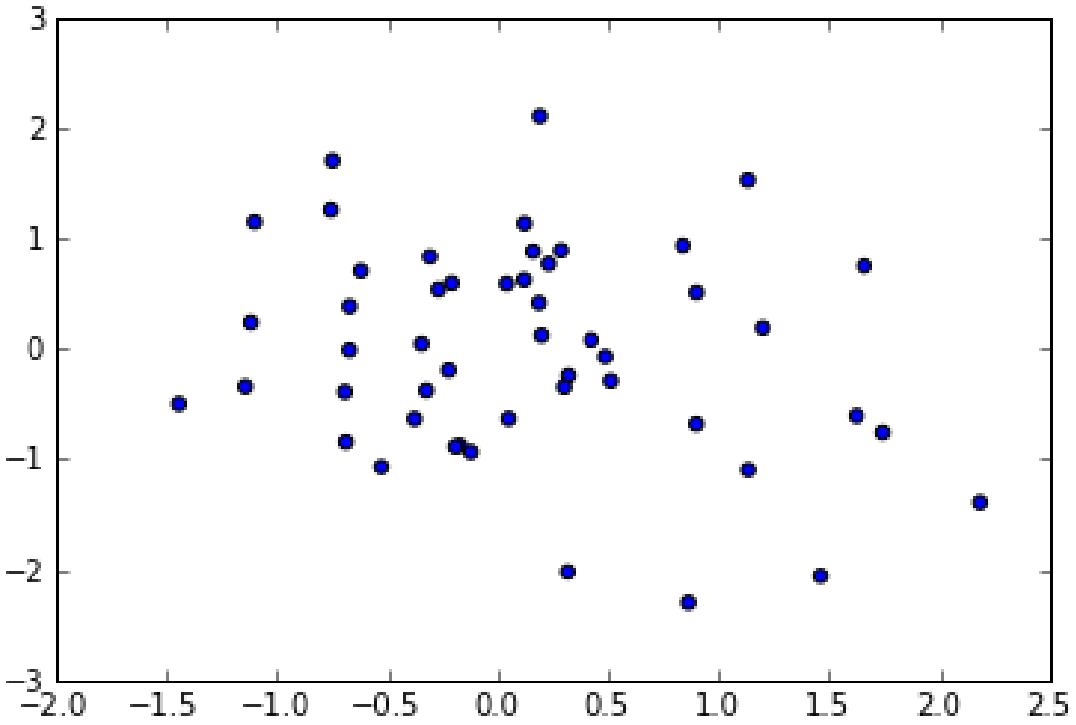
Example

Problem. Given a dataset $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbf{R}^m$, find $\mathbf{w}$, that minimizes

$$f(\mathbf{w}) = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{w}\|^2$$

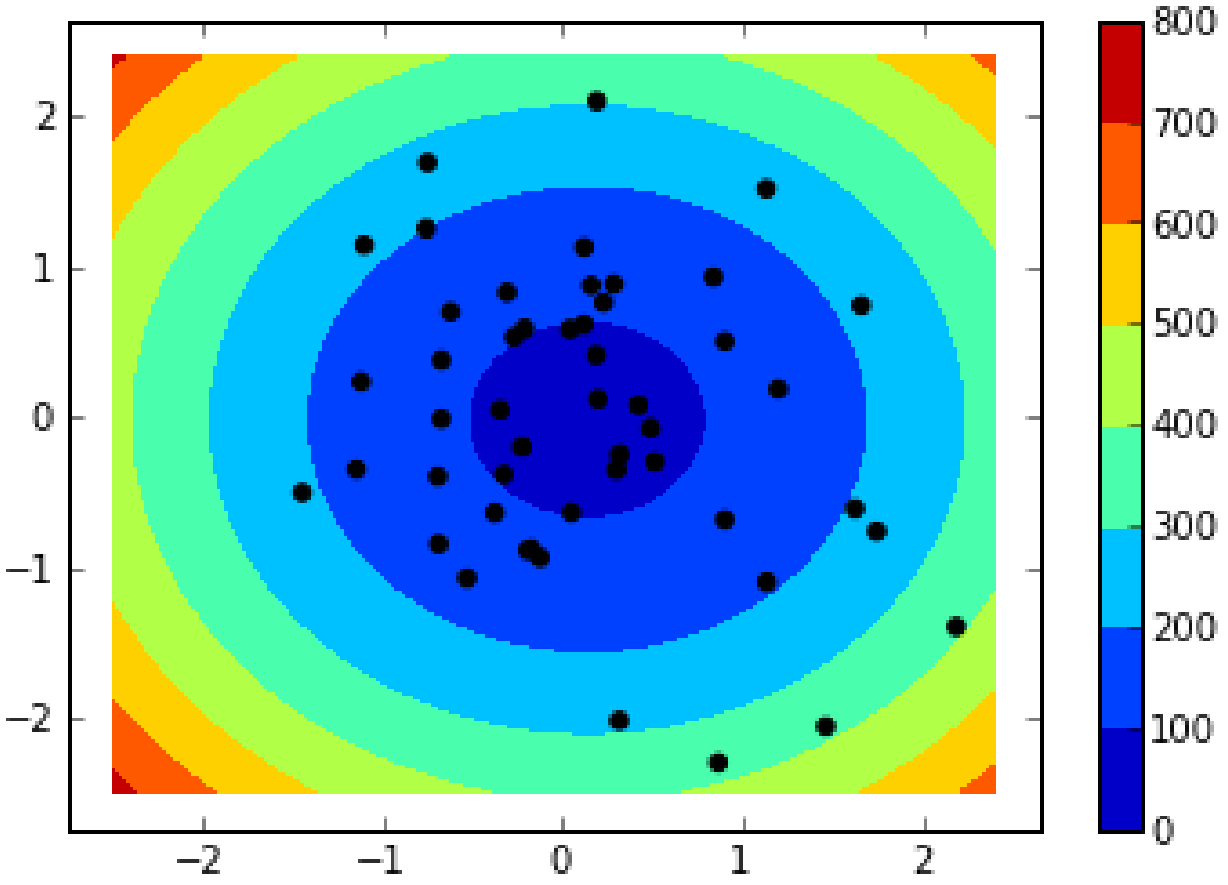
Propose an *analytical* as well as an *iterative* solution.

Example



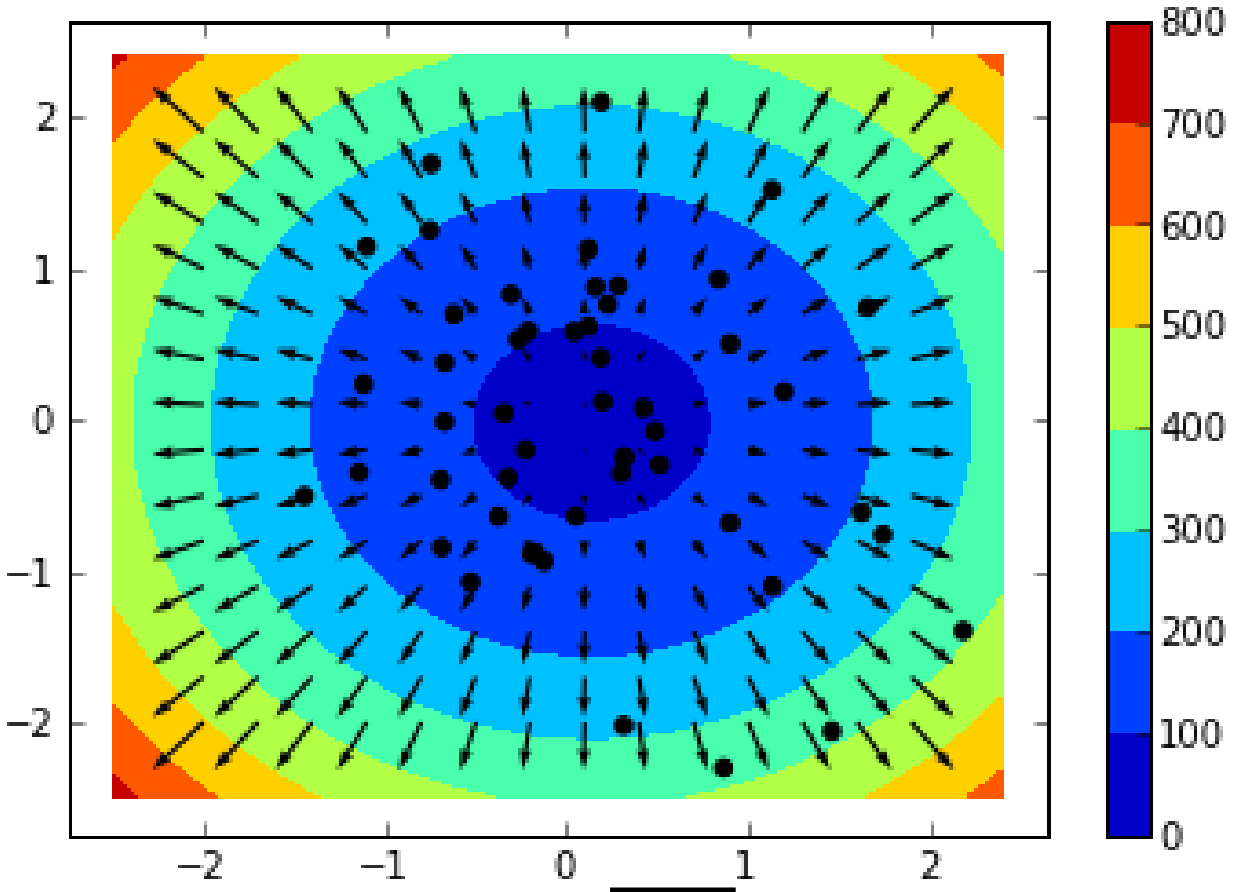
$$x_1, x_2, \ldots, x_{50}$$

Example



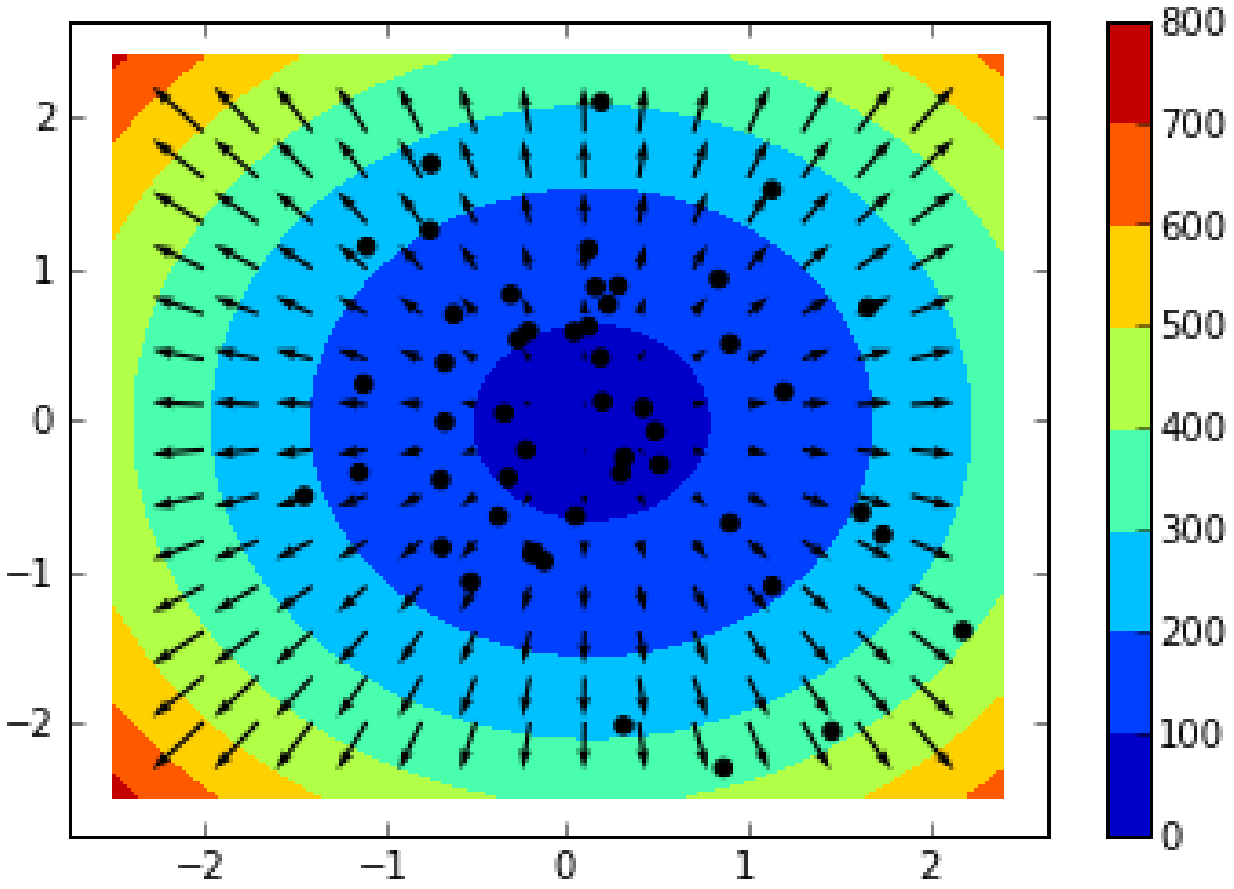
$$f(\boldsymbol{w}) = \sum_{\boldsymbol{i}} \|\boldsymbol{x_i} - \boldsymbol{w}\|^2$$

Example



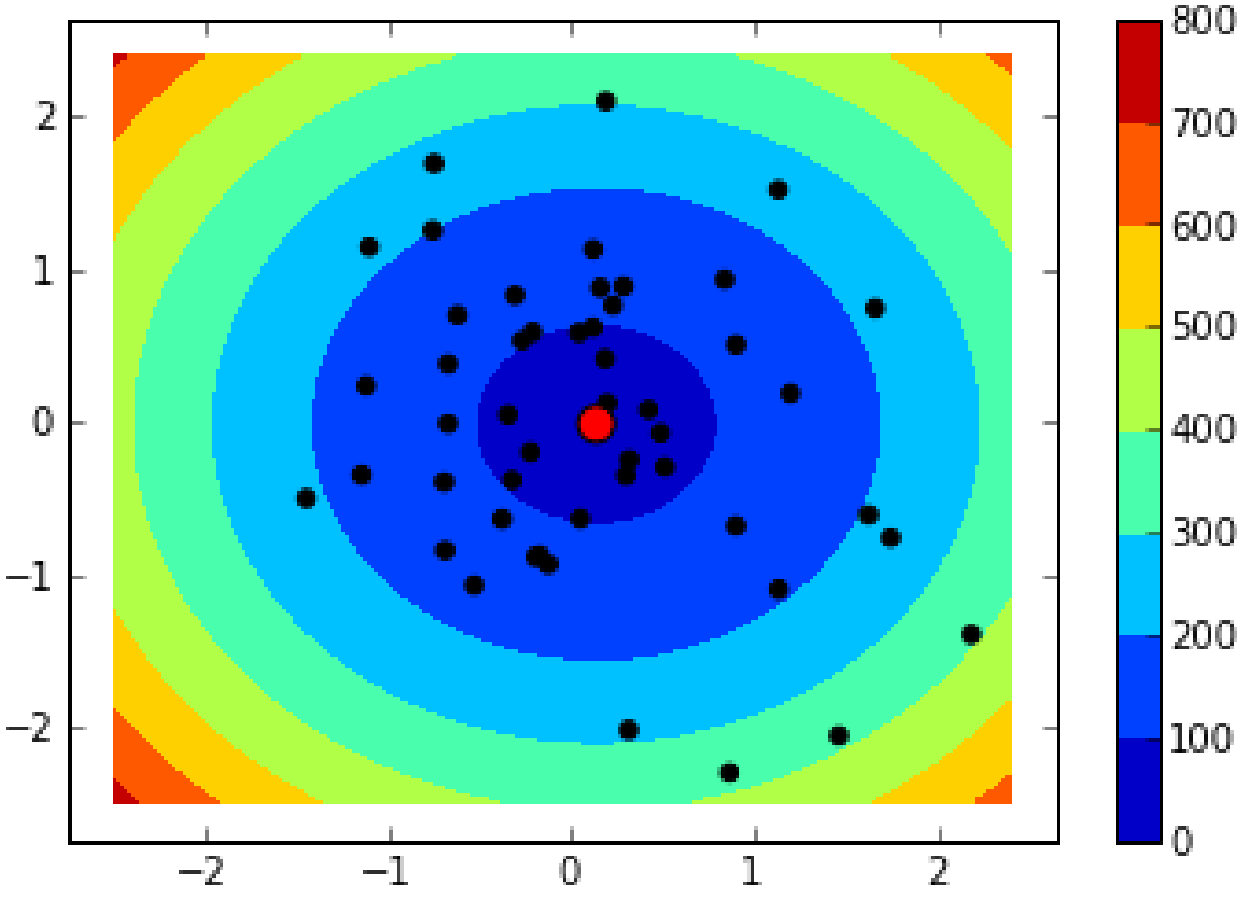
$$\nabla f(\boldsymbol{w}) = -\sum_{\boldsymbol{i}} 2(\boldsymbol{x_i} - \boldsymbol{w})$$

Example



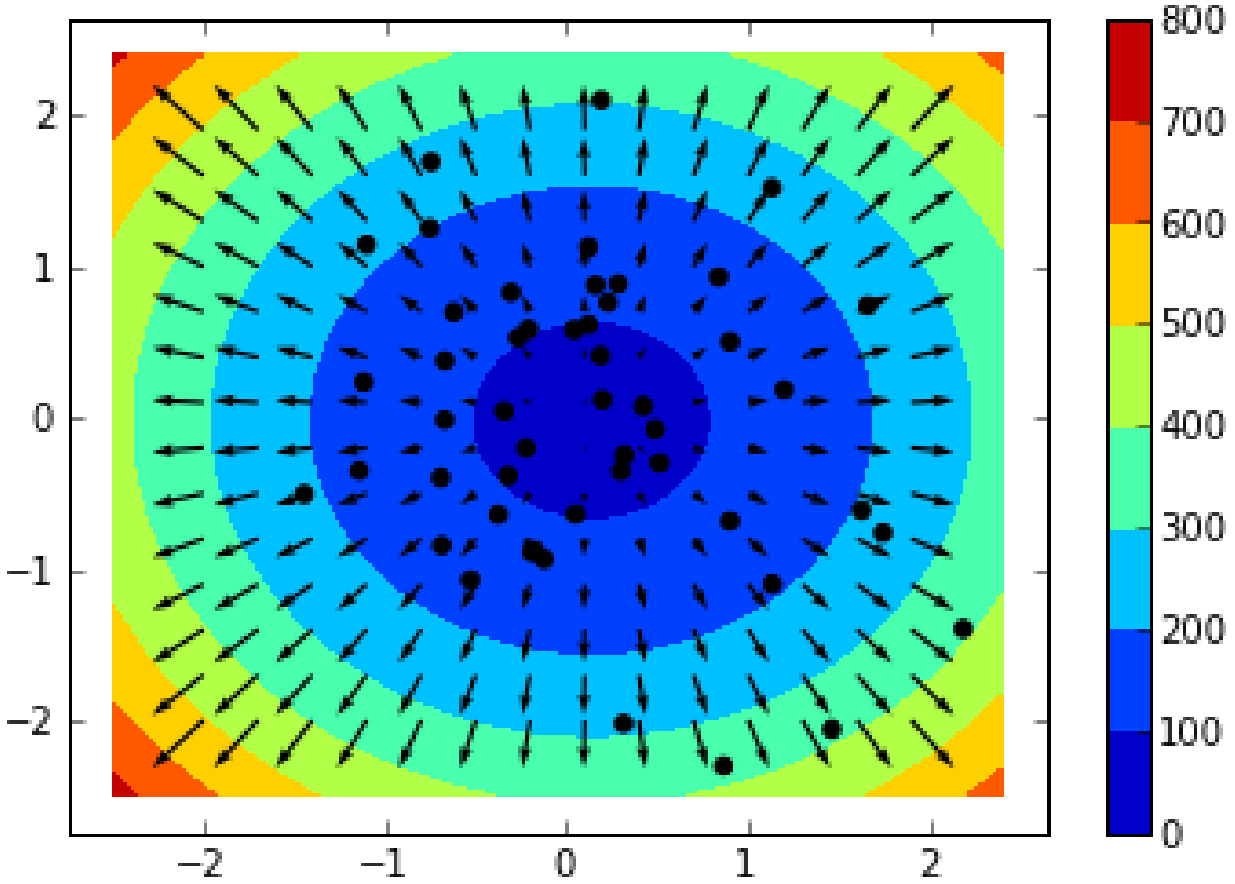
$$\nabla f(\boldsymbol{w}) = 2n(\boldsymbol{w} - \bar{\boldsymbol{x}})$$

Example



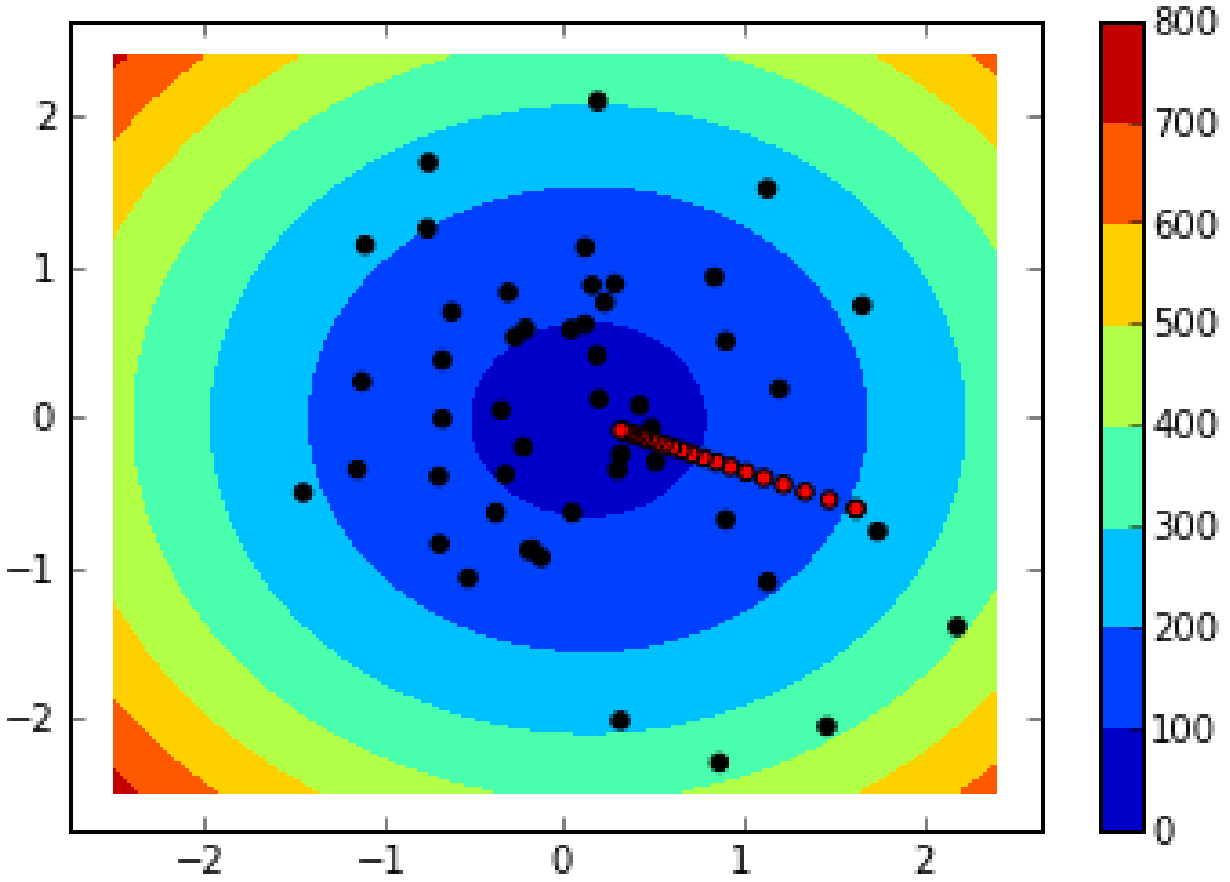
$$\nabla f(\boldsymbol{w}) = \mathbf{0}$$

Example



$$\nabla f(\boldsymbol{w})$$

Example



$$\Delta \mathbf{w} = -\mu \nabla f(\boldsymbol{w})$$

Stochastic gradient descent

- Whenever

$$f(\boldsymbol{w}) = \sum g(\boldsymbol{w}, \boldsymbol{x_k})$$

Stochastic gradient descent

- Whenever

$$f(\boldsymbol{w}) = \sum g(\boldsymbol{w}, \boldsymbol{x_k})$$

e.g.

$$\text{Error}(\boldsymbol{w}) = \sum (\text{model}_{\boldsymbol{w}}(\boldsymbol{x_k}) - \boldsymbol{y_k})^2$$

Stochastic gradient descent

- Whenever

$$f(\boldsymbol{w}) = \sum g(\boldsymbol{w}, \boldsymbol{x_k})$$

the gradient is also a sum:

$$\nabla f(\boldsymbol{w}) = \sum \nabla g(\boldsymbol{w}, \boldsymbol{x_k})$$

Stochastic gradient descent

- Whenever

$$f(\boldsymbol{w}) = \sum g(\boldsymbol{w}, \boldsymbol{x_k})$$

the gradient is also a sum:

$$\nabla f(\boldsymbol{w}) = \sum \nabla g(\boldsymbol{w}, \boldsymbol{x_k})$$

- The GD step is then also a sum

$$\Delta \boldsymbol{w_i} = -\mu \sum \nabla g(\boldsymbol{w_i}, \boldsymbol{x_k})$$

Stochastic gradient descent

- Whenever

$$f(\boldsymbol{w}) = \sum g(\boldsymbol{w}, \boldsymbol{x_k})$$

the gradient is also a sum:

$$\nabla f(\boldsymbol{w}) = \sum \nabla g(\boldsymbol{w}, \boldsymbol{x_k})$$

- The GD step is then also a sum

$$\Delta \boldsymbol{w_i} = -\mu \sum \nabla g(\boldsymbol{w_i}, \boldsymbol{x_k})$$

Stochastic gradient descent

- **Batch update:**

$$\Delta \boldsymbol{w_i} = -\mu \sum \nabla g(\boldsymbol{w_i}, \boldsymbol{x_k})$$

Stochastic gradient descent

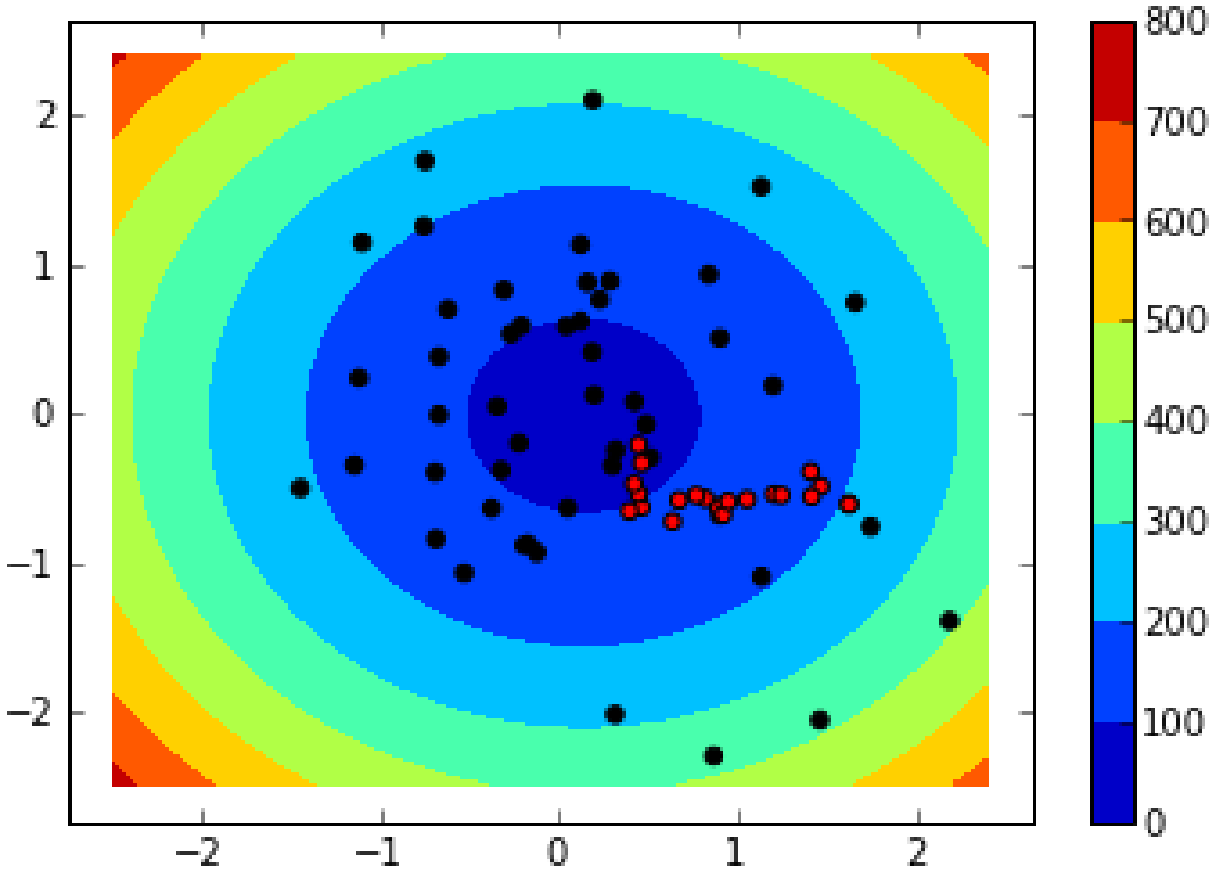
- **Batch update:**

$$\Delta \boldsymbol{w_i} = -\mu \sum \nabla g(\boldsymbol{w_i}, \boldsymbol{x_k})$$

- **On-line update:**

$$\Delta \boldsymbol{w_i} = -\mu \, \nabla g(\boldsymbol{w_i}, \boldsymbol{x_{random}})$$

Example



$$\Delta \mathbf{w} = \mu(\boldsymbol{x_i} - \boldsymbol{w})$$

Summary

- An **interpretation of the gradient**
- An **extremality criterion**
- An **iterative algorithm** for function minimization
- A **stochastic iterative algorithm** for function minimization

Quiz

- The symbol Δ is called _____ and denotes _____

- The symbol ∇ is called _____ and denotes _____

Quiz

- Gradient descent:

$$\Delta \boldsymbol{w}_i = \_\_\_\_\_$$

Quiz

- Gradient descent:

$$\Delta \boldsymbol{w}_i = -\mu \nabla_{\boldsymbol{w}} \mathbf{f}(\boldsymbol{w})$$

Quiz

- Gradient descent:

$$\Delta \boldsymbol{w}_i = -\mu \boldsymbol{c}$$

Quiz

- Gradient descent:

$$\Delta \boldsymbol{w}_i = -\mu \boldsymbol{c}$$

- Suppose the batch gradient descent step is

$$\Delta \boldsymbol{w}_i = \sum_{\boldsymbol{i}} \left(\boldsymbol{w}^T \boldsymbol{x}_i + |\boldsymbol{x}_i|^2 \right) \boldsymbol{e}$$

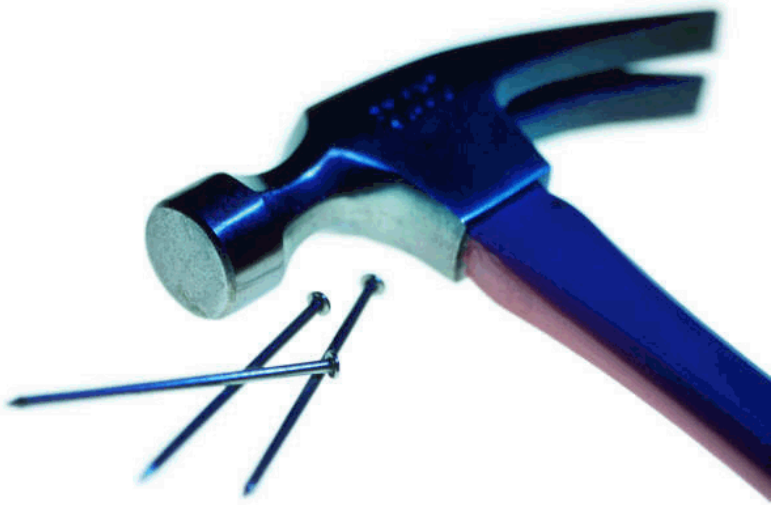
the corresponding stochastic gradient descent step is:

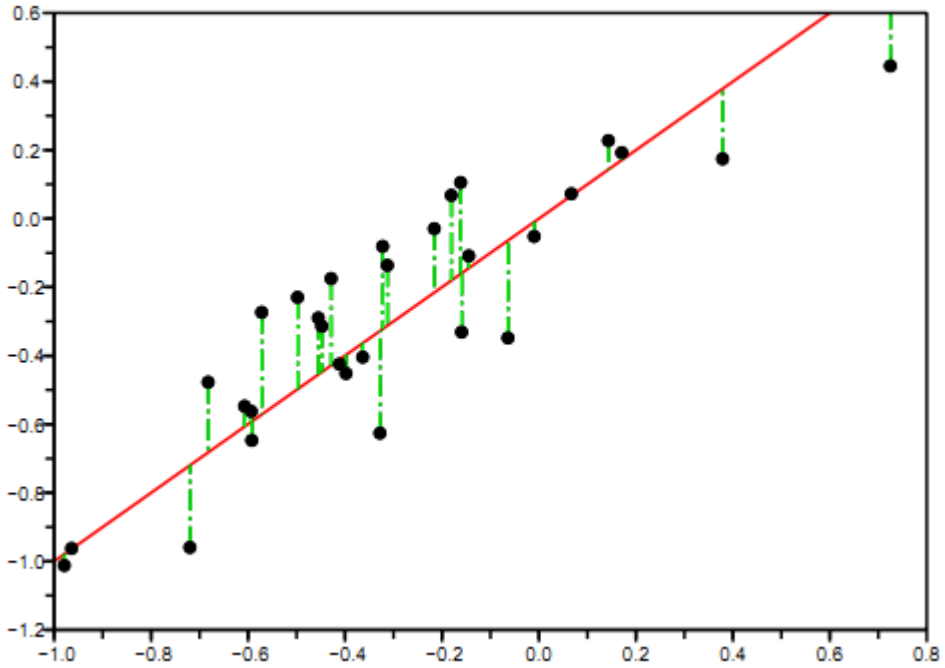
$$\Delta \boldsymbol{w}_i = \_\_\_\_\_$$

Quiz

- Can bacteria **learn**?

Questions?





Linear Regression

Konstantin Tretyakov
http://kt.era.ee

AACIMP 2012
August 2012

STACC Software Technology and Applications Competence Center

BIT

TARTU ÜLIKOOL 1632 UNIVERSITAS TARTUENSIS

Supervised Learning

- Let X and Y be some sets.
- Let there be a training dataset:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$
$$x_i \in X, y_i \in Y$$

Supervised Learning

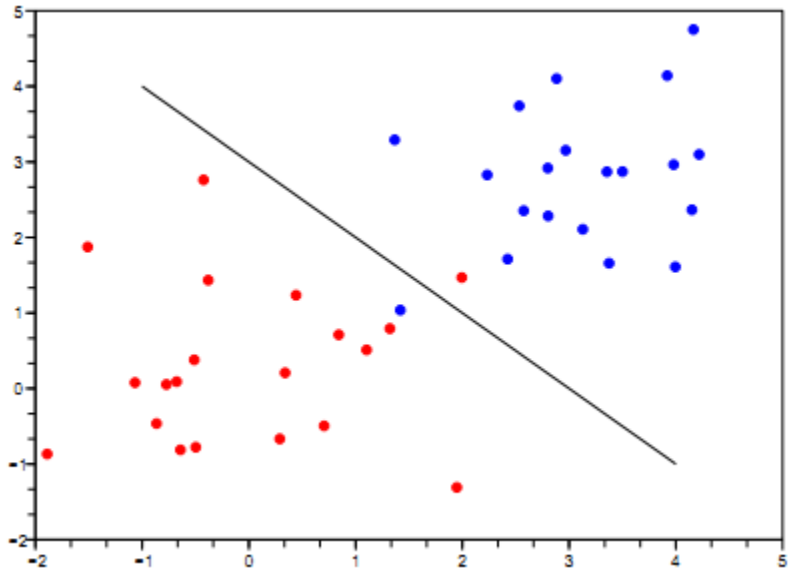
- Let X and Y be some sets.
- Let there be a training dataset:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$
$$x_i \in X, y_i \in Y$$

- **Supervised learning:**

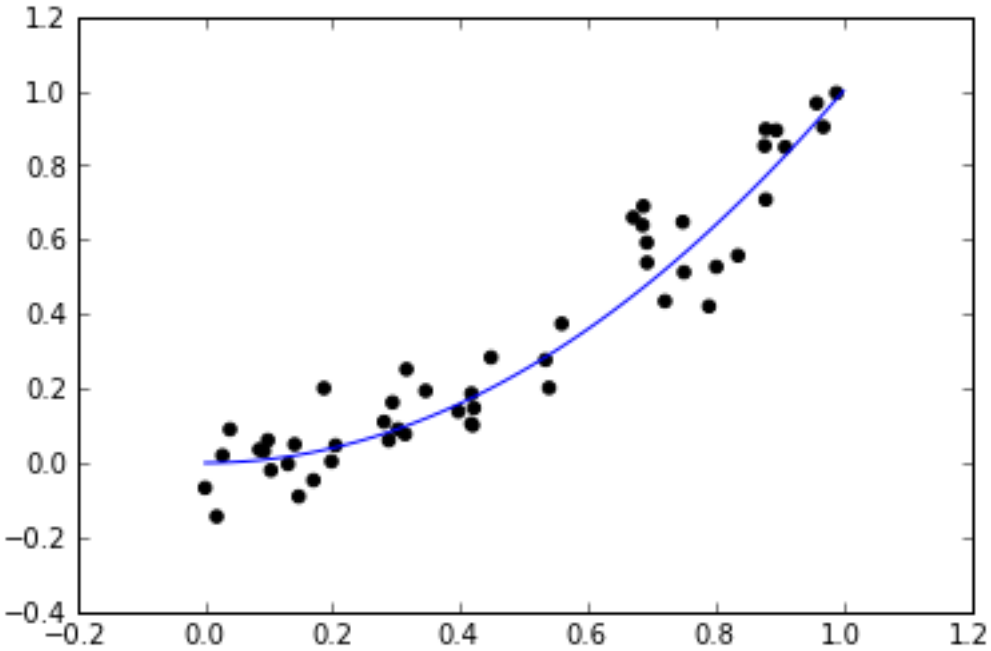
Find **a function** $f: X \to Y$, **generalizing the dependency** present in the data.

Classification



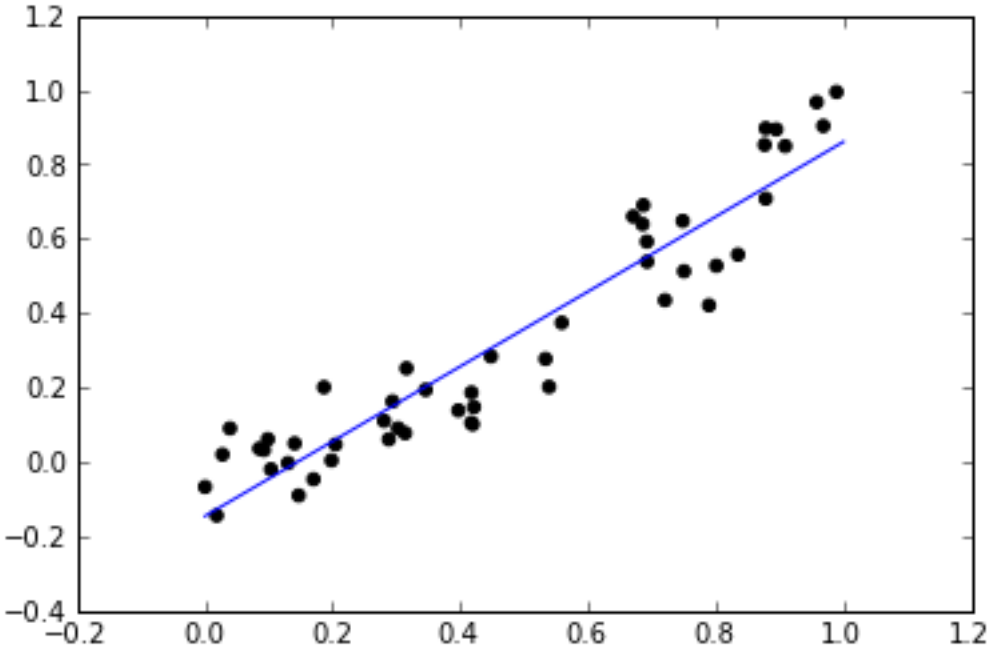
- $X = \mathbb{R}^2, \quad Y = \{\text{blue}, \text{red}\}$
- $D = \{\big((1.3, 0.8), \text{red}\big), \big((2.5,2.3), \text{blue}\big), \dots\}$
- $f(x_1, x_2) = \text{if } (x_1 + x_2) > 3 \text{ then blue else red}$

Regression



- $X = \mathbb{R}, \quad Y = \mathbb{R}$
- $D = \{(0.5, 0.26), (0.43, 0.08), \dots\}$
- $f(x) = x^2$

Linear Regression



- $X = \mathbb{R}, \qquad Y = \mathbb{R}$
- $D = \{(0.5, 0.26), (0.43, 0.08), \dots\}$
- $f(x) = -0.14 + 1.01\,x$

Linear Regression

$$X = \mathbb{R}^m, \qquad Y = \mathbb{R}$$

$$f(x_1, \ldots, x_m) = w_0 + w_1 x_1 + \cdots + w_m x_m$$

Linear Regression

$$X = \mathbb{R}^m, \qquad Y = \mathbb{R}$$

$$f(x_1, \ldots, x_m) = w_0 + w_1 x_1 + \cdots + w_m x_m$$

$$f(x_1, \ldots, x_m) = w_0 + \langle \boldsymbol{w}, \boldsymbol{x} \rangle$$

Inner product

Linear Regression

$$X = \mathbb{R}^m, \qquad Y = \mathbb{R}$$

$$f(x_1, \dots, x_m) = w_0 + w_1 x_1 + \cdots + w_m x_m$$

$$f(x_1, \dots, x_m) = w_0 + \langle \boldsymbol{w}, \boldsymbol{x} \rangle$$

$$f(x_1, \dots, x_m) = w_0 + (w_1, \dots, w_m) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}$$

Linear Regression

$$X = \mathbb{R}^m, \qquad Y = \mathbb{R}$$

$$f(x_1, \dots, x_m) = w_0 + w_1 x_1 + \cdots + w_m x_m$$

$$f(x_1, \dots, x_m) = w_0 + \langle \boldsymbol{w}, \boldsymbol{x} \rangle$$

$$f(x_1, \dots, x_m) = w_0 + (w_1, \dots, w_m) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}$$

$$f(x_1, \dots, x_m) = w_0 + \boldsymbol{w}^T \boldsymbol{x}$$

Linear Regression

$$f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^T \boldsymbol{x}$$

Linear Regression

$$f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^T \boldsymbol{x}$$

Bias term

Linear Regression

$$f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^T \boldsymbol{x}$$

$$f(x_1, \ldots, x_m) = (w_0, w_1, \ldots, w_m) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix}$$

Linear Regression

$$f(\boldsymbol{x}) = w_0 + \boldsymbol{w}^T \boldsymbol{x}$$

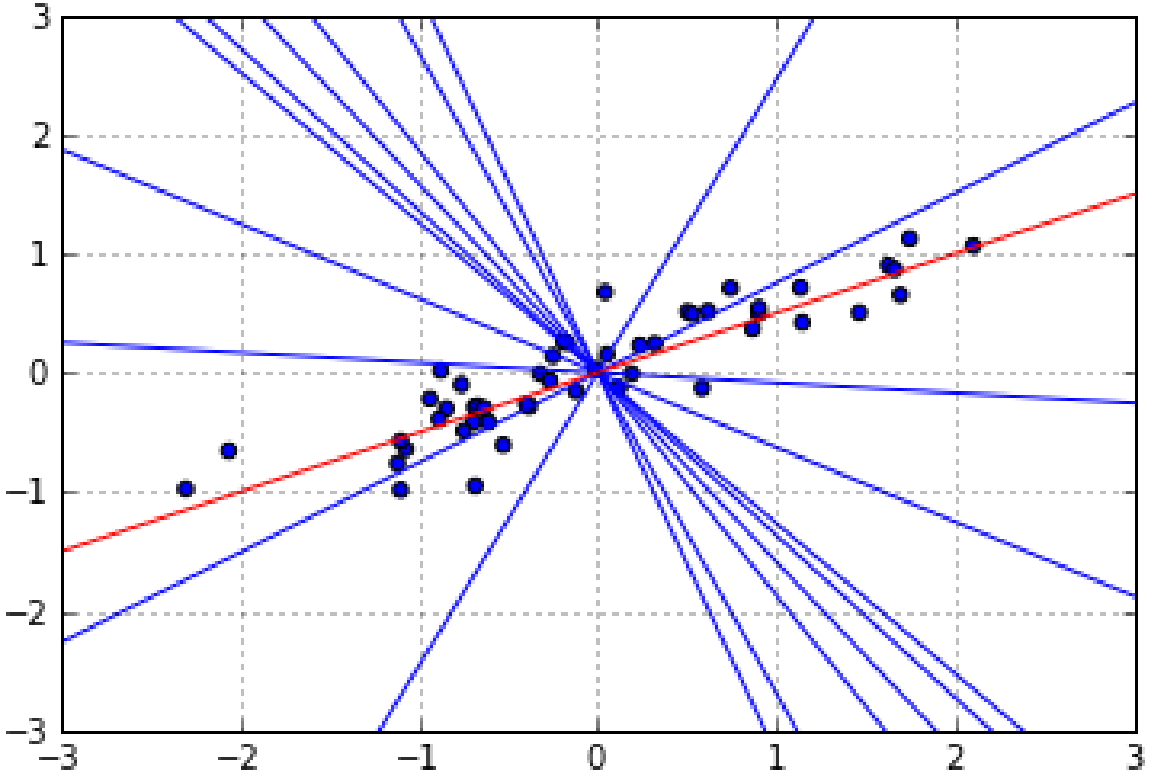
$$f(x_1, \ldots, x_m) = (w_0, w_1, \ldots, w_m) \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_m \end{pmatrix}$$

$$f(\boldsymbol{x}) = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}}$$

Linear Regression

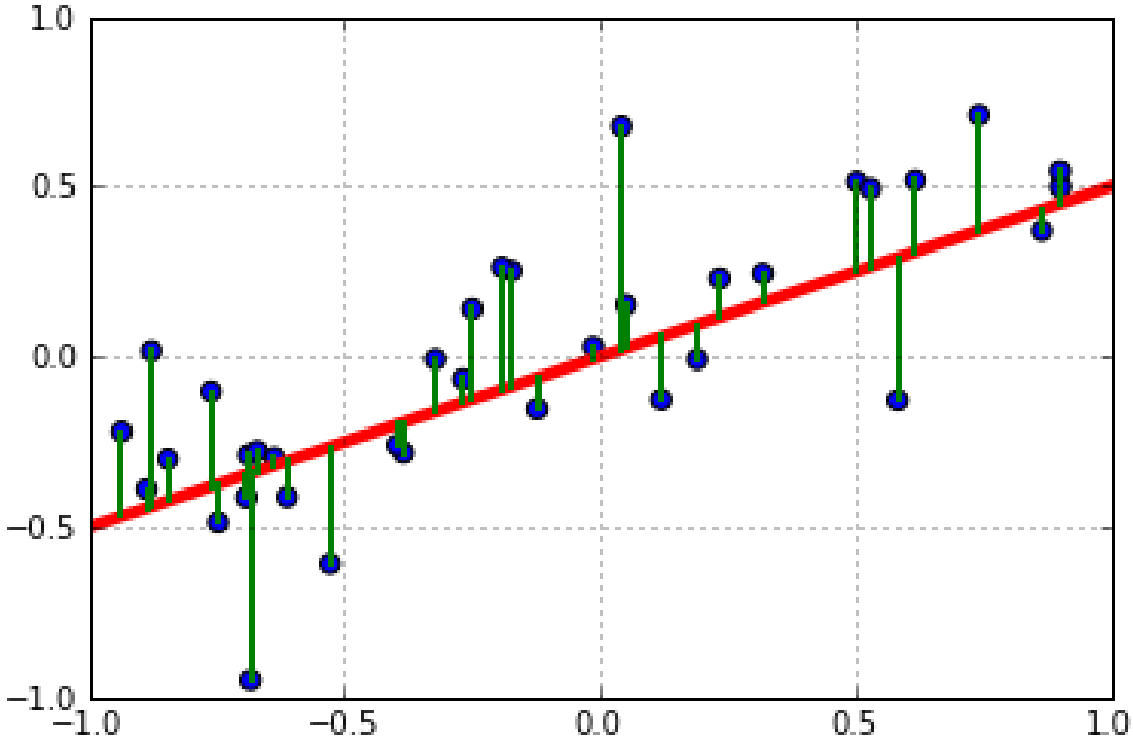
$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x}$$

Linear Regression



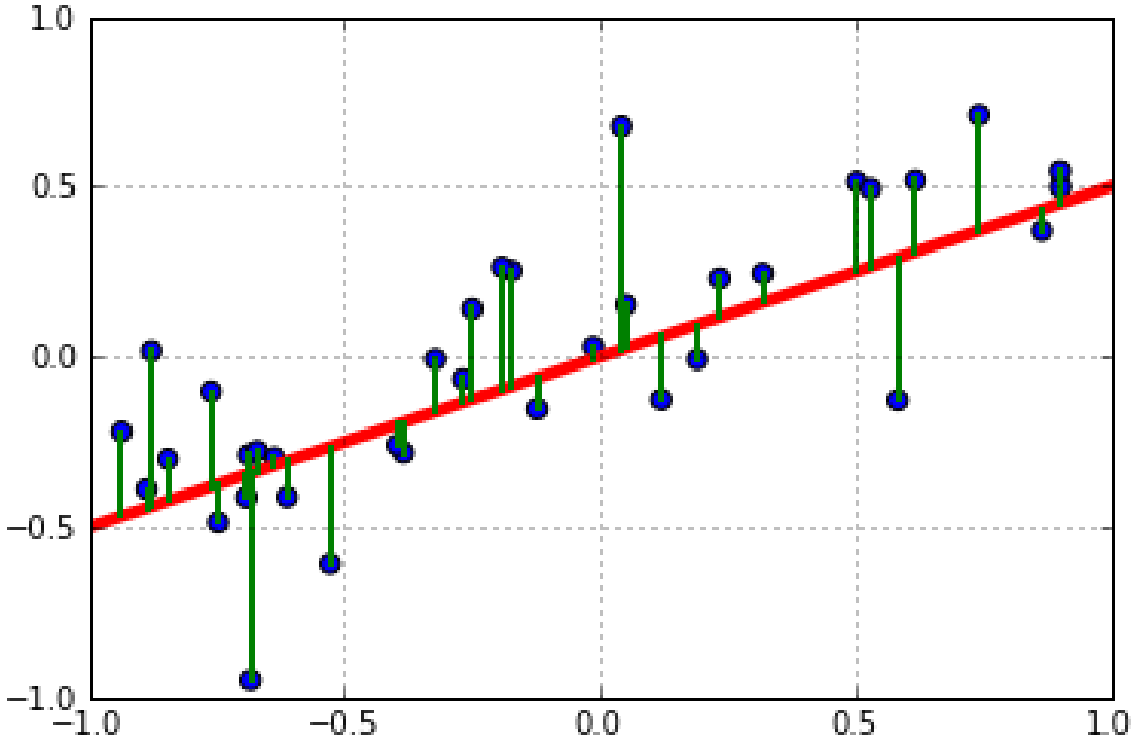
$$f(x) = wx$$

Objective Function



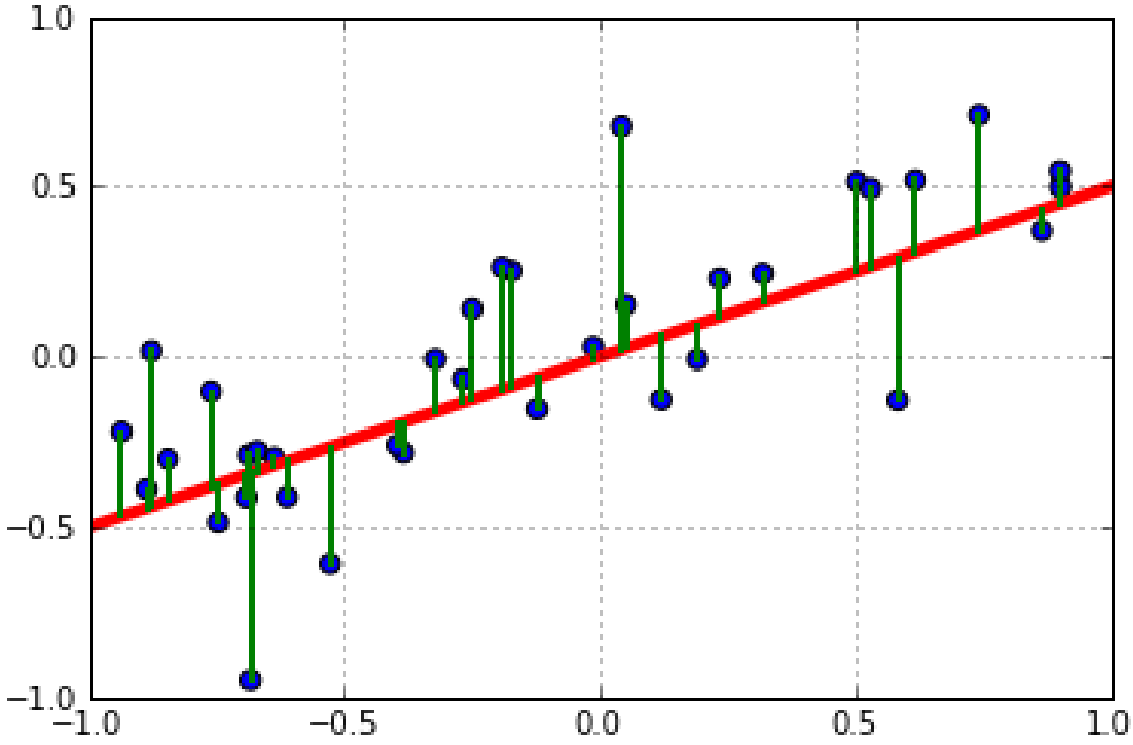
$$e_i = (f(x_i) - y_i)^2$$

Objective Function



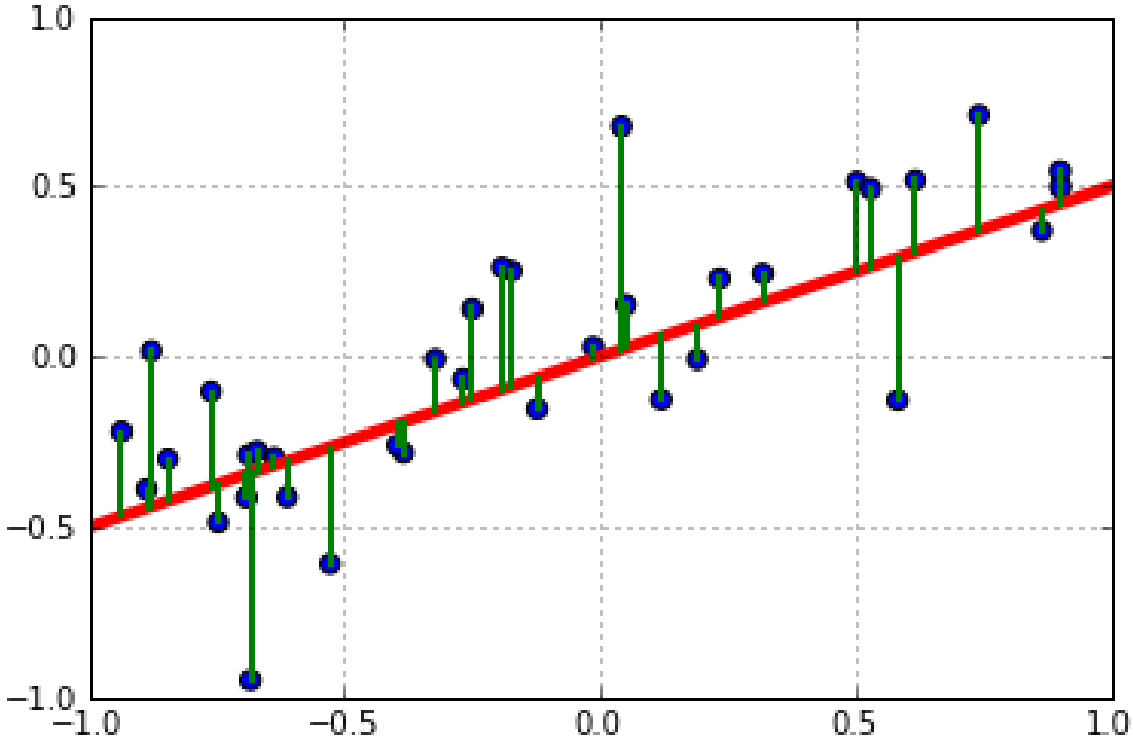
$$e_i = (\hat{y}_i - y_i)^2$$

Objective Function



$$e_i = (wx_i - y_i)^2$$

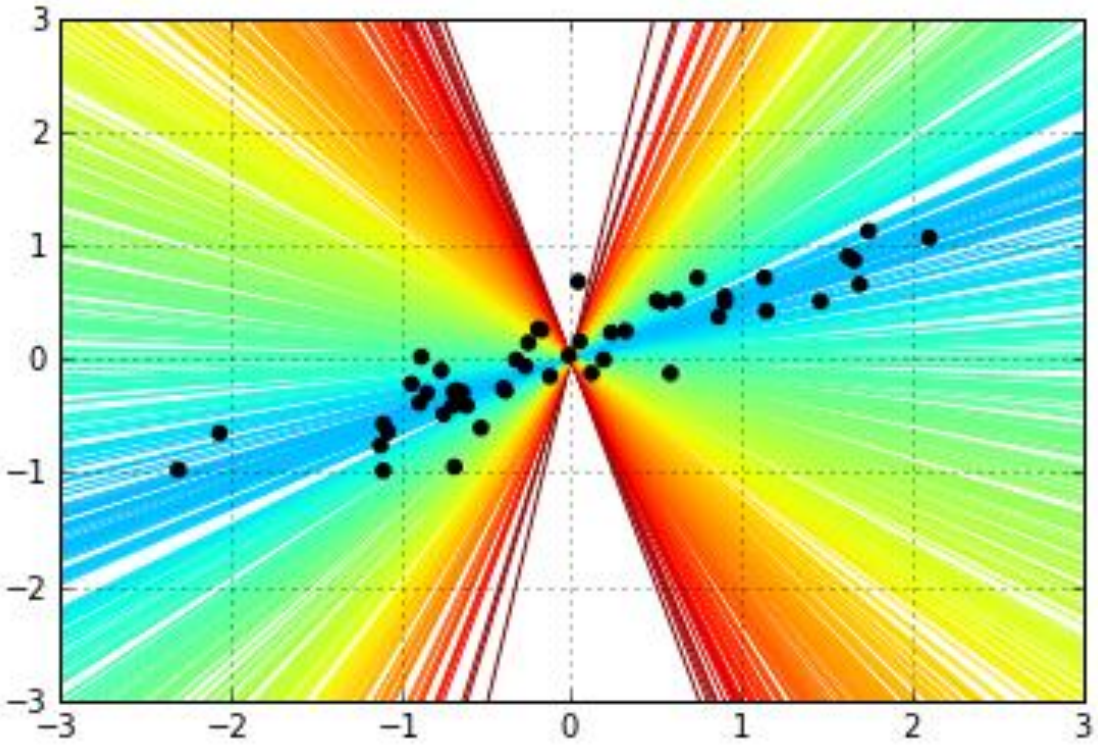
Objective Function



"OLS"

$$E_D(w) = \sum_i (wx_i - y_i)^2$$

Objective Function



$$E_D(w) = \sum_i (wx_i - y_i)^2$$

Objective Function

$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{w}^T \boldsymbol{x_i} - y_i\right)^2$$

Objective Function

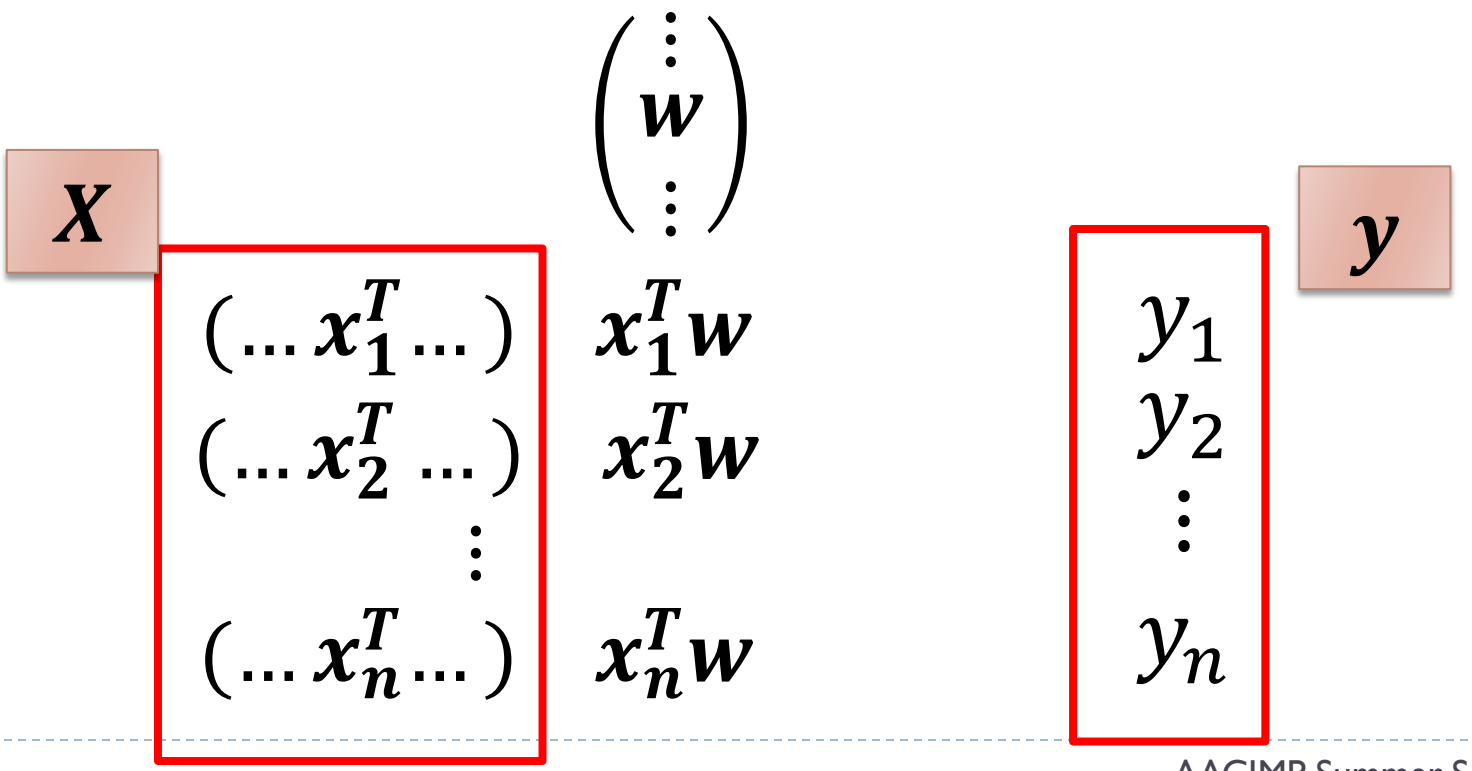
$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{x_i^T w} - y_i\right)^2$$

$$\begin{pmatrix} \vdots \\ \boldsymbol{w} \\ \vdots \end{pmatrix}$$

$$\begin{matrix} (\ldots \boldsymbol{x_1^T} \ldots) & \boldsymbol{x_1^T w} & y_1 \\ (\ldots \boldsymbol{x_2^T} \ldots) & \boldsymbol{x_2^T w} & y_2 \\ \vdots & & \vdots \\ (\ldots \boldsymbol{x_n^T} \ldots) & \boldsymbol{x_n^T w} & y_n \end{matrix}$$

Objective Function

$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{x}_i^T \boldsymbol{w} - y_i\right)^2$$



Objective Function

$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{x_i^T w} - y_i\right)^2$$

$$\boldsymbol{Xw - y}$$

Objective Function

$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{x_i^T w} - y_i\right)^2$$

$$\|\boldsymbol{Xw} - \boldsymbol{y}\|^2$$

Objective Function

$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{x_i^T w} - y_i\right)^2$$

$$\|\boldsymbol{Xw} - \boldsymbol{y}\|^2$$

$$\boldsymbol{Xw} \approx \boldsymbol{y}$$

Objective Function

$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{x_i^T w} - y_i\right)^2$$

$$\|\boldsymbol{Xw} - \boldsymbol{y}\|^2$$

$$\boldsymbol{Xw} \approx \boldsymbol{y}$$

$\boldsymbol{w} \approx \boldsymbol{X^{-1}y}$?

Objective Function

$$E_D(\boldsymbol{w}) = \sum_i \left(\boldsymbol{x_i^T w} - y_i\right)^2$$

$$\|\boldsymbol{Xw} - \boldsymbol{y}\|^2$$

$$\boldsymbol{Xw} \approx \boldsymbol{y}$$

$$\boldsymbol{w} = \boldsymbol{X}^+\boldsymbol{y}\,!$$

Optimization

$$\text{argmin}_{\mathbf{w}} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$$

Optimization

$$\operatorname{argmin}_{\mathbf{w}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$$

Optimization

$$\text{argmin}_{\mathbf{w}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$$

$$\|\boldsymbol{a}\|^2 = \boldsymbol{a}^T \boldsymbol{a}$$

$$\frac{1}{2} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

Optimization

$$\text{argmin}_{\mathbf{w}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2$$

$$\frac{1}{2} (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^T (\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

$$(\boldsymbol{a} + \boldsymbol{b})^T = (\boldsymbol{a}^T + \boldsymbol{b}^T)$$

$$\frac{1}{2} (\boldsymbol{w}^T \boldsymbol{X}^T - \boldsymbol{y}^T)(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

Optimization

$$\frac{1}{2}(\boldsymbol{w}^T\boldsymbol{X}^T - \boldsymbol{y}^T)(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

$$\boldsymbol{a}(\boldsymbol{b}+\boldsymbol{c}) = \boldsymbol{a}\boldsymbol{b} + \boldsymbol{a}\boldsymbol{c}$$

$$\frac{1}{2}\left(\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y}\right)$$

Optimization

$$\frac{1}{2}(\boldsymbol{w}^T\boldsymbol{X}^T - \boldsymbol{y}^T)(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$$

$$\frac{1}{2}(\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{y})$$

$$\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} = \boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{y} = \text{scalar}$$

$$\frac{1}{2}(\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T\boldsymbol{y})$$

Optimization

$$E(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{y}^T \boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T \boldsymbol{y})$$

Optimization

$$E(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{w}^T\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T\boldsymbol{y})$$

$$\nabla(f + g) = \nabla\mathbf{f} + \nabla\mathbf{g}$$
$$\nabla(\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w}) = 2\boldsymbol{A}\boldsymbol{w}$$
$$\nabla(\boldsymbol{a}^T\boldsymbol{w}) = \boldsymbol{a}$$

$$\nabla E(\boldsymbol{w}) = \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T\boldsymbol{y}$$

Optimization

$$E(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{y}^T \boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T \boldsymbol{y})$$

$$\nabla E(\boldsymbol{w}) = \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T \boldsymbol{y}$$
$$\boldsymbol{0} = \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T \boldsymbol{y}$$
$$\boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T \boldsymbol{y}$$

Optimization

$$E(\boldsymbol{w}) = \frac{1}{2}(\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - 2\boldsymbol{y}^T \boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^T \boldsymbol{y})$$

$$\nabla E(\boldsymbol{w}) = \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T \boldsymbol{y}$$
$$\boldsymbol{0} = \boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} - \boldsymbol{X}^T \boldsymbol{y}$$
$$\boldsymbol{X}^T \boldsymbol{X}\boldsymbol{w} = \boldsymbol{X}^T \boldsymbol{y}$$

$$\boldsymbol{w} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$$

Linear Regression Solution

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Linear Regression Solution

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

```
X = matrix(X)
y = matrix(y)
w =(X.T * X).I * X.T * y
```

Linear Regression Solution

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Moore-Penrose pseudoinverse

```
X = matrix(X)
y = matrix(y)
w =(X.T * X).I * X.T * y
```

Linear Regression Solution

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$\boldsymbol{w} = \boldsymbol{X}^{+}\boldsymbol{y}$$

Moore-Penrose pseudoinverse

```
X = matrix(X)
y = matrix(y)
w =(X.T * X).I * X.T * y
w = pinv(X) * y
```

Linear Regression & SKLearn

```
from sklearn.linear_model import
                        LinearRegression


model = LinearRegression()
model.fit(X, y)

w=(model.intercept_,model.coef_)
model.predict(X_new)
```

Stochastic Gradient Regression

$$\boldsymbol{\Delta w} = -\mu(\boldsymbol{w}^T \boldsymbol{x}_i - y_i)\boldsymbol{x}_i$$

Stochastic Gradient Regression

$$\boldsymbol{\Delta w} = -\mu e_i \boldsymbol{x_i}$$

Stochastic Gradient Regression

$$\Delta \boldsymbol{w} = -\mu e_i \boldsymbol{x_i}$$

```
from sklearn.linear_model import SGDRegressor

model = SGDRegressor(alpha=0, n_iter=30)
model.fit(X, y)
```

Polynomial Regression

Say we'd like to fit a model:

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2^2 + w_3 x_1 x_2$$

Polynomial Regression

Say we'd like to fit a model:

$$f(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2^2 + w_3 x_1 x_2$$

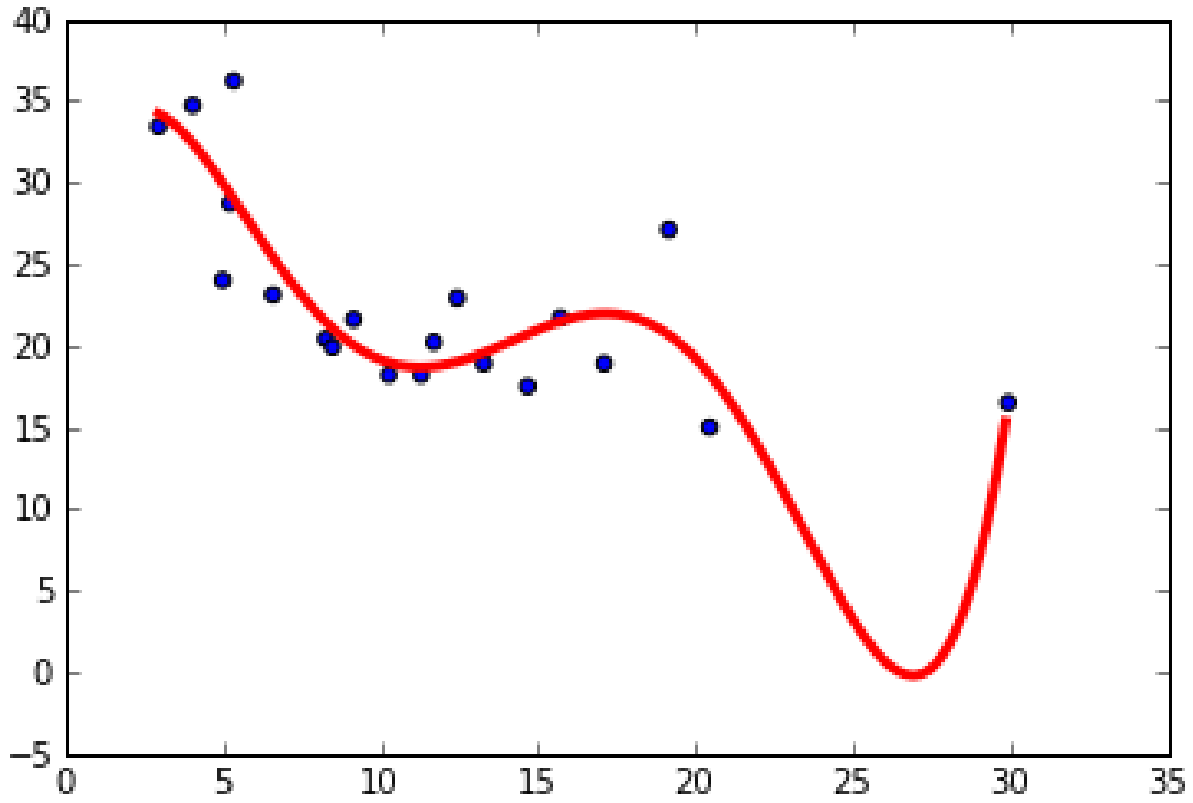
Simply transform the features and proceed as normal:

$$(x_1, x_2) \rightarrow (x_1, x_2^2, x_1 x_2)$$

Single-variable Polynomial OLS

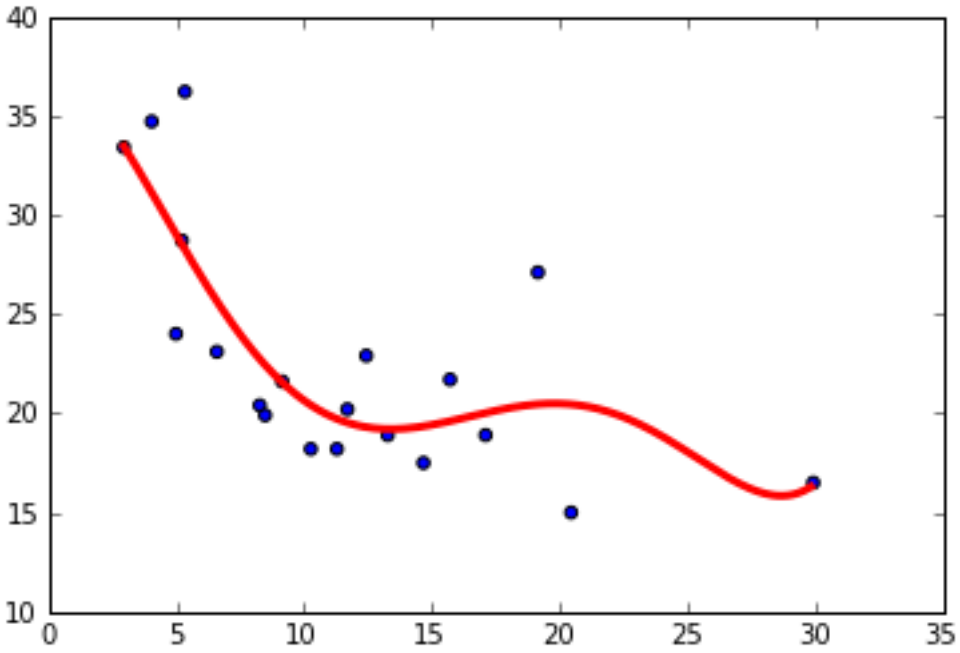
```
# n x 1 matrix
x = matrix(…)

# Add bias & square features
X = hstack([x**0, x**1, x**2])

# Solve for w
w = pinv(X) * y
```

Overfitting



Regularization

$$E(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{w}\|^2$$

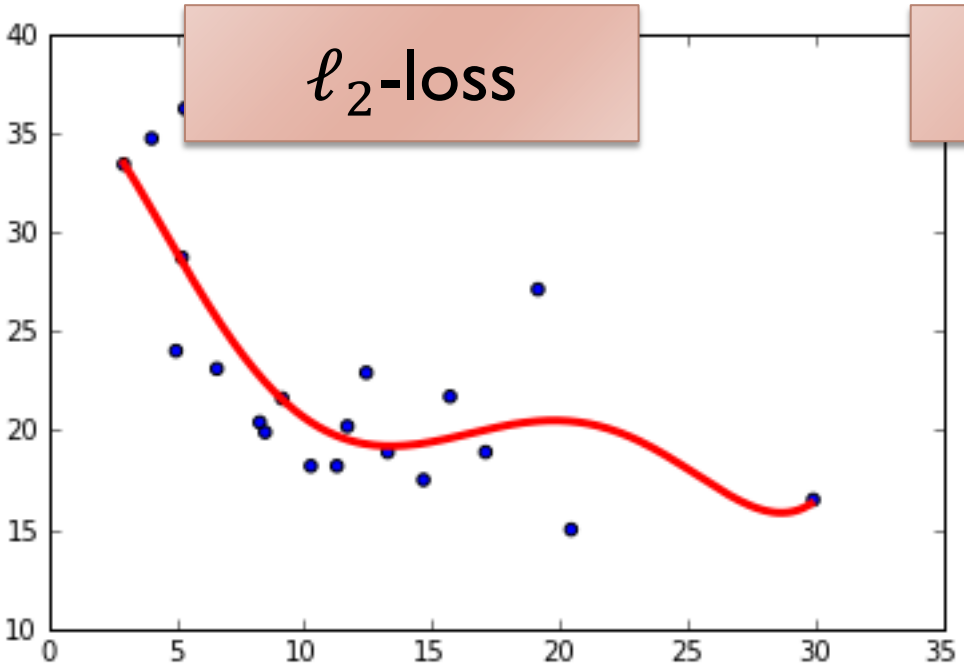


Regularization

$$E(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{w}\|^2$$

ℓ_2-loss

ℓ_2-penalty



Regularization

$$E(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda|w|_1$$

ℓ_2-loss

ℓ_1-penalty

Regularization

$$E(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda|w|_0$$

ℓ_2-loss

ℓ_0-penalty

Regularization

$$E(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda|w|_0$$

ℓ_2-loss

ℓ_0-penalty

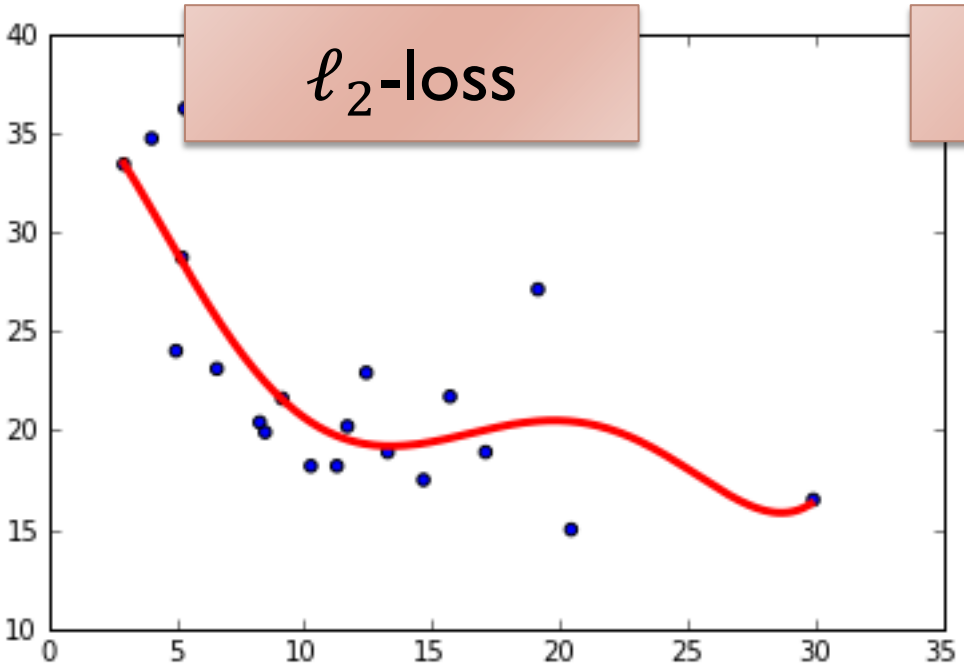
```
>>> SGDRegressor?
Parameters
----------
loss : str, 'squared_loss' or 'huber' ...
...
penalty : str, 'l2' or 'l1' or 'elasticnet'
...
```

Regularization

$$E(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{w}\|^2$$

ℓ_2-loss

ℓ_2-penalty



Ridge regression

$$\text{argmin}_{\boldsymbol{w}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{w}\|^2$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\, \boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

$$\boldsymbol{I} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Ridge regression

$$\text{argmin}_{\boldsymbol{w}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{w}_*\|^2$$

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\, \boldsymbol{I}_*)^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

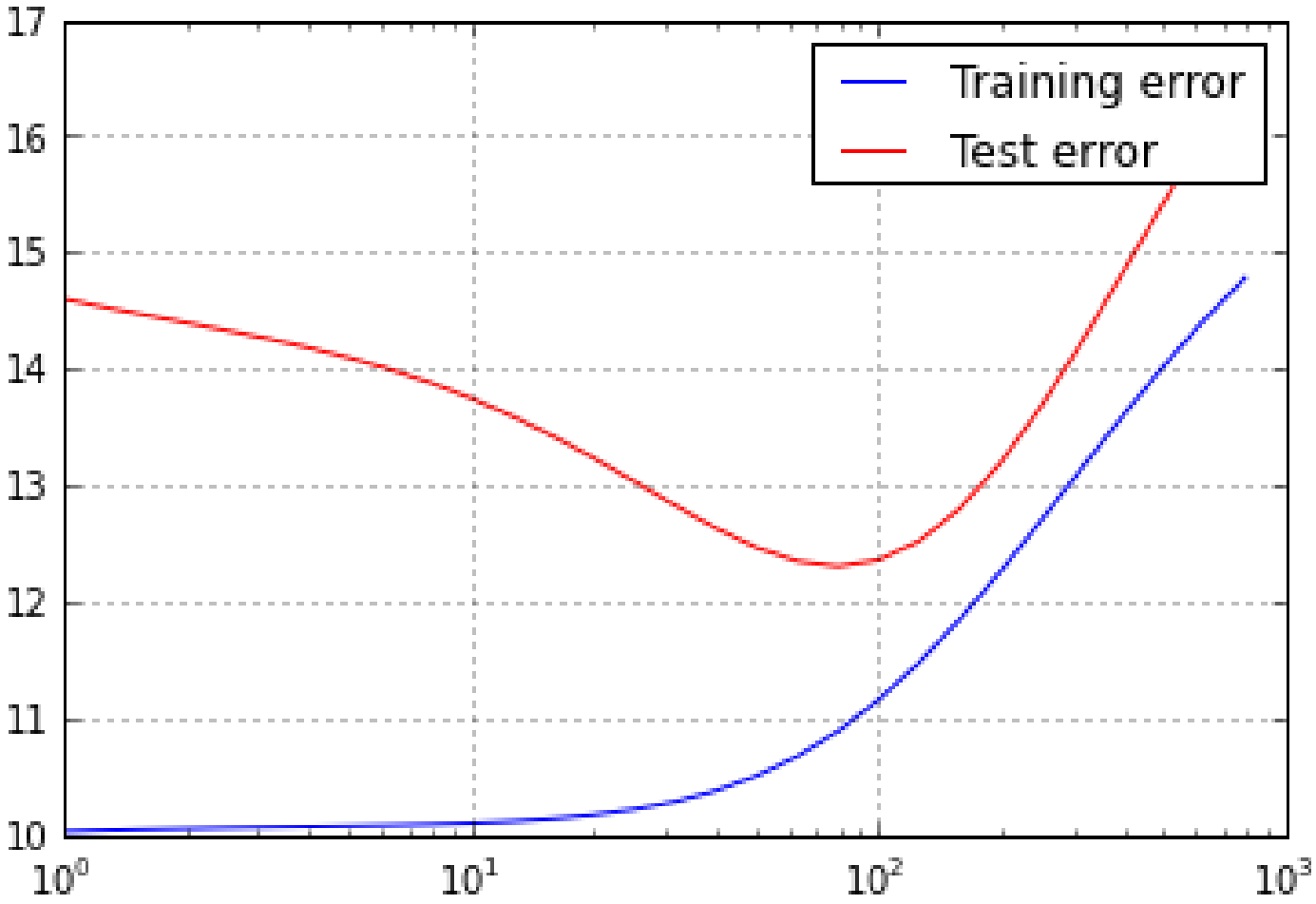
The bias term w_0 is usually **not penalized.**

$$\boldsymbol{I}_* = \begin{pmatrix} \mathbf{0} & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

Exercise

Derive an SGD algorithm for Ridge Regression.

Effects of Regularization



Quiz

- OLS linear regression searches for a _____ model that has the best _____.
- Analytic solution for OLS regression:

$$\boldsymbol{w} = \_\_\_\_\_$$

- Stochastic gradient solution for OLS regression:

$$\boldsymbol{\Delta w} = \_\_\_\_\_$$

Quiz

- Large number of model parameters and/or small data may lead to _____.
- We address overfitting by _____.
- "Ridge regression" means __-loss and ___-penalty.
- Analytic solution for Ridge regression:

$$\boldsymbol{w} = \_\_\_\_\_$$

Quiz

- As we increase regularization strength (i.e. increase λ), the training error _____.

- … and the test error _____.

Questions?