

# EEE5015: Machine Learning & Artificial Intelligence

Zhiyun Lin



南方科技大学  
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY  
电子与电气工程系  
DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING




### □ Regression

- Features
- Training examples
- Model
- Loss function
- Optimization

# Problems for Today

- o What should I watch this Friday?



Find Movies, TV shows, Celebrities and more...


All

Movies, TV & Showtimes

Celebs, Events & Photos

News & Community

Watchlist



[See More on IMDb Pro »](#)

## The Martian (2015)

PG-13 | 144 min | Adventure, Comedy, Drama |  
2 October 2015 (USA)

**Your rating:** ★★★★★★- /10

Ratings: **8.1**/10 from **271,829 users** Metascore: **80/100**

Reviews: **750 user** | **499 critic** | **46 from Metacritic.com**

During a manned mission to Mars, Astronaut Mark Watney is presumed dead after a fierce storm and left behind by his crew. But Watney has survived and finds himself stranded and alone on the hostile planet. With only meager supplies, he must draw upon his ingenuity, wit and spirit to subsist and find a way to signal to Earth that he is alive.

**Director:** [Ridley Scott](#)

**Writers:** [Drew Goddard](#) (screenplay), [Andy Weir](#) (book)

**Stars:** [Matt Damon](#), [Jessica Chastain](#), [Kristen Wiig](#) |  
[See full cast and crew »](#)

+ Watchlist

Watch Trailer

Share...

# Problems for Today

- o What should I watch this Friday?

All

[Movies, TV & Showtimes](#) [Celebs, Events & Photos](#) [News & Community](#) [Watchlist](#)



See More on IMDb Pro »

## Point Break (2015)

PG-13 | 114 min | Action, Crime, Sport | 25 December 2015 (USA)

**Your rating:** ★★★★★★☆☆ -/10  
Ratings: **5.4**/10 from 7,322 users Metascore: 34/100  
Reviews: 60 user | 84 critic | 19 from Metacritic.com

A young FBI agent infiltrates an extraordinary team of extreme sports athletes he suspects of masterminding a string of unprecedented, sophisticated corporate heists. "Point Break" is inspired by the classic 1991 hit.

**Director:** [Ericson Core](#)  
**Writers:** [Kurt Wimmer](#) (screenplay), [Rick King](#) (story), [5 more credits](#) »  
**Stars:** [Édgar Ramírez](#), [Luke Bracey](#), [Ray Winstone](#) | [See full cast and crew](#) »

[+ Watchlist](#) [Watch Trailer](#) [Share...](#)

 15

# Problems for Today

- o **Goal:** Predict movie rating automatically!



IMDb Find Movies, TV shows, Celebrities and more... All

Movies, TV & Showtimes Celebs, Events & Photos News & Community Watchlist

**Point Break** (2015) PG-13 | 114 min | 25 December 2015

**5.4** **Our rating:** ★★★★★★★★★★ -/10  
Ratings: 5.4/10 from 7,322 users Metascore: 34/100  
Reviews: 60 user | 84 critic | 19 from Metacritic.com

A young FBI agent infiltrates an extraordinary team of extreme sports athletes he suspects of masterminding a string of unprecedented, sophisticated corporate heists. "Point Break" is inspired by the classic 1991 hit.

**Director:** Ericson Core  
**Writers:** Kurt Wimmer (screenplay), Rick King (story), 5 more credits »  
**Stars:** Édgar Ramírez, Luke Bracey, Ray Winstone | See full cast and crew »

See More on IMDb Pro »

+ Watchlist Watch Trailer Share...

Predict this automatically!

Factors:

- Year of release of the film in cinemas
- Length of the film in minutes
- Budget for the film's production
- Number of positive votes received by viewers
- Genre of the film including Action, Animation, Comedy, Documentary, Drama, Romance and Short

# Problems for Today

- o **Goal:** How many followers will I get?

Red Leather Jacket

Updated on Jan 09, 2016



From This User

+1 282 VOTES

5 COMMENTS

67 FAVORITES

f Like 0

t Tweet

G+1 0

...

Pin it 2

Tags

Chic  
Everyday  
Winter

SHARE

# Problems for Today

- **Goal:** Predict the price of the house

The image is a screenshot of the Nationwide House Price Index website. The top navigation bar is blue with white text links: "Why choose Nationwide?", "Have your say", "Corporate information", "Media, Policy & Legal", "House Price Index" (which is highlighted), and "Investor relations". Below the navigation bar is a large banner image of a row of houses. Overlaid on the banner is a white box with the text "Nationwide" in red and "House Price Index" in blue. Below the banner is a row of five white buttons: "Headlines", "House Price calculator" (highlighted in red), "Report archive", "Download data", and "Methodology". Below the buttons is the "House Price Calculator" section. It has a sub-header "Instructions" in blue. There is a list of four instructions: 1. Property Value: Enter the price paid for, or a more recent valuation of your property. Please ensure the value is entered without commas, for example 150000, rather than 150,000. 2. Valuation Date 1: The date when your property was purchased, or revalued. 3. Valuation Date 2: Date for which you would like a new estimate of your property's value. 4. Region: Select region which the property is situated in. If you are not sure which region the property is in, click on the link below to find your region. To the right of the instructions is a "Please note" box with a light blue background. It states: "Please note: The Nationwide House Price Calculator is intended to illustrate general movement in prices only. The calculator is based on the Nationwide House Price Index. Results are based on movements in prices in the regions of the UK rather than in specific towns and cities. The data is based on movements in the price of a typical property in the region, and cannot show movement of differences in months of difference".

# Regression

- ❑ What do all these problems have in common?



# Regression

❑ What do all these problems have in common?

- Continuous **outputs**, we'll call these  $t$

(e.g., a rating: a real number between 0-10, # of followers, house price)

# Regression

- ❑ What do all these problems have in common?
  - Continuous **outputs**, we'll call these  $t$   
  
(e.g., a rating: a real number between 0-10, # of followers, house price)
- ❑ Predicting continuous outputs is called **regression**

# Regression

- ❑ What do all these problems have in common?
  - Continuous **outputs**, we'll call these  $t$   
  
(e.g., a rating: a real number between 0-10, # of followers, house price)
- ❑ Predicting continuous outputs is called **regression**
- ❑ What do I need in order to **predict** these outputs?

# Regression

- ❑ What do all these problems have in common?
  - Continuous **outputs**, we'll call these  $t$   
  
(e.g., a rating: a real number between 0-10, # of followers, house price)
- ❑ Predicting continuous outputs is called **regression**
- ❑ What do I need in order to **predict** these outputs?
  - **Features** (inputs), we'll call these  $x$  (or  $\mathbf{x}$  if vectors)

# Regression

- ❑ What do all these problems have in common?
  - Continuous **outputs**, we'll call these  $t$   
  
(e.g., a rating: a real number between 0-10, # of followers, house price)
- ❑ Predicting continuous outputs is called **regression**
- ❑ What do I need in order to **predict** these outputs?
  - **Features** (inputs), we'll call these  $x$  (or  $\mathbf{x}$  if vectors)
  - **Training examples**, many  $x(i)$  for which  $t(i)$  is known (e.g., many movies for which we know the rating)

# Regression

- ❑ What do all these problems have in common?
  - Continuous **outputs**, we'll call these  $t$   
  
(e.g., a rating: a real number between 0-10, # of followers, house price)
- ❑ Predicting continuous outputs is called **regression**
- ❑ What do I need in order to **predict** these outputs?
  - **Features** (inputs), we'll call these  $x$  (or  $\mathbf{x}$  if vectors)
  - **Training examples**, many  $x(i)$  for which  $t(i)$  is known (e.g., many movies for which we know the rating)
  - A **model**, a function that represents the relationship between  $x$  and  $t$

# Regression

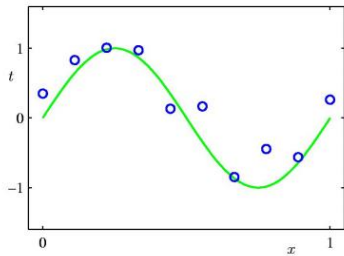
- ❑ What do all these problems have in common?
  - Continuous **outputs**, we'll call these  $t$   
  
(e.g., a rating: a real number between 0-10, # of followers, house price)
- ❑ Predicting continuous outputs is called **regression**
- ❑ What do I need in order to **predict** these outputs?
  - **Features** (inputs), we'll call these  $x$  (or  $\mathbf{x}$  if vectors)
  - **Training examples**, many  $x(i)$  for which  $t(i)$  is known (e.g., many movies for which we know the rating)
  - A **model**, a function that represents the relationship between  $x$  and  $t$
  - A **loss** or a **cost** or an **objective** function, which tells us how well our model approximates the training examples

# Regression

- ❑ What do all these problems have in common?
  - Continuous **outputs**, we'll call these  $t$   
  
(e.g., a rating: a real number between 0-10, # of followers, house price)
- ❑ Predicting continuous outputs is called **regression**
- ❑ What do I need in order to **predict** these outputs?
  - **Features** (inputs), we'll call these  $x$  (or  $\mathbf{x}$  if vectors)
  - **Training examples**, many  $x(i)$  for which  $t(i)$  is known (e.g., many movies for which we know the rating)
  - A **model**, a function that represents the relationship between  $x$  and  $t$
  - A **loss** or a **cost** or an **objective** function, which tells us how well our model approximates the training examples
  - **Optimization**, a way of finding the parameters of our model that minimizes the loss function

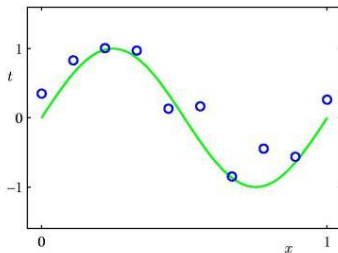


## Simple 1-D regression



- Circles are data points (i.e., training examples) that are given to us

## Simple 1-D regression

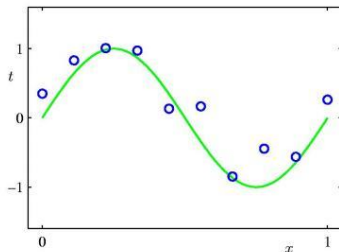


- Circles are data points (i.e., training examples) that are given to us
- The data points are uniform in  $x$ , but may be displaced in

$$t(x) = f(x) + \epsilon$$

with  $\epsilon$  some noise

# Simple 1-D regression



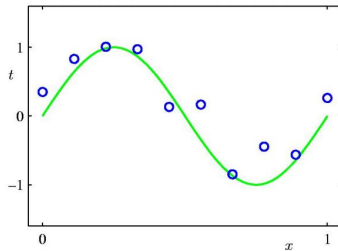
- Circles are data points (i.e., training examples) that are given to us
- The data points are uniform in  $x$ , but may be displaced in

$$t(x) = f(x) + \epsilon$$

with  $\epsilon$  some noise

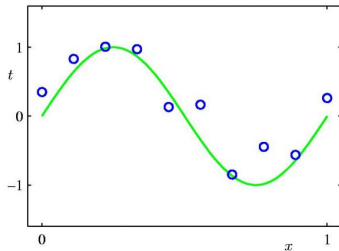
- In **green** is the "true" curve that we don't know
- **Goal**: We want to fit a curve to these points

# Simple 1-D regression



- Key Questions:
  - ▶ How do we parametrize the model?

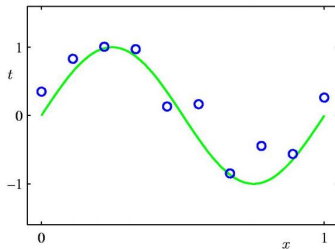
# Simple 1-D regression



- Key Questions:

- ▶ How do we parametrize the **model**?
- ▶ What **loss (objective) function** should we use to judge the fit?

# Simple 1-D regression



- Key Questions:

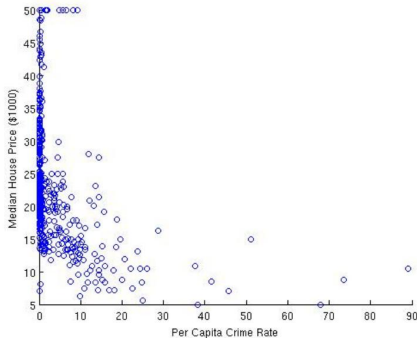
- ▶ How do we parametrize the **model**?
- ▶ What **loss (objective) function** should we use to judge the fit?
- ▶ How do we optimize fit to unseen test data (**generalization**)?

## Example: Boston Housing data

- Estimate median house price in a neighborhood based on neighborhood statistics

## Example: Boston Housing data

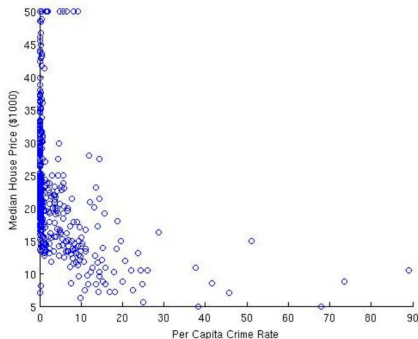
- Estimate median house price in a neighborhood based on neighborhood statistics
- Look at first possible attribute (feature): per capita crime rate





## Example: Boston Housing data

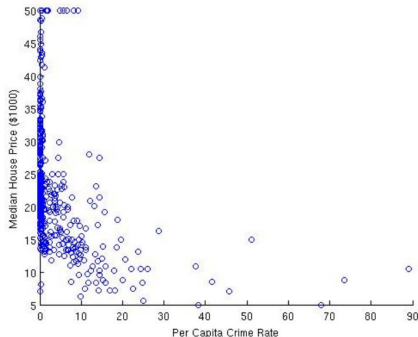
- Estimate median house price in a neighborhood based on neighborhood statistics
- Look at first possible attribute (feature): per capita crime rate



- Use this to predict house prices in other neighborhoods

## Example: Boston Housing data

- Estimate median house price in a neighborhood based on neighborhood statistics
- Look at first possible attribute (feature): per capita crime rate



- Use this to predict house prices in other neighborhoods
- Is this a **good input (attribute) to predict** house prices?

## Represent the Data

- Data is described as pairs  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$ 
  - ▶  $x \in \mathbb{R}$  is the **input feature** (per capita crime rate)
  - ▶  $t \in \mathbb{R}$  is the **target output** (median house price)
  - ▶  $(i)$  simply indicates the training examples (we have  $N$  in this case)

## Represent the Data

- Data is described as pairs  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$ 
  - ▶  $x \in \mathbb{R}$  is the **input feature** (per capita crime rate)
  - ▶  $t \in \mathbb{R}$  is the **target output** (median house price)
  - ▶  $(i)$  simply indicates the training examples (we have  $N$  in this case)
- Here  $t$  is continuous, so this is a **regression problem**

## Represent the Data

- Data is described as pairs  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$ 
  - ▶  $x \in \mathbb{R}$  is the **input feature** (per capita crime rate)
  - ▶  $t \in \mathbb{R}$  is the **target output** (median house price)
  - ▶  $(i)$  simply indicates the training examples (we have  $N$  in this case)
- Here  $t$  is continuous, so this is a **regression problem**
- Model outputs  $y$ , an estimate of  $t$

$$y(x) = w_0 + w_1x$$

# Represent the Data

- Data is described as pairs  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$ 
  - ▶  $x \in \mathbb{R}$  is the **input feature** (per capita crime rate)
  - ▶  $t \in \mathbb{R}$  is the **target output** (median house price)
  - ▶  $(i)$  simply indicates the training examples (we have  $N$  in this case)
- Here  $t$  is continuous, so this is a **regression problem**
- Model outputs  $y$ , an estimate of  $t$

$$y(x) = w_0 + w_1x$$

- What type of **model** did we choose?

## Represent the Data

- Data is described as pairs  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$ 
  - ▶  $x \in \mathbb{R}$  is the **input feature** (per capita crime rate)
  - ▶  $t \in \mathbb{R}$  is the **target output** (median house price)
  - ▶  $(i)$  simply indicates the training examples (we have  $N$  in this case)
- Here  $t$  is continuous, so this is a **regression problem**
- Model outputs  $y$ , an estimate of  $t$

$$y(x) = w_0 + w_1x$$

- What type of **model** did we choose?
- Divide the dataset into training and testing examples

# Represent the Data

- Data is described as pairs  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$ 
  - ▶  $x \in \mathbb{R}$  is the **input feature** (per capita crime rate)
  - ▶  $t \in \mathbb{R}$  is the **target output** (median house price)
  - ▶ <sup>(i)</sup> simply indicates the training examples (we have  $N$  in this case)
- Here  $t$  is continuous, so this is a **regression problem**
- Model outputs  $y$ , an estimate of  $t$

$$y(x) = w_0 + w_1 x$$

- What type of **model** did we choose?
- Divide the dataset into training and testing examples
  - ▶ Use the training examples to construct hypothesis, or function approximator, that maps  $x$  to predicted  $y$



# Represent the Data

- Data is described as pairs  $\mathcal{D} = \{(x^{(1)}, t^{(1)}), \dots, (x^{(N)}, t^{(N)})\}$ 
  - ▶  $x \in \mathbb{R}$  is the **input feature** (per capita crime rate)
  - ▶  $t \in \mathbb{R}$  is the **target output** (median house price)
  - ▶ <sup>(i)</sup> simply indicates the training examples (we have  $N$  in this case)
- Here  $t$  is continuous, so this is a **regression problem**
- Model outputs  $y$ , an estimate of  $t$

$$y(x) = w_0 + w_1x$$

- What type of **model** did we choose?
- Divide the dataset into training and testing examples
  - ▶ Use the training examples to construct hypothesis, or function approximator, that maps  $x$  to predicted  $y$
  - ▶ Evaluate hypothesis on test set

# Noise

- A simple model typically does not exactly fit the data
  - ▶ lack of fit can be considered **noise**

# Noise

- A simple model typically does not exactly fit the data
  - ▶ lack of fit can be considered **noise**
- Sources of noise:
  - ▶ Imprecision in data attributes (**input noise**, e.g., noise in per-capita crime)

# Noise

- A simple model typically does not exactly fit the data
  - ▶ lack of fit can be considered **noise**
- Sources of noise:
  - ▶ Imprecision in data attributes (**input noise**, e.g., noise in per-capita crime)
  - ▶ Errors in data targets (**mis-labeling**, e.g., noise in house prices)

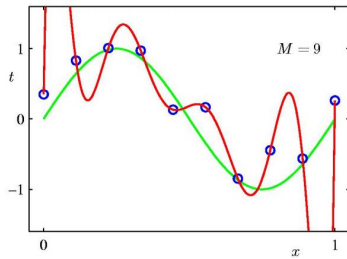
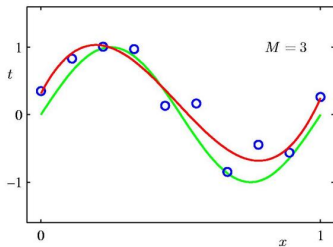
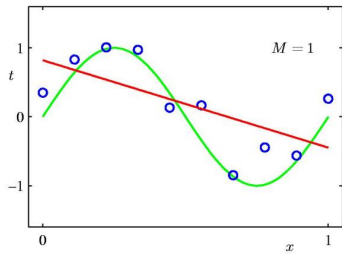
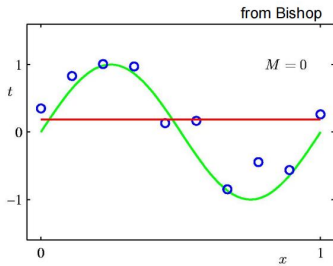
# Noise

- A simple model typically does not exactly fit the data
  - ▶ lack of fit can be considered **noise**
- Sources of noise:
  - ▶ Imprecision in data attributes (**input noise**, e.g., noise in per-capita crime)
  - ▶ Errors in data targets (**mis-labeling**, e.g., noise in house prices)
  - ▶ **Additional attributes** not taken into account by data attributes, affect target values (latent variables). In the example, what else could affect house prices?

# Noise

- A simple model typically does not exactly fit the data
  - ▶ lack of fit can be considered **noise**
- Sources of noise:
  - ▶ Imprecision in data attributes (**input noise**, e.g., noise in per-capita crime)
  - ▶ Errors in data targets (**mis-labeling**, e.g., noise in house prices)
  - ▶ **Additional attributes** not taken into account by data attributes, affect target values (latent variables). In the example, what else could affect house prices?
  - ▶ **Model** may be **too simple** to account for data targets

# Which fit is best?



## A toy example of linear regression

```
1  import torch
2  import torch.nn as nn
3  import numpy as np
4  import matplotlib.pyplot as plt
5
6
7  # Hyper-parameters
8  input_size = 1
9  output_size = 1
10 num_epochs = 60
11 learning_rate = 0.001
12
13 i_iter = np.array([])
14 loss_iter = np.array([])
15
16 # Toy dataset
17 x_train = np.array([[3.3], [4.4], [5.5], [6.71], [6.93], [4.168],
18                    [9.779], [6.182], [7.59], [2.167], [7.042],
19                    [10.791], [5.313], [7.997], [3.1]]], dtype=np.float32)
20
21 y_train = np.array([[1.7], [2.76], [2.09], [3.19], [1.694], [1.573],
22                    [3.366], [2.596], [2.53], [1.221], [2.827],
23                    [3.465], [1.65], [2.904], [1.3]]], dtype=np.float32)
24
25 # Linear regression model
26 model = nn.Linear(input_size, output_size)
```



## A toy example of linear regression

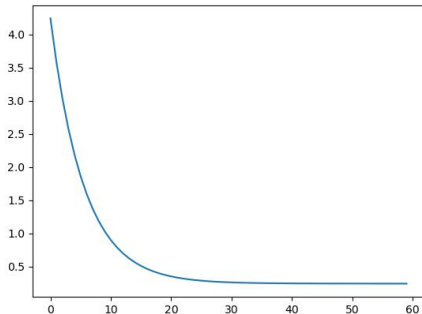
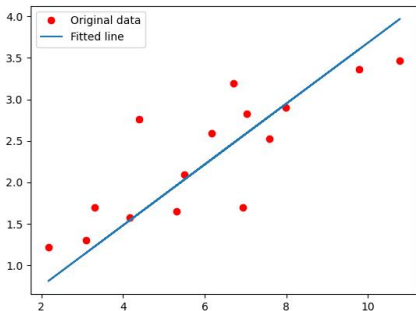
```
28 # Loss and optimizer
29 criterion = nn.MSELoss()
30 optimizer = torch.optim.SGD(model.parameters(), lr=learning_rate)
31
32 # Train the model
33 for epoch in range(num_epochs):
34     # Convert numpy arrays to torch tensors
35     inputs = torch.from_numpy(x_train)
36     targets = torch.from_numpy(y_train)
37
38     # Forward pass
39     outputs = model(inputs)
40     loss = criterion(outputs, targets)
41
42     # Backward and optimize
43     optimizer.zero_grad()
44     loss.backward()
45     optimizer.step()
46
47     i_iter = np.append(i_iter, epoch)
48     loss_iter = np.append(loss_iter, loss.item())
49
50     if (epoch+1) % 5 == 0:
51         print ('Epoch [{}/{}], Loss: {:.4f}'.format(epoch+1, num_epochs, loss.item()))
52
```

## A toy example of linear regression

```
53 # Plot the graph
54 predicted = model(torch.from_numpy(x_train)).detach().numpy()
55
56 plt.figure()
57 plt.plot(x_train, y_train, 'ro', label='Original data')
58 plt.plot(x_train, predicted, label='Fitted line')
59 plt.legend()
60
61 plt.figure()
62 plt.plot(i_iter, loss_iter)
63 plt.show()
64
65 # Save the model checkpoint
66 torch.save(model.state_dict(), 'model.ckpt')
```

# A toy example of linear regression

```
53 # Plot the graph
54 predicted = model(torch.from_numpy(x_train)).detach().numpy()
55
56 plt.figure()
57 plt.plot(x_train, y_train, 'ro', label='Original data')
58 plt.plot(x_train, predicted, label='Fitted line')
59 plt.legend()
60
61 plt.figure()
62 plt.plot(i_iter, loss_iter)
63 plt.show()
```



## Summary of Regression

### □ Regression: to predict continuous outputs $t$

- ▶ Consider proper **features** (inputs):  $x$  (or  $\mathbf{x}$  if vectors)
- ▶ **Training examples**, many  $x(i)$  for which  $t(i)$  is known (labeled)
- ▶ A **model**, a function that represents the relationship between  $x$  and  $t$

$$y = f(x, w)$$

- ▶ A **loss** or a **cost** or an **objective** function, which tells us how well our model approximates the training examples
- ▶ **Optimization**, a way of finding the parameters  $w$  of our model that minimizes the loss function