Optimization for Machine Learning

Introduction into supervised learning, stochastic gradient descent analysis and tricks

Lecturer: Robert M. Gower

28th of April to 5th of May 2020, Cornell mini-lecture series, online

Outline of my three classes

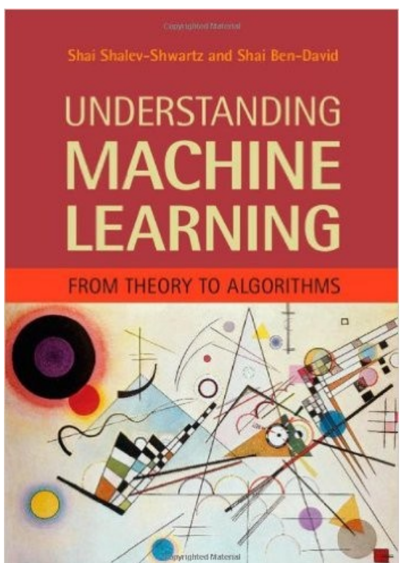
- 04/27/20 Intro to empirical risk problem and stochastic gradient descent (SGD)
- 04/29/20 SGD for convex optimization. Theory and variants
- 05/05/20 SGD with momentum and tricks

Part I: An Introduction to Supervised Learning

References classes today

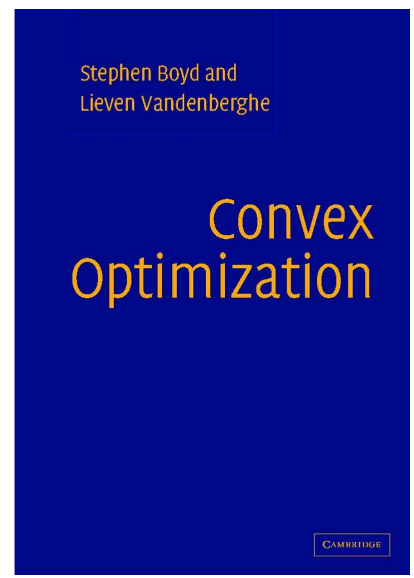
Chapter 2

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?

$$\xrightarrow{h} \left\{ \begin{array}{l} \textbf{Yes} \\ \\ \textbf{No} \end{array} \right.$$

x: Input/Feature

y: Output/Target

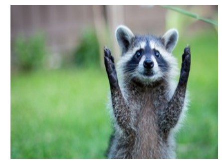
Find mapping *h* that assigns the "correct" target to each input

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

Labeled Data: The training set

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

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

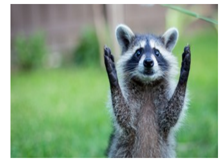
$x^3 \{$ 
$y^3 = -1$

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

Labeled Data: The training set

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

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

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

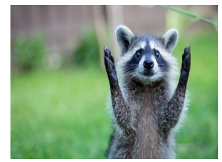
$\cdots \; x^n \{$  $y^n = 1$

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 { 
$y^n = 1$

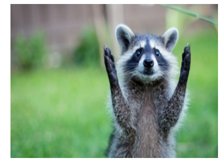
y= *-1* means no/false

Training 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

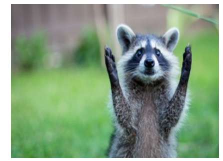
Training Algorithm

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

Labeled Data: The training set

x^1 { 
$y^1 = 1$

x^2 { 
$y^2 = 1$

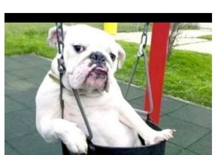
x^3 { 
$y^3 = -1$

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

y= -1 means no/false

Training Algorithm

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

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

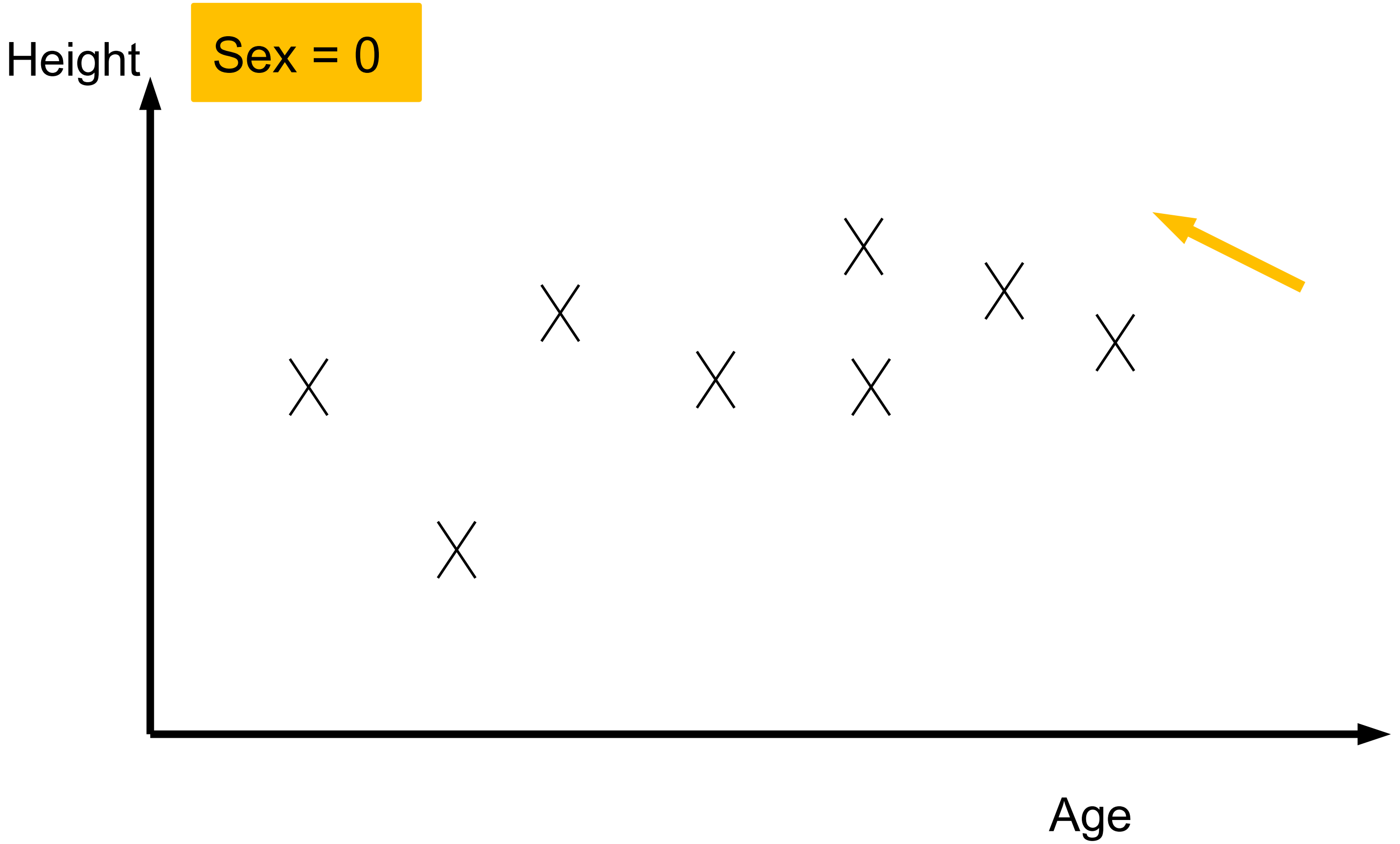
Example Hypothesis: Linear Model

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

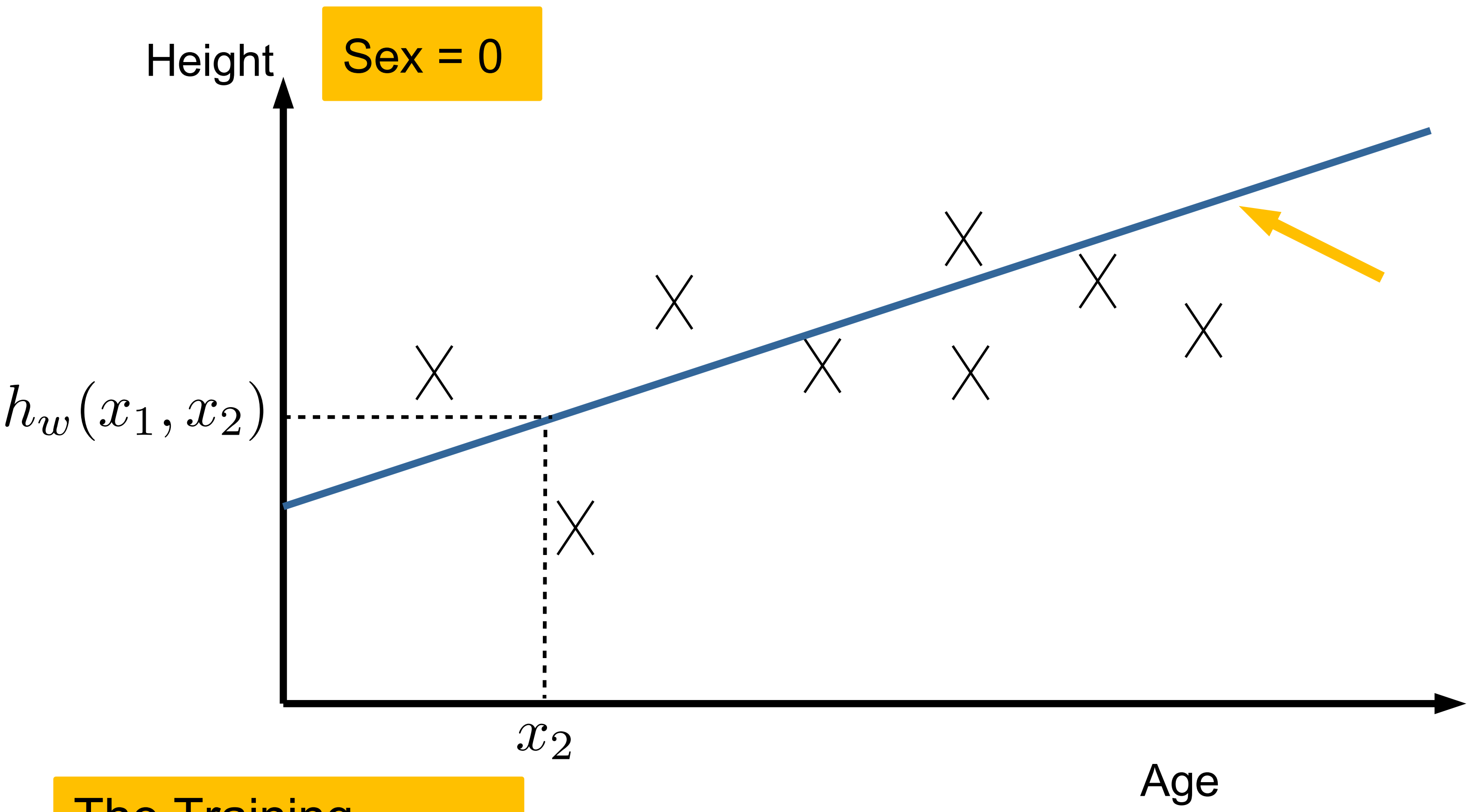
Example Training Problem:

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

Linear Regression for Height



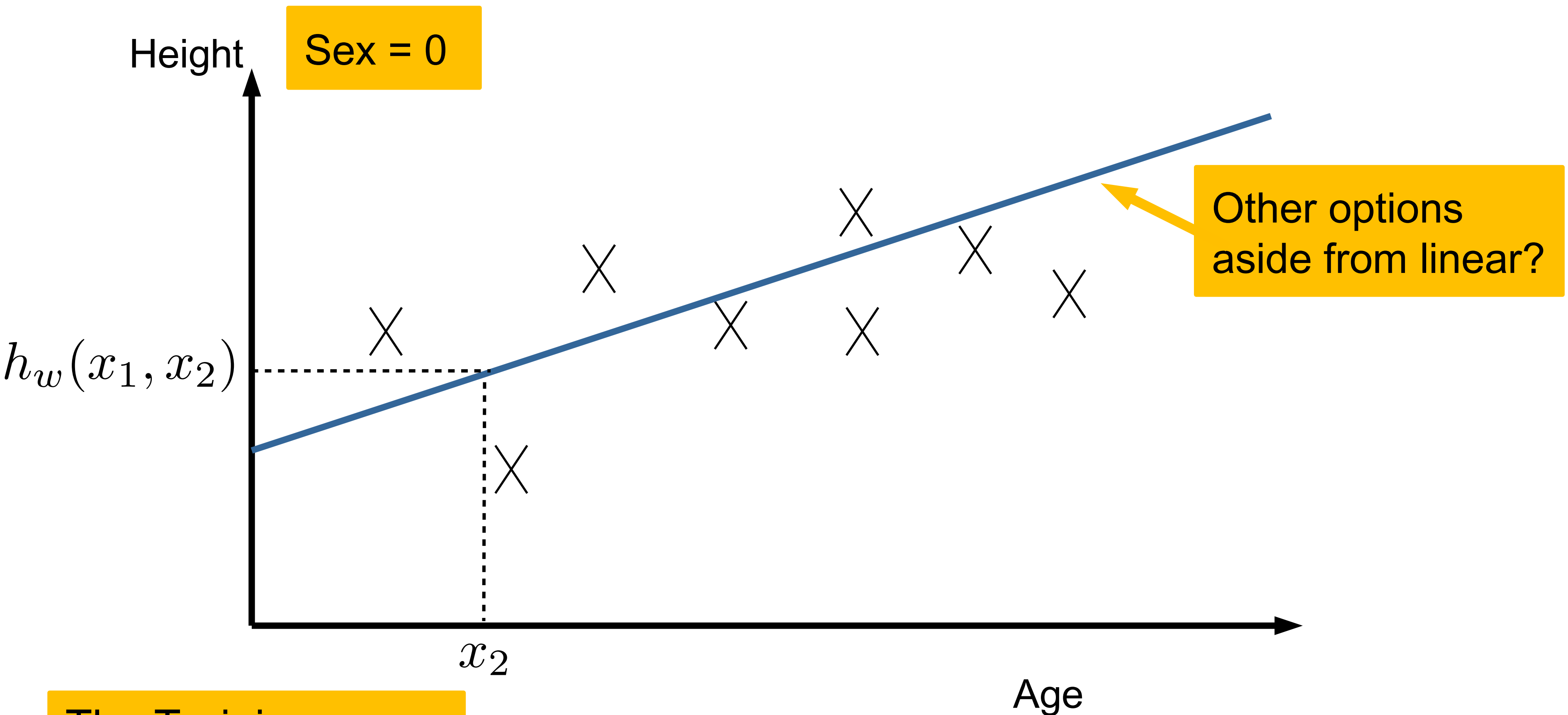
Linear Regression for Height



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



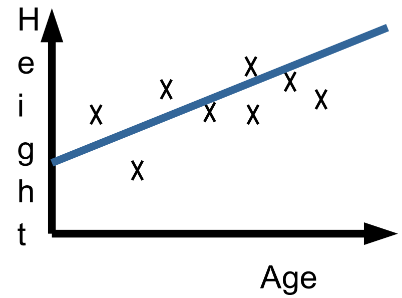
The Training Algorithm

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

Parametrizing the Hypothesis

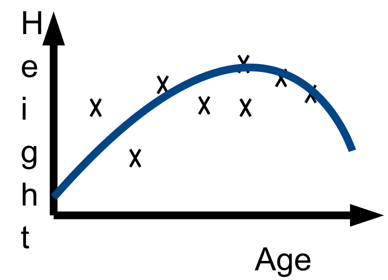
Linear:

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

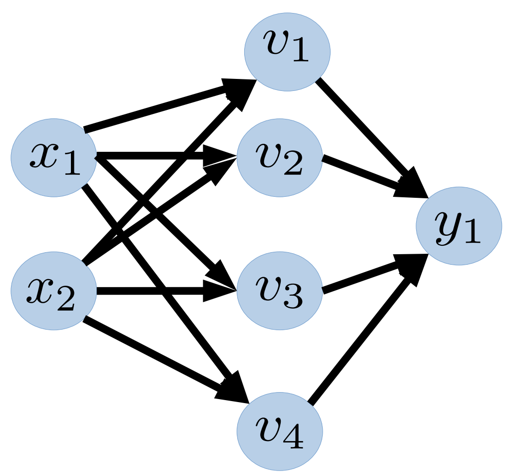


Polinomial:

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



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

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

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

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

Typically a convex function

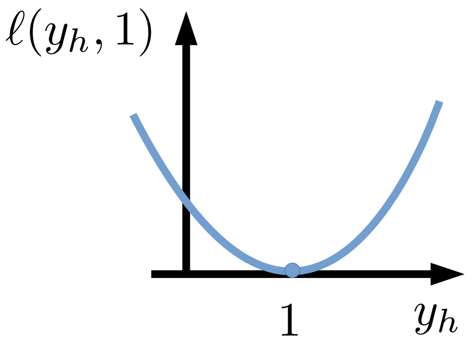
The Training Problem

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

Choosing the Loss Function

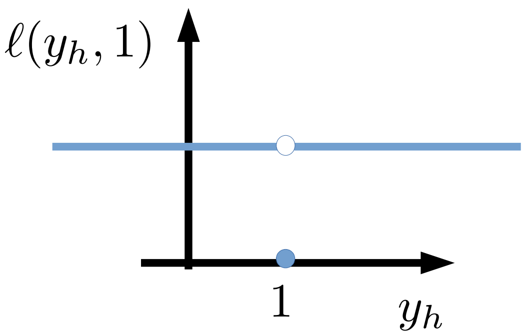
Let $y_h := h_w(x)$

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



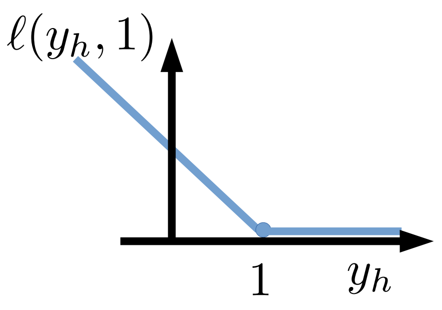
Binary Loss

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



Hinge Loss

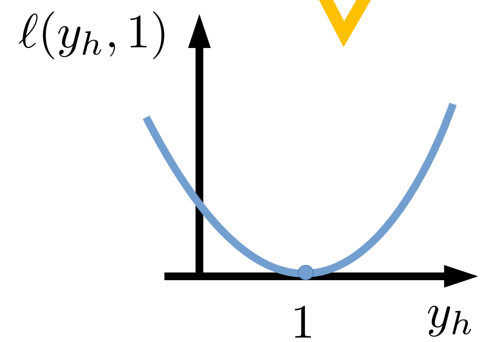
$$\ell(y_h, y) = \max\{0, 1 - y_h y\}$$



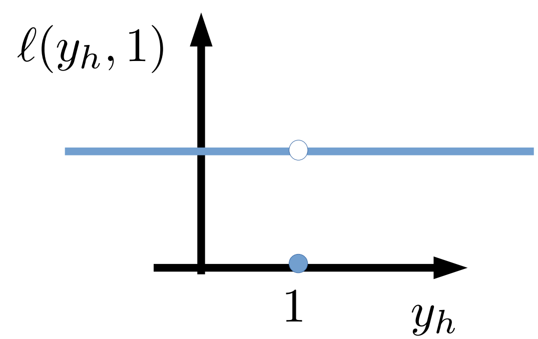
Choosing the Loss Function

y=1 in all figures

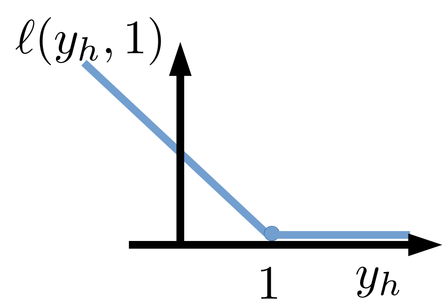
Let $y_h := h_w(x)$



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



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



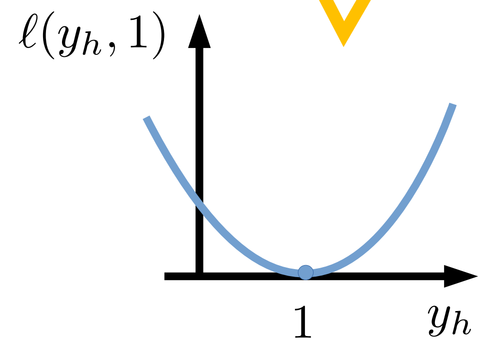
Hinge Loss $\ell(y_h, y) = \max\{0, 1 - y_h y\}$

Choosing the Loss Function

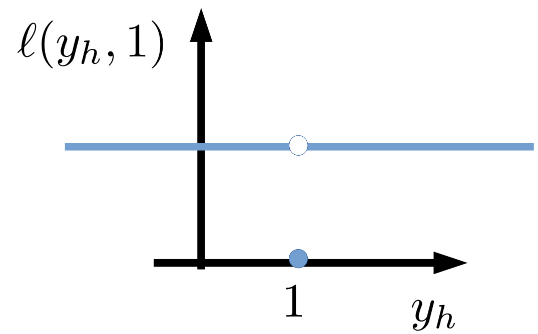
y=1 in all figures

Let $y_h := h_w(x)$

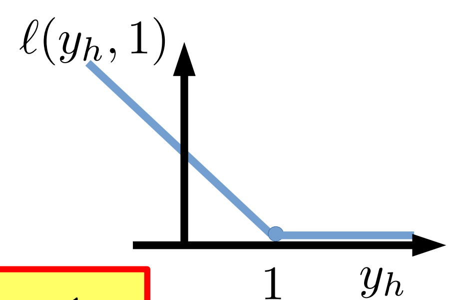
Quadratic Loss $\ell(y_h, y) = (y_h - y)^2$



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



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

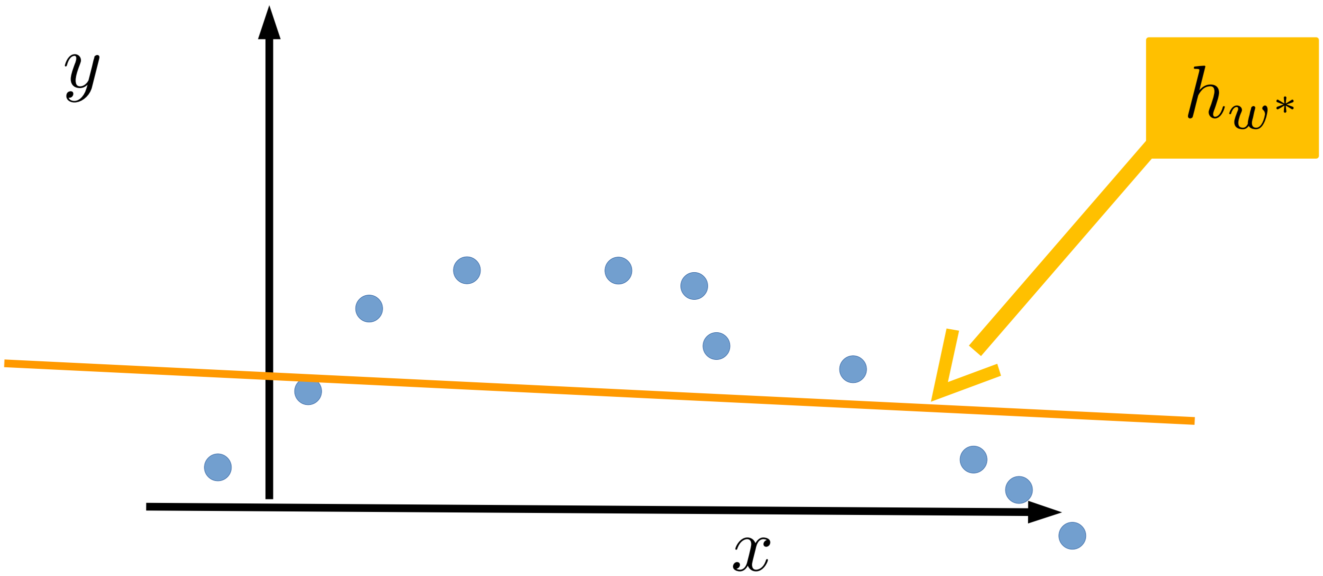
The Training Problem

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

Is a notion of Loss enough?

What happens when we do not have enough data?

Overfitting and Model Complexity

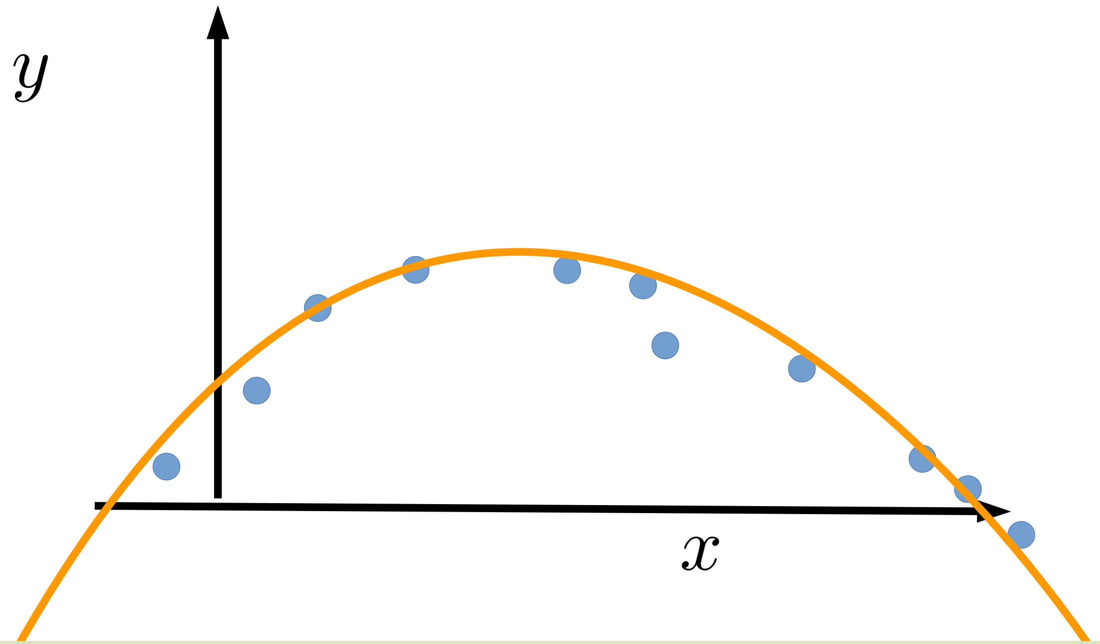


Fitting 1st order polynomial

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

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

Overfitting and Model Complexity

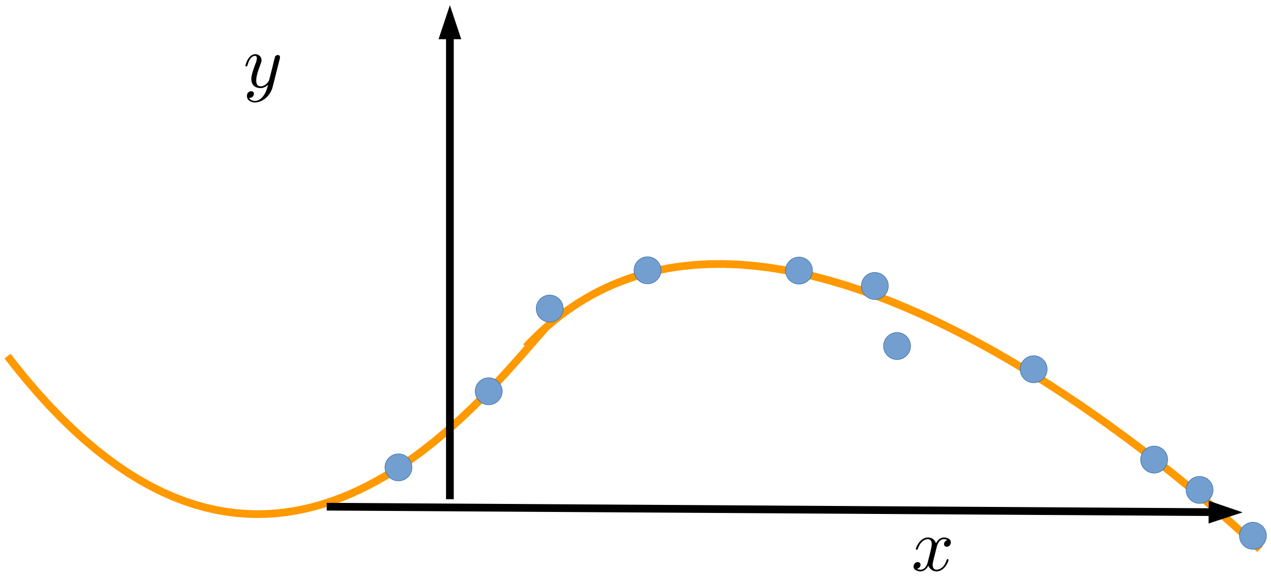


Fitting 2nd order polynomial

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

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

Overfitting and Model Complexity

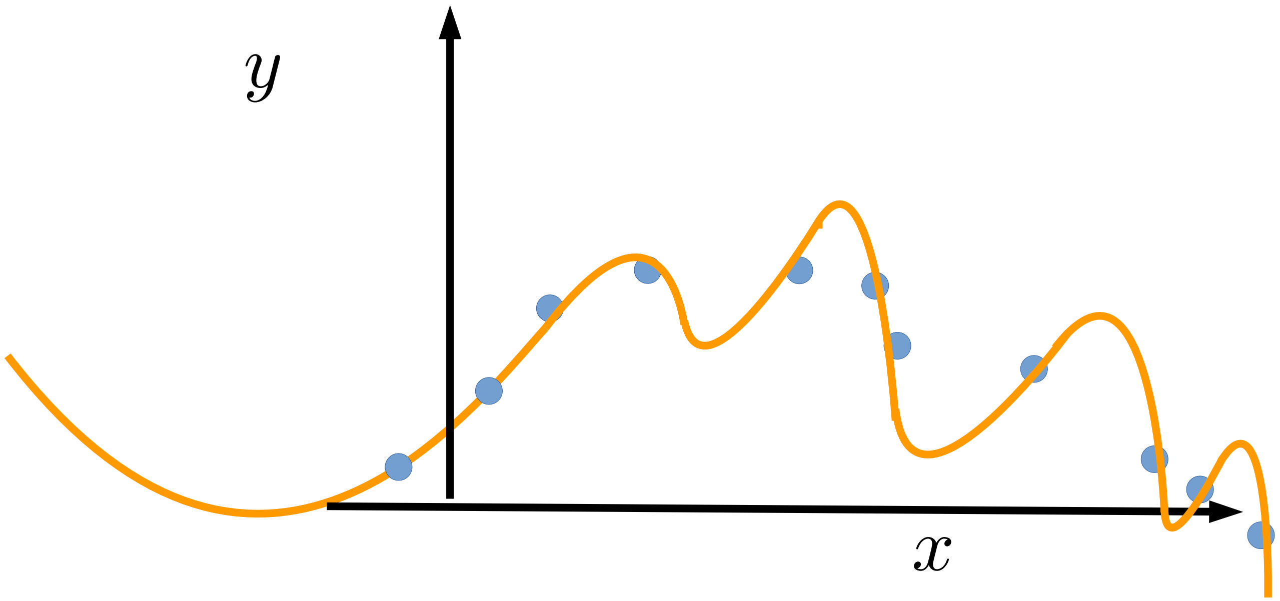


Fitting 3rd order polynomial

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

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

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

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

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

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

General Training Problem

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

Regularization

Regularizor Functions

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

General Training Problem

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

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

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

Regularization

Regularizor Functions

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

Controls tradeoff between fit and complexity

General Training Problem

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

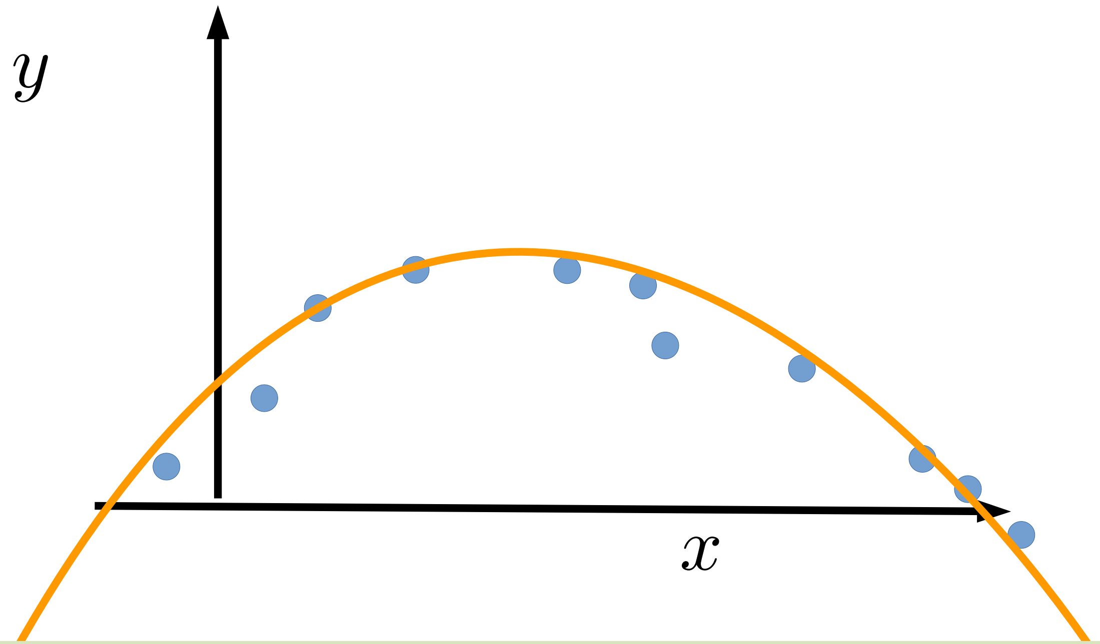
Goodness of fit, fidelity term ...etc

Penalizes complexity

Exe:

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

Overfitting and Model Complexity

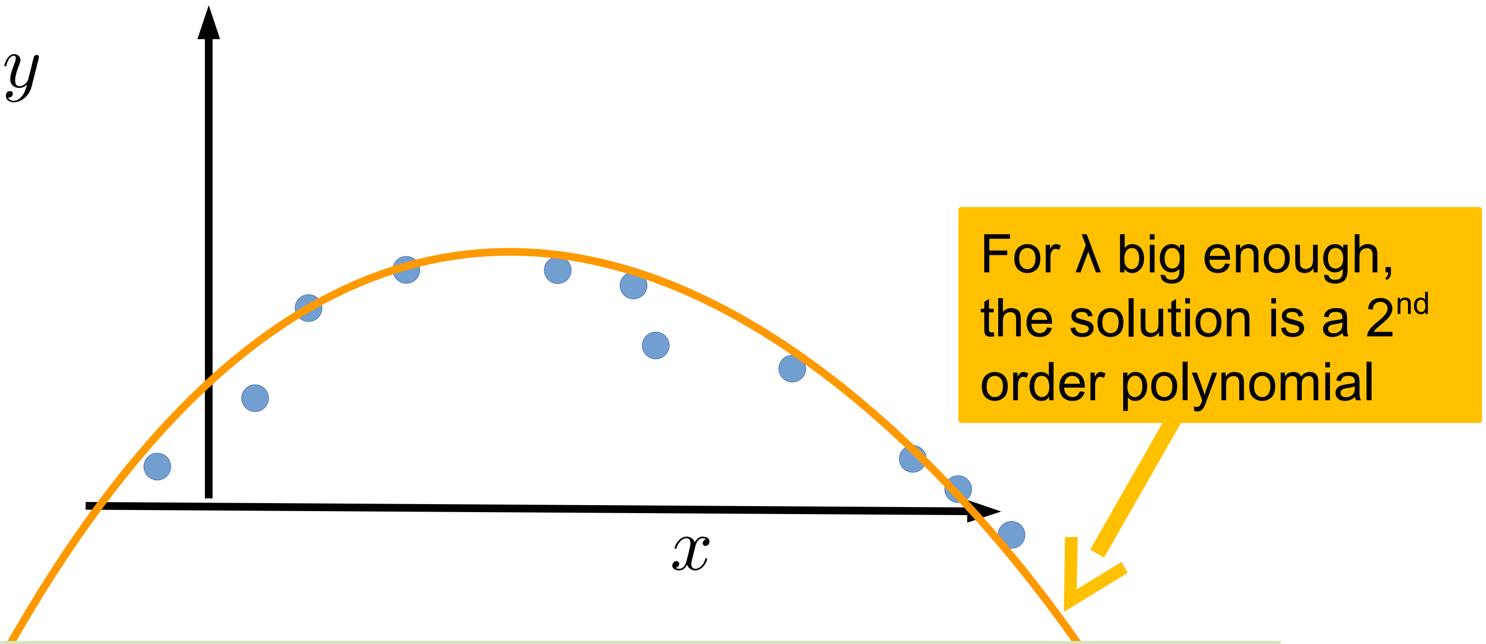


Fitting kth order polynomial

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

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

Overfitting and Model Complexity



Fitting kth 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

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

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: $(x^1, y^1), \ldots, (x^n, y^n)$

(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*:

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

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

Part II: Optimizing Empirical Risk

Re-writing as Sum of Terms

A Datum Function

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

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

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

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

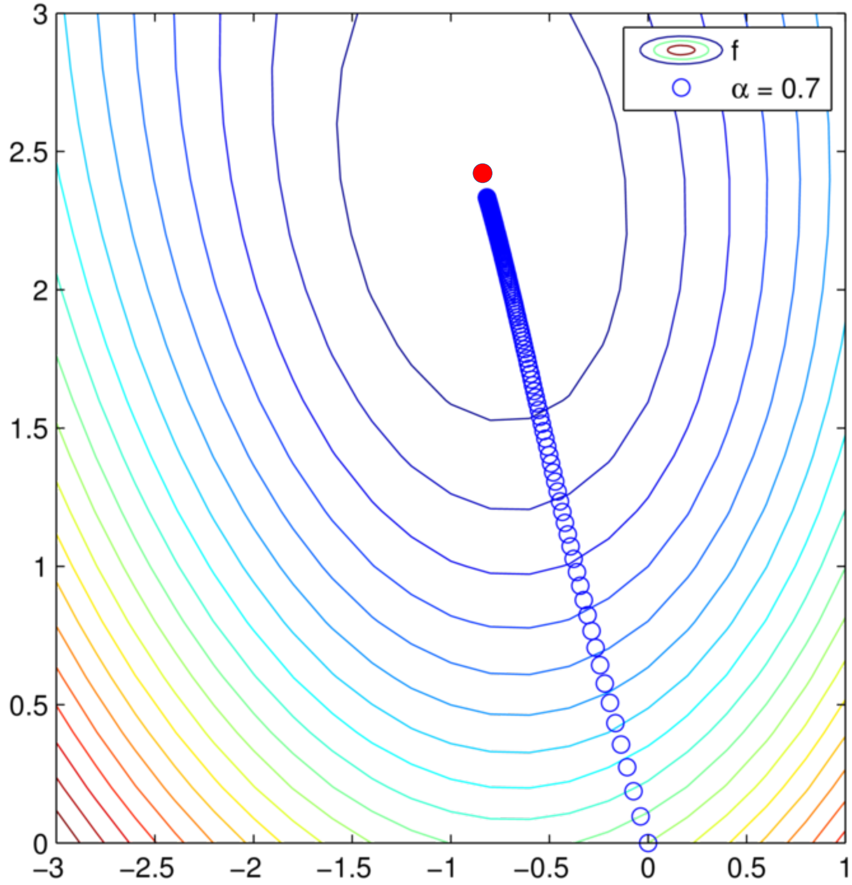
Output w^T

Gradient Descent Example

A Logistic Regression problem using the fourclass labelled data from LIBSVM
(n, d)= (862,2)

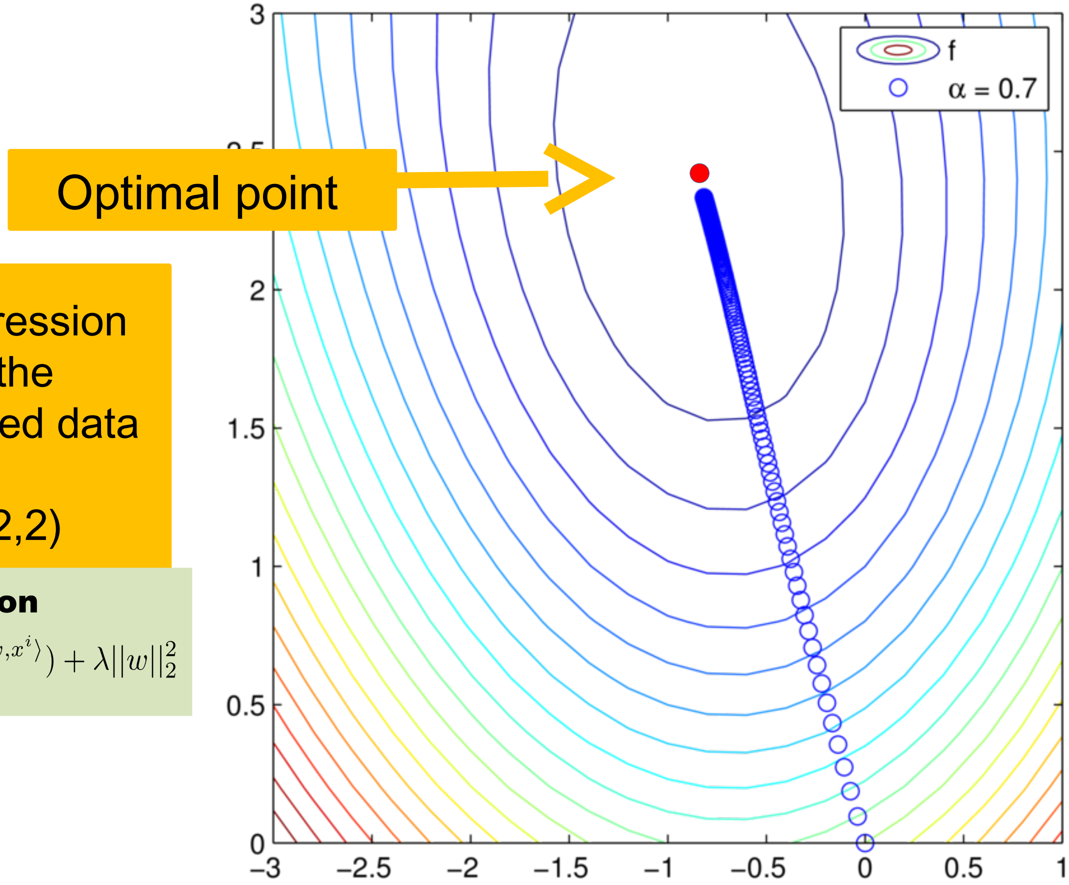
Logistic Regression

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



Can we prove that this always works?

Gradient Descent Example



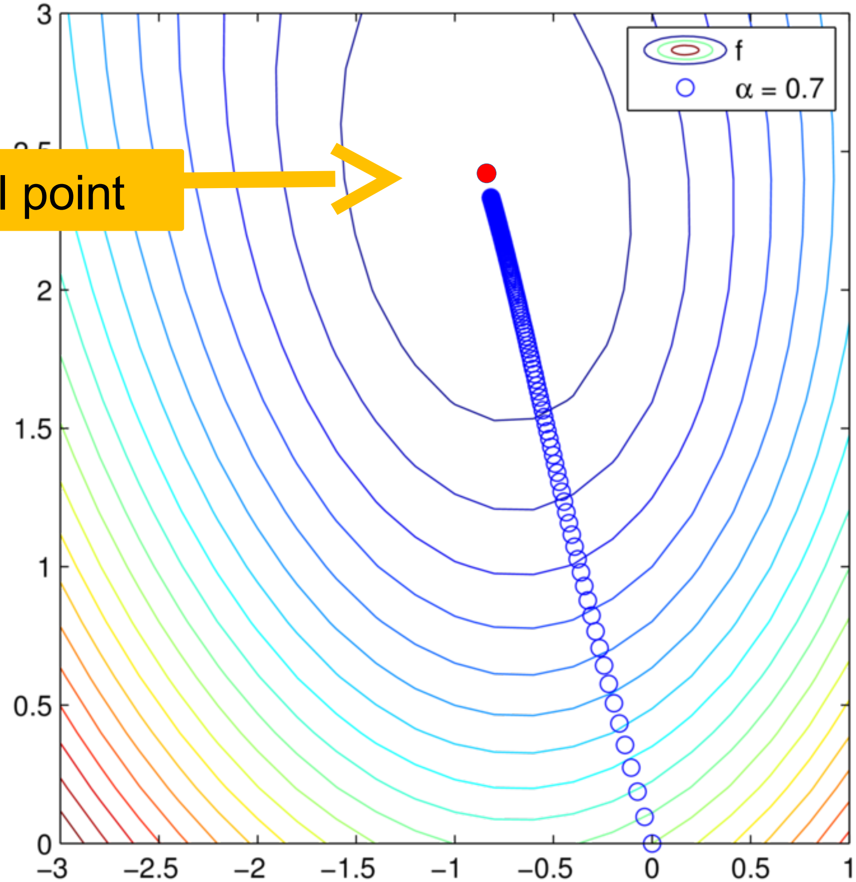
A Logistic Regression problem using the fourclass labelled data from LIBSVM
(n, d)= (862,2)

Logistic Regression

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

Can we prove that this always works?

Gradient Descent Example



Optimal point

A Logistic Regression problem using the fourclass labelled data from LIBSVM

$(n, d) = (862,2)$

Logistic Regression

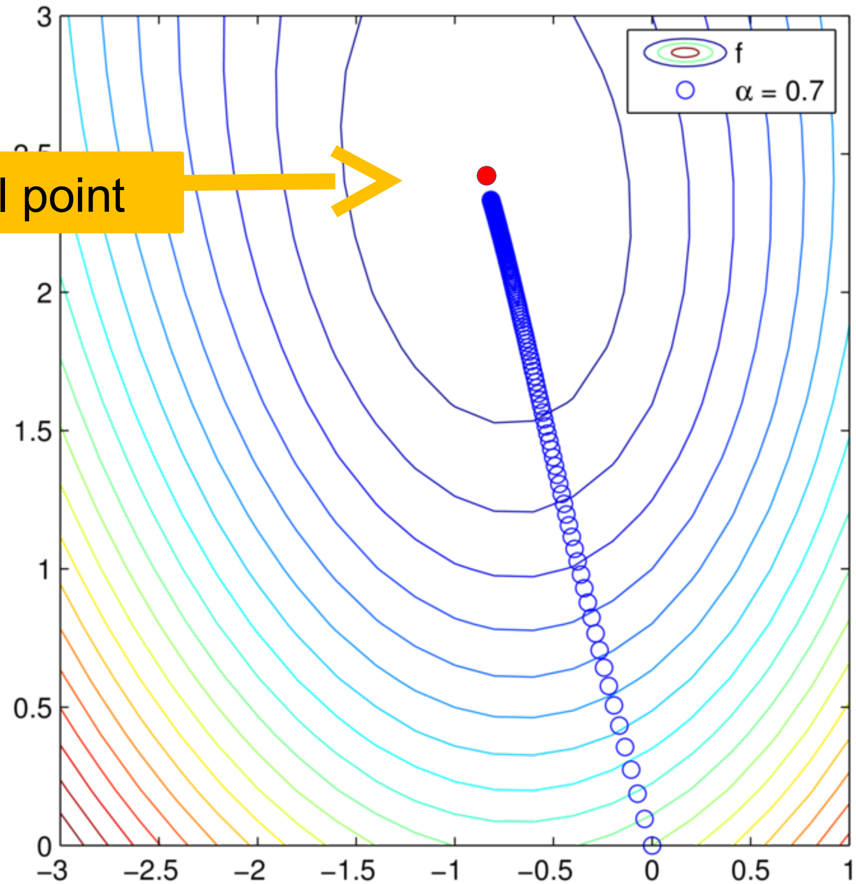
$$\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-y^i \langle w, x^i \rangle}) + \lambda ||w||_2^2$$

Can we prove that this always works?

No! There is no universal optimization method. The “no free lunch” of Optimization

Gradient Descent Example

Optimal point



A Logistic Regression problem using the fourclass labelled data from LIBSVM
$(n, d) = (862, 2)$

Logistic Regression

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

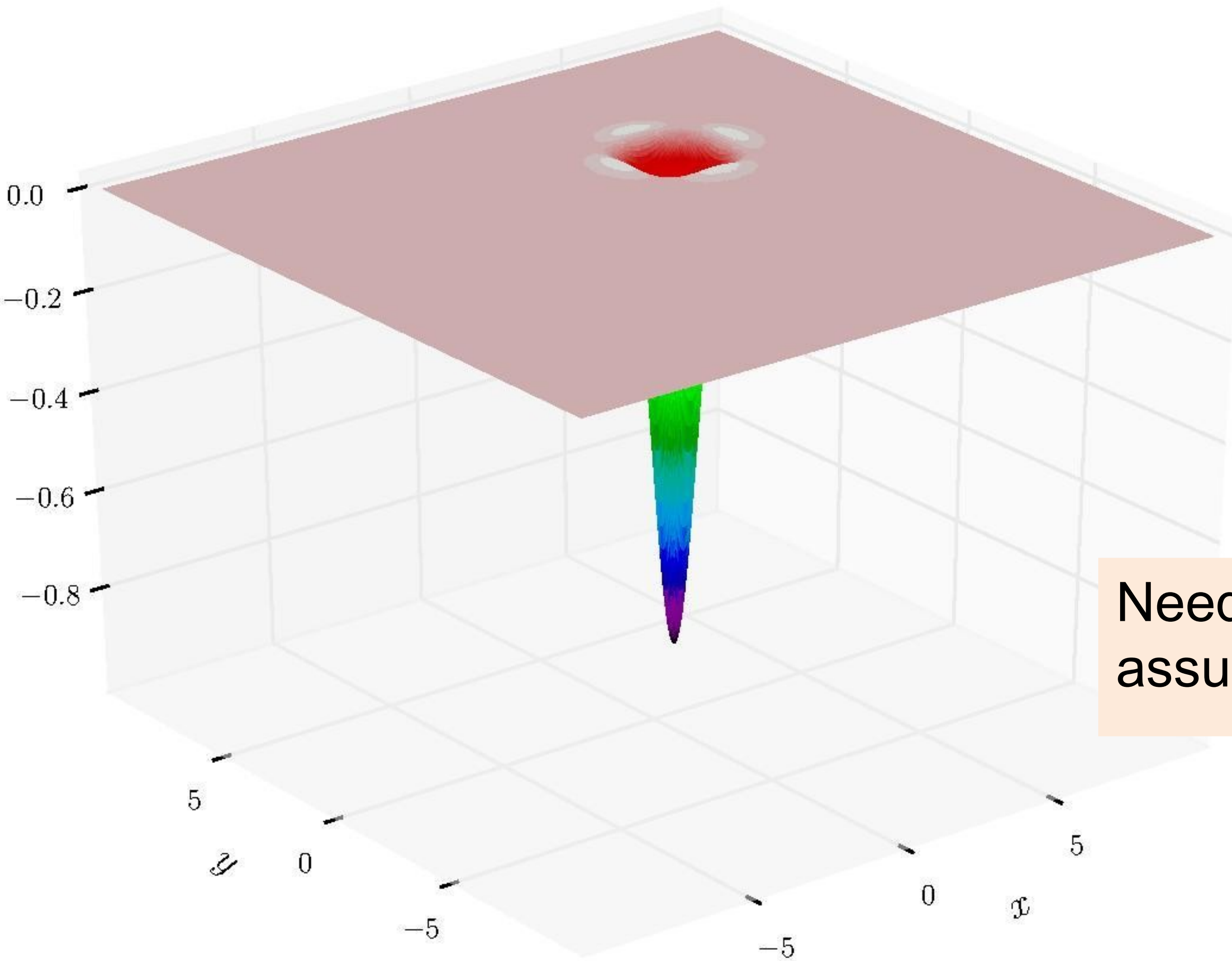
Can we prove that this always works?

No! There is no universal optimization method. The “no free lunch” of Optimization

Specialize

Convex and smooth training problems

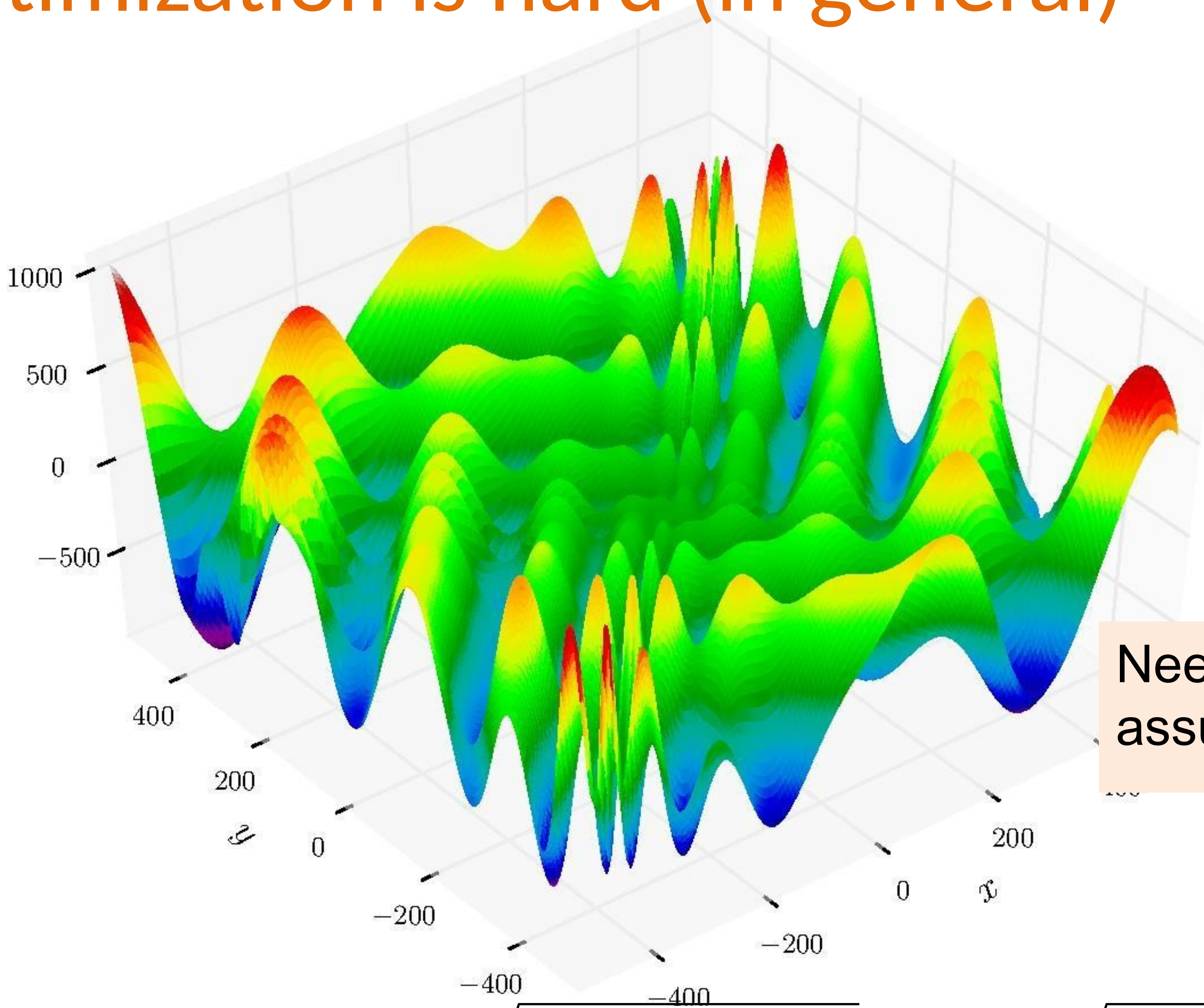
Optimization is hard (in general)



Need assumptions!

$$f(x,y) = -\cos(x)\cos(y)\exp\left(-(x-\pi)^2 - (y-\pi)^2\right)$$

Optimization is hard (in general)



Need assumptions!

$$f(x,y) = -(y+47)\sin\sqrt{\left|\tfrac{x}{2}+(y+47)\right|} - x\sin\sqrt{\left|\tfrac{x}{2}-(y+47)\right|}$$

Main assumption

Nice property

$$\text{If } \nabla f(w^*) = 0 \quad \text{then} \quad f(w^*) \leq f(w), \quad \forall w \in \mathbb{R}^d$$

All stationary points are global minima

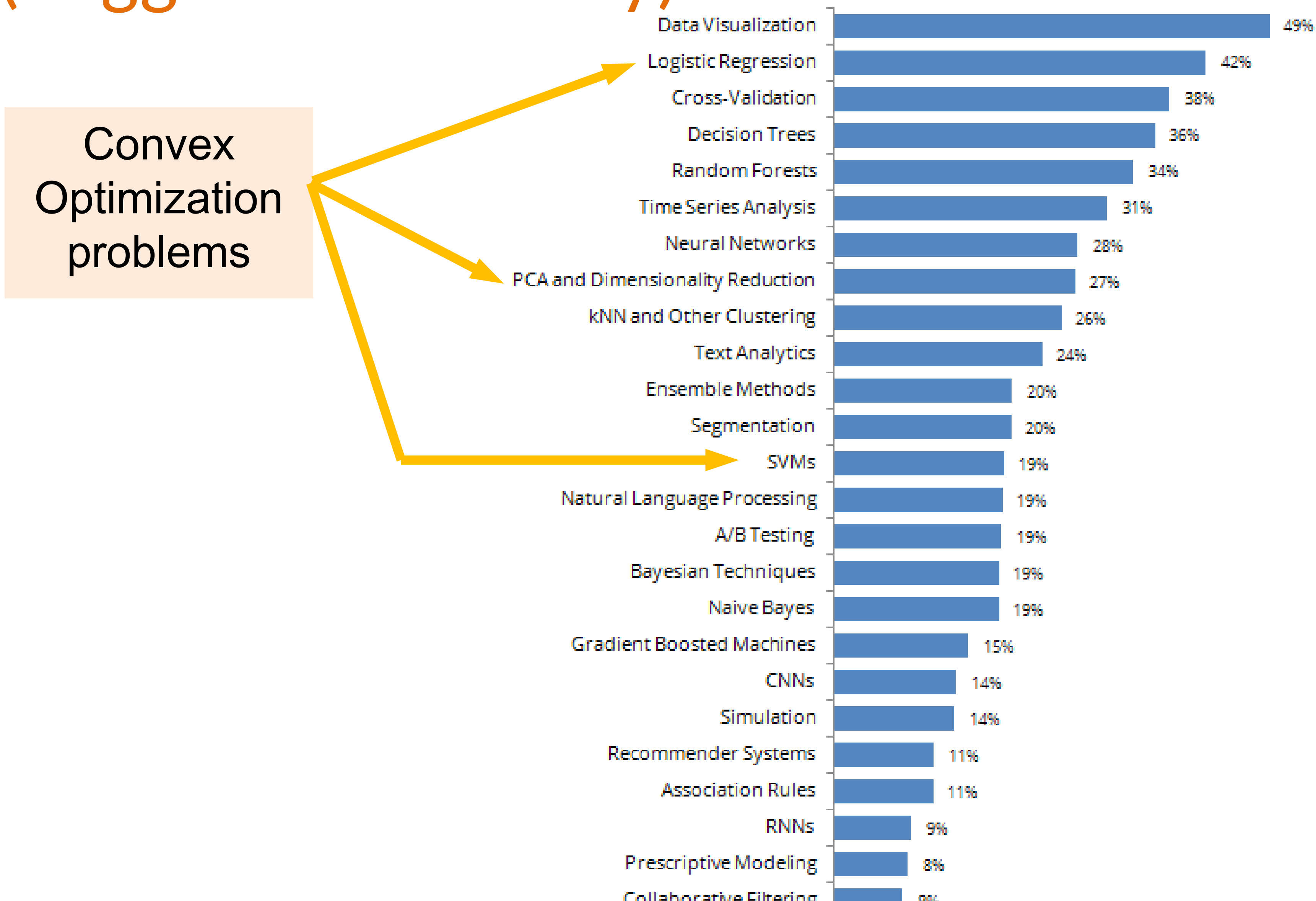
Lemma: Convexity => Nice property

$$\text{If } \; f(w) \geq f(y) + \langle \nabla f(y), w - y \rangle, \quad \forall w, y \in \mathbb{R}^d$$

$$\text{then nice property holds}$$

PROOF: Choose $y = w^*$

Data science methods most used (Kaggle 2017 survey)



Part III: Stochastic Gradient Descent

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 $f_n(w)$ 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*, …, *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*, …, *n*} selected uniformly at random. Then

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

$$\text{Use } \nabla f_j(w) \approx \nabla f(w)$$

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*, …, *n*} selected uniformly at random. Then

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

$$\text{Use } \nabla f_j(w) \approx \nabla f(w)$$

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 $\alpha > 0$

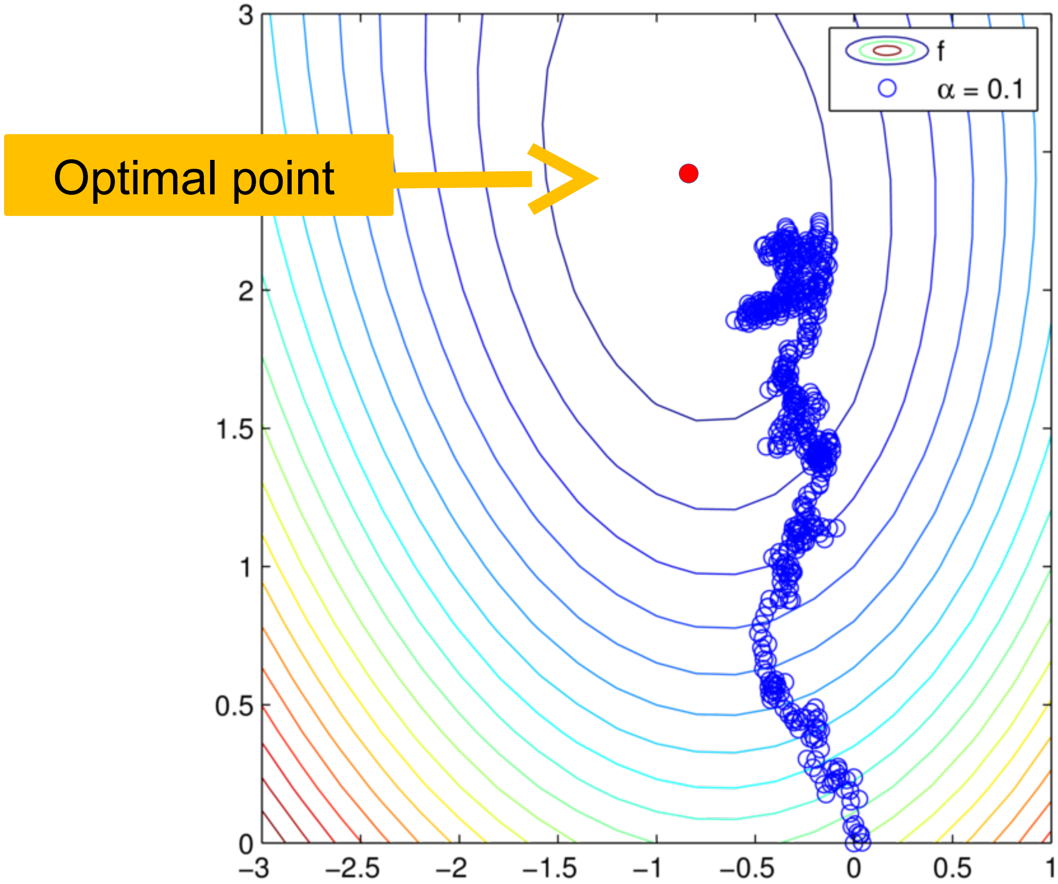
for $t = 0, 1, 2, \ldots, T-1$

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

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

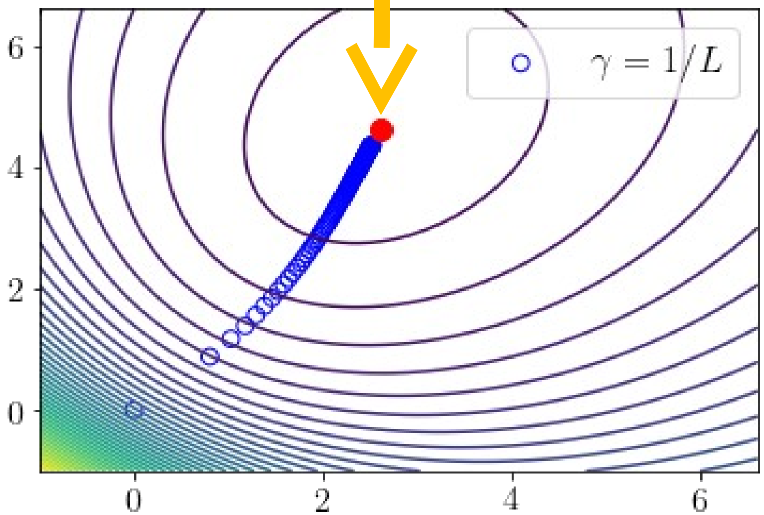
Output w^T

Stochastic Gradient Descent



GD vs Stochastic Gradient Descent

Optimal point w^*

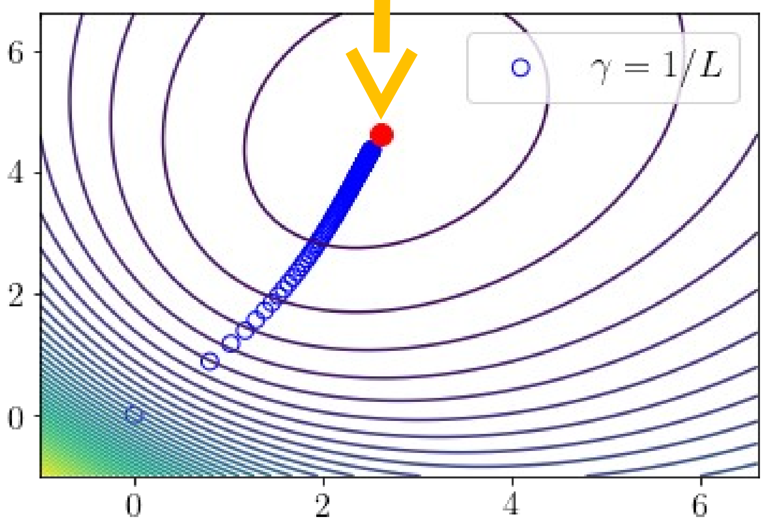


Gradient Descent

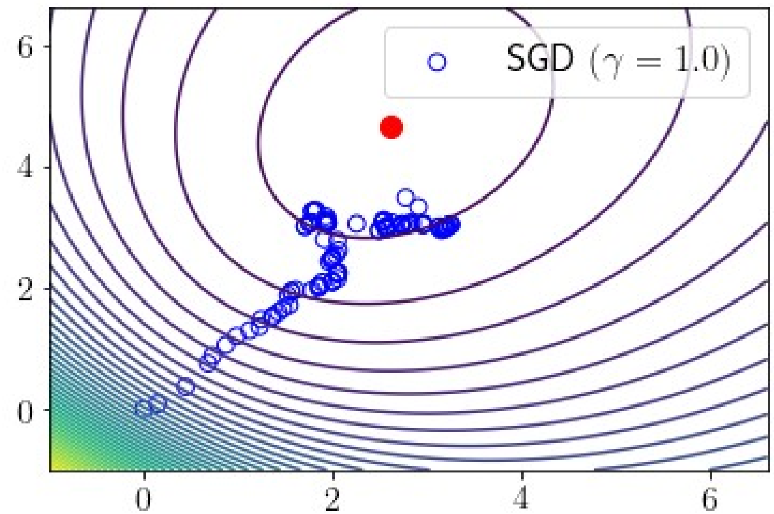
Need Assumptions

GD vs Stochastic Gradient Descent

Optimal point w^*



Gradient Descent



Stochastic Gradient Descent

Why does this happen?

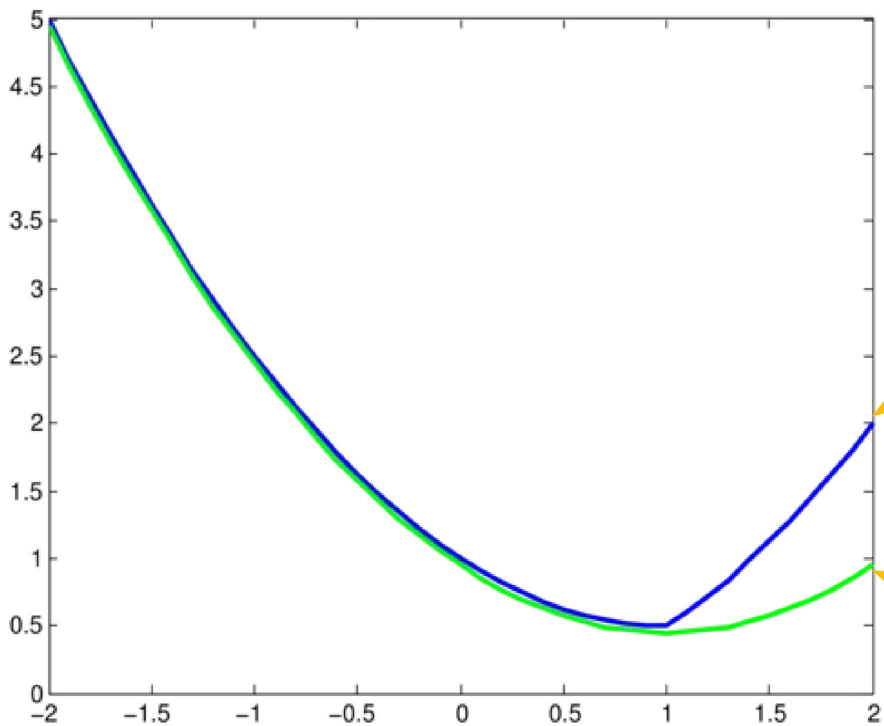
Need Assumptions

Assumption: Strong convexity

We say $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is μ–strongly convex if

$$f(w) \geq f(y) + \langle \nabla f(y), w - y \rangle + \frac{\mu}{2}||w - y||^2, \quad \forall w, y \in \mathbb{R}^n$$

EXE:



Hinge loss + L2

$$\max\{0, 1 - w\} + \frac{1}{2}||w||_2^2$$

Quadratic lower bound

Assumption: Strong convexity

$$f(w) := \frac{1}{n}\sum_{i=1}^{n} \underbrace{\ell\left(h_w(x^i), y^i\right)}_{\substack{\| \\ \text{convex}}} + \underbrace{\lambda R(w)}_{\substack{\| \\ \frac{1}{2}\|w\|^2}}$$

$$\text{strongly convex} \quad = \quad \text{convex} \quad + \quad \frac{1}{2}\|w\|^2$$

Example: SVM with soft margin

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

Not an Example: Neural networks, dictionary learning,
And more

Assumption: Smoothness

We say $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is smooth if

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||, \quad \forall x, y \in \mathbb{R}^n$$

Assumption: Smoothness

We say $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is smooth if

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||, \quad \forall x, y \in \mathbb{R}^n$$

If a twice differentiable $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is L–smooth then

1) $$d^\top \nabla^2 f(x) d \leq L \cdot ||d||_2^2, \quad \forall x, d \in \mathbb{R}^n$$

2) $$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}||x - y||^2, \quad \forall x, y \in \mathbb{R}^n$$

Assumption: Smoothness

We say $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is smooth if

$$||\nabla f(x) - \nabla f(y)|| \leq L||x - y||, \quad \forall x, y \in \mathbb{R}^n$$

If a twice differentiable $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is L–smooth then

1) $$d^\top \nabla^2 f(x) d \leq L \cdot ||d||_2^2, \quad \forall x, d \in \mathbb{R}^n$$

2) $$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}||x - y||^2, \quad \forall x, y \in \mathbb{R}^n$$

EXE: Using that $$\sigma_{\max}(X)^2 ||d||_2^2 \geq ||X^\top d||_2^2$$

Show that $\frac{1}{2}||X^\top w - b||_2^2$ is $\sigma_{\max}(X)^2$–smooth

Smoothness: Examples

Convex quadratics: $x \mapsto x^\top A x + b^\top x + c$

Logistic: $x \mapsto \log\left(1 + e^{-y\langle a, x\rangle}\right)$

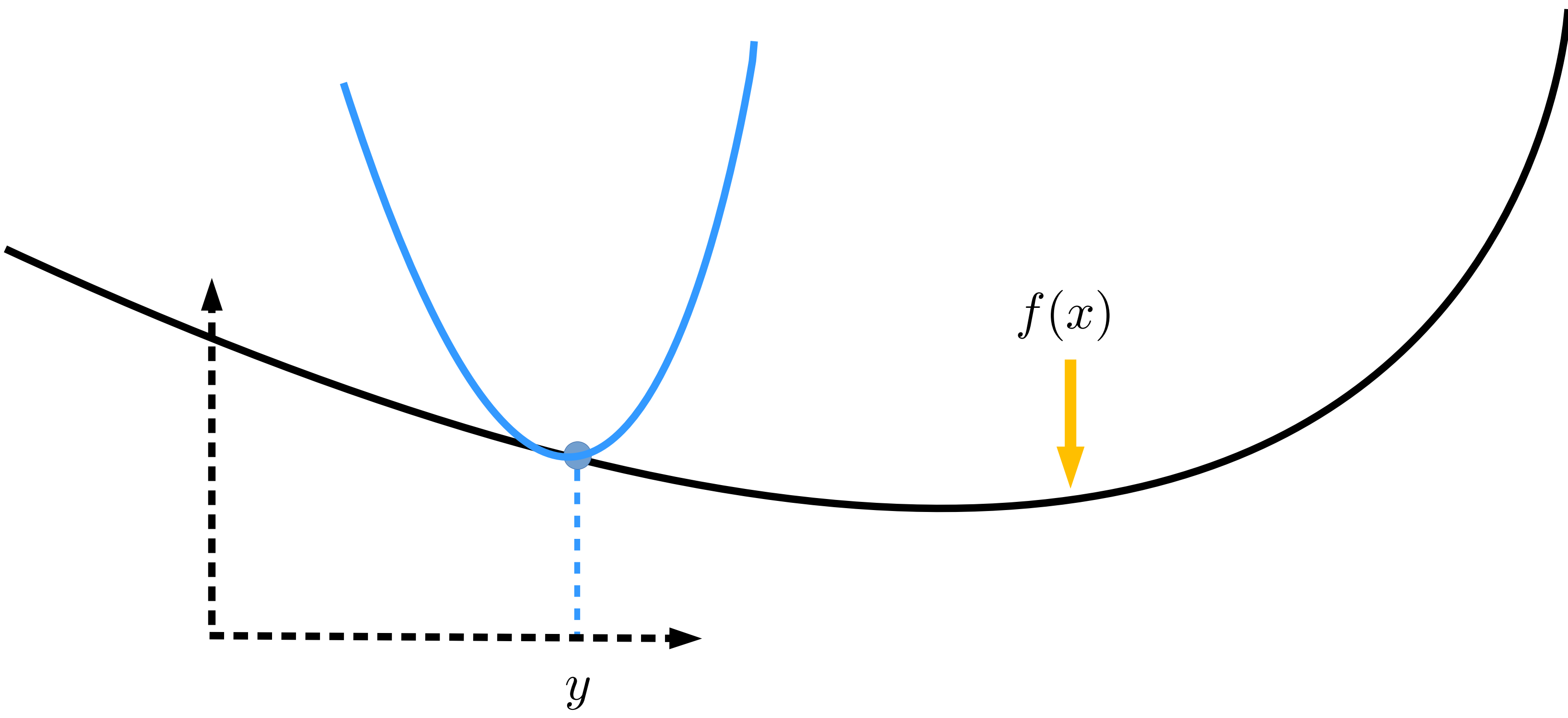
Trigonometric: $x \mapsto \cos(x), \sin(x)$

Proof is an exercise!

Important consequences of Smoothness

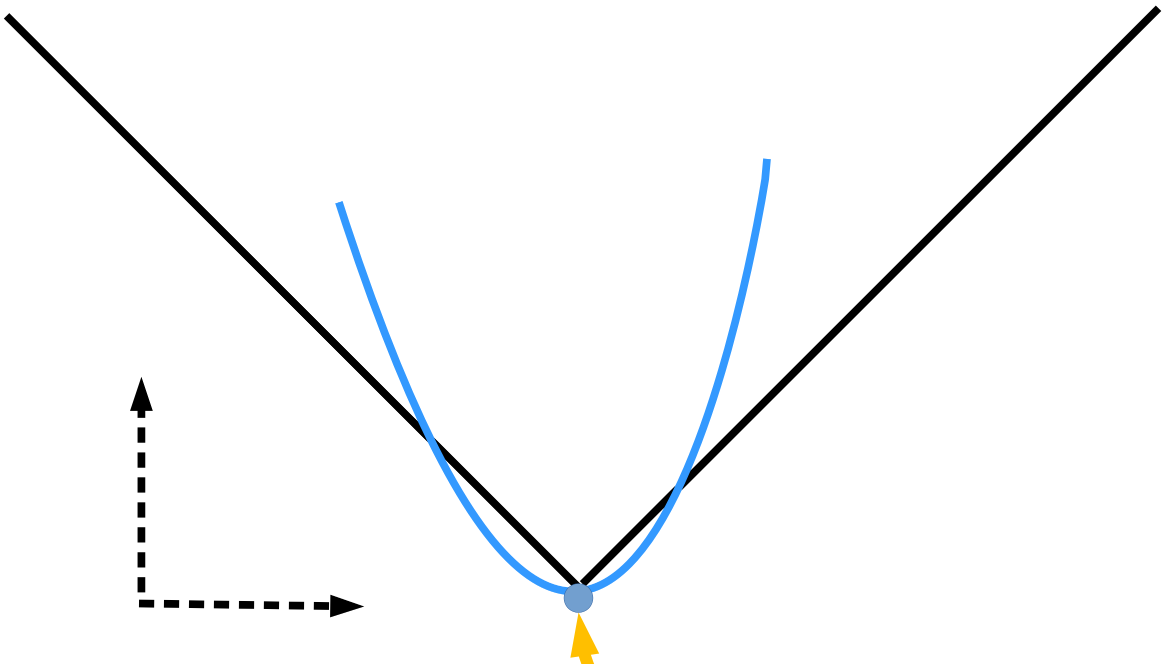
If $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is L–smooth then

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}||x - y||^2, \quad \forall x, y \in \mathbb{R}^n$$



Smoothness: Convex counter-example

$$f(w) = ||w||_1 = \sum_{i=1}^{n} |w_i|$$



Non-smooth can be solver with proximal SGD

Does not fit.
Not smooth

Assumptions for Convergence

Strongly quasi-convexity

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

Each f_i is convex and L_i smooth

$$f_i(y) \leq f_i(w) + \langle \nabla f_i(w), y - w \rangle + \frac{L_i}{2}||y - w||_2^2, \quad \forall w$$

$$L_{\max} := \max_{i=1,\ldots,n} L_i$$

Definition: Gradient Noise

$$\sigma^2 \quad := \quad \mathbb{E}_j[||\nabla f_j(w^*)||_2^2]$$

Assumptions for Convergence

EXE: Calculate the L_i's and L_{max} for

$$1. \quad f(w) = \frac{1}{2n}||X^\top w - y||_2^2 + \frac{\lambda}{2}||w||_2^2$$

HINT: A twice differentiable f_i is L_i -smooth if and only if

$$\nabla^2 f_i(w) \preceq L_i \, I \quad \Leftrightarrow \quad v^\top \nabla^2 f_i(w) v \leq L_i \|v\|^2, \forall v$$

Assumptions for Convergence

EXE: Calculate the L_i's and L_{max} for

$$1. \quad f(w) = \frac{1}{2n}||X^\top w - y||_2^2 + \frac{\lambda}{2}||w||_2^2$$

HINT: A twice differentiable f_i is L_i -smooth if and only if

$$\nabla^2 f_i(w) \preceq L_i \, I \quad \Leftrightarrow \quad v^\top \nabla^2 f_i(w) v \leq L_i \|v\|^2, \forall v$$

$$1. \quad f(w) = \frac{1}{2n}||X^\top w - y||_2^2 + \frac{\lambda}{2}||w||_2^2 = \frac{1}{n}\sum_{i=1}^n (\frac{1}{2}(x_i^\top w - y_i)^2 + \frac{\lambda}{2}||w||_2^2)$$

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

Assumptions for Convergence

EXE: Calculate the L_i's and L_{max} for

$$1. \quad f(w) = \frac{1}{2n}||X^\top w - y||_2^2 + \frac{\lambda}{2}||w||_2^2$$

HINT: A twice differentiable f_i is L_i-smooth if and only if

$$\nabla^2 f_i(w) \preceq L_i\ I \quad \Leftrightarrow \quad v^\top \nabla^2 f_i(w) v \leq L_i \|v\|^2, \forall v$$

$$1. \quad f(w) = \frac{1}{2n}||X^\top w - y||_2^2 + \frac{\lambda}{2}||w||_2^2 = \frac{1}{n}\sum_{i=1}^n \left(\frac{1}{2}(x_i^\top w - y_i)^2 + \frac{\lambda}{2}||w||_2^2\right)$$

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

$$\nabla^2 f_i(w) = x_i x_i^\top + \lambda \quad \preceq \quad (||x_i||_2^2 + \lambda) I \quad = \quad L_i\ I$$

Assumptions for Convergence

EXE: Calculate the L_i's and L_{max} for

$$1.\quad f(w) = \frac{1}{2n}||X^\top w - y||_2^2 + \frac{\lambda}{2}||w||_2^2$$

HINT: A twice differentiable f_i is L_i -smooth if and only if

$$\nabla^2 f_i(w) \preceq L_i\, I \quad \Leftrightarrow \quad v^\top \nabla^2 f_i(w) v \leq L_i \|v\|^2, \forall v$$

$$1.\quad f(w) = \frac{1}{2n}||X^\top w - y||_2^2 + \frac{\lambda}{2}||w||_2^2 = \frac{1}{n}\sum_{i=1}^n (\frac{1}{2}(x_i^\top w - y_i)^2 + \frac{\lambda}{2}||w||_2^2)$$

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

$$\nabla^2 f_i(w) = x_i x_i^\top + \lambda \quad \preceq \quad (||x_i||_2^2 + \lambda)I \quad = \quad L_i\, I$$

$$L_{\max} \quad = \quad \max_{i=1,\ldots,n} (||x_i||_2^2 + \lambda) \quad = \quad \max_{i=1,\ldots,n} ||x_i||_2^2 + \lambda$$

Assumptions for Convergence

EXE: Calculate the L_i's and L_{max} for

$$2. \quad f(w) = \frac{1}{n}\sum_{i=1}^{n} \ln(1+e^{-y_i\langle w, a_i\rangle}) + \frac{\lambda}{2}||w||_2^2$$

Assumptions for Convergence

EXE: Calculate the L_i's and L_{max} for

$$2. \quad f(w) = \frac{1}{n}\sum_{i=1}^{n} \ln(1+e^{-y_i\langle w, a_i\rangle}) + \frac{\lambda}{2}||w||_2^2$$

$$2. \quad f_i(w) = \ln(1+e^{-y_i\langle w, a_i\rangle}) + \frac{\lambda}{2}||w||_2^2,$$

Assumptions for Convergence

EXE: Calculate the L_i's and L_{max} for

$$2. \quad f(w) = \frac{1}{n}\sum_{i=1}^n \ln(1+e^{-y_i\langle w, a_i\rangle}) + \frac{\lambda}{2}||w||_2^2$$

$$2. \quad f_i(w) = \ln(1+e^{-y_i\langle w, a_i\rangle}) + \frac{\lambda}{2}||w||_2^2,$$

$$\nabla f_i(w) = \frac{-y_i a_i e^{-y_i\langle w, a_i\rangle}}{1+e^{-y_i\langle w, a_i\rangle}} + \lambda w$$

$$\nabla^2 f_i(w) = a_i a_i^\top \left(\frac{(1+e^{-y_i\langle w, a_i\rangle})e^{-y_i\langle w, a_i\rangle}}{(1+e^{-y_i\langle w, a_i\rangle})^2} - \frac{e^{-2y_i\langle w, a_i\rangle}}{(1+e^{-y_i\langle w, a_i\rangle})^2} \right) + \lambda I$$

$$= a_i a_i^\top \frac{e^{-y_i\langle w, a_i\rangle}}{(1+e^{-y_i\langle w, a_i\rangle})^2} + \lambda I \quad \preceq \quad \left(\frac{||a_i||_2^2}{4} + \lambda \right) I = L_i \; I$$

Complexity / Convergence

Theorem

If f is μ–str. convex, f_i is convex, L_i–smooth, $\alpha \in [0, \frac{1}{2L_{\max}}]$ then the iterates of the SGD satisfy

$\sigma^2 := \mathbb{E}_j[||\nabla f_j(w^*)||_2^2]$

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

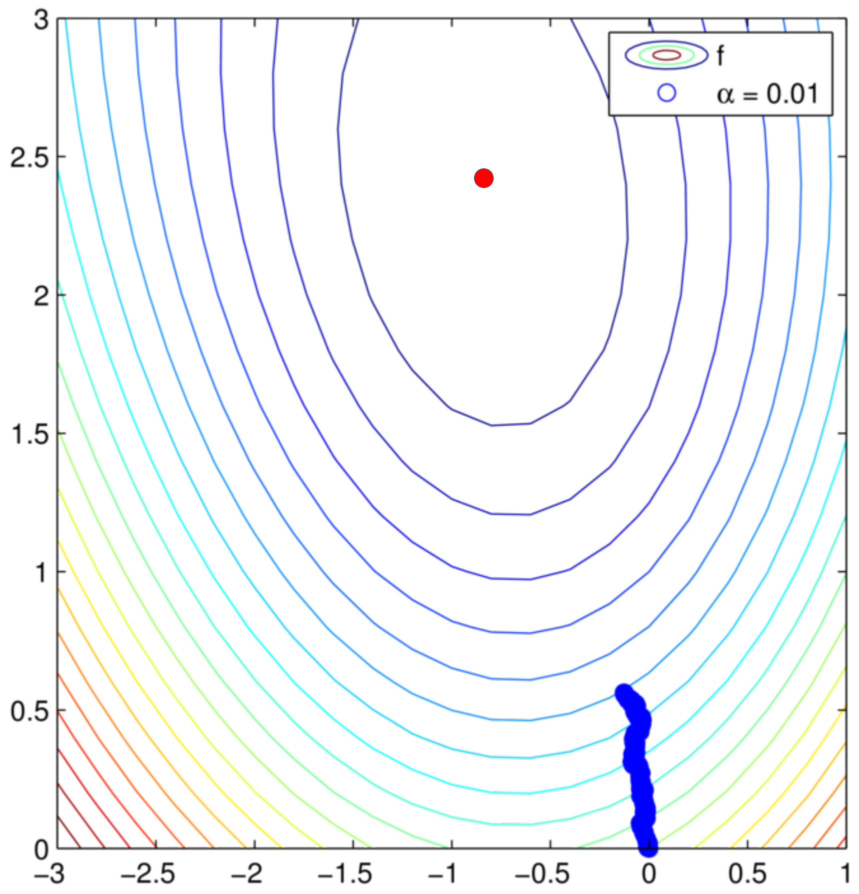
Shows that $\alpha \approx \frac{1}{\mu}$

Shows that $\alpha \approx 0$

RMG, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, P. Richtarik, ICML 2019, arXiv:1901.09401
SGD: General Analysis and Improved Rates.

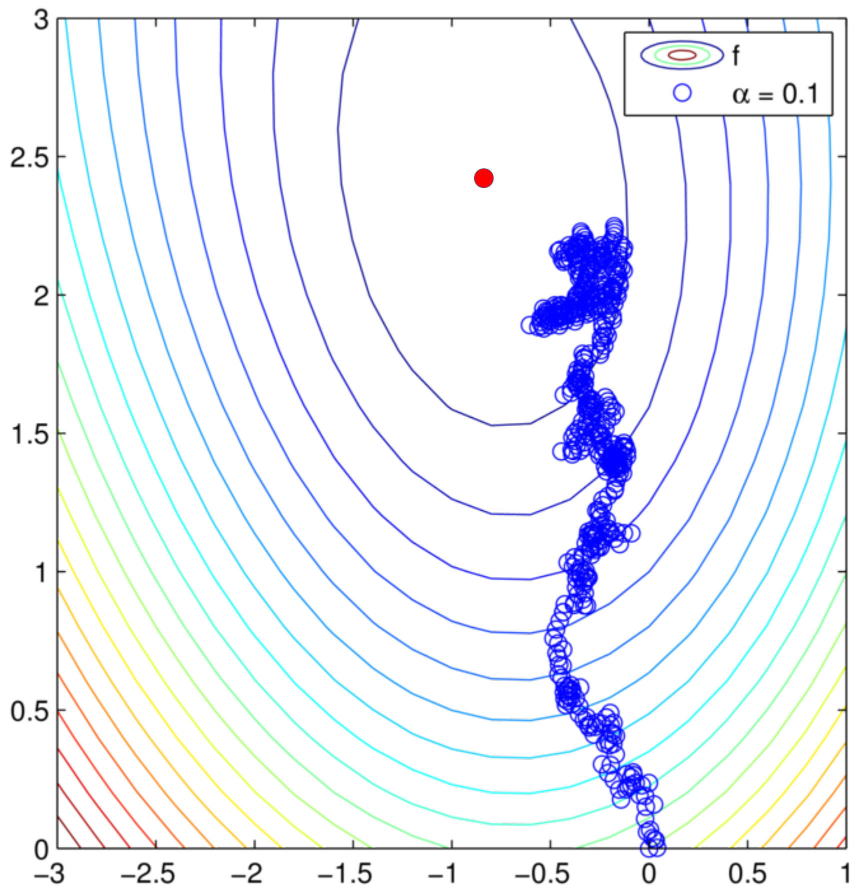
Stochastic Gradient Descent

α =0.01



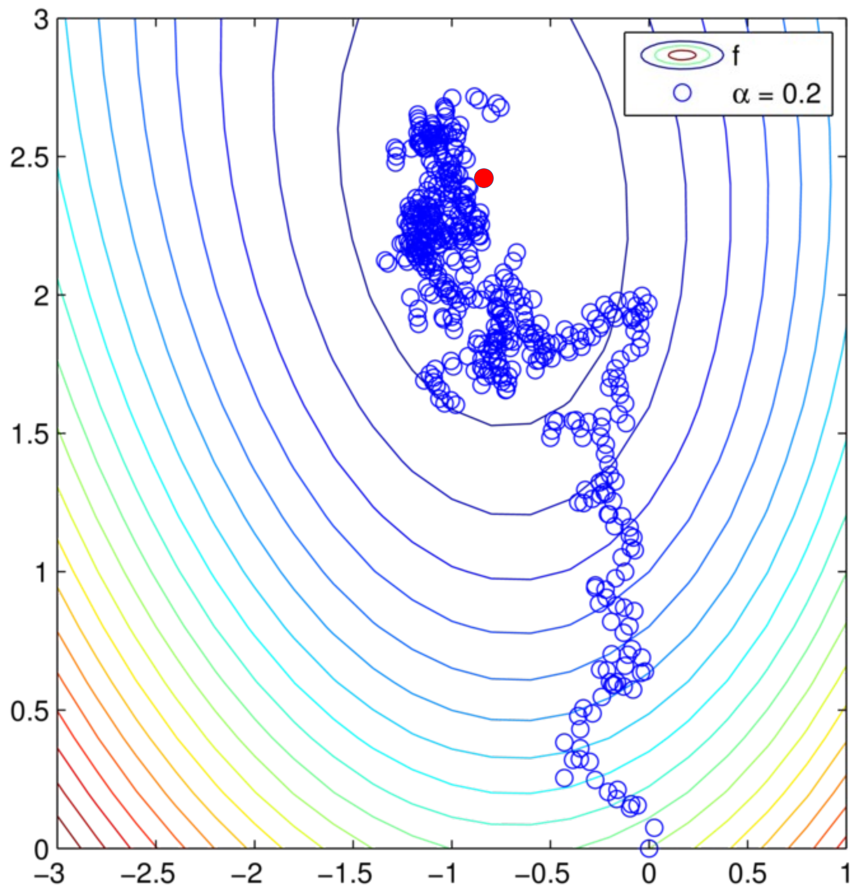
Stochastic Gradient Descent

α =0.1



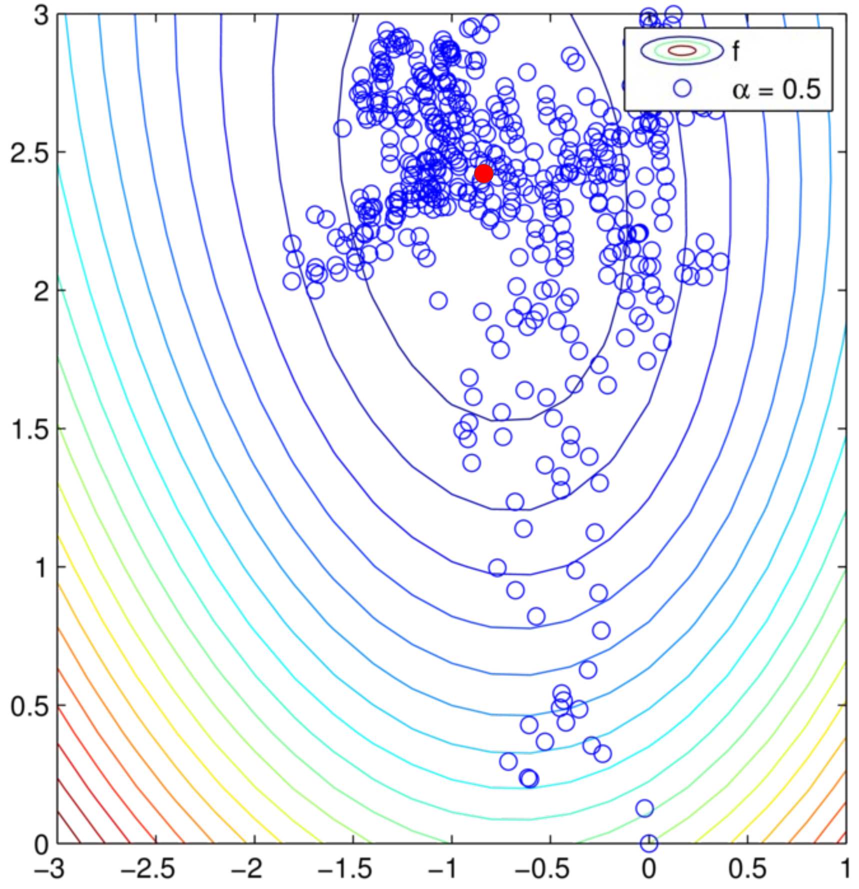
Stochastic Gradient Descent

α =0.2



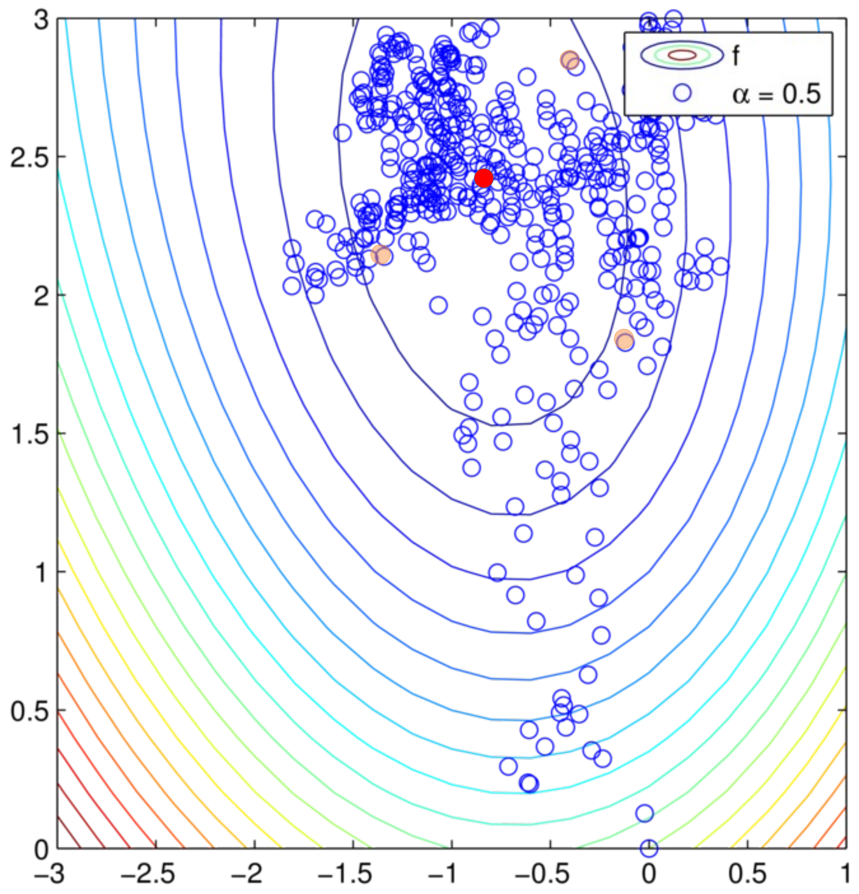
Stochastic Gradient Descent

α =0.5



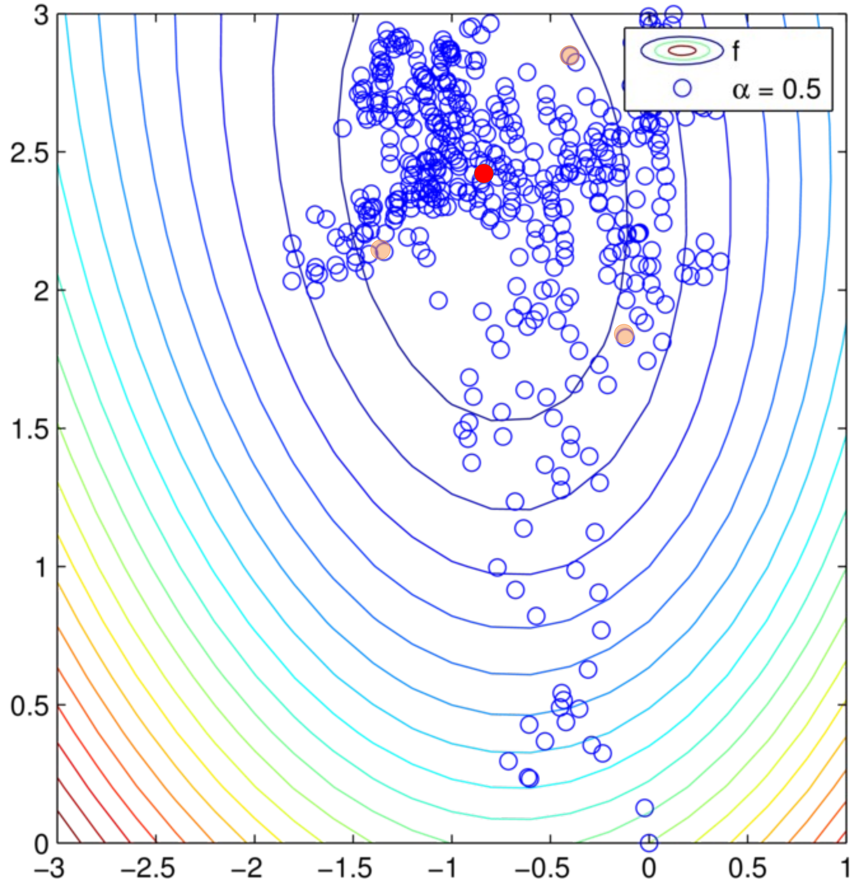
Stochastic Gradient Descent

α =0.5



Stochastic Gradient Descent

α =0.5

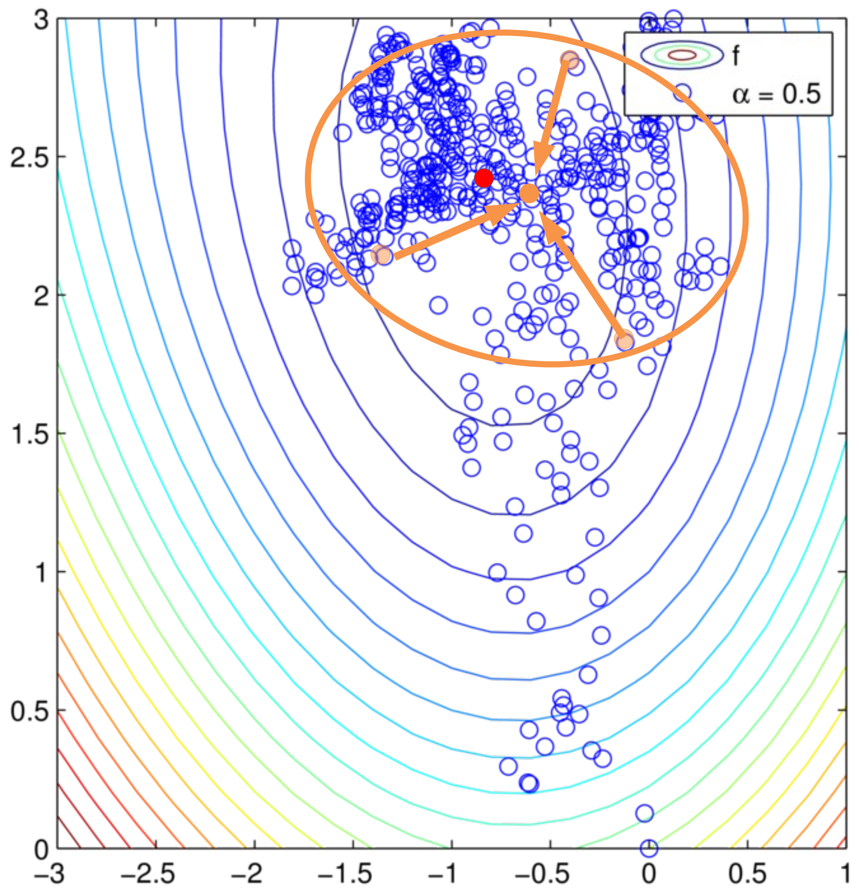


1) Start with big steps and end with smaller steps

2) Try averaging the points

Stochastic Gradient Descent

α =0.5



1) Start with big steps and end with smaller steps

2) Try averaging the points

SGD shrinking stepsize

SGD Shrinking stepsize

Set $w^0 = 0$

Choose $\alpha_t > 0$, $\alpha_t \to 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$

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

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

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

Output w^T

Shrinking Stepsize

SGD shrinking stepsize

SGD Shrinking stepsize

Set $w^0 = 0$

Choose $\alpha_t > 0,\ \alpha_t \to 0,\ \sum_{t=0}^{\infty} \alpha_t = \infty$

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

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

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

Output w^T

Shrinking Stepsize

How should we sample *j* ?

How fast $\alpha_t \to 0$?

Does this converge?

Complexity / Convergence

Theorem for switching to shrinking stepsizes

If f is μ–str. convex, f_i is convex and L_i–smooth.

Let $\mathcal{K} := L_{\max}/\mu$ and let

$$\alpha^t = \begin{cases} \frac{1}{2L_{\max}} & \text{for} \quad t \leq 4\lceil \mathcal{K} \rceil \\ \frac{2t+1}{(t+1)^2\mu} & \text{for} \quad t > 4\lceil \mathcal{K} \rceil. \end{cases}$$

If $t \geq 4\lceil \mathcal{K} \rceil$, then the SGD iterates converge

$$\mathbb{E}\|w^t - w^*\|^2 \leq \frac{\sigma^2}{\mu^2}\frac{8}{t} + \frac{16}{e^2}\frac{\lceil \mathcal{K} \rceil^2}{t^2}\|w^0 - w^*\|^2$$

Complexity / Convergence

Theorem for switching to shrinking stepsizes

If f is μ–str. convex, f_i is convex and L_i–smooth.

Let $\mathcal{K} := L_{\max}/\mu$ and let

$$\alpha^t = \begin{cases} \frac{1}{2L_{\max}} & \text{for} \quad t \leq 4\lceil \mathcal{K} \rceil \\ \frac{2t+1}{(t+1)^2\mu} & \text{for} \quad t > 4\lceil \mathcal{K} \rceil. \end{cases}$$

$\alpha^t = O(1/(t+1))$

If $t \geq 4\lceil \mathcal{K} \rceil$, then the SGD iterates converge

$$\mathbb{E}\|w^t - w^*\|^2 \leq \frac{\sigma^2}{\mu^2}\frac{8}{t} + \frac{16}{e^2}\frac{\lceil \mathcal{K} \rceil^2}{t^2}\|w^0 - w^*\|^2$$

Complexity / Convergence

Theorem for switching to shrinking stepsizes

If f is μ–str. convex, f_i is convex and L_i–smooth.

Let $\mathcal{K} := L_{\max}/\mu$ and let

$$\alpha^t = \begin{cases} \frac{1}{2L_{\max}} & \text{for} \quad t \leq 4\lceil\mathcal{K}\rceil \\ \frac{2t+1}{(t+1)^2\mu} & \text{for} \quad t > 4\lceil\mathcal{K}\rceil. \end{cases}$$

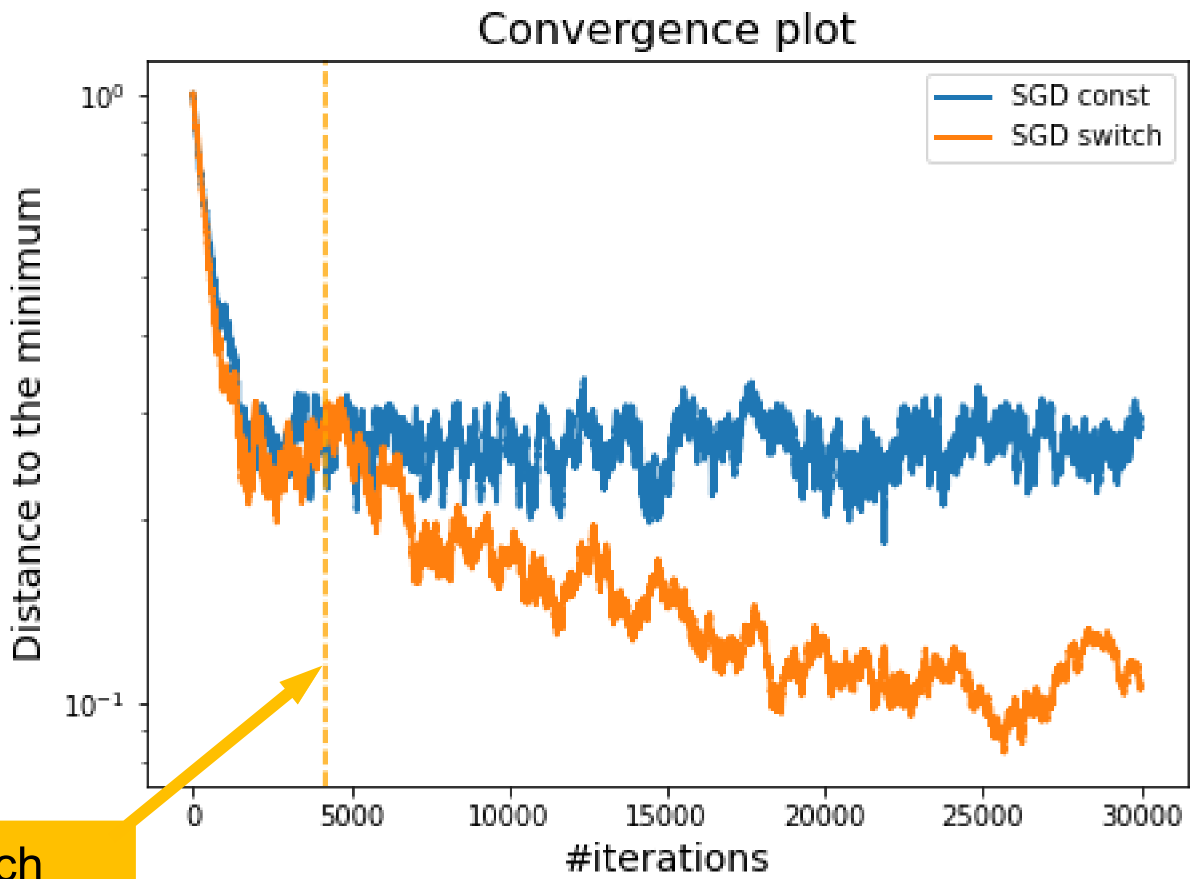
$\alpha^t = O(1/(t+1))$

If $t \geq 4\lceil\mathcal{K}\rceil$, then the SGD iterates converge

$$\mathbb{E}\|w^t - w^*\|^2 \leq \frac{\sigma^2}{\mu^2}\frac{8}{t} + \frac{16}{e^2}\frac{\lceil\mathcal{K}\rceil^2}{t^2}\|w^0 - w^*\|^2$$

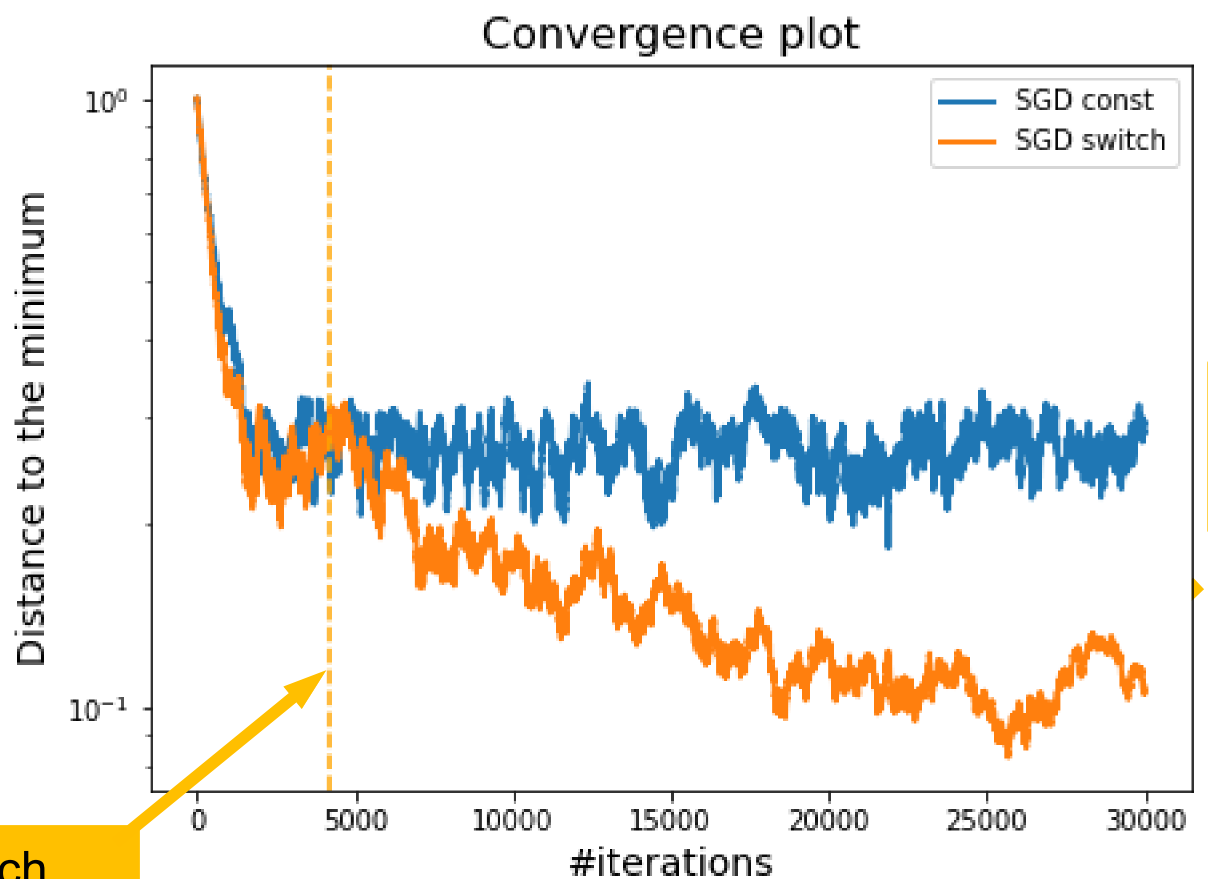
In practice often $\alpha^t = C/\sqrt{t+1}$ where C is tuned

Stochastic Gradient Descent with switch to decreasing stepsizes



Switch point
$t = 4\lceil \mathcal{K} \rceil$

Stochastic Gradient Descent with switch to decreasing stepsizes



Noisy iterates. Take averages?

Switch point $t = 4\lceil \mathcal{K} \rceil$

SGD with (late start) averaging

SGD with late averaging

Set $w^0 = 0$

Choose $\alpha_t > 0$, $\alpha_t \to 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$

Choose averaging start $s_0 \in \mathbb{N}$

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

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

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

if $t > s_0$

$$\overline{w} = \frac{1}{t-s_0} \sum_{i=s_0}^{t} w^t$$

else: $\overline{w} = w$

Output $\overline{w}$

B. T. Polyak and A. B. Juditsky, SIAM Journal on Control and Optimization (1992)
Acceleration of stochastic approximation by averaging

SGD with (late start) averaging

SGD with late averaging

Set $w^0 = 0$

Choose $\alpha_t > 0,\ \alpha_t \rightarrow 0,\ \sum_{t=0}^{\infty} \alpha_t = \infty$

Choose averaging start $s_0 \in \mathbb{N}$

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

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

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

  if $t > s_0$

    $\overline{w} = \frac{1}{t-s_0} \sum_{i=s_0}^{t} w^t$

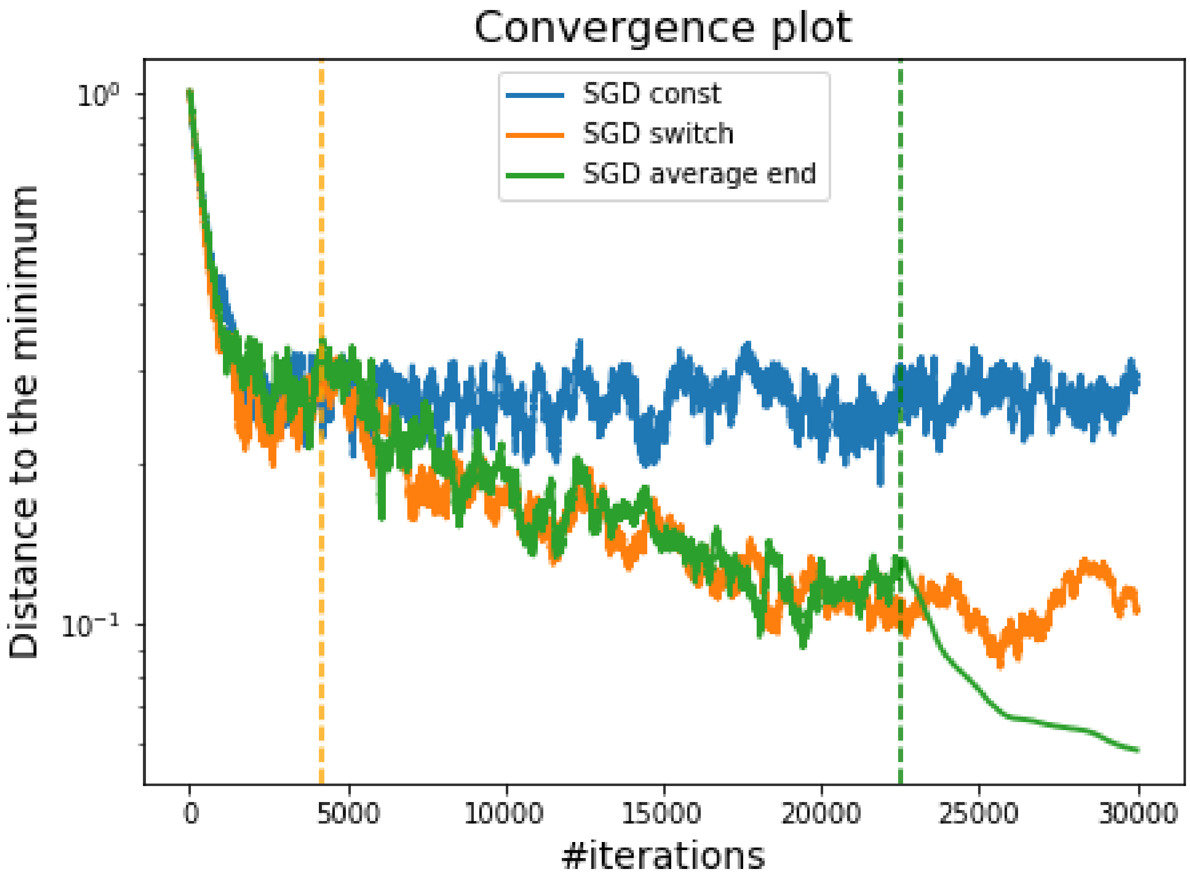
  else: $\overline{w} = w$

Output $\overline{w}$

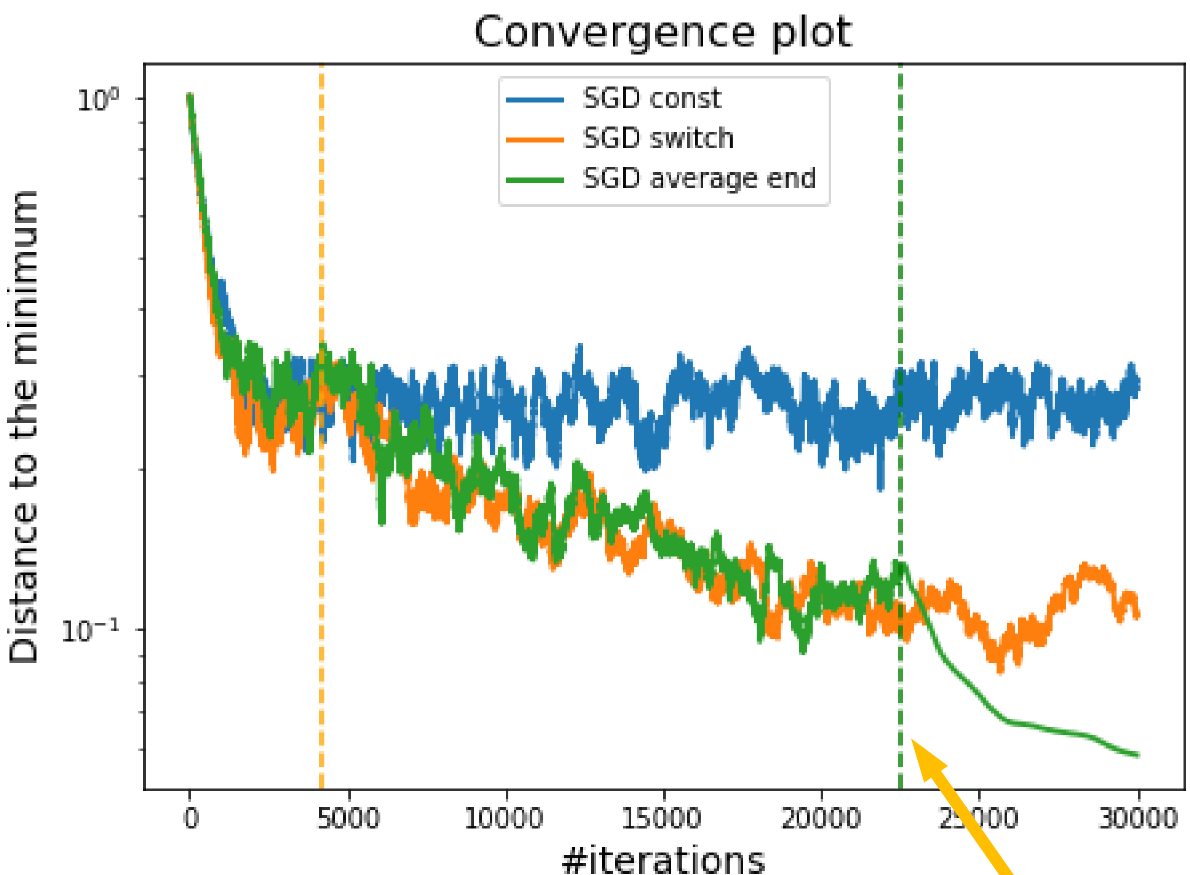
This is not efficient. How to make this efficient?

B. T. Polyak and A. B. Juditsky, SIAM Journal on Control and Optimization (1992)
Acceleration of stochastic approximation by averaging

Stochastic Gradient Descent
Averaging the last few iterates



Stochastic Gradient Descent
Averaging the last few iterates



Averaging starts here

Part III.2: Stochastic Gradient Descent for Sparse Data

Lazy SGD updates for Sparse Data

Finite Sum Training Problem

L2 regularizor + linear hypothesis

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

Let x^i have at most $s \in \mathbb{N}$ nonzero elements for all i.
How many operations does each SGD step cost?

Sparse Examples:
encoding of categorical variables (hot one encoding), word2vec, recommendation systems ...etc

Lazy SGD updates for Sparse Data

Finite Sum Training Problem

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

L2 regularizor + linear hypothesis

Let x^i have at most $s \in \mathbb{N}$ nonzero elements for all i.
How many operations does each SGD step cost?

$$\begin{aligned} w^{t+1} &= w^t - \alpha_t \left(\ell'(\langle w^t, x^i \rangle, y^i) x^i + \lambda w^t\right) \\ &= (1 - \lambda \alpha_t) w^t - \alpha_t \ell'(\langle w^t, x^i \rangle, y^i) x^i \end{aligned}$$

Sparse Examples:
encoding of categorical variables (hot one encoding), word2vec, recommendation systems ...etc

Lazy SGD updates for Sparse Data

Finite Sum Training Problem

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

L2 regularizor + linear hypothesis

Let x^i have at most $s \in \mathbb{N}$ nonzero elements for all i.
How many operations does each SGD step cost?

$$\begin{aligned} w^{t+1} &= w^t - \alpha_t \left(\ell'(\langle w^t, x^i \rangle, y^i) x^i + \lambda w^t\right) \\ &= \underbrace{(1 - \lambda\alpha_t) w^t}_{\text{Rescaling } O(d)} - \underbrace{\alpha_t \ell'(\langle w^t, x^i \rangle, y^i) x^i}_{\text{Addition sparse vector } O(s)} \end{aligned}$$

Rescaling $O(d)$ $+$ Addition sparse vector $O(s)$ $=$ $O(d)$

Sparse Examples:
encoding of categorical variables (hot one encoding), word2vec, recommendation systems ...etc

Lazy SGD updates for Sparse Data

SGD step

$$w^{t+1} = (1 - \lambda\alpha_t)w^t - \alpha_t \ell'(\langle w^t, x^i\rangle, y^i)x^i$$

EXE: re-write the iterates using $w^t = \beta_t z^t$ where $\beta_t \in \mathbb{R}$, $z^t \in \mathbb{R}^d$

Can you update β_t and z^t so that each iteration is $O(s)$?

Lazy SGD updates for Sparse Data

SGD step

$$w^{t+1} = (1-\lambda\alpha_t)w^t - \alpha_t \ell'(\langle w^t, x^i\rangle, y^i)x^i$$

EXE: re-write the iterates using $w^t = \beta_t z^t$ where $\beta_t \in \mathbb{R},\ z^t \in \mathbb{R}^d$

Can you update β_t and z^t so that each iteration is $O(s)$?

$$\beta_{t+1} z^{t+1} = (1-\lambda\alpha_t)\beta_t z^t - \alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)x^i$$

Lazy SGD updates for Sparse Data

SGD step

$$w^{t+1} = (1-\lambda\alpha_t)w^t - \alpha_t \ell'(\langle w^t, x^i\rangle, y^i)x^i$$

EXE: re-write the iterates using $w^t = \beta_t z^t$ where $\beta_t \in \mathbb{R},\ z^t \in \mathbb{R}^d$

Can you update β_t and z^t so that each iteration is $O(s)$?

$$\begin{aligned}\beta_{t+1} z^{t+1} &= (1-\lambda\alpha_t)\beta_t z^t - \alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)x^i \\ &= (1-\lambda\alpha_t)\beta_t\left(z^t - \frac{\alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)}{(1-\lambda\alpha_t)\beta_t}x^i\right)\end{aligned}$$

Lazy SGD updates for Sparse Data

SGD step

$$w^{t+1} = (1-\lambda\alpha_t)w^t - \alpha_t \ell'(\langle w^t, x^i\rangle, y^i)x^i$$

EXE: re-write the iterates using $w^t = \beta_t z^t$ where $\beta_t \in \mathbb{R}$, $z^t \in \mathbb{R}^d$

Can you update β_t and z^t so that each iteration is $O(s)$?

$$\begin{aligned}\beta_{t+1} z^{t+1} &= (1-\lambda\alpha_t)\beta_t z^t - \alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)x^i \\ &= \underbrace{(1-\lambda\alpha_t)\beta_t}_{\beta_{t+1}} \left(\underbrace{z^t - \frac{\alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)}{(1-\lambda\alpha_t)\beta_t} x^i}_{z^{t+1}}\right)\end{aligned}$$

Lazy SGD updates for Sparse Data

SGD step

$$w^{t+1} = (1-\lambda\alpha_t)w^t - \alpha_t \ell'(\langle w^t, x^i\rangle, y^i)x^i$$

EXE: re-write the iterates using $w^t = \beta_t z^t$ where $\beta_t \in \mathbb{R}, \; z^t \in \mathbb{R}^d$

Can you update β_t and z^t so that each iteration is $O(s)$?

$$\beta_{t+1} z^{t+1} = (1-\lambda\alpha_t)\beta_t z^t - \alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)x^i$$

$$= \underbrace{(1-\lambda\alpha_t)\beta_t}_{\beta_{t+1}} \left(\underbrace{z^t - \frac{\alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)}{(1-\lambda\alpha_t)\beta_t} x^i}_{z^{t+1}} \right)$$

$$\beta_{t+1} = (1-\lambda\alpha_t)\beta_t, \qquad z^{t+1} = z^t - \frac{\alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)}{(1-\lambda\alpha_t)\beta_t} x^i$$

Lazy SGD updates for Sparse Data

SGD step

$$w^{t+1} = (1-\lambda\alpha_t)w^t - \alpha_t \ell'(\langle w^t, x^i\rangle, y^i)x^i$$

EXE: re-write the iterates using $w^t = \beta_t z^t$ where $\beta_t \in \mathbb{R}$, $z^t \in \mathbb{R}^d$

Can you update β_t and z^t so that each iteration is $O(s)$?

$$\begin{aligned}\beta_{t+1} z^{t+1} &= (1-\lambda\alpha_t)\beta_t z^t - \alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)x^i \\ &= \underbrace{(1-\lambda\alpha_t)\beta_t}_{\beta_{t+1}} \underbrace{\left(z^t - \frac{\alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)}{(1-\lambda\alpha_t)\beta_t} x^i\right)}_{z^{t+1}}\end{aligned}$$

O(1) scaling +
O(s) sparse add =
O(s) update

$$\beta_{t+1} = (1-\lambda\alpha_t)\beta_t, \qquad z^{t+1} = z^t - \frac{\alpha_t \ell'(\beta_t\langle z^t, x^i\rangle, y^i)}{(1-\lambda\alpha_t)\beta_t} x^i$$

Part IV: Momentum and gradient descent

Back to Gradient Descent

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

Baseline method: Gradient Descent (GD)

$$w^{t+1} = w^t - \gamma \nabla f(w^t)$$

Step size/
Learning rate

GD motivated through local rate of change

Local rate of change

$$\Delta(d) \;:=\; \lim_{s\to 0^+} \frac{f(x+ds)-f(x)}{s}$$

GD motivated through local rate of change

Local rate of change

$$\Delta(d) \; := \; \lim_{s \to 0^+} \frac{f(x+ds) - f(x)}{s}$$

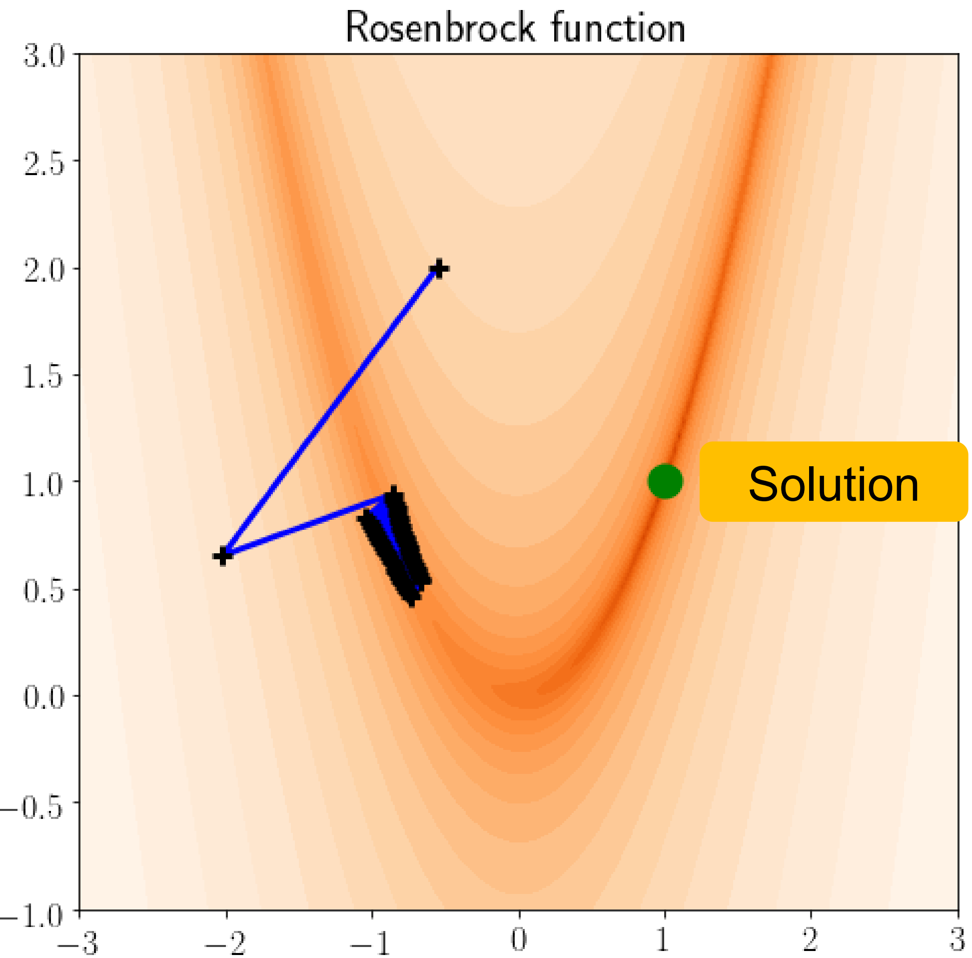
Max local rate

$$\frac{\nabla f(w^t)}{\|\nabla f(w^t)\|} \; := \; \max_{w \in \mathbb{R}^d} \Delta(d)$$

$$\text{subject to} \quad \|d\| \; = \; 1$$

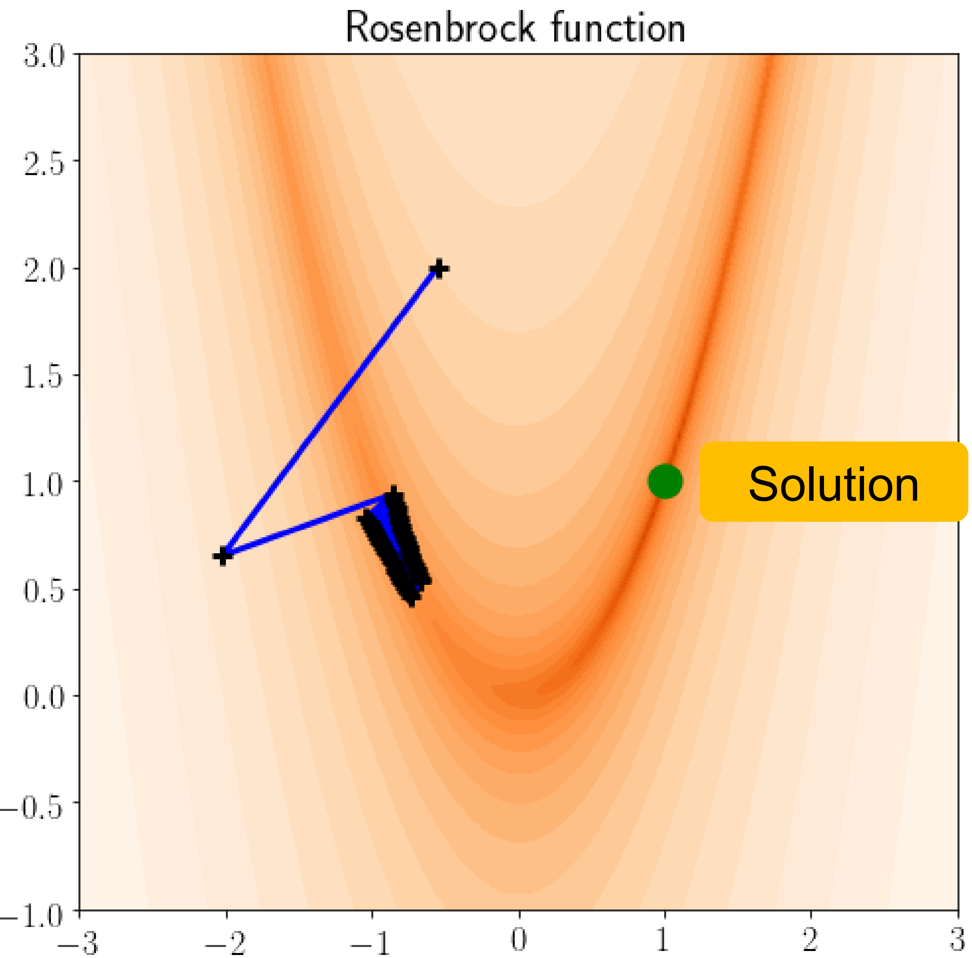
GD is the “steepest descent”

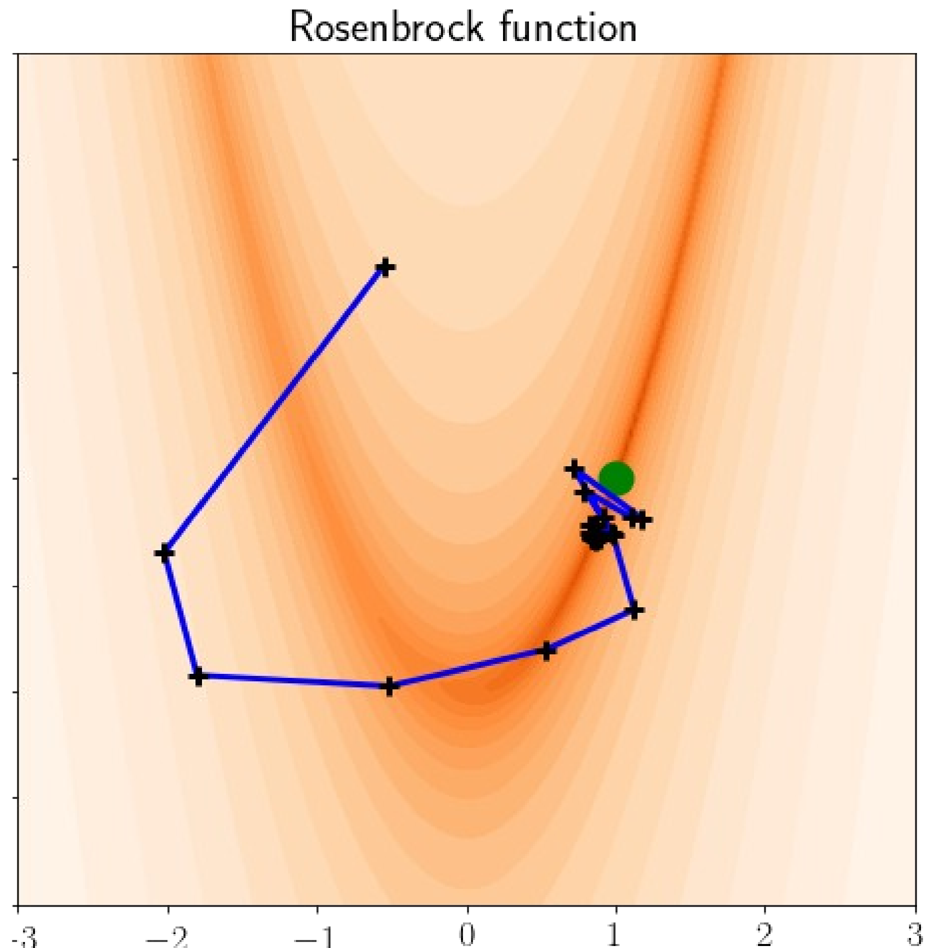
Local motivation not good for global



Get's stuck in "flat" valleys

Local motivation not good for global





Get’s stuck in “flat” valleys

Give momentum to keep going

Adding Momentum to GD

Additional momentum parameter ≈ 0.99

Heavey Ball Method:

$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1})$$

Adds “Inertia” to update, like friction for a heavy ball

Equivalent Momentum formulation

Heavey Ball Method:

$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \beta \underbrace{(w^t - w^{t-1})}_{\text{Adds “Inertia” to update}}$$

Equivalent Momentum formulation

Heavey Ball Method:

$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \beta\underbrace{(w^t - w^{t-1})}_{\text{Adds "Inertia" to update}}$$

Adds "Momentum" to update

GD with momentum (GDm):

$$m^t = \beta\, m^{t-1} + \nabla f(w^t)$$

$$w^{t+1} = w^t - \gamma\, m^t$$

Equivalent Momentum formulation

GD with momentum:

$$m^t = \beta\, m^{t-1} + \nabla f(w^t)$$

$$w^{t+1} = w^t - \gamma\, m^t$$

Equivalent Momentum formulation

GD with momentum:

$$m^t = \beta\, m^{t-1} + \nabla f(w^t)$$
$$w^{t+1} = w^t - \gamma\, m^t$$

$$\begin{aligned}
w^{t+1} &= w^t - \gamma\, m^t \\
&= w^t - \gamma\, (\beta m^{t-1} + \nabla f(w^t)) \\
&= w^t - \gamma\, \nabla f(w^t) - \gamma\beta\, m^{t-1} \\
&= w^t - \gamma\, \nabla f(w^t) + \frac{\gamma\beta}{\gamma}\, (w^t - w^{t-1})
\end{aligned}$$

Equivalent Momentum formulation

GD with momentum:

$$m^t = \beta\, m^{t-1} + \nabla f(w^t)$$
$$w^{t+1} = w^t - \gamma\, m^t$$

$$\begin{aligned}
w^{t+1} &= w^t - \gamma\, m^t \\
&= w^t - \gamma\left(\beta m^{t-1} + \nabla f(w^t)\right) \\
&= w^t - \gamma\, \nabla f(w^t) - \gamma\beta\, m^{t-1} \\
&= w^t - \gamma\, \nabla f(w^t) + \frac{\gamma\beta}{\gamma}\left(w^t - w^{t-1}\right)
\end{aligned}$$

Equivalent Momentum formulation

GD with momentum:

$$m^t = \beta\, m^{t-1} + \nabla f(w^t)$$
$$w^{t+1} = w^t - \gamma\, m^t$$

$$m^{t-1} = -\frac{1}{\gamma}(w^t - w^{t-1})$$

$$\begin{aligned}
w^{t+1} &= w^t - \gamma\, m^t \\
&= w^t - \gamma\,(\beta m^{t-1} + \nabla f(w^t)) \\
&= w^t - \gamma\,\nabla f(w^t) - \gamma\beta\, m^{t-1} \\
&= w^t - \gamma\,\nabla f(w^t) + \tfrac{\gamma\beta}{\gamma}\,(w^t - w^{t-1})
\end{aligned}$$

Equivalent Momentum formulation

GD with momentum:

$$m^t = \beta\, m^{t-1} + \nabla f(w^t)$$
$$w^{t+1} = w^t - \gamma\, m^t$$

$$m^{t-1} = -\frac{1}{\gamma}(w^t - w^{t-1})$$

$$\begin{aligned} w^{t+1} &= w^t - \gamma\, m^t \\ &= w^t - \gamma\, \underbrace{(\beta m^{t-1} + \nabla f(w^t))}_{} \\ &= w^t - \gamma\, \nabla f(w^t) - \gamma\beta\, m^{t-1} \\ &= w^t - \gamma\, \nabla f(w^t) + \frac{\gamma\beta}{\gamma}\, \underbrace{(w^t - w^{t-1})}_{} \end{aligned}$$

$$w^{t+1} = w^t - \gamma\, \nabla f(w^t) + \beta(w^t - w^{t-1})$$

Equivalent Momentum formulation

GD with momentum:

$$m^t = \beta\, m^{t-1} + \nabla f(w^t)$$
$$w^{t+1} = w^t - \gamma\, m^t$$

$$\begin{aligned} w^{t+1} &= w^t - \gamma\, m^t \\ &= w^t - \gamma\, (\beta m^{t-1} + \nabla f(w^t)) \\ &= w^t - \gamma\, \nabla f(w^t) - \gamma\beta\, m^{t-1} \\ &= w^t - \gamma\, \nabla f(w^t) + \tfrac{\gamma\beta}{\gamma}\, (w^t - w^{t-1}) \end{aligned}$$

$$m^{t-1} = -\frac{1}{\gamma}(w^t - w^{t-1})$$

Heavey Ball Method:

$$w^{t+1} = w^t - \gamma\, \nabla f(w^t) + \beta(w^t - w^{t-1})$$

Equivalent Iterate Averaging formulation

Heavey Ball Method:

$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \beta \underbrace{(w^t - w^{t-1})}_{\text{Adds “Inertia” to update}}$$

Equivalent Iterate Averaging formulation

Heavey Ball Method:

$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \underbrace{\beta(w^t - w^{t-1})}_{\text{Adds “Inertia” to update}}$$

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(w^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Equivalent Iterate Averaging formulation

Heavey Ball Method:

$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \underbrace{\beta(w^t - w^{t-1})}_{\text{Adds "Inertia" to update}}$$

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(w^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Additional sequence of variables

New parameters

Averaging of variables

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Define: $\gamma = \frac{\eta}{\alpha+1}$ and $\beta = \frac{\alpha}{\alpha+1}$

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Define: $\gamma = \frac{\eta}{\alpha+1}$ and $\beta = \frac{\alpha}{\alpha+1}$

$$w^{t+1} = \beta w^t + \frac{1}{\alpha+1} z^t$$

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Define: $\gamma = \dfrac{\eta}{\alpha+1}$ and $\beta = \dfrac{\alpha}{\alpha+1}$

$$\begin{aligned} w^{t+1} &= \beta w^t + \frac{1}{\alpha+1} z^t \\ &= \beta w^t + \frac{1}{\alpha+1}\left(z^{t-1} - \eta \nabla f(w^t)\right) \end{aligned}$$

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha + 1} w^t + \frac{1}{\alpha + 1} z^t$$

Define: $\gamma = \frac{\eta}{\alpha + 1}$ and $\beta = \frac{\alpha}{\alpha + 1}$

$t \leftarrow t - 1$

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

$$\begin{aligned} w^{t+1} &= \beta w^t + \frac{1}{\alpha + 1} z^t \\ &= \beta w^t + \frac{1}{\alpha + 1} \left(z^{t-1} - \eta \nabla f(w^t) \right) \end{aligned}$$

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Define: $\gamma = \frac{\eta}{\alpha+1}$ and $\beta = \frac{\alpha}{\alpha+1}$

$t \leftarrow t-1$

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

$$w^{t+1} = \beta w^t + \frac{1}{\alpha+1} z^t$$

$$= \beta w^t + \frac{1}{\alpha+1}\left(z^{t-1} - \eta \nabla f(w^t)\right)$$

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Define: $\gamma = \dfrac{\eta}{\alpha+1}$ and $\beta = \dfrac{\alpha}{\alpha+1}$

$t \leftarrow t-1$

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

$$\begin{aligned} w^{t+1} &= \beta w^t + \frac{1}{\alpha+1} z^t \\ &= \beta w^t + \frac{1}{\alpha+1} \left(z^{t-1} - \eta \nabla f(w^t)\right) \\ &= \beta w^t + \frac{1}{\alpha+1} \left((\alpha+1)w^t - \alpha w^{t-1} - \eta \nabla f(w^t)\right) \end{aligned}$$

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0,\ 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Define: $\gamma = \dfrac{\eta}{\alpha+1}$ and $\beta = \dfrac{\alpha}{\alpha+1}$

$t \leftarrow t-1$

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

$$\begin{aligned} w^{t+1} &= \beta w^t + \frac{1}{\alpha+1} z^t \\ &= \beta w^t + \frac{1}{\alpha+1}\left(z^{t-1} - \eta \nabla f(w^t)\right) \\ &= \beta w^t + \frac{1}{\alpha+1}\left((\alpha+1)w^t - \alpha w^{t-1} - \eta \nabla f(w^t)\right) \\ &= w^t - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1}) \end{aligned}$$

Equivalent Iterate Averaging formulation

Iterate Averaging: Let $\eta > 0, \alpha \in [0, 1]$

$$z^t = z^{t-1} - \eta \nabla f(x^t)$$

$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Define: $\gamma = \dfrac{\eta}{\alpha+1}$ and $\beta = \dfrac{\alpha}{\alpha+1}$

$t \leftarrow t-1$

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

$$\begin{aligned} w^{t+1} &= \beta w^t + \frac{1}{\alpha+1} z^t \\ &= \beta w^t + \frac{1}{\alpha+1}\left(z^{t-1} - \eta \nabla f(w^t)\right) \\ &= \beta w^t + \frac{1}{\alpha+1}\left((\alpha+1)w^t - \alpha w^{t-1} - \eta \nabla f(w^t)\right) \end{aligned}$$

Heavey Ball Method:

$$= w^t - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1})$$

Part IV.2: Convergence of Momentum with gradient descent

Convergence of Gradient Descent

Theorem Let f be μ–strongly convex and L–smooth, that is

stepsize

$$\mu I \quad \preceq \quad \nabla^2 f(w) \quad \preceq \quad LI, \quad \forall w \in \mathbb{R}^d$$

If $\gamma = \dfrac{2}{L+\mu}$ then Gradient Descent converges

$$\|w^t - w^*\| \; \leq \; \left(\frac{\kappa - 1}{\kappa + 1}\right)^t \|w^0 - w^*\|$$

$\kappa := L/\mu \geq 1$

Convergence of Gradient Descent

Theorem Let f be μ–strongly convex and L–smooth, that is

stepsize

$$\mu I \quad \preceq \quad \nabla^2 f(w) \quad \preceq \quad LI, \quad \forall w \in \mathbb{R}^d$$

If $\gamma = \dfrac{2}{L+\mu}$ then Gradient Descent converges

$$\Rightarrow \quad \|w^t - w^*\| \; \leq \; \left(\frac{\kappa - 1}{\kappa + 1}\right)^t \|w^0 - w^*\|$$

$\kappa := L/\mu \geq 1$

Corollary $t \; \geq \; \dfrac{1}{\kappa + 1} \log\left(\dfrac{1}{\epsilon}\right) \quad \Rightarrow \quad \dfrac{\|w^t - w^*\|}{\|w^0 - w^*\|} \leq \epsilon$

Convergence of Gradient Descent with Momentum

Polyak 1964

Theorem Let $f \in C^2$ be μ–strongly convex and L–smooth, that is

stepsize

$$\mu I \quad \preceq \quad \nabla^2 f(w) \quad \preceq \quad LI, \quad \forall w \in \mathbb{R}^d$$

If $\gamma = \dfrac{4}{(\sqrt{L}+\sqrt{\mu})^2}$ and $\beta = \dfrac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ then SGDm converges

$$\|w^t - w^*\| \;\leq\; \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t \|w^0 - w^*\|$$

$\kappa := L/\mu \geq 1$

Convergence of Gradient Descent with Momentum

Polyak 1964

Theorem Let $f \in C^2$ be μ–strongly convex and L–smooth, that is

stepsize

$$\mu I \quad \preceq \quad \nabla^2 f(w) \quad \preceq \quad LI, \quad \forall w \in \mathbb{R}^d$$

If $\gamma = \dfrac{4}{(\sqrt{L}+\sqrt{\mu})^2}$ and $\beta = \dfrac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ then SGDm converges

$$\Rightarrow \quad \|w^t - w^*\| \; \leq \; \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t \|w^0 - w^*\|$$

$\kappa := L/\mu \geq 1$

Corollary $t \;\geq\; \dfrac{1}{\sqrt{\kappa}+1} \log\left(\dfrac{1}{\epsilon}\right) \quad \Rightarrow \quad \dfrac{\|w^t - w^*\|}{\|w^0 - w^*\|} \leq \epsilon$

Convergence of Gradient Descent with Momentum

Polyak 1964

Theorem Let $f \in C^2$ be μ–strongly convex and L–smooth, that is

stepsize

$$\mu I \preceq \nabla^2 f(w) \preceq LI, \quad \forall w \in \mathbb{R}^d$$

If $\gamma = \dfrac{4}{(\sqrt{L}+\sqrt{\mu})^2}$ and $\beta = \dfrac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ then SGDm converges

$$\Rightarrow \quad \|w^t - w^*\| \leq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^t \|w^0 - w^*\|$$

Optimal iteration complexity for this function class

$\kappa := L/\mu \geq 1$

Corollary $t \geq \dfrac{1}{\sqrt{\kappa}+1} \log\left(\dfrac{1}{\epsilon}\right) \quad \Rightarrow \quad \dfrac{\|w^t - w^*\|}{\|w^0 - w^*\|} \leq \epsilon$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

$$\int_{s=0}^{1} \nabla^2 f(\underbrace{w^s}_{w^s := w^* + s(w^t - w^*)}) ds (w^t - w^*) = \nabla f(w^t) - \nabla f(w^*) = \nabla f(w^t)$$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

$$\int_{s=0}^{1} \nabla^2 f(\underbrace{w^s}_{w^s := w^* + s(w^t - w^*)})ds(w^t - w^*) = \nabla f(w^t) - \nabla f(w^*) = \nabla f(w^t)$$

$$\begin{aligned}
w^{t+1} - w^* &= w^t - w^* - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1}) \\
&= \left(I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)\right)(w^t - w^*) + \beta(w^t - w^{t-1}) \\
&= \left((1+\beta)I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)\right)(w^t - w^*) - \beta(w^{t-1} - w^*)
\end{aligned}$$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

$$\int_{s=0}^{1} \nabla^2 f(\underbrace{w^s}_{w^s := w^* + s(w^t - w^*)})ds(w^t - w^*) = \nabla f(w^t) - \nabla f(w^*) = \nabla f(w^t)$$

$+w^* - w^*$

$$\begin{aligned}
w^{t+1} - w^* &= w^t - w^* - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1}) \\
&= \left(I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)\right)(w^t - w^*) + \beta(w^t - w^{t-1}) \\
&= \left((1+\beta)I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)\right)(w^t - w^*) - \beta(w^{t-1} - w^*)
\end{aligned}$$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

$$\int_{s=0}^{1} \nabla^2 f(\underbrace{w^s}_{w^s := w^* + s(w^t - w^*)}) ds (w^t - w^*) = \nabla f(w^t) - \nabla f(w^*) = \nabla f(w^t)$$

$$\begin{aligned}
w^{t+1} - w^* &= w^t - w^* - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1}) \qquad +w^* - w^* \\
&= \left(I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)\right)(w^t - w^*) + \beta(w^t - w^{t-1}) \\
&= \left(\underbrace{(1+\beta)I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)}_{=: A_\gamma}\right)(w^t - w^*) - \beta(w^{t-1} - w^*)
\end{aligned}$$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

$$\int_{s=0}^{1} \nabla^2 f(\underbrace{w^s}_{w^s := w^* + s(w^t - w^*)}) ds (w^t - w^*) = \nabla f(w^t) - \nabla f(w^*) = \nabla f(w^t)$$

$+w^* - w^*$

$$\begin{aligned}
w^{t+1} - w^* &= w^t - w^* - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1}) \\
&= \left(I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)\right)(w^t - w^*) + \beta(w^t - w^{t-1}) \\
&= \left(\underbrace{(1+\beta)I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)}_{=: A_\gamma}\right)(w^t - w^*) - \beta(w^{t-1} - w^*) \\
&= A_\gamma (w^t - w^*) - \beta(w^{t-1} - w^*)
\end{aligned}$$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

$$\int_{s=0}^{1} \nabla^2 f(\underbrace{w^s}_{w^s := w^* + s(w^t - w^*)})ds(w^t - w^*) = \nabla f(w^t) - \nabla f(w^*) = \nabla f(w^t)$$

$$\begin{aligned}
w^{t+1} - w^* &= w^t - w^* - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1}) \qquad +w^* - w^* \\
&= \left(I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)\right)(w^t - w^*) + \beta(w^t - w^{t-1}) \\
&= \left(\underbrace{(1+\beta)I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)}_{=:\, A_\gamma}\right)(w^t - w^*) - \beta(w^{t-1} - w^*) \\
&= A_\gamma(w^t - w^*) - \beta(w^{t-1} - w^*)
\end{aligned}$$

Depends on two times steps

Proof: Convergence of Heavy Ball

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} \in \mathbb{R}^{2d}$$

Proof: Convergence of Heavy Ball

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} \in \mathbb{R}^{2d}$$

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_\gamma(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix}$$

Proof: Convergence of Heavy Ball

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} \in \mathbb{R}^{2d}$$

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_\gamma (w^t - w^*) - \beta (w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix}$$

$$= \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix}$$

Proof: Convergence of Heavy Ball

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} \in \mathbb{R}^{2d}$$

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_\gamma(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix}$$

$$= \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix}$$

$$= \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} z^t$$

Proof: Convergence of Heavy Ball

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} \in \mathbb{R}^{2d}$$

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_\gamma(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix}$$

$$= \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix}$$

$$= \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} z^t$$

Simple recurrence!

Proof: Convergence of Heavy Ball

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} \in \mathbb{R}^{2d}$$

$$z^{t+1} = \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_\gamma(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix}$$

$$= \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix}$$

$$= \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} z^t$$

Simple recurrence!

$$\|z^{t+1}\| \leq \left\| \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \right\| \|z^t\|$$

Proof: Convergence of Heavy Ball

$$\|z^{t+1}\| \quad \leq \quad \left\| \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \right\| \; \|z^t\|$$

Proof: Convergence of Heavy Ball

$$\|z^{t+1}\| \quad \leq \quad \left\| \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \right\| \; \|z^t\|$$

EXE on Eigenvalues:

If $\gamma = \dfrac{4}{(\sqrt{L}+\sqrt{\mu})^2}$ and $\beta = \dfrac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ then

$$\left\| \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \right\| \quad = \quad \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$$

Proof: Convergence of Heavy Ball

$$\|z^{t+1}\| \quad \leq \quad \left\| \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \right\| \; \|z^t\|$$

$$(1+\beta)I - \gamma \int_{s=0}^{1} \nabla^2 f(w^s)$$

EXE on Eigenvalues:

If $\gamma = \dfrac{4}{(\sqrt{L}+\sqrt{\mu})^2}$ and $\beta = \dfrac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ then

$$\left\| \begin{bmatrix} A_\gamma & -I\beta \\ I & 0 \end{bmatrix} \right\| \quad = \quad \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$$

Part V: Momentum with SGD

Adding Momentum to SGD

Rumelhart, Hinton, Geoffrey, Ronald, 1986, Nature

Stochastic Heavey Ball Method:

$$w^{t+1} = w^t - \gamma \nabla f_{j_t}(w^t) + \beta(w^t - w^{t-1})$$

SGD with momentum:

$$m^t = \beta m^{t-1} + \nabla f_{j_t}(w^t)$$
$$w^{t+1} = w^t - \gamma m^t$$

Iterate Averaging:

$$z^t = z^{t-1} - \eta \nabla f_{j_t}(x^t)$$
$$w^{t+1} = \frac{\alpha}{\alpha+1} w^t + \frac{1}{\alpha+1} z^t$$

Sampled i.i.d

$j_t \in \{1, \ldots, n\}$

$\mathbb{P}[j = j_t] = 1/n$

SGDm and Averaging

$$\begin{aligned} m^t &= \beta\, m^{t-1} + \nabla f_{j_t}(w^t) \\ &= \beta\, m^{t-2} + \nabla f_{j_t}(w^t) + \beta \nabla f_{j_{t-1}}(w^{t-1}) \\ &= \textstyle\sum_{i=1}^t \beta^i \nabla f_{j_{t-i}}(w^{t-i}) \end{aligned}$$

http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html

SGDm and Averaging

$$
\begin{aligned}
m^t &= \beta\, m^{t-1} + \nabla f_{j_t}(w^t) \\
&= \beta\, m^{t-2} + \nabla f_{j_t}(w^t) + \beta \nabla f_{j_{t-1}}(w^{t-1}) \\
&= \textstyle\sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i})
\end{aligned}
$$

$m^0 = 0$

http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html

SGDm and Averaging

$$\begin{aligned} m^t &= \beta\, m^{t-1} + \nabla f_{j_t}(w^t) \\ &= \beta\, m^{t-2} + \nabla f_{j_t}(w^t) + \beta \nabla f_{j_{t-1}}(w^{t-1}) \\ &= \textstyle\sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i}) \end{aligned}$$

$m^0 = 0$

Momentum as exponentiated average:

$$w^{t+1} = w^t - \gamma \sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i})$$

http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html

SGDm and Averaging

$$\begin{aligned} m^t &= \beta\, m^{t-1} + \nabla f_{j_t}(w^t) \\ &= \beta\, m^{t-2} + \nabla f_{j_t}(w^t) + \beta \nabla f_{j_{t-1}}(w^{t-1}) \\ &= \textstyle\sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i}) \end{aligned}$$

$m^0 = 0$

Momentum as exponentiated average:

$$w^{t+1} = w^t - \gamma \sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i})$$

Acts like an approximate variance reduction since

http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html

SGDm and Averaging

$$
\begin{aligned}
m^t &= \beta\, m^{t-1} + \nabla f_{j_t}(w^t) \\
&= \beta\, m^{t-2} + \nabla f_{j_t}(w^t) + \beta \nabla f_{j_{t-1}}(w^{t-1}) \\
&= \textstyle\sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i})
\end{aligned}
$$

$m^0 = 0$

Momentum as exponentiated average:

$$
w^{t+1} = w^t - \gamma \sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i})
$$

Acts like an approximate variance reduction since

$$
\sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i}) \approx \sum_{i=1}^{n} \frac{1}{n} \nabla f_i(w^t)
$$

http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html

SGDm and Averaging

$$\begin{aligned} m^t &= \beta\, m^{t-1} + \nabla f_{j_t}(w^t) \\ &= \beta\, m^{t-2} + \nabla f_{j_t}(w^t) + \beta \nabla f_{j_{t-1}}(w^{t-1}) \\ &= \textstyle\sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i}) \end{aligned}$$

$m^0 = 0$

Momentum as exponentiated average:

$$w^{t+1} = w^t - \gamma \sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i})$$

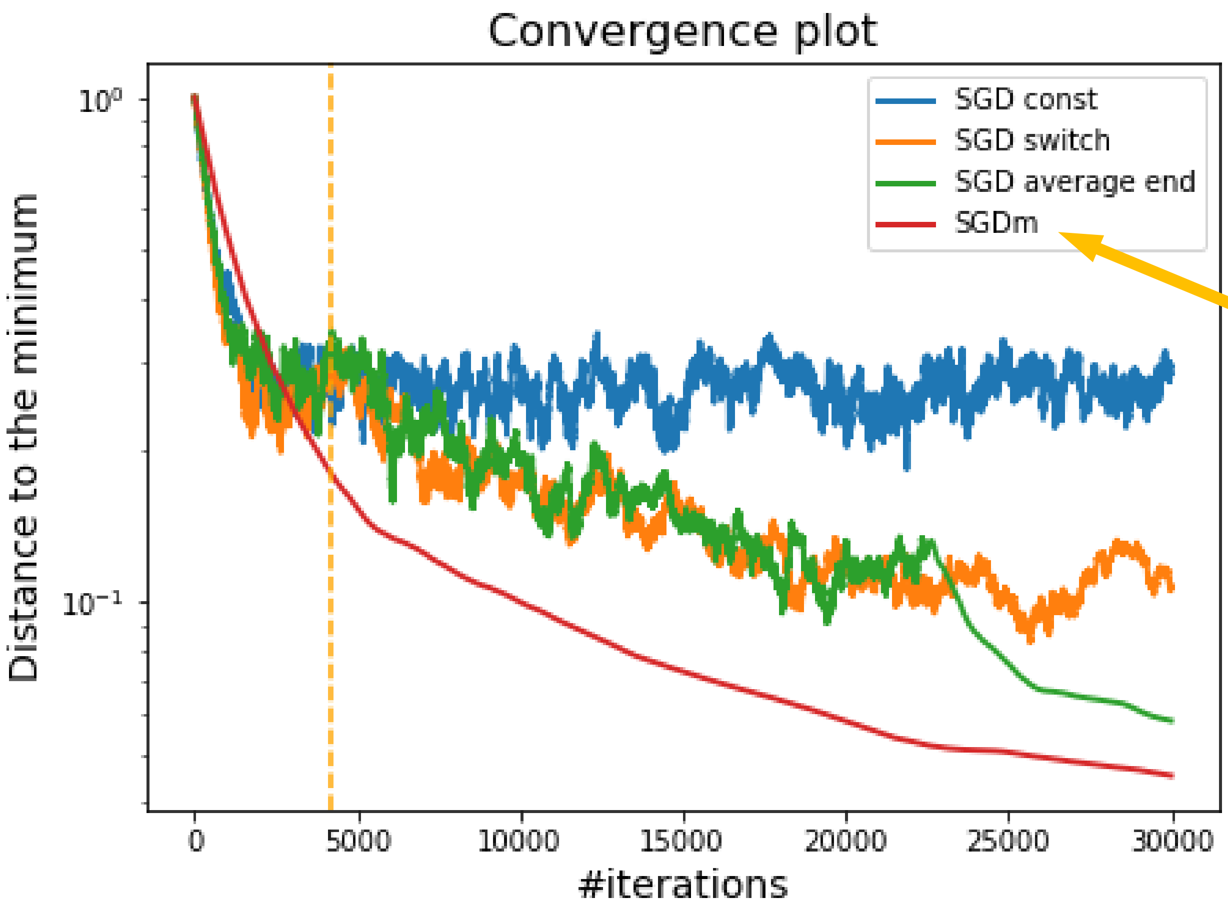
Acts like an approximate variance reduction since

This is why momentum works well with SGD

$$\sum_{i=1}^{t} \beta^i \nabla f_{j_{t-i}}(w^{t-i}) \approx \sum_{i=1}^{n} \frac{1}{n} \nabla f_i(w^t)$$

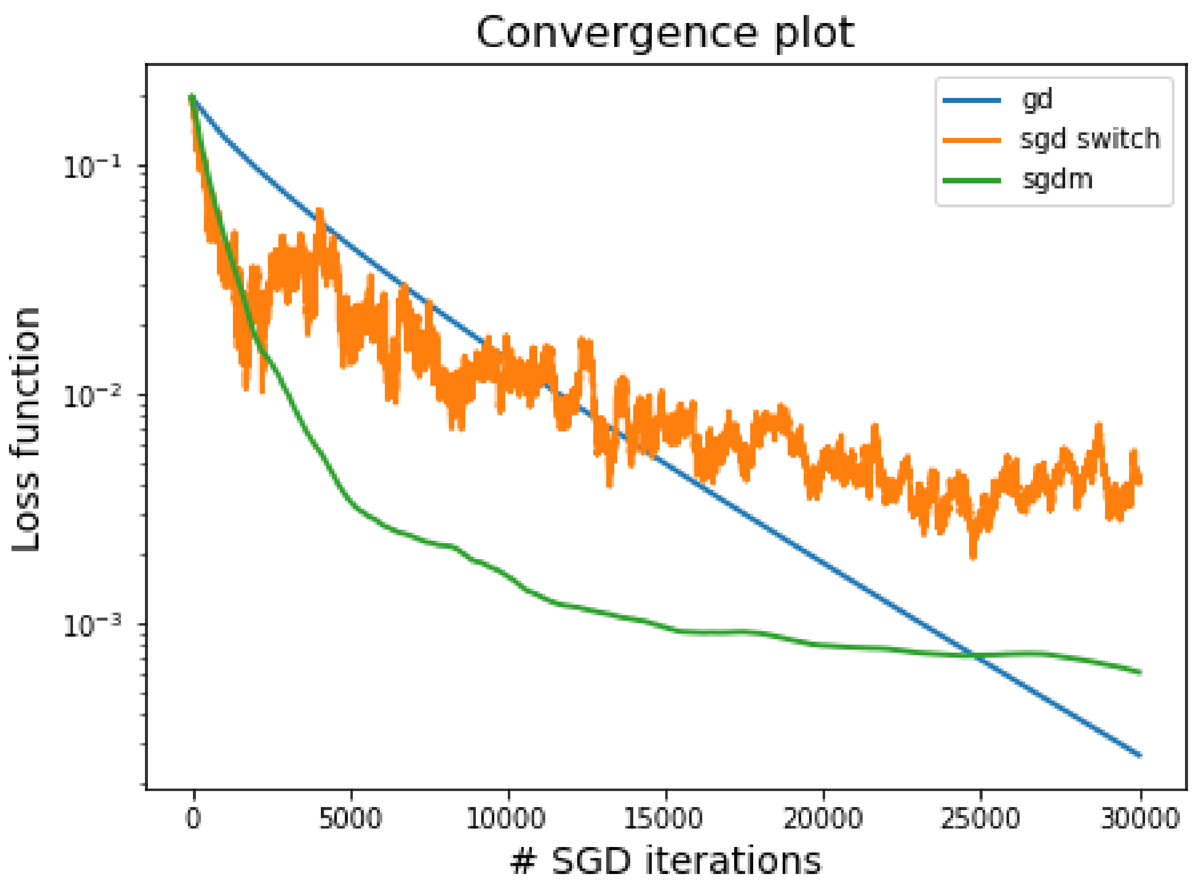
http://fa.bianp.net/teaching/2018/COMP-652/stochastic_gradient.html

Stochastic Gradient Descent with momentum



SGDm = SGD with momentum

Stochastic Gradient Descent with momentum vs GD



Can we prove momentum always works?

Difficult: Recent 2019 results only

Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?

Not clear, recently same rate as SGD + averaging

Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?

Not clear, recently same rate as SGD + averaging

Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?

Not clear, recently same rate as SGD + averaging

f is μ–strongly convex,

f_i is convex and L_i–smooth

$$t \geq O\left(\frac{1}{\epsilon}\right)$$

Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?

Not clear, recently same rate as SGD + averaging

f is μ–strongly convex,
f_i is convex and L_i–smooth

$$t \geq O\left(\frac{1}{\epsilon}\right)$$

f_i is convex and L_i–smooth

$$t \geq O\left(\frac{1}{\epsilon^2}\right)$$

Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?

Not clear, recently same rate as SGD + averaging

f is μ–strongly convex,
f_i is convex and L_i–smooth

$$t \geq O\left(\frac{1}{\epsilon}\right)$$

f_i is convex and L_i–smooth

$$t \geq O\left(\frac{1}{\epsilon^2}\right)$$

Sebbouth, Defazio, RMG, online soon, 2020

Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?

Not clear, recently same rate as SGD + averaging

f is μ–strongly convex,

f_i is convex and L_i–smooth

f_i is convex and L_i–smooth

Results use iterate averaging to crack the proof!

$$t \geq O\left(\frac{1}{\epsilon}\right)$$

$$t \geq O\left(\frac{1}{\epsilon^2}\right)$$

Sebbouth, Defazio, RMG, online soon, 2020

Part V: Test error and Validation

Generalization

We have been solving:

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

Generalization

We have been solving:

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

But we already know these labels

Generalization

We have been solving:

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

But we already know these labels

We want our model to correctly label unseen data. We want to generalize

Generalization

We have been solving:

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

But we already know these labels

We want our model to correctly label unseen data. We want to generalize

We would like to solve:

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

$$\min_{w \in \mathbf{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(h_w(x), y\right)\right]$$

Validation Error

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_T & x_{T+1} & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$y := \begin{bmatrix} y_1 & y_2 & \cdots & y_T & y_{T+1} & \cdots & y_n \end{bmatrix} \in \mathbb{R}^n$$

Validation Error

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_T & \vline & x_{T+1} & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$y := \begin{bmatrix} y_1 & y_2 & \cdots & y_T & \vline & y_{T+1} & \cdots & y_n \end{bmatrix} \in \mathbb{R}^{n}$$

Validation Error

Train set	Validation set

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_T & x_{T+1} & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$y := \begin{bmatrix} y_1 & y_2 & \cdots & y_T & y_{T+1} & \cdots & y_n \end{bmatrix} \in \mathbb{R}^{n}$$

Validation Error

Train set | Validation set

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_T & x_{T+1} & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$y := \begin{bmatrix} y_1 & y_2 & \cdots & y_T & y_{T+1} & \cdots & y_n \end{bmatrix} \in \mathbb{R}^{n}$$

Use to train

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

Validation Error

Train set

Validation set

$$X := \begin{bmatrix} x_1 & x_2 & \cdots & x_T \end{bmatrix} \; \Big| \; \begin{bmatrix} x_{T+1} & \cdots & x_n \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$y := \begin{bmatrix} y_1 & y_2 & \cdots & y_T \end{bmatrix} \; \Big| \; \begin{bmatrix} y_{T+1} & \cdots & y_n \end{bmatrix} \in \mathbb{R}^{n}$$

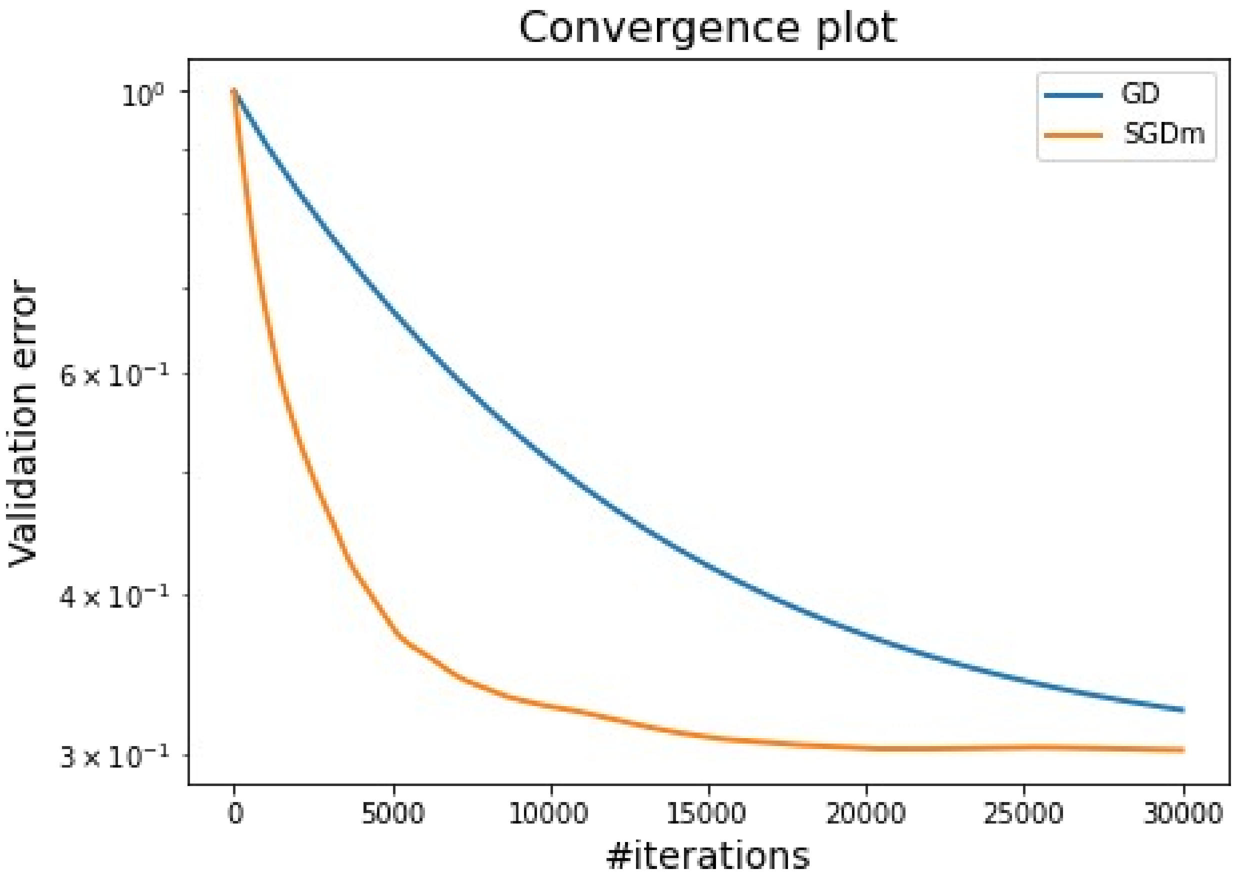
Use to train

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

Use to validate

$$\mathrm{loss}(w^t) = \frac{1}{n-T} \sum_{i=T+1}^{n} \ell\left(h_{w^t}(x^i), y^i\right) + \lambda R(w^t)$$

Stochastic Gradient Descent with momentum vs GD on validation set



This why SGD is popular in ML

More reason why ML likes SGD

We have been solving:

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

But we already know these labels

But we want to solve:

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

$$\min_{w \in \mathbf{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell\left(h_w(x), y\right)\right]$$

SGD can be applied to the statistical learning problem!

Why Machine Learners like SGD

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

$$\min_{w \in \mathbf{R}^d} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\ell\left(h_w(x), y\right)\right]$$

SGD for learning

Set $w^0 = 0,\ \alpha_t > 0$

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

sample $(x, y) \sim \mathcal{D}$

$w^{t+1} = w^t - \alpha_t \nabla \ell(h_{w^t}(x), y)$

Output $\overline{w}^T = \frac{1}{T}\sum_{t=1}^{T} w^t$