

Optimization for Machine Learning

Introduction into supervised learning, stochastic gradient descent analysis and tricks

Lecturer: Robert M. Gower



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

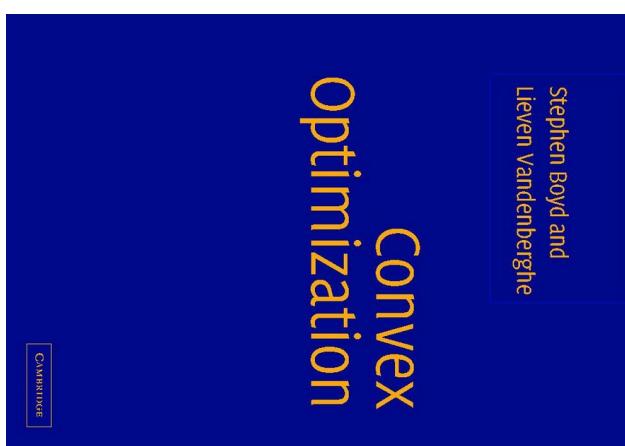
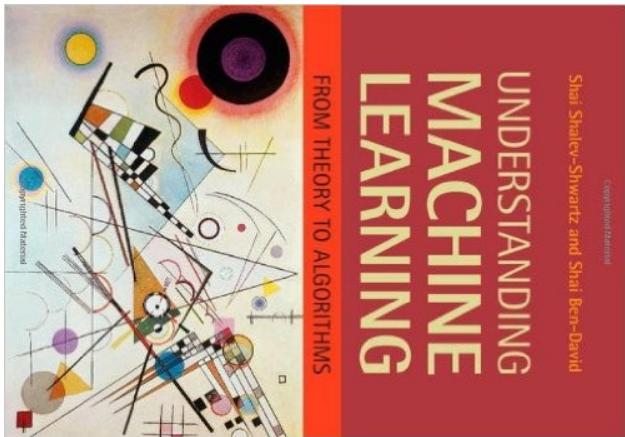
References classes today

Chapter 2

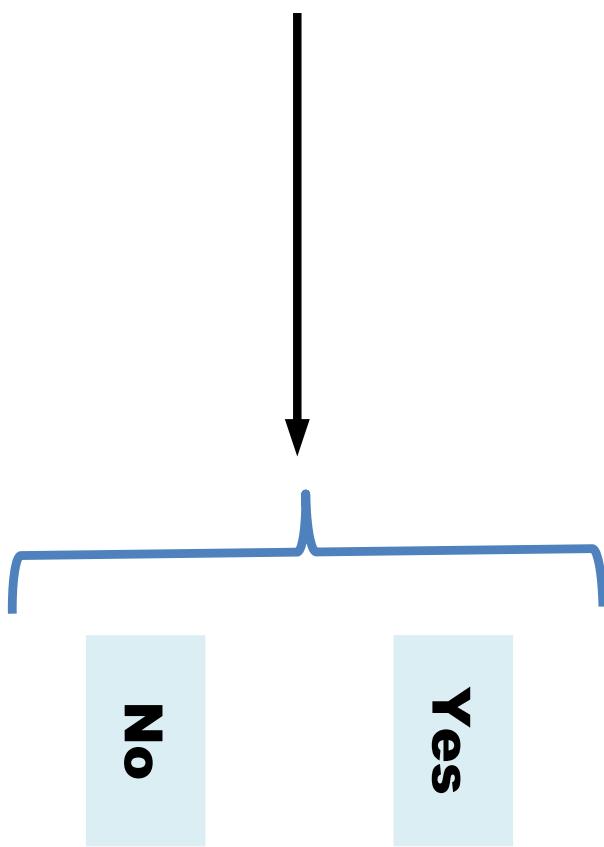
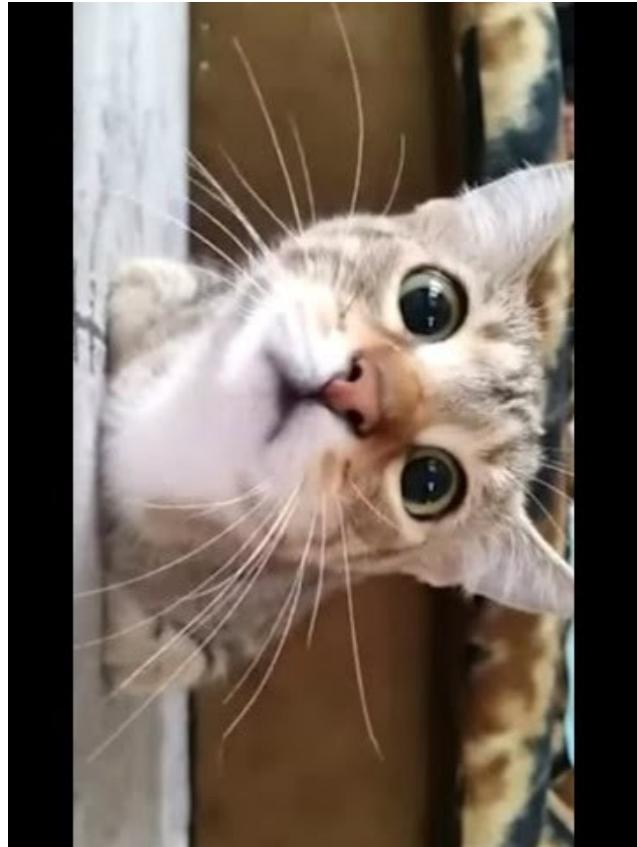
Pages 67 to 79

Understanding Machine
Learning: From Theory to
Algorithms

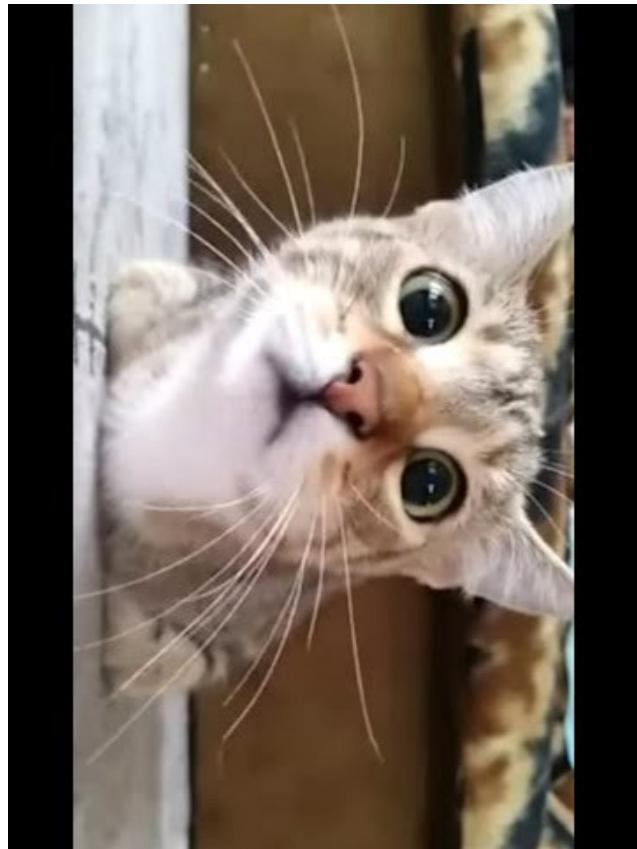
Convex Optimization,
Stephen Boyd



Is There a Cat in the Photo?



Is There a Cat in the Photo?



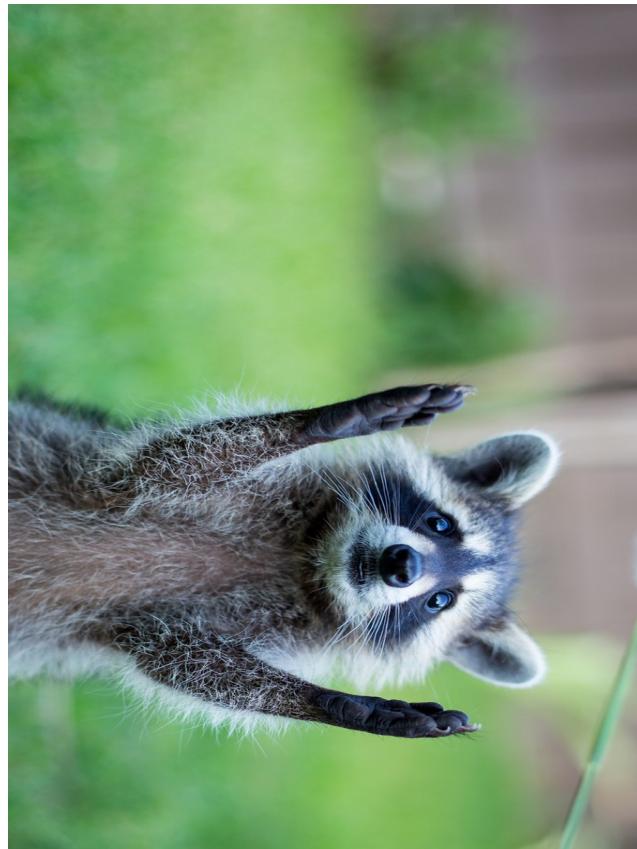
Yes

Is There a Cat in the Photo?



Yes

Is There a Cat in the Photo?



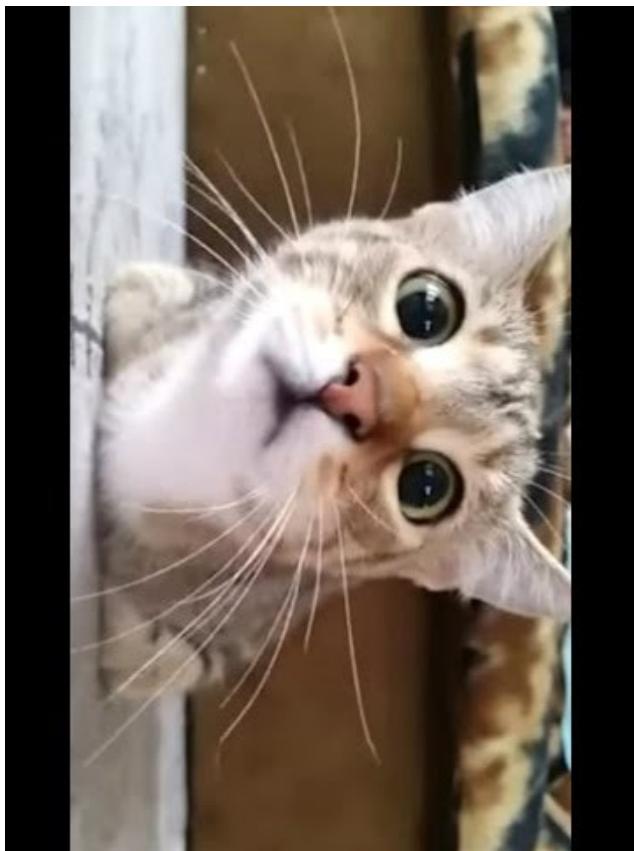
No

Is There a Cat in the Photo?



Yes

Is There a Cat in the Photo?



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

| | |
|----------------|--|
| $x^1 \{$ |  |
| $y^1 = 1$ | |
| $x^2 \{$ |  |
| $y^2 = 1$ | |
| $x^3 \{$ |  |
| $y^3 = -1$ | |
| $\dots x^n \{$ | |
| $y^n = 1$ |  |



Labeled Data: The training set

| | |
|----------------|--|
| $x^1 \{$ |  |
| $y^1 = 1$ | |
| $x^2 \{$ |  |
| $y^2 = 1$ | |
| $x^3 \{$ |  |
| $y^3 = -1$ | |
| $\dots x^n \{$ | |
| $y^n = 1$ |  |

$y = -1$ means no/false



Labeled Data: The training set

| | |
|----------------|--|
| $x^1 \{$ |  |
| $y^1 = 1$ | |
| $x^2 \{$ |  |
| $y^2 = 1$ | |
| $x^3 \{$ |  |
| $y^3 = -1$ | |
| $\dots x^n \{$ | |
| $y^n = 1$ | |



$y = -1$ means no/false

Training
Algorithm

Labeled Data: The training set

| | |
|----------------|--|
| $x^1 \{$ |  |
| $y^1 = 1$ | |
| $x^2 \{$ |  |
| $y^2 = 1$ | |
| $x^3 \{$ |  |
| $y^3 = -1$ | |
| $\dots x^n \{$ | |
| $y^n = 1$ | |



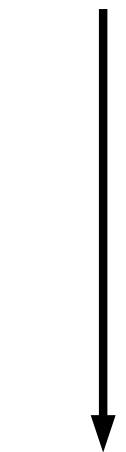
$y = -1$ means no/false



Training
Algorithm

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

Labeled Data: The training set



-1

Training
Algorithm

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



$y = -1$ means no/false

| | |
|----------------|--|
| $x^1 \{$ | A photograph of a white cat looking up. |
| $y^1 = 1$ | |
| $x^2 \{$ | A photograph of a cat with a clown mask. |
| $y^2 = 1$ | |
| $x^3 \{$ | A photograph of a raccoon. |
| $y^3 = -1$ | |
| $\dots x^n \{$ | |
| $y^n = 1$ | A photograph of a fluffy orange cat. |

Example: Linear Regression for Height

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

Male = 0
Female = 1

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

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

Example: Linear Regression for Height

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

Male = 0
Female = 1

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

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

Example Hypothesis: Linear Model

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

Example: Linear Regression for Height

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 |

Male = 0
Female = 1

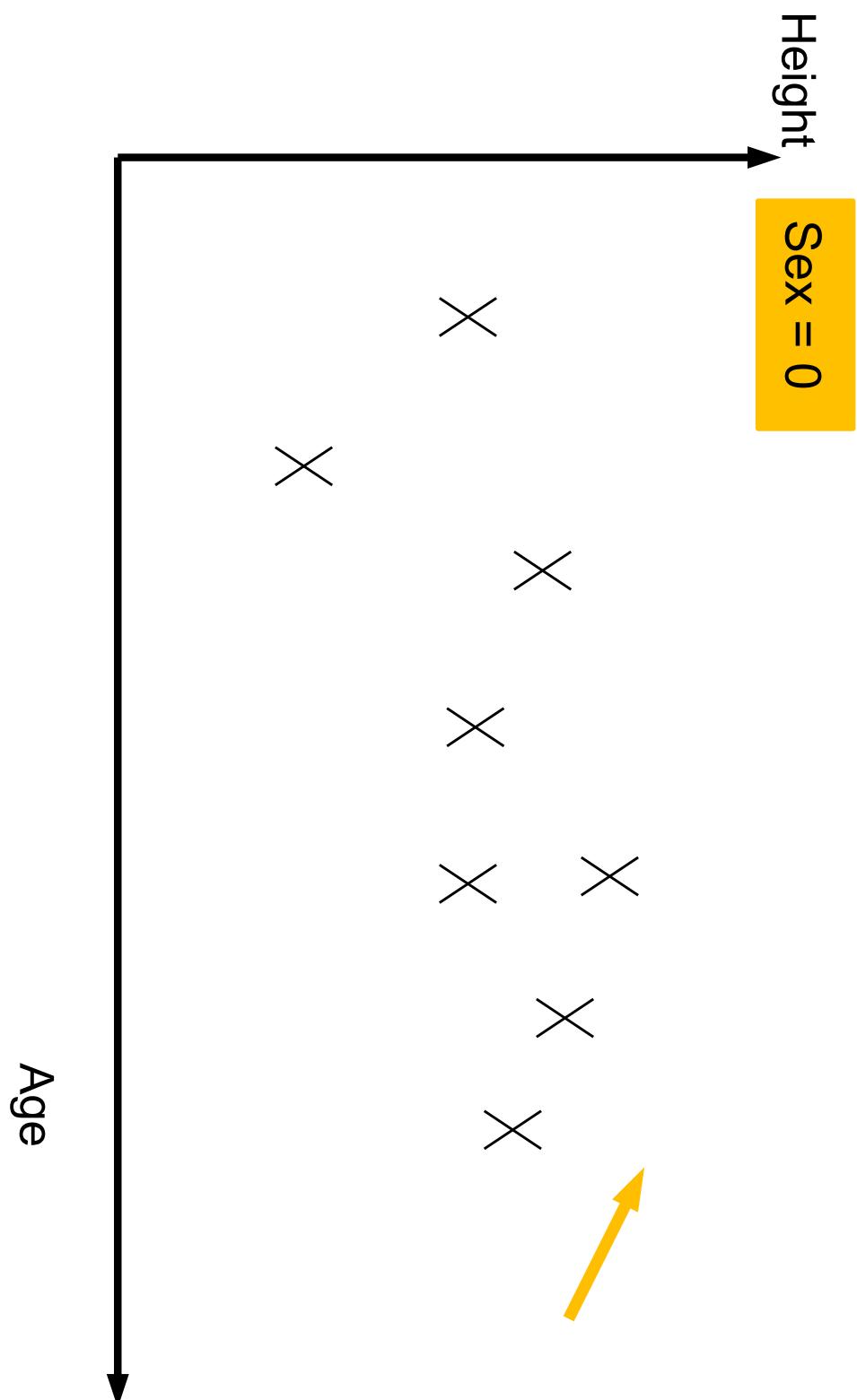
| | | |
|---------|----------|---------|
| x_1^n | { Sex | 1 |
| x_2^n | { Age | 70 |
| y^n | { Height | 1,52 cm |

Example Hypothesis: Linear Model

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

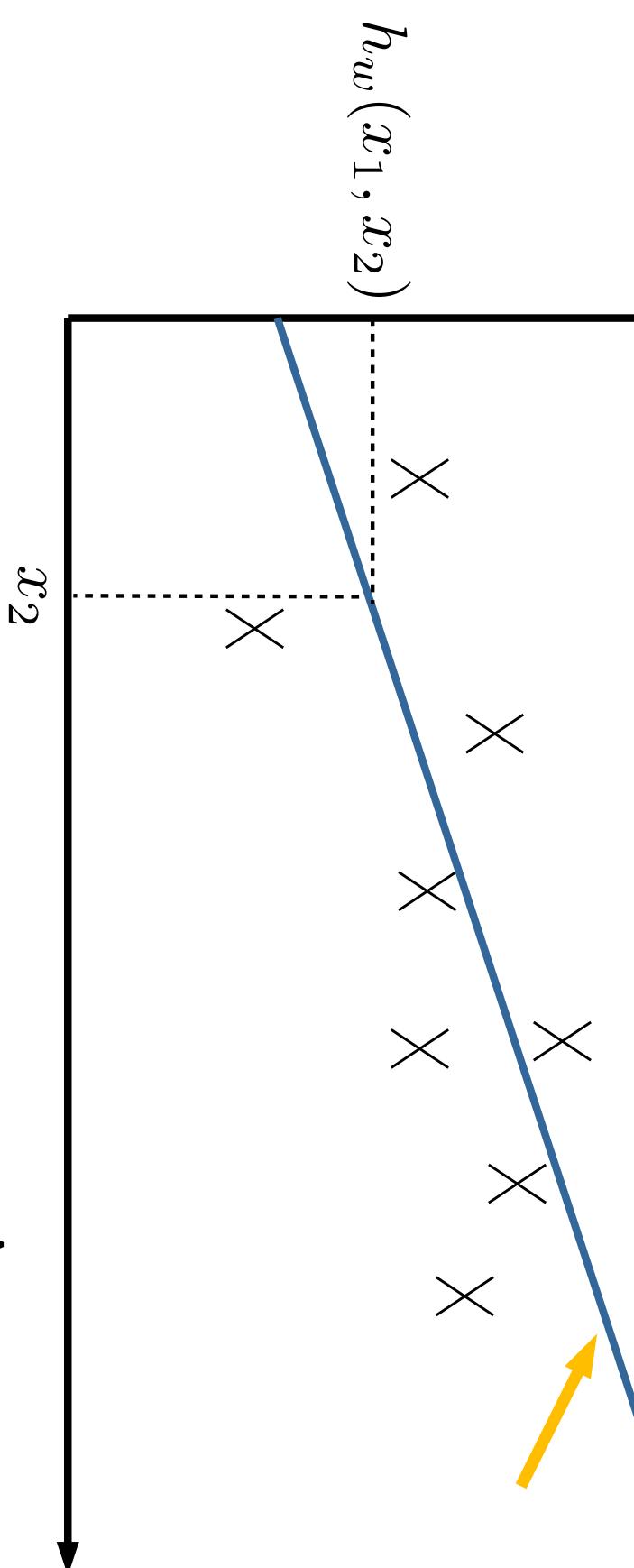
$$\min_{w \in \mathbf{R}^3} \frac{1}{n} \sum_{i=1}^n (h_w(x_1^i, x_2^i) - y^i)^2$$

Linear Regression for Height



Linear Regression for Height

Height
Sex = 0

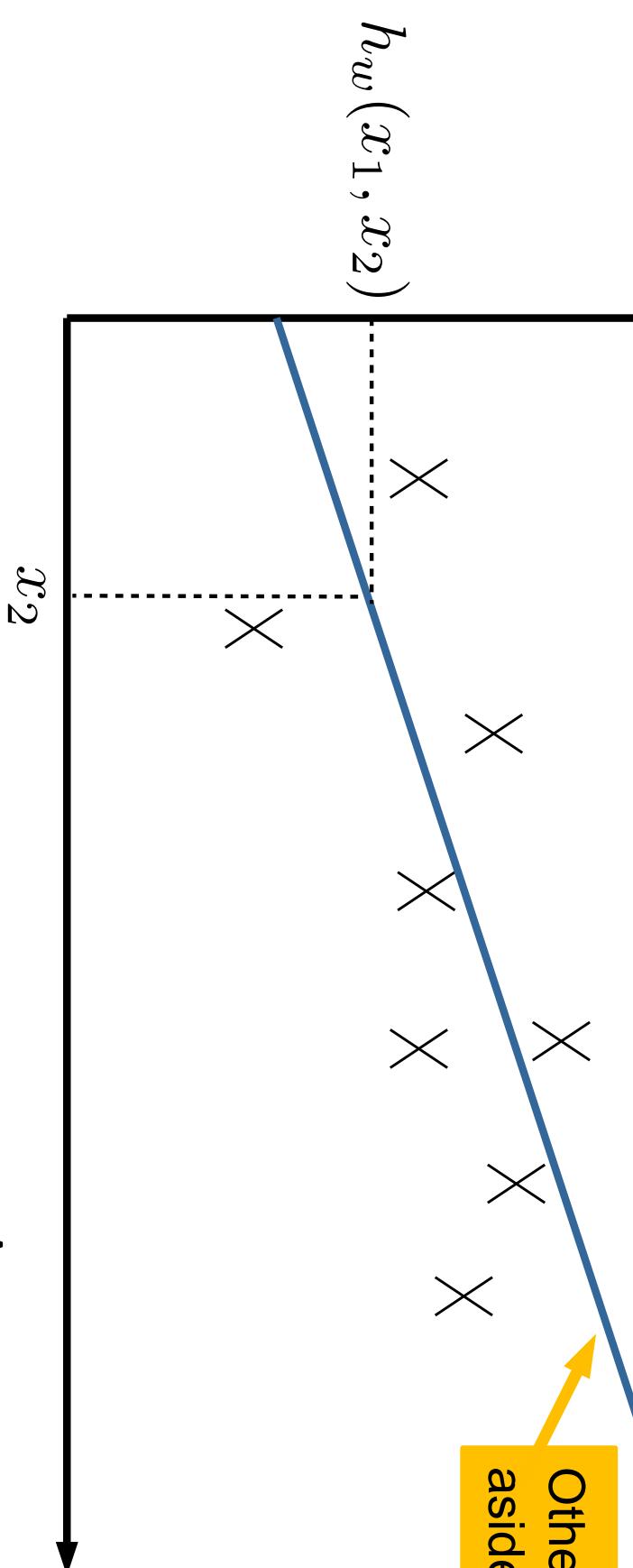


The Training
Algorithm

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

Linear Regression for Height

Height
Sex = 0



Other options
aside from linear?

The Training
Algorithm

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

Parametrizing the Hypothesis

Linear:

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

Polynomial:

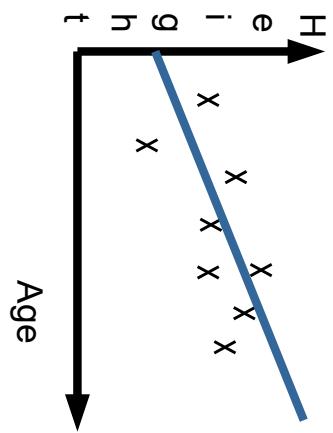
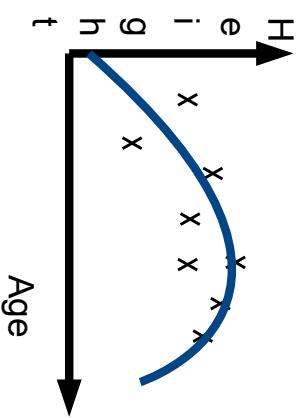
$$h_w(x) = \sum_{i,j=0}^d w_{ij} x_i x_j$$

exe :

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

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

Neural Net:



LOSS Functions

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

Why a
Squared
Loss?

LOSS Functions

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

Let $y_h := h_w(x)$

Why a
Squared
Loss?

Loss Functions

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

The Training Problem

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

LOSS Functions

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

Let $y_h := h_w(x)$

Loss Functions

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

Typically a
convex function

Why a
Squared
Loss?

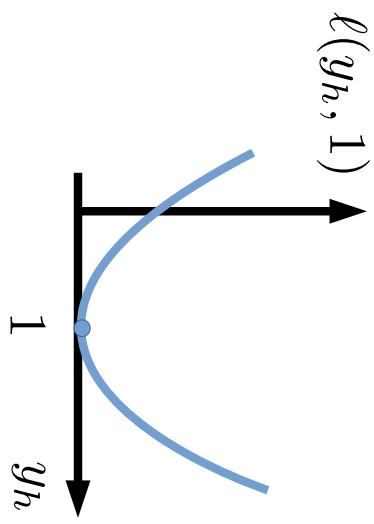
The Training Problem

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

Choosing the Loss Function

Let $y_h := h_w(x)$

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

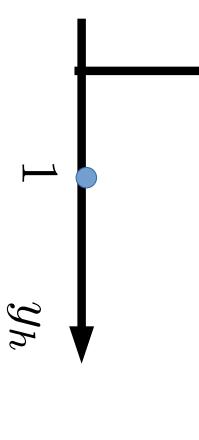


$\ell(y_h, 1)$

y_h

Binary Loss

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

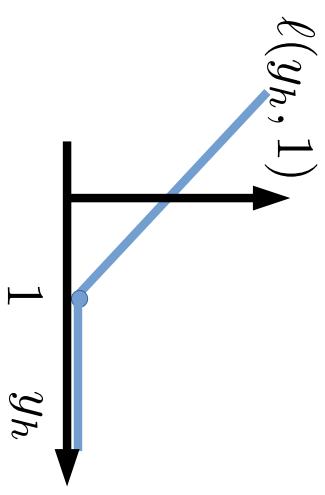


$\ell(y_h, 1)$

y_h

Hinge Loss

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

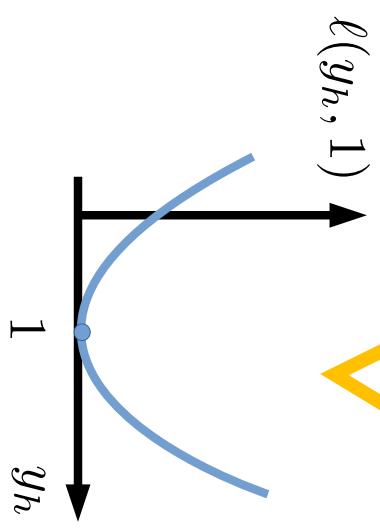


y_h

Choosing the Loss Function

Let $y_h := h_w(x)$

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

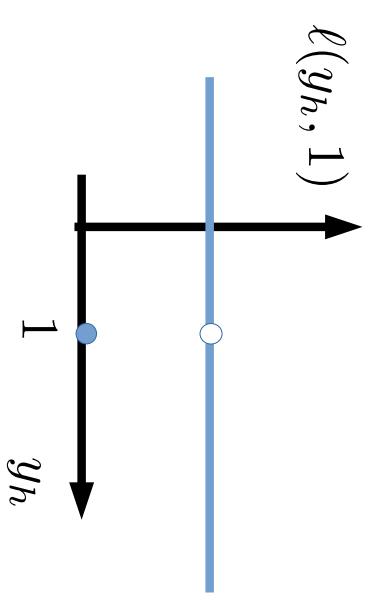


$\ell(y_h, 1)$

1

y_h

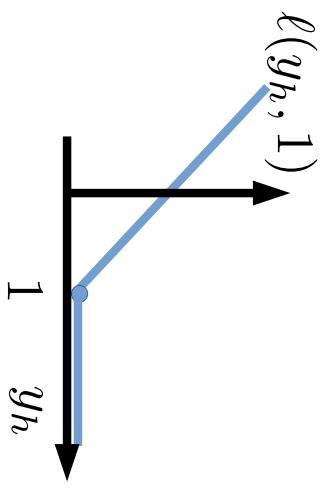
Binary Loss
$$\ell(y_h, y) = \begin{cases} 0 & \text{if } y_h = y \\ 1 & \text{if } y_h \neq y \end{cases}$$



1

y_h

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



1

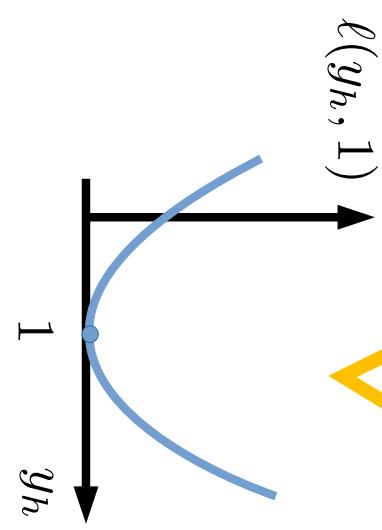
y_h

y=1 in all
figures

Choosing the Loss Function

Let $y_h := h_w(x)$

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

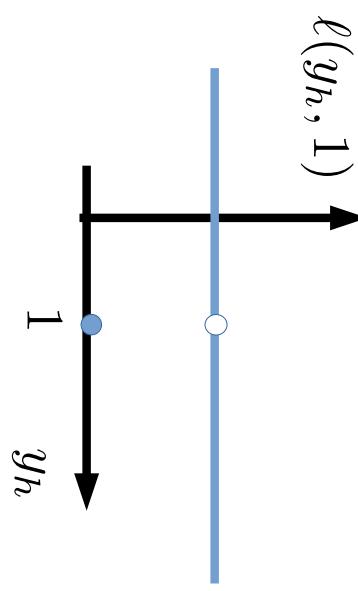


$\ell(y_h, 1)$

y_h

Binary Loss

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



$\ell(y_h, 1)$

y_h

Hinge Loss

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

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

$y=1$ in all figures

LOSS Functions

Is a notion of Loss enough?

What happens when we do not have enough data?

LOSS Functions

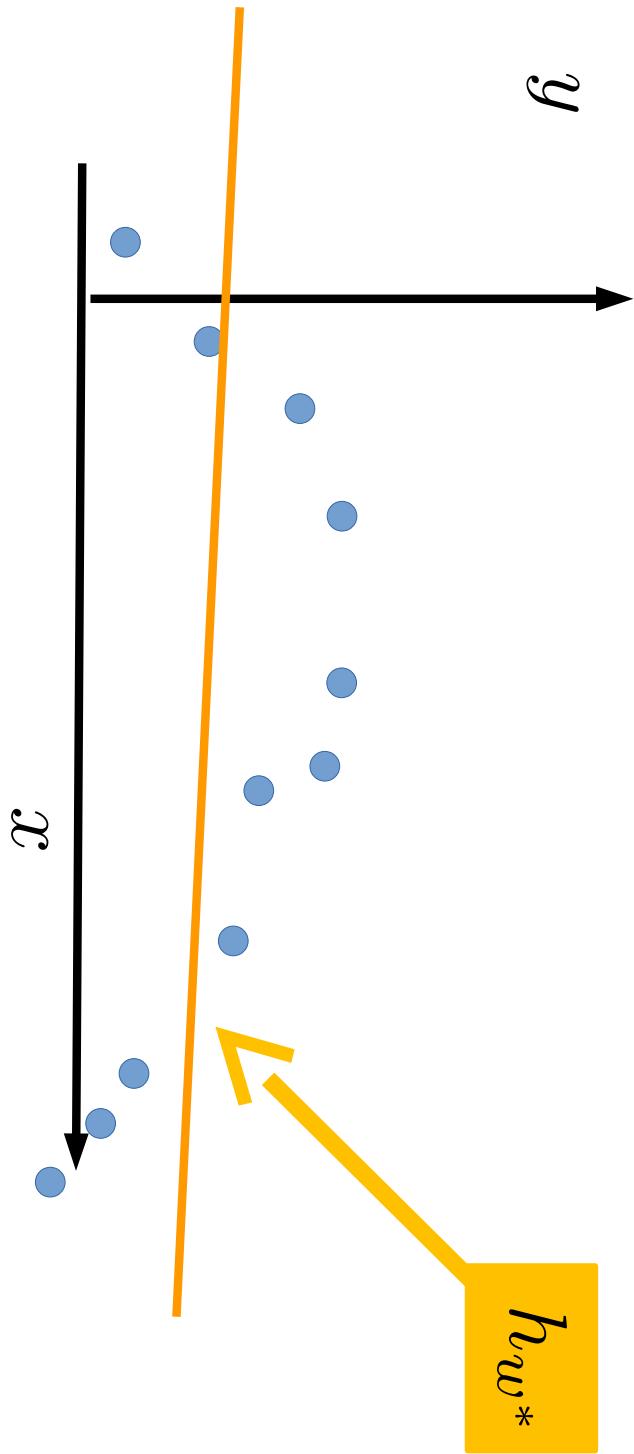
The Training Problem

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

Is a notion of Loss enough?

What happens when we do not have enough data?

Overfitting and Model Complexity

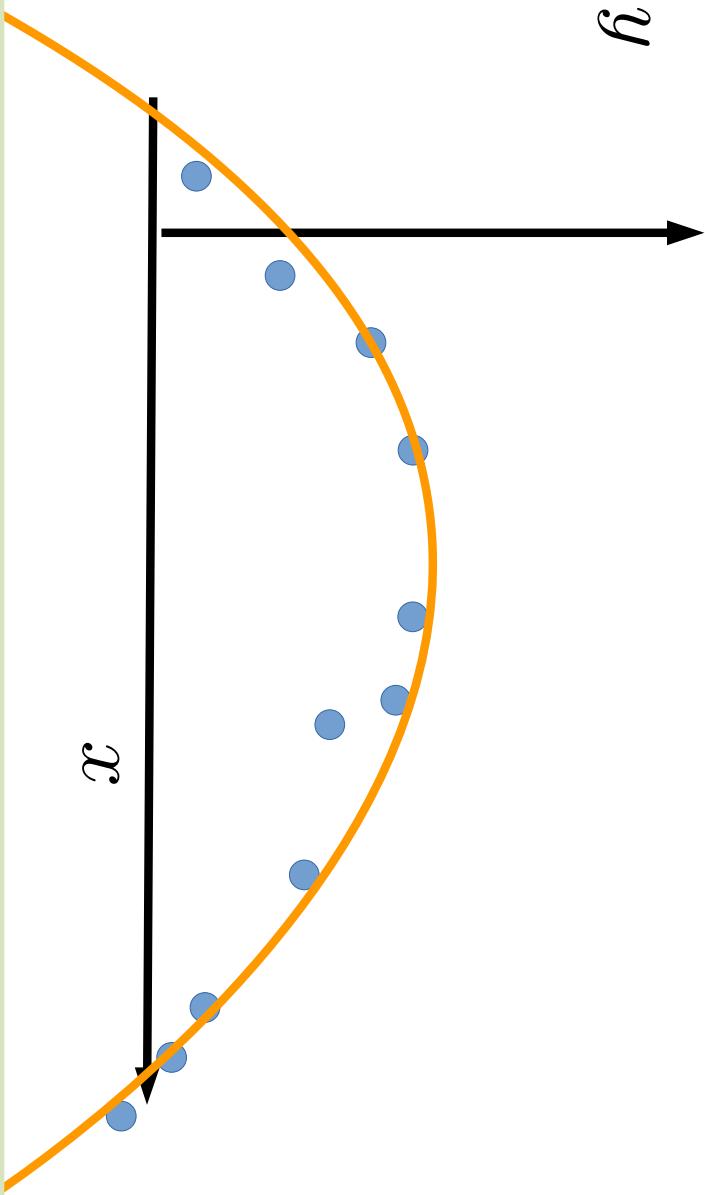


Fitting 1st order polynomial

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

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

Overfitting and Model Complexity

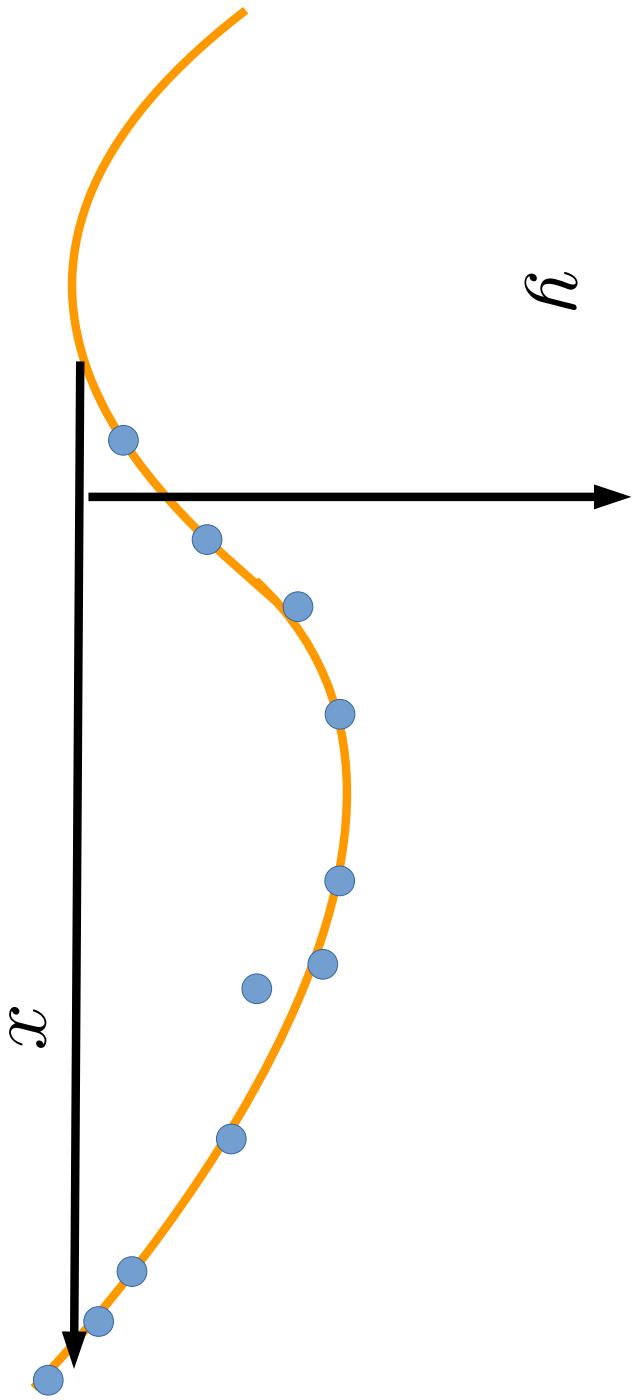


Fitting 2nd order polynomial

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

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

Overfitting and Model Complexity

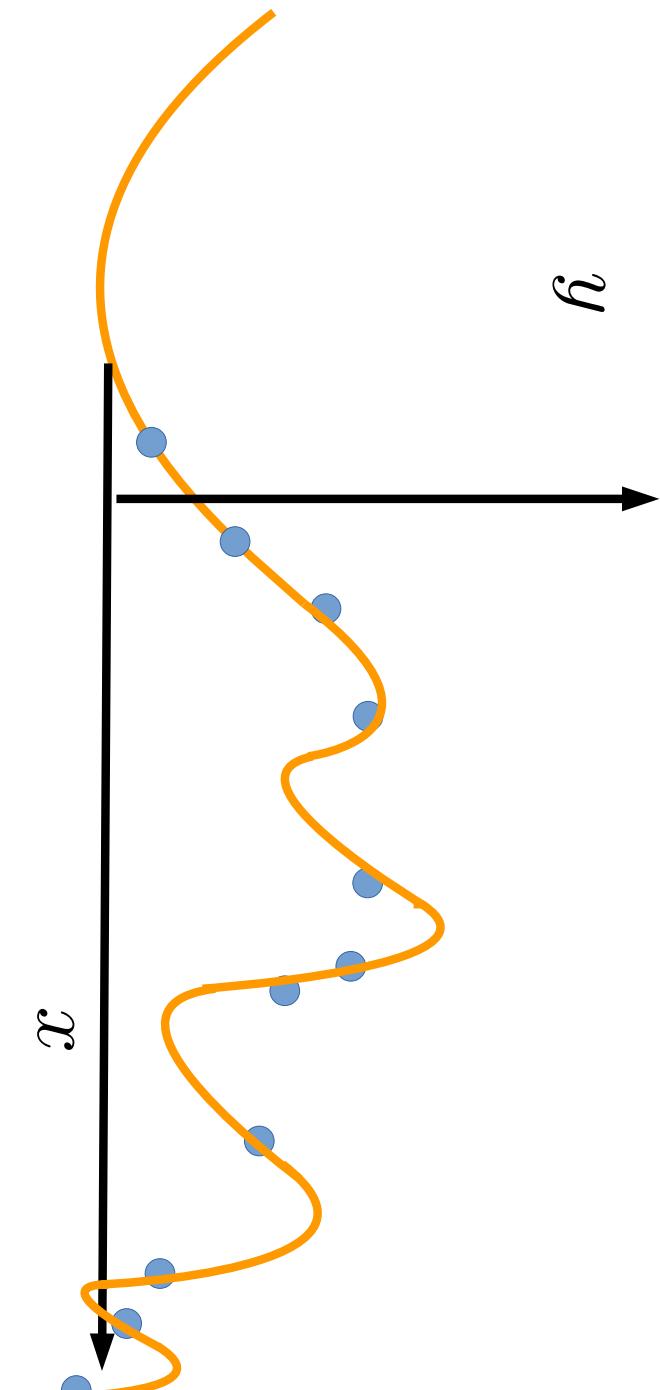


Fitting 3rd order polynomial

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

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

Overfitting and Model Complexity



Fitting 9th order polynomial

$$h_w = \sum_{i=0}^9 w_i x^i$$
$$w^* = \arg \min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^n (h_w(x^i) - y^i)^2$$

Regularization

Regularizer Functions

$$\begin{array}{ccc} R : & \mathbf{R}^d & \rightarrow \\ & w & \rightarrow R(w) \end{array}$$

General Training Problem

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

Regularization

Regularizer Functions

$$\begin{array}{ccc} R : & \mathbf{R}^d & \rightarrow \\ & w & \rightarrow R(w) \end{array}$$

General Training Problem

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

Goodness of fit,
fidelity term ...etc

Regularization

Regularizer Functions

$$\begin{array}{ccc} R : & \mathbf{R}^d & \rightarrow \\ & w & \rightarrow R(w) \end{array}$$

General Training Problem

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

Goodness of fit,
fidelity term ...etc

Penalizes
complexity

Regularization

Regularizer Functions

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

Controls tradeoff
between fit and
complexity

General Training Problem

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

Goodness of fit,
fidelity term ...etc

Penalizes
complexity

Regularization

Regularizer Functions

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

Controls tradeoff
between fit and
complexity

General Training Problem

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

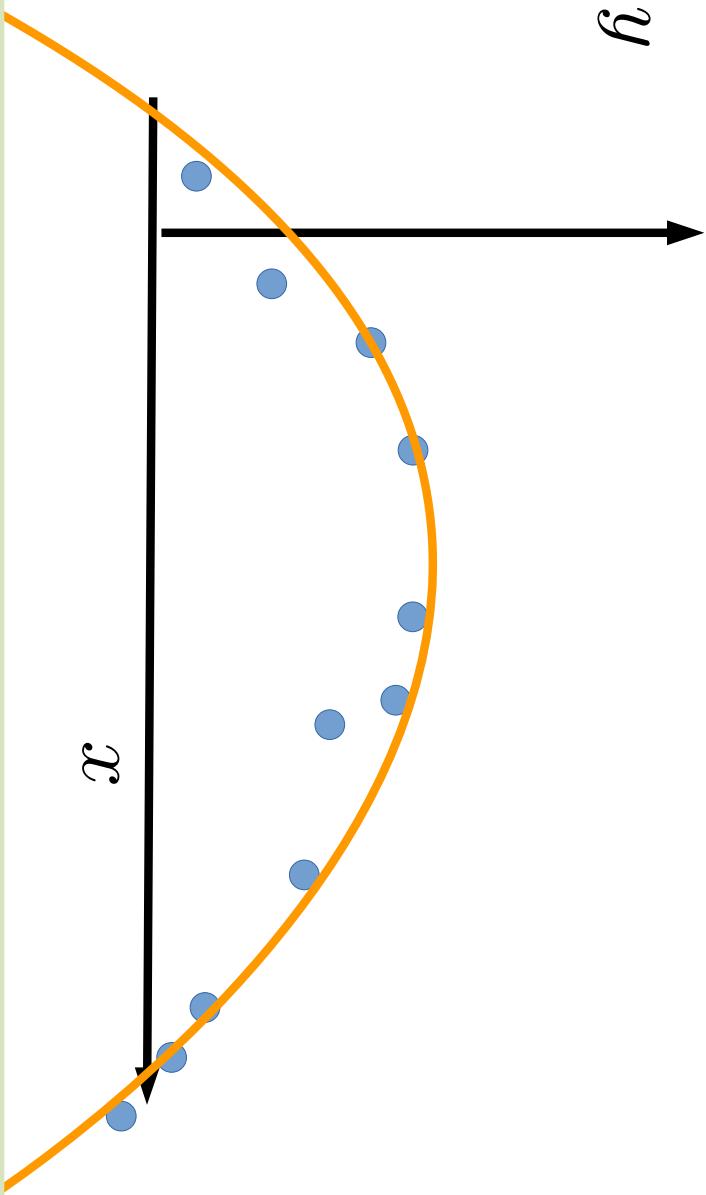
Goodness of fit,
fidelity term ...etc

Penalizes
complexity

Exe:

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

Overfitting and Model Complexity

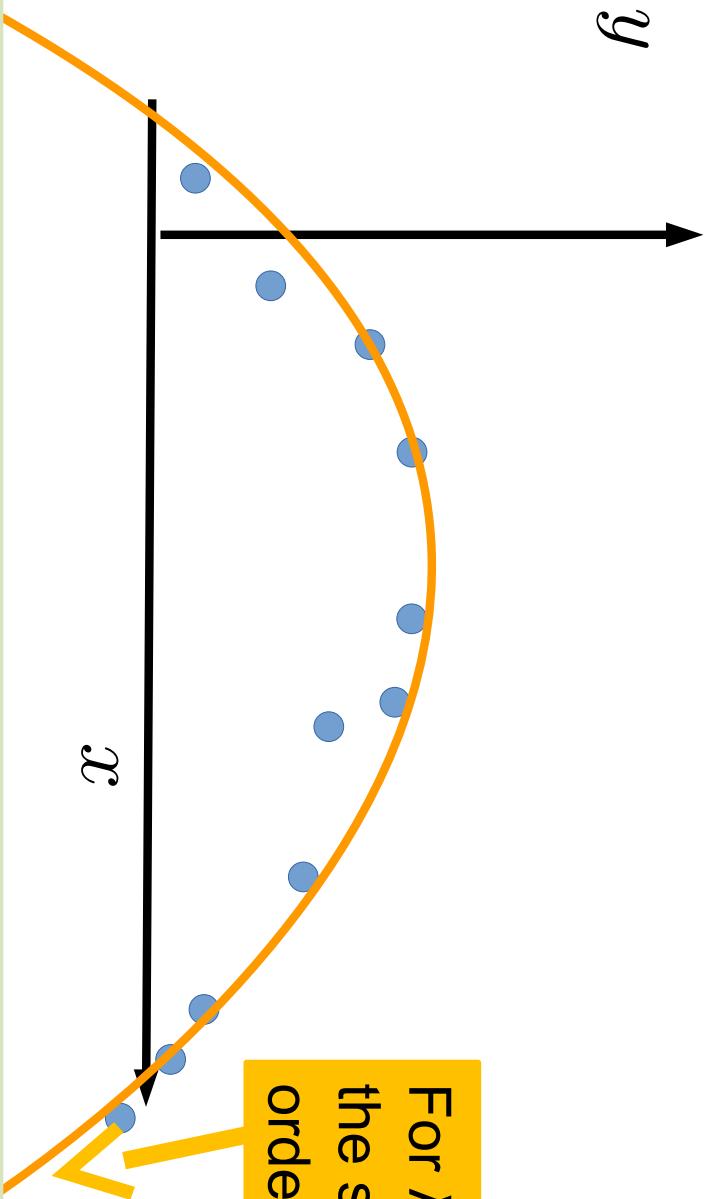


Fitting kth order polynomial

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

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

Overfitting and Model Complexity



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

Fitting kth order polynomial

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

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

Ex: Ridge Regression

Linear hypothesis

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

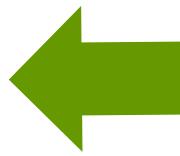


L2 regularizer

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

L2 loss

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



Ridge Regression

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

Ex: Support Vector Machines

Linear hypothesis

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

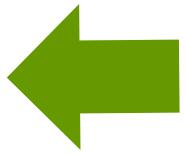


L2 regularizer

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

Hinge loss

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



SVM with soft margin

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

Ex: Logistic Regression

Linear hypothesis

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

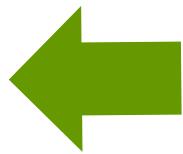


L2 regularizer

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

Logistic loss

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



Logistic Regression

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

The Machine Learners Job

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

The Machine Learners Job

- (1) Get the labeled data: $(x^1, y^1), \dots, (x^n, y^n)$
- (2) Choose a parametrization for hypothesis: $h_w(x)$

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: $\ell(h_w(x), y) \geq 0$

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

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

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

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

Finite Sum Training Problem

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

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

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

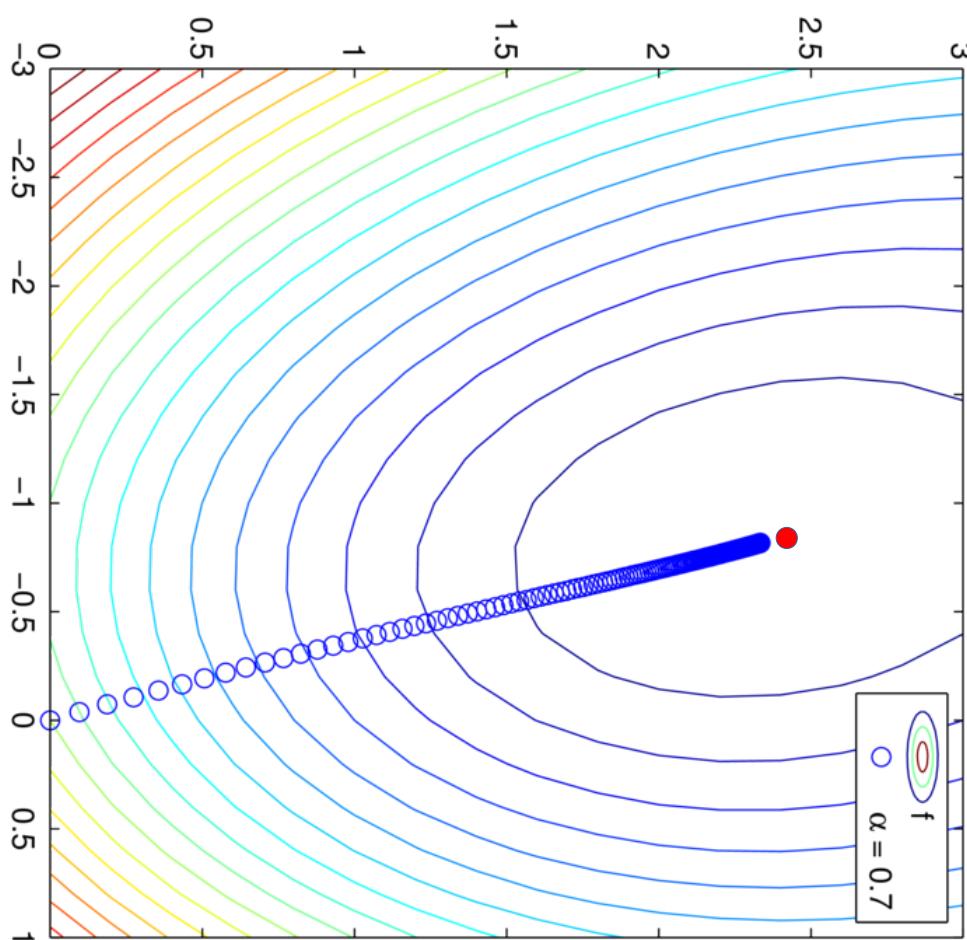
Output w^T

Gradient Descent Example

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

Logistic Regression

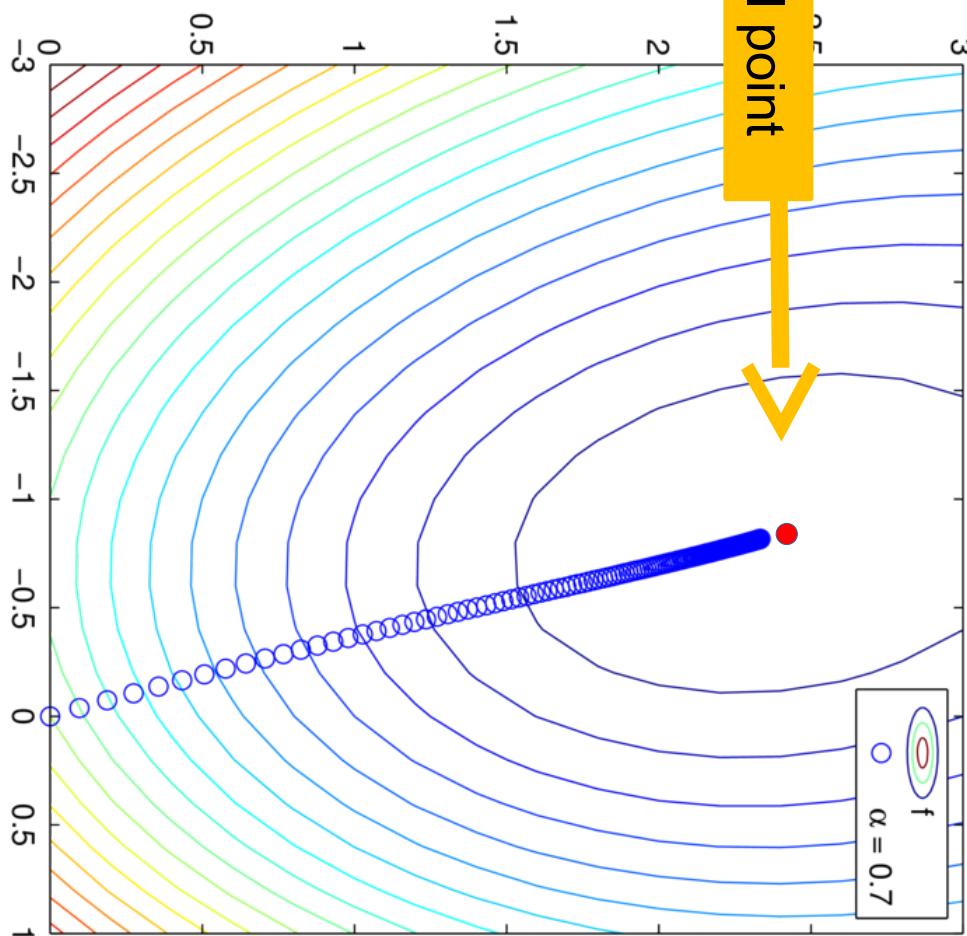
$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ln(1 + e^{-y^i \langle w, x^i \rangle}) + \lambda \|w\|_2^2$$



Can we prove
that this always
works?

Gradient Descent Example

Optimal point



Logistic Regression

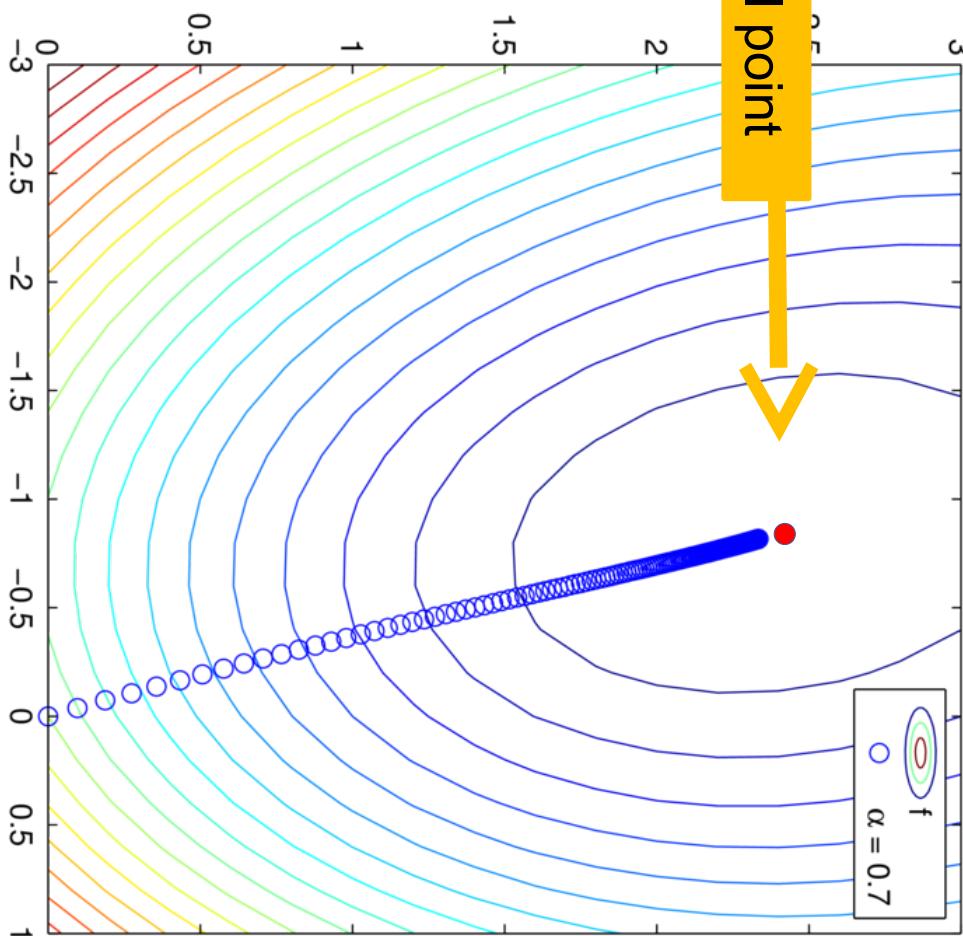
$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ln(1 + e^{-y^i \langle w, x^i \rangle}) + \lambda \|w\|_2^2$$

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

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

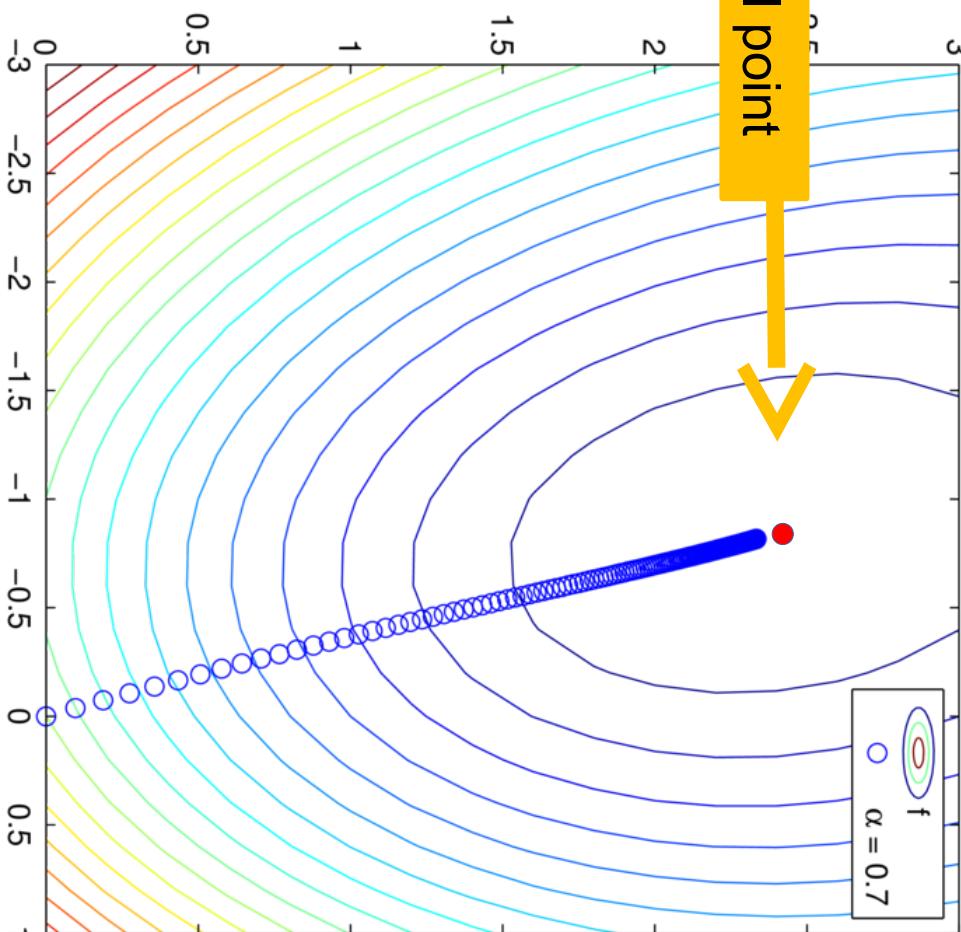
$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n \ln(1 + e^{-y^i \langle w, x^i \rangle}) + \lambda \|w\|_2^2$$

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



Logistic Regression

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

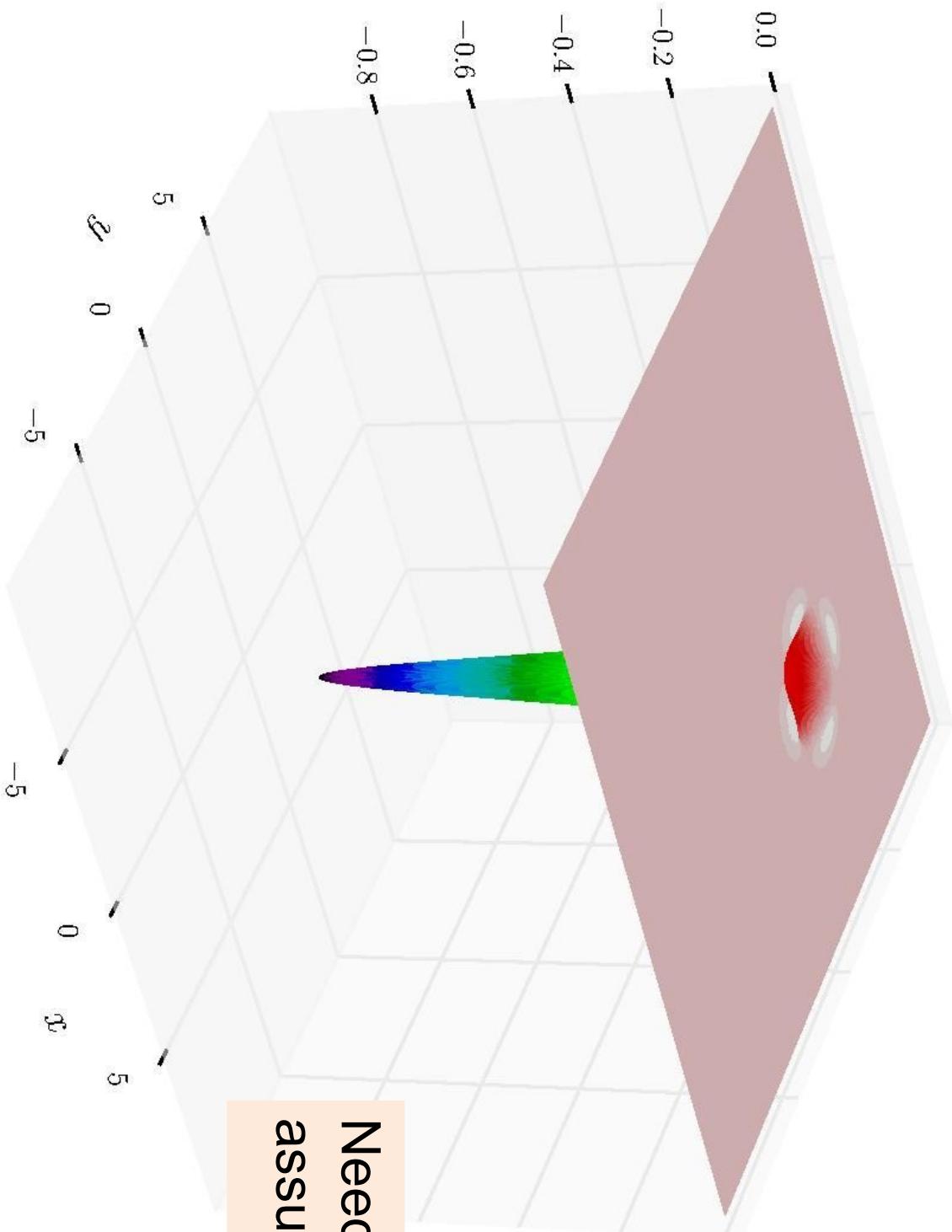
A Logistic Regression problem using the fourclass labelled data from LIBSVM
(n, d)=(862,2)

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

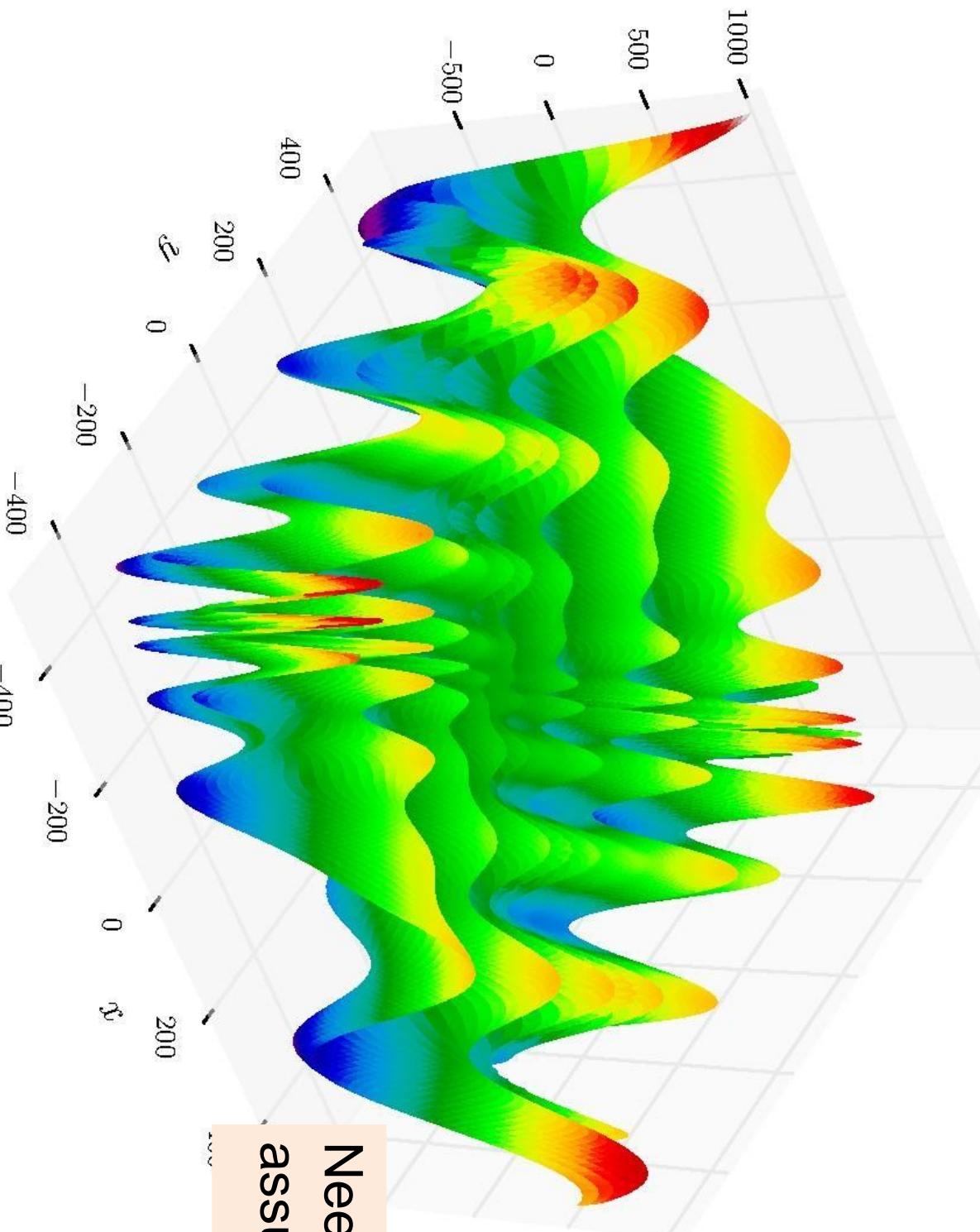
Optimization is hard (in general)



Need
assumptions!

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

Optimization is hard (in general)



Need
assumptions!

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

Main assumption

Nice property

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

All stationary points are
global minima

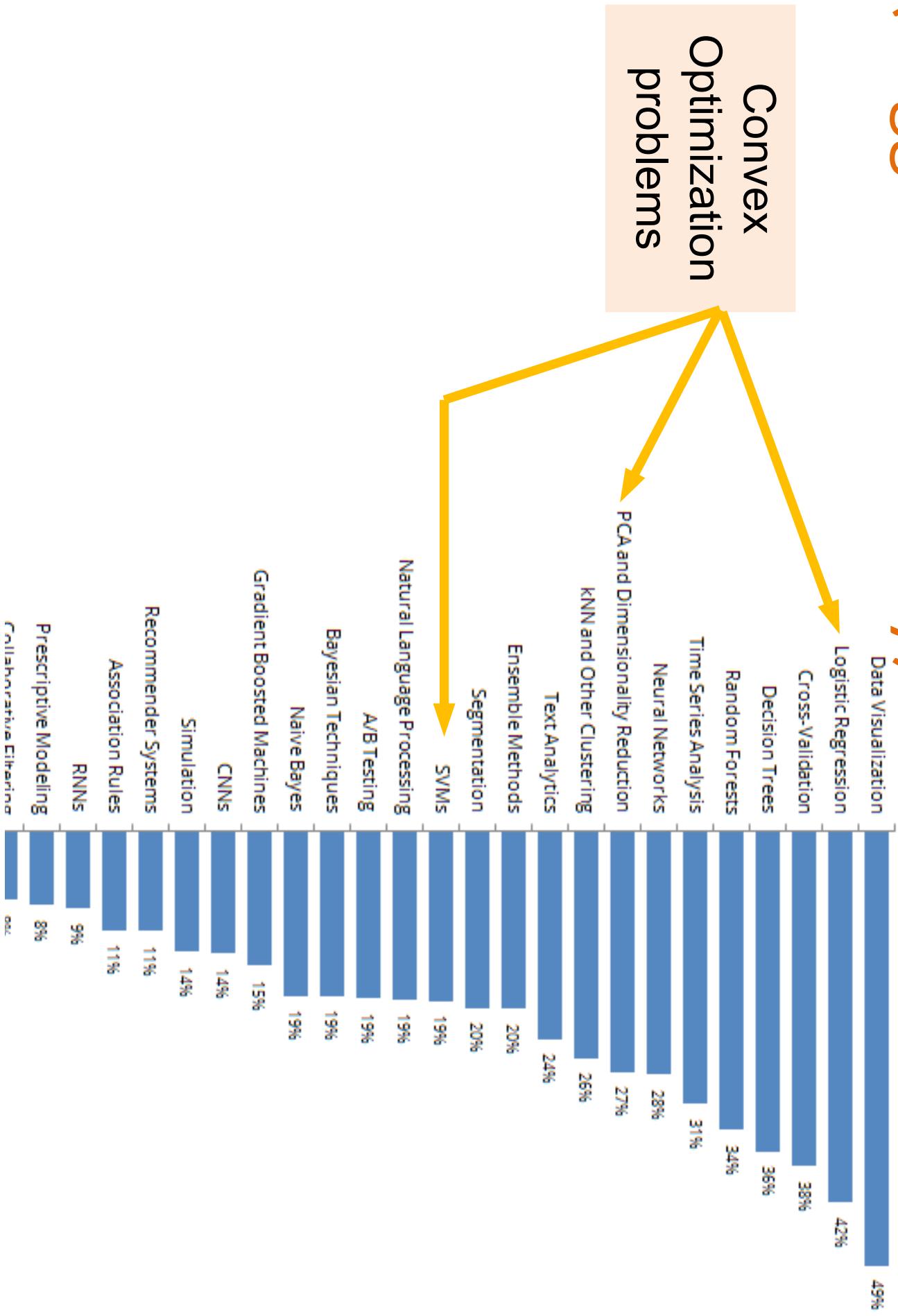
Lemma: Convexity => Nice property

If $f(w) \geq f(y) + \langle \nabla f(y), w - y \rangle$, $\forall w, y \in \mathbb{R}^d$
then nice property holds

PROOF: Choose $y = w^*$

Data science methods most used

(Kaggle 2017 survey)



Part I: 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 $t = 0, 1, 2, \dots, T$

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

Output w^T

Stochastic Gradient Descent

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

Stochastic Gradient Descent

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

Unbiased Estimate

Let j be a random index sampled from $\{1, \dots, n\}$ 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 at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \dots, n\}$ 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 at each iteration?

Unbiased Estimate

Let j be a random index sampled from $\{1, \dots, n\}$ 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

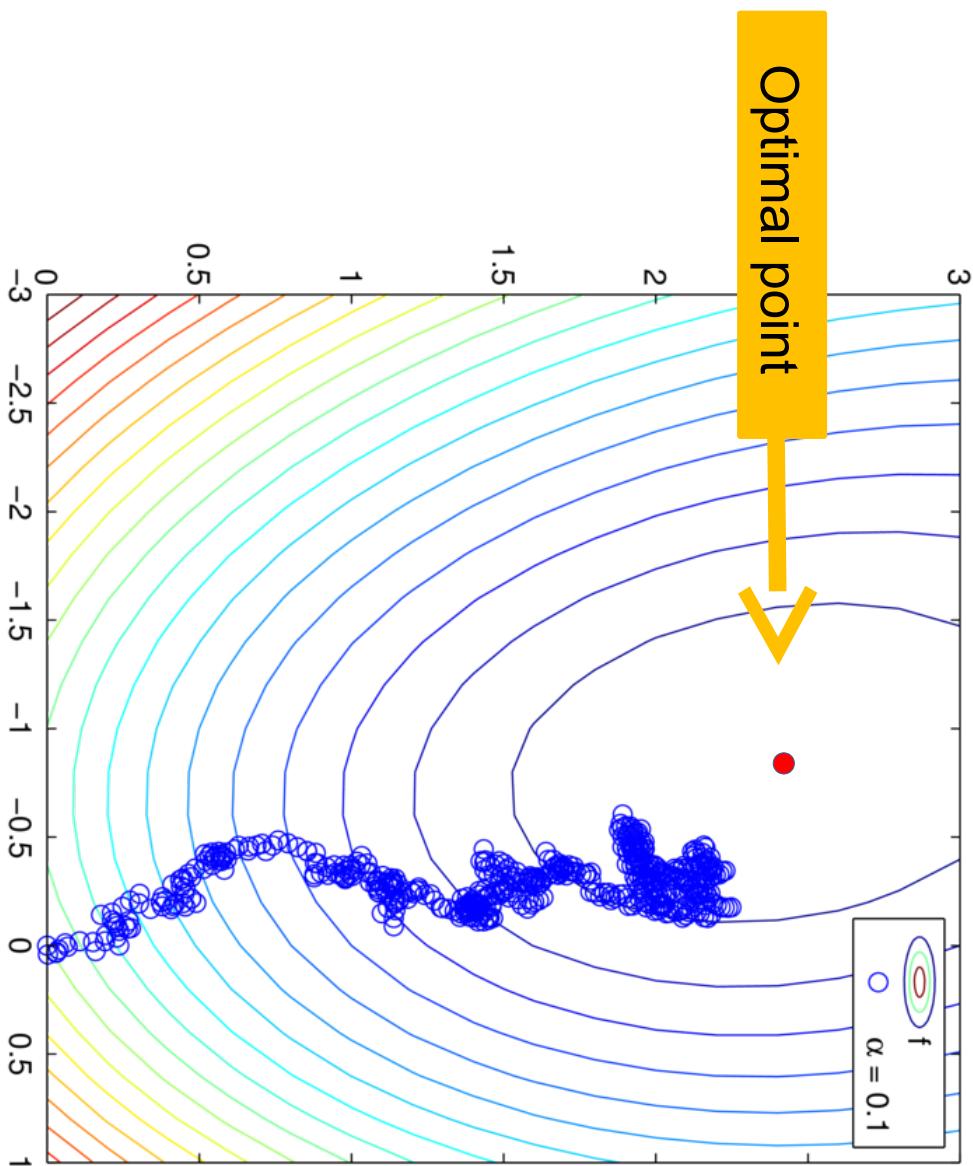
Set $w^0 = 0$, choose $\alpha > 0$
for $t = 0, 1, 2, \dots, T - 1$

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

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

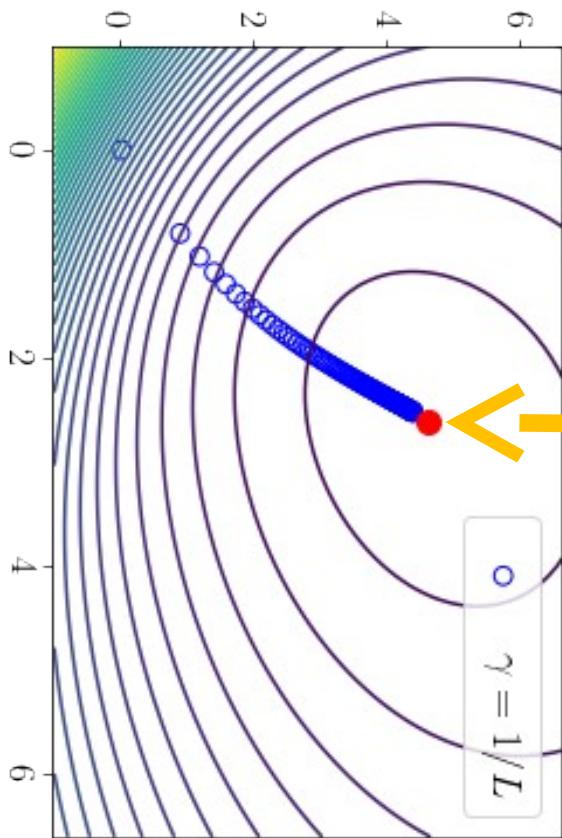
Output w^T

Stochastic Gradient Descent



GD vs Stochastic Gradient Descent

Optimal point w^*



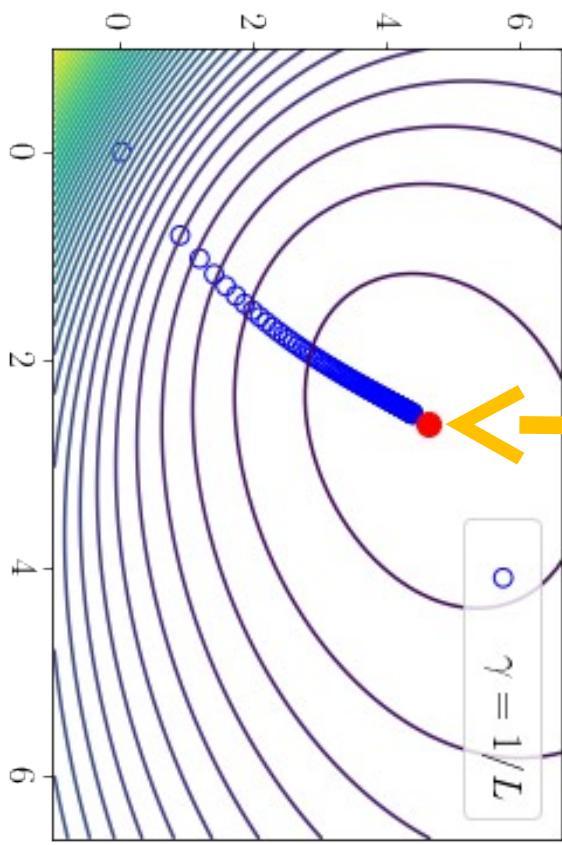
Gradient Descent

Need Assumptions

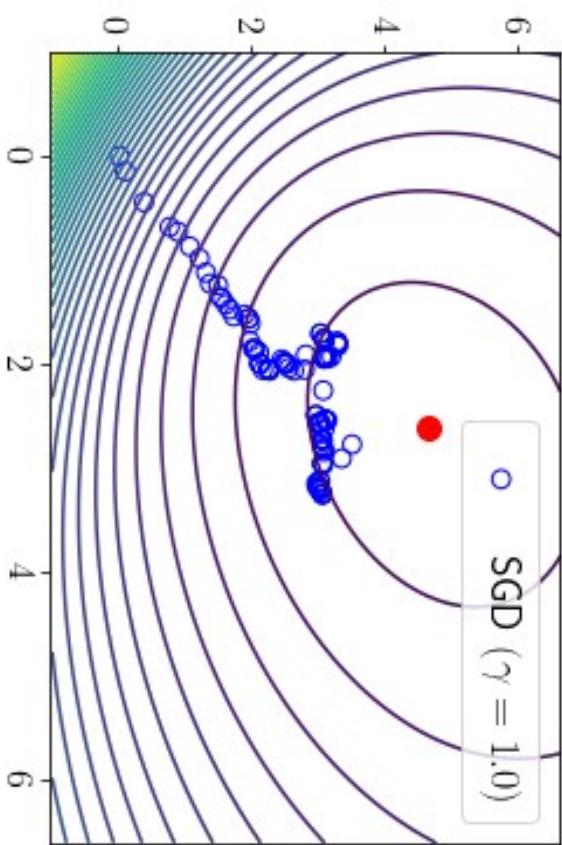


GD vs Stochastic Gradient Descent

Why does this happen?



Optimal point w^*



Gradient Descent
Stochastic Gradient
Descent

Need Assumptions

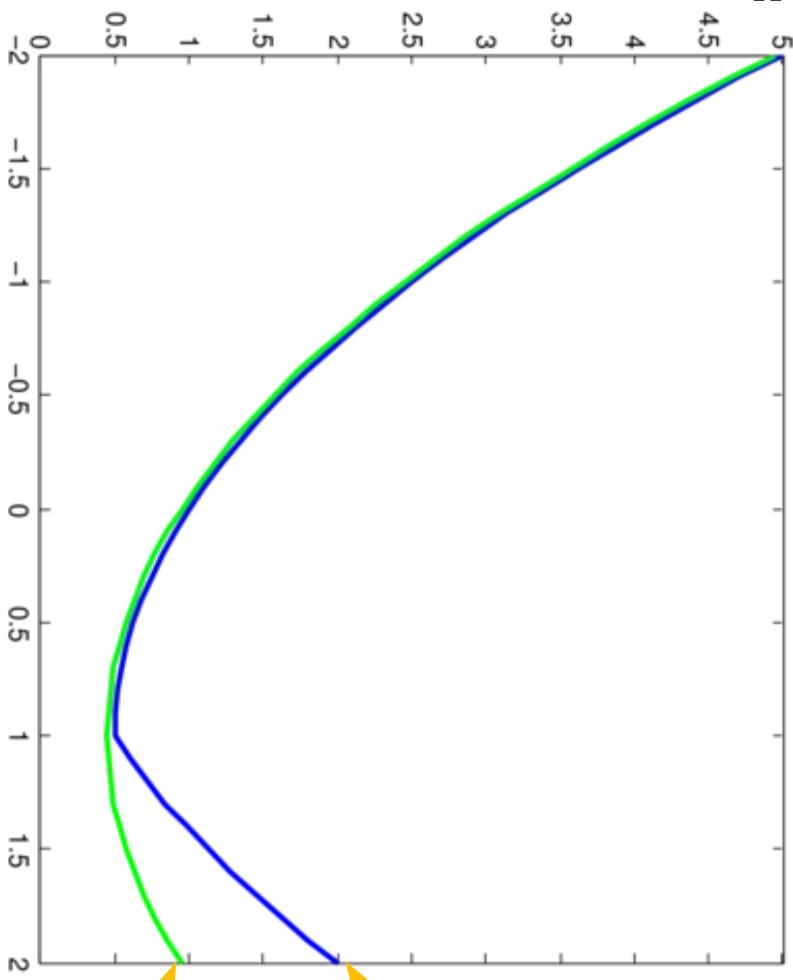


Assumption: Strong convexity

We say $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ 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

$$f(w) := \frac{1}{n} \sum_{i=1}^n \underbrace{\ell(h_w(x^i), y^i)}_{\parallel} + \underbrace{\lambda R(w)}_{\parallel}$$

\parallel \parallel

Strongly convex = convex + $\frac{1}{2} \|w\|^2$

Example: SVM with soft margin

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

Not an Example: Neural networks, dictionary learning,

And more

Assumption: Smoothness

We say $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ 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 $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ 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 $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is L -smooth then

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

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

Assumption: Smoothness

We say $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ 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 $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is L -smooth then

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

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

EXE: Using that

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

Show that

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

Smoothness: Examples

Convex quadratics:

$$x \mapsto x^\top Ax + b^\top x + c$$

Logistic:

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

Trigonometric:

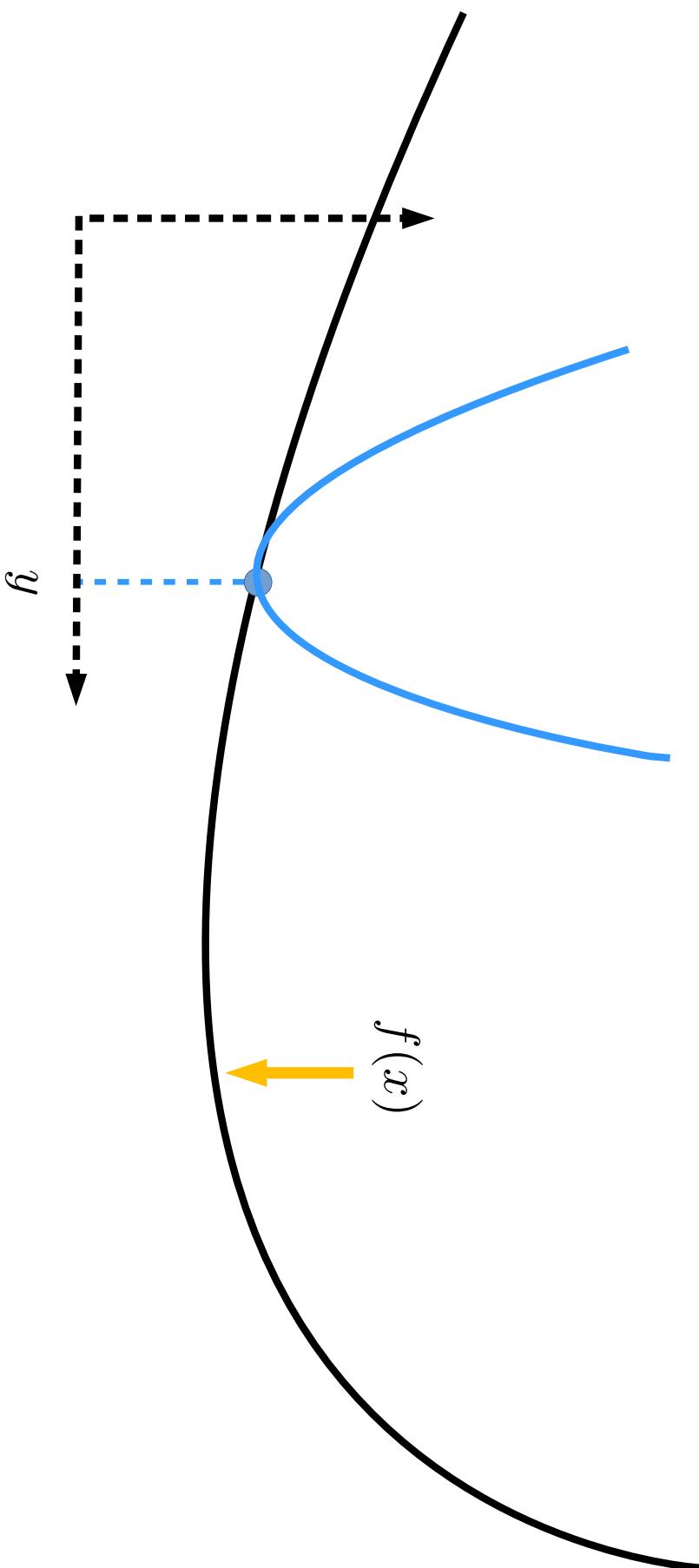
$$x \mapsto \cos(x), \sin(x)$$

Proof is an
exercise!

Important consequences of Smoothness

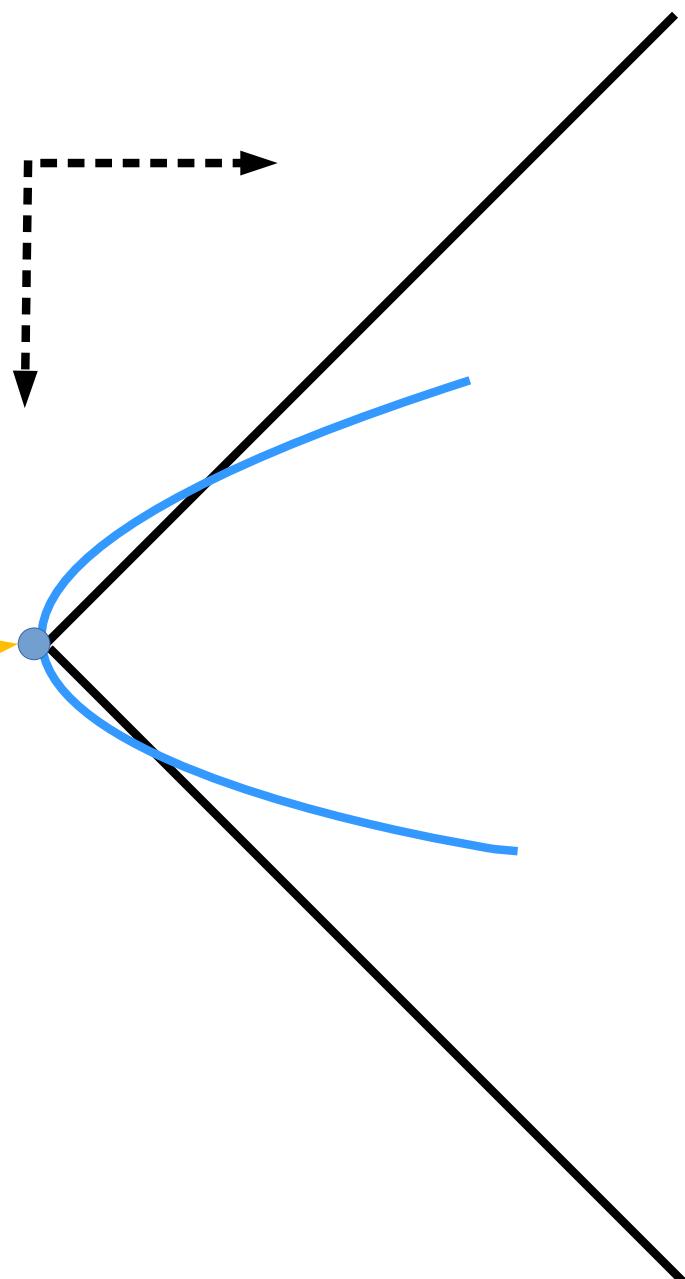
If $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is L -smooth then

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



Smoothness: Convex counter-example

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



Non-smooth can be solved 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

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

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

Definition: Gradient Noise

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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



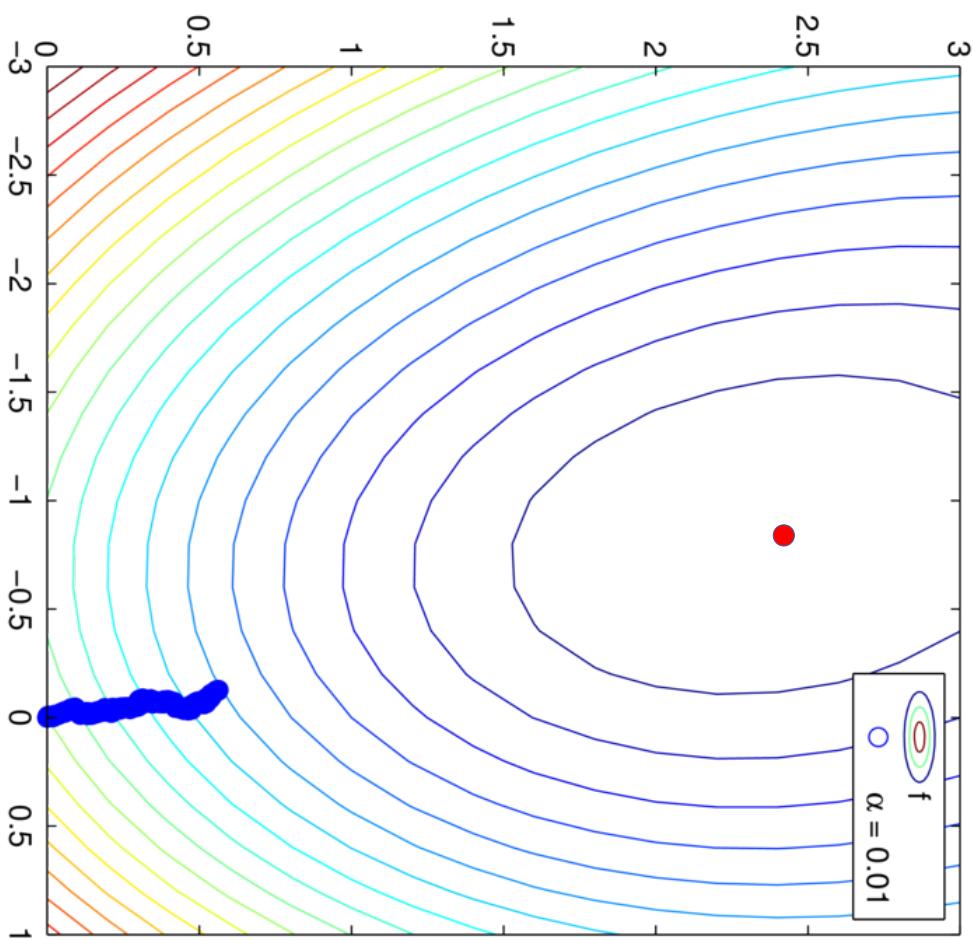
Shows that $\alpha \approx \frac{1}{\mu}$

Shows that $\alpha \approx 0$



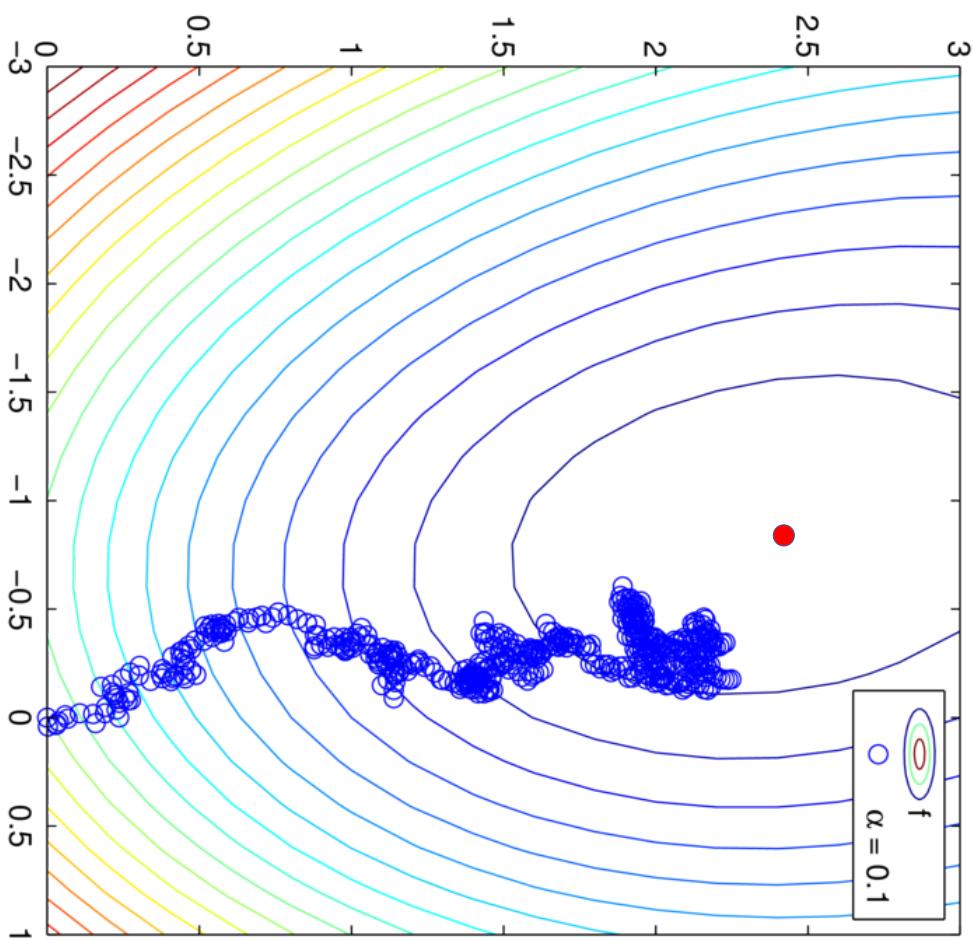
Stochastic Gradient Descent

$\alpha = 0.01$



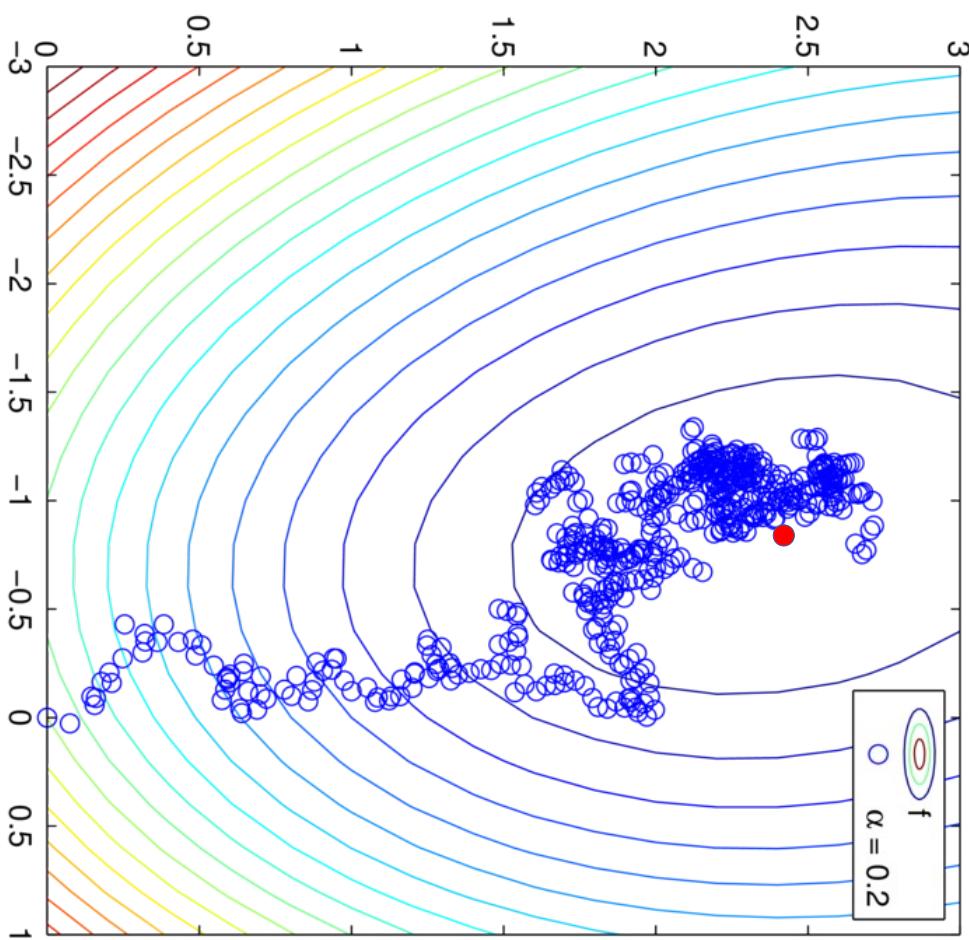
Stochastic Gradient Descent

$\alpha = 0.1$



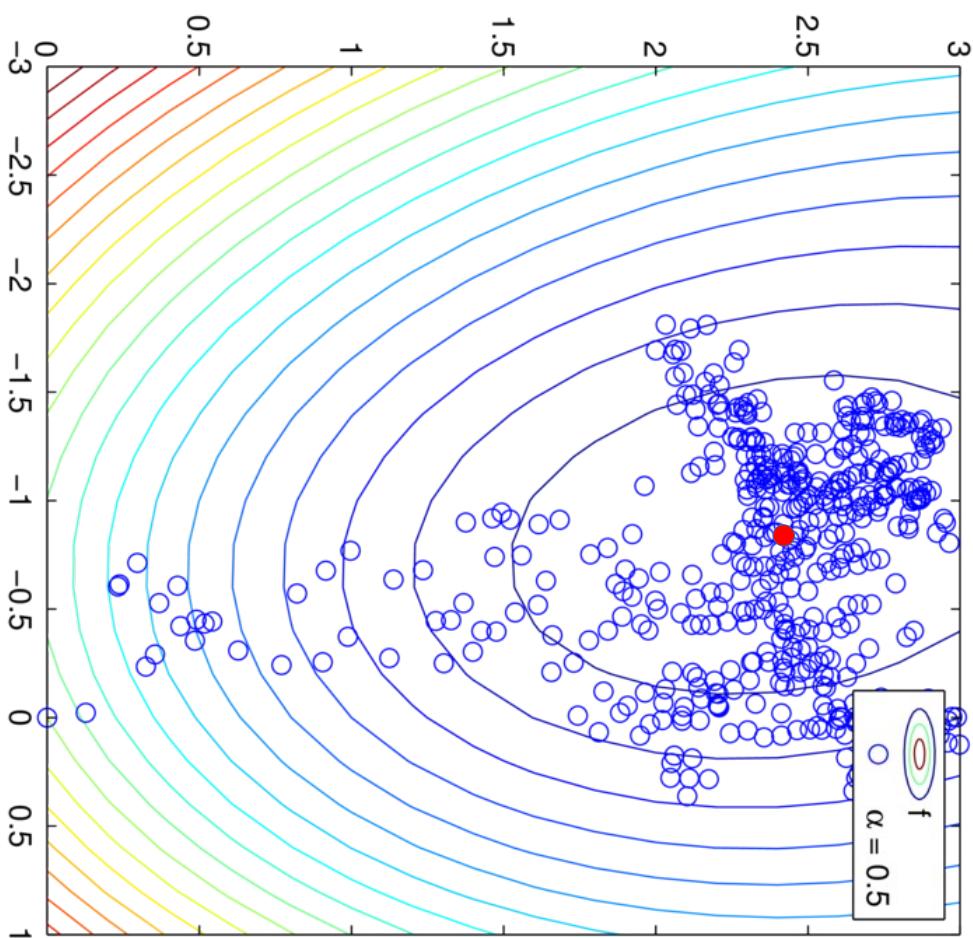
Stochastic Gradient Descent

$\alpha = 0.2$



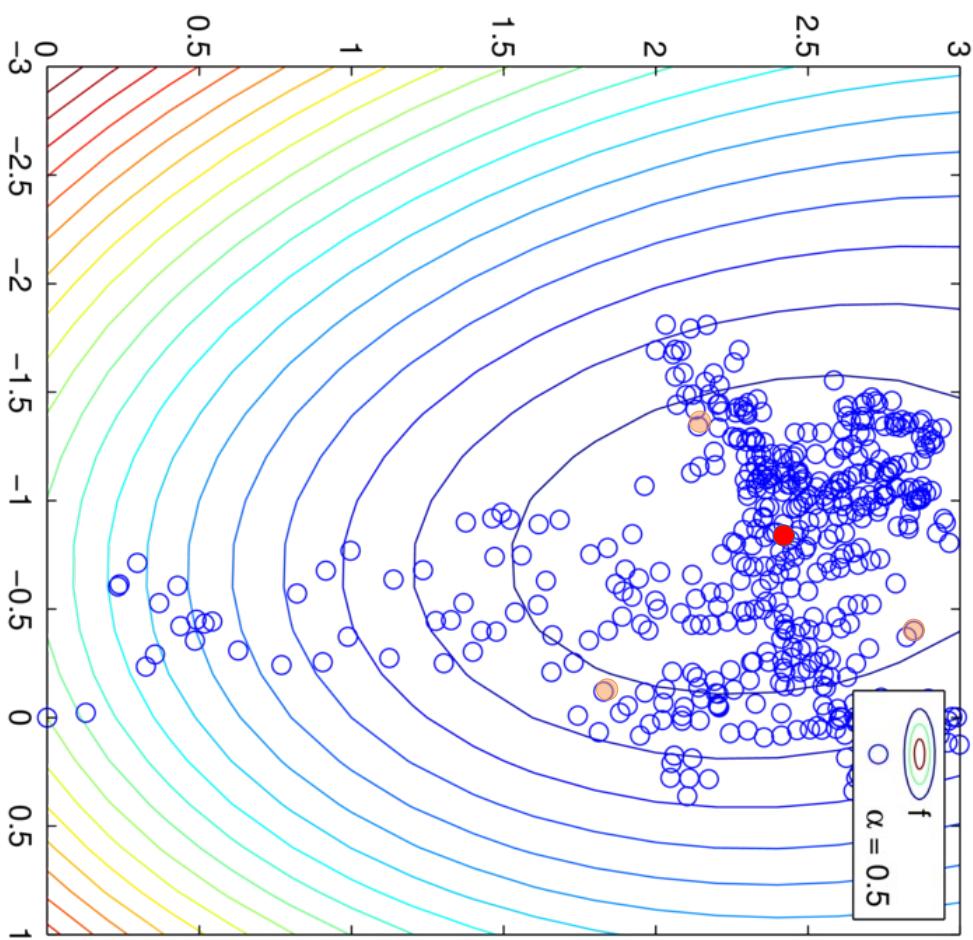
Stochastic Gradient Descent

$\alpha = 0.5$



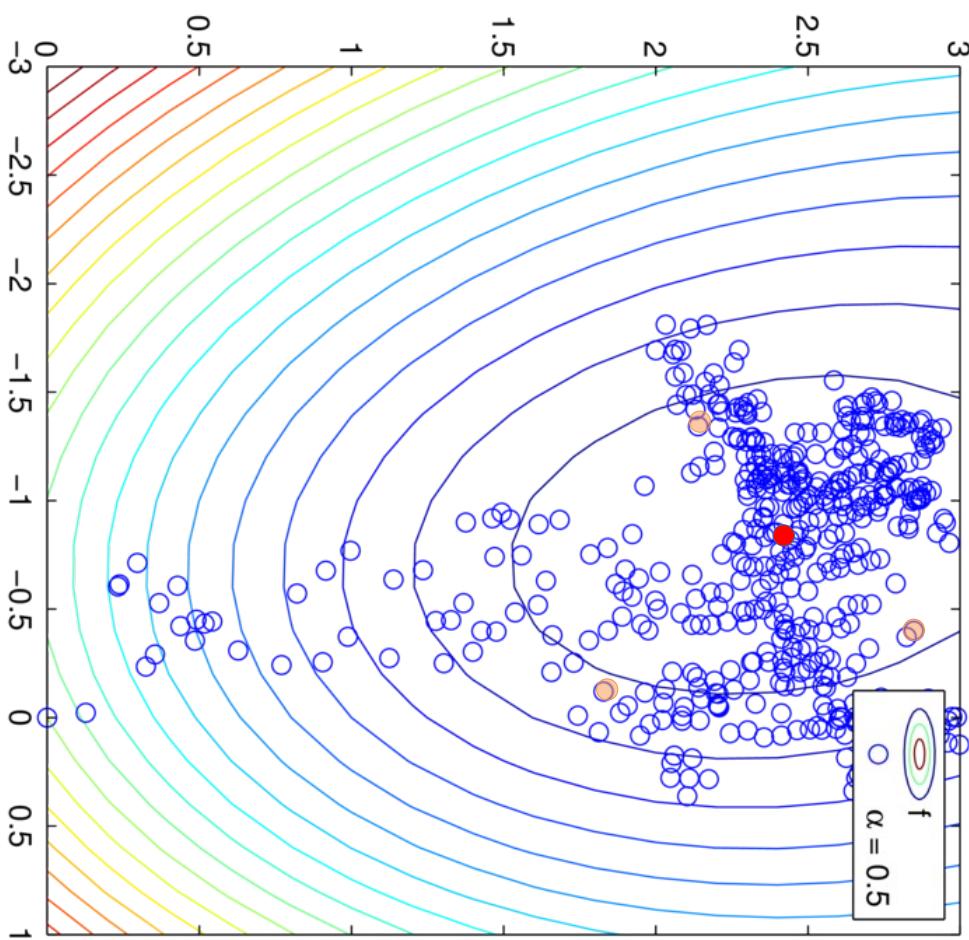
Stochastic Gradient Descent

$\alpha = 0.5$



Stochastic Gradient Descent

$\alpha = 0.5$

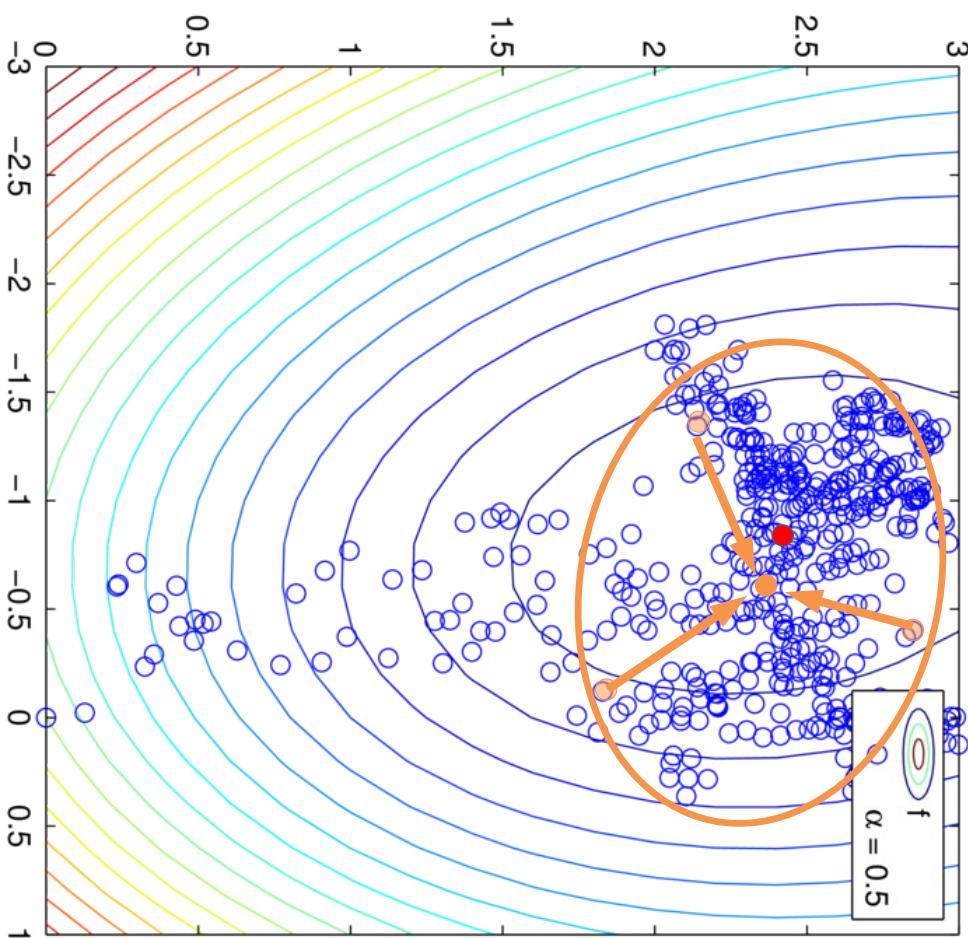


2) Try
averaging
the points

1) Start with
big steps and
end with
smaller steps

Stochastic Gradient Descent

$\alpha = 0.5$



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 $\alpha_t > 0$, $\alpha_t \rightarrow 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$
for $t = 0, 1, 2, \dots, T - 1$

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

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

Output w^T

Shrinking
Stepsize



SGD shrinking stepsize

98

SGD Shrinking stepsize

Set $w^0 = 0$

Choose $\alpha_t > 0$, $\alpha_t \rightarrow 0$, $\sum_{t=0}^{\infty} \alpha_t = \infty$
for $t = 0, 1, 2, \dots, T - 1$

Sample $j \in \{1, \dots, n\}$

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

Output w^T



How should we
sample j ?

Shrinking
Step size

How fast $\alpha_t \rightarrow 0$?

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

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

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

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

Complexity / Convergence

102

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

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

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

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

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

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

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



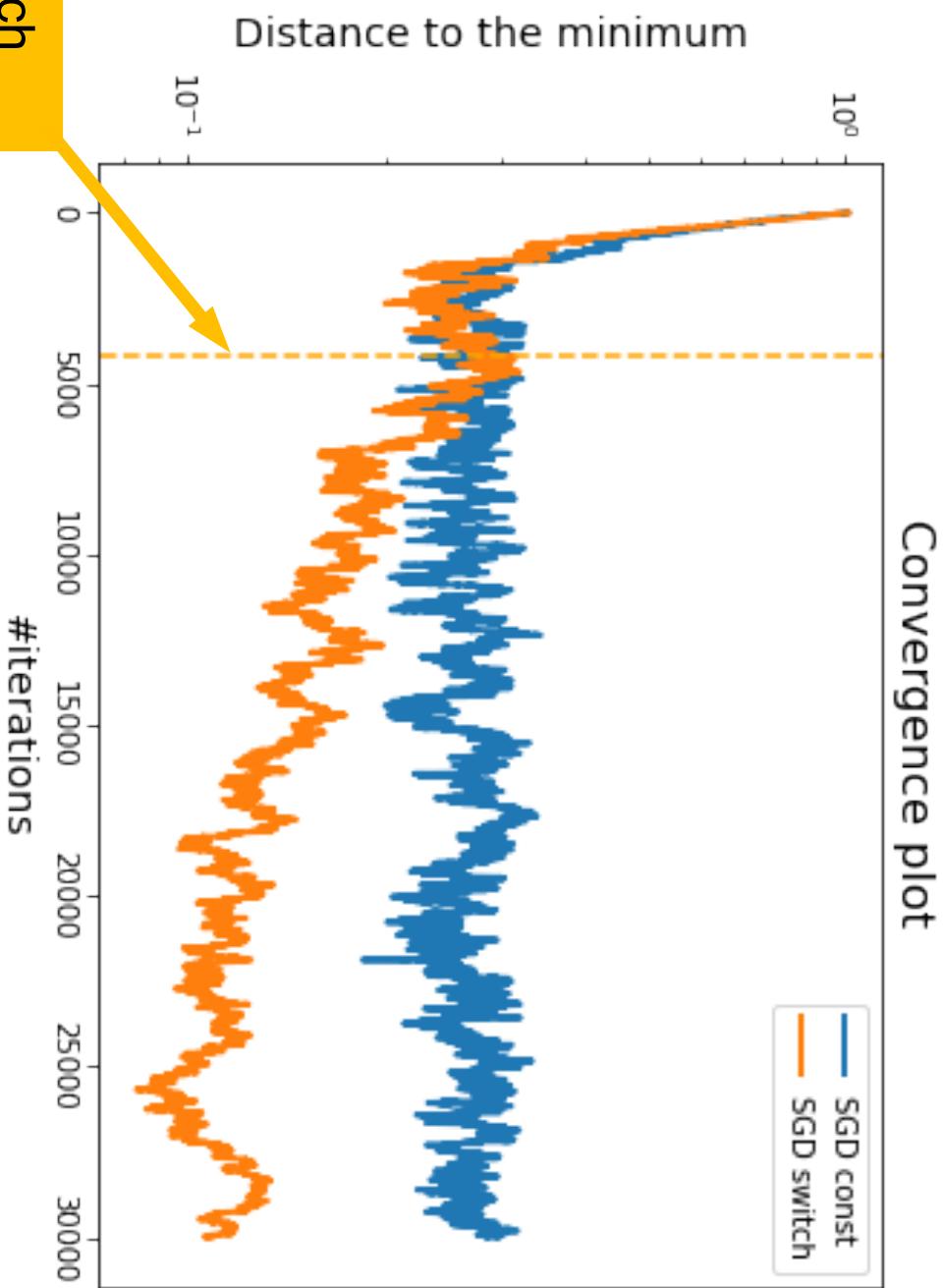
$$\alpha^t = O(1/(t+1))$$

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

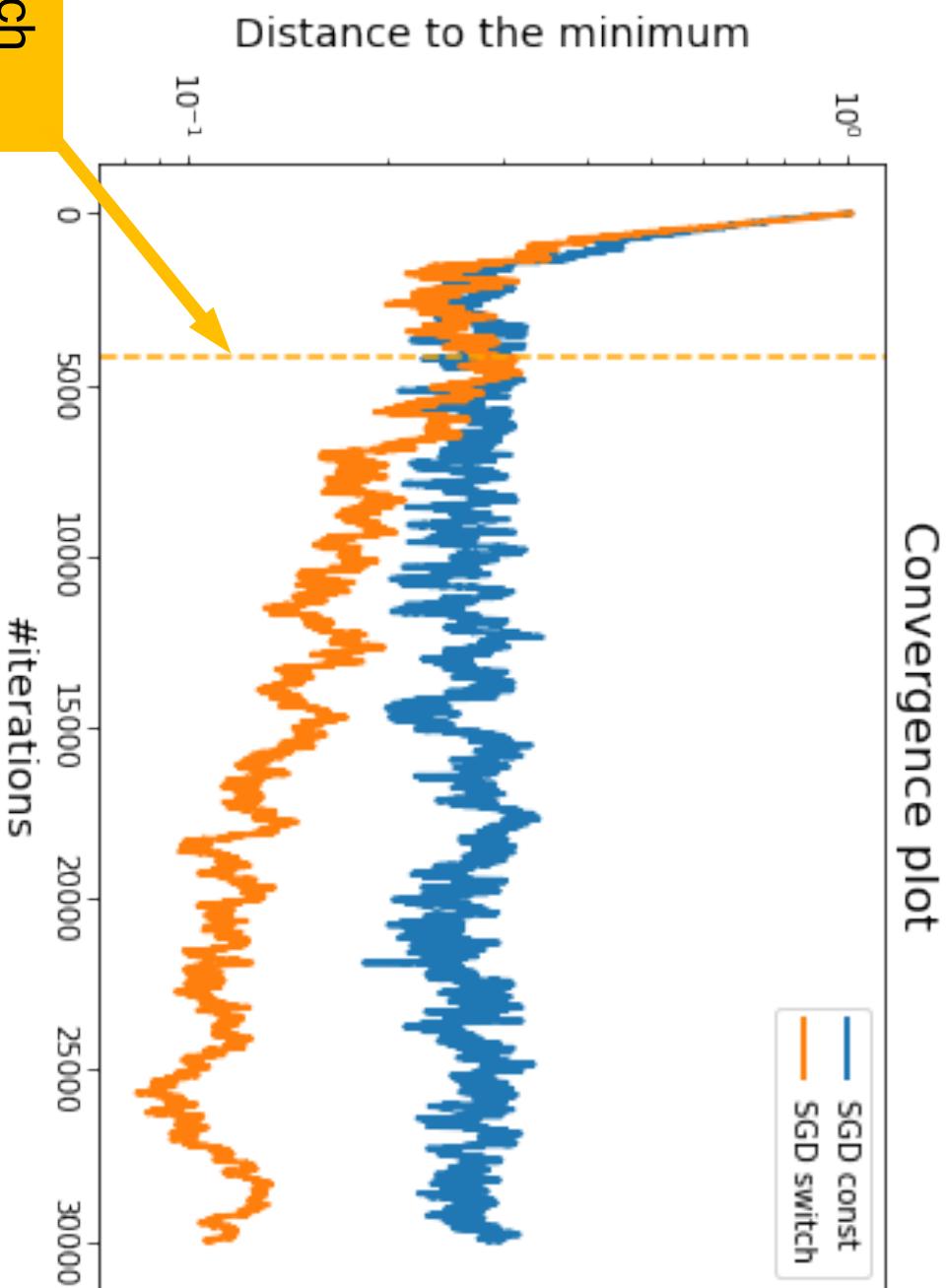
$$\mathbb{E}\|w^t - w^*\|^2 \leq \frac{\sigma^2}{\mu^2} \frac{8}{t} + \frac{16}{e^2} \frac{[\mathcal{K}]^2}{t^2} \|w^0 - w^*\|^2$$

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

Stochastic Gradient Descent with switch to decreasing stepsizes



Stochastic Gradient Descent with switch to decreasing stepsizes



Noisy iterates.
Take averages?

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

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

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

if $t > s_0$

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

else: $\bar{w} = w$

Output \bar{w}



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

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

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

if $t > s_0$

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

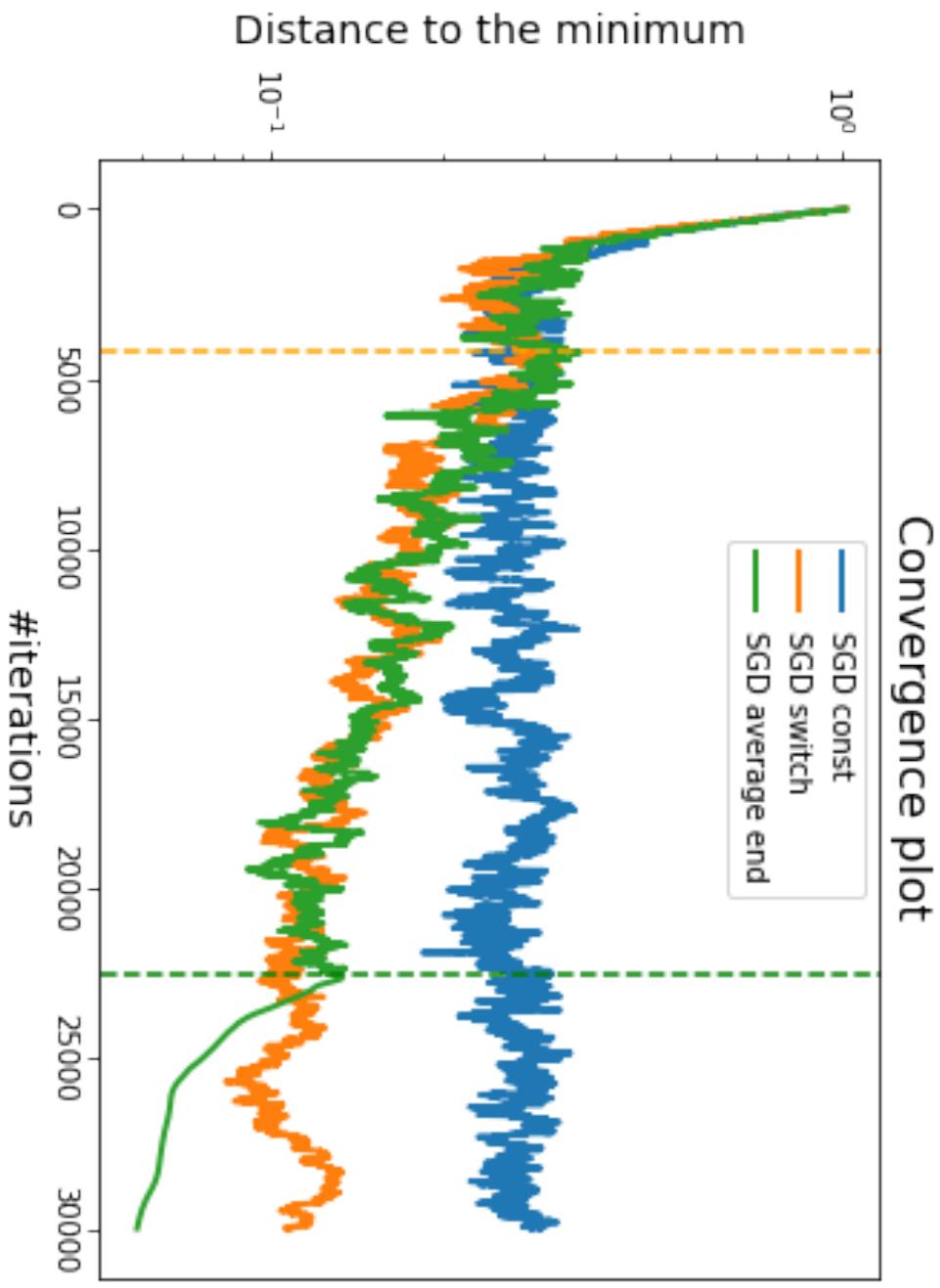
else: $\bar{w} = w$

Output \bar{w}

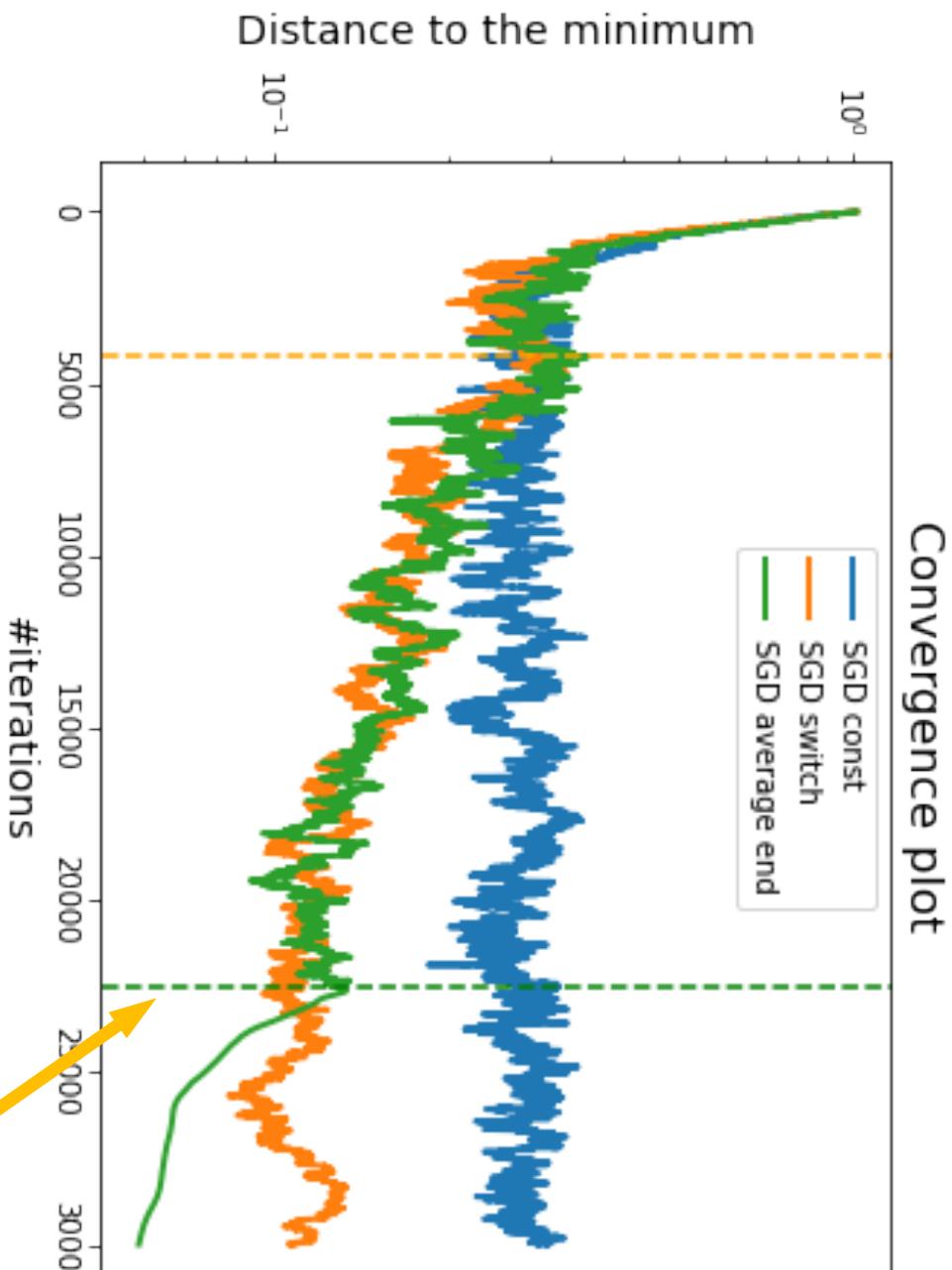
This is not efficient.
 How to make this
 efficient?




Stochastic Gradient Descent Averaging the last few iterates



Stochastic Gradient Descent Averaging the last few iterates



Part III.2: Stochastic Gradient Descent for Sparse Data

Lazy SGD updates for Sparse Data

Finite Sum Training Problem

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

L2 regularizer + 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?

Sparse Examples:

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

Lazy SGD updates for Sparse Data

Finite Sum Training Problem

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

L2 regularizer + 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?

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

Sparse Examples:

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

Lazy SGD updates for Sparse Data

Finite Sum Training Problem

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

L2 regularizer + 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?

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



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 $O(s)$?

$$\begin{aligned} \beta_{t+1} z^{t+1} &= (1 - \lambda\alpha_t)\beta_t z^t - \alpha_t \ell'(\beta_t \langle z^t, x^i \rangle, y^i) x^i \\ &= (1 - \lambda\alpha_t)\beta_t \left(z^t - \frac{\alpha_t \ell'(\beta_t \langle z^t, x^i \rangle, y^i)}{(1 - \lambda\alpha_t)\beta_t} x^i \right) \\ &\quad \underbrace{\hspace{-10em}}_{\beta_{t+1}} \quad \underbrace{\hspace{-10em}}_{z^{t+1}} \end{aligned}$$

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 $O(s)$?

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

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

$$\beta_{t+1}$$

$$z^{t+1}$$

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

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 $O(s)$?

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

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

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

$$\beta_{t+1} \quad \quad \quad z^{t+1}$$

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

gradient descent

Part IV: Momentum and

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

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

GD motivated through local rate of change

Local rate of change

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

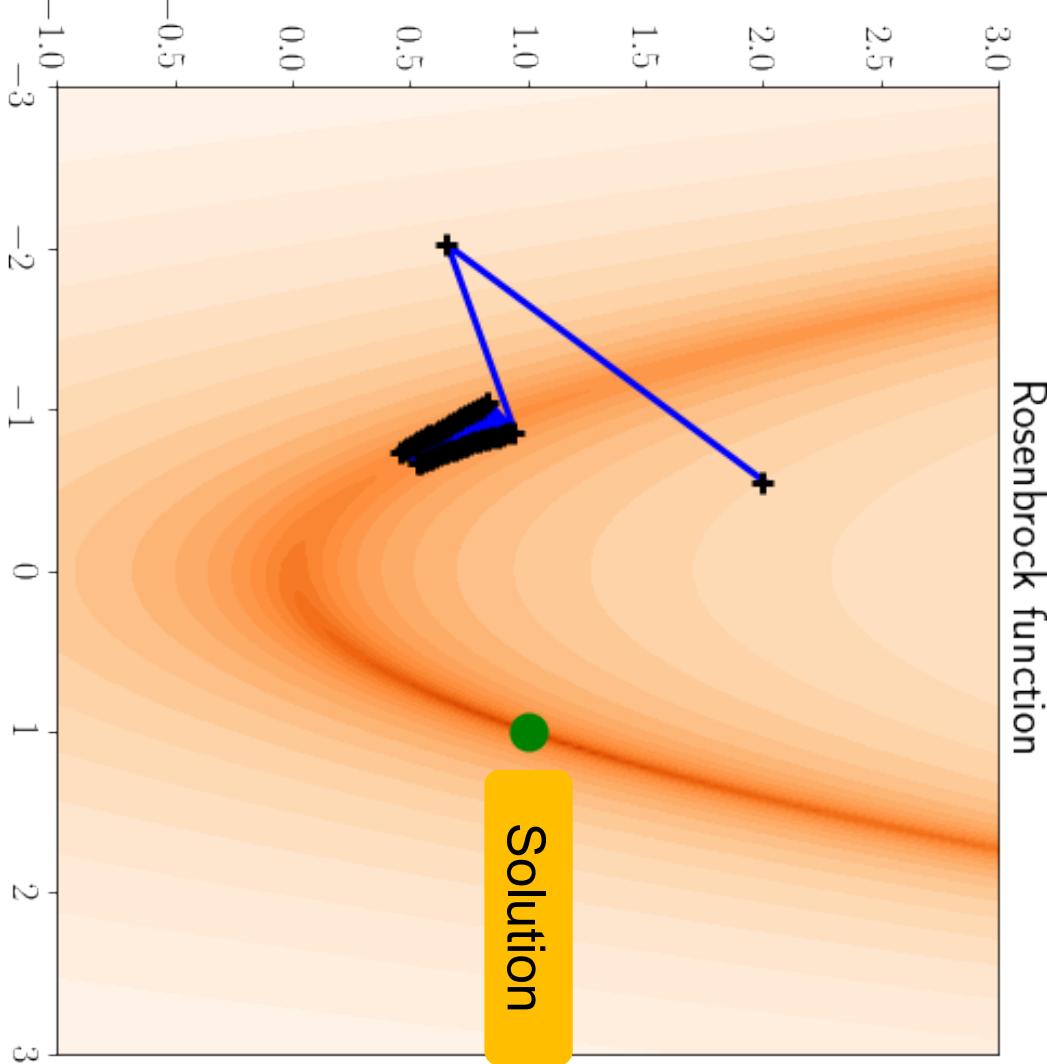
Max local rate

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

subject to $\|d\| = 1$

GD is the “steepest descent”

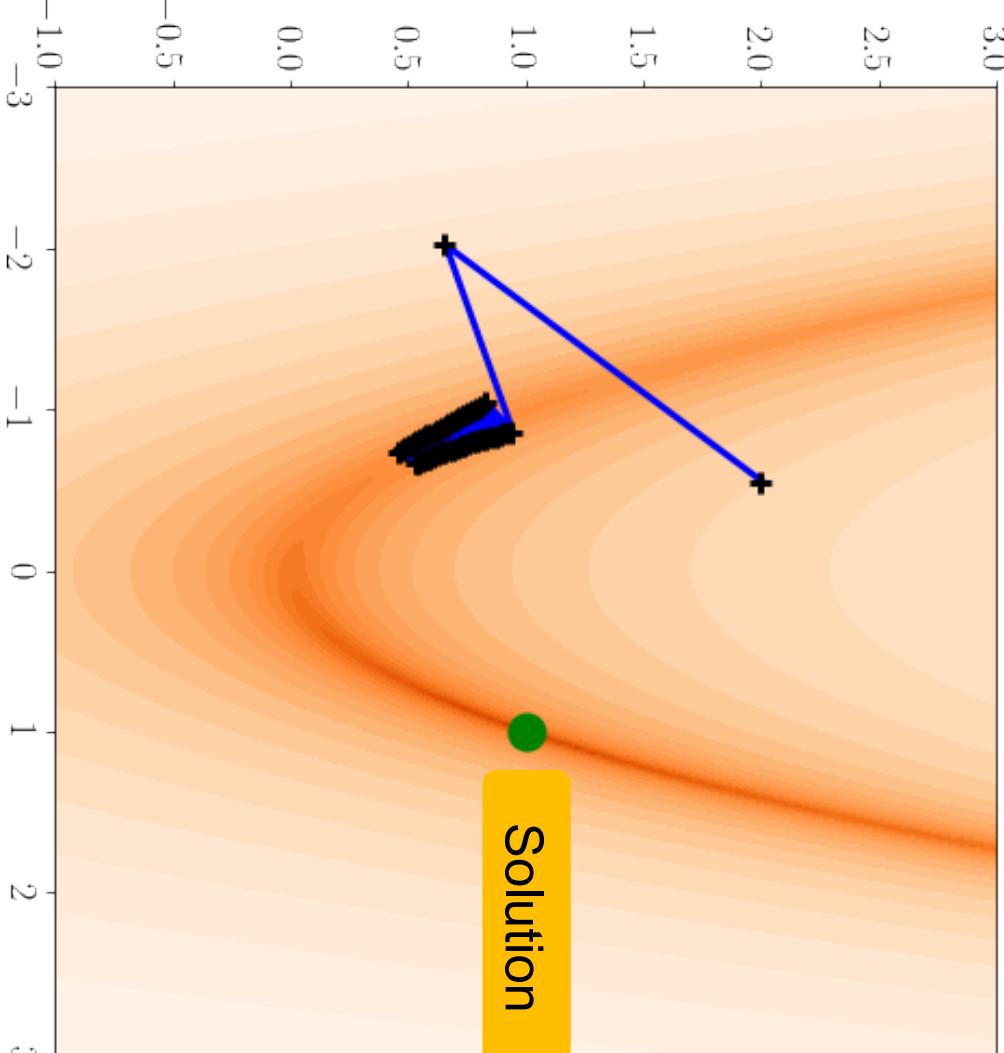
Local motivation not good for global



Get's stuck in "flat" valleys

Local motivation not good for global

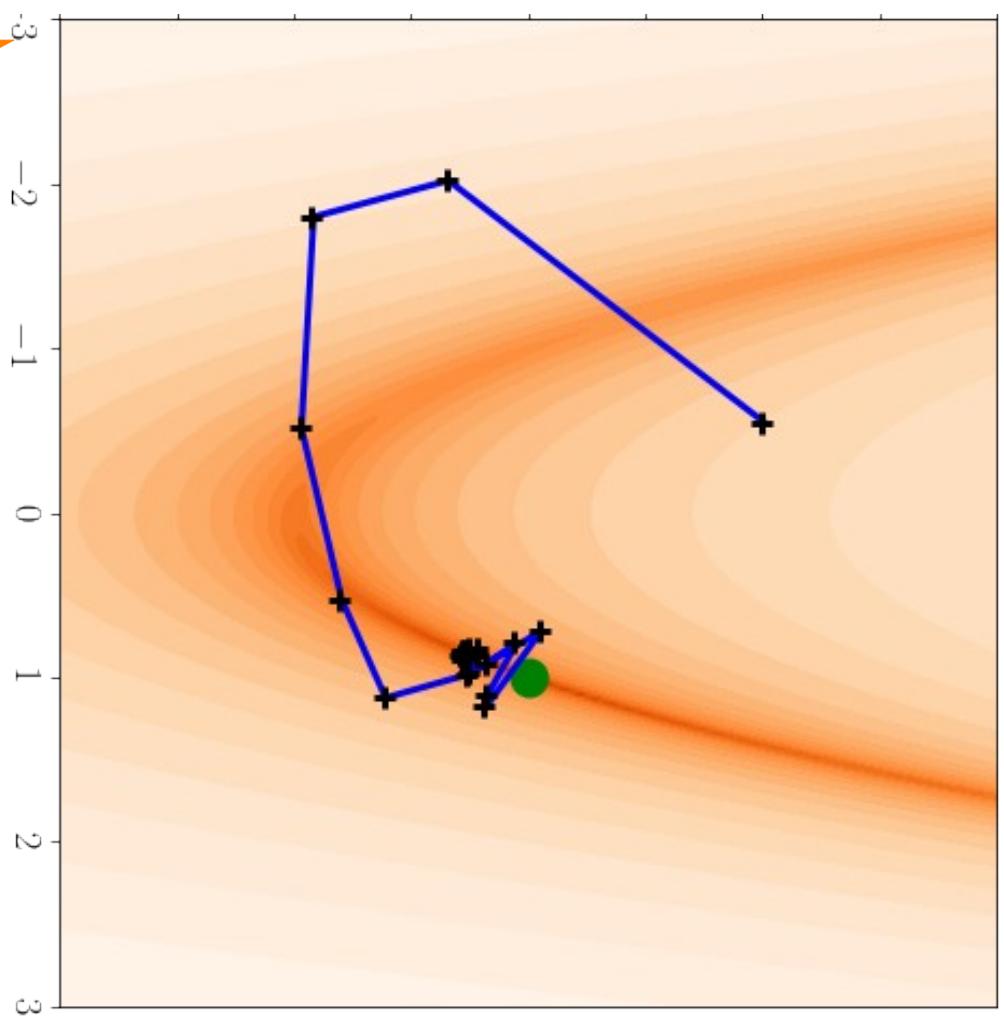
Rosenbrock function



Get's stuck in "flat" valleys

Solution

Rosenbrock function



Give momentum to keep going

Adding Momentum to GD

Heavy Ball Method:

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

Additional momentum parameter ≈ 0.99

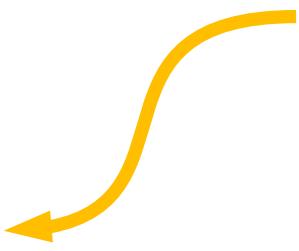
Adds “Inertia” to update, like friction for a heavy ball

Equivalent Momentum formulation

Heavey Ball Method:

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

Adds “Inertia” to update



Equivalent Momentum formulation

Heavey Ball Method:

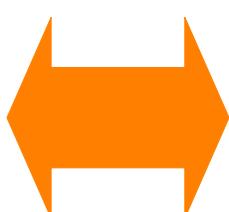
$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1})$$

Adds “Momentum”
to update

GD with momentum (GDm):

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

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



Adds “Inertia” to update

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:

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

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

Equivalent Momentum formulation

GD with momentum:

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

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

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

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

Equivalent Momentum formulation

GD with momentum:

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

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

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

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

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

$$(w^t - w^{t-1})$$

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

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

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

Equivalent Momentum formulation

GD with momentum:

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

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

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

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

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:

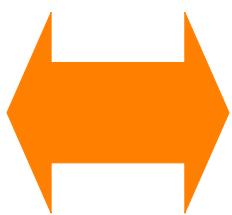
$$w^{t+1} = w^t - \gamma \nabla f(w^t) + \beta(w^t - w^{t-1})$$

Adds “Inertia” to update

Equivalent Iterate Averaging formulation

Heavey Ball Method:

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



Adds “Inertia” to update

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

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

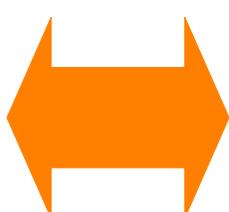
$$w^{t+1} = \frac{\alpha}{\alpha + 1} w^t + \frac{1}{\alpha + 1} z^t$$

Equivalent Iterate Averaging formulation

Heavey Ball Method:

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

Additional sequence
of variables



Adds “Inertia” to update

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

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

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

New parameters

Averaging of
variables



Equivalent Iterate Averaging formulation

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

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

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

Equivalent Iterate Averaging formulation

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

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

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 $\eta > 0, \alpha \in [0, 1]$

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

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

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

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 $\beta = \frac{\alpha}{\alpha + 1}$

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

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

$$t \leftarrow t - 1$$

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 $\beta = \frac{\alpha}{\alpha + 1}$

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

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

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

$$t \leftarrow t - 1$$

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 $\beta = \frac{\alpha}{\alpha + 1}$

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

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

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

$t \leftarrow t - 1$

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

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 $\beta = \frac{\alpha}{\alpha + 1}$

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

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

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

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

$t \leftarrow t - 1$

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

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 $\beta = \frac{\alpha}{\alpha + 1}$

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

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

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

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

Heavy Ball Method:

$t \leftarrow t - 1$

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

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

stepsize

If $\gamma = \frac{2}{L + \mu}$ 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

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

stepsize

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

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

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

Corollary

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

Convergence of Gradient Descent with Momentum



Polyak 1964

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

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

stepsize

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

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

stepsize

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

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

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

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

Convergence of Gradient Descent with Momentum

Momentum

 Polyak 1964

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

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

stepsize

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

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

Optimal iteration complexity
for this function class

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

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

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

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

$$w^s := w^* + s(w^t - w^*)$$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

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

$$w^s := w^* + s(w^t - w^*)$$

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

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

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

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

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

$$w^s := w^* + s(w^t - w^*)$$

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

$$+ w^* - w^*$$



Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

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

$$w^s := w^* + s(w^t - w^*)$$

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

$$+ w^* - w^*$$

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

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

$$=: A_s$$

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

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

$$w^s := w^* + s(w^t - w^*)$$

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

Proof: Convergence of Heavy Ball. Two time steps

Fundamental Theorem of Calculus

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

$$w^s := w^* + s(w^t - w^*)$$

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

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

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

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

Proof: Convergence of Heavy Ball

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

$$\begin{aligned} z^{t+1} &= \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_s(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix} \\ &= \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix} \end{aligned}$$

Proof: Convergence of Heavy Ball

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

$$\begin{aligned} z^{t+1} &= \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_s(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix} \\ &= \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix} \\ &= \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} z^t \end{aligned}$$

Proof: Convergence of Heavy Ball

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

$$\begin{aligned} z^{t+1} &= \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_s(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix} \\ &= \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix} \\ &= \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} z^t \end{aligned}$$

Simple recurrence!

Proof: Convergence of Heavy Ball

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

$$\begin{aligned} z^{t+1} &= \begin{bmatrix} w^{t+1} - w^* \\ w^t - w^* \end{bmatrix} = \begin{bmatrix} A_s(w^t - w^*) - \beta(w^{t-1} - w^*) \\ w^t - w^* \end{bmatrix} \\ &= \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} \begin{bmatrix} w^t - w^* \\ w^{t-1} - w^* \end{bmatrix} \\ &= \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} z^t \end{aligned}$$

Simple recurrence!

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

Proof: Convergence of Heavy Ball

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

$$\|A\| := \max_{i=1,\dots,2n} |\lambda_i(A)|$$

Proof: Convergence of Heavy Ball

$$\|z^{t+1}\| \leq \left\| \begin{bmatrix} A_s & -I\beta \\ I & 0 \end{bmatrix} \right\| \|z^t\|$$
$$\|A\| := \max_{i=1,\dots,2n} |\lambda_i(A)|$$

EXE on Eigenvalues:

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

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

Proof: Convergence of Heavy Ball

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

$$\|A\| := \max_{i=1,\dots,2n} |\lambda_i(A)|$$

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

EXE on Eigenvalues:

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

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

Part V: Momentum with SGD

Adding Momentum to SGD



Rumelhart, Hinton,
Geoffrey, Ronald,
1986, Nature

Stochastic Heavy 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:

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

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

Sampled i.i.d

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

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

SGDm and Averaging

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

SGDm and Averaging

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

$m^0 = 0$



SGDm and Averaging

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

$m^0 = 0$

Momentum as exponentiated average:

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

SGDm and Averaging

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

$m^0 = 0$

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

SGDm and Averaging

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

$m^0 = 0$

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

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

SGDm and Averaging

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

$m^0 = 0$

Momentum as exponentiated average:

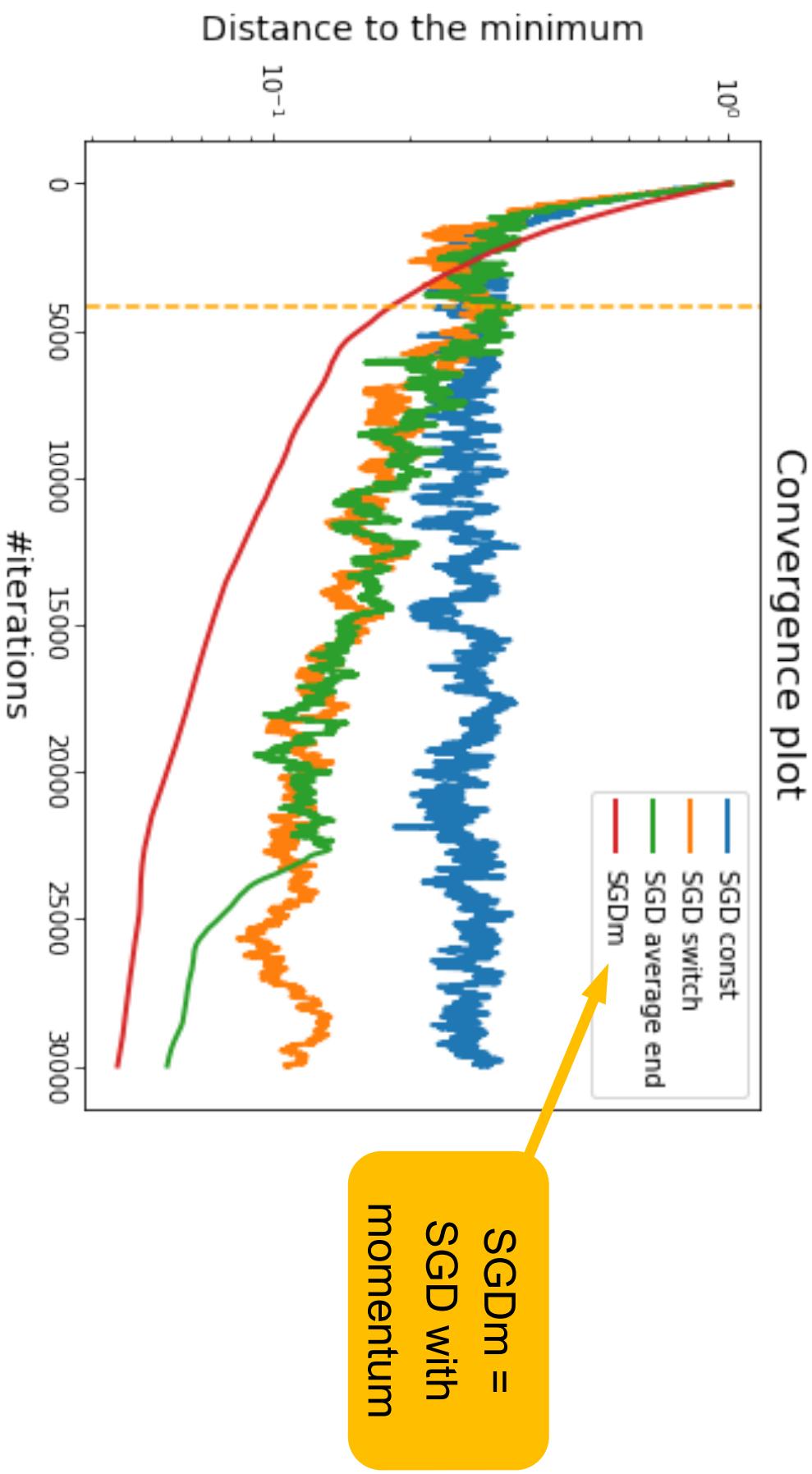
$$w^{t+1} = w^t - \gamma \sum_{i=1}^t \beta^i \nabla f_{j_{t-i}}(w^{t-i})$$

Acts like an approximate variance reduction since

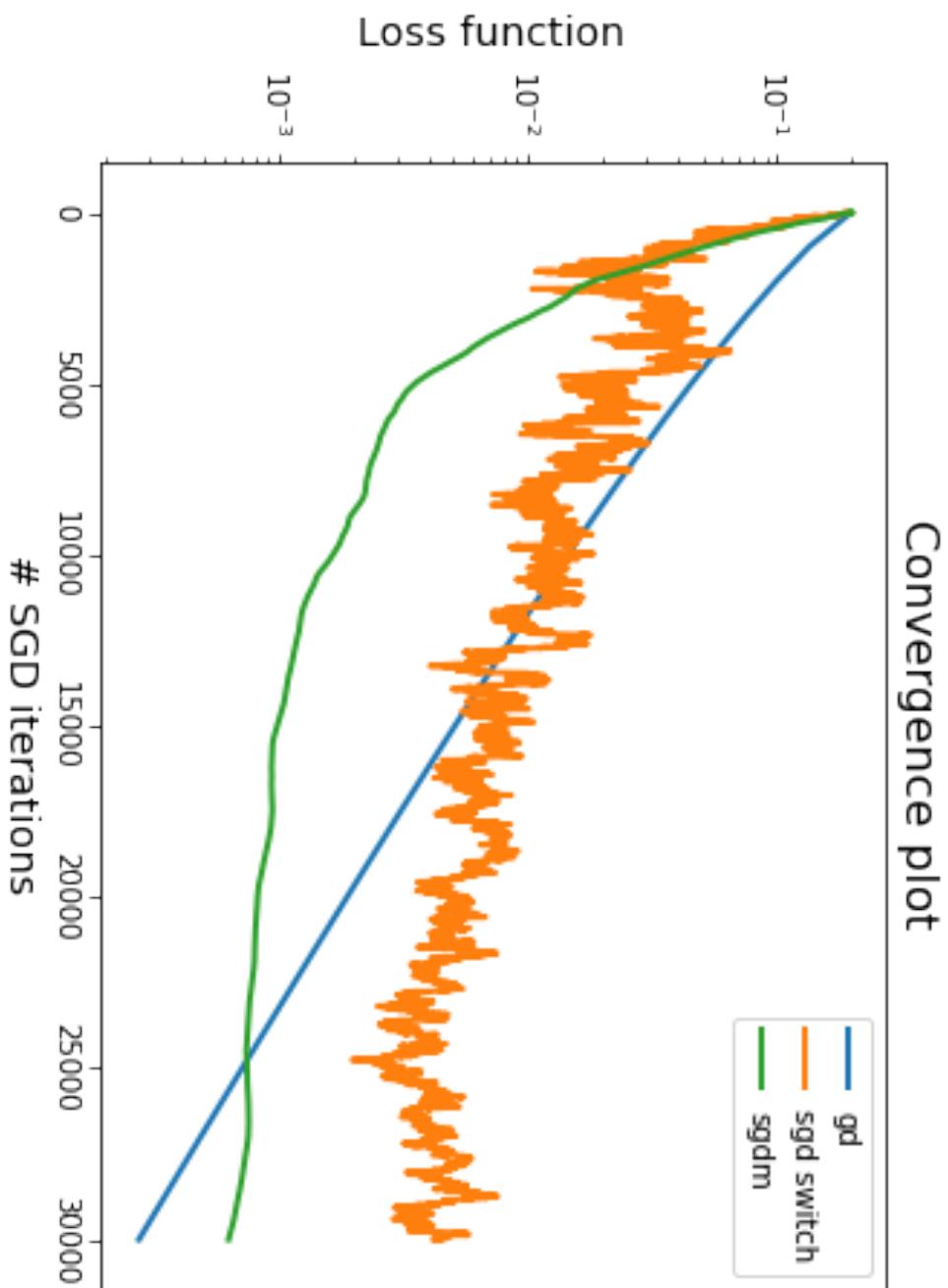
This is why momentum works well with SGD

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

Stochastic Gradient Descent with momentum



Stochastic Gradient Descent with momentum vs GD



Can we prove momentum always works?



Difficult: Recent 2019 results only

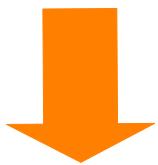
Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?

Not clear, recently same rate as SGD + averaging

Convergence of Gradient Descent with Momentum

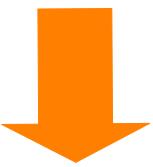
Does momentum make SGD converge faster?



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

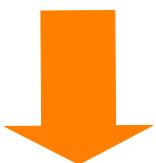
f is μ -strongly convex,
 f_i is convex and L_i -smooth



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

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

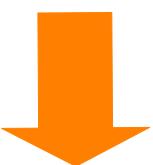


$$t \geq O\left(\frac{1}{\epsilon}\right)$$

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

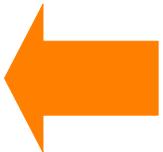
Convergence of Gradient Descent with Momentum

Does momentum make SGD converge faster?



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

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



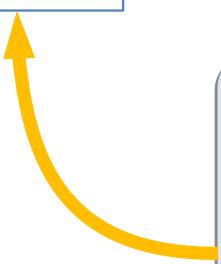
Not clear, recently same rate as SGD + averaging

f_i is convex and L_i -smooth

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

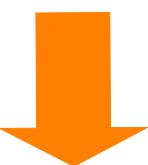


Sebbouth, Defazio,
RMG, online soon,
2020



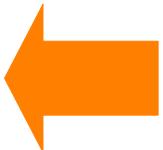
Convergence of Gradient Descent with Momentum

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



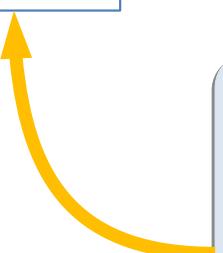
Results use iterate averaging
to crack the proof!

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

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



Sebbouth, Defazio,
RMG, online soon,
2020



Part V: Test error and validation

Generalization

We have been solving:

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

Generalization

We have been solving:

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

But we already know these labels

Generalization

We have been solving:

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

But we already know these labels

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

Generalization

We have been solving:

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

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

But we already know these labels

We would like to solve:

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

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

Validation Error

$$\begin{aligned} X &:= \begin{bmatrix} x_1 & x_2 & \dots & x_T & x_{T+1} & \dots & x_n \end{bmatrix} \in \mathbb{R}^{d \times n} \\ y &:= \begin{bmatrix} y_1 & y_2 & \dots & y_T & y_{T+1} & \dots & y_n \end{bmatrix} \in \mathbb{R}^n \end{aligned}$$

Validation Error

192

$$X := \begin{bmatrix} x_1 & x_2 & \dots & x_T \\ y_1 & y_2 & \dots & y_T \end{bmatrix} \in \mathbb{R}^{d \times n}$$

$$y := \begin{bmatrix} x_{T+1} & \dots & x_n \\ y_{T+1} & \dots & y_n \end{bmatrix} \in \mathbb{R}^n$$

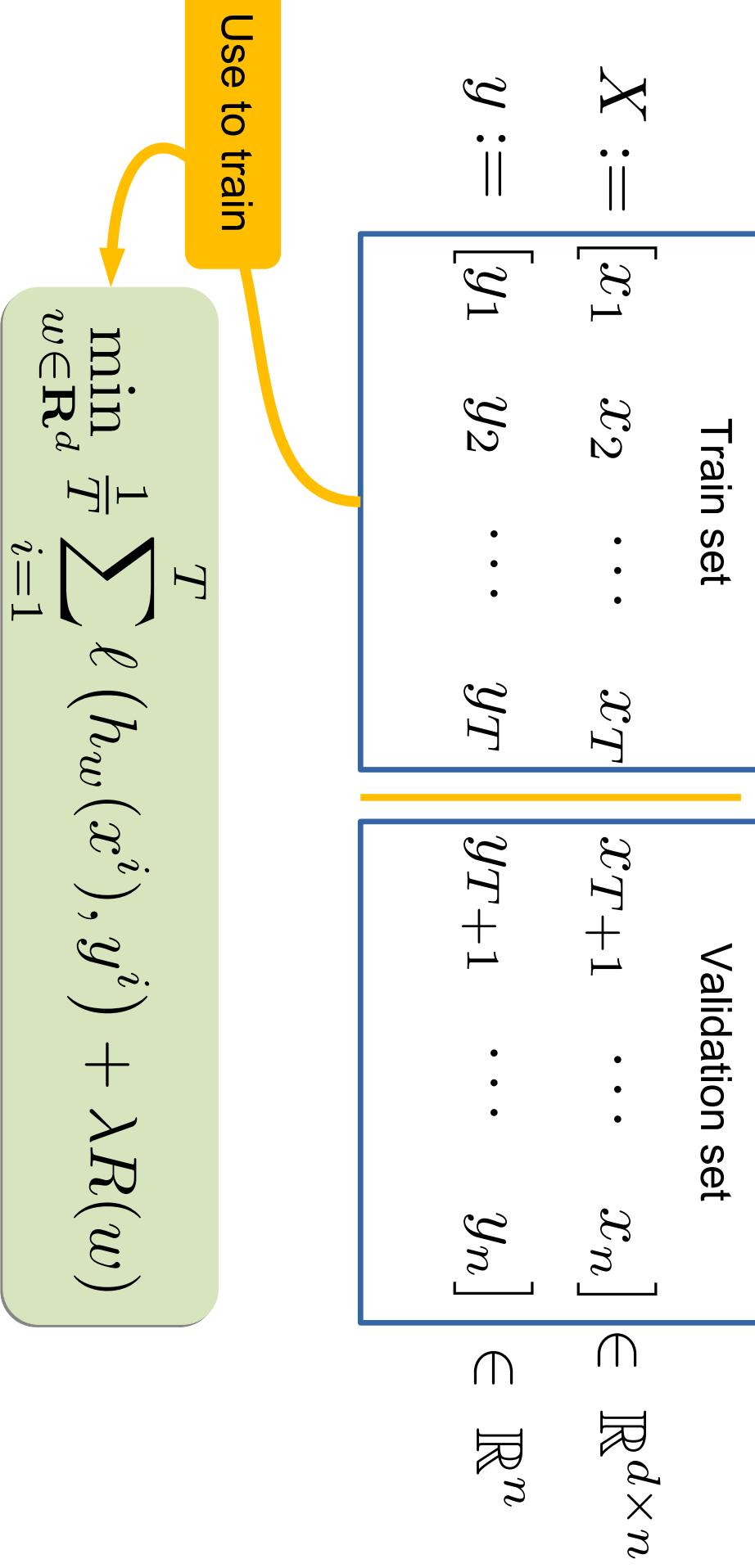
Validation Error

$$\begin{aligned} X &:= \begin{bmatrix} x_1 & x_2 & \cdots & x_T \end{bmatrix} \in \mathbb{R}^{d \times n} \\ y &:= \begin{bmatrix} y_1 & y_2 & \cdots & y_T \\ y_{T+1} & \cdots & & y_n \end{bmatrix} \in \mathbb{R}^n \end{aligned}$$

Train set Validation set

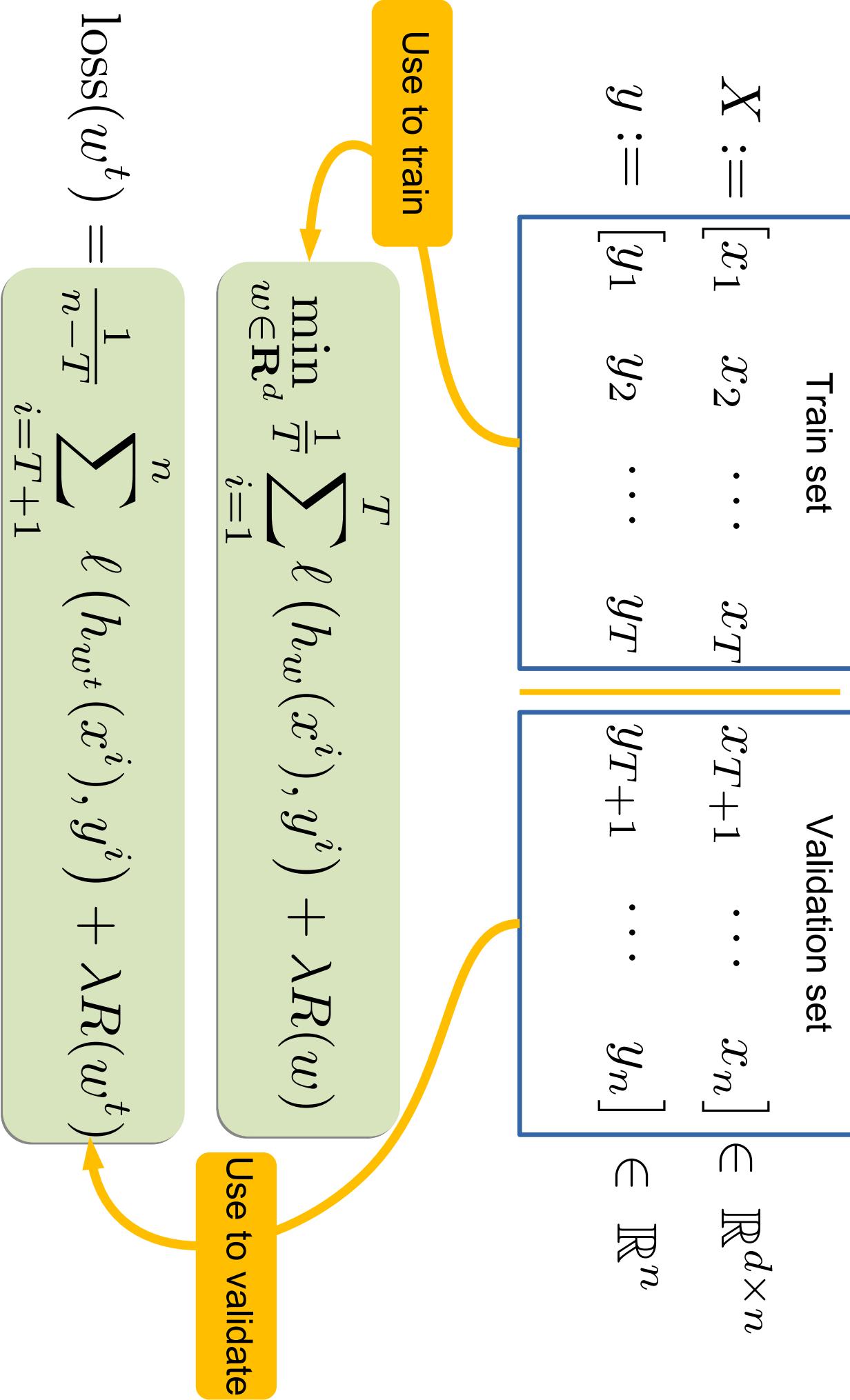
Validation Error

194

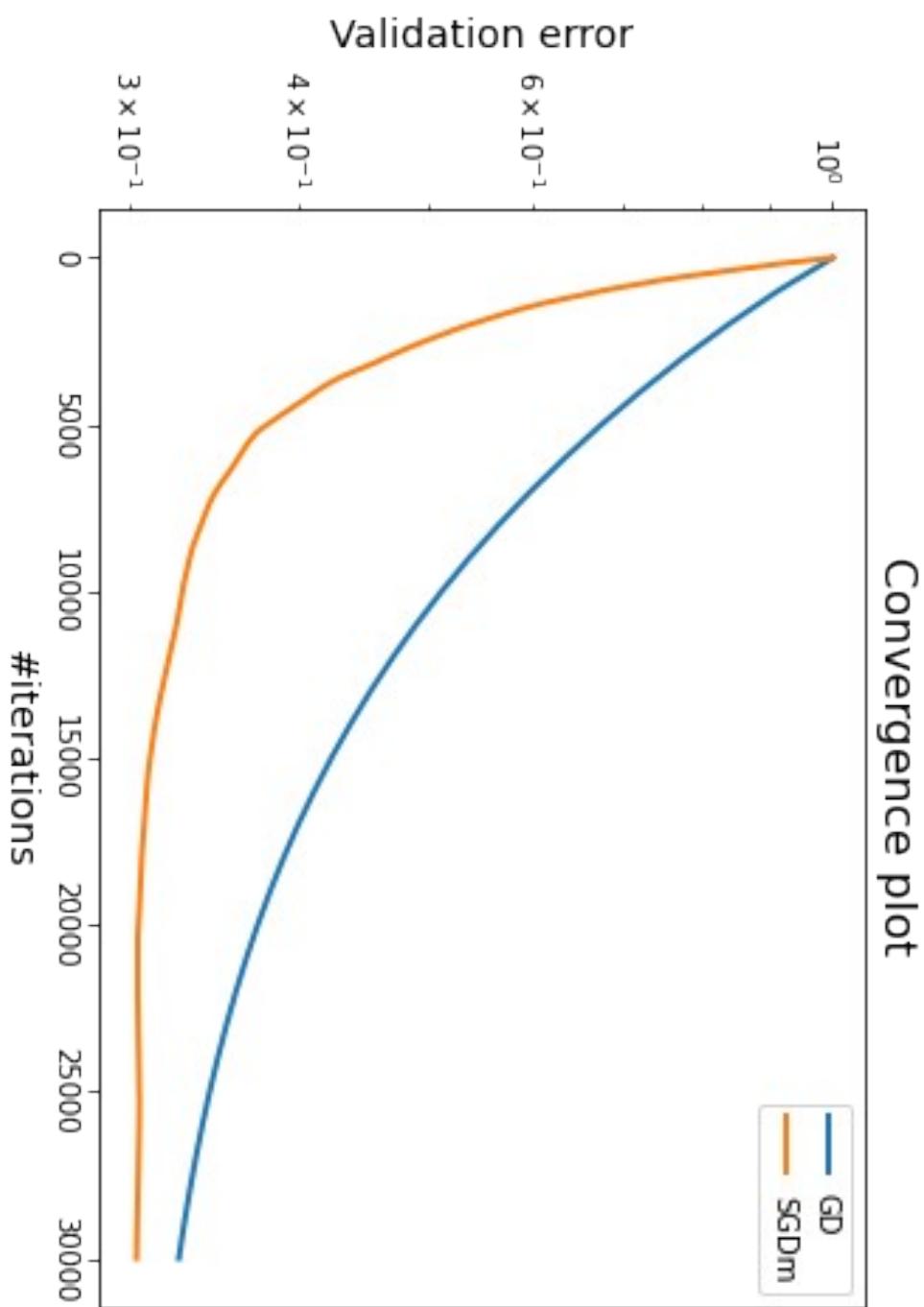


Validation Error

195



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:

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

But we want to solve:

But we already know these labels

The statistical learning problem:

Minimize the expected loss over an *unknown* expectation

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

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

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

SGD for learning

- Set $w^0 = 0$, $\alpha_t > 0$
- for $t = 0, 1, 2, \dots, T - 1$
- sample $(x, y) \sim \mathcal{D}$
- $w^{t+1} = w^t - \alpha_t \nabla \ell(h_{w^t}(x), y)$
- Output $\bar{w}^T = \frac{1}{T} \sum_{t=1}^T w^t$