Optimization for Machine Learning

Introduction into supervised learning

Lecturer: Robert M. Gower

Master IASD: AI Systems and Data Science, 2019

Core Info

- **Where**: ENS: 07/11 amphi Langevin, 03/12 U209, 05/12 amphi Langevin.
- **Online:** Teaching materials for these 3 classes: https://gowerrobert.github.io/
- **Google docs with course info:** Can also be found on https://gowerrobert.github.io/

Outline of my three classes

- 07/11/19 Foundations and the empirical risk problem, revision probability, SGD (Stochastic Gradient Descent) for ridge regression
- 03/12/19 SGD for convex optimization. Theory and variants
- 05/12/19 Lab on SGD and variants

Detailed Outline today

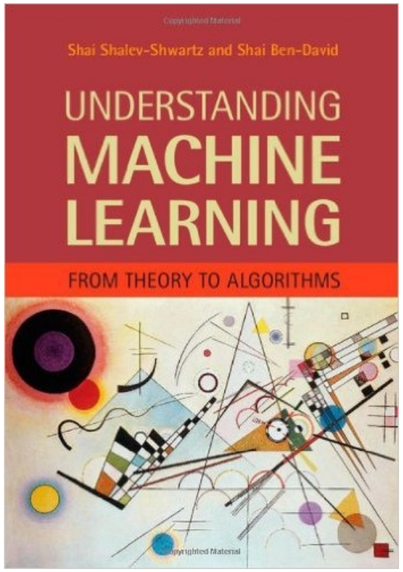
- 13:30 – 14:00: Introduction to empirical risk minimization and classification and SGD
- 14:00 – 15:00 Revision on probability
- 15:00 – 15:30: Tea Time! Break
- 15:30 – 17:00: Exercises and proof of convergence of SGD for ridge regression

An Introduction to Supervised Learning

References classes today

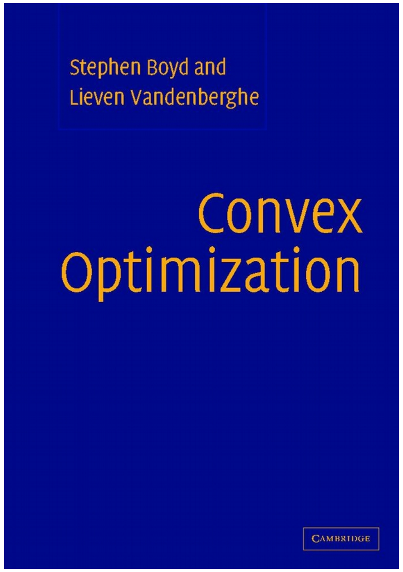
Chapter 2

Understanding Machine Learning: From Theory to Algorithms



Pages 67 to 79

Convex Optimization, Stephen Boyd



Is There a Cat in the Photo?

Yes

No

Is There a Cat in the Photo?

Yes

Is There a Cat in the Photo?

Yes

Is There a Cat in the Photo?

No

Is There a Cat in the Photo?

Yes

Is There a Cat in the Photo?

$\xrightarrow{h}$

Yes

No

x: Input/Feature

y: Output/Target

Find mapping h that assigns the "correct" target to each input

$$h : x \in \mathbf{R}^d \longrightarrow y \in \mathbf{R}$$

Labeled Data: The training set

$x^1 \{$  $y^1 = 1$

$x^2 \{$  $y^2 = 1$

$x^3 \{$  $y^3 = -1$

$\cdots \; x^n \{$  $y^n = 1$

Labeled Data: The training set

x^1 { 
$y^1 = 1$

x^2 { 
$y^2 = 1$

x^3 { 
$y^3 = -1$

$\cdots$ x^n { 
$y^n = 1$

y= *-1* means no/false

Labeled Data: The training set

x^1 { 
$y^1 = 1$

x^2 { 
$y^2 = 1$

x^3 { 
$y^3 = -1$

$\cdots$ x^n { 
$y^n = 1$

y= -1 means no/false

Learning Algorithm

Labeled Data: The training set

$x^1 \{$ 
$y^1 = 1$

$x^2 \{$ 
$y^2 = 1$

$x^3 \{$ 
$y^3 = -1$

$\cdots\ x^n \{$ 
$y^n = 1$

y= -1 means no/false

Learning Algorithm

$$h : x \in X \rightarrow y \in \mathbf{R}$$

Labeled Data: The training set

x^1 {  $y^1 = 1$

x^2 {  $y^2 = 1$

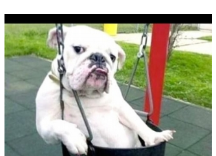
x^3 {  $y^3 = -1$

$\cdots$ x^n {  $y^n = 1$

y= -1 means no/false

Learning Algorithm

$h : x \in X \rightarrow y \in \mathbf{R}$

h () $\longrightarrow$ **-1**

Example: Linear Regression for Height

Male $= 0$
Female $= 1$

Labelled data $x \in \mathbf{R}^2, y \in \mathbf{R}_+$

x_1^1	Sex	0
x_2^1	Age	30
y^1	Height	1,72 cm

$\cdots$

x_1^n	Sex	1
x_2^n	Age	70
y^n	Height	1,52 cm

Example: Linear Regression for Height

Male $= 0$
Female $= 1$

Labelled data $x \in \mathbf{R}^2, y \in \mathbf{R}_+$

x_1^1 {	Sex	0	
x_2^1 {	Age	30	
y^1 {	Height	1,72 cm	

$\cdots$

x_1^n {	Sex	1
x_2^n {	Age	70
y^n {	Height	1,52 cm

Example Hypothesis: Linear Model

$$h_w(x_1, x_2) = w_0 + x_1 w_1 + x_2 w_2 \overset{x_0=1}{=} \langle w, x \rangle$$

Example: Linear Regression for Height

Male $= 0$
Female $= 1$

Labelled data $x \in \mathbf{R}^2, y \in \mathbf{R}_+$

x_1^1	Sex	0
x_2^1	Age	30
y^1	Height	1,72 cm

$\cdots$

x_1^n	Sex	1
x_2^n	Age	70
y^n	Height	1,52 cm

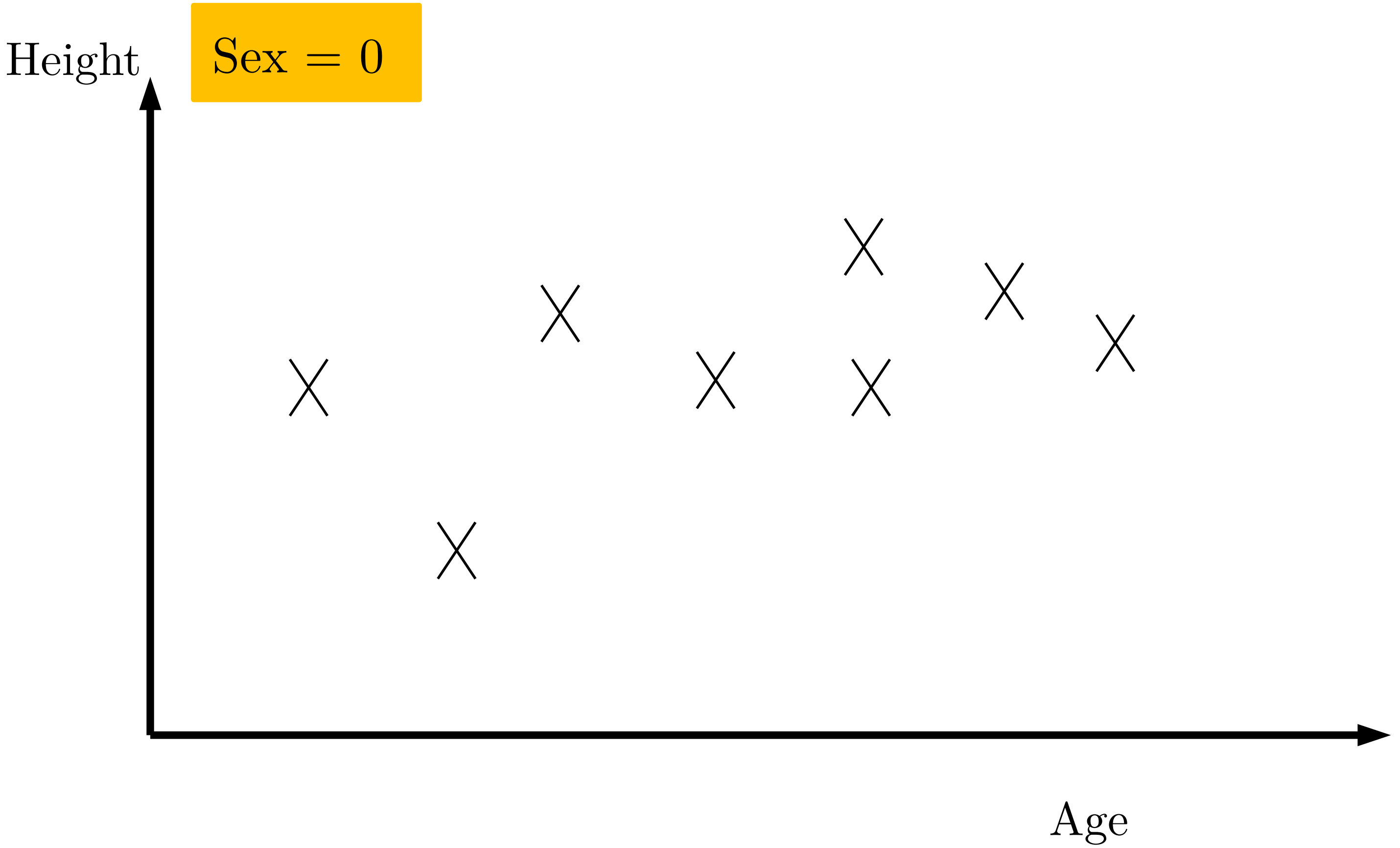
Example Hypothesis: Linear Model

$$h_w(x_1, x_2) = w_0 + x_1 w_1 + x_2 w_2 \overset{x_0=1}{=} \langle w, x \rangle$$

Example Training Problem:

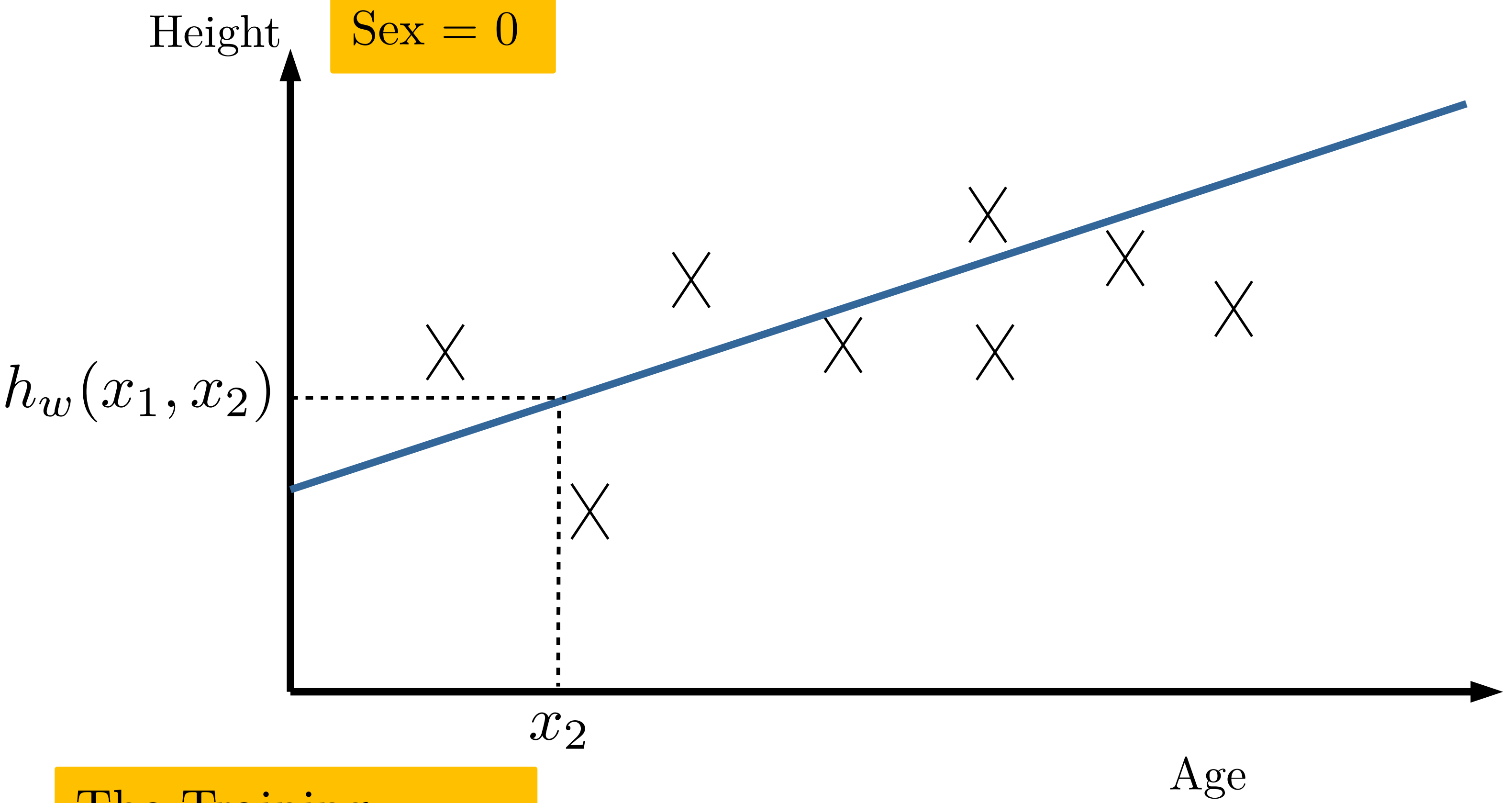
$$\min_{w \in \mathbf{R}^3} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x_1^i, x_2^i) - y^i \right)^2$$

Linear Regression for Height



Linear Regression for Height

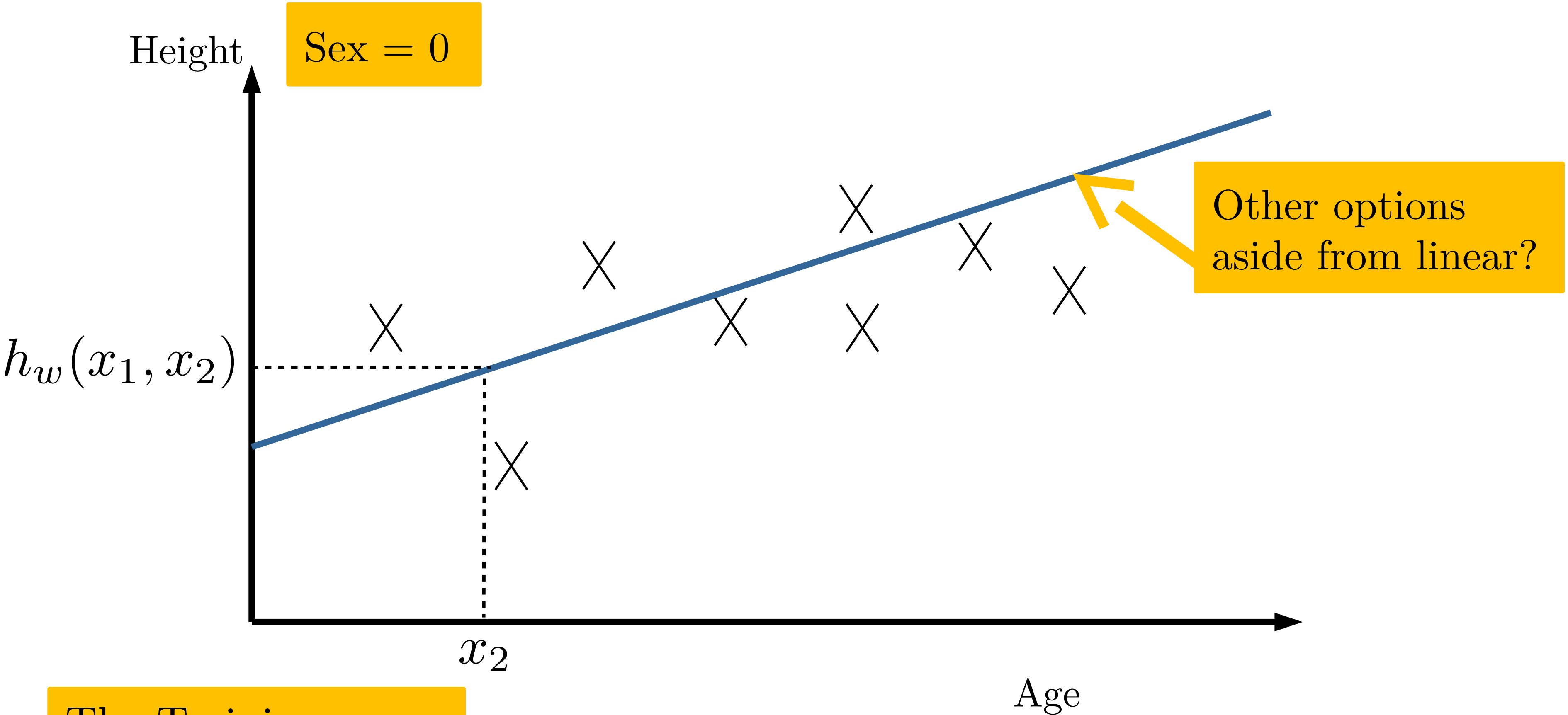
Sex = 0



The Training Algorithm

$$\min_{w \in \mathbf{R}^3} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x_1^i, x_2^i) - y^i \right)^2$$

Linear Regression for Height

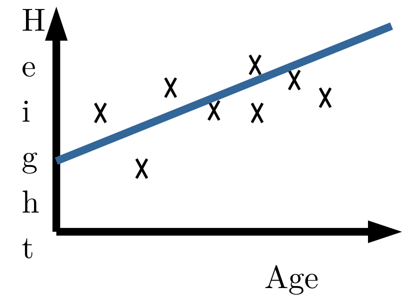


The Training Algorithm

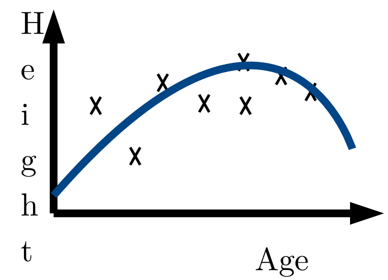
$$\min_{w \in \mathbf{R}^3} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x_1^i, x_2^i) - y^i\right)^2$$

Parametrizing the Hypothesis

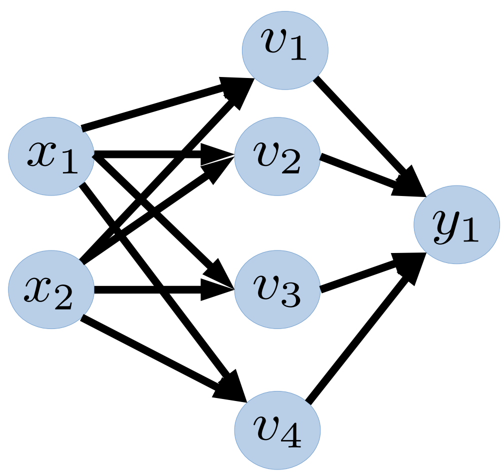
Linear: $$h_w(x) = \sum_{i=0}^{d} w_i x_i$$



Polinomial: $$h_w(x) = \sum_{i,j=0}^{d} w_{ij} x_i x_j$$



Neural Net:



exe :

$$v_1 = \mathrm{sign}(w_{11}x_1 + w_{12}x_2)$$

$$v_4 = 1/\left(1 + \exp(w_{41}x_1 + w_{42}x_2)\right)$$

Loss Functions

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i \right)^2$$

Why a Squared Loss?

Loss Functions

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i\right)^2$$

Why a Squared Loss?

$$\text{Let } y_h := h_w(x)$$

Loss Functions

$$\begin{array}{rccc} \ell: & \mathbf{R} \times \mathbf{R} & \rightarrow & \mathbf{R}_+ \\ & (y_h, y) & \rightarrow & \ell(y_h, y) \end{array}$$

The Training Problem

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right)$$

Loss Functions

$$\min_{w\in\mathbf{R}^d} \frac{1}{n}\sum_{i=1}^{n}\left(h_w(x^i) - y^i\right)^2$$

Why a Squared Loss?

$$\text{Let } y_h := h_w(x)$$

Loss Functions

$$\begin{aligned}\ell : \quad \mathbf{R}\times\mathbf{R} &\rightarrow \quad \mathbf{R}_+\\ (y_h, y) &\rightarrow \quad \ell(y_h, y)\end{aligned}$$

Typically a convex function

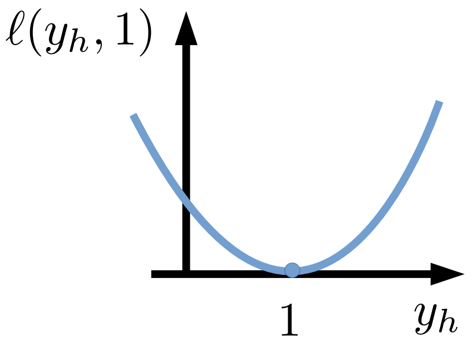
The Training Problem

$$\min_{w\in\mathbf{R}^d} \frac{1}{n}\sum_{i=1}^{n}\ell\left(h_w(x^i), y^i\right)$$

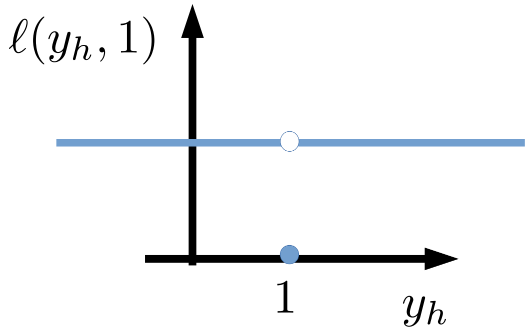
Choosing the Loss Function

Let $y_h := h_w(x)$

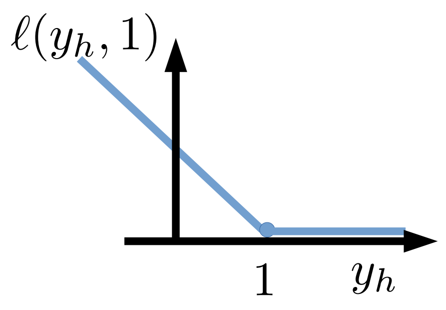
Quadratic Loss $\ell(y_h, y) = (y_h - y)^2$



Binary Loss $\ell(y_h, y) = \begin{cases} 0 & \text{if } y_h = y \\ 1 & \text{if } y_h \neq y \end{cases}$



Hinge Loss $\ell(y_h, y) = \max\{0, 1 - y_h y\}$

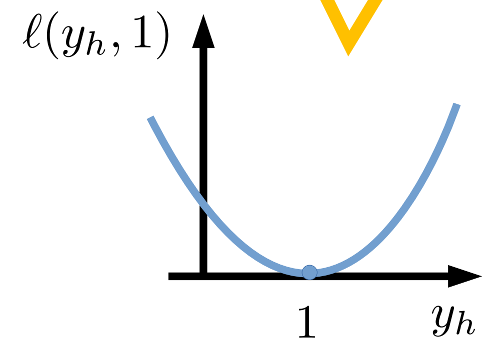


Choosing the Loss Function

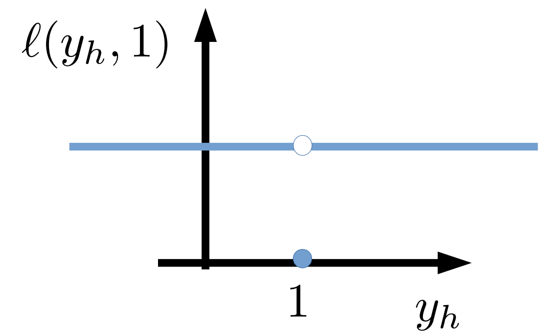
y=1 in all figures

Let $y_h := h_w(x)$

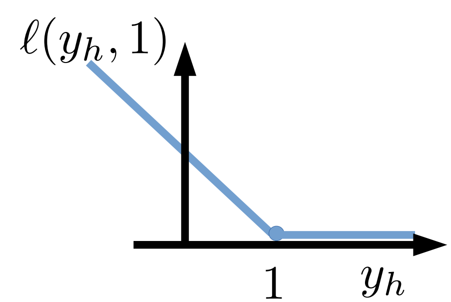
Quadratic Loss $\ell(y_h, y) = (y_h - y)^2$



Binary Loss $\ell(y_h, y) = \begin{cases} 0 & \text{if } y_h = y \\ 1 & \text{if } y_h \neq y \end{cases}$



Hinge Loss $\ell(y_h, y) = \max\{0, 1 - y_h y\}$

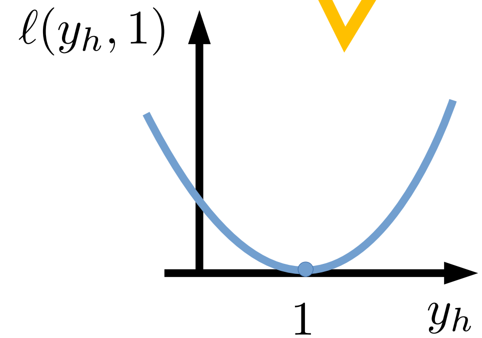


Choosing the Loss Function

y=1 in all figures

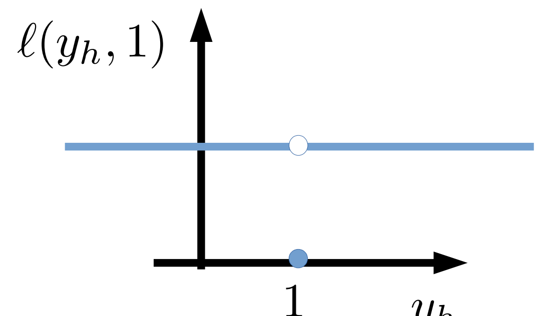
Let $y_h := h_w(x)$

Quadratic Loss $\quad \ell(y_h, y) = (y_h - y)^2$

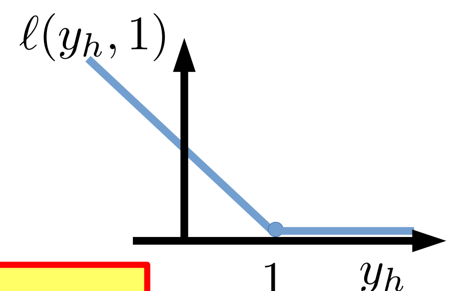


Binary Loss

$$\ell(y_h, y) = \begin{cases} 0 & \text{if } y_h = y \\ 1 & \text{if } y_h \neq y \end{cases}$$



Hinge Loss $\quad \ell(y_h, y) = \max\{0, 1 - y_h y\}$



EXE: Plot the binary and hinge loss function in when $y = -1$

Loss Functions

Is a notion of Loss enough?

What happens when we do not have enough data?

Loss Functions

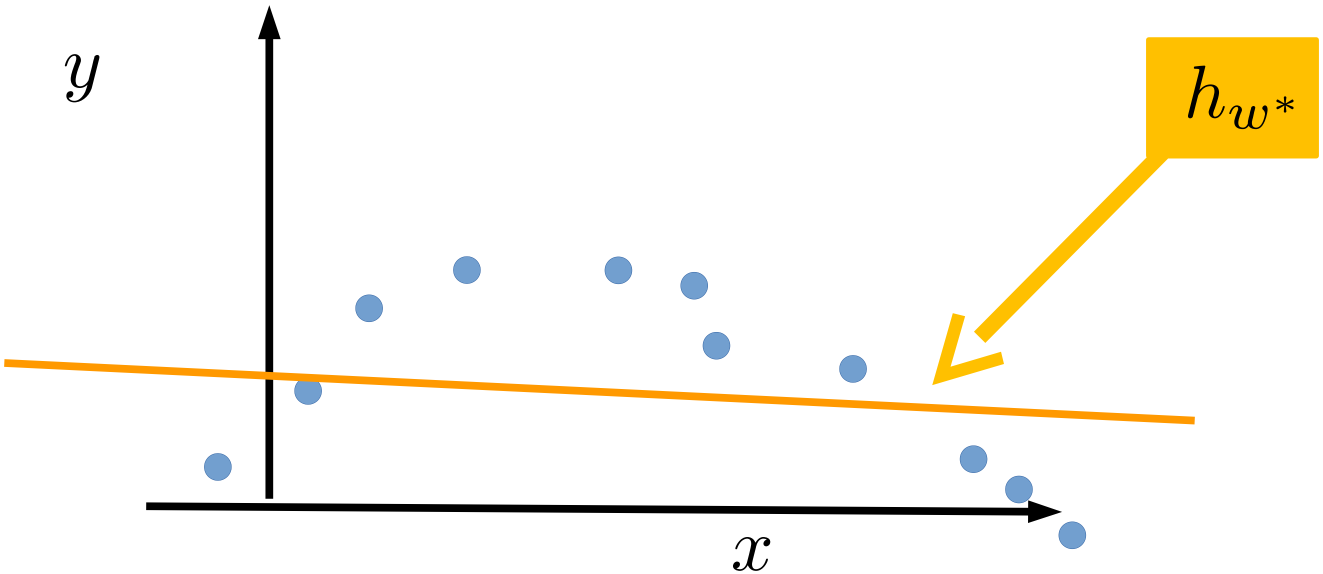
The Training Problem

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right)$$

Is a notion of Loss enough?

What happens when we do not have enough data?

Overfitting and Model Complexity

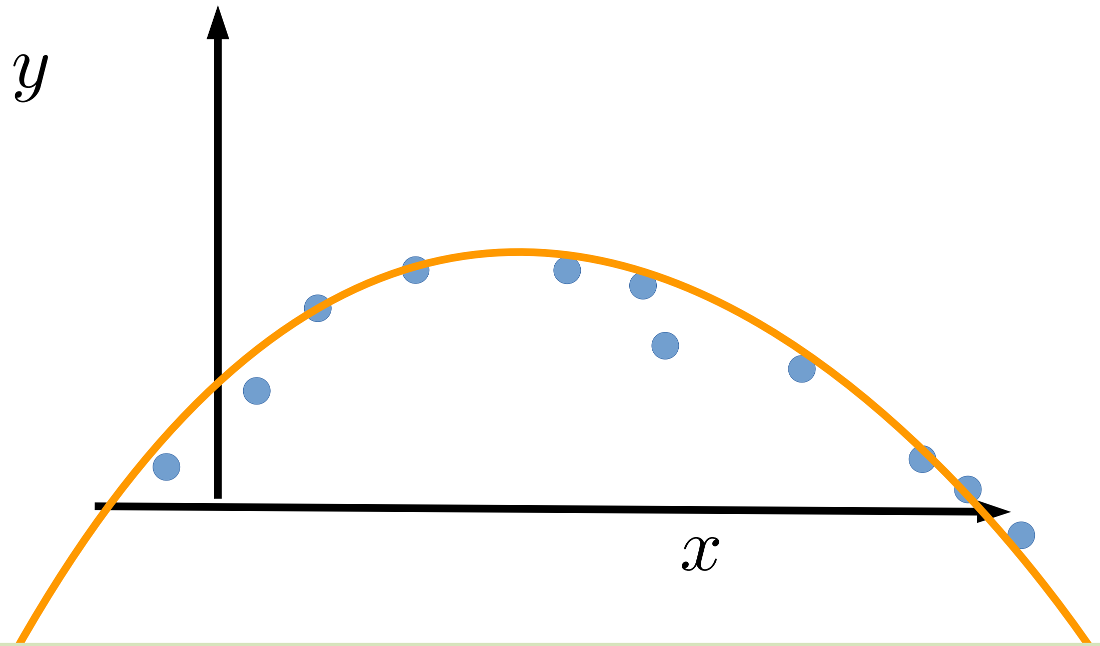


Fitting 1st order polynomial

$$h_w = \langle w, x \rangle$$

$$w^* = \arg \min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i \right)^2$$

Overfitting and Model Complexity

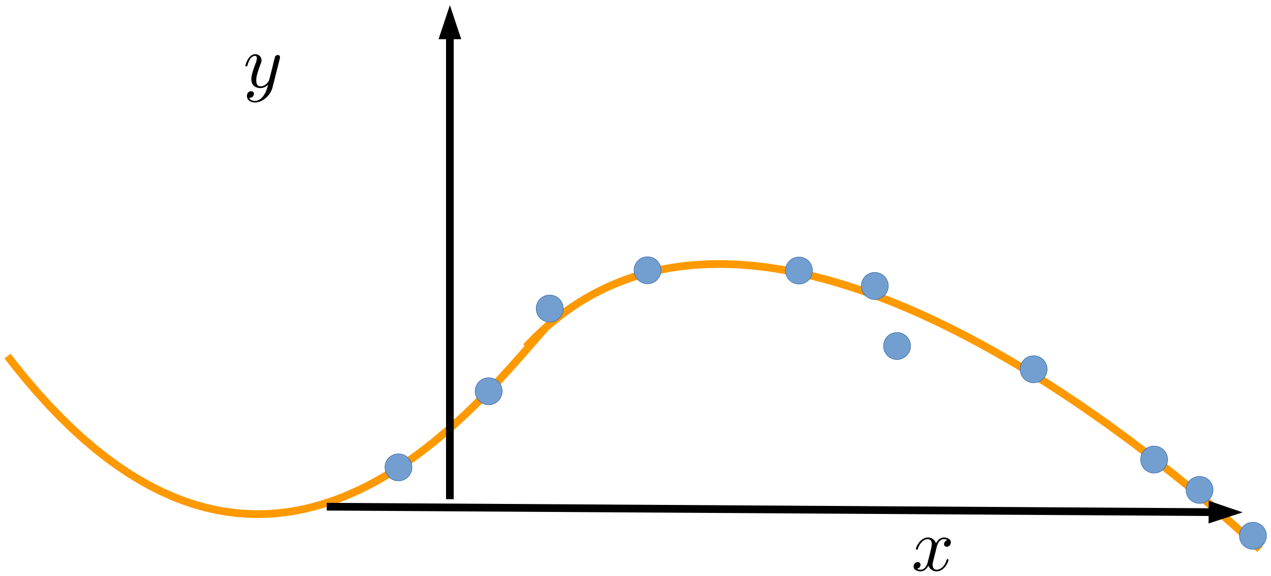


Fitting 2nd order polynomial

$$h_w = w_0 + w_1 x + w_2 x^2$$

$$w^* = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i\right)^2$$

Overfitting and Model Complexity

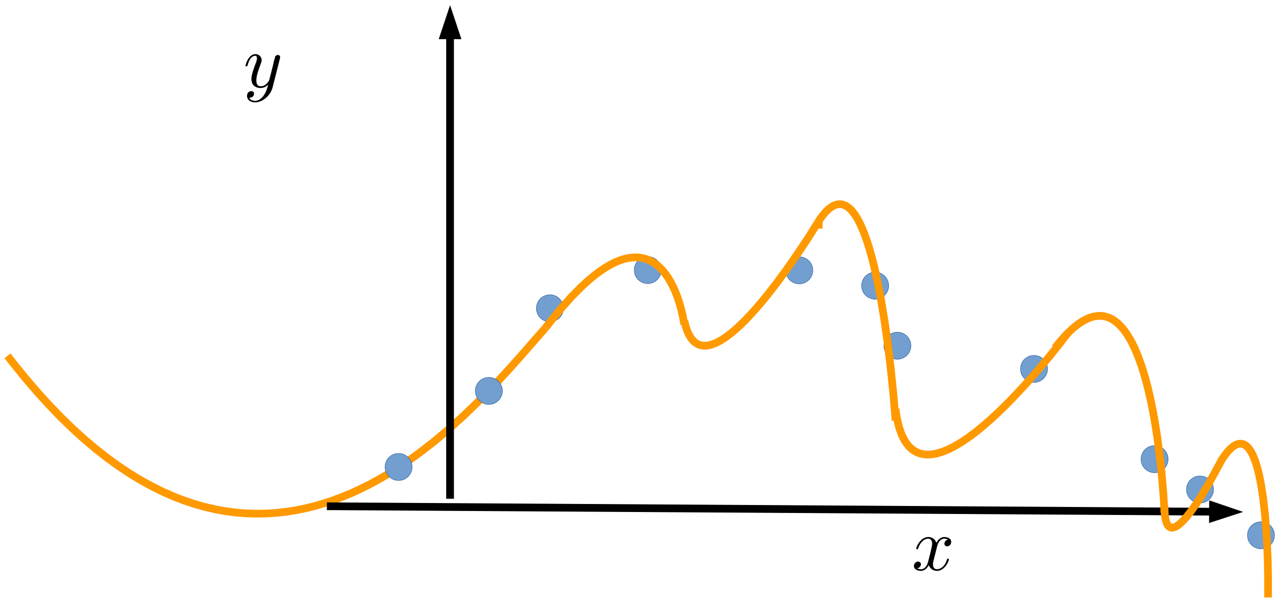


Fitting 3rd order polynomial

$$h_w = \sum_{i=0}^{3} w_i x^i$$

$$w^* = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i\right)^2$$

Overfitting and Model Complexity



Fitting 9th order polynomial

$$h_w = \sum_{i=0}^{9} w_i x^i$$

$$w^* = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i \right)^2$$

Regularization

Regularizor Functions

$$\begin{aligned} R: \quad \mathbf{R}^d &\rightarrow \mathbf{R}_+ \\ w &\rightarrow R(w) \end{aligned}$$

General Training Problem

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w)$$

Regularization

Regularizor Functions

$$R: \quad \mathbf{R}^d \to \mathbf{R}_+ \qquad w \to R(w)$$

General Training Problem

$$\min_{w\in\mathbf{R}^d} \frac{1}{n}\underbrace{\sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right)}_{\text{Goodness of fit, fidelity term ...etc}} + \lambda R(w)$$

Regularization

Regularizor Functions

$$
\begin{aligned}
R: \quad \mathbf{R}^d &\rightarrow \mathbf{R}_+ \\
w &\rightarrow R(w)
\end{aligned}
$$

General Training Problem

$$
\min_{w \in \mathbf{R}^d} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right)}_{\text{Goodness of fit, fidelity term ...etc}} + \underbrace{\lambda R(w)}_{\text{Penalizes complexity}}
$$

Regularization

Regularizor Functions

$$\begin{aligned} R : \quad \mathbf{R}^d &\rightarrow \mathbf{R}_+ \\ w &\rightarrow R(w) \end{aligned}$$

Controls tradeoff between fit and complexity

General Training Problem

$$\min_{w \in \mathbf{R}^d} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell \left(h_w(x^i), y^i \right)}_{\text{Goodness of fit, fidelity term ...etc}} + \underbrace{\lambda R(w)}_{\text{Penalizes complexity}}$$

Goodness of fit, fidelity term ...etc

Penalizes complexity

Regularization

Regularizor Functions

$$\begin{aligned} R: \quad \mathbf{R}^d &\rightarrow \mathbf{R}_+ \\ w &\rightarrow R(w) \end{aligned}$$

Controls tradeoff between fit and complexity

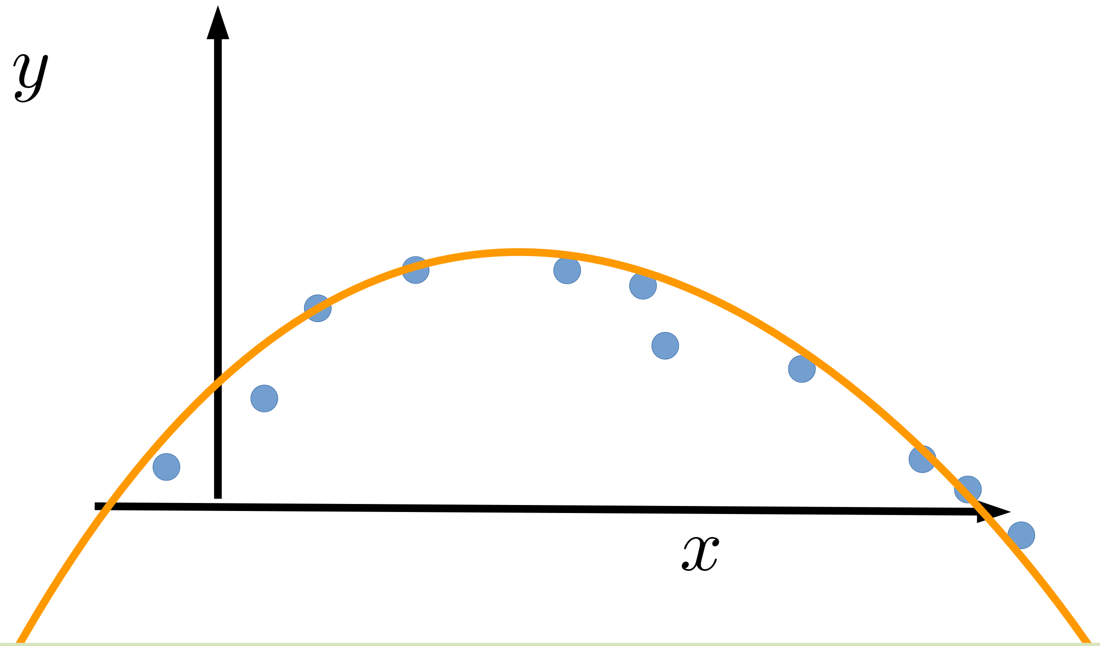
General Training Problem

$$\min_{w \in \mathbf{R}^d} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right)}_{\text{Goodness of fit, fidelity term ...etc}} + \underbrace{\lambda R(w)}_{\text{Penalizes complexity}}$$

Exe:

$$R(w) = ||w||_2^2, \quad ||w||_1, \quad ||w||_p, \quad \text{other norms} \ \ldots$$

Overfitting and Model Complexity

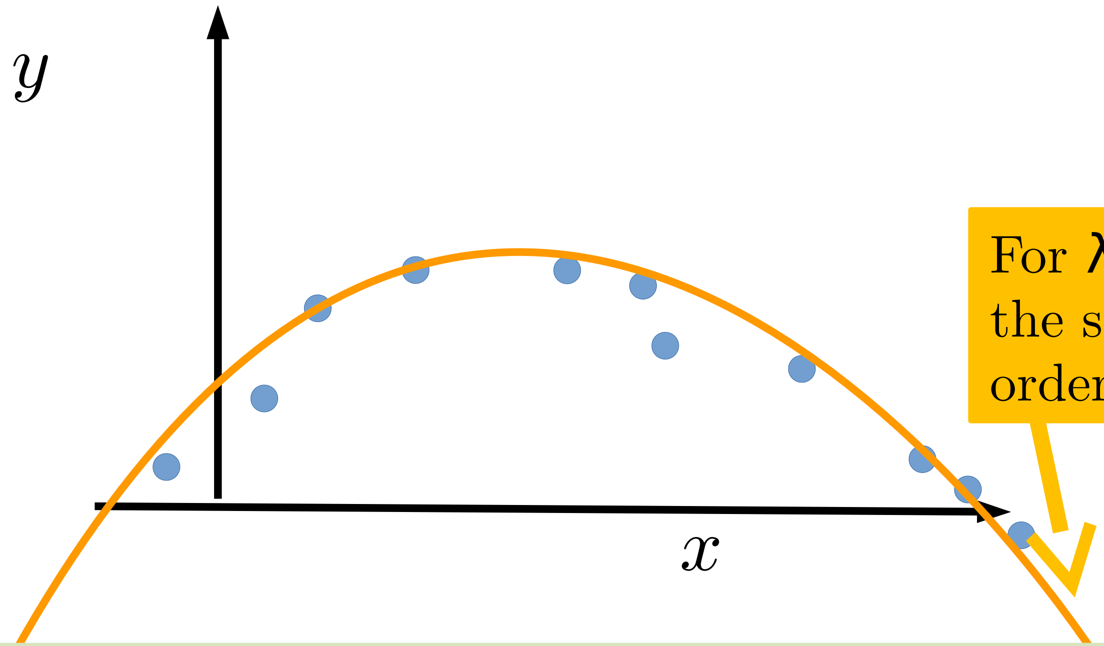


Fitting k^th order polynomial

$$h_w = \sum_{i=0}^{k} w_i x^i$$

$$w^* = \arg \min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i\right)^2 + \lambda ||w||_1$$

Overfitting and Model Complexity



For λ big enough, the solution is a 2nd order polynomial

Fitting k^th^ order polynomial

$$h_w = \sum_{i=0}^{k} w_i x^i$$

$$w^* = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(h_w(x^i) - y^i\right)^2 + \lambda ||w||_1$$

Exe: Ridge Regression

Linear hypothesis

$$h_w(x) = \langle w, x \rangle$$

L2 regularizor

$$R(w) = ||w||_2^2$$

L2 loss

$$\ell(y_h, y) = (y_h - y)^2$$

Ridge Regression

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y^i - \langle w, x^i \rangle)^2 + \lambda ||w||_2^2$$

Exe: Support Vector Machines

Linear hypothesis

$$h_w(x) = \langle w, x \rangle$$

L2 regularizor

$$R(w) = ||w||_2^2$$

Hinge loss

$$\ell(y_h, y) = \max\{0, 1 - y_h y\}$$

SVM with soft margin

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y^i \langle w, x^i \rangle\} + \lambda ||w||_2^2$$

Exe: Logistic Regression

Linear hypothesis

$$h_w(x) = \langle w, x \rangle$$

L2 regularizor

$$R(w) = ||w||_2^2$$

Logistic loss

$$\ell(y_h, y) = \ln(1 + e^{-y y_h})$$

Logistic Regression

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-y^i \langle w, x^i \rangle}) + \lambda ||w||_2^2$$

The Machine Learners Job

(1) Get the labeled data: $(x^1, y^1), \ldots, (x^n, y^n)$

The Machine Learners Job

(1) Get the labeled data: $(x^1, y^1), \ldots, (x^n, y^n)$

(2) Choose a parametrization for hypothesis: $h_w(x)$

The Machine Learners Job

(1) Get the labeled data: $(x^1, y^1), \ldots, (x^n, y^n)$

(2) Choose a parametrization for hypothesis: $h_w(x)$

(3) Choose a loss function: $\ell(h_w(x), y) \geq 0$

The Machine Learners Job

(1) Get the labeled data: $(x^1, y^1), \ldots, (x^n, y^n)$

(2) Choose a parametrization for hypothesis: $h_w(x)$

(3) Choose a loss function: $\ell(h_w(x), y) \geq 0$

(4) Solve the *training problem*:

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w)$$

The Machine Learners Job

(1) Get the labeled data: $(x^1, y^1), \ldots, (x^n, y^n)$

(2) Choose a parametrization for hypothesis: $h_w(x)$

(3) Choose a loss function: $\ell(h_w(x), y) \geq 0$

(4) Solve the *training problem*:

$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w)$$

(5) Test and cross-validate. If fail, go back a few steps

The Machine Learners Job

(1) Get the labeled data: $(x^1, y^1), \dots, (x^n, y^n)$

(2) Choose a parametrization for hypothesis: $h_w(x)$

(3) Choose a loss function: $\ell(h_w(x), y) \geq 0$

(4) Solve the *training problem*:

$$\boxed{\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w)}$$

(5) Test and cross-validate. If fail, go back a few steps

Re-writing as Sum of Terms

A Datum Function

$$f_i(w) := \ell\left(h_w(x^i), y^i\right) + \lambda R(w)$$

$$\frac{1}{n}\sum_{i=1}^{n} \ell\left(h_w(x^i), y^i\right) + \lambda R(w) = \frac{1}{n}\sum_{i=1}^{n} \left(\ell\left(h_w(x^i), y^i\right) + \lambda R(w)\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

Finite Sum Training Problem

$$\min_{w \in \mathbf{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(w) =: f(w)$$

Can we use this sum structure?

The Training Problem

Solving the *training problem*:

$$\min_{w\in\mathbf{R}^d} \frac{1}{n}\sum_{i=1}^n f_i(w)$$

Reference method: Gradient descent

$$\nabla\left(\frac{1}{n}\sum_{i=1}^n f_i(w)\right) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(w)$$

Gradient Descent Algorithm

Set $w^0 = 0$, choose $\alpha > 0$.

for $t = 0, 1, 2, \ldots, T-1$

$$w^{t+1} = w^t - \tfrac{\alpha}{n}\textstyle\sum_{i=1}^n \nabla f_i(w^t)$$

Output w^T

The Training Problem

Solving the *training problem*: $$\min_{w\in\mathbf{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

Problem with Gradient Descent:
Each iteration requires computing a gradient $\nabla f_i(w)$ for each data point. One gradient for each cat on the internet!

Gradient Descent Algorithm

Set $w^0 = 0$, choose $\alpha > 0$.
for $t = 0, 1, 2, \ldots, T$
$$w^{t+1} = w^t - \frac{\alpha}{n}\sum_{i=1}^{n} \nabla f_i(w^t)$$
Output w^T

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \ldots, n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = \nabla f(w)$$

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \ldots, n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(w) = \nabla f(w)$$

Use $\nabla f_j(w) \approx \nabla f(w)$

Stochastic Gradient Descent

Is it possible to design a method that uses only the gradient of a **single** data function $f_i(w)$ at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \ldots, n\}$ selected uniformly at random. Then

$$\mathbb{E}_j[\nabla f_j(w)] = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w) = \nabla f(w)$$

Use $\nabla f_j(w) \approx \nabla f(w)$

EXE: Let $\sum_{i=1}^{n} p_i = 1$ and $j \sim p_j$. Show $\mathbb{E}[\nabla f_j(w)/(np_j)] = \nabla f(w)$

Stochastic Gradient Descent

SGD 0.0 Constant stepsize

Set $w^0 = 0$, choose $\alpha > 0$

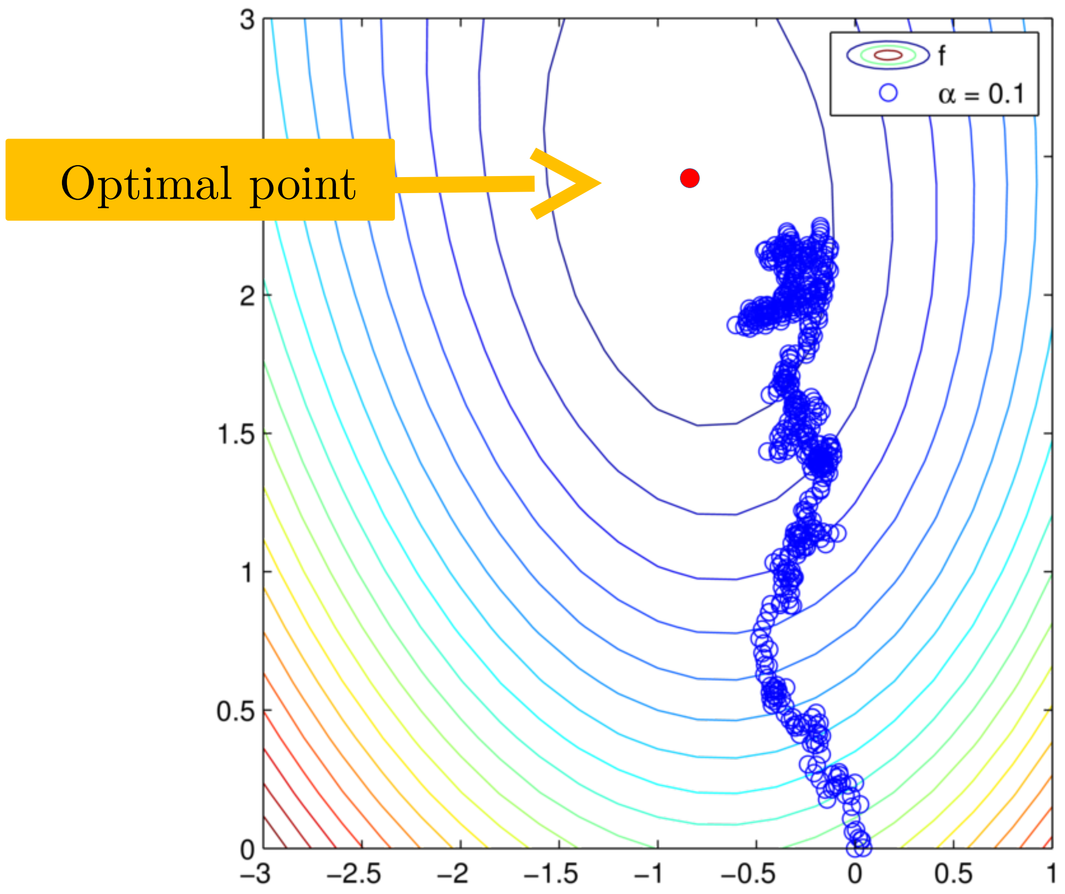
for $t = 0, 1, 2, \ldots, T-1$

  sample $j \in \{1, \ldots, n\}$

  $w^{t+1} = w^t - \alpha \nabla f_j(w^t)$

Output w^T

Stochastic Gradient Descent



Detailed Outline today

- 13:30 – 14:00: Introduction to empirical risk minimization and classification and SGD
- **14:00 – 15:00 Revision on probability**
- **15:00 – 15:30: Tea Time! Break**
- **15:30 – 17:00: Exercises and proof of convergence of SGD for ridge regression**

Assumptions for Convergence

Strong Convexity

$$f(y) \geq f(w) + \langle \nabla f(w), y - w \rangle + \frac{\lambda}{2} ||y - w||_2^2, \quad \forall w, y$$

$$\Big\downarrow \; y = w^*$$

$$2\langle \nabla f(w), w - w^* \rangle \geq \lambda ||w - w^*||_2^2$$

Assumptions for Convergence

Strong Convexity

$$f(y) \geq f(w) + \langle \nabla f(w), y - w \rangle + \frac{\lambda}{2}||y - w||_2^2, \quad \forall w, y$$

$$\Big\Downarrow \; y = w^*$$

$$2\langle \nabla f(w), w - w^* \rangle \geq \lambda ||w - w^*||_2^2$$

Assumptions for Convergence

Strong Convexity

$$f(y) \geq f(w) + \langle \nabla f(w), y - w \rangle + \frac{\lambda}{2} ||y - w||_2^2, \quad \forall w, y$$

$$y = w^*$$

$$2\langle \nabla f(w), w - w^* \rangle \geq \lambda ||w - w^*||_2^2$$

Expected Bounded Stochastic Gradients

$$\mathbb{E}_j [||\nabla f_j(w^t)||_2^2] \leq B^2, \text{ for all iterates } w^t \text{ of SGD}$$

Assumptions for Convergence

Strong Convexity

$$f(y) \geq f(w) + \langle \nabla f(w), y - w \rangle + \frac{\lambda}{2} ||y - w||_2^2, \quad \forall w, y$$

$$\Big\Downarrow \; y = w^*$$

$$2\langle \nabla f(w), w - w^* \rangle \geq \lambda ||w - w^*||_2^2$$

Expected Bounded Stochastic Gradients

$$\mathbb{E}_j[||\nabla f_j(w^t)||_2^2] \leq B^2, \text{ for all iterates } w^t \text{ of SGD}$$

Complexity / Convergence

Theorem

If $0 < \alpha \leq \frac{1}{\lambda}$ then the iterates of the SGD 0.0 method satisfy

$$\mathbb{E}\left[||w^t - w^*||_2^2\right] \leq (1-\alpha\lambda)^t ||w^0 - w^*||_2^2 + \frac{\alpha}{\lambda} B^2$$

EXE: Do exercises on convergence of random sequences.

Complexity / Convergence

Theorem

If $0 < \alpha \leq \frac{1}{\lambda}$ then the iterates of the SGD 0.0 method satisfy

$$\mathbb{E}\left[||w^t - w^*||_2^2\right] \leq (1-\alpha\lambda)^t ||w^0 - w^*||_2^2 + \frac{\alpha}{\lambda} B^2$$

Shows that $\alpha \approx \frac{1}{\lambda}$

EXE: Do exercises on convergence of random sequences.

Complexity / Convergence

Theorem

If $0 < \alpha \leq \frac{1}{\lambda}$ then the iterates of the SGD 0.0 method satisfy

$$\mathbb{E}\left[||w^t - w^*||_2^2\right] \leq (1 - \alpha\lambda)^t ||w^0 - w^*||_2^2 + \frac{\alpha}{\lambda} B^2$$

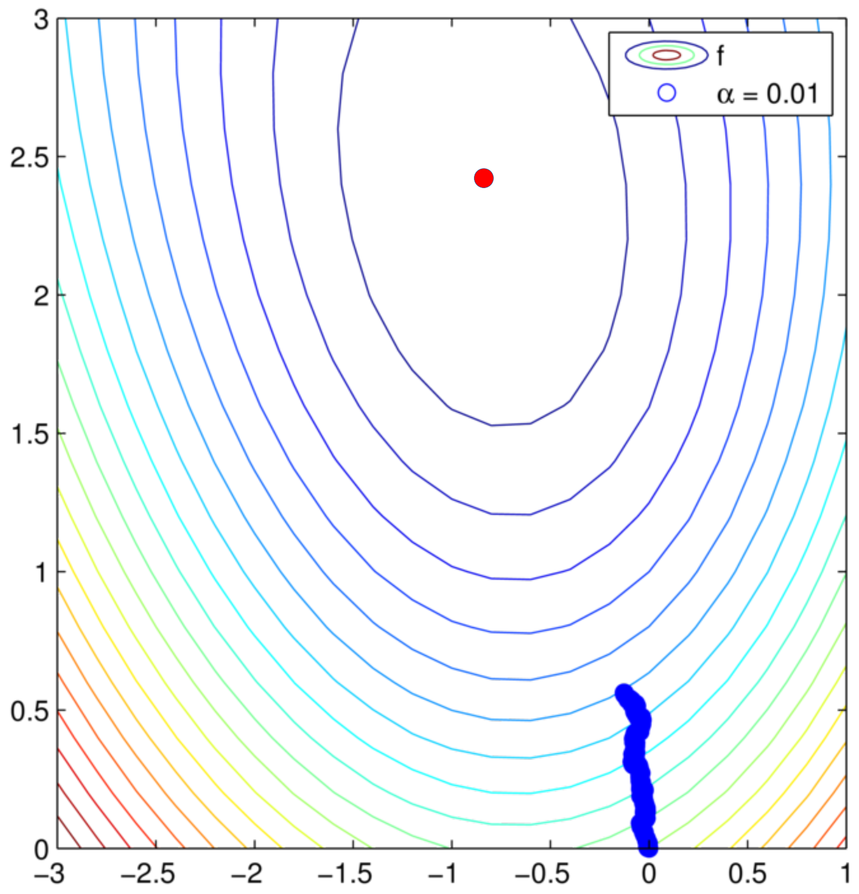
Shows that $\alpha \approx \frac{1}{\lambda}$

Shows that $\alpha \approx 0$

EXE: Do exercises on convergence of random sequences.

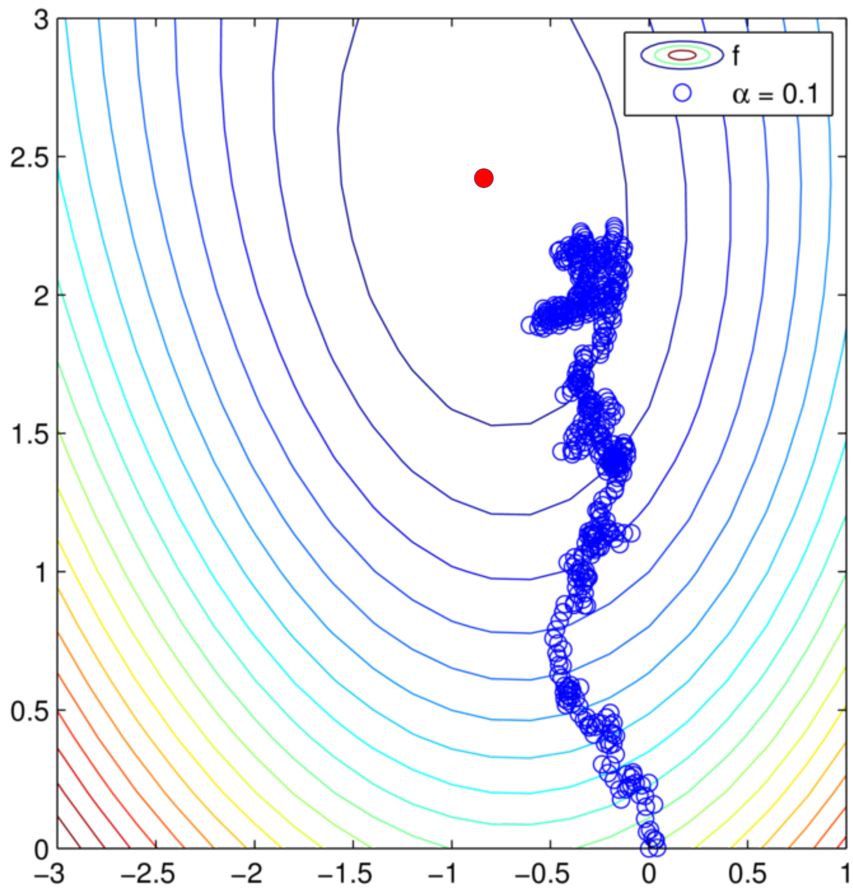
Stochastic Gradient Descent

α =0.01



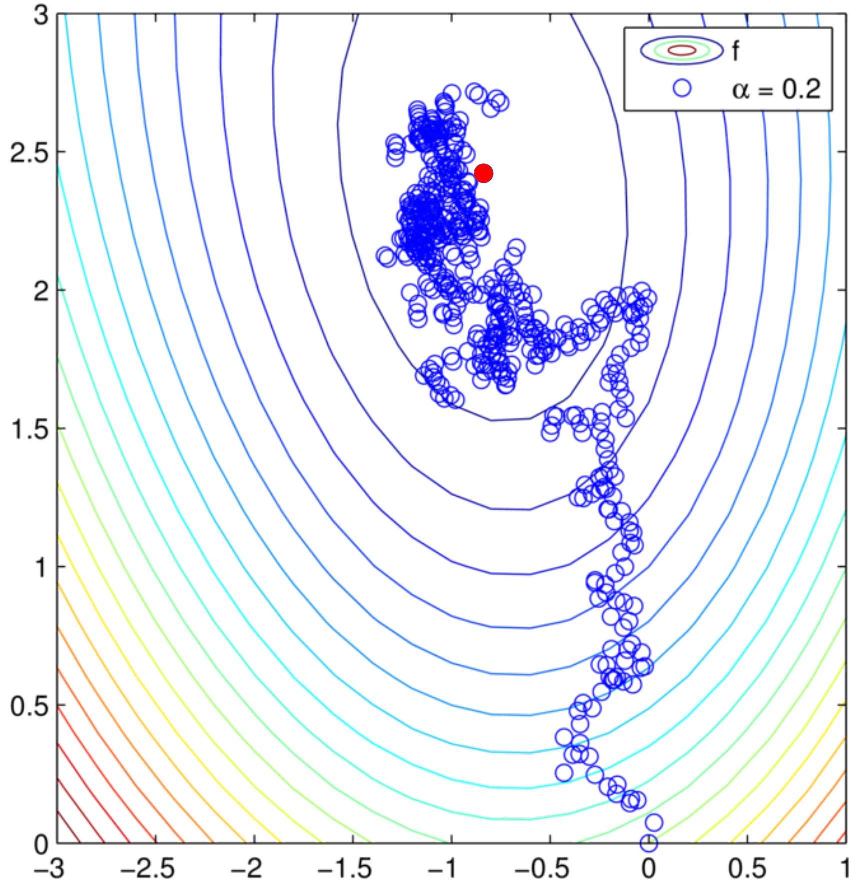
Stochastic Gradient Descent

$\alpha = 0.1$



Stochastic Gradient Descent

$\alpha = 0.2$



Stochastic Gradient Descent

α =0.5