What is Machine Learning? (Part I)

Neil D. Lawrence

School of Computer Science, University of Manchester, U.K.
(from August 1st Sheffield Institute of Translational Neuroscience
and University of Sheffield, U.K.)
Machine Learning and CogSci Summer School, Pula, Sardinia

6th May 2010

Outline

Outline

What is Machine Learning?

Equipping Computers with Human Like Capabilities.

- Endow computers with the ability to “learn” from “data”.
- Present data from sensors, the internet, experiments.
- Expect computer to make “sensible” decisions.
- Traditionally categorized as:
  - Supervised learning: classification, regression.
  - Unsupervised learning: dimensionality reduction, clustering.
  - Reinforcement learning: learning from delayed feedback. Planning. Difficult stuff!

What is Machine Learning?

Equipping Computers with Human Like Capabilities.

- Endow computers with the ability to "learn" from "data".
- Present data from sensors, the internet, experiments.
- Expect computer to make "sensible" decisions.
- Traditionally categorized as:
  - Supervised learning: classification, regression.
  - Unsupervised learning: dimensionality reduction, clustering.
  - Reinforcement learning: learning from delayed feedback. Planning. Difficult stuff!

What is Machine Learning?

Equipping Computers with Human Like Capabilities.

- Endow computers with the ability to “learn” from “data”.
- Present data from sensors, the internet, experiments.
- Expect computer to make “sensible” decisions.
- Traditionally categorized as:
  - Supervised learning: classification, regression.
  - Unsupervised learning: dimensionality reduction, clustering.
  - Reinforcement learning: learning from delayed feedback. Planning. Difficult stuff!

What is Machine Learning?

Equipping Computers with Human Like Capabilities.

- Endow computers with the ability to “learn” from “data”.
- Present data from sensors, the internet, experiments.
- Expect computer to make “sensible” decisions.
- Traditionally categorized as:
  - Supervised learning: classification, regression.
  - Unsupervised learning: dimensionality reduction, clustering.
  - Reinforcement learning: learning from delayed feedback. Planning. Difficult stuff!

What is Machine Learning?

Equipping Computers with Human Like Capabilities.

- Endow computers with the ability to "learn" from "data".
- Present data from sensors, the internet, experiments.
- Expect computer to make "sensible" decisions.
- Traditionally categorized as:
  - Supervised learning: classification, regression.
  - Unsupervised learning: dimensionality reduction, clustering.
  - Reinforcement learning: learning from delayed feedback. Planning. Difficult stuff!

What is Machine Learning?

Equipping Computers with Human Like Capabilities.

- Endow computers with the ability to “learn” from “data”.
- Present data from sensors, the internet, experiments.
- Expect computer to make “sensible” decisions.
- Traditionally categorized as:
  - Supervised learning: classification, regression.
  - Unsupervised learning: dimensionality reduction, clustering.
  - Reinforcement learning: learning from delayed feedback. Planning. Difficult stuff!

What is Machine Learning?

Equipping Computers with Human Like Capabilities.

- Endow computers with the ability to "learn" from "data".
- Present data from sensors, the internet, experiments.
- Expect computer to make "sensible" decisions.
- Traditionally categorized as:
  - Supervised learning: classification, regression.
  - Unsupervised learning: dimensionality reduction, clustering.
  - Reinforcement learning: learning from delayed feedback. Planning. Difficult stuff!

History of Machine Learning (personal)

Rosenblatt to Vapnik

- Early connectionist research focused on models of the brain.
- Rosenblatt's perceptron (Rosenblatt, 1962) based on simple model of a neuron (McCulloch and Pitts, 1943) and a learning algorithm.
- Later machine learning research focused on theoretical foundations of such models and their capacity to learn (Vapnik, 1998).
- Personal view: machine learning benefited greatly by incorporating ideas from psychology, but not being afraid to incorporate rigorous theory.

History of Machine Learning (personal)

Rosenblatt to Vapnik

- Early connectionist research focused on models of the brain.
- Rosenblatt's perceptron (Rosenblatt, 1962) based on simple model of a neuron (McCulloch and Pitts, 1943) and a learning algorithm.
- Later machine learning research focused on theoretical foundations of such models and their capacity to learn (Vapnik, 1998).
- Personal view: machine learning benefited greatly by incorporating ideas from psychology, but not being afraid to incorporate rigorous theory.

History of Machine Learning (personal)

Rosenblatt to Vapnik

- Early connectionist research focused on models of the brain.
- Rosenblatt's perceptron (Rosenblatt, 1962) based on simple model of a neuron (McCulloch and Pitts, 1943) and a learning algorithm.
- Later machine learning research focused on theoretical foundations of such models and their capacity to learn (Vapnik, 1998).
- Personal view: machine learning benefited greatly by incorporating ideas from psychology, but not being afraid to incorporate rigorous theory.

History of Machine Learning (personal)

Rosenblatt to Vapnik

- Early connectionist research focused on models of the brain.
- Rosenblatt's perceptron (Rosenblatt, 1962) based on simple model of a neuron (McCulloch and Pitts, 1943) and a learning algorithm.
- Later machine learning research focused on theoretical foundations of such models and their capacity to learn (Vapnik, 1998).
- Personal view: machine learning benefited greatly by incorporating ideas from psychology, but not being afraid to incorporate rigorous theory.

Machine Learning Today

An extension of statistics?

- Early machine learning viewed with scepticism by statisticians.
- Modern machine learning and statistics interact to both communities benefits.
- Personal view: statistics and machine learning are fundamentally different. Statistics aims to provide a human with the tools to analyze data. Machine learning wants to replace the human in the processing of data.

Machine Learning Today

An extension of statistics?

- Early machine learning viewed with scepticism by statisticians.
- Modern machine learning and statistics interact to both communities benefits.
- Personal view: statistics and machine learning are fundamentally different. Statistics aims to provide a human with the tools to analyze data. Machine learning wants to replace the human in the processing of data.

- Early machine learning viewed with scepticism by statisticians.
- Modern machine learning and statistics interact to both communities benefits.
- Personal view: statistics and machine learning are fundamentally different. Statistics aims to provide a human with the tools to analyze data. Machine learning wants to replace the human in the processing of data.

Machine Learning Today

Mathematics and Bumblebees

- For the moment the two overlap strongly. But they are not the same field!
- This summer school reflects that. ML has a lot still to learn from CogSci.
- Mathematical formalisms of a problem are helpful, but they can hide facts: i.e. the fallacy that "aerodynamically a bumble bee can't fly". Clearly a limitation of the model rather than fact.
- Mathematical foundations are still very important though: they help us understand the capabilities of our algorithms.
- But we mustn't restrict our ambitions to the limitations of current mathematical formalisms. That is where humans give inspiration.

- For the moment the two overlap strongly. But they are not the same field!
- This summer school reflects that. ML has a lot still to learn from CogSci.
- Mathematical formalisms of a problem are helpful, but they can hide facts: i.e. the fallacy that "aerodynamically a bumble bee can't fly". Clearly a limitation of the model rather than fact.
- Mathematical foundations are still very important though: they help us understand the capabilities of our algorithms.
- But we mustn't restrict our ambitions to the limitations of current mathematical formalisms. That is where humans give inspiration.

- For the moment the two overlap strongly. But they are not the same field!
- This summer school reflects that. ML has a lot still to learn from CogSci.
- Mathematical formalisms of a problem are helpful, but they can hide facts: i.e. the fallacy that "aerodynamically a bumble bee can't fly". Clearly a limitation of the model rather than fact.
- Mathematical foundations are still very important though: they help us understand the capabilities of our algorithms.
- But we mustn't restrict our ambitions to the limitations of current mathematical formalisms. That is where humans give inspiration.

- For the moment the two overlap strongly. But they are not the same field!
- This summer school reflects that. ML has a lot still to learn from CogSci.
- Mathematical formalisms of a problem are helpful, but they can hide facts: i.e. the fallacy that "aerodynamically a bumble bee can't fly". Clearly a limitation of the model rather than fact.
- Mathematical foundations are still very important though: they help us understand the capabilities of our algorithms.
- But we mustn't restrict our ambitions to the limitations of current mathematical formalisms. That is where humans give inspiration.

- For the moment the two overlap strongly. But they are not the same field!
- This summer school reflects that. ML has a lot still to learn from CogSci.
- Mathematical formalisms of a problem are helpful, but they can hide facts: i.e. the fallacy that "aerodynamically a bumble bee can't fly". Clearly a limitation of the model rather than fact.
- Mathematical foundations are still very important though: they help us understand the capabilities of our algorithms.
- But we mustn't restrict our ambitions to the limitations of current mathematical formalisms. That is where humans give inspiration.

- Early statistics had great success with the idea of statistical proof.
  - Question: I computed the mean of these two tables of numbers (a statistic). They are different. Does this "prove" anything?
  - Answer: it depends on how the numbers are generated, how many there are and how big the difference. Randomization is important.
- Hypothesis testing: questions you can ask about your data are quite limiting.
- This can have the affect of limiting science too.
- Many successes: crop fertilization, clinical trials, brewing, polling.
- Many open questions: e.g. causality.

Early 20th Century Statistics

- Many statisticians were Edwardian English gentleman.

Figure: William Sealy Gosset in 1908

Outline

Supervised Learning

Outline

Classification

- We are given data set containing "inputs", $\mathbf{X}$, and "targets", $\mathbf{y}$.
- Each data point consists of an input vector $\mathbf{x}_{i,:}$ and a class label, y_i.
- For binary classification assume y_i should be either 1 (yes) or -1 (no).
- Input vector can be thought of as features.

Classification Examples

- Classifying hand written digits from binary images (automatic zip code reading).
- Detecting faces in images (e.g. digital cameras).
- Who a detected face belongs to (e.g. Picasa).
- Classifying type of cancer given gene expression data.
- Categorization of document types (different types of news article on the internet).

The Perceptron

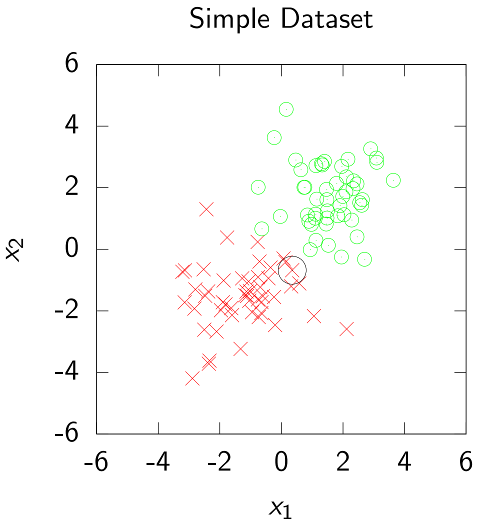
- Developed in 1957 by Rosenblatt.
- Take a data point at, $\mathbf{x}_i$.
- Predict it belongs to a class, $y_i = 1$ if $\sum_j w_j \mathbf{x}_{i,j} + b > 0$ i.e. $\mathbf{w}^\top \mathbf{x}_i + b > 0$. Otherwise assume $y_i = -1$.

Perceptron-like Algorithm

1. Select a random data point i.
2. Ensure i is correctly classified by setting $\mathbf{w} = y_i \mathbf{x}_i$.
   - i.e. $\text{sign}\left(\mathbf{w}^\top \mathbf{x}_{i,:}\right) = \text{sign}\left(y_i \mathbf{x}_{i,:}^\top \mathbf{x}_{i,:}\right) = \text{sign}\left(y_i\right) = y_i$
3. Iterate: increment k and select a misclassified point, i.
4. Set $\mathbf{w} \leftarrow \mathbf{w} + \eta y_i \mathbf{x}_{i,:}$.
   - If η is large enough this will guarantee this point becomes correctly classified.
5. Repeat until there are no misclassified points..

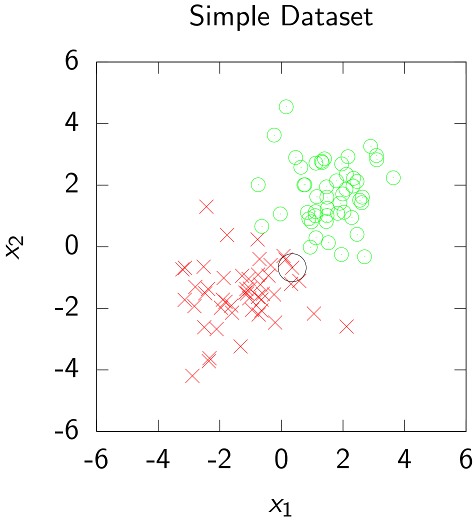
Perceptron Algorithm

- Iteration 1 data no 29



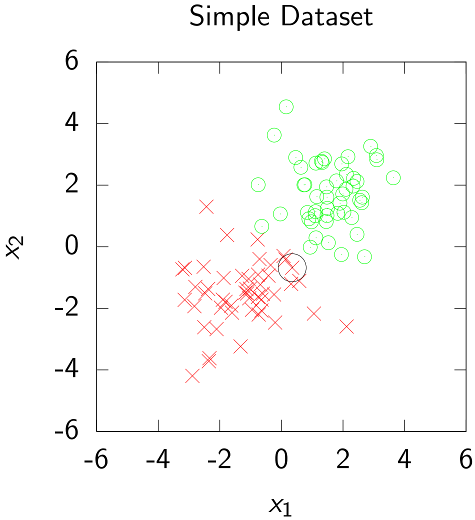
Perceptron Algorithm

- Iteration 1 data no 29
- $w_1 = 0,\ w_2 = 0$



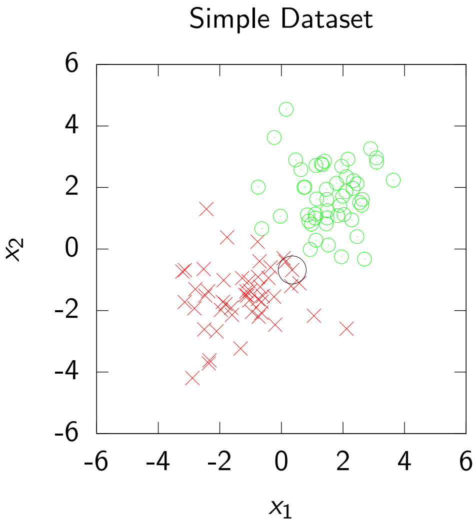
Perceptron Algorithm

- Iteration 1 data no 29
- $w_1 = 0, \; w_2 = 0$
- First Iteration



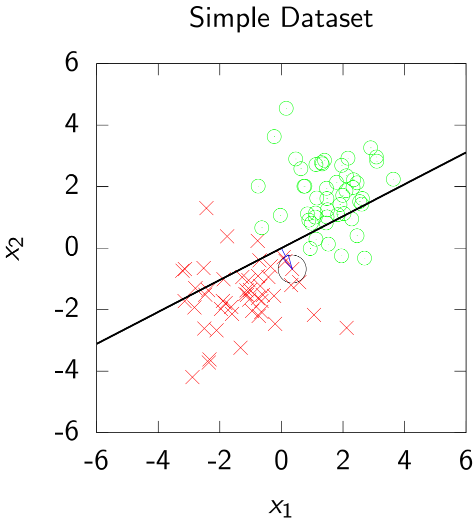
Perceptron Algorithm

- Iteration 1 data no 29
- $w_1 = 0, \ w_2 = 0$
- First Iteration
- Set weight vector to data point.



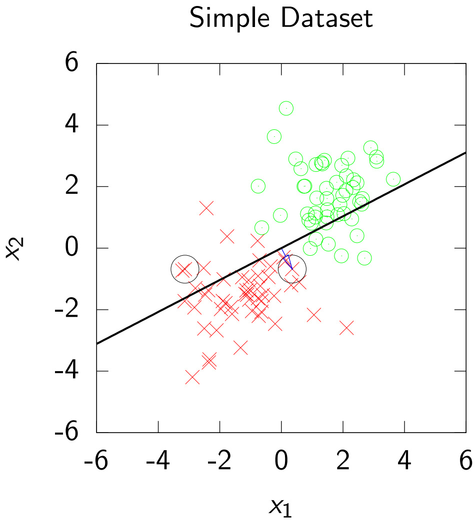
Perceptron Algorithm

- Iteration 1 data no 29
- $w_1 = 0,\ w_2 = 0$
- First Iteration
- Set weight vector to data point.
- $\mathbf{w} = y_{29}\mathbf{x}_{29,:}$



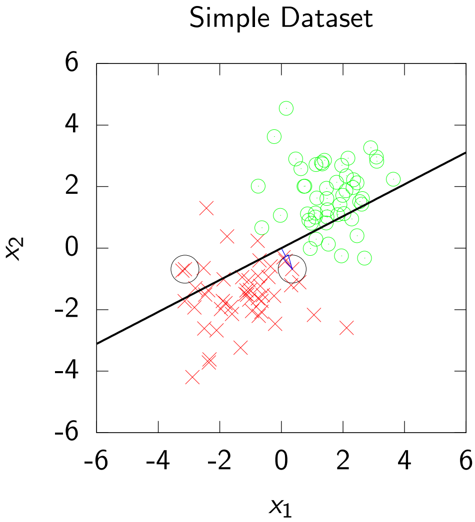
Perceptron Algorithm

- Iteration 1 data no 29
- $w_1 = 0, \; w_2 = 0$
- First Iteration
- Set weight vector to data point.
- $\mathbf{w} = y_{29}\mathbf{x}_{29,:}$
- Select new incorrectly classified data point.



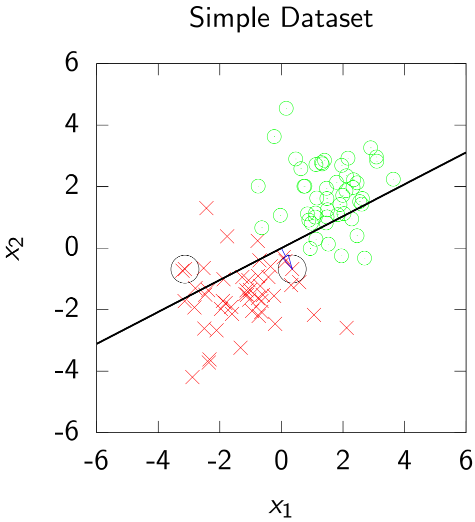
Perceptron Algorithm

- Iteration 2 data no 16



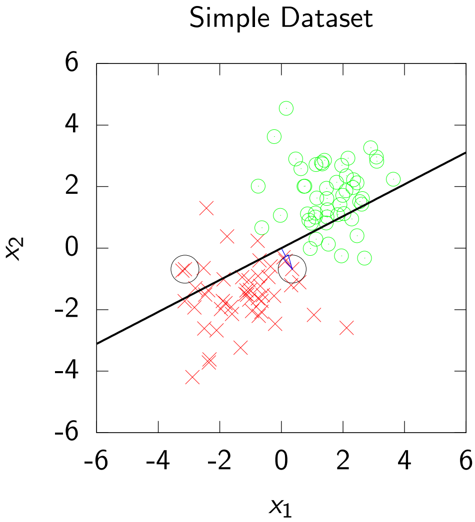
Perceptron Algorithm

- Iteration 2 data no 16
- $w_1 = 0.3519$,
  $w_2 = -0.6787$



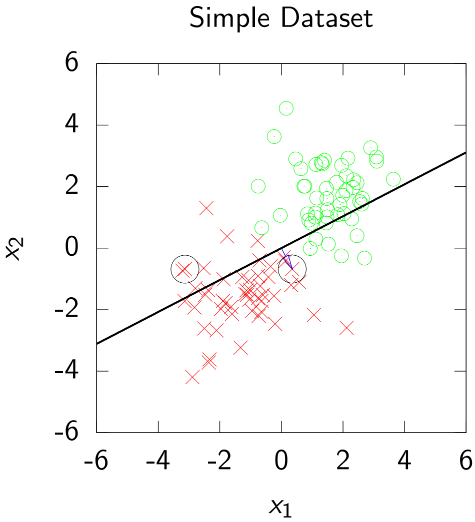
Perceptron Algorithm

- Iteration 2 data no 16
- $w_1 = 0.3519$,
  $w_2 = -0.6787$
- Incorrect classification



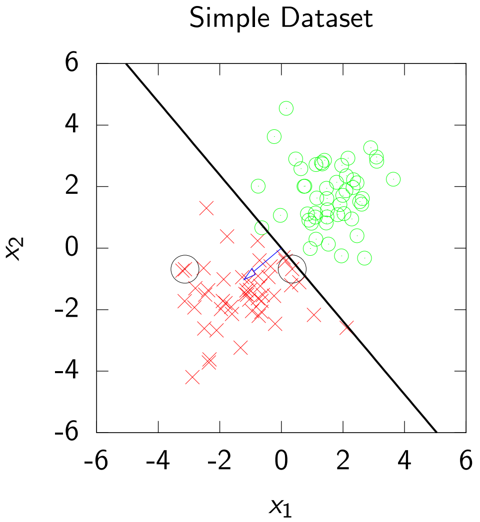
Perceptron Algorithm

- Iteration 2 data no 16
- $w_1 = 0.3519$,
  $w_2 = -0.6787$
- Incorrect classification
- Adjust weight vector with new data point.



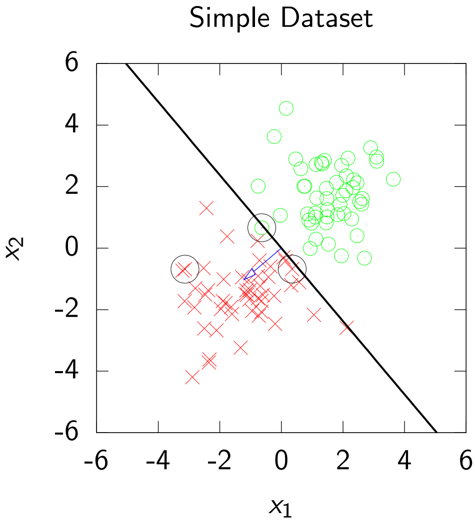
Perceptron Algorithm

- Iteration 2 data no 16
- $w_1 = 0.3519$,
  $w_2 = -0.6787$
- Incorrect classification
- Adjust weight vector with new data point.
- $\mathbf{w} \leftarrow \mathbf{w} + \eta y_{16} \mathbf{x}_{16,:}$



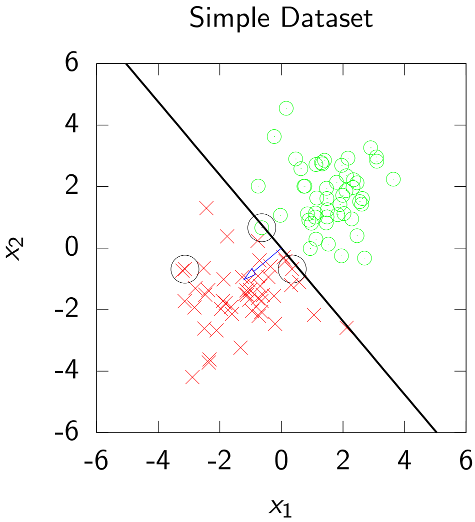
Perceptron Algorithm

- Iteration 2 data no 16
- $w_1 = 0.3519$,
  $w_2 = -0.6787$
- Incorrect classification
- Adjust weight vector with new data point.
- $\mathbf{w} \leftarrow \mathbf{w} + \eta y_{16} \mathbf{x}_{16,:}$
- Select new incorrectly classified data point.



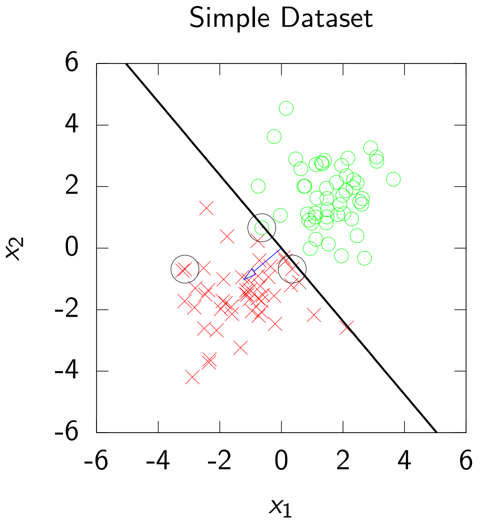
Perceptron Algorithm

- Iteration 3 data no 58



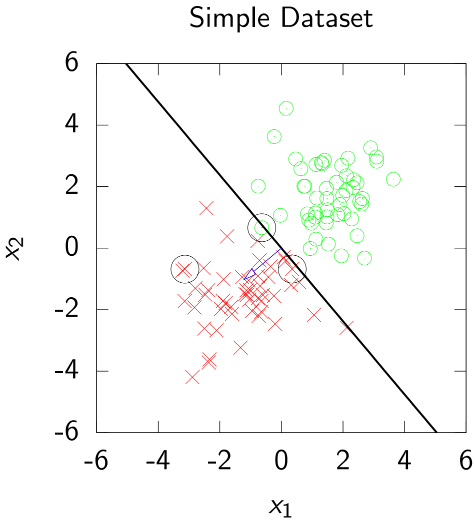
Perceptron Algorithm

- Iteration 3 data no 58
- $w_1 = -1.2143$,
  $w_2 = -1.0217$



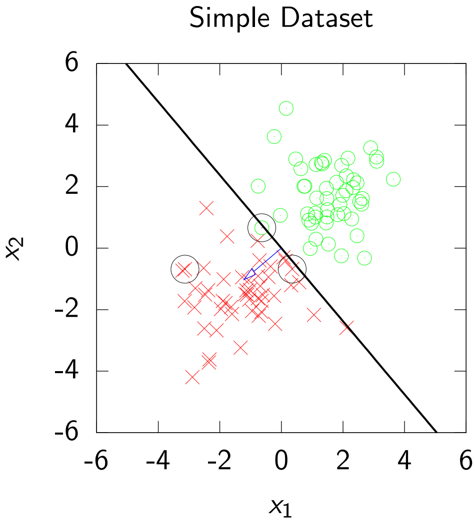
Perceptron Algorithm

- Iteration 3 data no 58
- $w_1 = -1.2143$,
  $w_2 = -1.0217$
- Incorrect classification



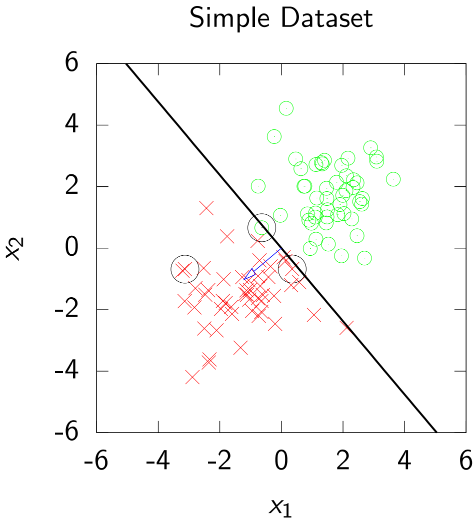
Perceptron Algorithm

- Iteration 3 data no 58
- $w_1 = -1.2143$, $w_2 = -1.0217$
- Incorrect classification
- Adjust weight vector with new data point.



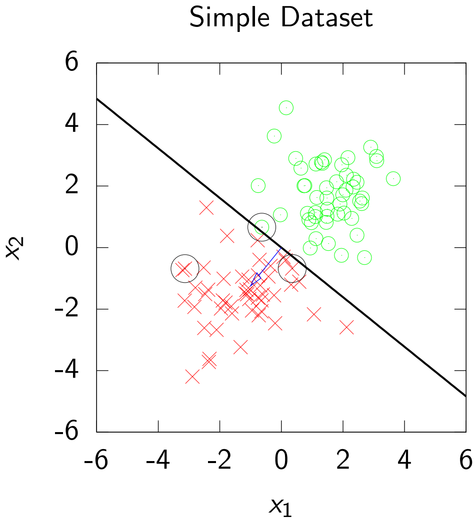
Perceptron Algorithm

- Iteration 3 data no 58
- $w_1 = -1.2143$, $w_2 = -1.0217$
- Incorrect classification
- Adjust weight vector with new data point.
- $\mathbf{w} \leftarrow \mathbf{w} + \eta y_{58} \mathbf{x}_{58,:}$



Perceptron Algorithm

- Iteration 3 data no 58
- $w_1 = -1.2143$,
  $w_2 = -1.0217$
- Incorrect classification
- Adjust weight vector with new data point.
- $\mathbf{w} \leftarrow \mathbf{w} + \eta y_{58} \mathbf{x}_{58,:}$
- All data correctly classified.



Outline

Regression Examples

- Predict a real value, y_i given some inputs $\mathbf{x}_i$.
- Predict quality of meat given spectral measurements (Tecator data).
- Radiocarbon dating, the C14 calibration curve: predict age given quantity of C14 isotope.
- Predict quality of different Go or Backgammon moves given expert rated training data.

Linear Regression

Is there an equivalent learning rule for regression?

- Predict a real value y given x.
- We can also construct a learning rule for regression.
  - Define our prediction
$$f(x) = mx + c.$$
  - Define an error
$$\Delta y_i = y_i - f(x_i).$$

Updating Bias/Intercept

- c represents bias. Add portion of error to bias.

$$c \rightarrow c + \eta \Delta y_i.$$

$$\Delta y_i = y_i - mx_i - c.$$

1. For $+$ve error, c and therefore $f(x_i)$ become larger and error magnitude becomes smaller.
2. For -ve error, c and therefore $f(x_i)$ become smaller and error magnitude becomes smaller.

Updating Slope

- m represents Slope. Add portion of error $\times$ input to slope.

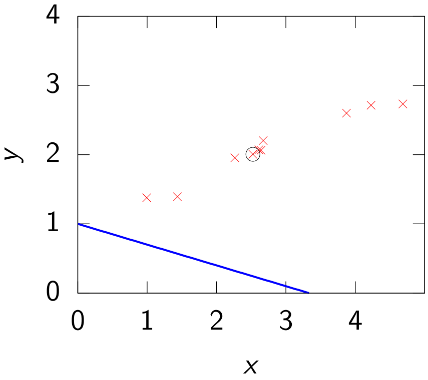
$$m \to m + \eta \Delta y_i x_i.$$

$$\Delta y_i = y_i - m x_i - c.$$

1. For $+$ve error and $+$ve input, m becomes larger and $f(x_i)$ becomes larger: error magnitude becomes smaller.
2. For $+$ve error and -ve input, m becomes smaller and $f(x_i)$ becomes larger: error magnitude becomes smaller.
3. For -ve error and -ve slope, m becomes larger and $f(x_i)$ becomes smaller: error magnitude becomes smaller.
4. For -ve error and $+$ve input, m becomes smaller and $f(x_i)$ becomes smaller: error magnitude becomes smaller.

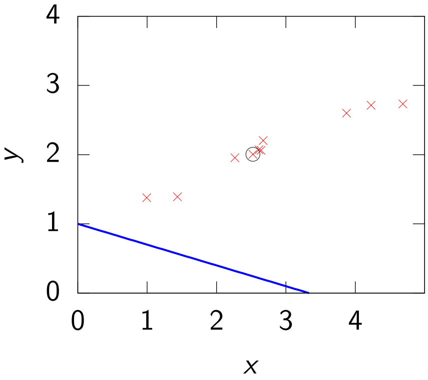
Linear Regression Example

- Iteration 1 $\hat{m} = -0.3$ $\hat{c} = 1$



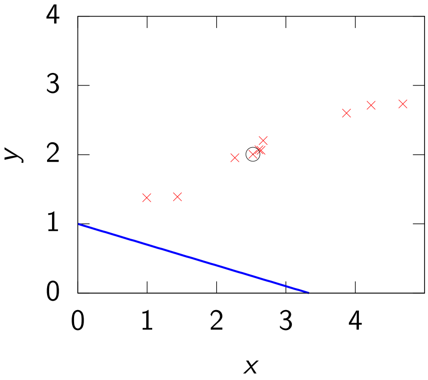
Linear Regression Example

- Iteration 1 $\hat{m} = -0.3$ $\hat{c} = 1$
  - Present data point 4



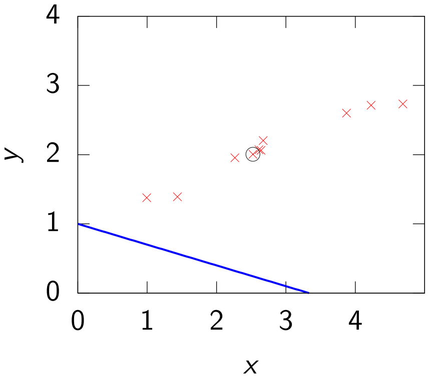
Linear Regression Example

- Iteration 1 $\hat{m} = -0.3$ $\hat{c} = 1$
  - Present data point 4
  - $\Delta y_4 = (y_4 - \hat{m}x_4 - \hat{c})$



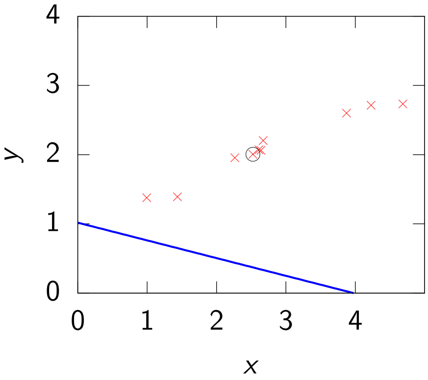
Linear Regression Example

- Iteration 1 $\hat{m} = -0.3$ $\hat{c} = 1$
  - Present data point 4
  - $\Delta y_4 = (y_4 - \hat{m}x_4 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_4 \Delta y_4$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_4$



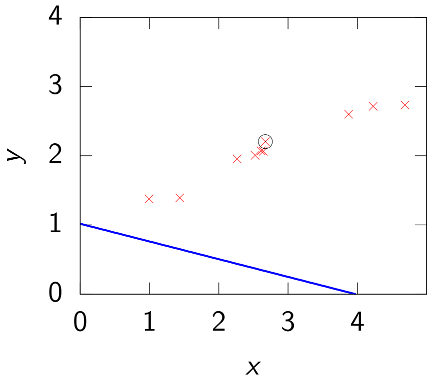
Linear Regression Example

- Iteration 1 $\hat{m} = -0.3$ $\hat{c} = 1$
  - Present data point 4
  - $\Delta y_4 = (y_4 - \hat{m}x_4 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_4 \Delta y_4$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_4$
- Updated values
  $\hat{m} = -0.25593$ $\hat{c} = 1.0175$



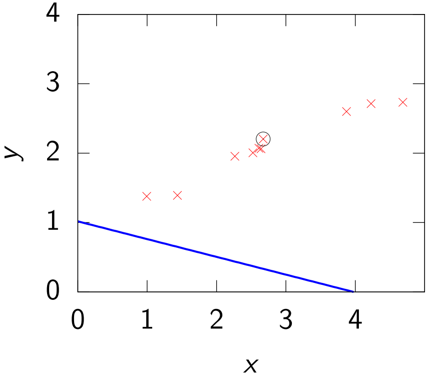
Linear Regression Example

- Iteration 2
  $\hat{m} = -0.25593$
  $\hat{c} = 1.0175$



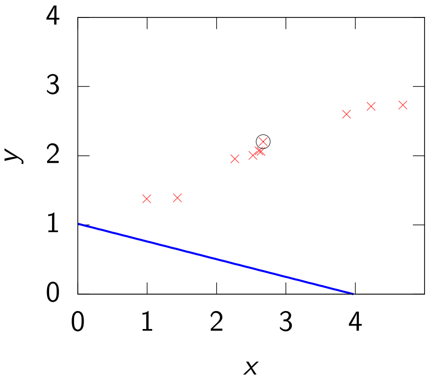
Linear Regression Example

- Iteration 2
  $\hat{m} = -0.25593$
  $\hat{c} = 1.0175$
  - Present data point 7



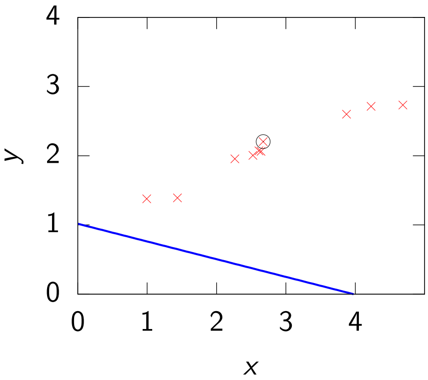
Linear Regression Example

- Iteration 2
  $\hat{m} = -0.25593$
  $\hat{c} = 1.0175$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m}x_7 - \hat{c})$



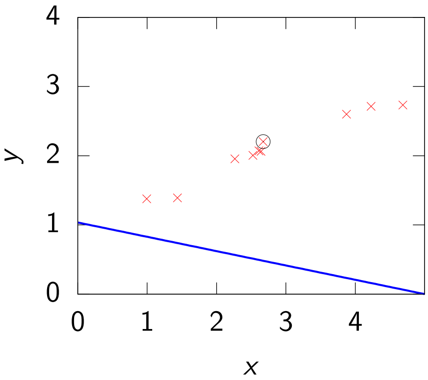
Linear Regression Example

- Iteration 2
  $\hat{m} = -0.25593$
  $\hat{c} = 1.0175$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m} x_7 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_7 \Delta y_7$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_7$



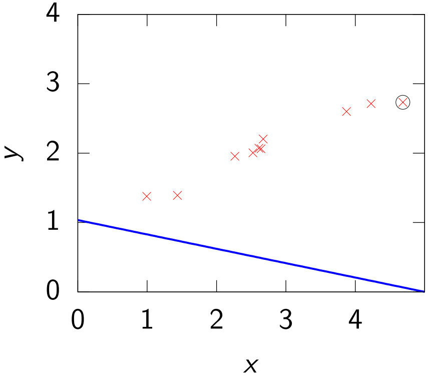
Linear Regression Example

- Iteration 2
  $\hat{m} = -0.25593$
  $\hat{c} = 1.0175$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m} x_7 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_7 \Delta y_7$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_7$
- Updated values
  $\hat{m} = -0.20693$ $\hat{c} = 1.0358$



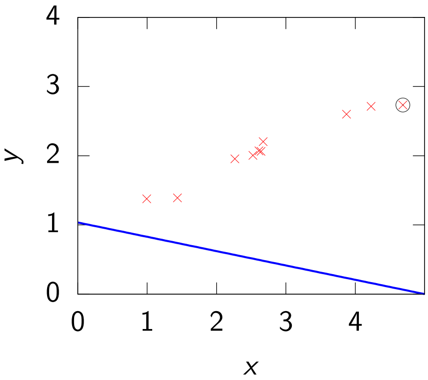
Linear Regression Example

- Iteration 3
  $\hat{m} = -0.20693$
  $\hat{c} = 1.0358$



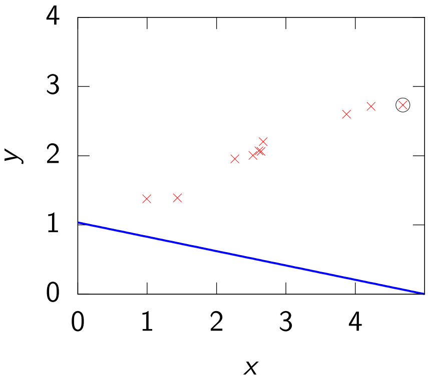
Linear Regression Example

- Iteration 3
  $\hat{m} = -0.20693$
  $\hat{c} = 1.0358$
  - Present data point 10



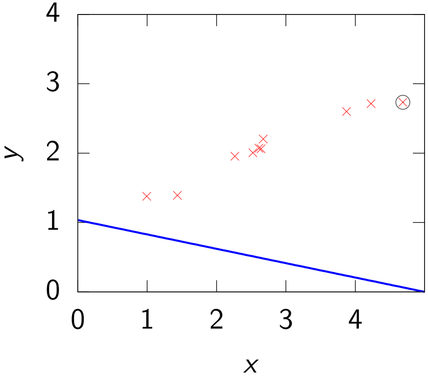
Linear Regression Example

- Iteration 3
  $\hat{m} = -0.20693$
  $\hat{c} = 1.0358$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$



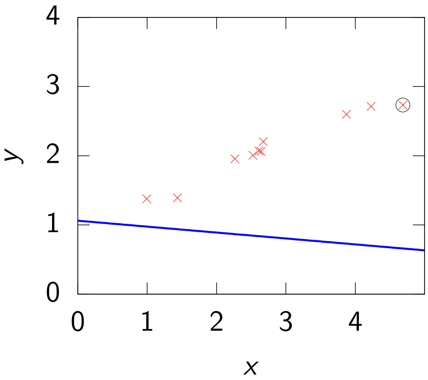
Linear Regression Example

- Iteration 3
  $\hat{m} = -0.20693$
  $\hat{c} = 1.0358$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m} x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$



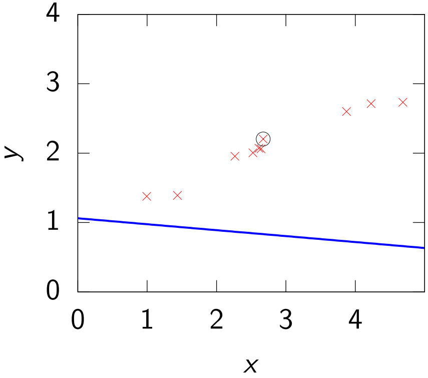
Linear Regression Example

- Iteration 3
  $\hat{m} = -0.20693$
  $\hat{c} = 1.0358$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$
- Updated values
  $\hat{m} = -0.085591$ $\hat{c} = 1.0617$



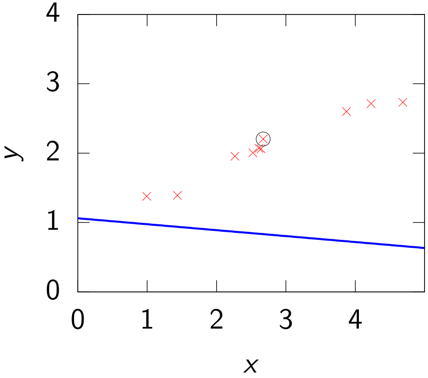
Linear Regression Example

- Iteration 4
  $\hat{m} = -0.085591$
  $\hat{c} = 1.0617$



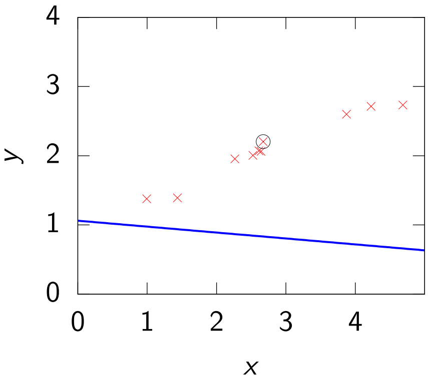
Linear Regression Example

- Iteration 4
  $\hat{m} = -0.085591$
  $\hat{c} = 1.0617$
  - Present data point 7



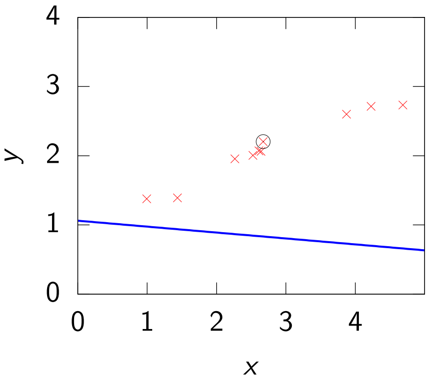
Linear Regression Example

- Iteration 4
  $\hat{m} = -0.085591$
  $\hat{c} = 1.0617$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m}x_7 - \hat{c})$



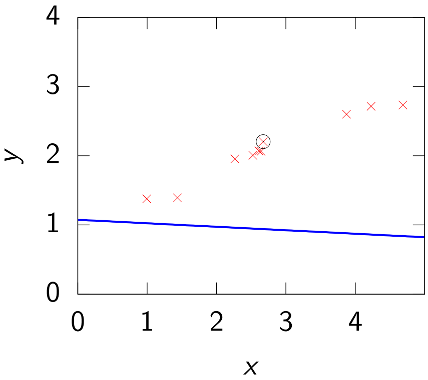
Linear Regression Example

- Iteration 4
  $\hat{m} = -0.085591$
  $\hat{c} = 1.0617$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m}x_7 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_7 \Delta y_7$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_7$



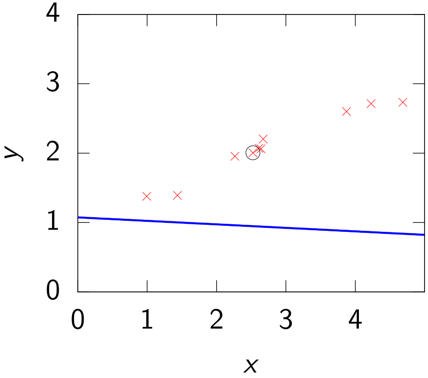
Linear Regression Example

- Iteration 4
  $\hat{m} = -0.085591$
  $\hat{c} = 1.0617$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m}x_7 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_7 \Delta y_7$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_7$
- Updated values
  $\hat{m} = -0.050355$ $\hat{c} = 1.0749$



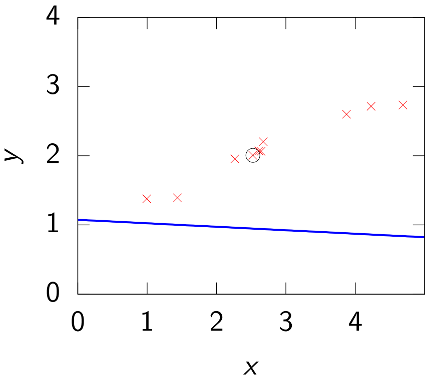
Linear Regression Example

- Iteration 5
  $\hat{m} = -0.050355$
  $\hat{c} = 1.0749$



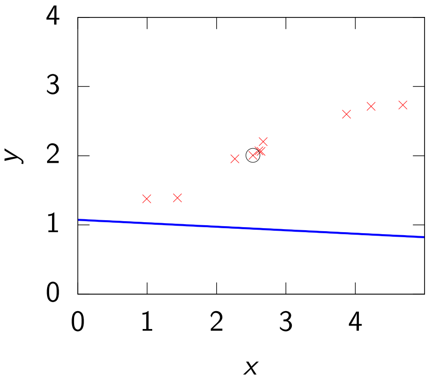
Linear Regression Example

- Iteration 5
  $\hat{m} = -0.050355$
  $\hat{c} = 1.0749$
  - Present data point 4



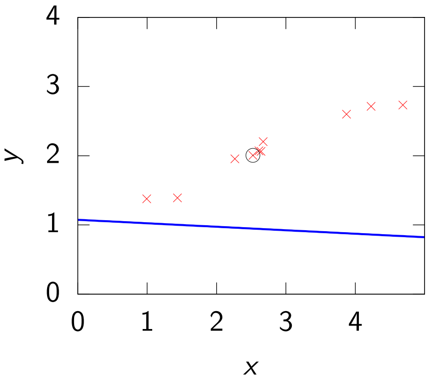
Linear Regression Example

- Iteration 5
  $\hat{m} = -0.050355$
  $\hat{c} = 1.0749$
  - Present data point 4
  - $\Delta y_4 = (y_4 - \hat{m}x_4 - \hat{c})$



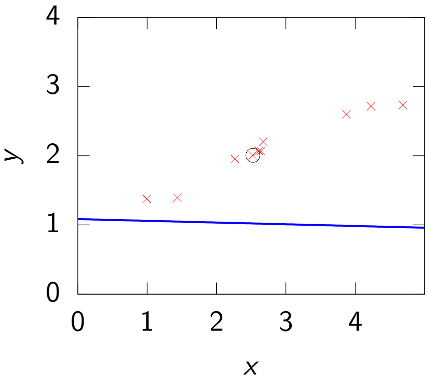
Linear Regression Example

- Iteration 5
  $\hat{m} = -0.050355$
  $\hat{c} = 1.0749$
  - Present data point 4
  - $\Delta y_4 = (y_4 - \hat{m} x_4 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_4 \Delta y_4$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_4$



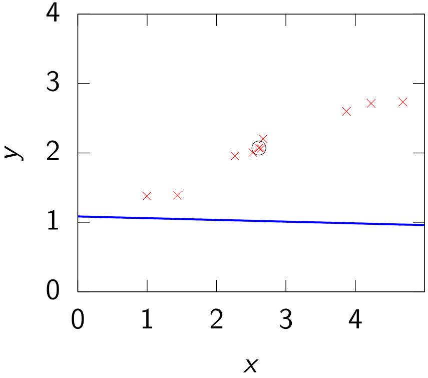
Linear Regression Example

- Iteration 5
  $\hat{m} = -0.050355$
  $\hat{c} = 1.0749$
  - Present data point 4
  - $\Delta y_4 = (y_4 - \hat{m}x_4 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_4 \Delta y_4$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_4$
- Updated values
  $\hat{m} = -0.024925$ $\hat{c} = 1.0849$



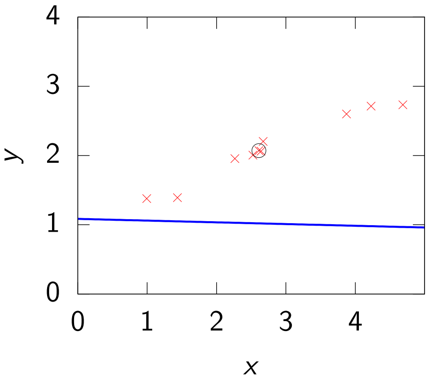
Linear Regression Example

- Iteration 6
  $\hat{m} = -0.024925$
  $\hat{c} = 1.0849$



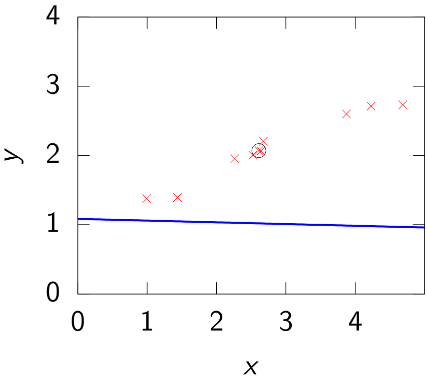
Linear Regression Example

- Iteration 6
  $\hat{m} = -0.024925$
  $\hat{c} = 1.0849$
  - Present data point 5



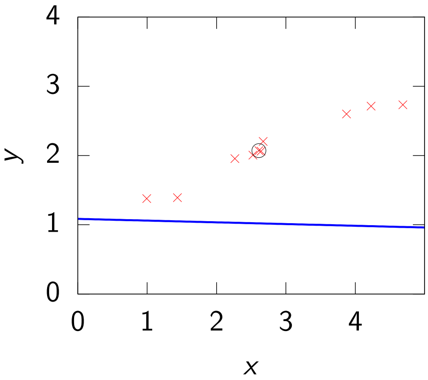
Linear Regression Example

- Iteration 6
  $\hat{m} = -0.024925$
  $\hat{c} = 1.0849$
  - Present data point 5
  - $\Delta y_5 = (y_5 - \hat{m}x_5 - \hat{c})$



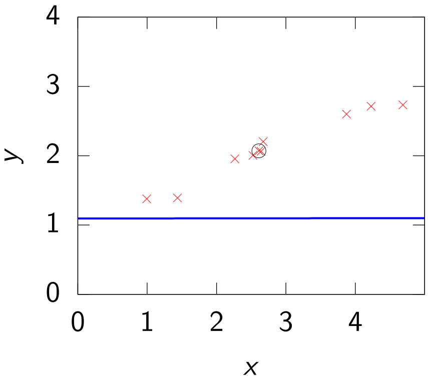
Linear Regression Example

- Iteration 6
  $\hat{m} = -0.024925$
  $\hat{c} = 1.0849$
  - Present data point 5
  - $\Delta y_5 = (y_5 - \hat{m}x_5 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_5 \Delta y_5$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_5$



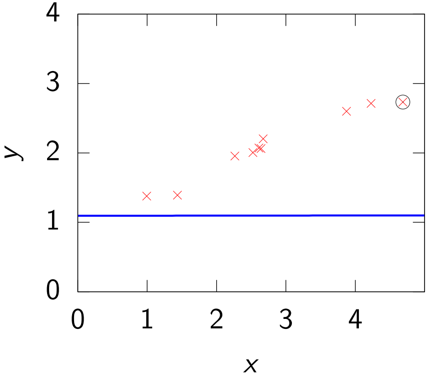
Linear Regression Example

- Iteration 6
  $\hat{m} = -0.024925$
  $\hat{c} = 1.0849$
  - Present data point 5
  - $\Delta y_5 = (y_5 - \hat{m}x_5 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_5 \Delta y_5$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_5$
- Updated values
  $\hat{m} = 0.00098511$ $\hat{c} = 1.0949$



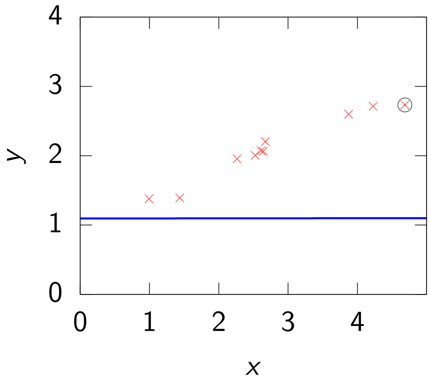
Linear Regression Example

- Iteration 7
  $\hat{m} = 0.00098511$
  $\hat{c} = 1.0949$



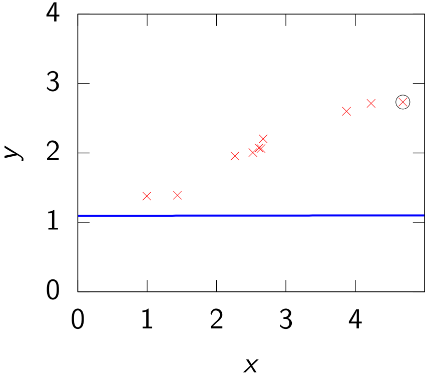
Linear Regression Example

- Iteration 7
  $\hat{m} = 0.00098511$
  $\hat{c} = 1.0949$
    - Present data point 10



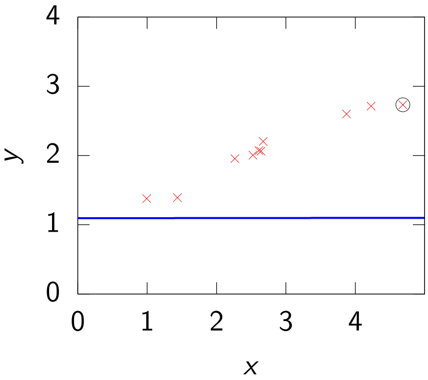
Linear Regression Example

- Iteration 7
  $\hat{m} = 0.00098511$
  $\hat{c} = 1.0949$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$



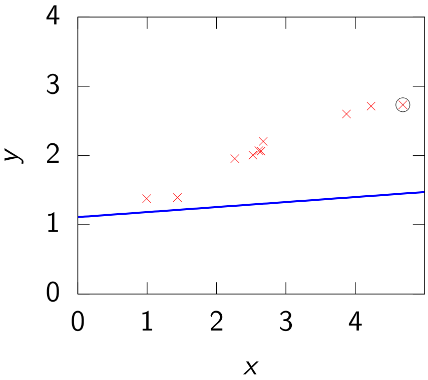
Linear Regression Example

- Iteration 7
  $\hat{m} = 0.00098511$
  $\hat{c} = 1.0949$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$



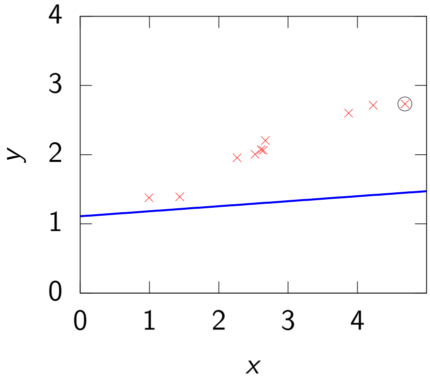
Linear Regression Example

- Iteration 7
  $\hat{m} = 0.00098511$
  $\hat{c} = 1.0949$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$
- Updated values
  $\hat{m} = 0.072529$ $\hat{c} = 1.1101$



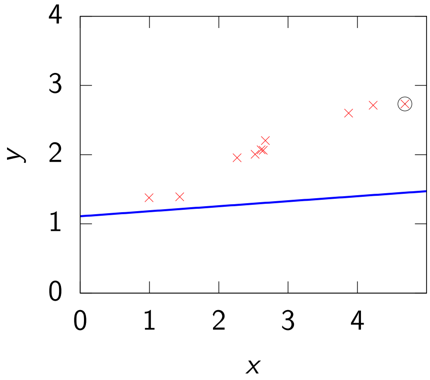
Linear Regression Example

- Iteration 8 $\hat{m} = 0.072529$ $\hat{c} = 1.1101$



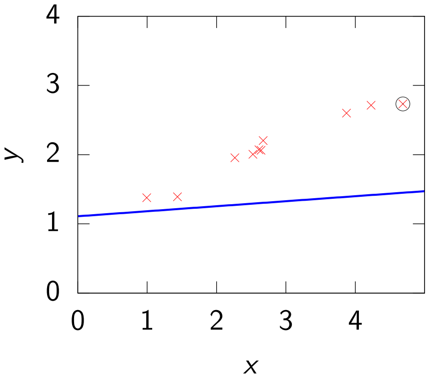
Linear Regression Example

- Iteration 8 $\hat{m} = 0.072529$ $\hat{c} = 1.1101$
  - Present data point 10



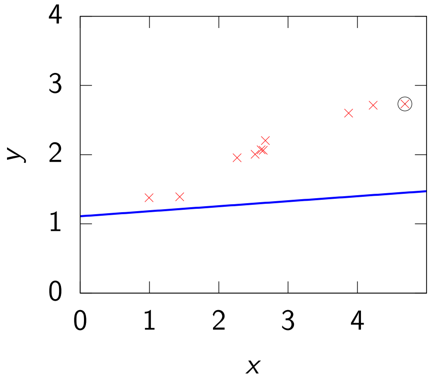
Linear Regression Example

- Iteration 8 $\hat{m} = 0.072529$ $\hat{c} = 1.1101$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$



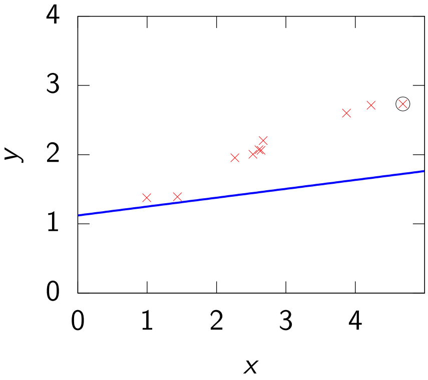
Linear Regression Example

- Iteration 8 $\hat{m} = 0.072529$ $\hat{c} = 1.1101$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$



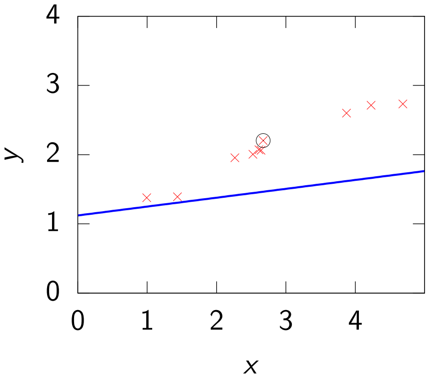
Linear Regression Example

- Iteration 8 $\hat{m} = 0.072529$ $\hat{c} = 1.1101$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$
- Updated values
  $\hat{m} = 0.1282$ $\hat{c} = 1.122$



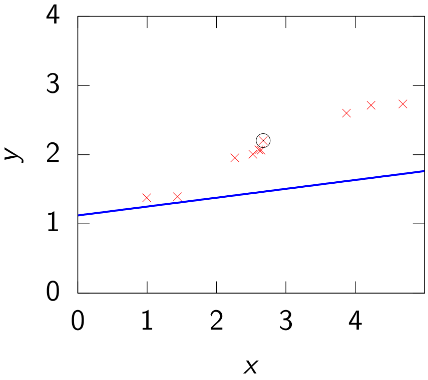
Linear Regression Example

- Iteration 9 $\hat{m} = 0.1282$ $\hat{c} = 1.122$



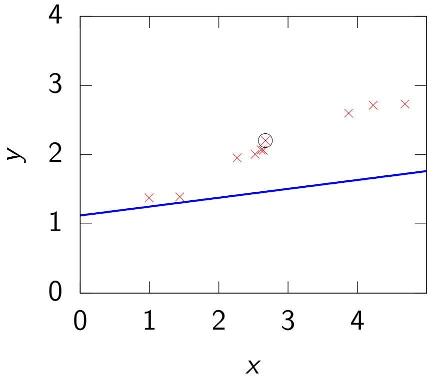
Linear Regression Example

- Iteration 9 $\hat{m} = 0.1282$ $\hat{c} = 1.122$
  - Present data point 7



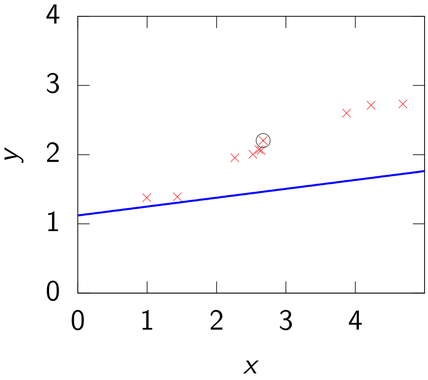
Linear Regression Example

- Iteration 9 $\hat{m} = 0.1282$ $\hat{c} = 1.122$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m}x_7 - \hat{c})$



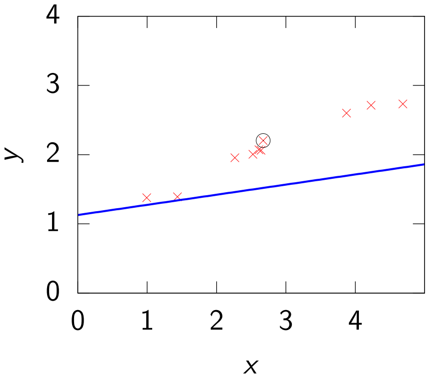
Linear Regression Example

- Iteration 9 $\hat{m} = 0.1282$ $\hat{c} = 1.122$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m}x_7 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_7 \Delta y_7$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_7$



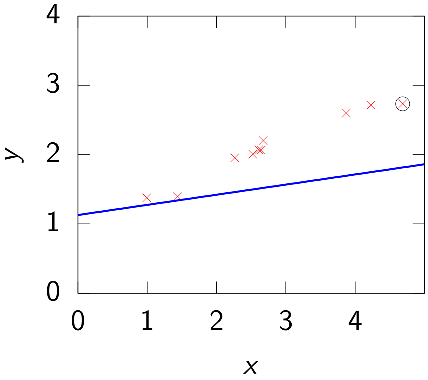
Linear Regression Example

- Iteration 9 $\hat{m} = 0.1282$ $\hat{c} = 1.122$
  - Present data point 7
  - $\Delta y_7 = (y_7 - \hat{m}x_7 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_7 \Delta y_7$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_7$
- Updated values
  $\hat{m} = 0.14634$ $\hat{c} = 1.1288$



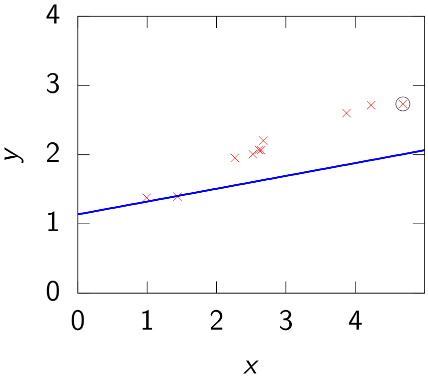
Linear Regression Example

- Iteration 10 $\hat{m} = 0.14634$ $\hat{c} = 1.1288$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m} x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$



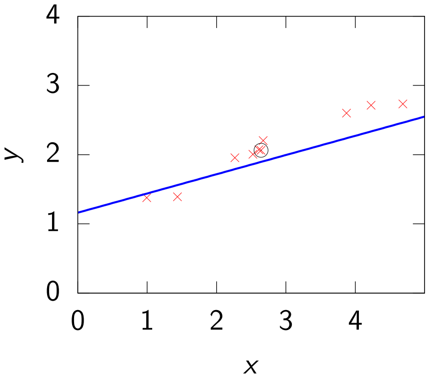
Linear Regression Example

- Iteration 10 $\hat{m} = 0.14634$ $\hat{c} = 1.1288$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$
- Updated values
  $\hat{m} = 0.18547$ $\hat{c} = 1.1372$



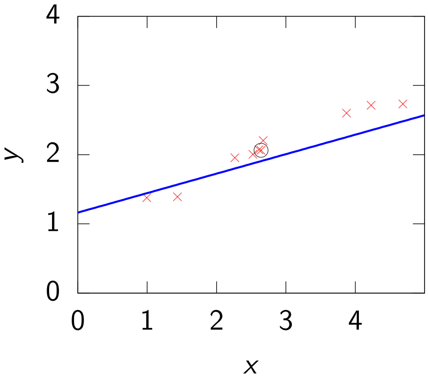
Linear Regression Example

- Iteration 20 $\hat{m} = 0.27764$ $\hat{c} = 1.1621$
  - Present data point 6
  - $\Delta y_6 = (y_6 - \hat{m}x_6 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_6 \Delta y_6$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_6$



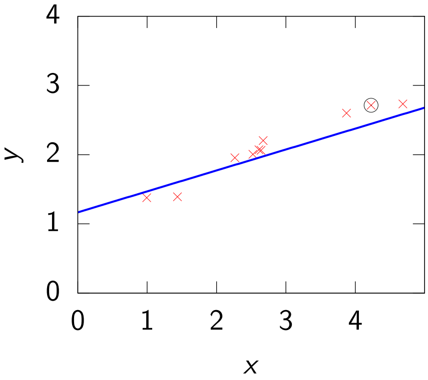
Linear Regression Example

- Iteration 20 $\hat{m} = 0.27764$ $\hat{c} = 1.1621$
  - Present data point 6
  - $\Delta y_6 = (y_6 - \hat{m}x_6 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_6 \Delta y_6$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_6$
- Updated values
  $\hat{m} = 0.28135$ $\hat{c} = 1.1635$



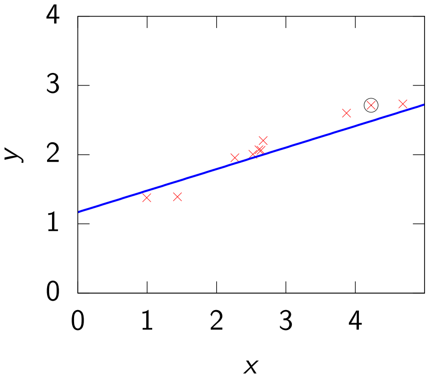
Linear Regression Example

- Iteration 30 $\hat{m} = 0.30249$ $\hat{c} = 1.1673$
  - Present data point 9
  - $\Delta y_9 = (y_9 - \hat{m}x_9 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_9 \Delta y_9$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_9$



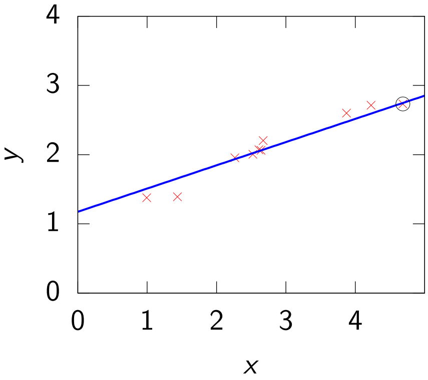
Linear Regression Example

- Iteration 30 $\hat{m} = 0.30249$ $\hat{c} = 1.1673$
  - Present data point 9
  - $\Delta y_9 = (y_9 - \hat{m}x_9 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_9 \Delta y_9$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_9$
- Updated values
  $\hat{m} = 0.31119$ $\hat{c} = 1.1693$



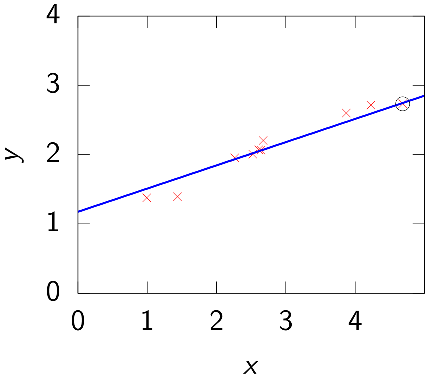
Linear Regression Example

- Iteration 40 $\hat{m} = 0.33551$ $\hat{c} = 1.1754$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$



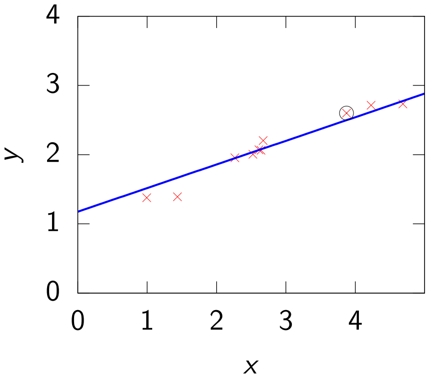
Linear Regression Example

- Iteration 40 $\hat{m} = 0.33551$ $\hat{c} = 1.1754$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m} x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$
- Updated values
  $\hat{m} = 0.33503$ $\hat{c} = 1.1753$



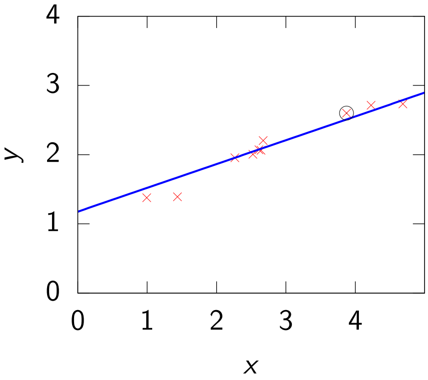
Linear Regression Example

- Iteration 50 $\hat{m} = 0.34126$ $\hat{c} = 1.1763$
  - Present data point 8
  - $\Delta y_8 = (y_8 - \hat{m}x_8 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_8 \Delta y_8$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_8$



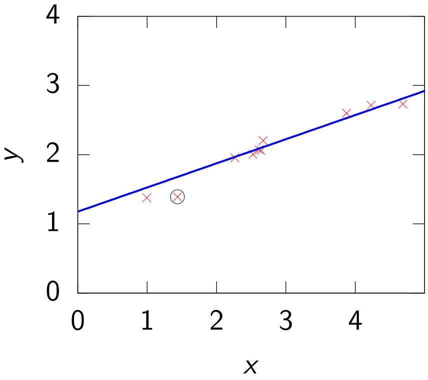
Linear Regression Example

- Iteration 50 $\hat{m} = 0.34126$ $\hat{c} = 1.1763$
  - Present data point 8
  - $\Delta y_8 = (y_8 - \hat{m}x_8 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_8 \Delta y_8$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_8$
- Updated values
  $\hat{m} = 0.3439$ $\hat{c} = 1.177$



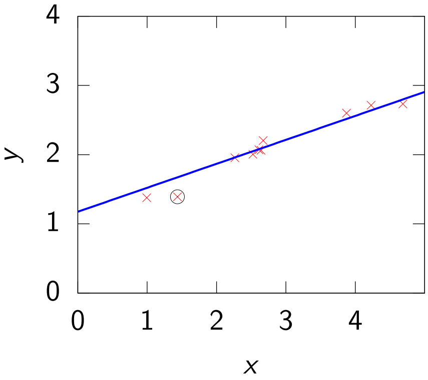
Linear Regression Example

- Iteration 60 $\hat{m} = 0.34877$ $\hat{c} = 1.1775$
  - Present data point 2
  - $\Delta y_2 = (y_2 - \hat{m}x_2 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_2 \Delta y_2$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_2$



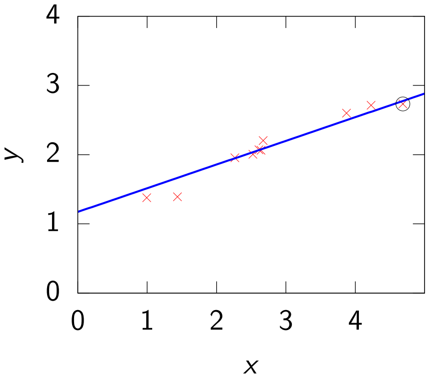
Linear Regression Example

- Iteration 60 $\hat{m} = 0.34877$ $\hat{c} = 1.1775$
  - Present data point 2
  - $\Delta y_2 = (y_2 - \hat{m}x_2 - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_2 \Delta y_2$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_2$
- Updated values
  $\hat{m} = 0.34621$ $\hat{c} = 1.1757$



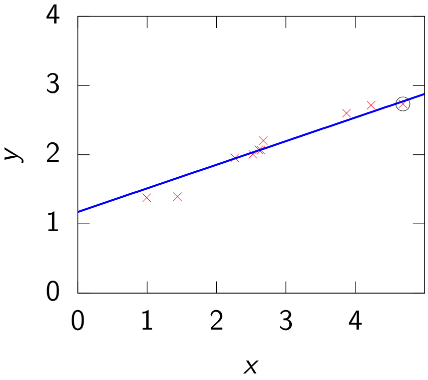
Linear Regression Example

- Iteration 70 $\hat{m} = 0.34207$ $\hat{c} = 1.1734$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$



Linear Regression Example

- Iteration 70 $\hat{m} = 0.34207$ $\hat{c} = 1.1734$
  - Present data point 10
  - $\Delta y_{10} = (y_{10} - \hat{m}x_{10} - \hat{c})$
  - Adjust $\hat{m}$ and $\hat{c}$
    $\hat{m} \leftarrow \hat{m} + \eta x_{10} \Delta y_{10}$
    $\hat{c} \leftarrow \hat{c} + \eta \Delta y_{10}$
- Updated values
  $\hat{m} = 0.34088$ $\hat{c} = 1.1732$



- Problem with Linear Regression—$\mathbf{x}$ may not be linearly related to $\mathbf{y}$.
- Potential solution: create a feature space: define $\phi(\mathbf{x})$ where $\phi(\cdot)$ is a nonlinear function of $\mathbf{x}$.
- Model for target is a linear combination of these nonlinear functions

$$f(\mathbf{x}) = \sum_{j=1}^{K} w_j \phi_j(\mathbf{x}) \tag{1}$$

Quadratic Basis

- Basis functions can be global. E.g. quadratic basis:

$$[1, x, x^2]$$

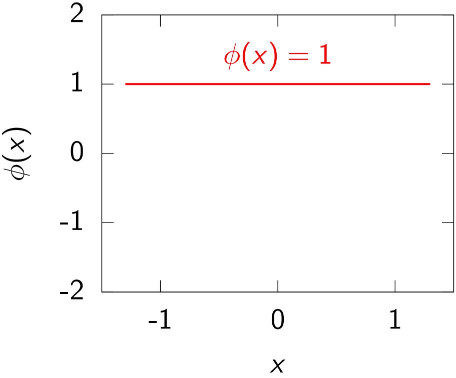


Figure: A quadratic basis.

Quadratic Basis

- Basis functions can be global. E.g. quadratic basis:

$$[1, x, x^2]$$

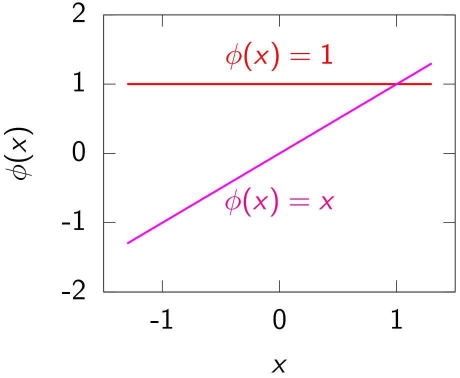


Figure: A quadratic basis.

Quadratic Basis

- Basis functions can be global. E.g. quadratic basis:

$$[1, x, x^2]$$

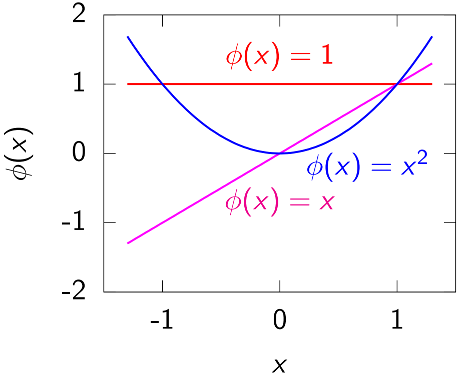


Figure: A quadratic basis.

Functions Derived from Quadratic Basis

$$f(x) = w_1 + w_2 x + w_3 x^2$$

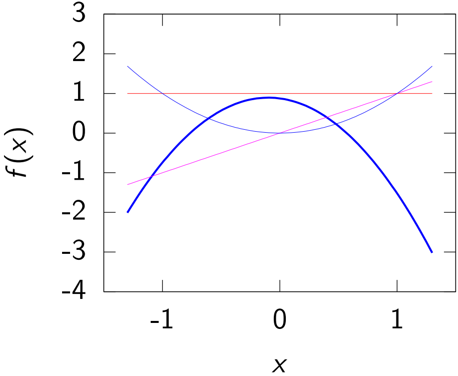


Figure: Function from quadratic basis with weights $w_1 = 0.87466$, $w_2 = -0.38835$, $w_3 = -2.0058$.

Functions Derived from Quadratic Basis

$$f(x) = w_1 + w_2 x + w_3 x^2$$

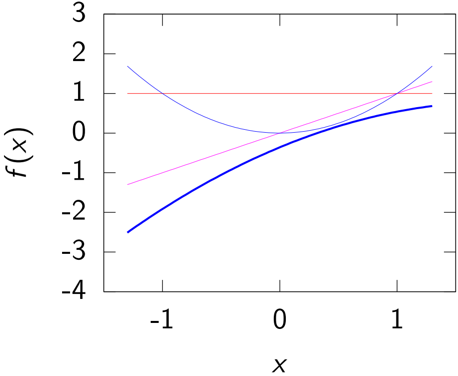


Figure: Function from quadratic basis with weights $w_1 = -0.35908$, $w_2 = 1.2274$, $w_3 = -0.32825$.

Functions Derived from Quadratic Basis

$$f(x) = w_1 + w_2 x + w_3 x^2$$

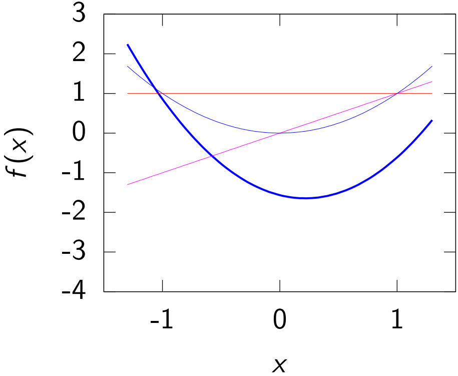


Figure: Function from quadratic basis with weights $w_1 = -1.5638$, $w_2 = -0.73577$, $w_3 = 1.6861$.

Radial Basis Functions

- Or they can be local. E.g. radial (or Gaussian) basis $\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{\ell^2}\right)$

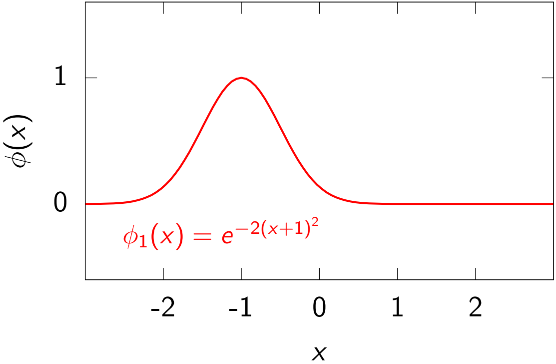


Figure: Radial basis functions.

Radial Basis Functions

- Or they can be local. E.g. radial (or Gaussian) basis $\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{\ell^2}\right)$

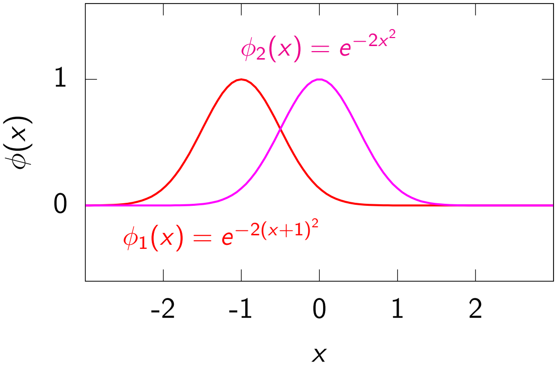


Figure: Radial basis functions.

Radial Basis Functions

- Or they can be local. E.g. radial (or Gaussian) basis $\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{\ell^2}\right)$

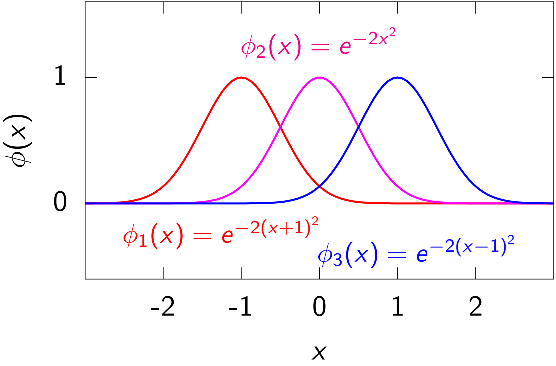


Figure: Radial basis functions.

Functions Derived from Radial Basis

$$f(x) = w_1 e^{-2(x+1)^2} + w_2 e^{-2x^2} + w_3 e^{-2(x-1)^2}$$

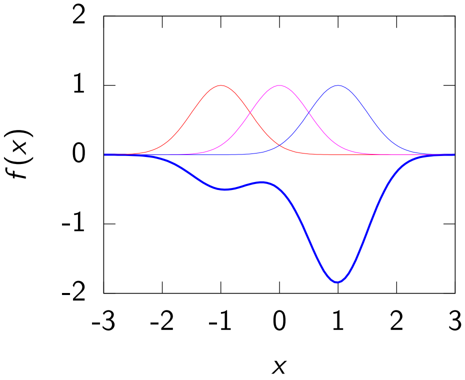


Figure: Function from radial basis with weights $w_1 = -0.47518$, $w_2 = -0.18924$, $w_3 = -1.8183$.

Functions Derived from Radial Basis

$$f(x) = w_1 e^{-2(x+1)^2} + w_2 e^{-2x^2} + w_3 e^{-2(x-1)^2}$$

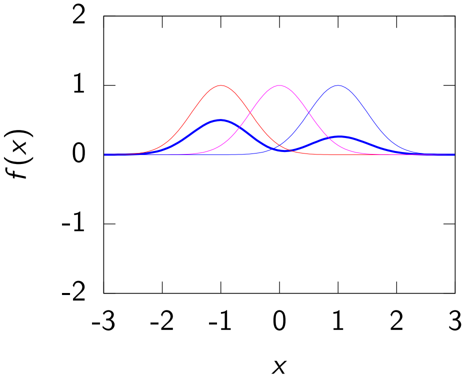


Figure: Function from radial basis with weights $w_1 = 0.50596$, $w_2 = -0.046315$, $w_3 = 0.26813$.

Functions Derived from Radial Basis

$$f(x) = w_1 e^{-2(x+1)^2} + w_2 e^{-2x^2} + w_3 e^{-2(x-1)^2}$$

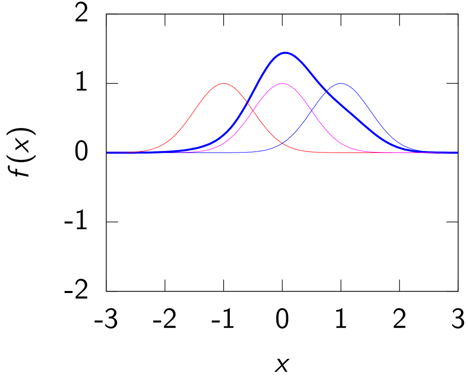
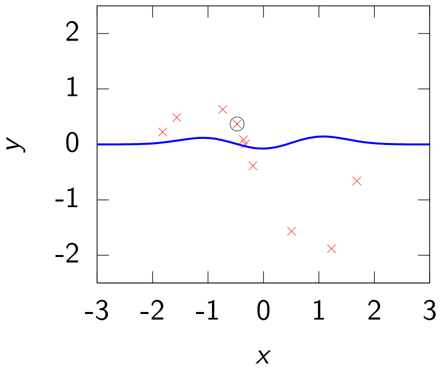


Figure: Function from radial basis with weights $w_1 = 0.07179$, $w_2 = 1.3591$, $w_3 = 0.50604$.

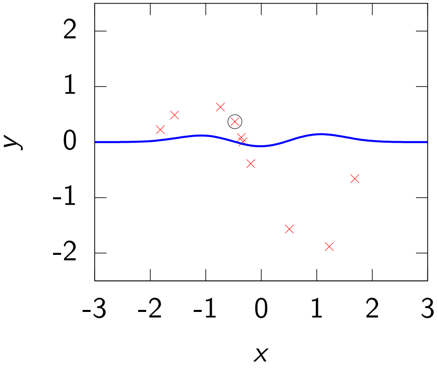
Nonlinear Regression Example

- Iteration 1
  - $w_1 = 0.13018$,
    $w_2 = -0.11355$,
    $w_3 = 0.15448$
  - Present data point 4



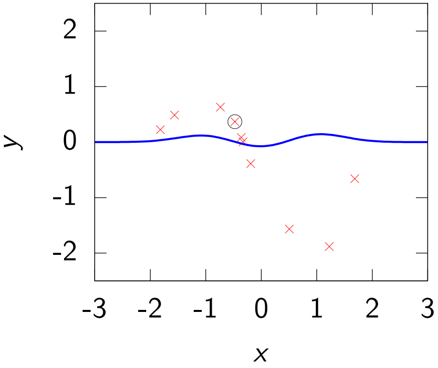
Nonlinear Regression Example

- Iteration 1
  - $w_1 = 0.13018$,
    $w_2 = -0.11355$,
    $w_3 = 0.15448$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$



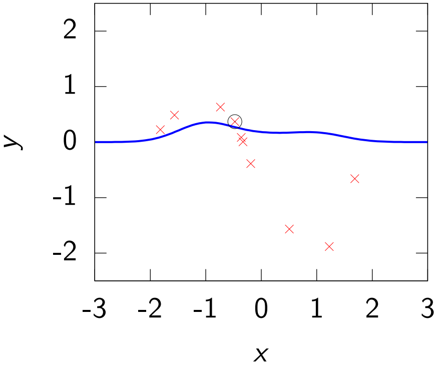
Nonlinear Regression Example

- Iteration 1
  - $w_1 = 0.13018$,
    $w_2 = -0.11355$,
    $w_3 = 0.15448$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$



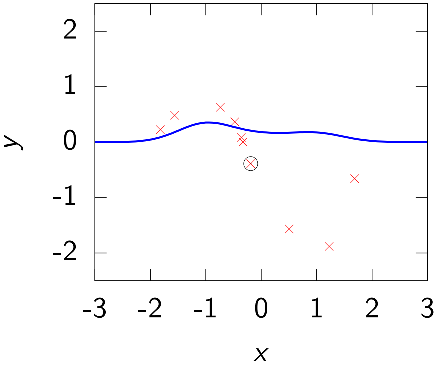
Nonlinear Regression Example

- Iteration 1
  - $w_1 = 0.13018$,
    $w_2 = -0.11355$,
    $w_3 = 0.15448$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_4 \Delta y_4$



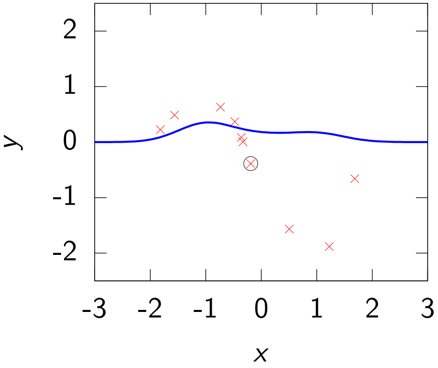
Nonlinear Regression Example

- Iteration 2
  - $w_1 = 0.33696$,
    $w_2 = 0.11481$,
    $w_3 = 0.1591$
  - Present data point 7



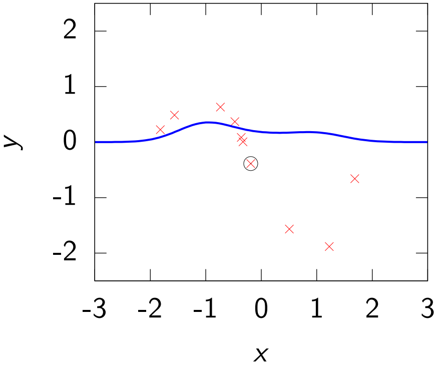
Nonlinear Regression Example

- Iteration 2
  - $w_1 = 0.33696$,
    $w_2 = 0.11481$,
    $w_3 = 0.1591$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$



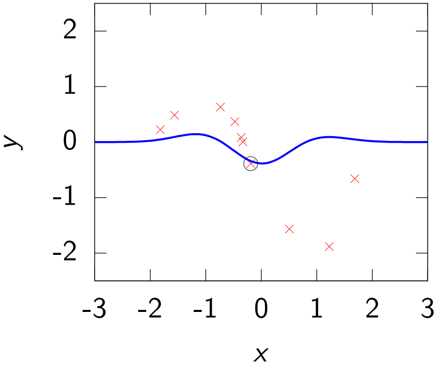
Nonlinear Regression Example

- Iteration 2
  - $w_1 = 0.33696$,
    $w_2 = 0.11481$,
    $w_3 = 0.1591$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$



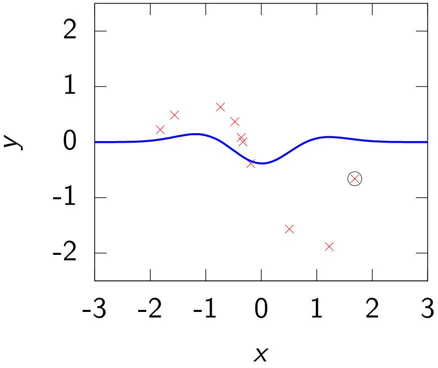
Nonlinear Regression Example

- Iteration 2
  - $w_1 = 0.33696$,
    $w_2 = 0.11481$,
    $w_3 = 0.1591$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_7 \Delta y_7$



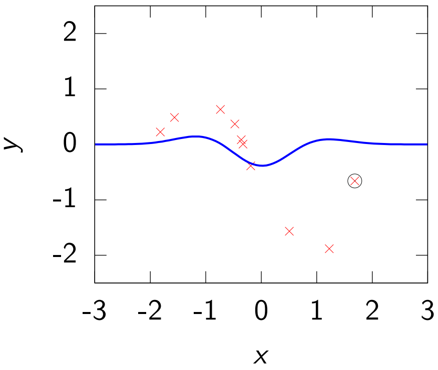
Nonlinear Regression Example

- Iteration 3
  - $w_1 = 0.18076$,
    $w_2 = -0.4266$,
    $w_3 = 0.12473$
  - Present data point 10



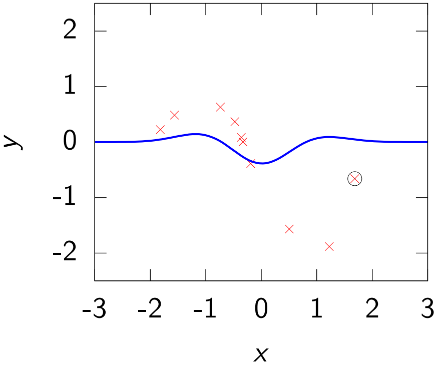
Nonlinear Regression Example

- Iteration 3
  - $w_1 = 0.18076$,
    $w_2 = -0.4266$,
    $w_3 = 0.12473$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^\top \mathbf{w}$



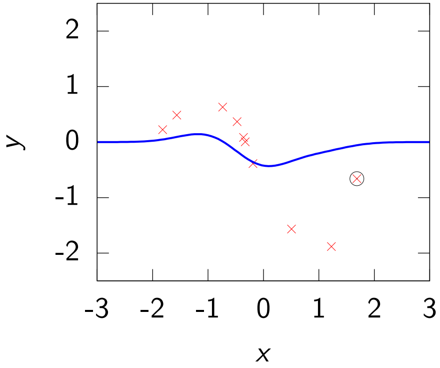
Nonlinear Regression Example

- Iteration 3
  - $w_1 = 0.18076$,
    $w_2 = -0.4266$,
    $w_3 = 0.12473$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$



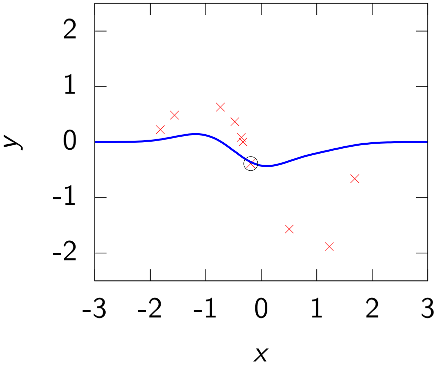
Nonlinear Regression Example

- Iteration 3
  - $w_1 = 0.18076$,
    $w_2 = -0.4266$,
    $w_3 = 0.12473$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta\phi_{10}\Delta y_{10}$



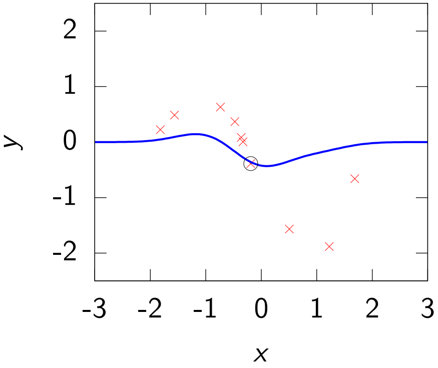
Nonlinear Regression Example

- Iteration 4
  - $w_1 = 0.18076$,
    $w_2 = -0.42893$,
    $w_3 = -0.14306$
  - Present data point 7



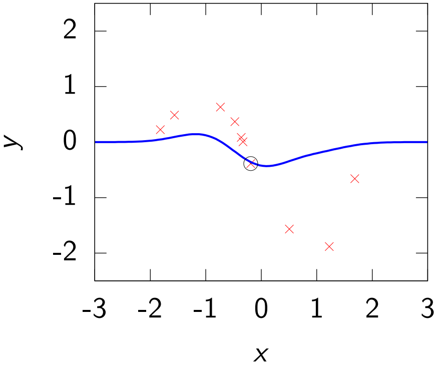
Nonlinear Regression Example

- Iteration 4
  - $w_1 = 0.18076$,
    $w_2 = -0.42893$,
    $w_3 = -0.14306$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$



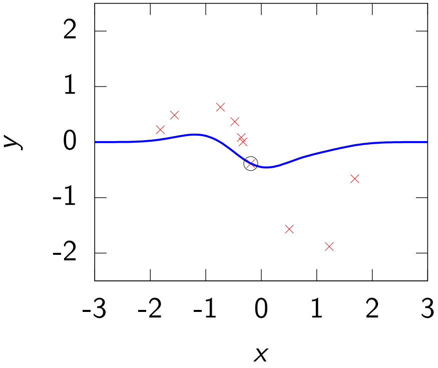
Nonlinear Regression Example

- Iteration 4
  - $w_1 = 0.18076$,
    $w_2 = -0.42893$,
    $w_3 = -0.14306$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$



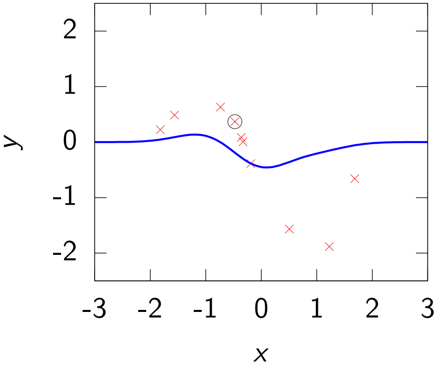
Nonlinear Regression Example

- Iteration 4
  - $w_1 = 0.18076$,
    $w_2 = -0.42893$,
    $w_3 = -0.14306$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_7 \Delta y_7$



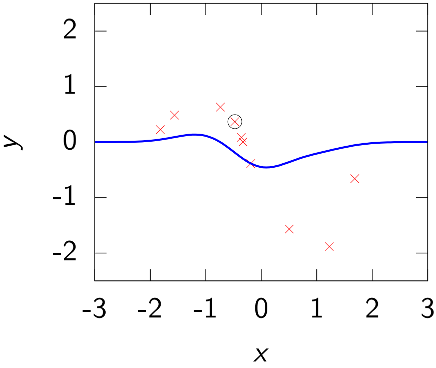
Nonlinear Regression Example

- Iteration 5
  - $w_1 = 0.17372$,
    $w_2 = -0.45335$,
    $w_3 = -0.14461$
  - Present data point 4



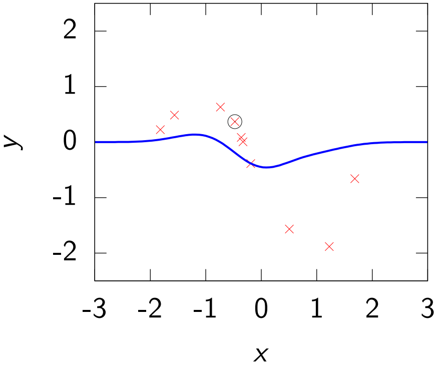
Nonlinear Regression Example

- Iteration 5
  - $w_1 = 0.17372$,
    $w_2 = -0.45335$,
    $w_3 = -0.14461$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$



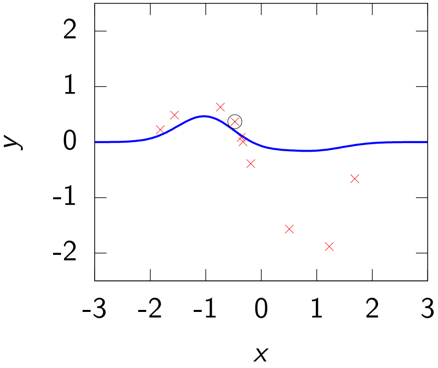
Nonlinear Regression Example

- Iteration 5
  - $w_1 = 0.17372$,
    $w_2 = -0.45335$,
    $w_3 = -0.14461$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$



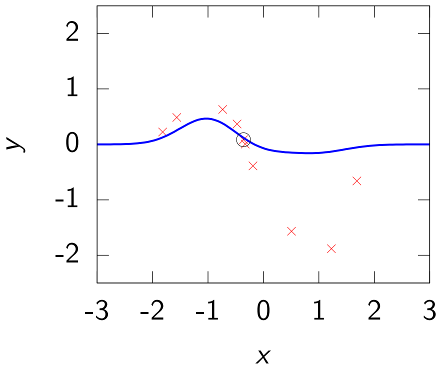
Nonlinear Regression Example

- Iteration 5
  - $w_1 = 0.17372$,
    $w_2 = -0.45335$,
    $w_3 = -0.14461$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_4 \Delta y_4$



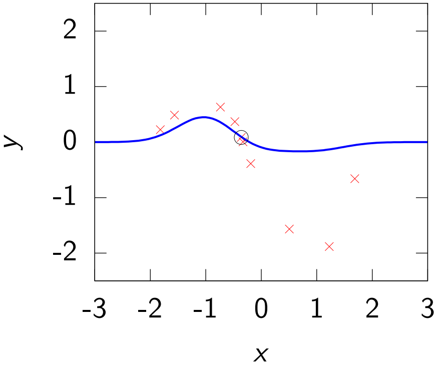
Nonlinear Regression Example

- Iteration 6
  - $w_1 = 0.47971$,
    $w_2 = -0.11541$,
    $w_3 = -0.13778$
  - Present data point 5
  - $\Delta y_5 = y_5 - \phi_5^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_5 \Delta y_5$



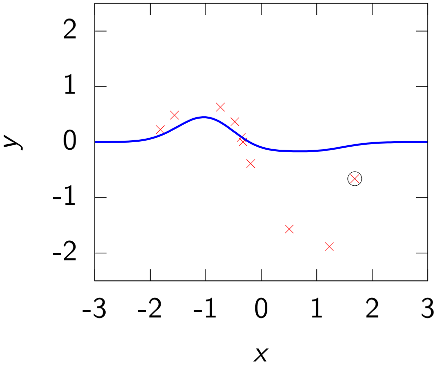
Nonlinear Regression Example

- Iteration 6
  - $w_1 = 0.47971,$
    $w_2 = -0.11541,$
    $w_3 = -0.13778$
  - Present data point 5
  - $\Delta y_5 = y_5 - \phi_5^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_5 \Delta y_5$



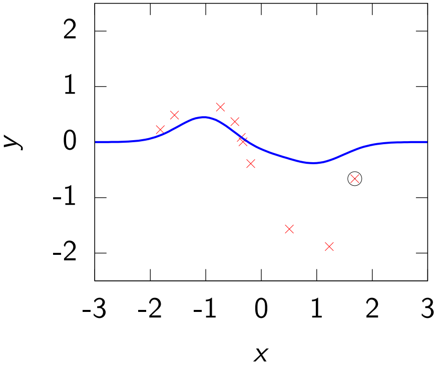
Nonlinear Regression Example

- Iteration 7
  - $w_1 = 0.46599$,
    $w_2 = -0.13952$,
    $w_3 = -0.13855$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta\phi_{10}\Delta y_{10}$



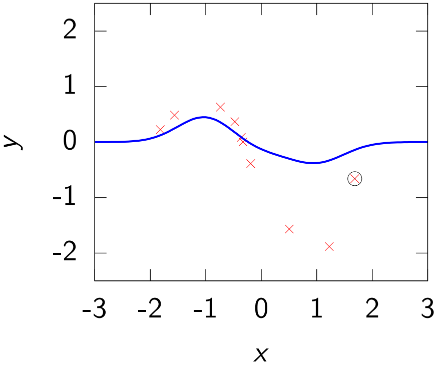
Nonlinear Regression Example

- Iteration 7
  - $w_1 = 0.46599$,
    $w_2 = -0.13952$,
    $w_3 = -0.13855$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta\phi_{10}\Delta y_{10}$



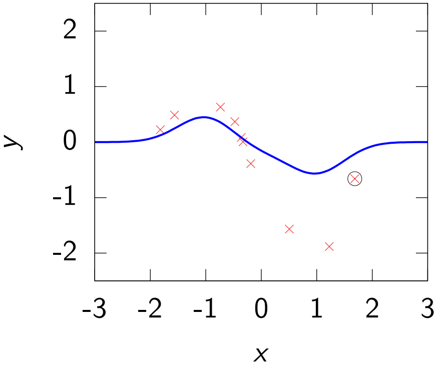
Nonlinear Regression Example

- Iteration 8
  - $w_1 = 0.46599$,
    $w_2 = -0.14144$,
    $w_3 = -0.35924$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_{10} \Delta y_{10}$



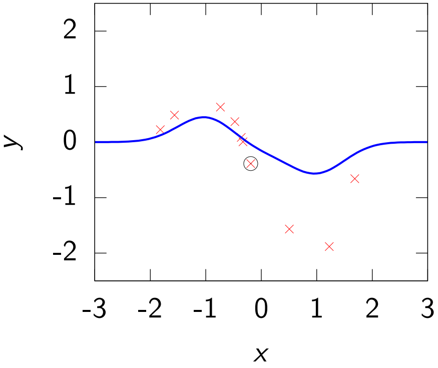
Nonlinear Regression Example

- Iteration 8
  - $w_1 = 0.46599$,
    $w_2 = -0.14144$,
    $w_3 = -0.35924$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_{10} \Delta y_{10}$



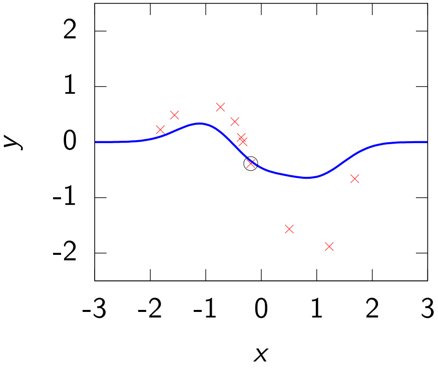
Nonlinear Regression Example

- Iteration 9
  - $w_1 = 0.46599$,
    $w_2 = -0.14307$,
    $w_3 = -0.54679$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_7 \Delta y_7$



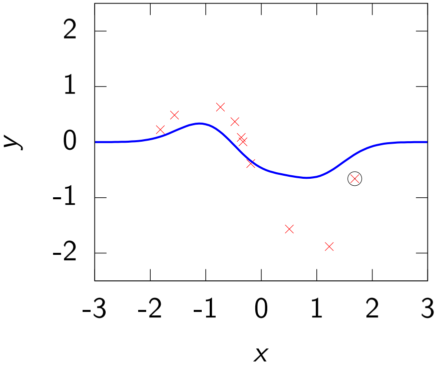
Nonlinear Regression Example

- Iteration 9
  - $w_1 = 0.46599$,
    $w_2 = -0.14307$,
    $w_3 = -0.54679$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_7 \Delta y_7$



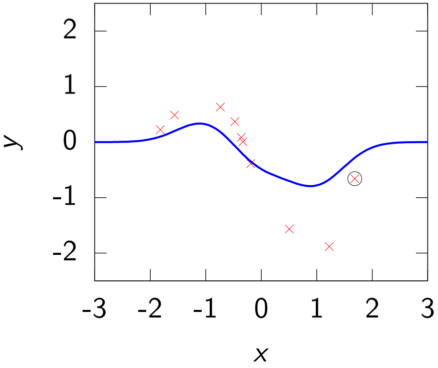
Nonlinear Regression Example

- Iteration 10
  - $w_1 = 0.38071$,
    $w_2 = -0.43867$,
    $w_3 = -0.56556$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta\phi_{10}\Delta y_{10}$



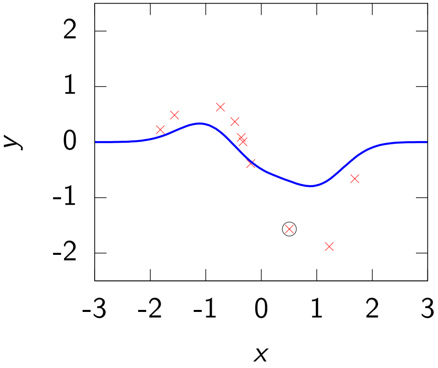
Nonlinear Regression Example

- Iteration 10
  - $w_1 = 0.38071,$
    $w_2 = -0.43867,$
    $w_3 = -0.56556$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta\phi_{10}\Delta y_{10}$



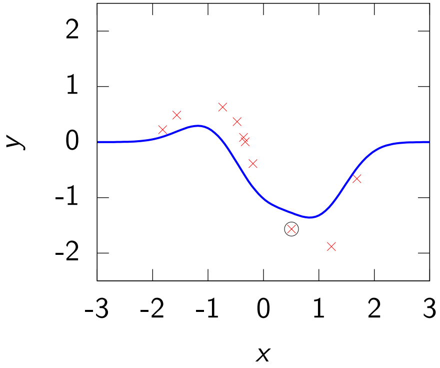
Nonlinear Regression Example

- Iteration 11
  - $w_1 = 0.38071$,
    $w_2 = -0.44002$,
    $w_3 = -0.7208$
  - Present data point 8
  - $\Delta y_8 = y_8 - \phi_8^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_8 \Delta y_8$



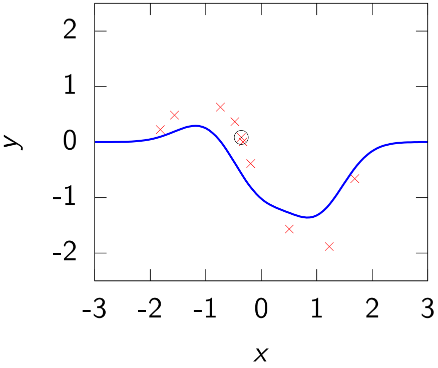
Nonlinear Regression Example

- Iteration 11
  - $w_1 = 0.38071$,
    $w_2 = -0.44002$,
    $w_3 = -0.7208$
  - Present data point 8
  - $\Delta y_8 = y_8 - \phi_8^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_8 \Delta y_8$



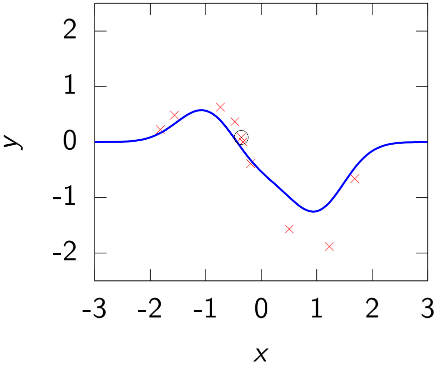
Nonlinear Regression Example

- Iteration 12
  - $w_1 = 0.37237$,
    $w_2 = -0.90666$,
    $w_3 = -1.1987$
  - Present data point 5
  - $\Delta y_5 = y_5 - \phi_5^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_5 \Delta y_5$



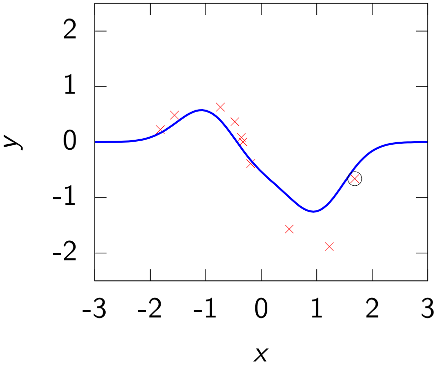
Nonlinear Regression Example

- Iteration 12
  - $w_1 = 0.37237$,
    $w_2 = -0.90666$,
    $w_3 = -1.1987$
  - Present data point 5
  - $\Delta y_5 = y_5 - \phi_5^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_5 \Delta y_5$



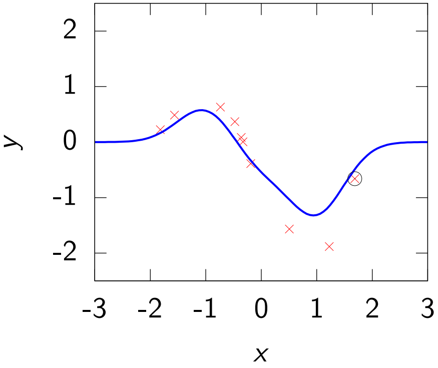
Nonlinear Regression Example

- Iteration 13
  - $w_1 = 0.62833$,
    $w_2 = -0.45691$,
    $w_3 = -1.1842$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_{10} \Delta y_{10}$



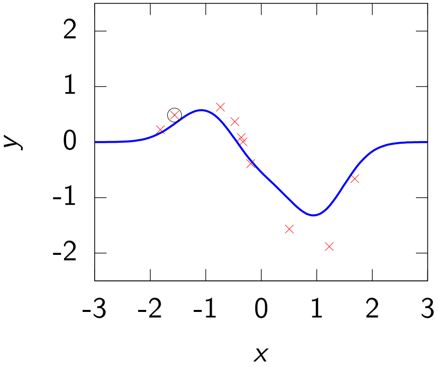
Nonlinear Regression Example

- Iteration 13
  - $w_1 = 0.62833$,
    $w_2 = -0.45691$,
    $w_3 = -1.1842$
  - Present data point 10
  - $\Delta y_{10} = y_{10} - \phi_{10}^{\top}\mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta\phi_{10}\Delta y_{10}$



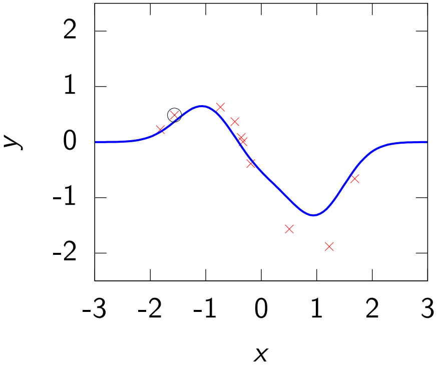
Nonlinear Regression Example

- Iteration 14
  - $w_1 = 0.62833$,
    $w_2 = -0.4575$,
    $w_3 = -1.252$
  - Present data point 2
  - $\Delta y_2 = y_2 - \phi_2^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_2 \Delta y_2$



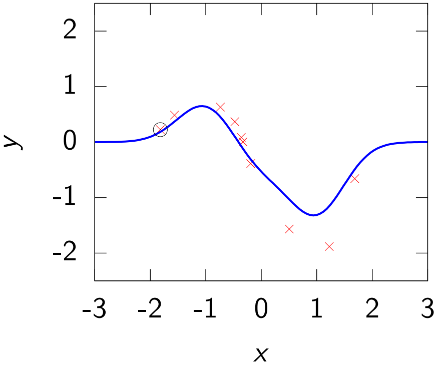
Nonlinear Regression Example

- Iteration 14
  - $w_1 = 0.62833$,
    $w_2 = -0.4575$,
    $w_3 = -1.252$
  - Present data point 2
  - $\Delta y_2 = y_2 - \phi_2^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_2 \Delta y_2$



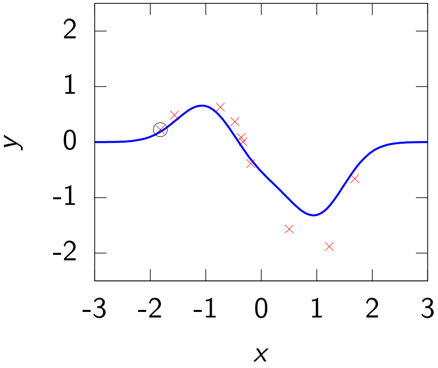
Nonlinear Regression Example

- Iteration 15
  - $w_1 = 0.7016$,
    $w_2 = -0.45646$,
    $w_3 = -1.252$
  - Present data point 1
  - $\Delta y_1 = y_1 - \phi_1^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_1 \Delta y_1$



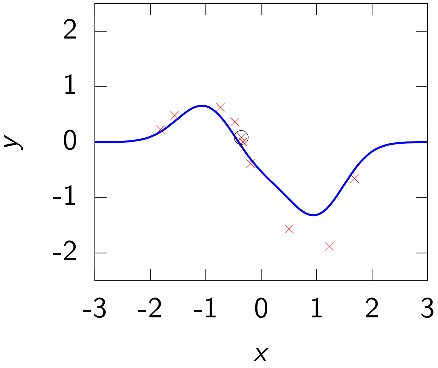
Nonlinear Regression Example

- Iteration 15
  - $w_1 = 0.7016$,
    $w_2 = -0.45646$,
    $w_3 = -1.252$
  - Present data point 1
  - $\Delta y_1 = y_1 - \phi_1^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_1 \Delta y_1$



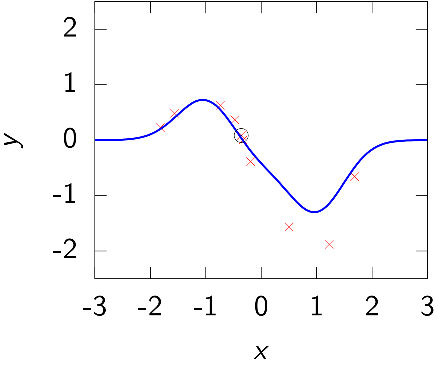
Nonlinear Regression Example

- Iteration 16
  - $w_1 = 0.7109$,
    $w_2 = -0.45641$,
    $w_3 = -1.252$
  - Present data point 5
  - $\Delta y_5 = y_5 - \phi_5^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_5 \Delta y_5$



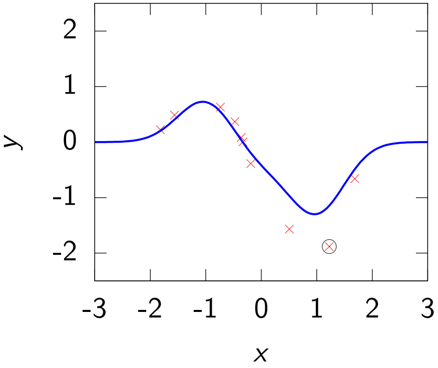
Nonlinear Regression Example

- Iteration 16
  - $w_1 = 0.7109$,
    $w_2 = -0.45641$,
    $w_3 = -1.252$
  - Present data point 5
  - $\Delta y_5 = y_5 - \phi_5^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_5 \Delta y_5$



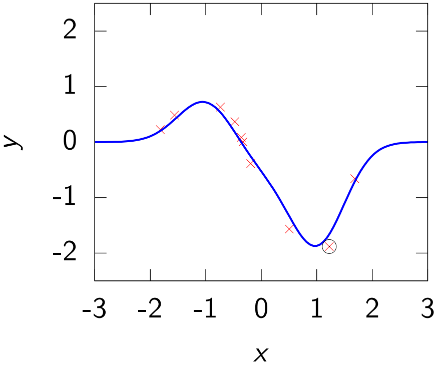
Nonlinear Regression Example

- Iteration 17
  - $w_1 = 0.77022$,
    $w_2 = -0.35219$,
    $w_3 = -1.2487$
  - Present data point 9
  - $\Delta y_9 = y_9 - \phi_9^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_9 \Delta y_9$



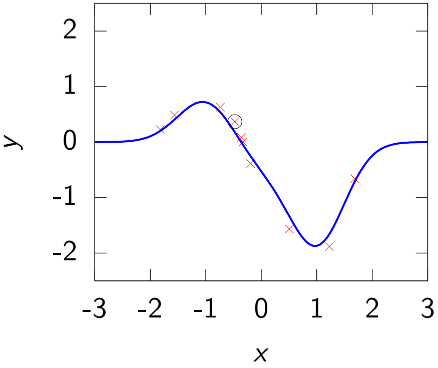
Nonlinear Regression Example

- Iteration 17
  - $w_1 = 0.77022$,
    $w_2 = -0.35219$,
    $w_3 = -1.2487$
  - Present data point 9
  - $\Delta y_9 = y_9 - \phi_9^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_9 \Delta y_9$



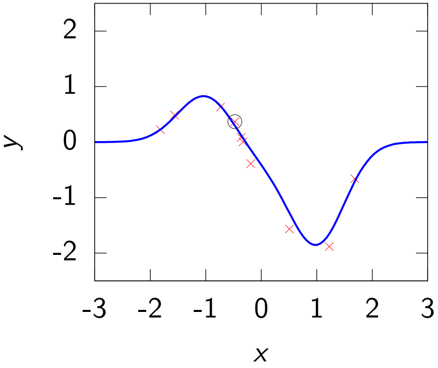
Nonlinear Regression Example

- Iteration 18
  - $w_1 = 0.77019$,
    $w_2 = -0.3832$,
    $w_3 = -1.8175$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_4 \Delta y_4$



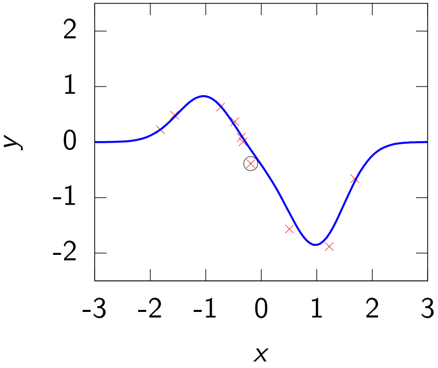
Nonlinear Regression Example

- Iteration 18
  - $w_1 = 0.77019,$
    $w_2 = -0.3832,$
    $w_3 = -1.8175$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_4 \Delta y_4$



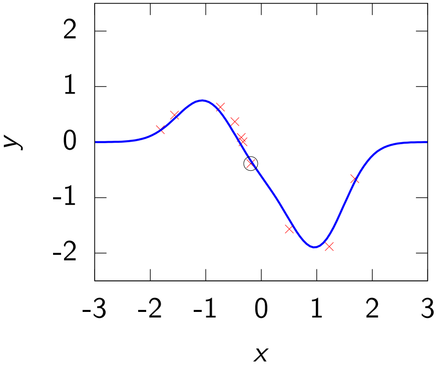
Nonlinear Regression Example

- Iteration 19
  - $w_1 = 0.86321$,
    $w_2 = -0.28046$,
    $w_3 = -1.8154$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_7 \Delta y_7$



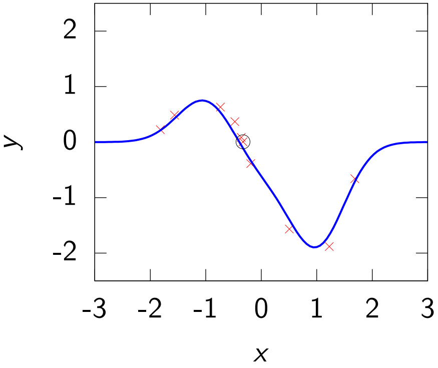
Nonlinear Regression Example

- Iteration 19
  - $w_1 = 0.86321$,
    $w_2 = -0.28046$,
    $w_3 = -1.8154$
  - Present data point 7
  - $\Delta y_7 = y_7 - \phi_7^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_7 \Delta y_7$



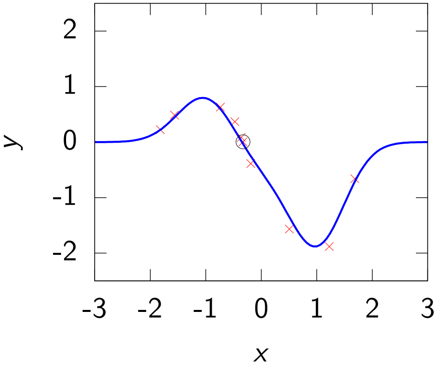
Nonlinear Regression Example

- Iteration 20
  - $w_1 = 0.80681$,
    $w_2 = -0.47597$,
    $w_3 = -1.8278$
  - Present data point 6
  - $\Delta y_6 = y_6 - \phi_6^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_6 \Delta y_6$



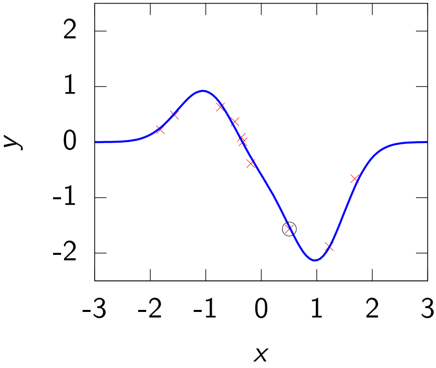
Nonlinear Regression Example

- Iteration 20
  - $w_1 = 0.80681$,
    $w_2 = -0.47597$,
    $w_3 = -1.8278$
  - Present data point 6
  - $\Delta y_6 = y_6 - \phi_6^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_6 \Delta y_6$



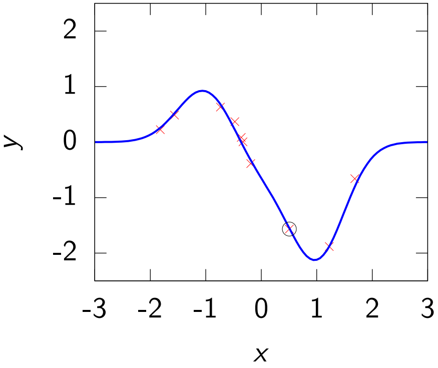
Nonlinear Regression Example

- Iteration 50
  - $w_1 = 0.9777$,
    $w_2 = -0.4076$,
    $w_3 = -2.038$
  - Present data point 8
  - $\Delta y_8 = y_8 - \phi_8^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_8 \Delta y_8$



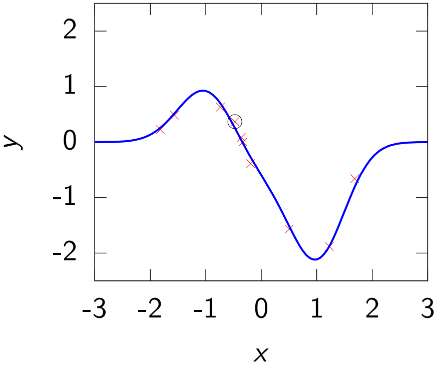
Nonlinear Regression Example

- Iteration 100
  - $w_1 = 0.98593,$
    $w_2 = -0.49744,$
    $w_3 = -2.046$
  - Present data point 8
  - $\Delta y_8 = y_8 - \phi_8^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_8 \Delta y_8$



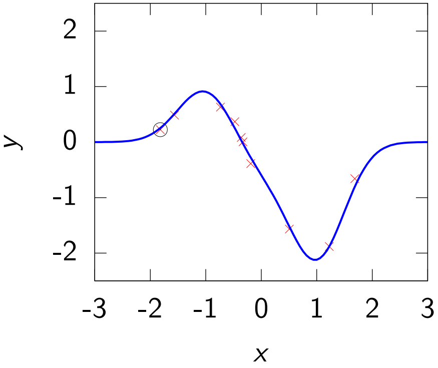
Nonlinear Regression Example

- Iteration 200
  - $w_1 = 0.95307$,
    $w_2 = -0.48041$,
    $w_3 = -2.0553$
  - Present data point 4
  - $\Delta y_4 = y_4 - \phi_4^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_4 \Delta y_4$



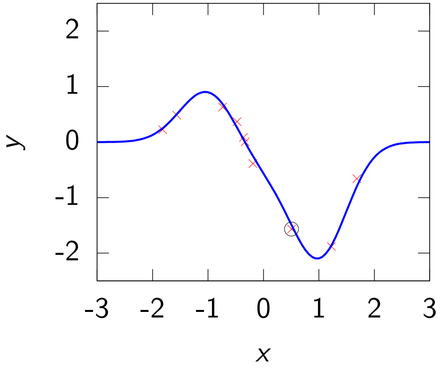
Nonlinear Regression Example

- Iteration 300
  - $w_1 = 0.97066$,
    $w_2 = -0.44667$,
    $w_3 = -2.0588$
  - Present data point 1
  - $\Delta y_1 = y_1 - \phi_1^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_1 \Delta y_1$



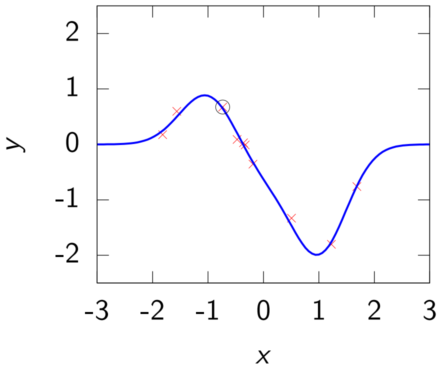
Nonlinear Regression Example

- Iteration 400
  - $w_1 = 0.95515$,
    $w_2 = -0.40611$,
    $w_3 = -2.0289$
  - Present data point 8
  - $\Delta y_8 = y_8 - \phi_8^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_8 \Delta y_8$



Nonlinear Regression Example

- Iteration 500
  - $w_1 = 0.94178$,
    $w_2 = -0.49879$,
    $w_3 = -1.9209$
  - Present data point 5
  - $\Delta y_5 = y_5 - \phi_5^\top \mathbf{w}$
  - Adjust $\hat{\mathbf{w}}$
- Updated values
  $\hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \eta \phi_5 \Delta y_5$



Outline

Mathematical Interpretation

- What is the mathematical interpretation?
  - There is a cost function.
  - It expresses mismatch between your prediction and reality.

$$E(\mathbf{w}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{K} w_j \phi_j(x_i) - y_i \right)^2$$

  - This is known as the sum of squares error.

Mathematical Interpretation

- What is the mathematical interpretation?
  - There is a cost function.
  - It expresses mismatch between your prediction and reality.

$$E(\mathbf{w}) = \sum_{i=1}^{n} \left(\mathbf{w}^\top \boldsymbol{\phi}_i - y_i\right)^2$$

  - This is known as the sum of squares error.
  - Defining $\boldsymbol{\phi}_i = [\phi_1(x_i), \ldots, \phi_K(x_i)]^\top$.

Learning is Optimization

- Learning is minimization of the cost function.
- At the minima the gradient is zero.
- Gradient of error function:

$$\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}\mathbf{w}} = -2\sum_{i=1}^{n} \phi_i \left(y_i - \mathbf{w}^\top \phi_i\right)$$

Learning is Optimization

- Learning is minimization of the cost function.
- At the minima the gradient is zero.
- Gradient of error function:

$$\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}\mathbf{w}} = -2\sum_{i=1}^{n} \phi_i \Delta y_i$$

- Where $\Delta y_i = \left(y_i - \mathbf{w}^\top \phi_i\right)$.

Minimization via Gradient Descent

- One way of minimizing is steepest descent.
- Initialize algorithm with $\mathbf{w}$.
- Compute gradient of error function, $\frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}\mathbf{w}}$.
- Change $\mathbf{w}$ by moving in steepest downhill direction.

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}\mathbf{w}}$$

Steepest Descent

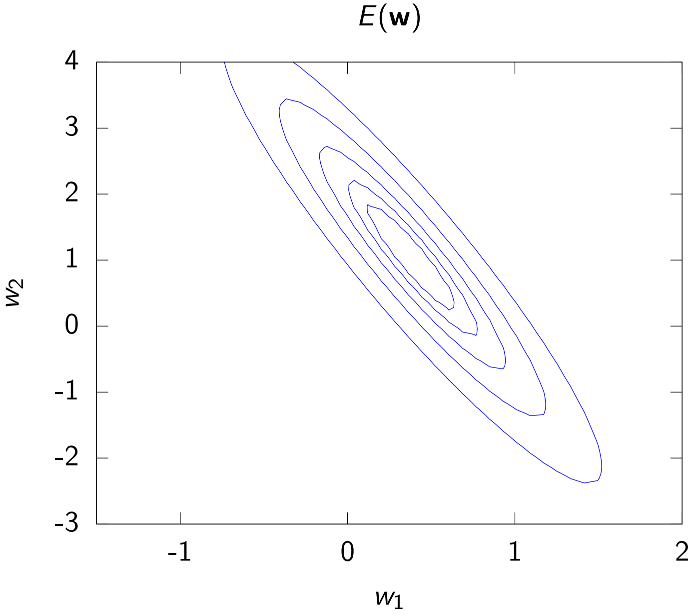


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

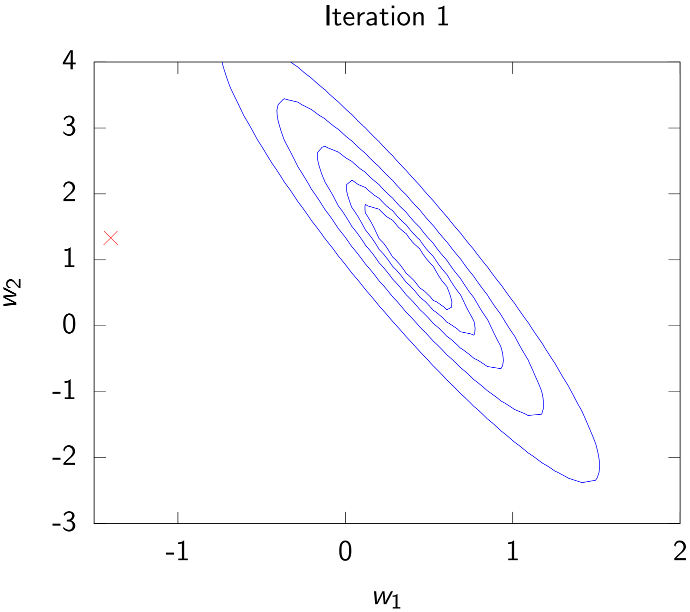


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

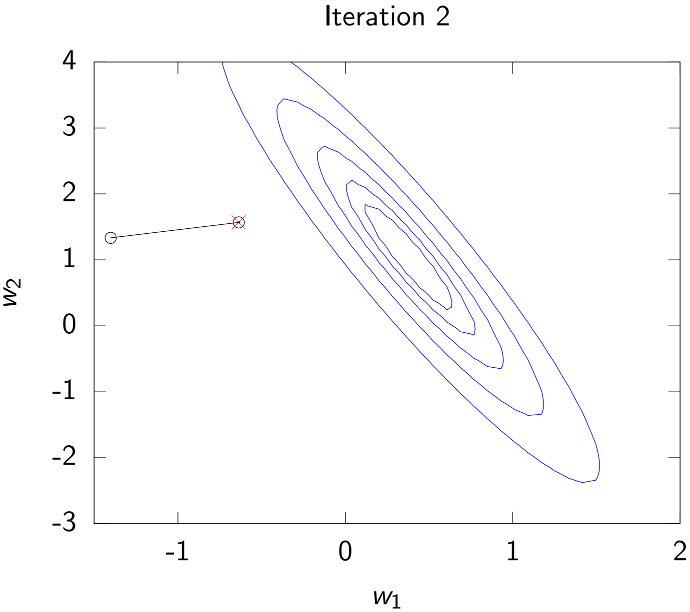


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

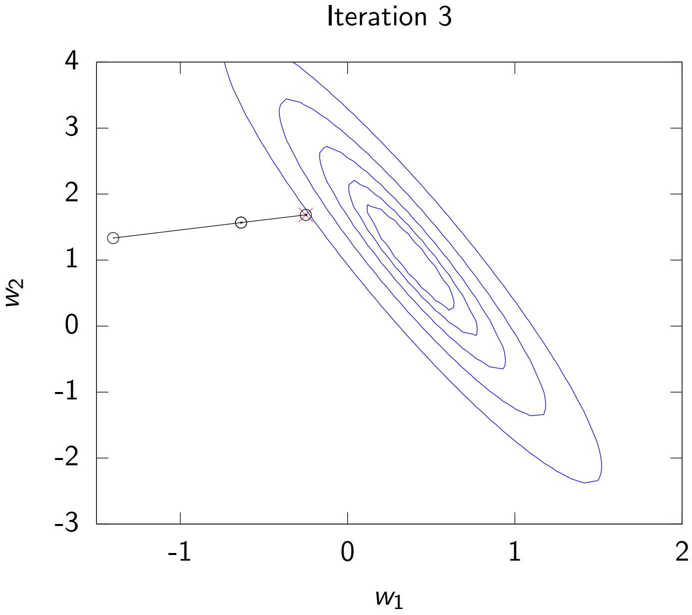


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

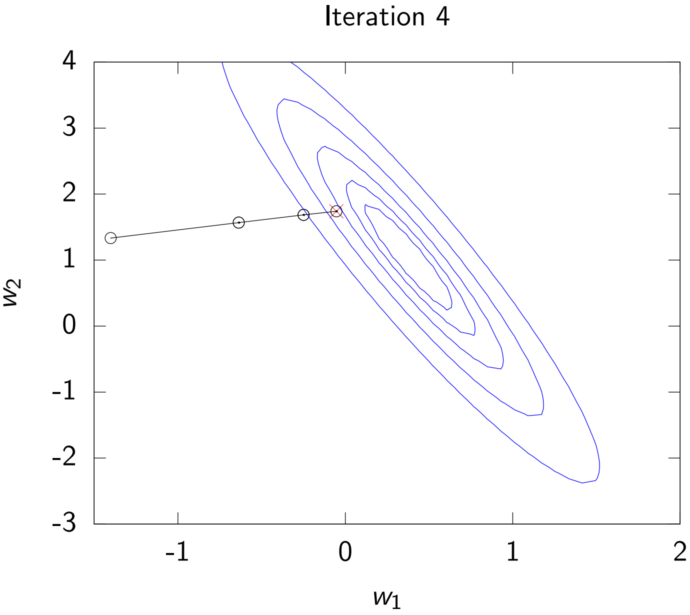


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

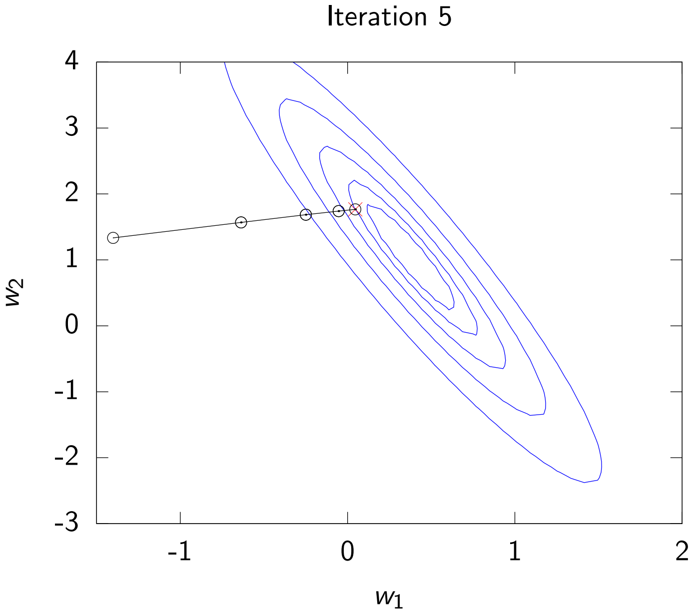


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

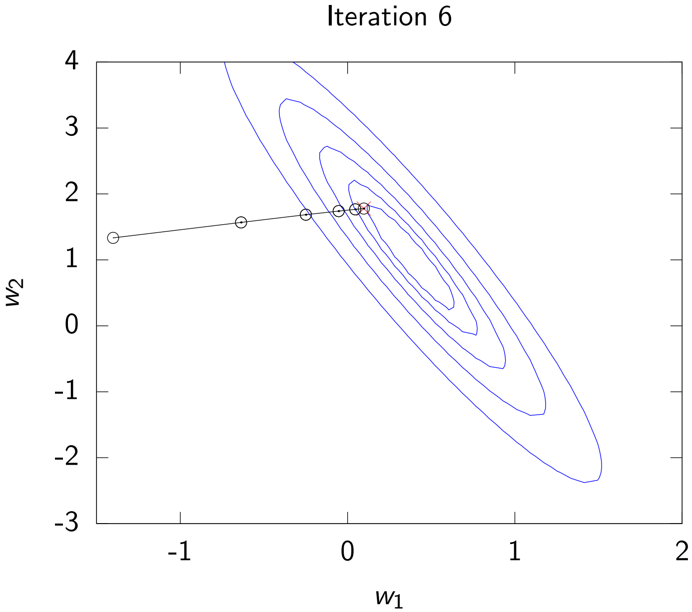


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

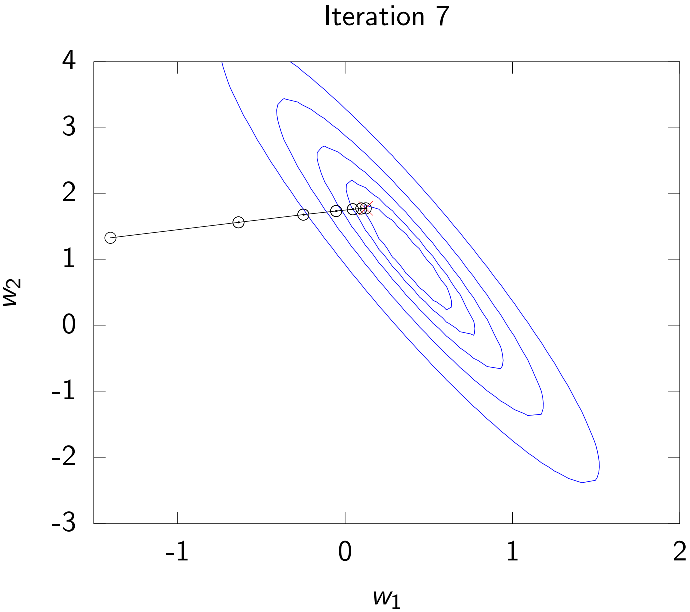


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

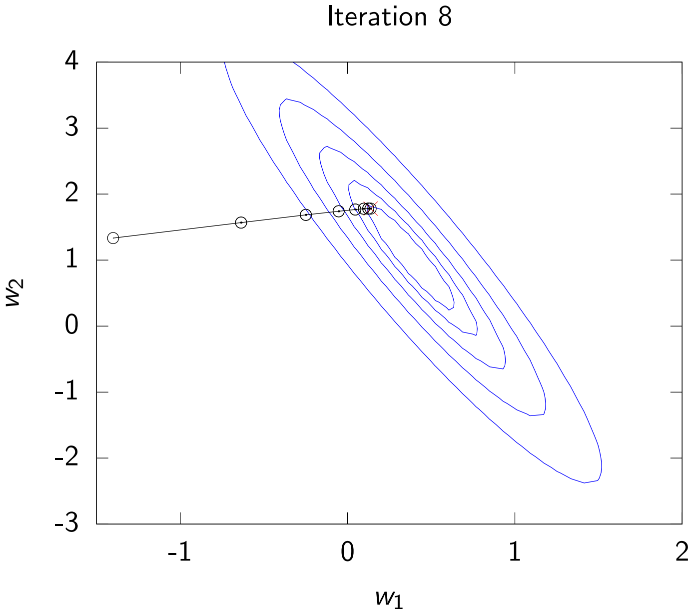


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

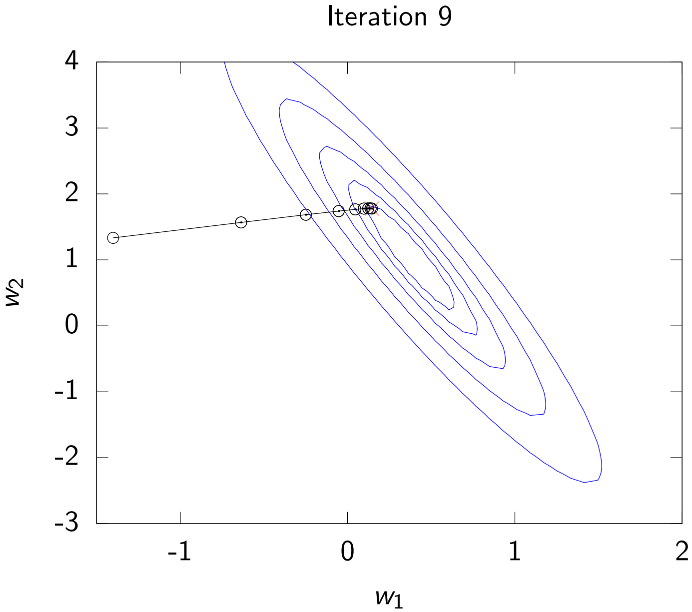


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

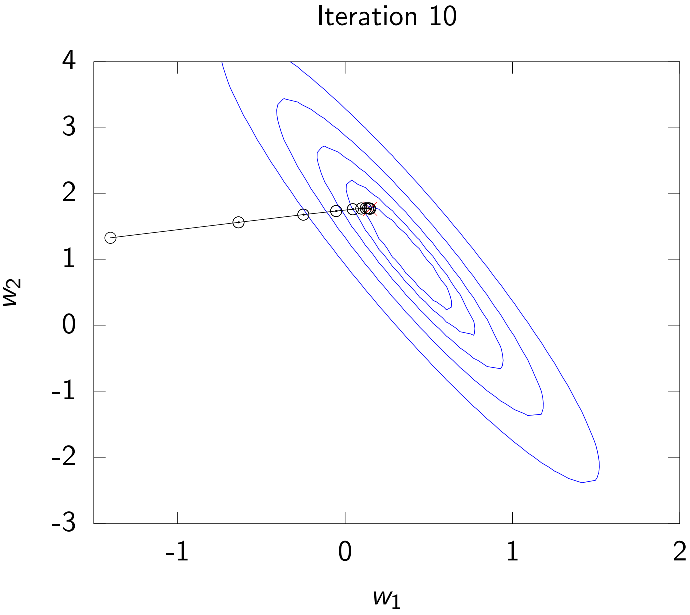


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

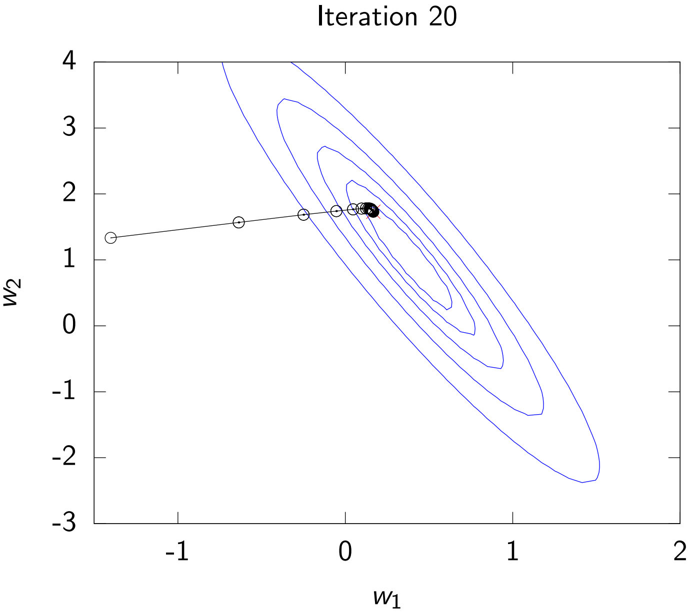


Figure: Steepest descent on a quadratic error surface.

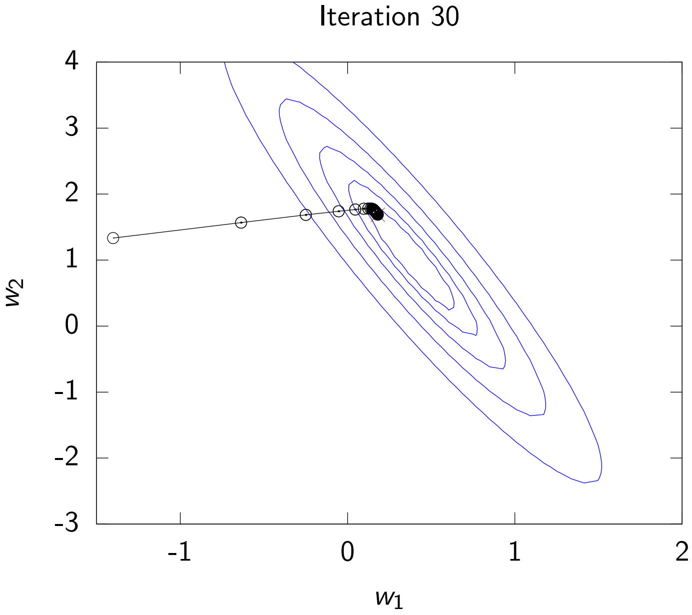


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

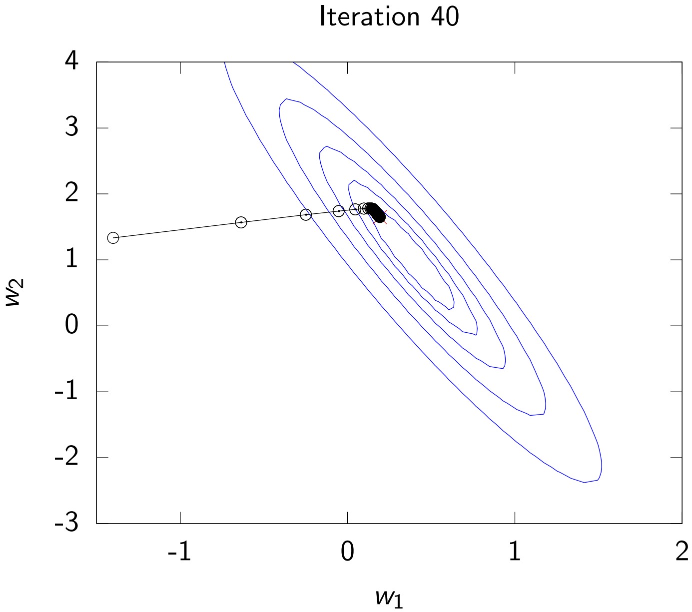


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

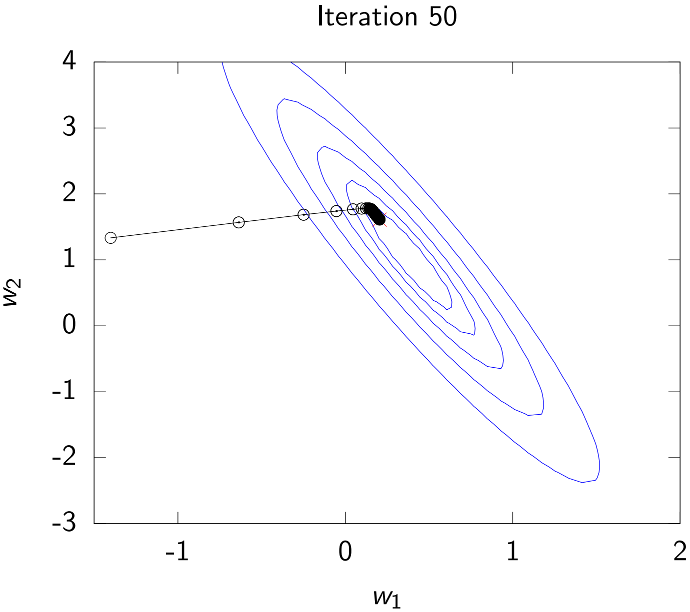


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

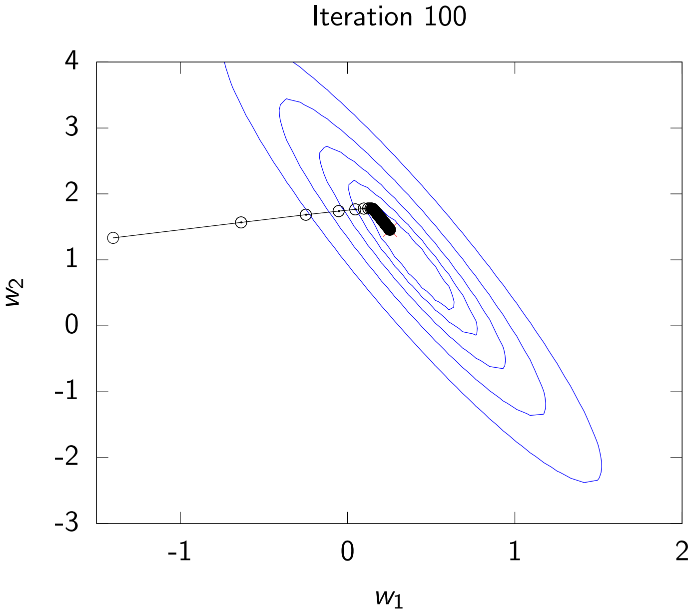


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

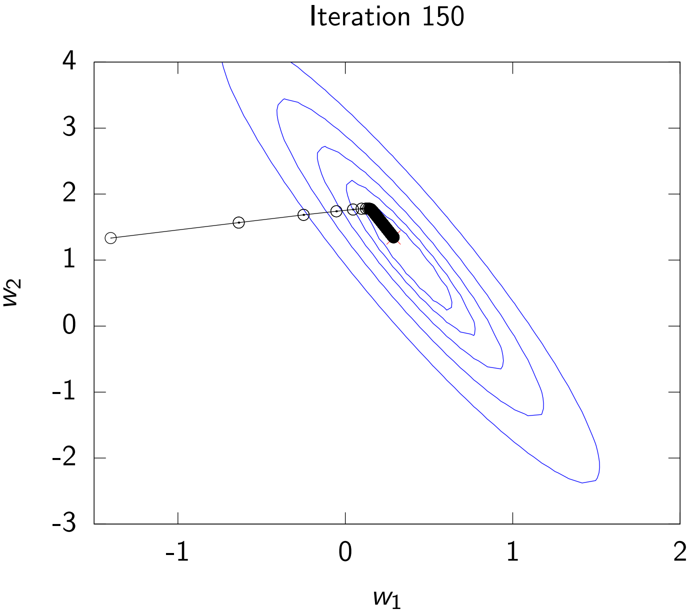


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

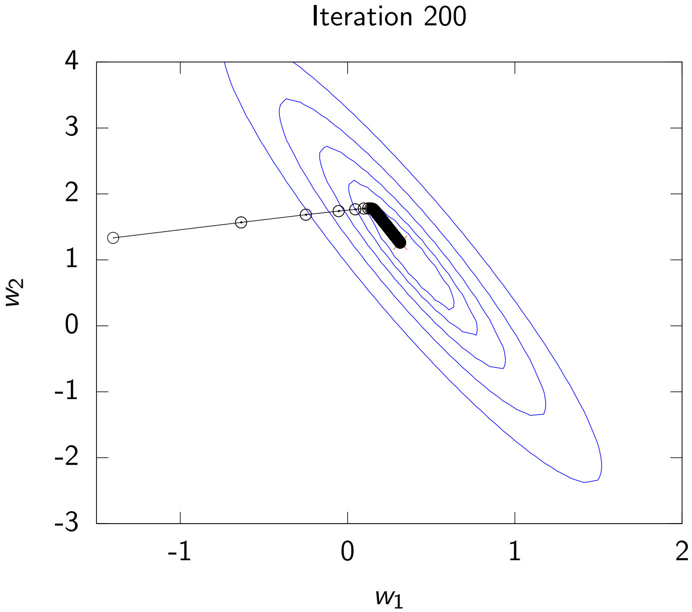


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

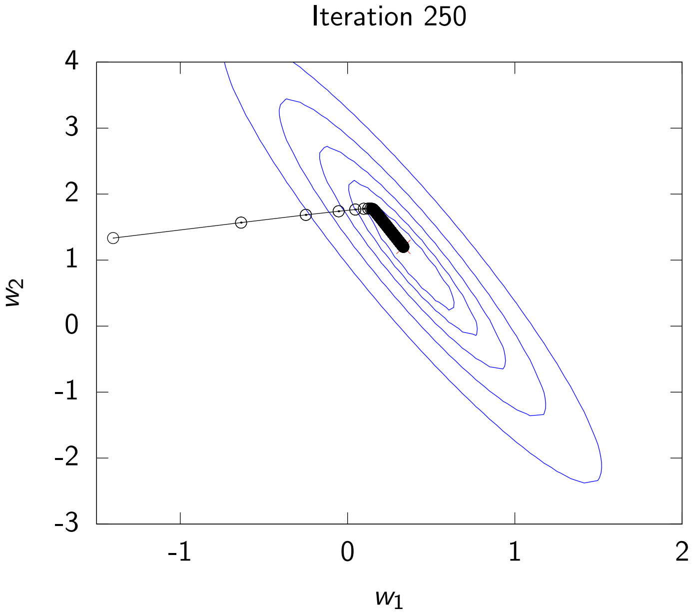


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

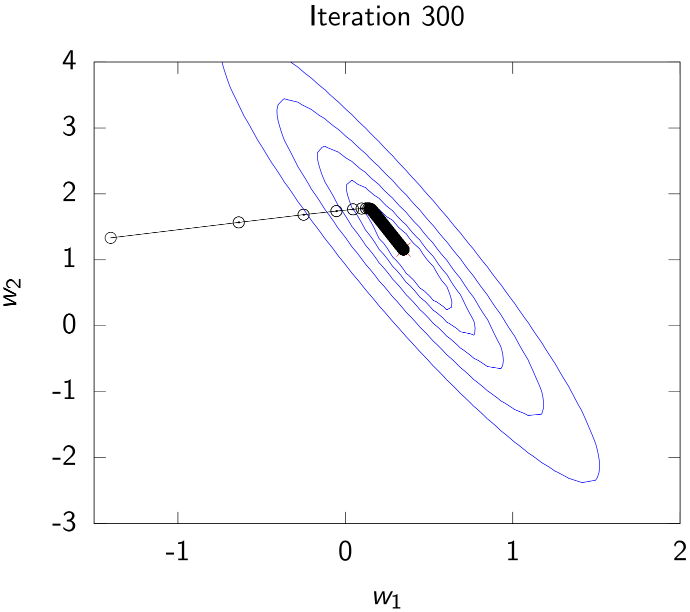


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

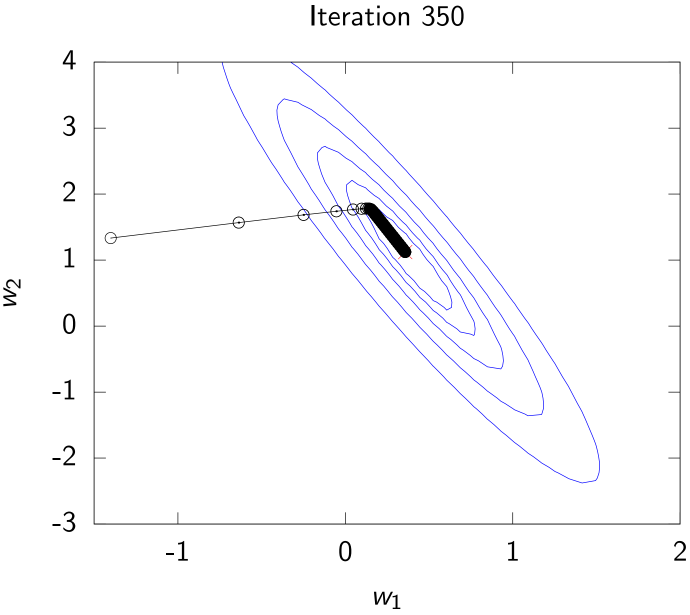


Figure: Steepest descent on a quadratic error surface.

Steepest Descent

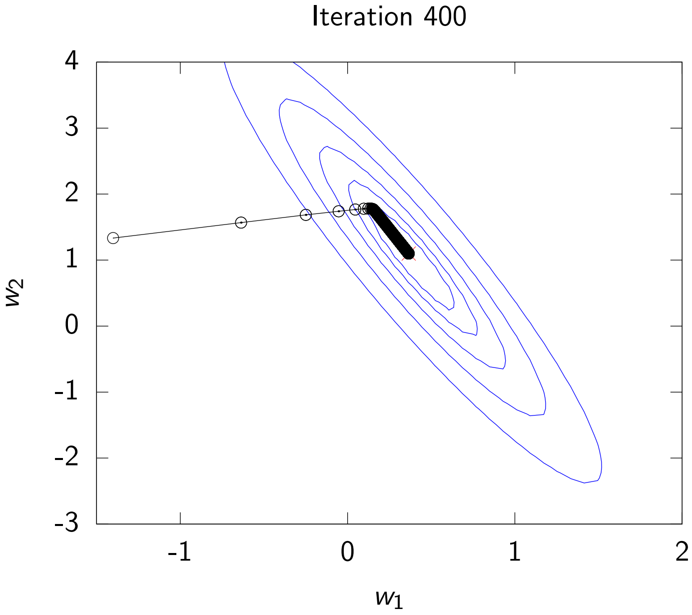


Figure: Steepest descent on a quadratic error surface.

Stochastic Gradient Descent

How does this relate to learning rules we presented?

- For regression, the learning rule can be seen as a variant of gradient descent.
- This variant is known as stochastic gradient descent.
- For regression steepest descent gives

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \frac{\mathrm{d}E(\mathbf{w})}{\mathrm{d}\mathbf{w}}$$

Stochastic Gradient Descent

How does this relate to learning rules we presented?

- For regression, the learning rule can be seen as a variant of gradient descent.
- This variant is known as stochastic gradient descent.
- For regression steepest descent gives

$$\mathbf{w} \leftarrow \mathbf{w} - 2\eta \sum_{i=1}^{n} \phi_i \left(\mathbf{w}^\top \phi_i - y_i\right)$$

Stochastic Gradient Descent

How does this relate to learning rules we presented?

- For regression, the learning rule can be seen as a variant of gradient descent.
- This variant is known as stochastic gradient descent.
- For regression steepest descent gives

$$\mathbf{w} \leftarrow \mathbf{w} - \eta' \sum_{i=1}^{n} \phi_i \left(\mathbf{w}^\top \phi_i - y_i \right)$$

Stochastic Gradient Descent

How does this relate to learning rules we presented?

- For regression, the learning rule can be seen as a variant of gradient descent.
- This variant is known as stochastic gradient descent.
- For regression steepest descent gives

$$\mathbf{w} \leftarrow \mathbf{w} - \eta' \sum_{i=1}^{n} \phi_i \left(\mathbf{w}^\top \phi_i - y_i \right)$$

Stochastic Gradient Descent

How does this relate to learning rules we presented?

- For regression, the learning rule can be seen as a variant of gradient descent.
- This variant is known as stochastic gradient descent.
- For regression steepest descent gives

$$\mathbf{w} \leftarrow \mathbf{w} - \eta' \sum_{i=1}^{n} \phi_i \Delta y_i$$

Stochastic Gradient Descent

How does this relate to learning rules we presented?

- For regression, the learning rule can be seen as a variant of gradient descent.
- This variant is known as stochastic gradient descent.
- For regression steepest descent gives

$$\mathbf{w} \leftarrow \mathbf{w} - \eta' \sum_{i=1}^{n} \phi_i \Delta y_i$$

- And the stochastic approximation is

$$\mathbf{w} \leftarrow \mathbf{w} + \eta' \phi_i \Delta y_i$$

Stochastic Gradient Descent

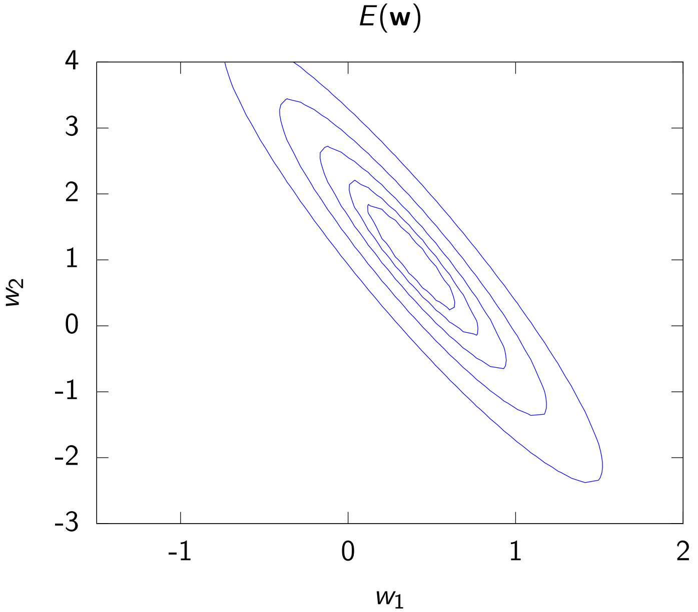


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

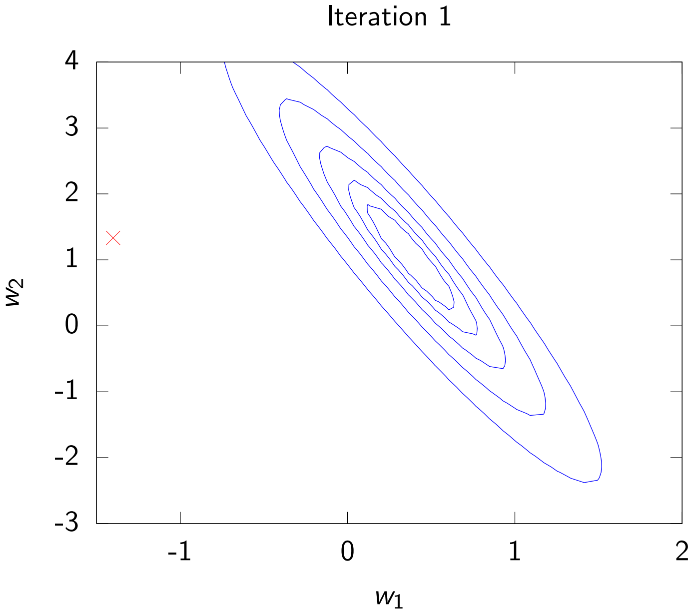


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

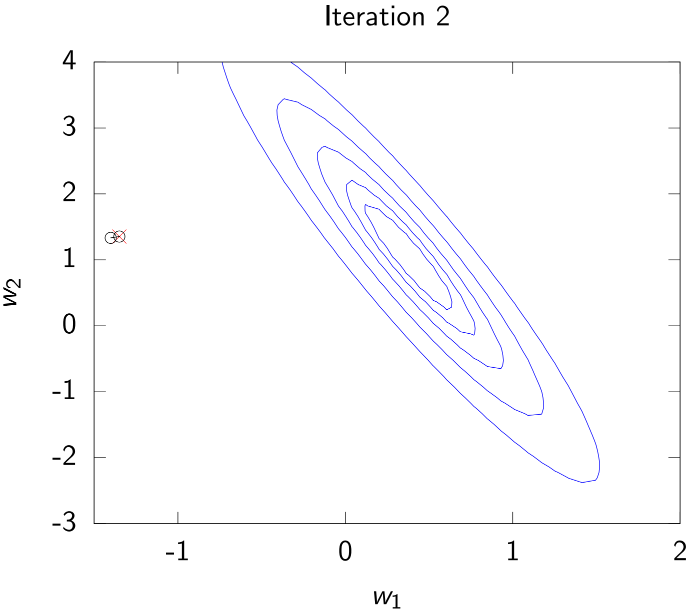


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

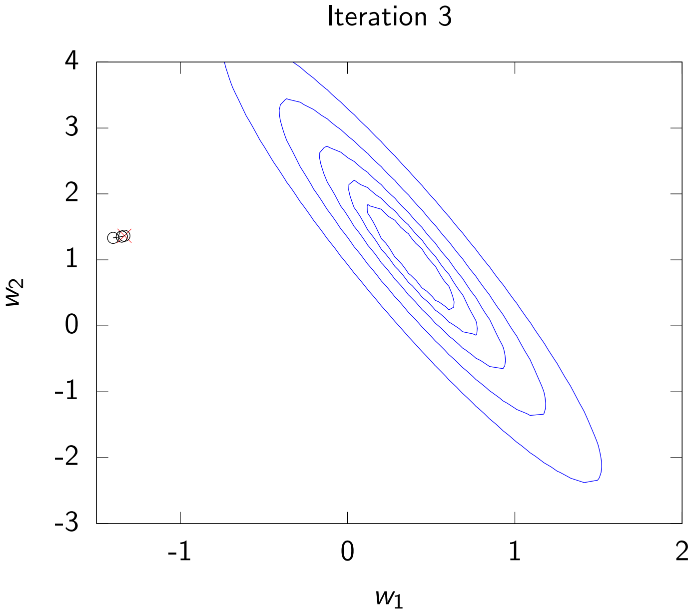


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

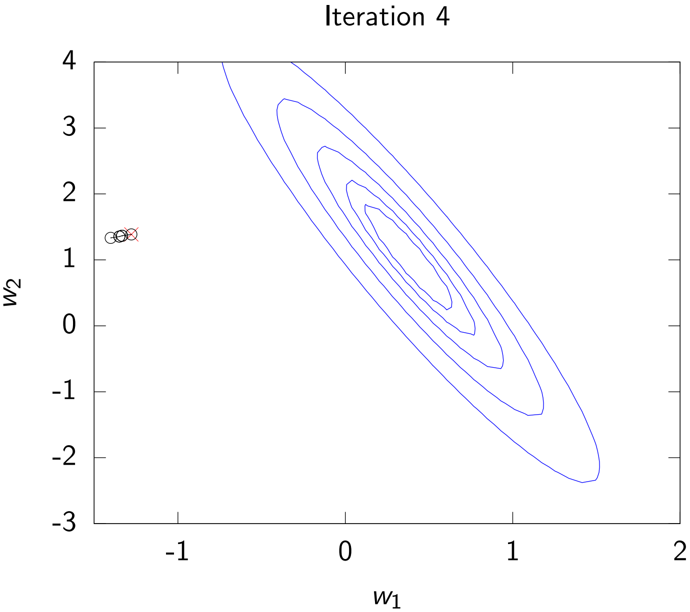


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

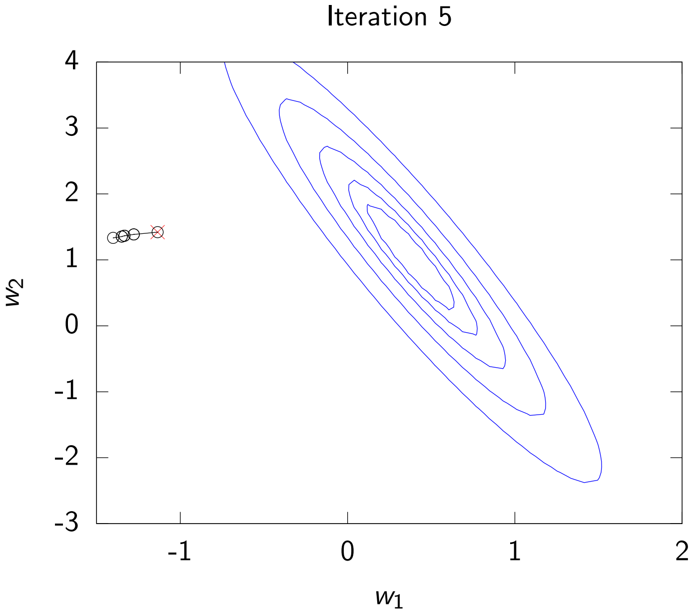


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

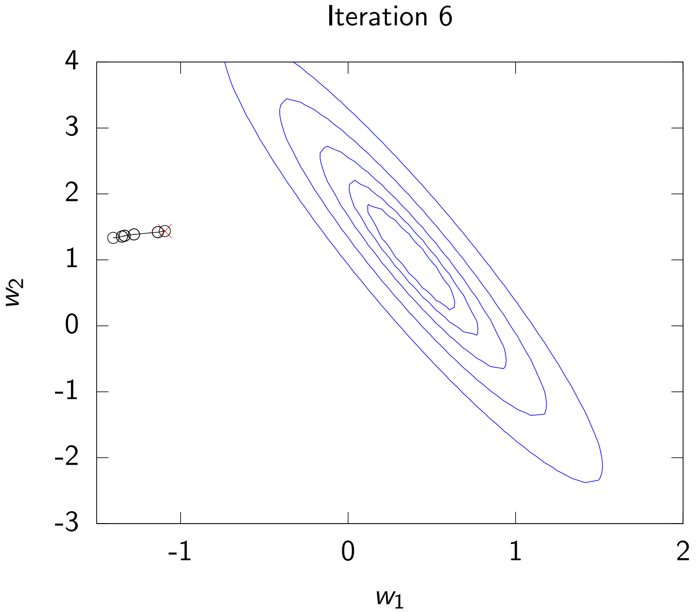


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

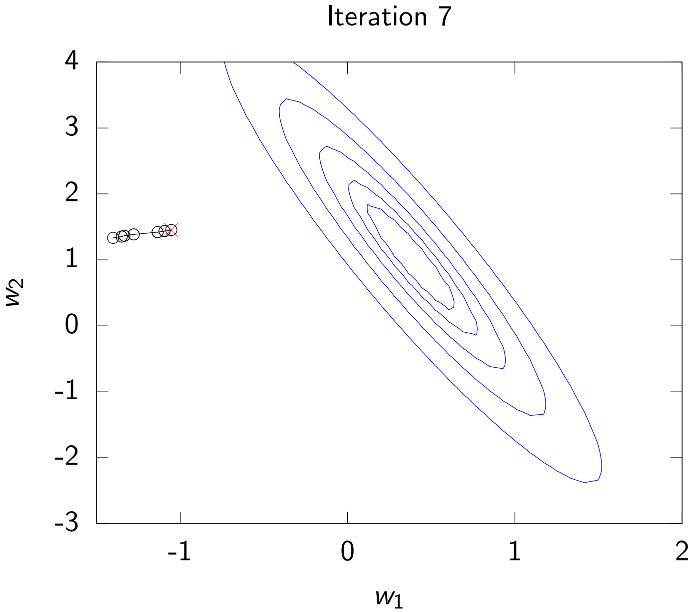


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

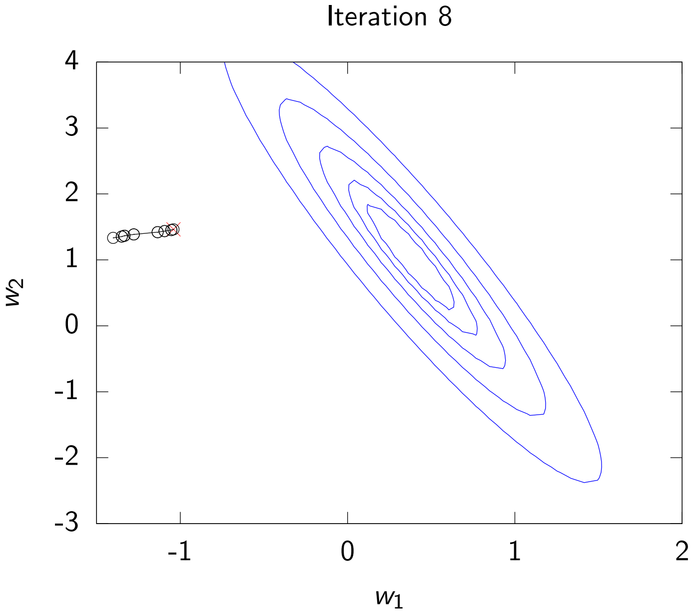


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

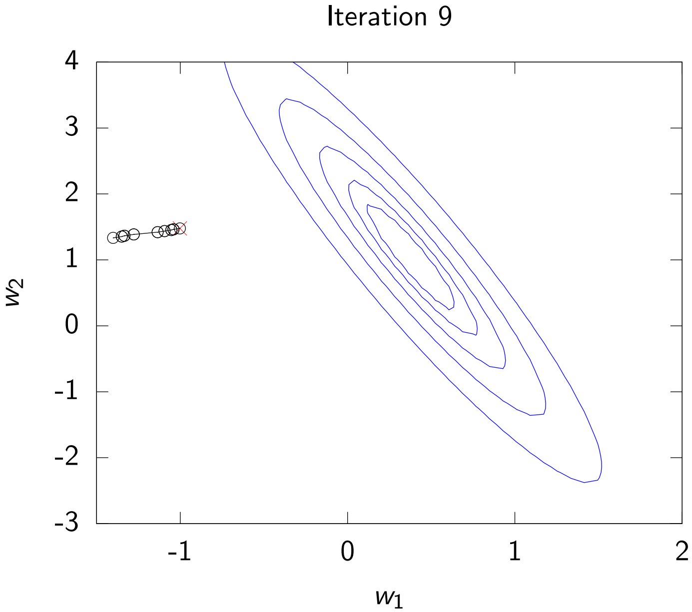


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

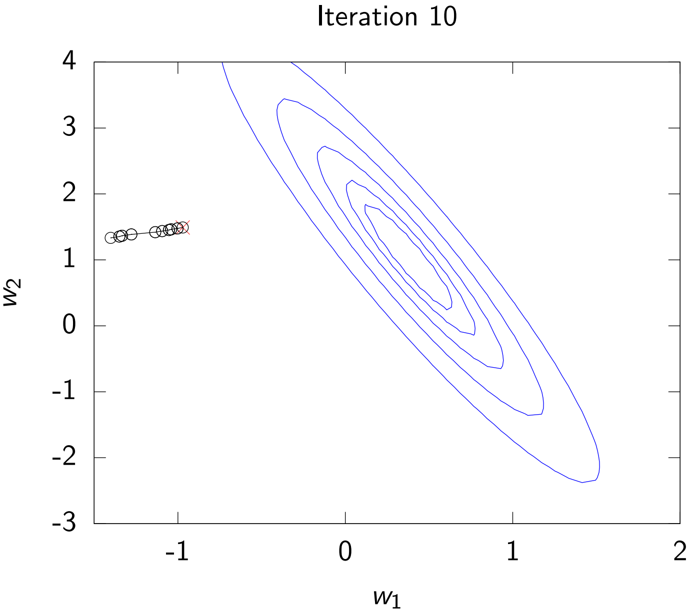


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

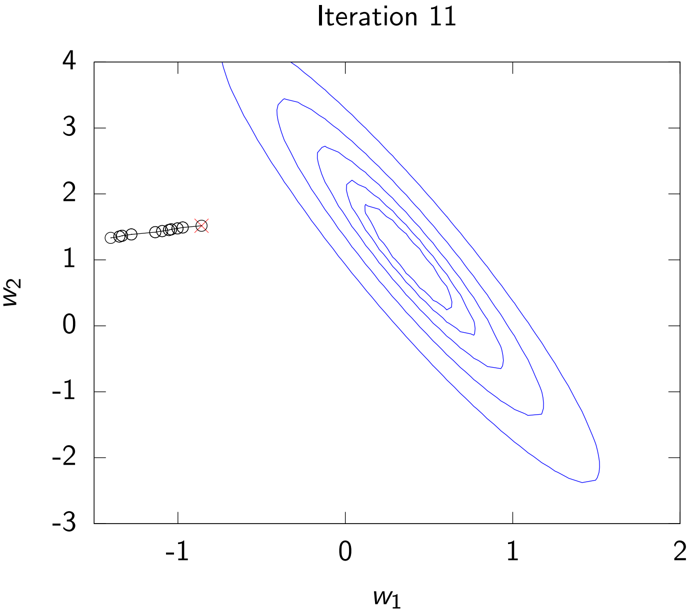


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

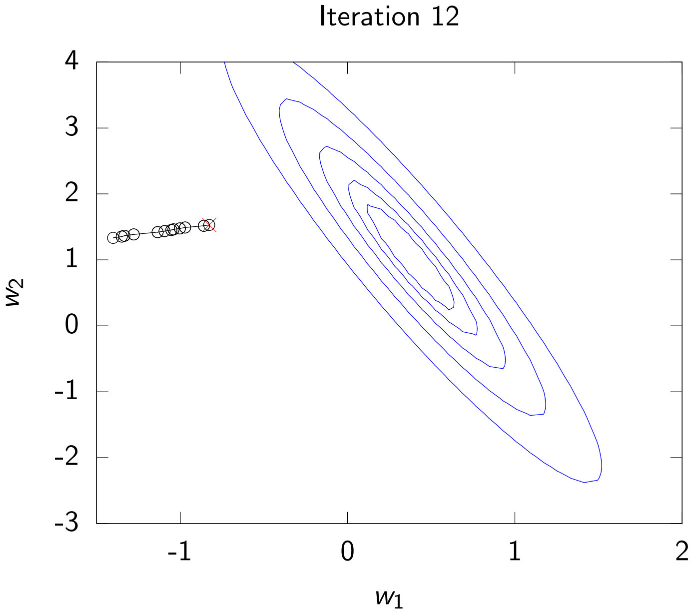


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

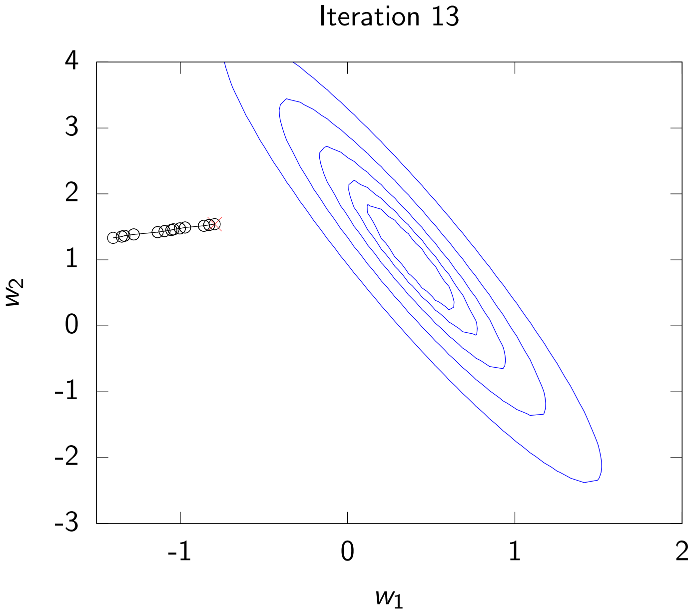


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

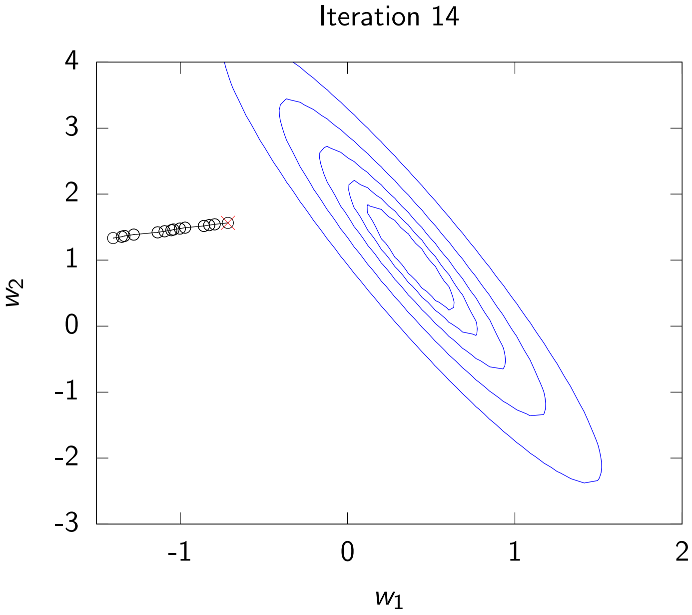


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

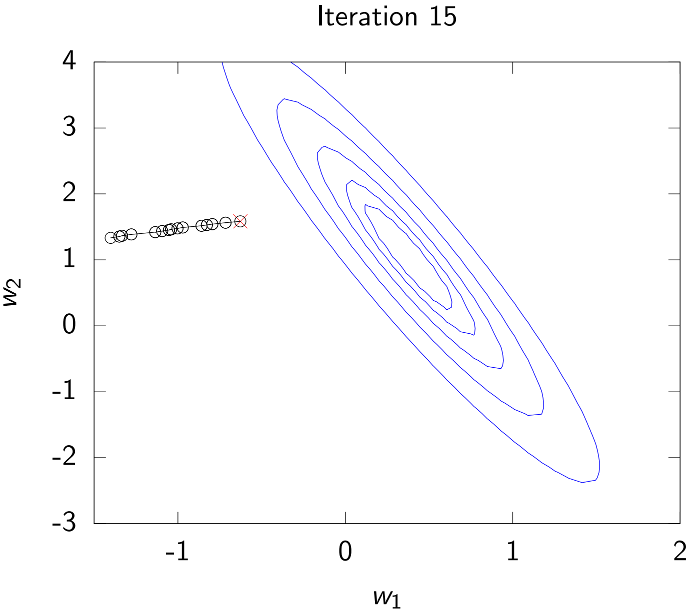


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

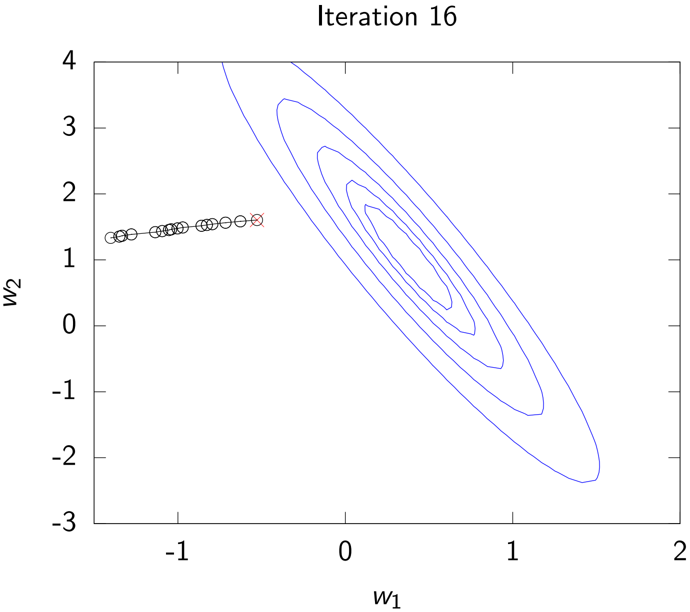


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

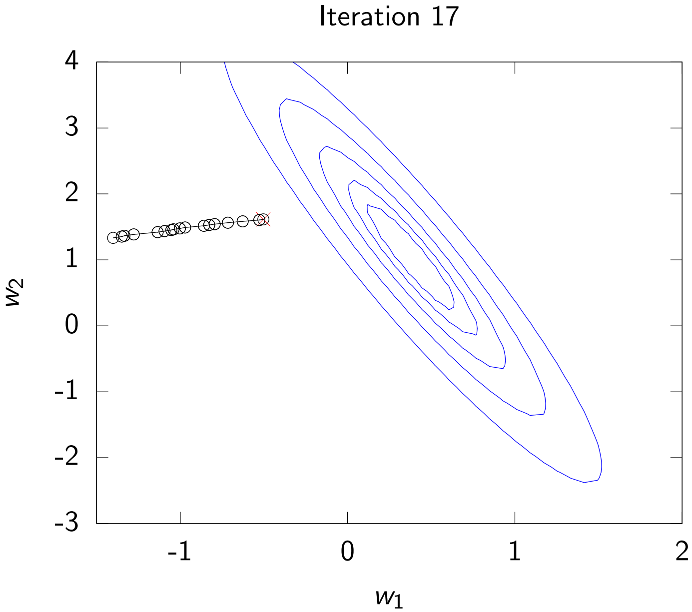


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

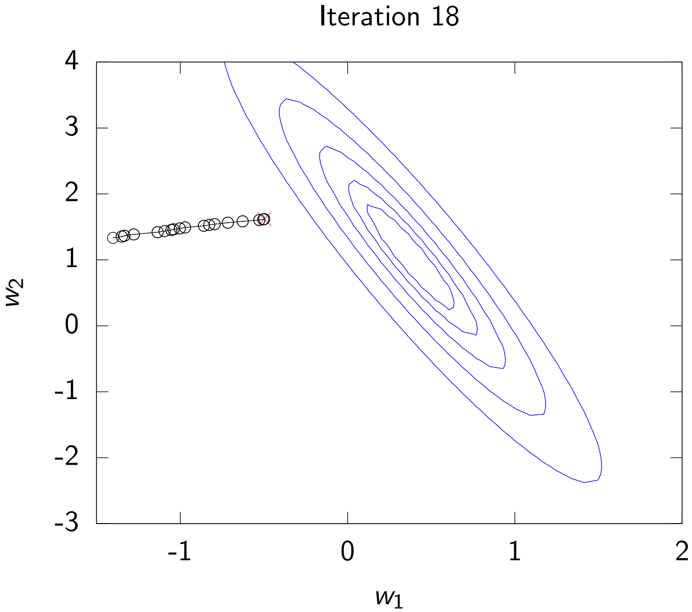


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

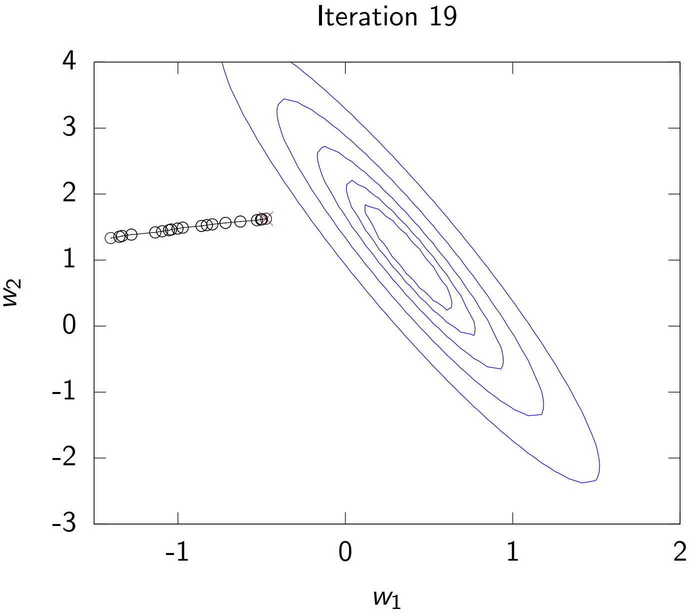


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

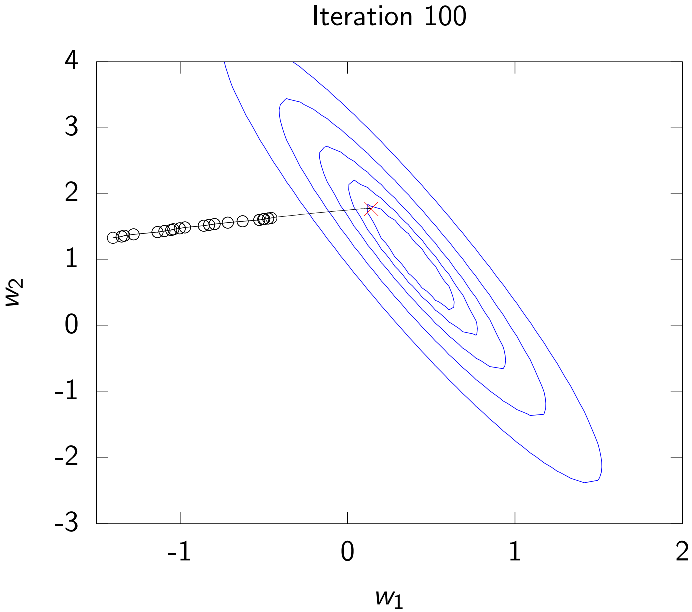


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

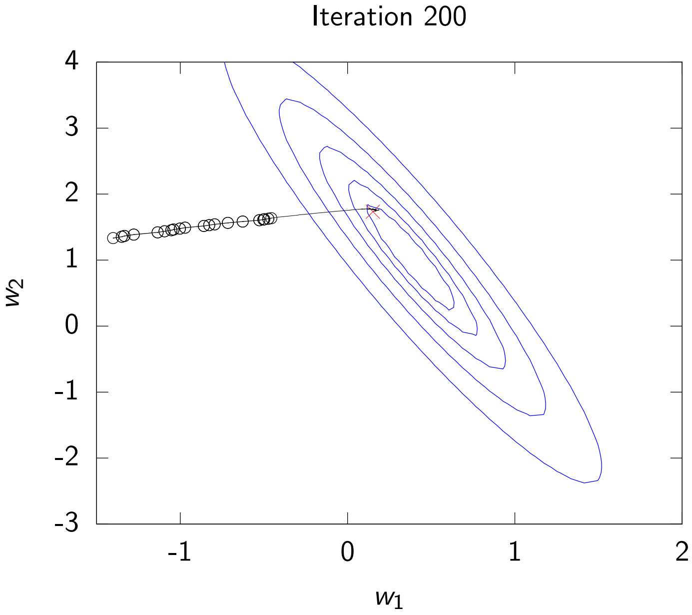


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

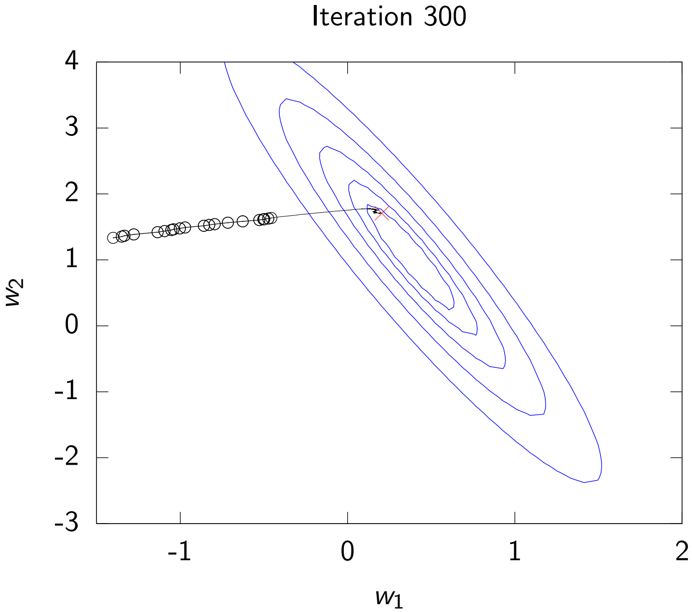


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

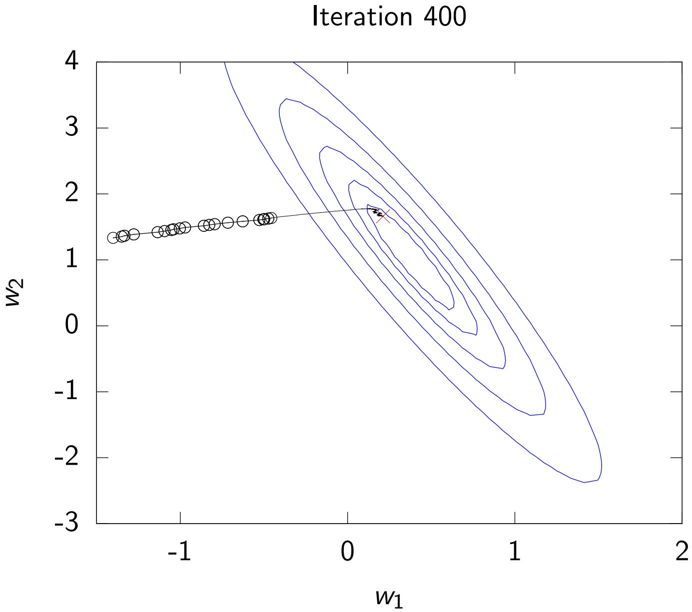


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

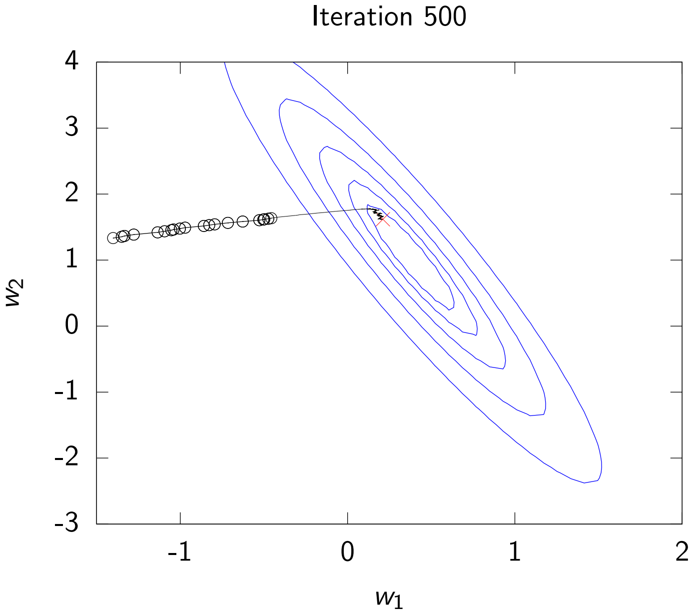


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

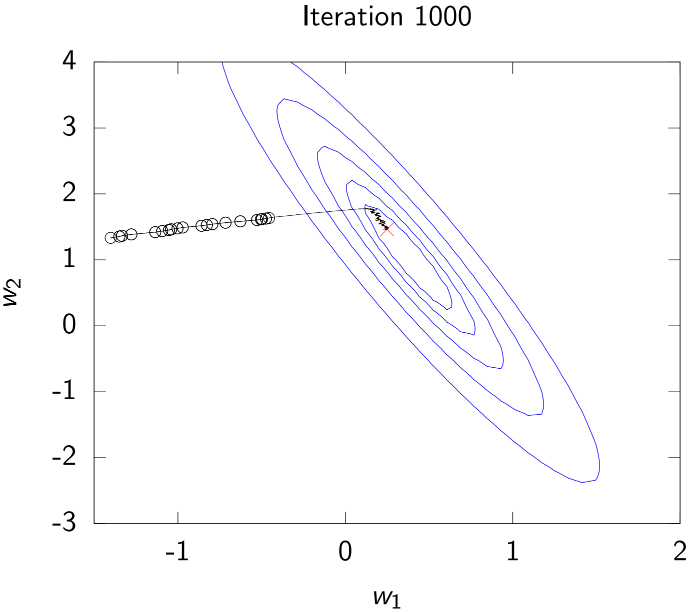


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

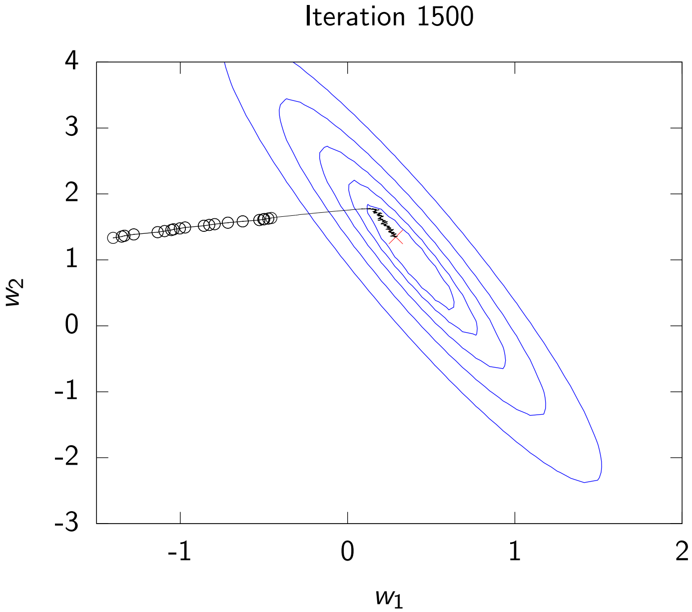


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

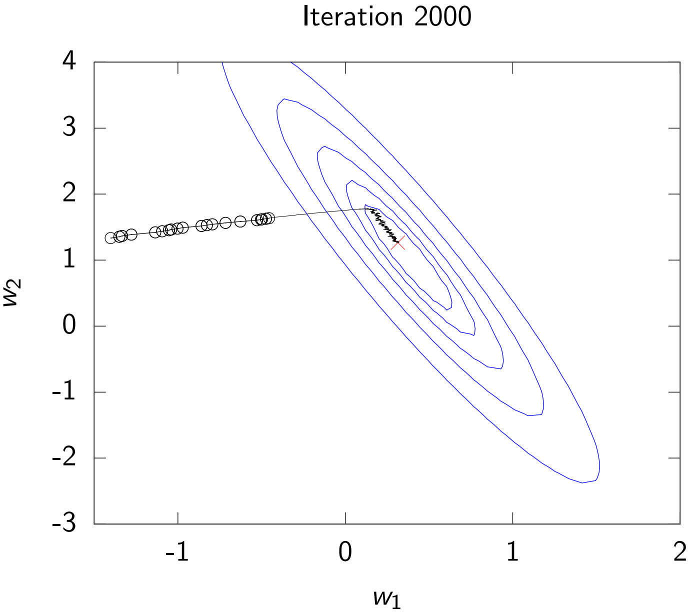


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

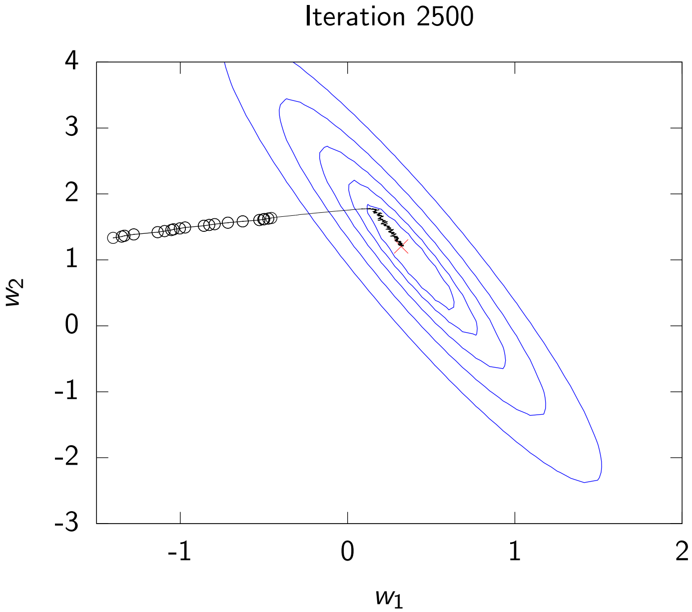


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

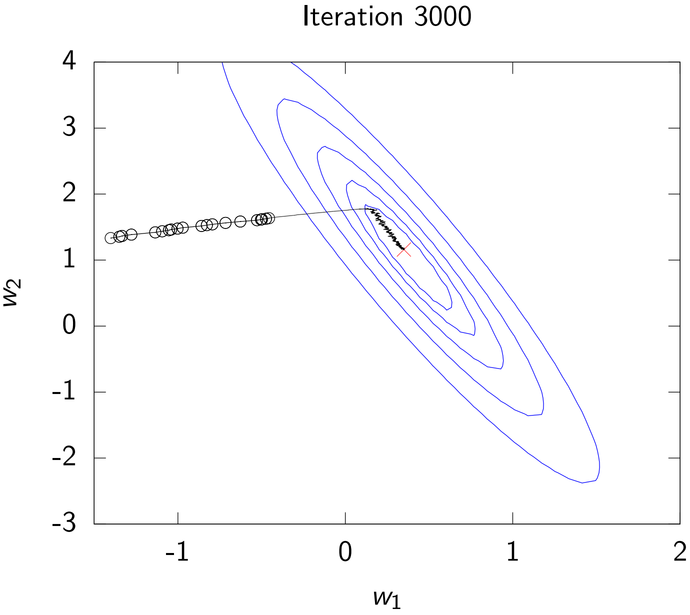


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

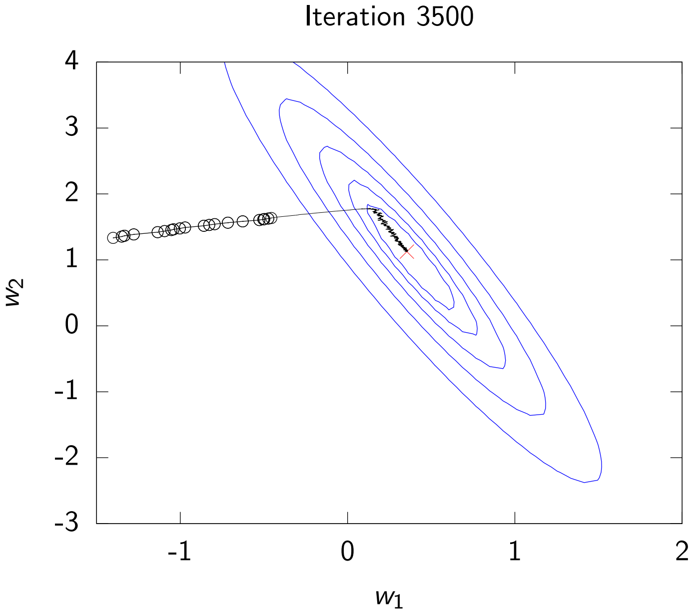


Figure: Stochastic gradient descent on a quadratic error surface.

Stochastic Gradient Descent

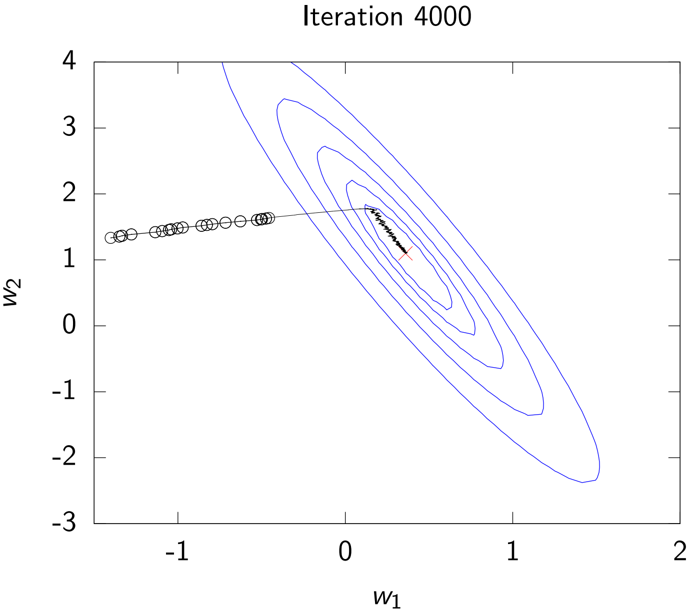


Figure: Stochastic gradient descent on a quadratic error surface.

Modern View of Error Functions

- Error function has a probabilistic interpretation (maximum likelihood).
- Error function is an actual loss function that you want to minimize (empirical risk minimization).
- For these interpretations probability and optimization theory become important.
- Much of the last 15 years of machine learning research has focused on probabilistic interpretations or clever relaxations of difficult objective functions.

Important Concepts Not Covered

- Optimization methods.
  - Second order methods, conjugate gradient, quasi-Newton and Newton.
  - Effective heuristics such as momentum.
- Local vs global solutions.

Outline

Outline

Clustering

- Divide data into discrete groups according to characteristics.
  - For example different animal species.
  - Different political parties.
- Determine the allocation to the groups and (harder) number of different groups.

K-means Clustering
An Algorithm

- *Require*: Set of K cluster centers & assignment of each point to a cluster.
    - Initialize cluster centers as data points.
    - Assign each data point to nearest cluster center.
    - Update each cluster center by setting it to the mean of assigned data points.

Objective Function

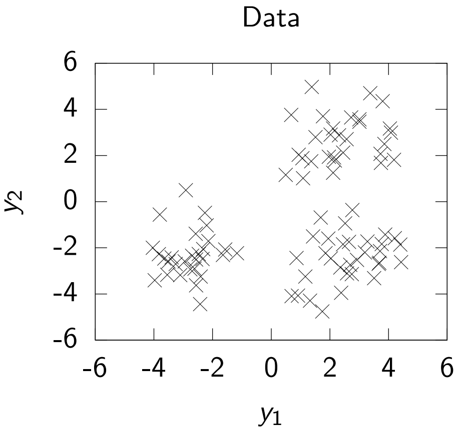
- This minimizes the objective:

$$\sum_{j=1}^{K} \sum_{i \text{ allocated to } j} \left(\mathbf{y}_{i,:} - \boldsymbol{\mu}_{j,:}\right)^\top \left(\mathbf{y}_{i,:} - \boldsymbol{\mu}_{j,:}\right)$$

- i.e. it minimizes the sum of Euclidean squared distances between points and their associated centers.
- The minimum is not guaranteed to be *global* or *unique*.
  - This objective is a non-convex optimization problem.

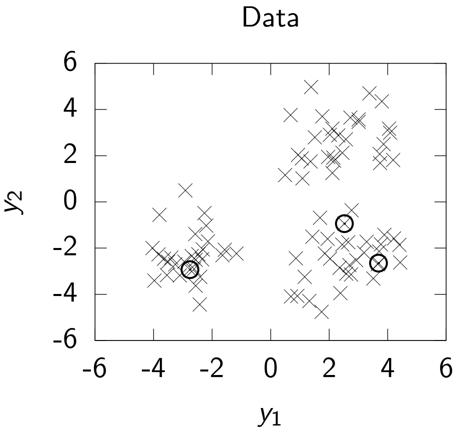
K-means Clustering

- K-means clustering.
  - Data set to be analyzed. Initialize cluster centers.



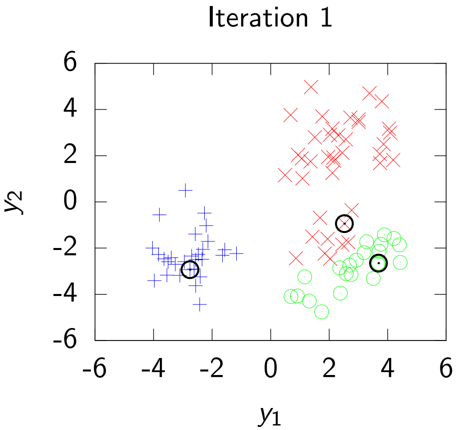
K-means Clustering

- K-means clustering.
  - Allocate each point to the cluster with the nearest center



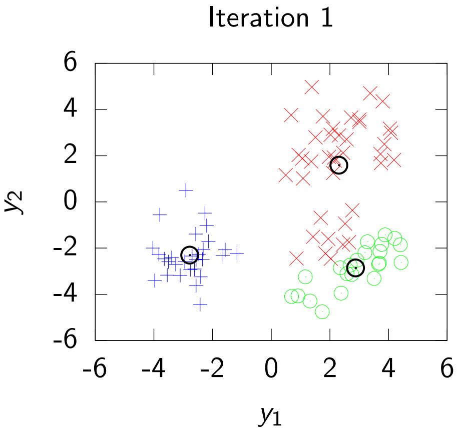
K-means Clustering

- K-means clustering.
  - Update each center by setting to the mean of the allocated points.



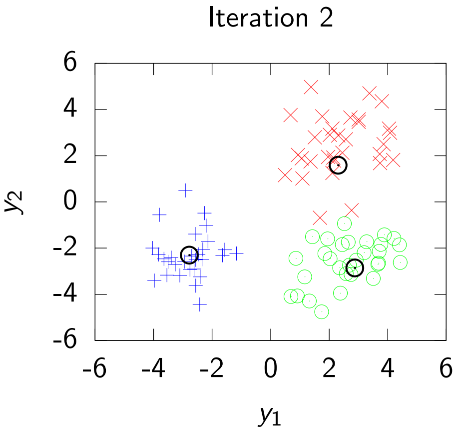
K-means Clustering

- K-means clustering.
  - Allocate each data point to the nearest cluster center.



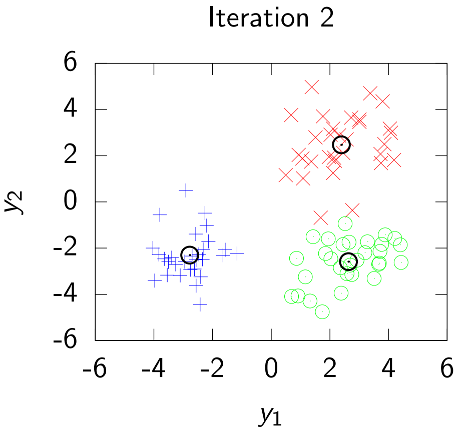
K-means Clustering

- K-means clustering.
  - Update each center by setting to the mean of the allocated points.



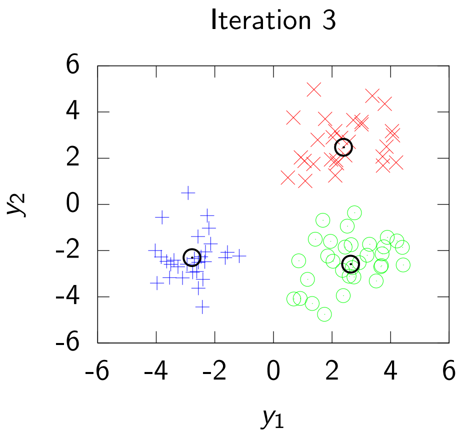
K-means Clustering

- K-means clustering.
  - Allocate each data point to the nearest cluster center.



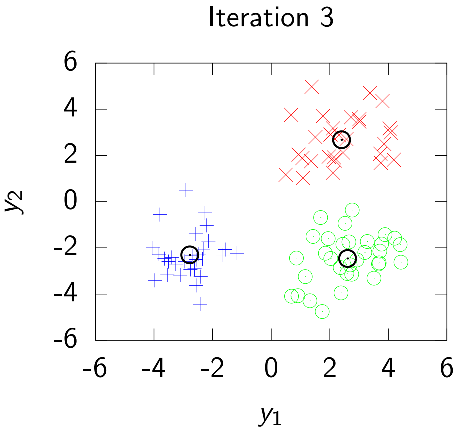
K-means Clustering

- K-means clustering.
  - Update each center by setting to the mean of the allocated points.



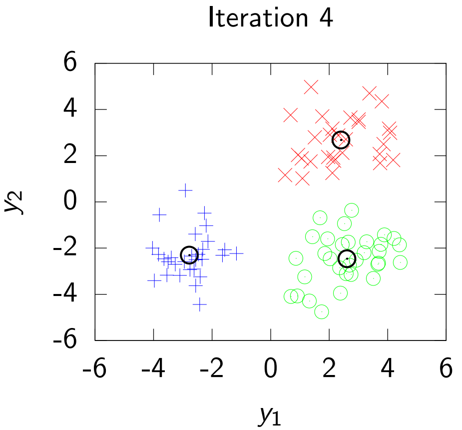
K-means Clustering

- K-means clustering.
  - Allocate each data point to the nearest cluster center.



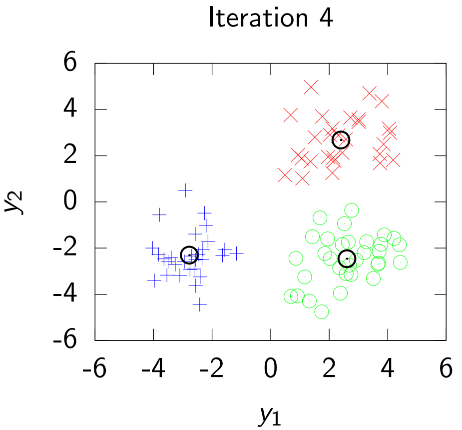
K-means Clustering

- K-means clustering.
  - Update each center by setting to the mean of the allocated points.



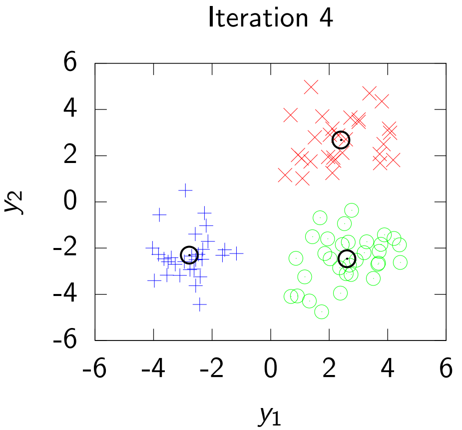
K-means Clustering

- K-means clustering.
  - Allocate each data point to the nearest cluster center.



K-means Clustering

- K-means clustering.
  - Allocation doesn't change so stop.



Other Clustering Approaches

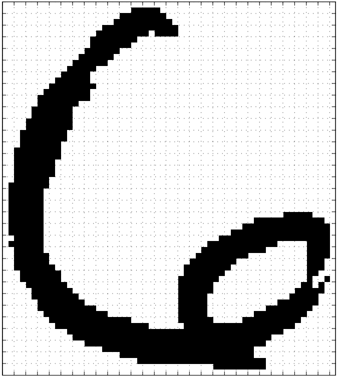
- Spectral clustering (Shi and Malik, 2000; Ng et al., 2002).
  - Allows clusters which aren't convex hulls.
- Dirichlet processes
  - A probabilistic formulation for a clustering algorithm that is non-parameteric.

Outline

High Dimensional Data

USPS Data Set Handwritten Digit

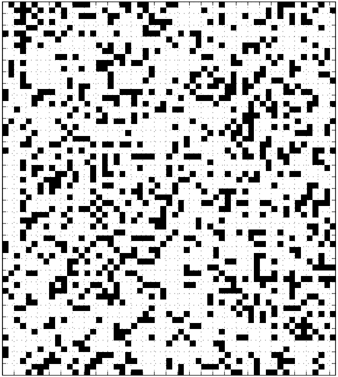
- 3648 Dimensions
- 64 rows by 57 columns



High Dimensional Data

USPS Data Set Handwritten Digit

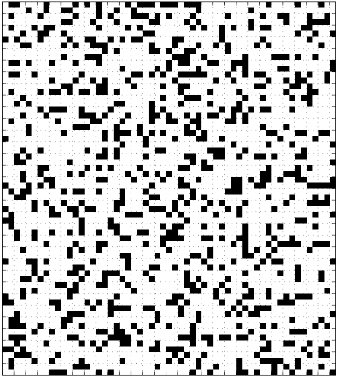
- 3648 Dimensions
- 64 rows by 57 columns
- Space contains more than just this digit.



High Dimensional Data

USPS Data Set Handwritten Digit

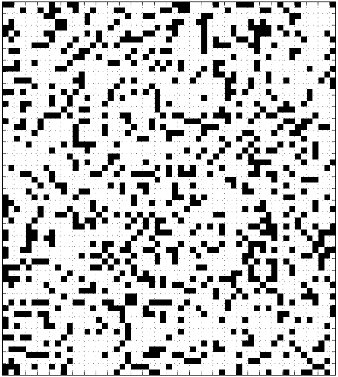
- 3648 Dimensions
- 64 rows by 57 columns
- Space contains more than just this digit.
- Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



High Dimensional Data

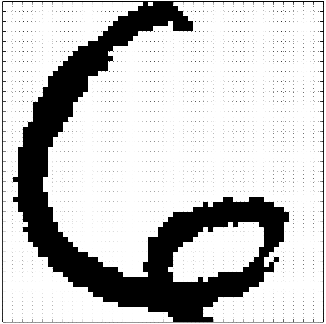
USPS Data Set Handwritten Digit

- 3648 Dimensions
- 64 rows by 57 columns
- Space contains more than just this digit.
- Even if we sample every nanosecond from now until the end of the universe, you won't see the original six!



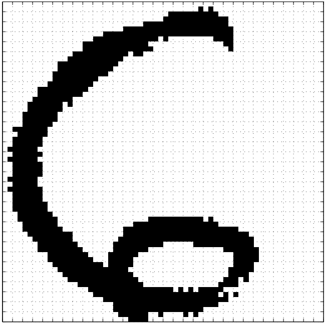
Simple Model of Digit

- Rotate a 'Prototype'



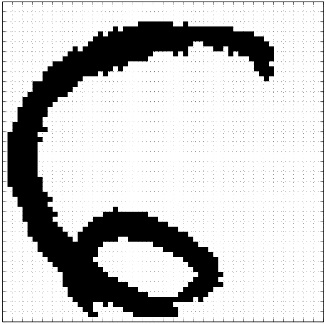
Simple Model of Digit

- Rotate a 'Prototype'



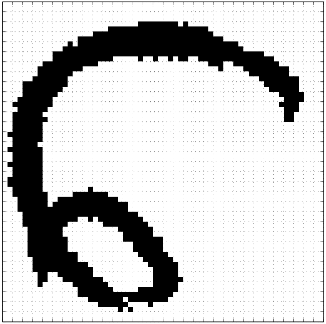
Simple Model of Digit

- Rotate a 'Prototype'



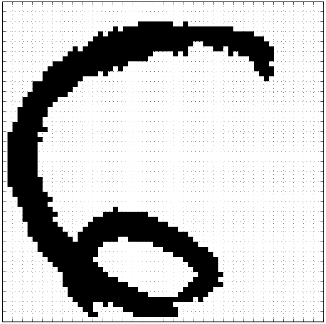
Simple Model of Digit

- Rotate a 'Prototype'



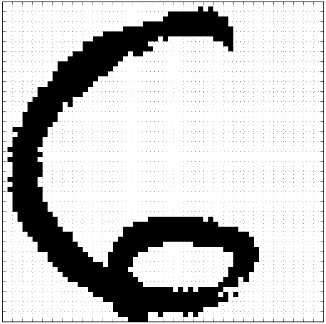
Simple Model of Digit

- Rotate a 'Prototype'



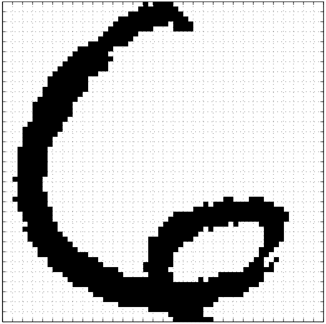
Simple Model of Digit

- Rotate a 'Prototype'



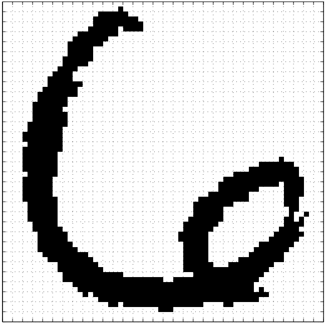
Simple Model of Digit

- Rotate a 'Prototype'



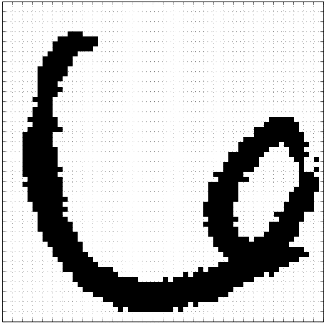
Simple Model of Digit

- Rotate a 'Prototype'



Simple Model of Digit

- Rotate a 'Prototype'

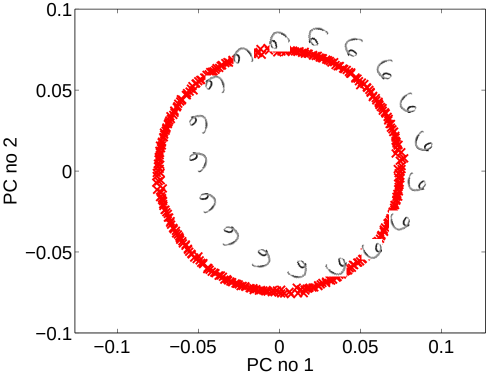


MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```

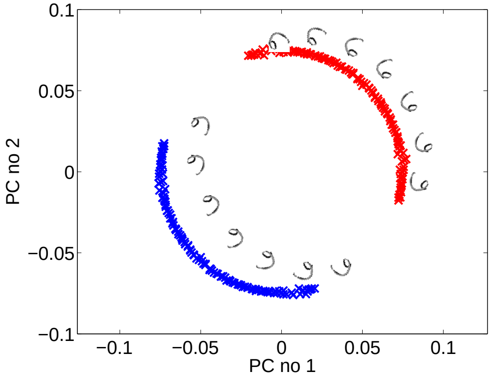
MATLAB Demo

```
demDigitsManifold([1 2], 'all')
```



MATLAB Demo

```
demDigitsManifold([1 2], 'sixnine')
```



Pure Rotation is too Simple

- In practice the data may undergo several distortions.
  - *e.g.* digits undergo 'thinning', translation and rotation.
- For data with 'structure':
- we expect fewer distortions than dimensions;
- we therefore expect the data to live on a lower dimensional manifold.
- Conclusion: deal with high dimensional data by looking for lower dimensional non-linear embedding.

Notation

q— dimension of latent/embedded space

p— dimension of data space

n— number of data points

data matrix, $\mathbf{Y} = [\mathbf{y}_{1,:}, \ldots, \mathbf{y}_{n,:}]^{\mathrm{T}} = [\mathbf{y}_{:,1}, \ldots, \mathbf{y}_{:,p}] \in \Re^{n \times p}$

latent variables, $\mathbf{X} = [\mathbf{x}_{1,:}, \ldots, \mathbf{x}_{n,:}]^{\mathrm{T}} = [\mathbf{x}_{:,1}, \ldots, \mathbf{x}_{:,q}] \in \Re^{n \times q}$

mapping matrix, $\mathbf{W} \in \Re^{p \times q}$

centering matrix, $\mathbf{H} = \mathbf{I} - n^{-1}\mathbf{1}\mathbf{1}^{\top} \in \Re^{n \times n}$

Reading Notation

- $\mathbf{a}_{i,:}$ is a vector from the ith row of a given matrix $\mathbf{A}$.
- $\mathbf{a}_{:,j}$ is a vector from the jth row of a given matrix $\mathbf{A}$.
- $\mathbf{X}$ and $\mathbf{Y}$ are *design matrices.*
- If we assume that the data matrix, $\mathbf{Y}$, is centered (i.e. has mean zero) then
  - Sample covariance given by

$$\mathbf{S} = n^{-1}\mathbf{Y}^{\top}\mathbf{Y}.$$

Data Representation

- Think of the data represented by interpoint distances.

$$d_{i,j} = \|\mathbf{y}_{i,:} - \mathbf{y}_{j,:}\|_2 = \sqrt{\left(\mathbf{y}_{i,:} - \mathbf{y}_{j,:}\right)^{\mathrm{T}}\left(\mathbf{y}_{i,:} - \mathbf{y}_{j,:}\right)}$$

- This is the Euclidean distance between any two data points.
- For any data set can display as a matrix, $\mathbf{D}$, where i, jth element is given by $d_{i,j}$.

Interpoint Distances for Rotated Sixes

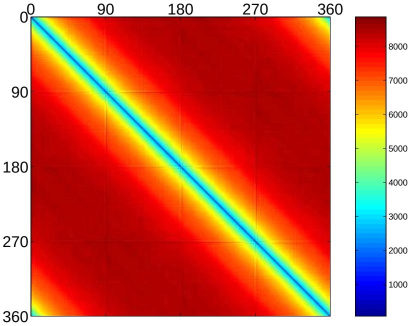


Figure: Interpoint distances for the rotated six data.

Multidimensional Scaling

- We want to find a low dimensional representation of the data.
- Find a configuration of points, $\mathbf{X}$, such that each

$$\delta_{i,j} = \left\|\mathbf{x}_{i,:} - \mathbf{x}_{j,:}\right\|_2$$

  closely matches the corresponding $d_{i,j}$ in the distance matrix.
- Need an objective function for matching the matrix of latent distances, which we denote $\boldsymbol{\Delta}$, to the matrix of observed distances, $\mathbf{D}$.

Feature Selection

- A possible error function:
  - An entrywise L_1 norm on difference between squared distances

$$E(\mathbf{X}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \left| d_{i,j}^2 - \delta_{i,j}^2 \right| .$$

- A possible dimensionality reduction algorithm:
  - Retain q columns of $\mathbf{Y}$ which minimize the error.
- To minimize $E(\mathbf{Y})$ we need to retain for $\mathbf{X}$ the columns of $\mathbf{Y}$ that have the largest variance.

Feature Selection

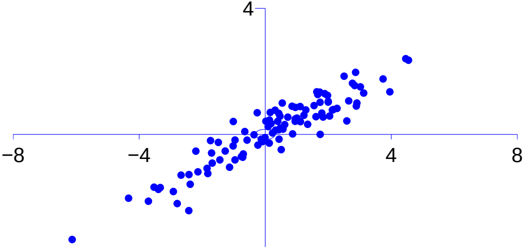


Figure: Feature selection via distance preservation.

Feature Selection

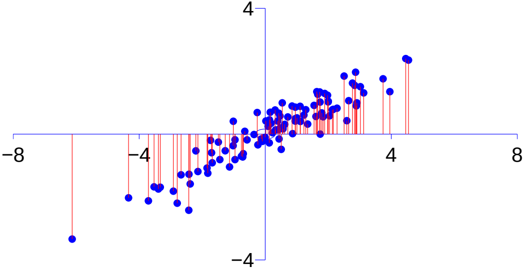


Figure: Feature selection via distance preservation.

Feature Selection

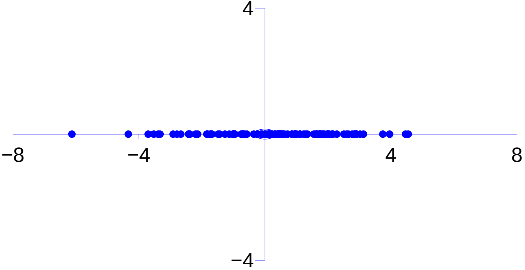
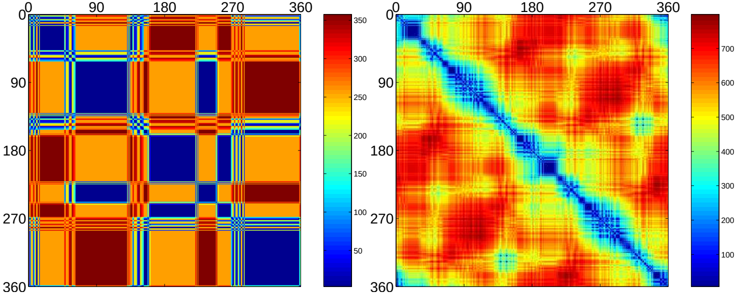


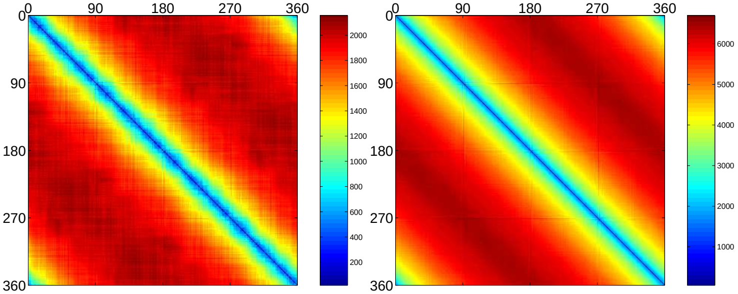
Figure: Feature selection via distance preservation.

Reconstruction from Latent Space



Left: distances reconstructed with two dimensions. *Right*: distances reconstructed with 10 dimensions.

Reconstruction from Latent Space



Left: distances reconstructed with 100 dimensions. *Right*: distances reconstructed with 1000 dimensions.

Considering Rotations

- Extracting only columns of data is a very simple approach to dimensionality reduction.
- We can extend our approach by considering rotations of the data before we take the columns.

Feature Extraction

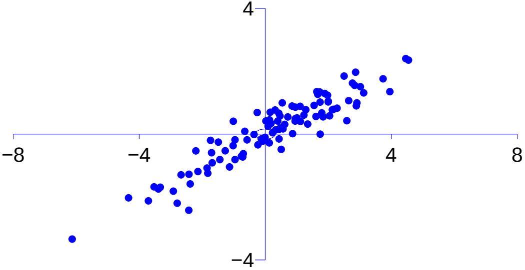


Figure: Rotation preserves interpoint distances. .

Feature Extraction

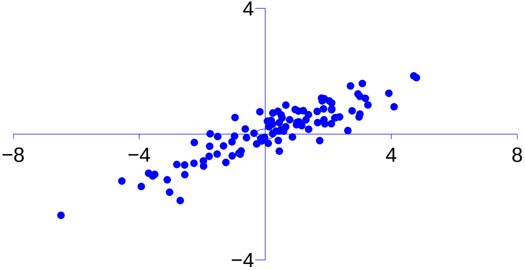


Figure: Rotation preserves interpoint distances. .

Feature Extraction

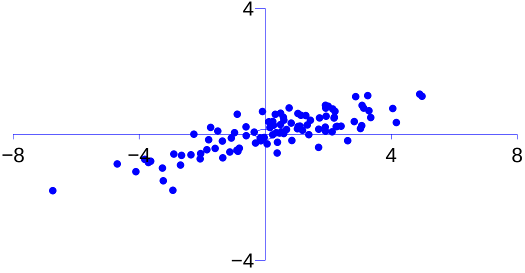


Figure: Rotation preserves interpoint distances. .

Feature Extraction

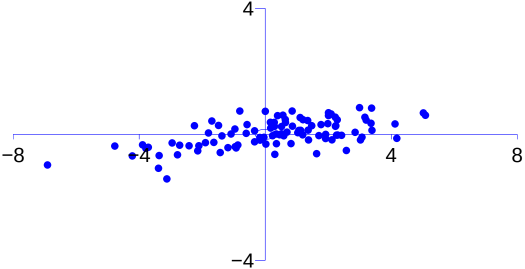


Figure: Rotation preserves interpoint distances. .

Feature Extraction

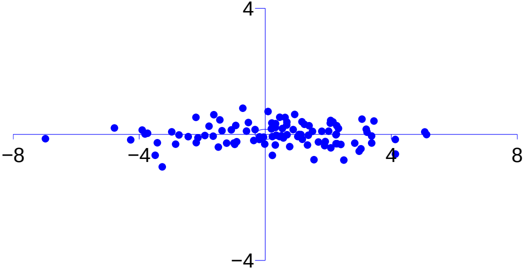


Figure: Rotation preserves interpoint distances. .

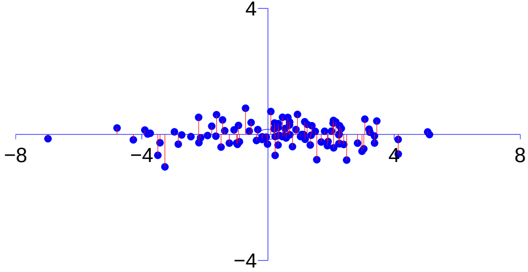


Figure: Rotation preserves interpoint distances. Residuals are much reduced.

Feature Extraction

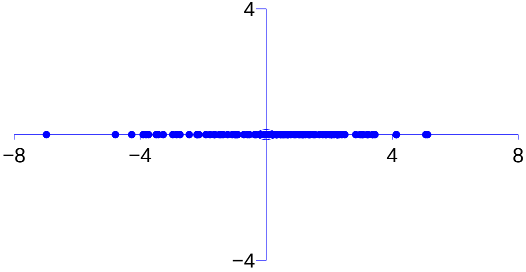


Figure: Rotation preserves interpoint distances. Residuals are much reduced.

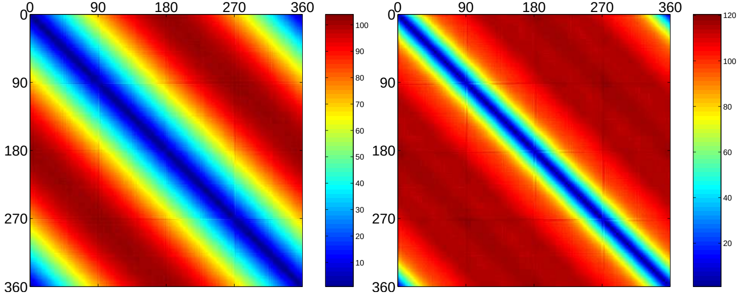
Which Rotation?

- We need the rotation that will minimise residual error.
- Discard direction with *maximum variance*.
- Error is then given by the sum of residual variances.

$$E(\mathbf{X}) \propto \sum_{k=q+1}^{p} \sigma_k^2.$$

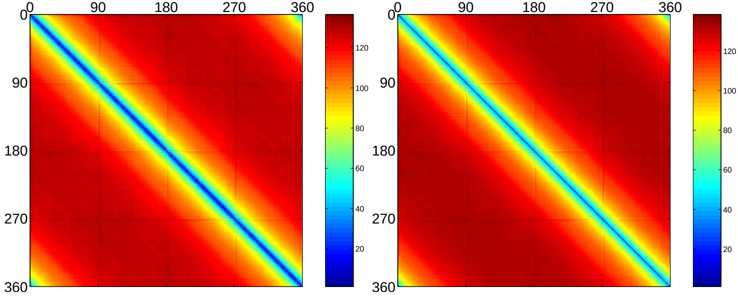
- Rotations of data matrix *do not* effect this analysis.
- Algorithm:
  - Rotate data to find directions of maximum variance.
  - Retain these directions for the low dimensional representation.

Rotation Reconstruction from Latent Space



Left: distances reconstructed with two dimensions. *Right*: distances reconstructed with 10 dimensions.

Rotation Reconstruction from Latent Space



Left: distances reconstructed with 100 dimensions. *Right*: distances reconstructed with 360 dimensions.

Outline

Principal Component Analysis

- How do we find these directions?
- Rotate to find directions in data with maximal variance.
  - This is known as PCA (Hotelling, 1933).
- Rotate data to extract directions of maximum variance.
- Do this by diagonalizing the sample covariance matrix

$$\mathbf{S} = n^{-1}\mathbf{Y}^\top\mathbf{Y}.$$

Principal Component Analysis

- Find a direction in the data, $\mathbf{x}_{:,1} = \mathbf{Y}\mathbf{r}_1$, for which variance is maximized.

$$\begin{aligned} \mathbf{r}_1 =& \text{argmax}_{\mathbf{r}_1} \operatorname{var}\left(\mathbf{Y}\mathbf{r}_1\right) \\ \text{subject to:} \quad & \mathbf{r}_1^\top \mathbf{r}_1 = 1 \end{aligned}$$

- Can rewrite in terms of sample covariance

$$\text{var}\left(\mathbf{x}_{:,1}\right) = n^{-1}\left(\mathbf{Y}\mathbf{r}_1\right)^\top \mathbf{Y}\mathbf{r}_1 = \mathbf{r}_1^\top \underbrace{\left(n^{-1}\mathbf{Y}^\top\mathbf{Y}\right)}_{\text{sample covariance}} \mathbf{r}_1 = \mathbf{r}_1^\top \mathbf{S}\mathbf{r}_1$$

Lagrangian

- Solution via constrained optimisation (Lagrange multipliers):

$$L(\mathbf{r}_1, \lambda_1) = \mathbf{r}_1^\top \mathbf{S}\mathbf{r}_1 + \lambda_1 \left(1 - \mathbf{r}_1^\top \mathbf{r}_1\right)$$

- Gradient with respect to $\mathbf{r}_1$

$$\frac{\mathrm{d}L(\mathbf{r}_1, \lambda_1)}{\mathrm{d}\mathbf{r}_1} = 2\mathbf{S}\mathbf{r}_1 - 2\lambda_1 \mathbf{r}_1$$

  rearrange to form

$$\mathbf{S}\mathbf{r}_1 = \lambda_1 \mathbf{r}_1.$$

  Which is recognised as an eigenvalue problem.

- Further directions can also be shown to be eigenvectors of the covariance.

Conclusions

- Machine learning has slightly different roots from statistics.
- Has inspiration from psychology and computer science.
- Modern machine learning is more mathematically motivated.
- Many of the modern challenges are strongly related to statistics.
- Personal view: we can benefit greatly by more interaction with cognitive science.

References I

J. A. Anderson and E. Rosenfeld, editors. *Neurocomputing: Foundations of Research*, Cambridge, MA, 1988. MIT Press.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press, London, 1979. [Google Books] .

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943. Reprinted in Anderson and Rosenfeld (1988).

A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14, Cambridge, MA, 2002. MIT Press.

F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Spartan, 1962.

References II

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8): 888–905, 2000.

V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.

Outline

PCA Further Directions

Lagrange Multiplier

- Recall the gradient,

$$\frac{\mathrm{d}L\left(\mathbf{r}_1, \lambda_1\right)}{\mathrm{d}\mathbf{r}_1} = 2\mathbf{S}\mathbf{r}_1 - 2\lambda_1\mathbf{r}_1 \tag{2}$$

  to find λ_1 premultiply (2) by $\mathbf{r}_1^\top$ and rearrange giving

$$\lambda_1 = \mathbf{r}_1^\top \mathbf{S}\mathbf{r}_1.$$

- Maximum variance is therefore *necessarily* the maximum eigenvalue of $\mathbf{S}$.
- This is the *first principal component.*

Further Directions

- Find orthogonal directions to earlier extracted directions with maximal variance.
- Orthogonality constraints, for $j < k$ we have

$$\mathbf{r}_j^\top \mathbf{r}_k = \mathbf{0} \quad \mathbf{r}_k^\top \mathbf{r}_k = 1$$

- Lagrangian

$$L\left(\mathbf{r}_k, \lambda_k, \gamma\right) = \mathbf{r}_k^\top \mathbf{S} \mathbf{r}_k + \lambda_k \left(1 - \mathbf{r}_k^\top \mathbf{r}_k\right) + \sum_{j=1}^{k-1} \gamma_j \mathbf{r}_j^\top \mathbf{r}_k$$

$$\frac{\mathrm{d}\left(\mathbf{r}_k, \lambda_k\right)}{\mathrm{d}\mathbf{r}_k} = 2\mathbf{S}\mathbf{r}_k - 2\lambda_k \mathbf{r}_k + \sum_{j=1}^{k-1} \gamma_j \mathbf{r}_j$$

Further Eigenvectors

- Gradient of Lagrangian:

$$\frac{\mathrm{d}L\left(\mathbf{r}_k, \lambda_k\right)}{\mathrm{d}\mathbf{r}_k} = 2\mathbf{S}\mathbf{r}_k - 2\lambda_k\mathbf{r}_k + \sum_{j=1}^{k-1}\gamma_j\mathbf{r}_j \tag{3}$$

- Premultipling (3) by $\mathbf{r}_i$ with $i < k$ implies

$$\gamma_i = 0$$

  which allows us to write

$$\mathbf{S}\mathbf{r}_k = \lambda_k\mathbf{r}_k.$$

- Premultiplying (3) by $\mathbf{r}_k$ implies

$$\lambda_k = \mathbf{r}_k^\top\mathbf{S}\mathbf{r}_k.$$

- This is the *kth principal component*.

Principal Coordinates Analysis

- The rotation which finds directions of maximum variance is the eigenvectors of the covariance matrix.
- The variance in each direction is given by the eigenvalues.
- **Problem:** working directly with the sample covariance, $\mathbf{S}$, may be impossible.
- For example: perhaps we are given distances between data points, but not absolute locations.
  - No access to absolute positions: cannot compute original sample covariance.

An Alternative Formalism

- Matrix representation of eigenvalue problem for first q eigenvectors.

$$\mathbf{Y}^\top \mathbf{Y}\mathbf{R}_q = \mathbf{R}_q \boldsymbol{\Lambda}_q \quad \mathbf{R}_q \in \Re^{p\times q} \tag{4}$$

- Premultiply by $\mathbf{Y}$:

$$\mathbf{Y}\mathbf{Y}^\top \mathbf{Y}\mathbf{R}_q = \mathbf{Y}\mathbf{R}_q \boldsymbol{\Lambda}_q$$

- Postmultiply by $\boldsymbol{\Lambda}_q^{-\frac{1}{2}}$

$$\mathbf{Y}\mathbf{Y}^\top \mathbf{Y}\mathbf{R}_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}} = \mathbf{Y}\mathbf{R}_q \boldsymbol{\Lambda}_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

An Alternative Formalism

- Matrix representation of eigenvalue problem for first q eigenvectors.

$$\mathbf{Y}^\top\mathbf{Y}\mathbf{R}_q = \mathbf{R}_q\boldsymbol{\Lambda}_q \quad \mathbf{R}_q \in \Re^{p\times q} \tag{4}$$

- Premultiply by $\mathbf{Y}$:

$$\mathbf{Y}\mathbf{Y}^\top\mathbf{Y}\mathbf{R}_q = \mathbf{Y}\mathbf{R}_q\boldsymbol{\Lambda}_q$$

- Postmultiply by $\boldsymbol{\Lambda}_q^{-\frac{1}{2}}$

$$\mathbf{Y}\mathbf{Y}^\top\mathbf{Y}\mathbf{R}_q\boldsymbol{\Lambda}_q^{-\frac{1}{2}} = \mathbf{Y}\mathbf{R}_q\boldsymbol{\Lambda}_q^{-\frac{1}{2}}\boldsymbol{\Lambda}_q$$

An Alternative Formalism

- Matrix representation of eigenvalue problem for first q eigenvectors.

$$\mathbf{Y}^\top \mathbf{Y} \mathbf{R}_q = \mathbf{R}_q \boldsymbol{\Lambda}_q \quad \mathbf{R}_q \in \Re^{p \times q} \tag{4}$$

- Premultiply by $\mathbf{Y}$:

$$\mathbf{Y}\mathbf{Y}^\top \mathbf{Y} \mathbf{R}_q = \mathbf{Y}\mathbf{R}_q \boldsymbol{\Lambda}_q$$

- Postmultiply by $\boldsymbol{\Lambda}_q^{-\frac{1}{2}}$

$$\mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \mathbf{U}_q \boldsymbol{\Lambda}_q \quad \mathbf{U}_q = \mathbf{Y}\mathbf{R}_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \boldsymbol{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \mathbf{Y}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{Y}\mathbf{R}_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \mathbf{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \left(\mathbf{Y}^\top \mathbf{Y}\right)^2 \mathbf{R}_q \mathbf{\Lambda}_q^{-\frac{1}{2}}$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \boldsymbol{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \left(\mathbf{Y}^\top \mathbf{Y}\right)^2 \mathbf{R}_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

- Full eigendecomposition of sample covariance

$$\mathbf{Y}^\top \mathbf{Y} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^\top$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \boldsymbol{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \left(\mathbf{Y}^\top \mathbf{Y}\right)^2 \mathbf{R}_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

- Full eigendecomposition of sample covariance

$$\mathbf{Y}^\top \mathbf{Y} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^\top$$

- Implies that

$$\left(\mathbf{Y}^\top \mathbf{Y}\right)^2 = \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^\top \mathbf{R}\boldsymbol{\Lambda}\mathbf{R}^\top = \mathbf{R}\boldsymbol{\Lambda}^2\mathbf{R}^\top.$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \boldsymbol{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \mathbf{R} \boldsymbol{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

- Full eigendecomposition of sample covariance

$$\mathbf{Y}^\top \mathbf{Y} = \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^\top$$

- Implies that

$$\left(\mathbf{Y}^\top \mathbf{Y}\right)^2 = \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^\top \mathbf{R} \boldsymbol{\Lambda} \mathbf{R}^\top = \mathbf{R} \boldsymbol{\Lambda}^2 \mathbf{R}^\top.$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \mathbf{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \mathbf{R} \mathbf{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q \mathbf{\Lambda}_q^{-\frac{1}{2}}$$

- Product of the first q eigenvectors with the rest,

$$\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \in \Re^{p \times q}$$

where we have used $\mathbf{I}_q$ to denote a $q \times q$ identity matrix.

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \mathbf{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \mathbf{R} \mathbf{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q \mathbf{\Lambda}_q^{-\frac{1}{2}}$$

- Product of the first q eigenvectors with the rest,

$$\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \in \Re^{p \times q}$$

where we have used $\mathbf{I}_q$ to denote a $q \times q$ identity matrix.

- Premultiplying by eigenvalues gives,

$$\mathbf{\Lambda} \mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{\Lambda}_q \\ \mathbf{0} \end{bmatrix}$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \mathbf{\Lambda}_q^{-\frac{1}{2}} \mathbf{R}_q^\top \mathbf{R} \mathbf{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q \mathbf{\Lambda}_q^{-\frac{1}{2}}$$

- Product of the first q eigenvectors with the rest,

$$\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \in \Re^{p \times q}$$

  where we have used $\mathbf{I}_q$ to denote a $q \times q$ identity matrix.
- Premultiplying by eigenvalues gives,

$$\mathbf{\Lambda} \mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{\Lambda}_q \\ \mathbf{0} \end{bmatrix}$$

- Multiplying by self transpose gives

$$\mathbf{R}_q^\top \mathbf{R} \mathbf{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q = \mathbf{\Lambda}_q^2$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \boldsymbol{\Lambda}_q^{-\frac{1}{2}} \left[\mathbf{R}_q^\top \mathbf{R}\boldsymbol{\Lambda}^2\mathbf{R}^\top \mathbf{R}_q\right] \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

- Product of the first q eigenvectors with the rest,

$$\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \in \Re^{p\times q}$$

  where we have used $\mathbf{I}_q$ to denote a $q \times q$ identity matrix.
- Premultiplying by eigenvalues gives,

$$\boldsymbol{\Lambda}\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \boldsymbol{\Lambda}_q \\ \mathbf{0} \end{bmatrix}$$

- Multiplying by self transpose gives

$$\mathbf{R}_q^\top \mathbf{R}\boldsymbol{\Lambda}^2\mathbf{R}^\top \mathbf{R}_q = \boldsymbol{\Lambda}_q^2$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \boldsymbol{\Lambda}_q^{-\frac{1}{2}} \left[\mathbf{R}_q^\top \mathbf{R}\boldsymbol{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q\right] \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

- Product of the first q eigenvectors with the rest,

$$\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \in \Re^{p\times q}$$

  where we have used $\mathbf{I}_q$ to denote a $q \times q$ identity matrix.

- Premultiplying by eigenvalues gives,

$$\boldsymbol{\Lambda}\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \boldsymbol{\Lambda}_q \\ \mathbf{0} \end{bmatrix}$$

- Multiplying by self transpose gives

$$\mathbf{R}_q^\top \mathbf{R}\boldsymbol{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q = \boldsymbol{\Lambda}_q^2$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \boldsymbol{\Lambda}_q^{-\frac{1}{2}} \boldsymbol{\Lambda}_q^2 \boldsymbol{\Lambda}_q^{-\frac{1}{2}}$$

- Product of the first q eigenvectors with the rest,

$$\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \in \Re^{p \times q}$$

  where we have used $\mathbf{I}_q$ to denote a $q \times q$ identity matrix.

- Premultiplying by eigenvalues gives,

$$\boldsymbol{\Lambda}\mathbf{R}^\top \mathbf{R}_q = \begin{bmatrix} \boldsymbol{\Lambda}_q \\ \mathbf{0} \end{bmatrix}$$

- Multiplying by self transpose gives

$$\mathbf{R}_q^\top \mathbf{R}\boldsymbol{\Lambda}^2 \mathbf{R}^\top \mathbf{R}_q = \boldsymbol{\Lambda}_q^2$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{U}_q^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{U}_q = \mathbf{\Lambda}_q$$

$\mathbf{U}_q$ Diagonalizes the Inner Product Matrix

- Need to prove that $\mathbf{U}_q$ are eigenvectors of inner product matrix.

$$\mathbf{Y}\mathbf{Y}^\top\mathbf{U}_q = \mathbf{U}_q\boldsymbol{\Lambda}_q$$

Equivalent Eigenvalue Problems

- Two eigenvalue problems are equivalent. One solves for the rotation, the other solves for the location of the rotated points.
- When $p < n$ it is easier to solve for the rotation, $\mathbf{R}_q$. But when $p > n$ we solve for the embedding (principal coordinate analysis).
- In MDS we may not know $\mathbf{Y}$, cannot compute $\mathbf{Y}^\top\mathbf{Y}$ from distance matrix.
- Can we compute $\mathbf{Y}\mathbf{Y}^\top$ instead?

The Covariance Interpretation

- $n^{-1}\mathbf{Y}^\top\mathbf{Y}$ is the data covariance.
- $\mathbf{Y}\mathbf{Y}^\top$ is a centred inner product matrix.
  - Also has an interpretation as a covariance matrix (Gaussian processes).
  - It expresses correlation and anti correlation between *data points*.
  - Standard covariance expresses correlation and anti correlation between *data dimensions*.

Distance to Similarity: A Gaussian Covariance Interpretation

- Translate between covariance and distance.
  - Consider a vector sampled from a zero mean Gaussian distribution,
$$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}).$$
  - Expected square distance between two elements of this vector is
$$d_{i,j}^2 = \left\langle (z_i - z_j)^2 \right\rangle$$
$$d_{i,j}^2 = \left\langle z_i^2 \right\rangle + \left\langle z_j^2 \right\rangle - 2\left\langle z_i z_j \right\rangle$$
under a zero mean Gaussian with covariance given by $\mathbf{K}$ this is
$$d_{i,j}^2 = k_{i,i} + k_{j,j} - 2k_{i,j}.$$
Take the distance to be square root of this,
$$d_{i,j} = \left(k_{i,i} + k_{j,j} - 2k_{i,j}\right)^{\frac{1}{2}}.$$

Standard Transformation

- This transformation is known as the *standard transformation* between a similarity and a distance (Mardia et al., 1979, pg 402).
- If the covariance is of the form $\mathbf{K} = \mathbf{Y}\mathbf{Y}^\top$ then $k_{i,j} = \mathbf{y}_{i,:}^\top \mathbf{y}_{j,:}$ and

$$d_{i,j} = \left(\mathbf{y}_{i,:}^\top \mathbf{y}_{i,:} + \mathbf{y}_{j,:}^\top \mathbf{y}_{j,:} - 2\mathbf{y}_{i,:}^\top \mathbf{y}_{j,:}\right)^{\frac{1}{2}} = \left|\mathbf{y}_{i,:} - \mathbf{y}_{j,:}\right|_2 .$$

- For other distance matrices this gives us an approach to covert to a similarity matrix or kernel matrix so we can perform classical MDS.