NYU

Introduction to Data Science

Center for Data Science
Iddo Drori, Spring 2019

NYU

Agenda

- Supervised learning: classification, fitting
- Logistic regression
- Support vector machines
- Overfitting, bias and variance
- Regularization
- Cross validation

Optional

- Ensemble models
- Optimization

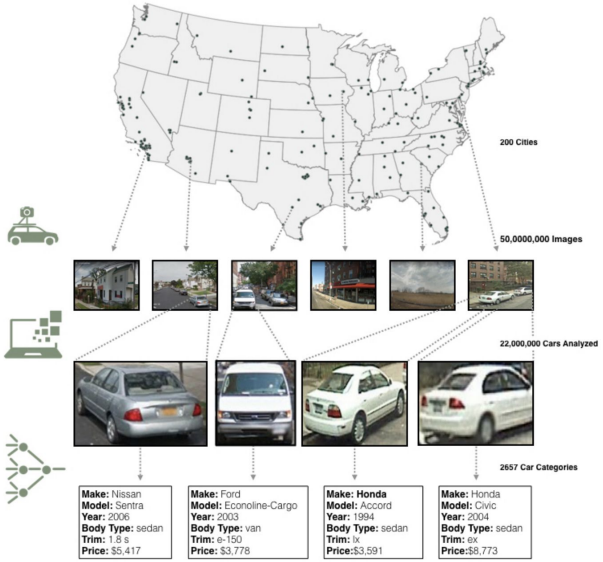
NYU

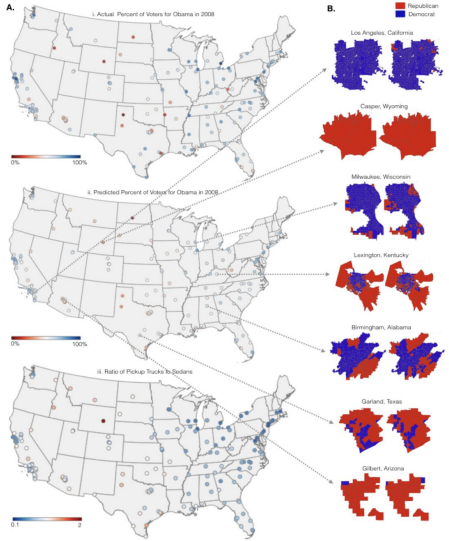
Introduction

NYU Classification and Regression Example in Data Science Project

Source: Using deep learning and Google Street View to estimate the demographic makeup of the US, Gebru et al, PNAS, 2018.

- Determine make, model, year of all vehicles in neighborhoods, enumerate 22M automobiles (8% of US)
- Accurately estimate income, race, education, and voting patterns
- If # sedans encountered during 15-minute drive through city > # pickup trucks then city likely to vote for Democrat during next Presidential election (88%); otherwise, likely to vote Republican (82%)





Machine Learning

- **Supervised**
- Unsupervised
- Reinforcement

Supervised Learning

- Given $D = \{x^{(i)}, y^{(i)}\}$ learn a model or function f: $x^{(k)}$ -> $y^{(k)}$

Training and Test Datasets

- Ideally from same distribution
- Disjoint training and test datasets

```
def intersection(a, b):
    return not set(a).isdisjoint(b)
```

Generalization

- Training data is sample from population
- We would like our model to generalize well to unseen test data drawn from same population
- Generalization error is difference between empirical loss and expected loss.

$$G(f(X,W)) = \frac{1}{m}\sum_i L(y^i, f(x^i, W)) - E_{(X,Y)\sim P}(L(y^i, f(x^i, W)))$$

Task Types and Sub-Types

Task type	Task sub-type
classification	binary, multi-class
regression	univariate, multivariate

Notation

Example i $x^{(i)}$

Label i $y^{(i)}$

Feature j x_j

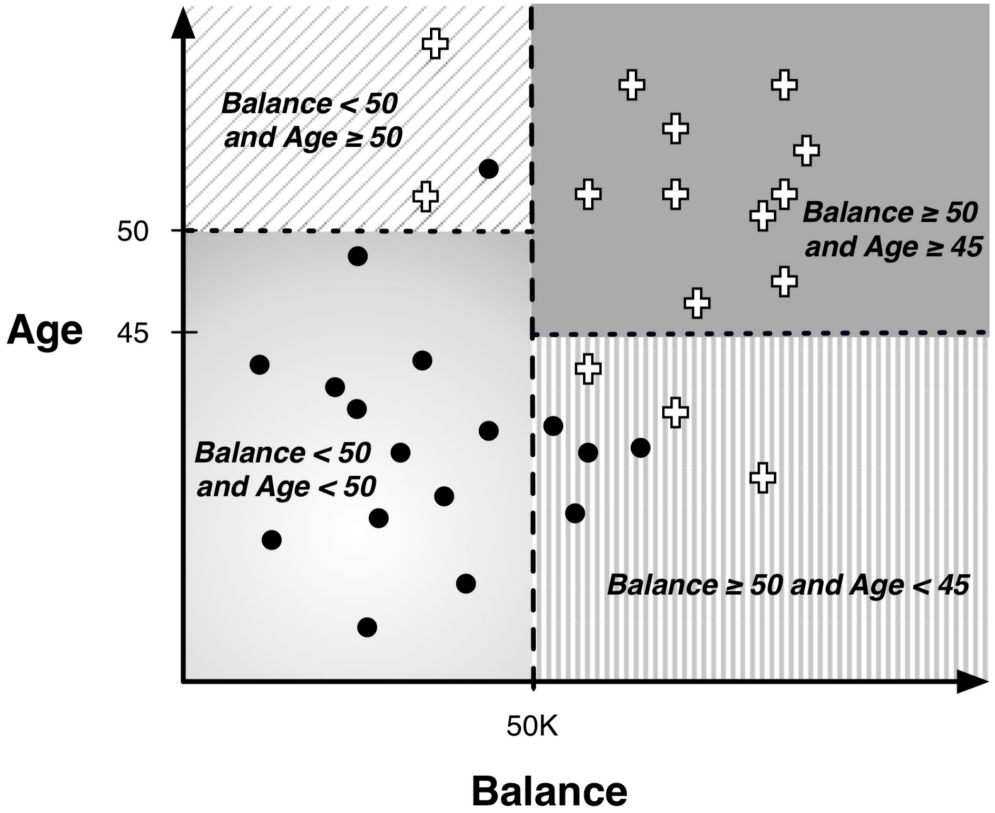
Supervised Learning

data -> model, parameters

data, model -> parameters

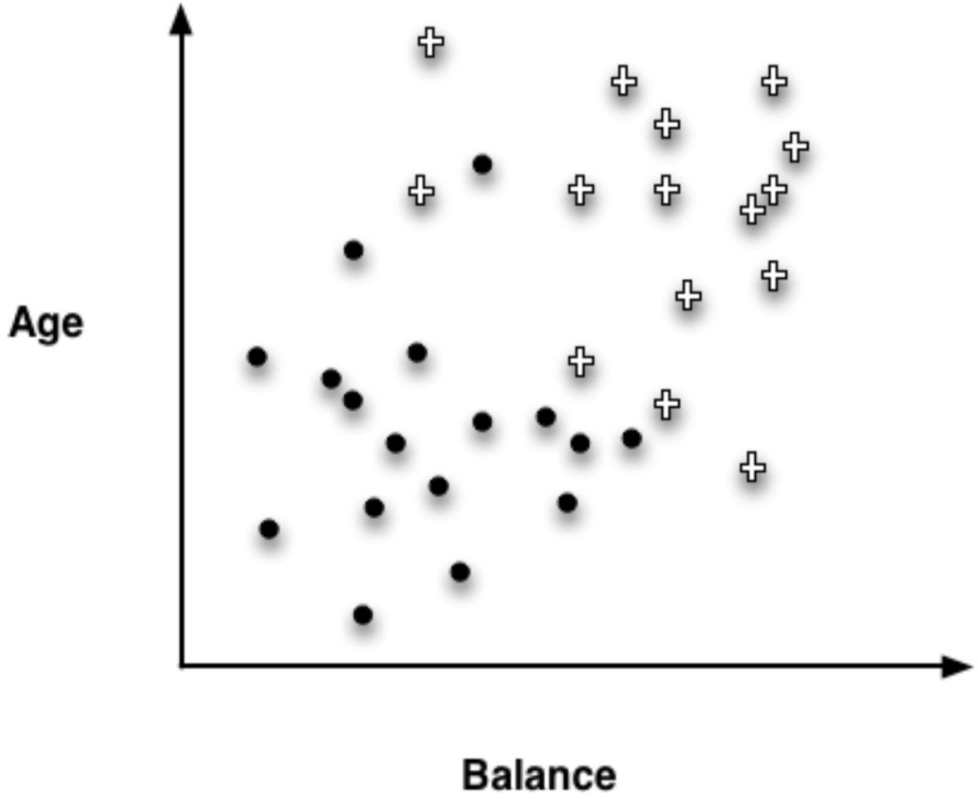
Decision Tree

perpendicular
recursive
explainable



Source: DSB

Data Points



Source: DSB

Linear Classifier

$$f(x) = w^T x$$

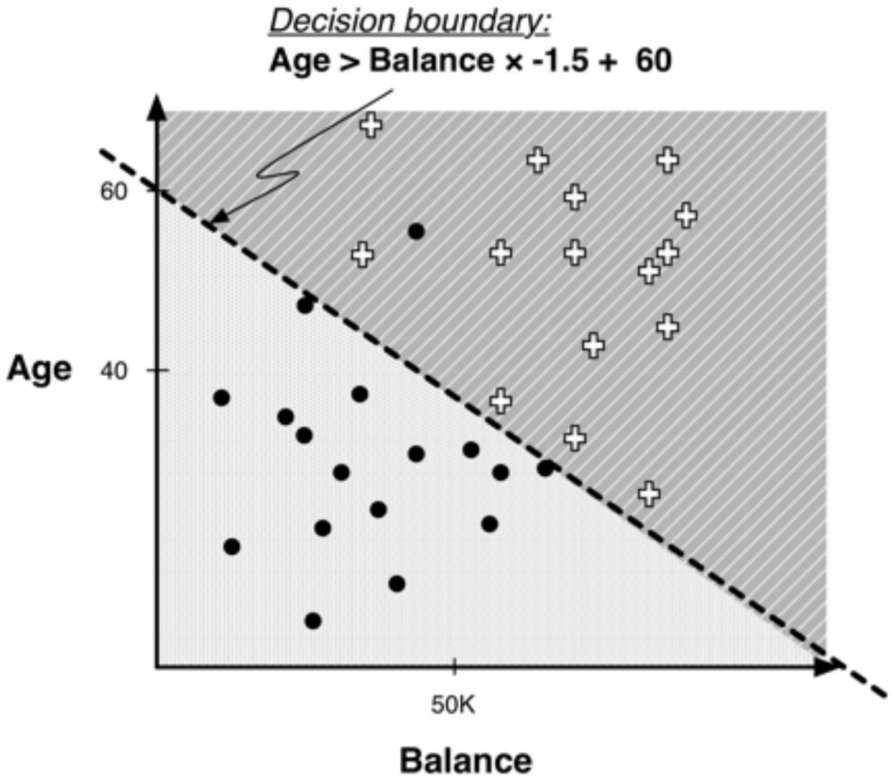
$$class(x) = \begin{cases} 1 & f(x) > 0 \\ 0 & otherwise \end{cases}$$

angle

once

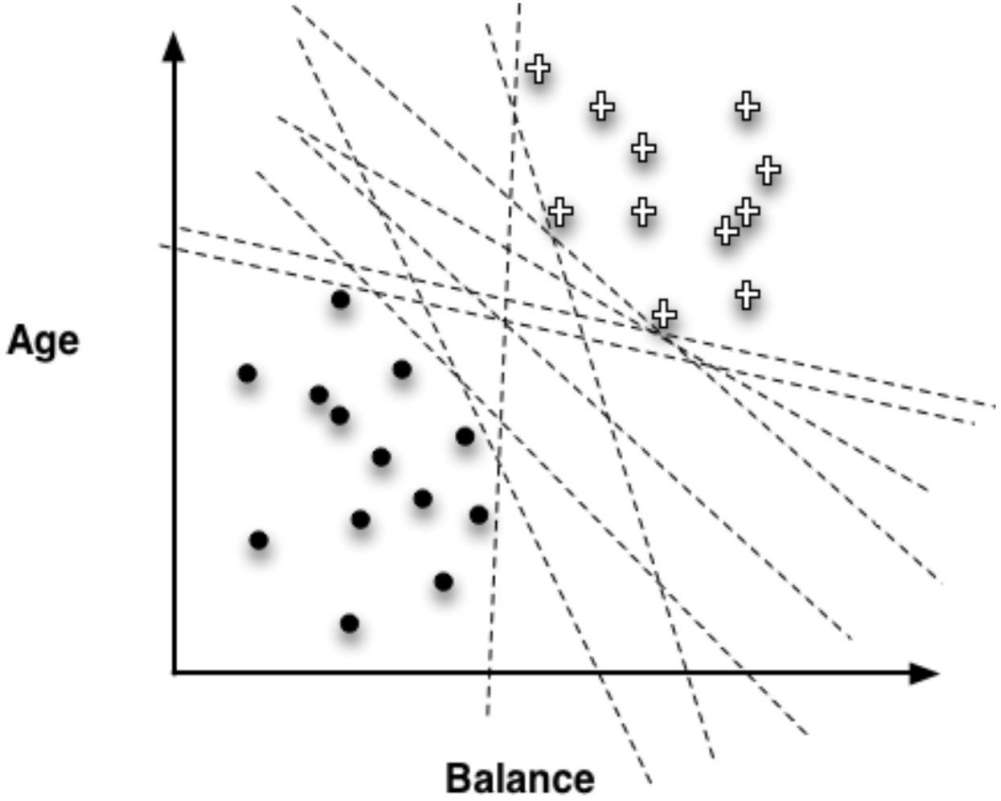
w measures importance of features

explainable?



Source: DSB

Linear Classifiers

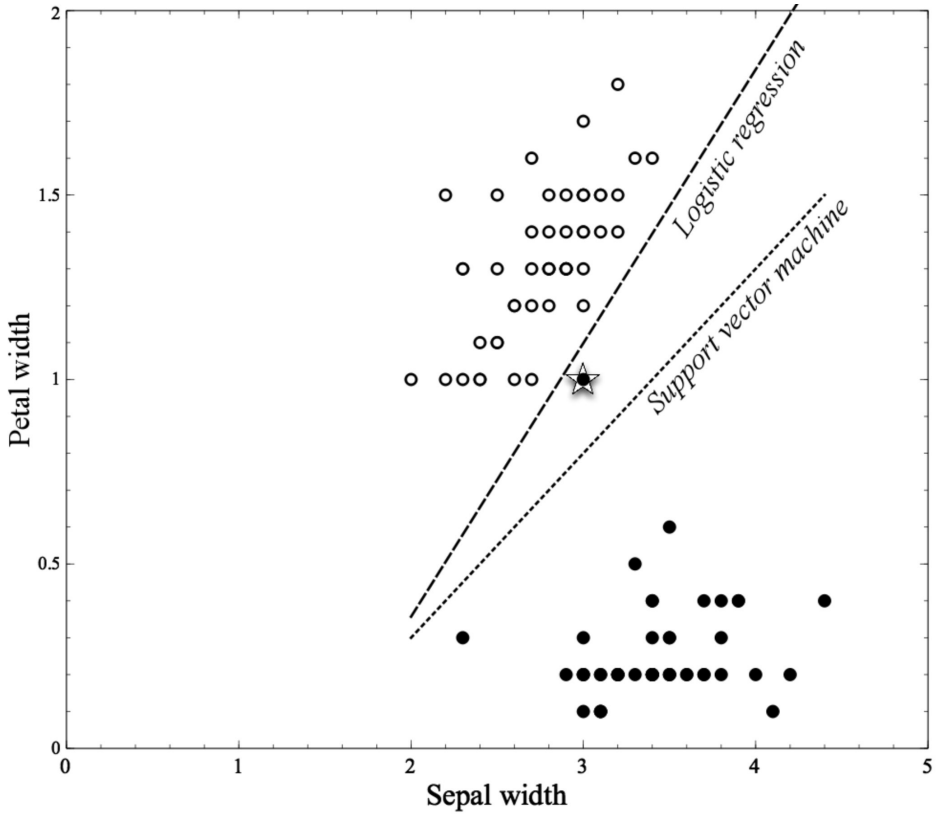


Source: DSB

Logistic Regression and Support Vector Machine

fitting linear model to data

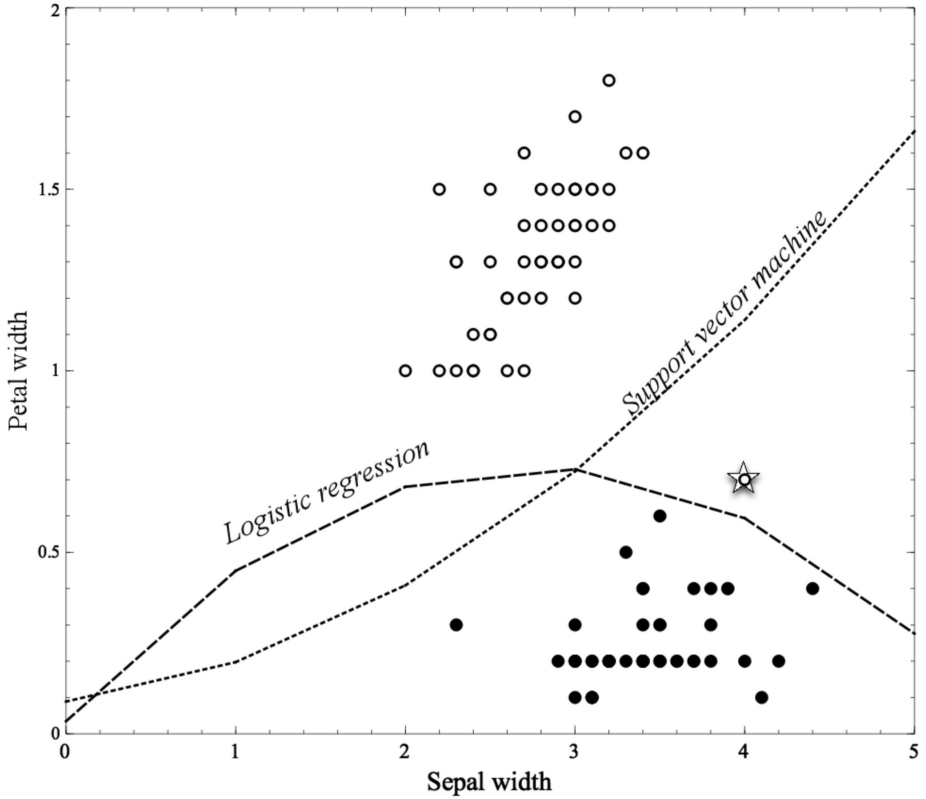
using different objective functions



Source: DSB

Using Non-Linear Features

linear models using non-linear feature create non-linear model boundary



Source: DSB

Cost Function

Compare predicted output with ground truth label

For a single example $\mathcal{L}(y, f(x, W))$

Average loss over all examples $\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(y^{(i)}, \hat{y}^{(i)})$ $\hat{y}^{(i)} = f(x^{(i)}, W)$

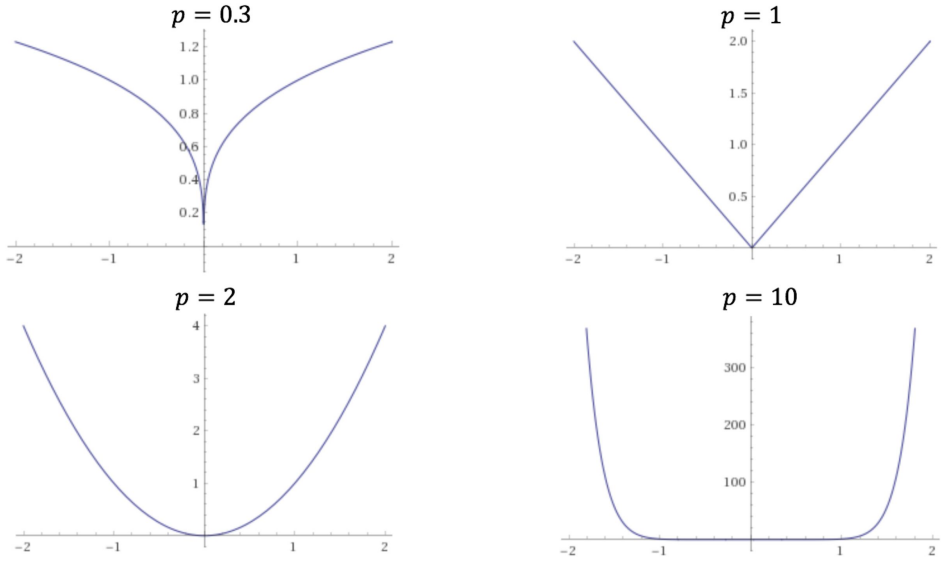
Optimization $\underset{W}{\text{minimize}}\frac{1}{m}\sum_{i=1}^{m}\mathcal{L}\left(y^{(i)}, f(x^{(i)}, W)\right)$

Regularization term may be added to prefer simple models, avoid overfitting.

Loss Function

Input: ground truth and prediction
Output: measure of their difference

For example:



$$\mathcal{L}(y, a) = |y - a|^p$$

NYU

Logistic Regression

Logistic Regression

- Applies linear model to class probability estimation
- Does not do regression

Classification: Logistic Regression

Probability $p(x)$

Odds *if $p(x)=½$ then odds are 50:50 or 1; if $p(x)=0.9$ then odds are 90:10 or 9*

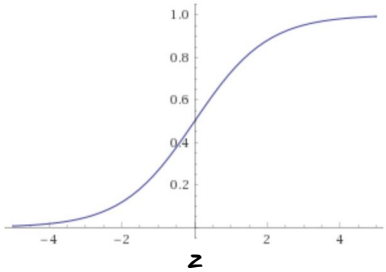
Log-odds *$\log(p(x)/(1-p(x)))$; if $p(x)=½$ then log-odds are 0; if $p(x)=0.9$ then log-odds are 2.19*

Set log-odds to be a linear classifier *$\log(g(x)/(1-g(x))) = \theta^T x$*

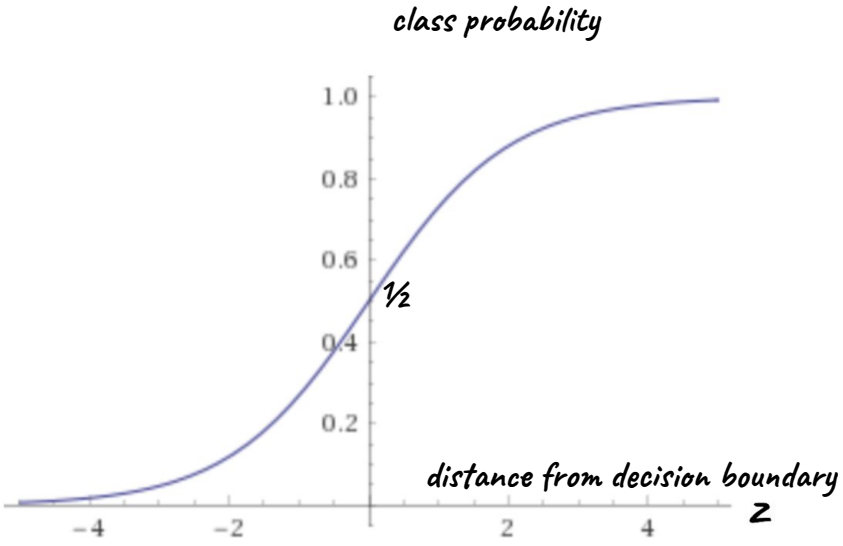
Result is sigmoid function $g(z) = \frac{1}{1+e^{-z}}$

Logistic regression $f_\theta(x) = g(\theta^T x)$

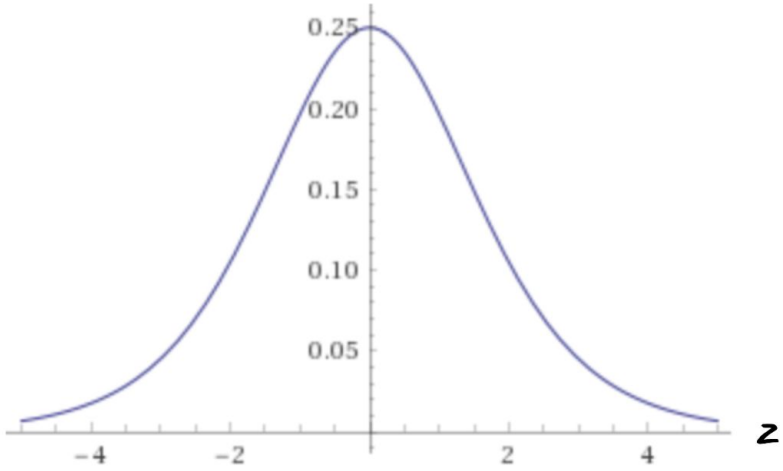
$g(z)$



Sigmoid



$$g(z) = \frac{1}{1 + e^{-z}}$$



$$g'(z) = g(z)\big(1 - g(z)\big)$$

Logistic Regression

Prediction $a = f_\theta(x) = g(\theta^T x)$

Loss function between ground truth label and prediction $\mathcal{L}(y, a) = \mathcal{L}(y, f_\theta(x))$

Log-loss $-y \log(a) - (1-y)\log(1-a)$

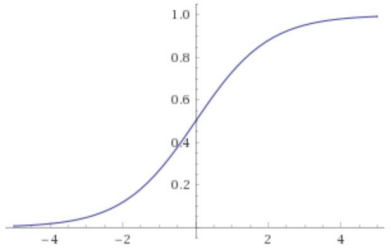
Cost: average over all examples $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}\left(y^{(i)}, f_\theta\left(x^{(i)}\right)\right)$

Find parameter to minimize cost

Logistic Regression Classification

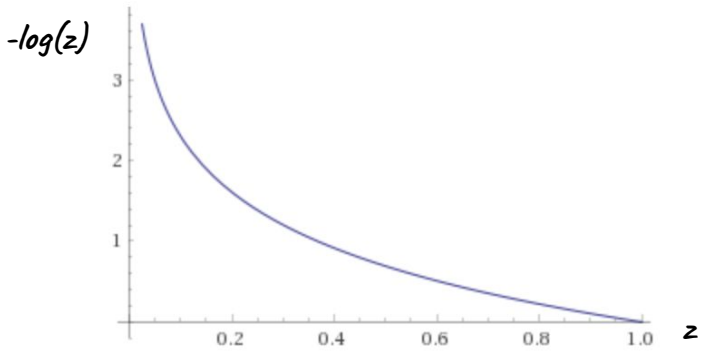
Binary classification

$$h_w(x) = \begin{cases} 1 & w^T x \geq 0 \\ 0 & otherwise \end{cases}$$



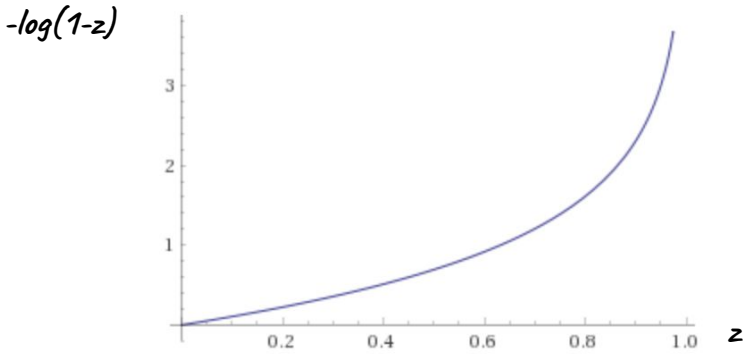
Logistic Regression Loss

y=1



if y=1 and z=1 then loss=0
if y=1 and z=0 then loss=inf

y=0



if y=0 and z=0 then loss=0
if y=0 and z=1 then loss=inf

Logistic Regression Loss

Binary classification $h_w(x) = \begin{cases} 1 & w^T x \geq 0 \\ 0 & otherwise \end{cases}$

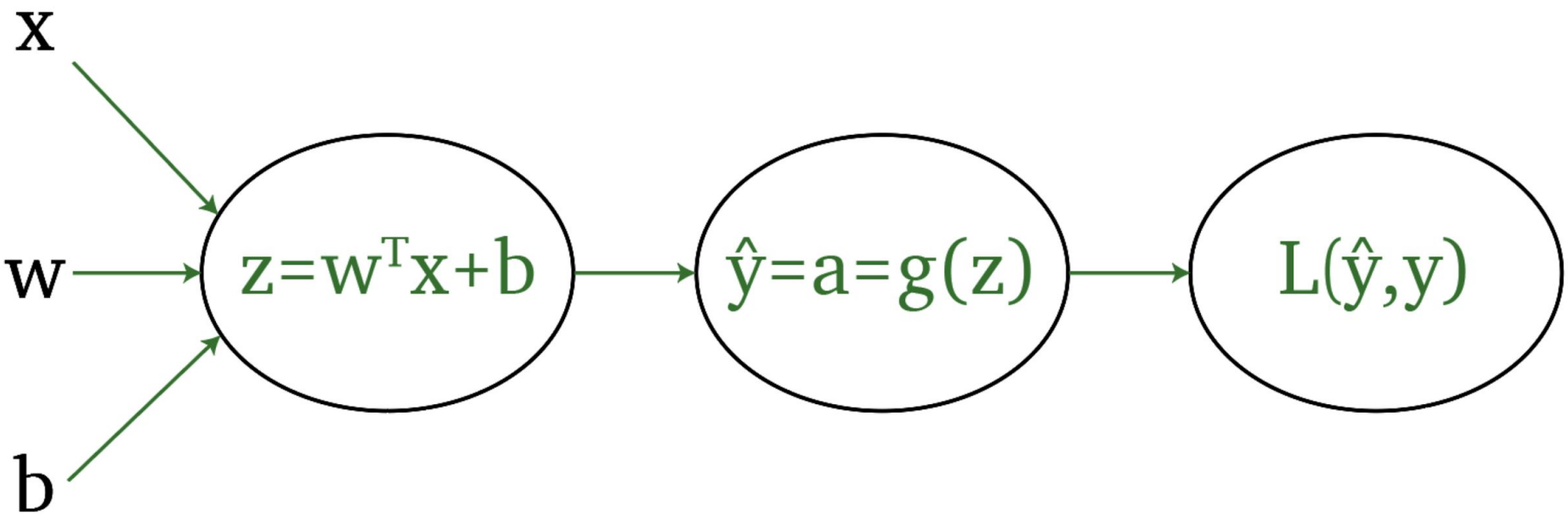
$$-y \log \frac{1}{1+e^{-w^T x}} - (1-y) \log \left(1 - \frac{1}{1+e^{-w^T x}}\right)$$

Logistic Regression Cost

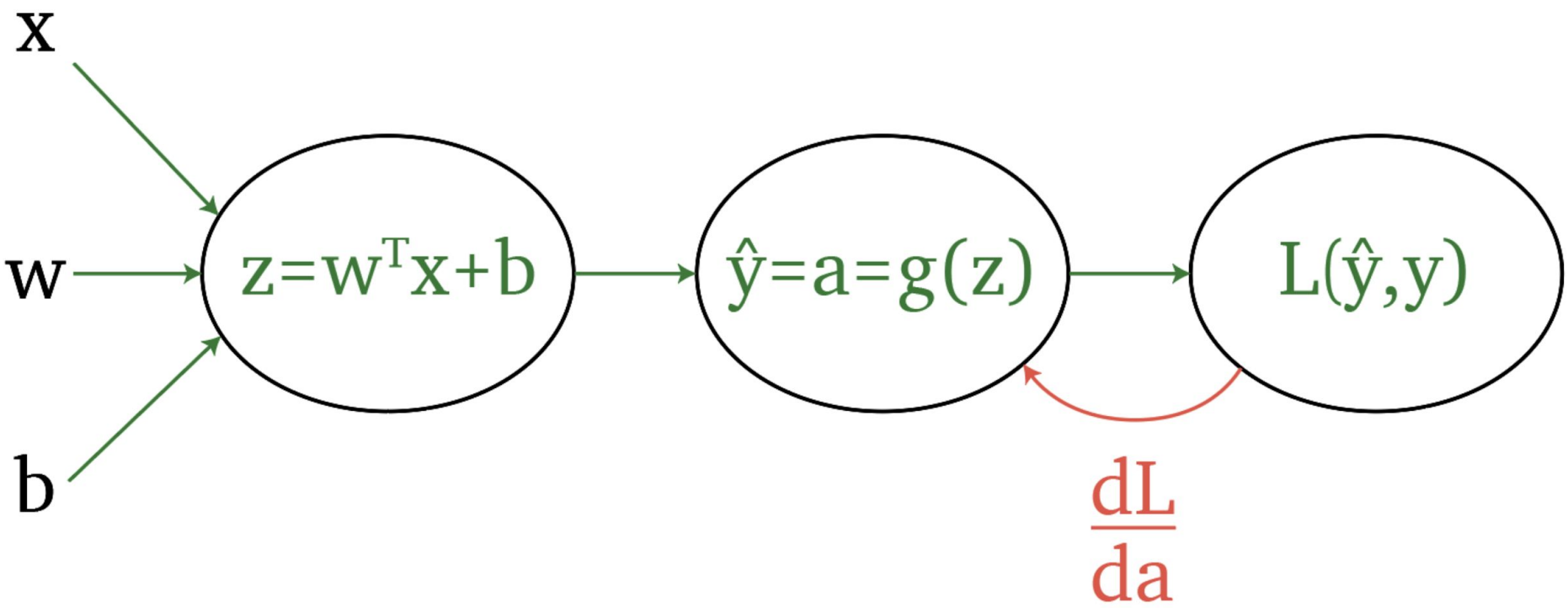
Binary classification

$$\underset{w}{\text{minimize}} \frac{1}{m}\left(\sum_{i=1}^{m} y^{(i)}\left(-\log h\left(w^T x^{(i)}\right)\right) + \left(1-y^{(i)}\right)\left(1-\log\left(1-h\left(w^T x^{(i)}\right)\right)\right)\right) + \frac{\lambda}{2m}\|w\|$$

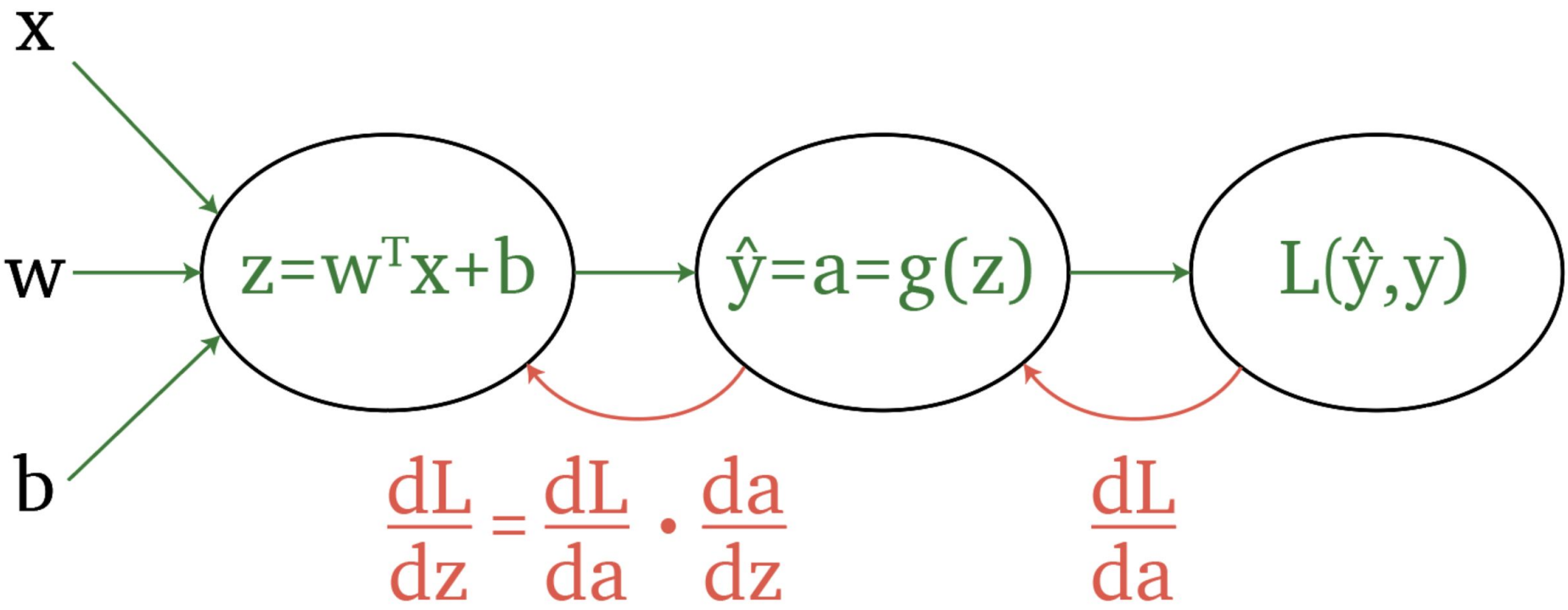
Logistic Regression



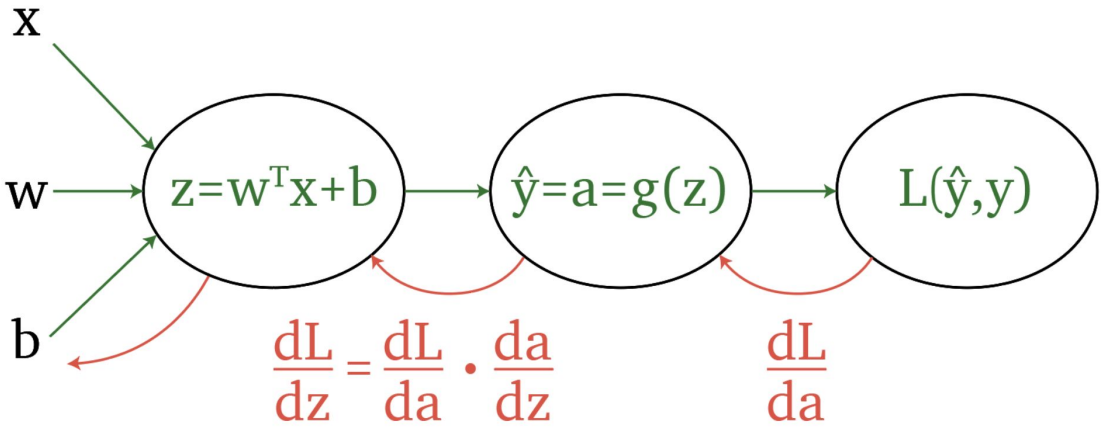
Logistic Regression



Logistic Regression



Logistic Regression



$$\frac{dL}{dw} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{dw} = \frac{dL}{dz} \cdot x$$

$$\frac{dL}{db} = \frac{dL}{da} \cdot \frac{da}{dz} \cdot \frac{dz}{db} = \frac{dL}{dz}$$

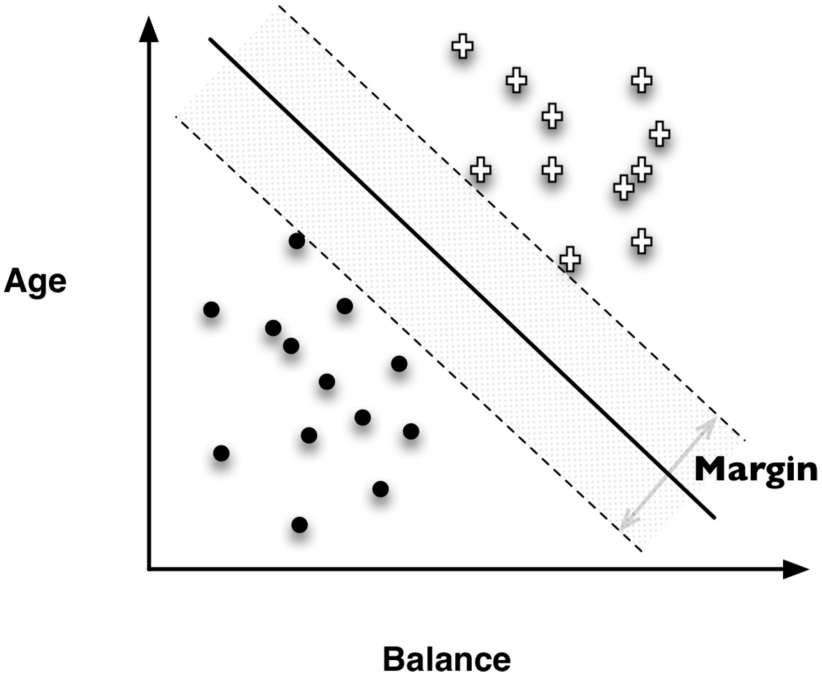
NYU

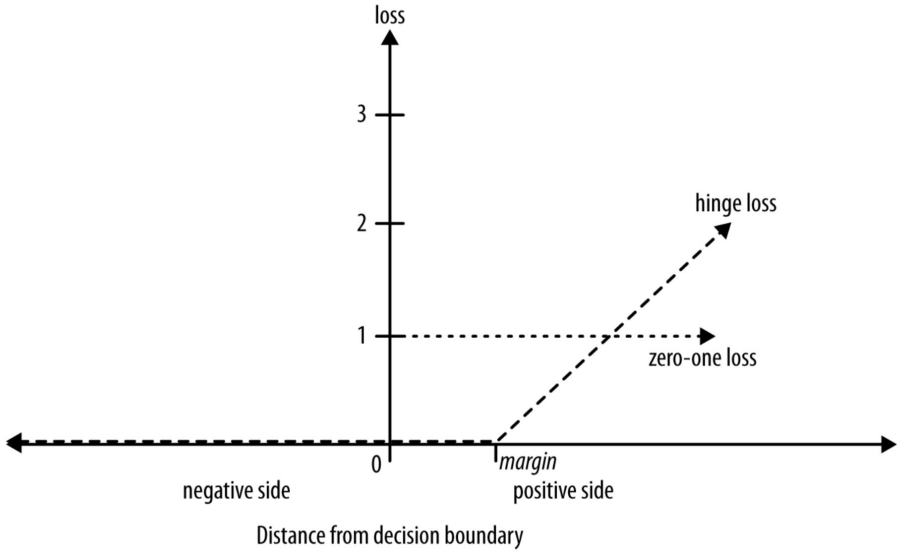
Support Vector Machine

Support Vector Machine

- Linear discriminant, binary classification
- Instead of separating by line, separate by widest bar
- Penalty for misclassification is proportional to distance from boundary

Support Vector Machine





Source: DSB

Optimal Margin Classifier

Classifier

$$h(x) = sign(w^T x - b)$$

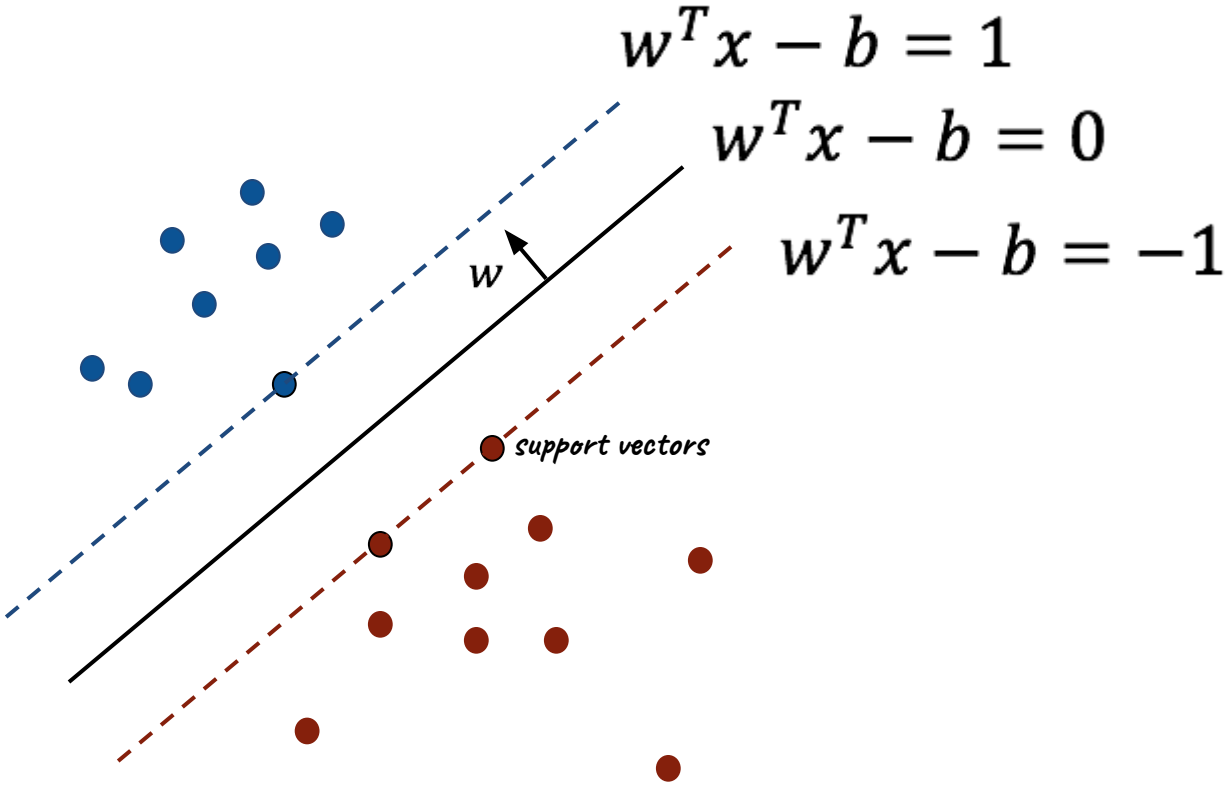
which is the solution to

$$\underset{w}{\text{minimize}} \frac{1}{2}\|w\|_2^2$$

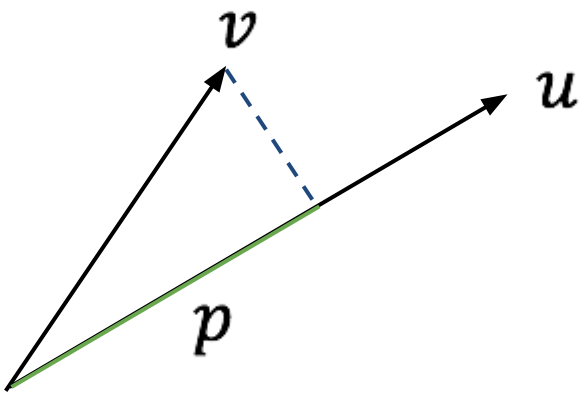
such that

$$y^{(i)}\left(w^T x^{(i)} - b\right) \geq 1$$

Optimal Margin Classifier



Optimal Margin Classifier

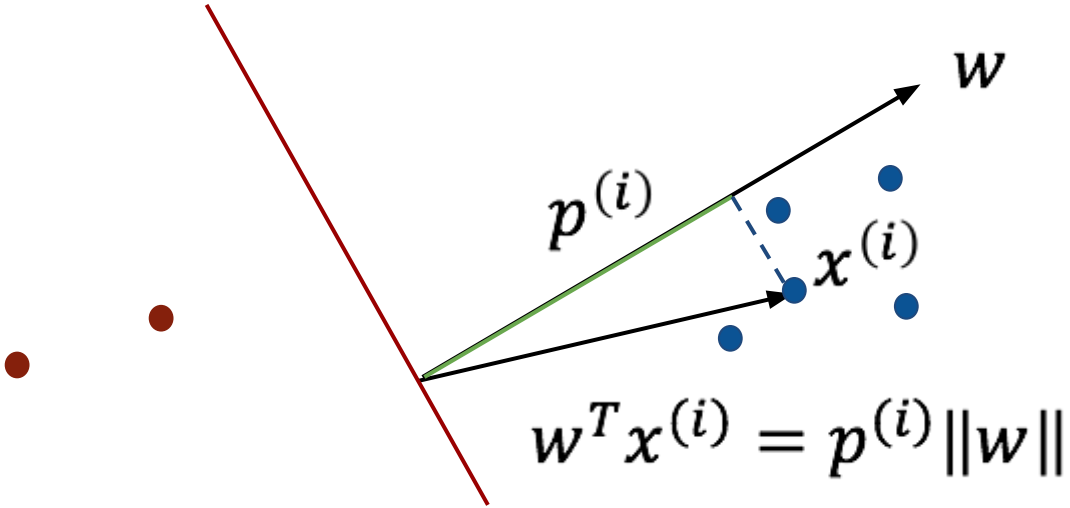


$$u^T v = p\|u\| = p\sqrt{u_1^2 + u_2^2}$$

Optimal Margin Classifier

$$\underset{w}{\text{minimize}} \frac{1}{2}\|w\|_2^2$$

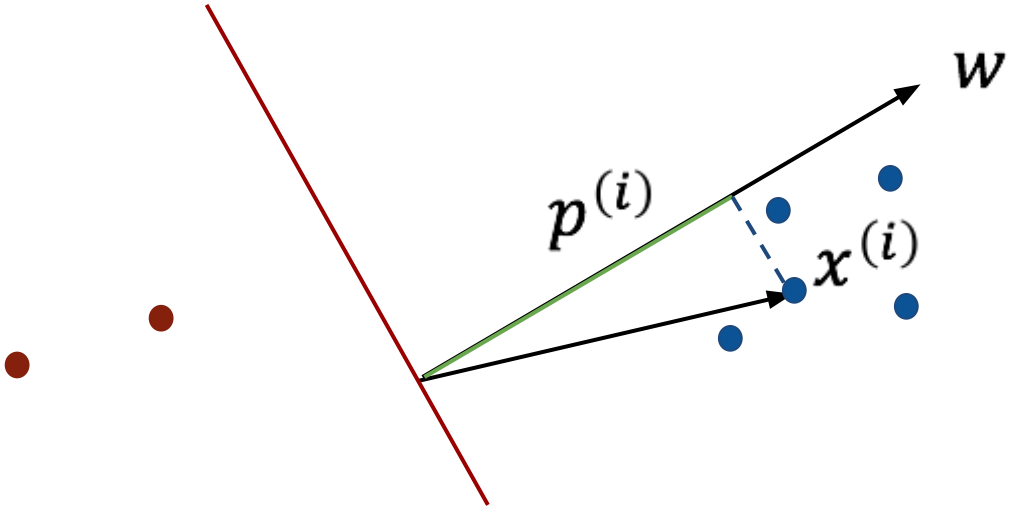
$$w^T x^{(i)} \geq 1 \quad if \quad y^{(i)} = 1 \qquad w^T x^{(i)} \leq -1 \quad if \quad y^{(i)} = 0$$



Optimal Margin Classifier

$$\underset{w}{\text{minimize}} \frac{1}{2}\|w\|_2^2$$

$$p^{(i)}\|w\| \geq 1 \quad if \quad y^{(i)} = 1 \qquad p^{(i)}\|w\| \leq -1 \quad if \quad y^{(i)} = 0$$



Hinge Loss

$$L(z, y) = (1 - yz)_+ = \max(0, 1 - yz)$$

Logistic Regression Classification

Binary classification

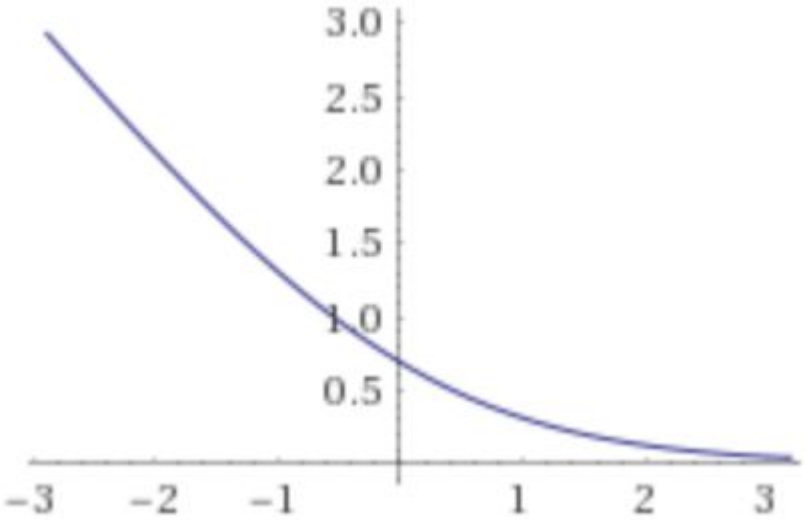
$$h_w(x) = \begin{cases} 1 & w^T x \geq 0 \\ 0 & otherwise \end{cases}$$

Logistic Regression Loss

Binary classification $h_w(x) = \begin{cases} 1 & w^T x \geq 0 \\ 0 & otherwise \end{cases}$

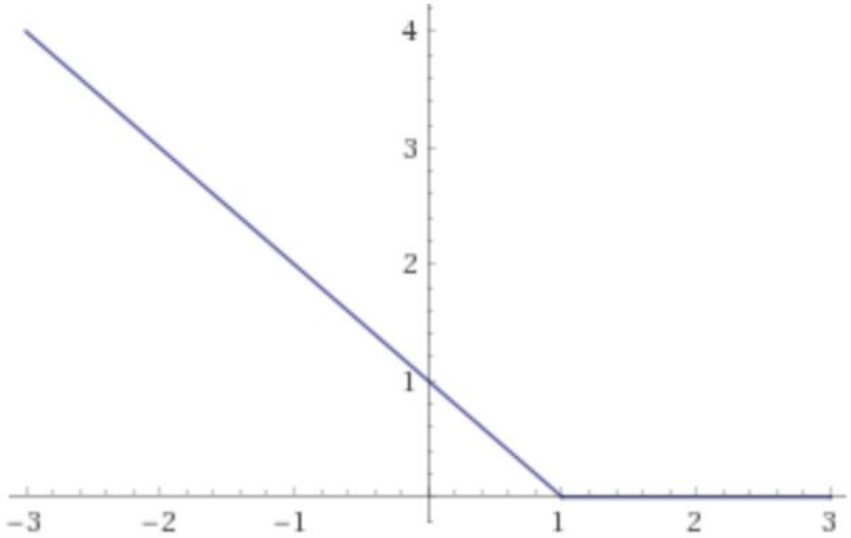
$$-y \log \frac{1}{1+e^{-w^T x}} - (1-y) \log \left(1 - \frac{1}{1+e^{-w^T x}}\right)$$

Logistic Regression Loss



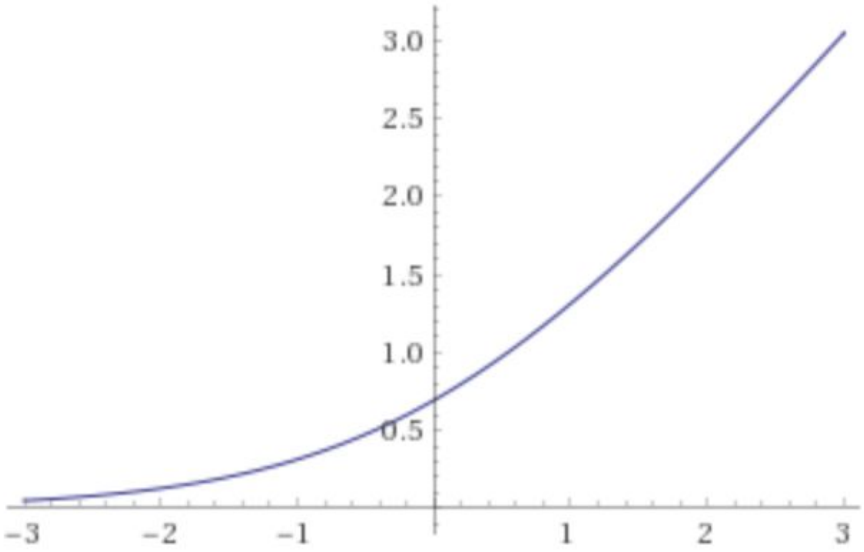
$$-\log\left(\frac{1}{1+e^{|-z|}}\right)$$

Hinge Loss



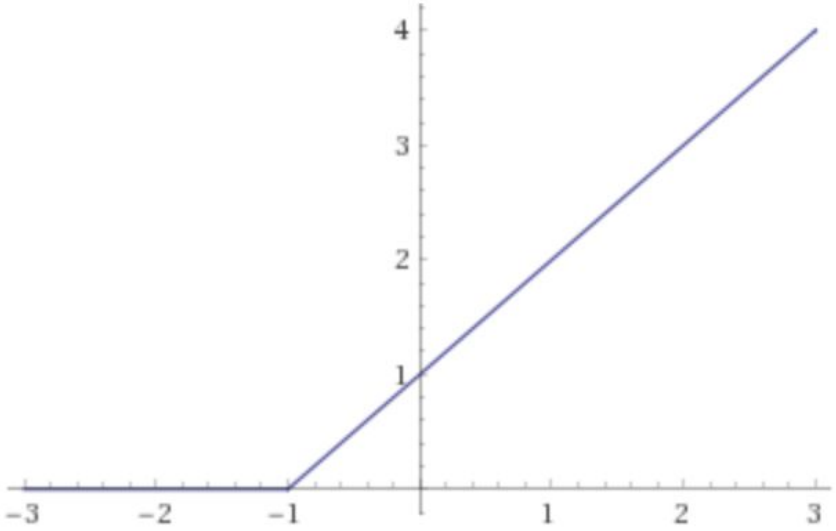
$\max(1 - z, 0)$

Logistic Regression Loss



$$-\log\left(1 - \frac{1}{1 + e^{\{-z\}}}\right)$$

Hinge Loss



$\max(z + 1, 0)$

Logistic Regression Cost

Binary classification

$$\underset{w}{\text{minimize}} \frac{1}{m}\left(\sum_{i=1}^{m} y^{(i)}\left(-\log h\left(w^T x^{(i)}\right)\right) + \left(1-y^{(i)}\right)\left(1-\log\left(1-h\left(w^T x^{(i)}\right)\right)\right)\right) + \frac{\lambda}{2m}\|w\|$$

Support Vector Machine Cost

$$\underset{w}{\text{minimize}}\ C\left(\sum_{i=1}^{m} y^{(i)}\, hinge\left(w^T x^{(i)}\right) + \left(1 - y^{(i)}\right)\left(1 - hinge\left(w^T x^{(i)}\right)\right)\right) + \frac{1}{2}\|w\|_2^2$$

$$C = \frac{1}{\lambda}$$

$$\|w\|_2^2 = w^T w$$

Support Vector Machine Classification

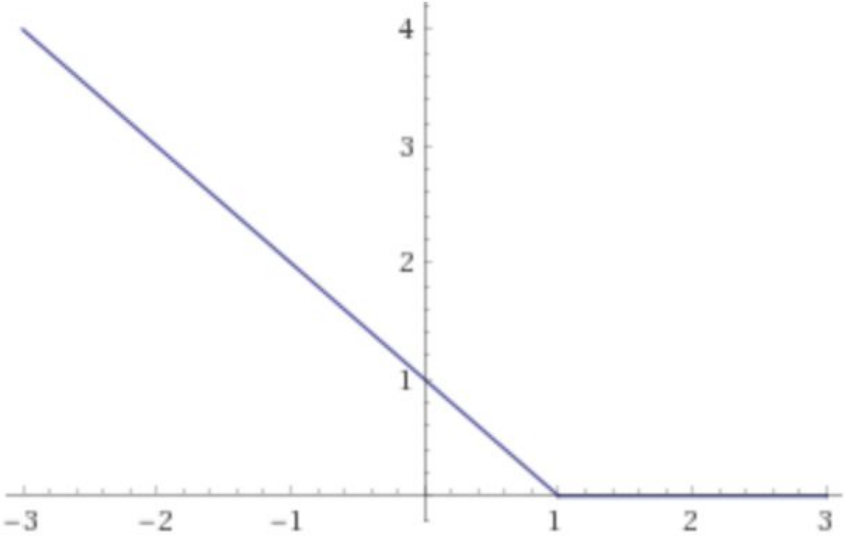
Binary classification

$$h_w(x) = \begin{cases} 1 & w^T x \geq 1 \\ 0 & w^T x \leq -1 \end{cases}$$

Support Vector Machine Classification

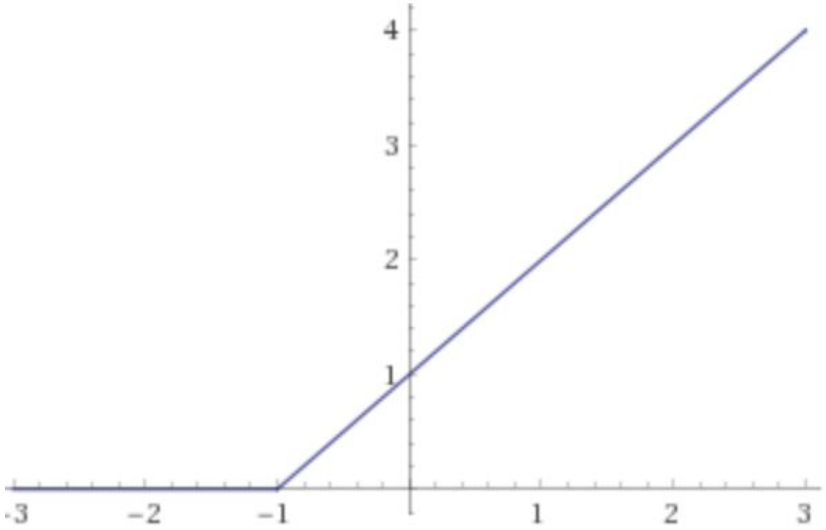
Binary classification

$$h_w(x) = \begin{cases} 1 & w^T x \geq 1 \\ 0 & w^T x \leq -1 \end{cases}$$



$\max(1 - z, 0)$

$y = 1$



$\max(z + 1, 0)$

$y = 0$

Support Vector Machine Cost

$$\underset{w}{\text{minimize}} \ \frac{1}{2}\|w\|_2^2$$

s.t.

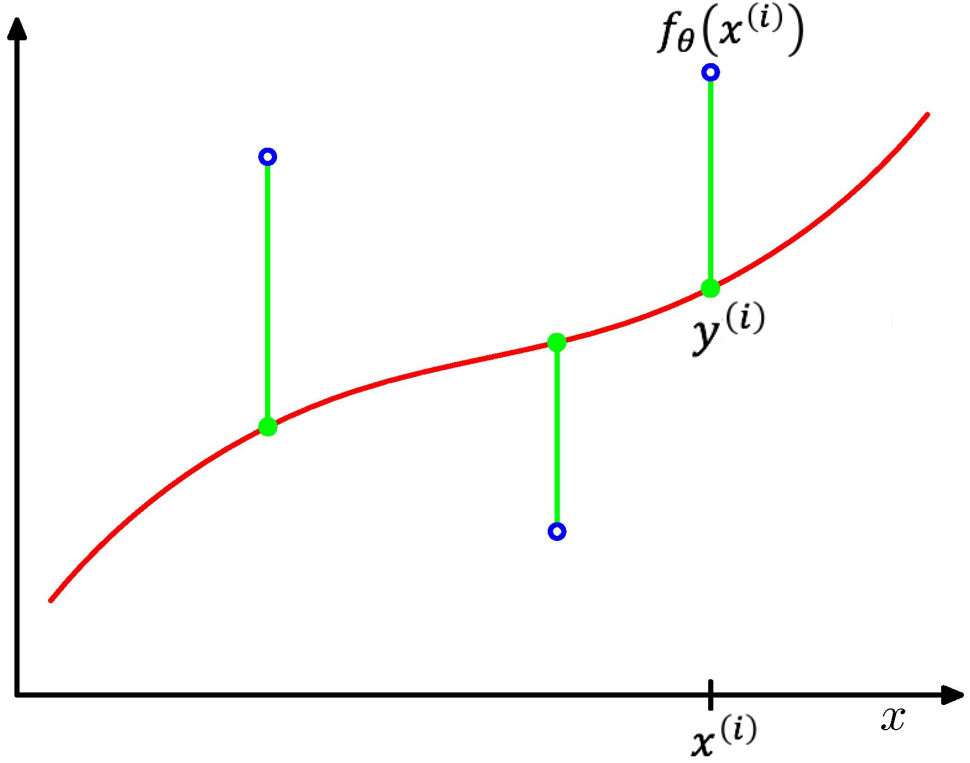
$$w^T x^{(i)} \geq 1 \quad if \quad y^{(i)} = 1$$

$$w^T x^{(i)} \leq -1 \quad if \quad y^{(i)} = 0$$

NYU

Regression

Sum of Squares Error

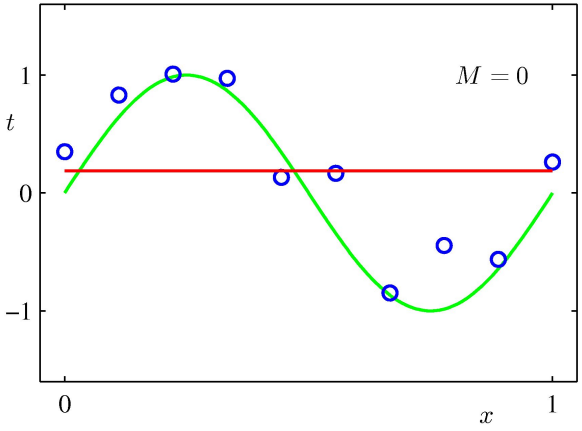


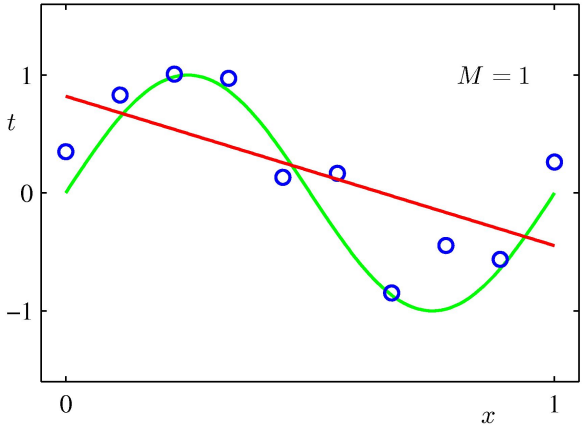
$$f_\theta(x) = \sum_{j=0}^{M} \theta_j x^j$$

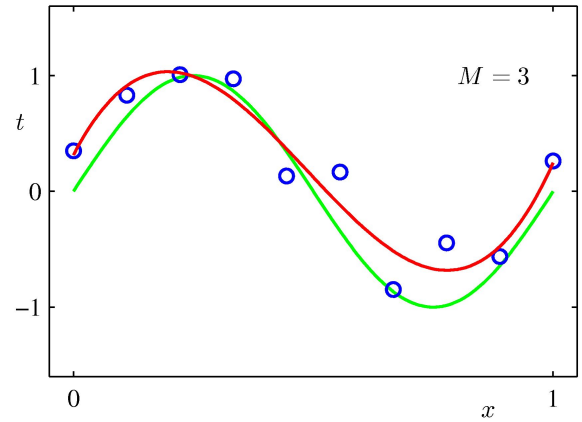
$$\mathcal{L}(\theta) = \frac{1}{2} \sum_{i=1}^{n} \left(f_\theta(x^{(i)}) - y^{(i)} \right)^2$$

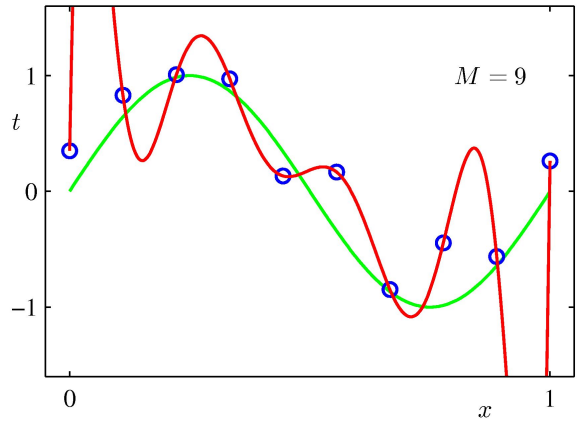
Source: Bishop Ch. 1

Problem: Overfitting

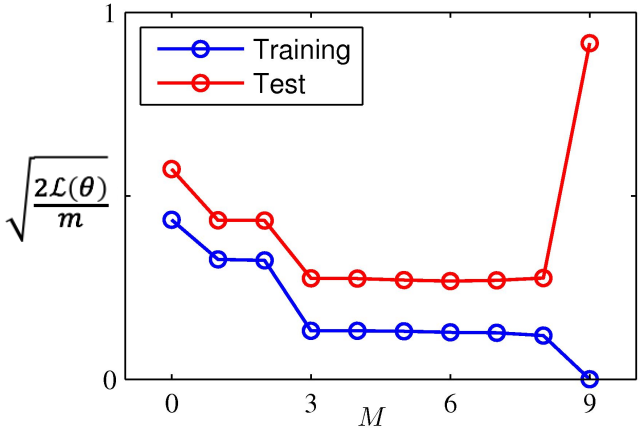






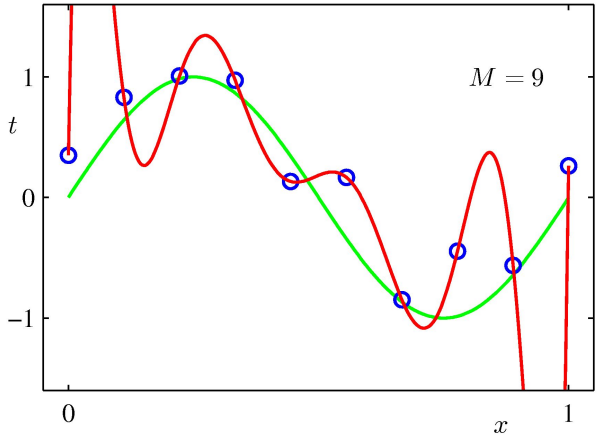


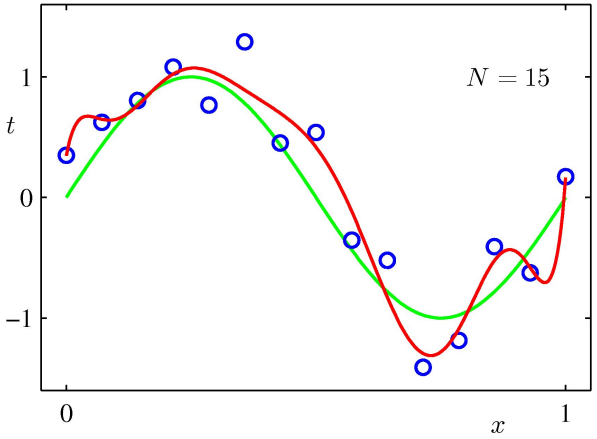
$$f_\theta(x) = \sum_{j=0}^{M} \theta_j x^j$$

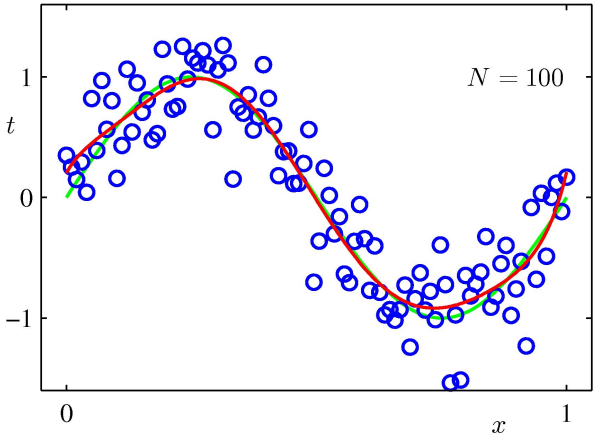


Source: Bishop Ch. 1

Solution 1: Data

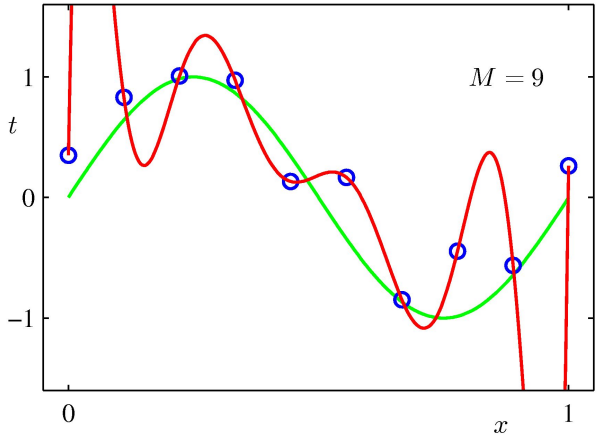


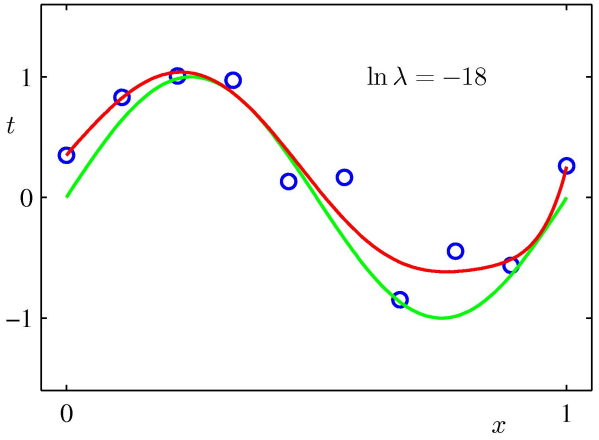




Source: Bishop Ch. 1

Solution 2: Regularization





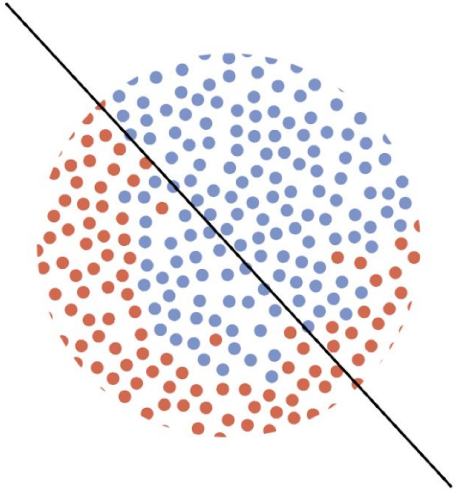
Source: Bishop Ch. 1

NYU

Overfitting

Regularization

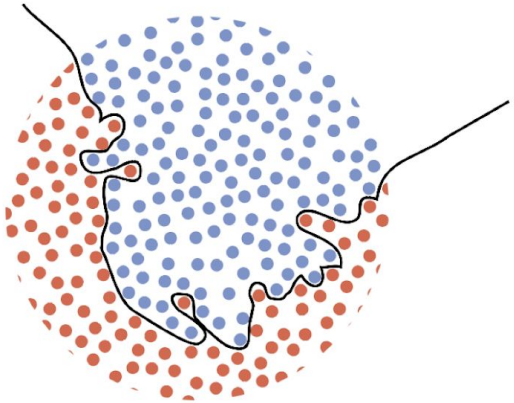
underfitting



high bias

low variance

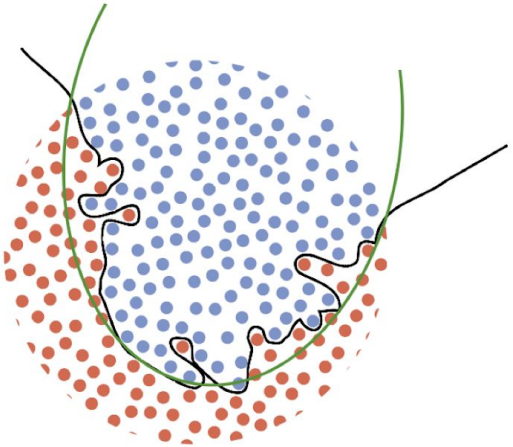
overfitting



low bias

high variance

regularized



low bias

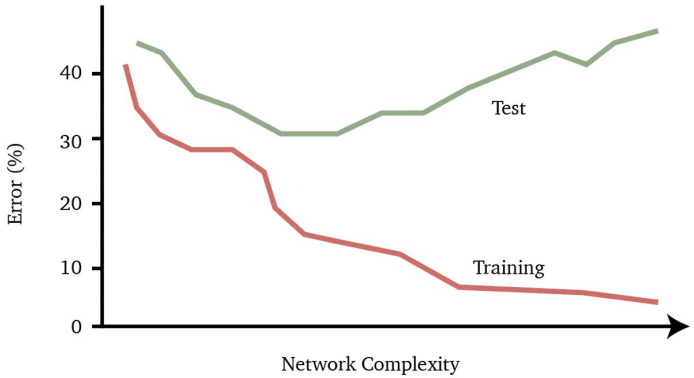
low variance

Overfitting

- Trees: continue splitting
- Linear models: adding variables, features

Fitting Curves

- Training and test accuracy as a function of model complexity for given dataset.
- Overfitting: large gap between training error and test error.



NYU

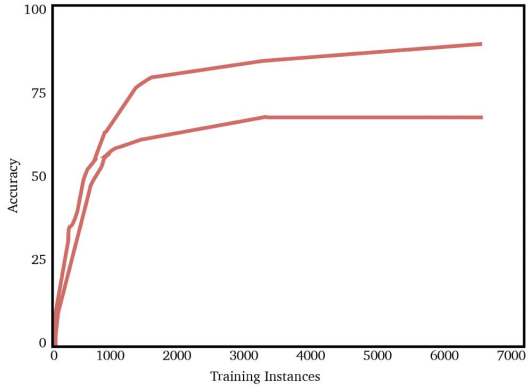
Avoid Overfitting: Trees

- Early stopping on minimum # of instances in leaf
- Many trees with different stopping criteria, check performance on test set
- Pruning

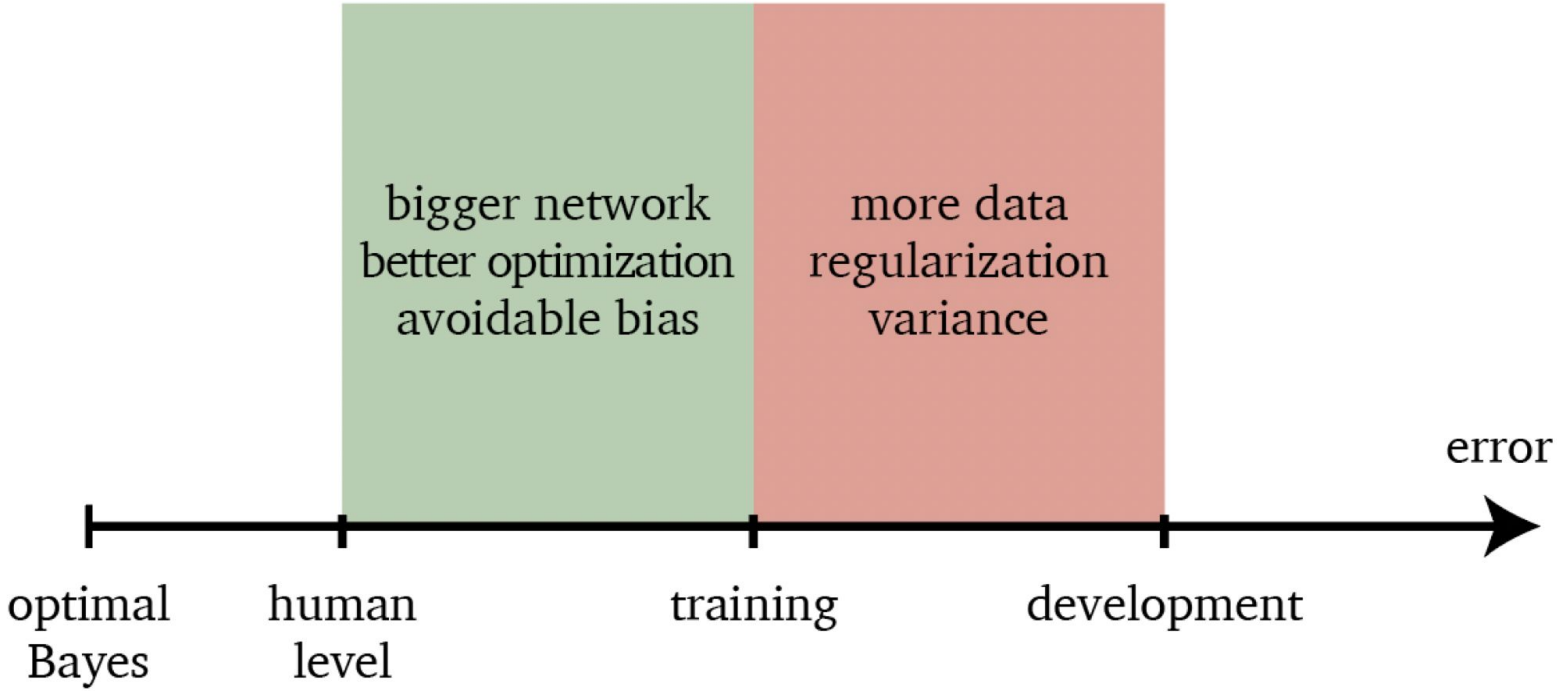
Learning Curves

Test accuracy as function of amount of training data for various models.

Adding training data increases generalization accuracy until a limit, reducing generalization error as expected.



Bias and Variance

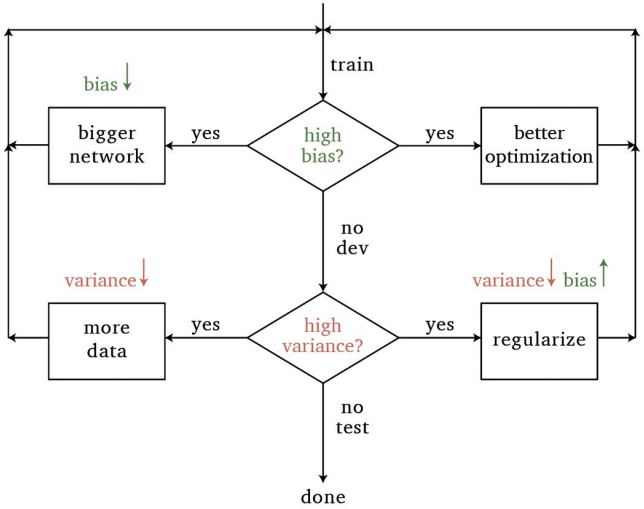


Bias and Variance

Our goal is to reduce both bias and variance.

Reduce bias by training a deeper and wider network.

Reduce variance by obtaining more data or by regularization.



Bias and Variance

Variance: amount by which approximation would change for different training sets.

Bias: error introduced by approximating complicated model by a simpler model.

Average test MSE for multiple training sets.

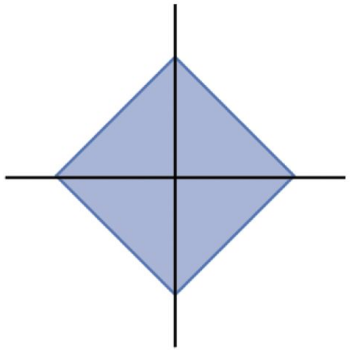
$$E\left(y - \hat{f}(x)\right)^2 = var\left(\hat{f}(x)\right) + \left(bias\left(\hat{f}(x)\right)\right)^2 + var(\varepsilon)$$

NYU

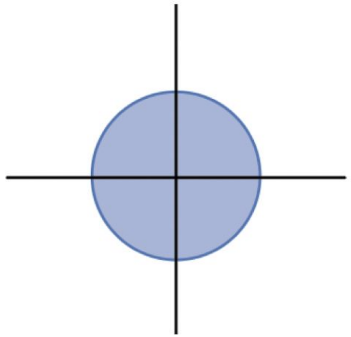
Regularization

Norms

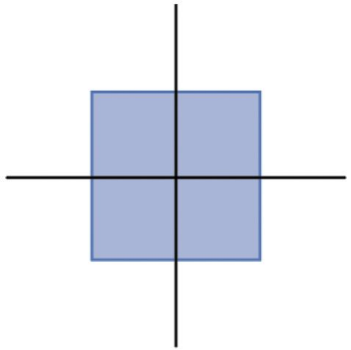
$\|x\| \geq 0$ and $\|x\| = 0$ iff $x = 0$ $\quad \|x + y\| \leq \|x\| + \|y\|$ $\quad \|\alpha x\| = |\alpha|\|x\|$



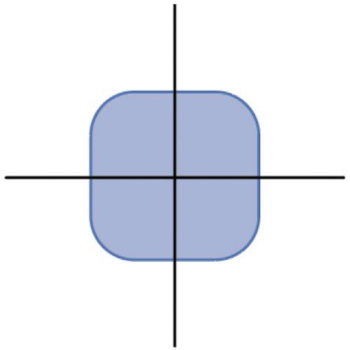
$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$



$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$



$$\|x\|_\infty = \max_i \|x_i\|$$



$$\|x\|_p = \left(\sum_{i=1}^{n} (\|x_i\|^p)\right)^{\frac{1}{p}}$$

$$1 \leq p \leq \infty$$

Regularization

$$\frac{1}{m}\sum_{i=1}^{m} L(y^i, f_w(x^i)) + R(w)$$

$$R(w) = \lambda \|w\|_p \qquad \lambda > 0$$

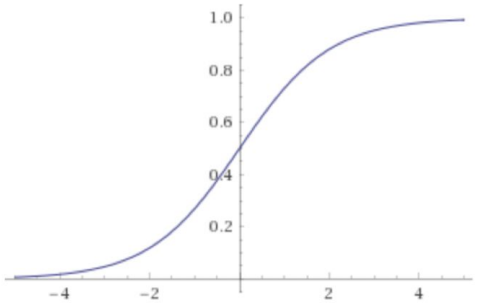
Regularization

$$\frac{1}{m}\sum_{i=1}^{m} L(y^i, f_w(x^i)) + R(w)$$

$$R(w) = \lambda \|w\|_p \qquad \lambda > 0$$

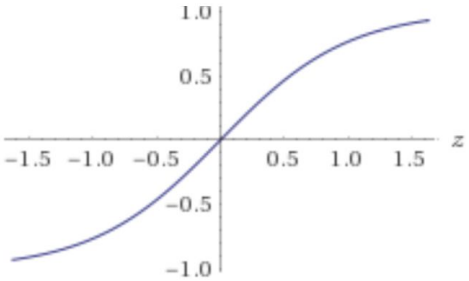
$$Z^l = W^l A^{l-1} \qquad A^l = g^l(Z^l)$$

Sigmoid

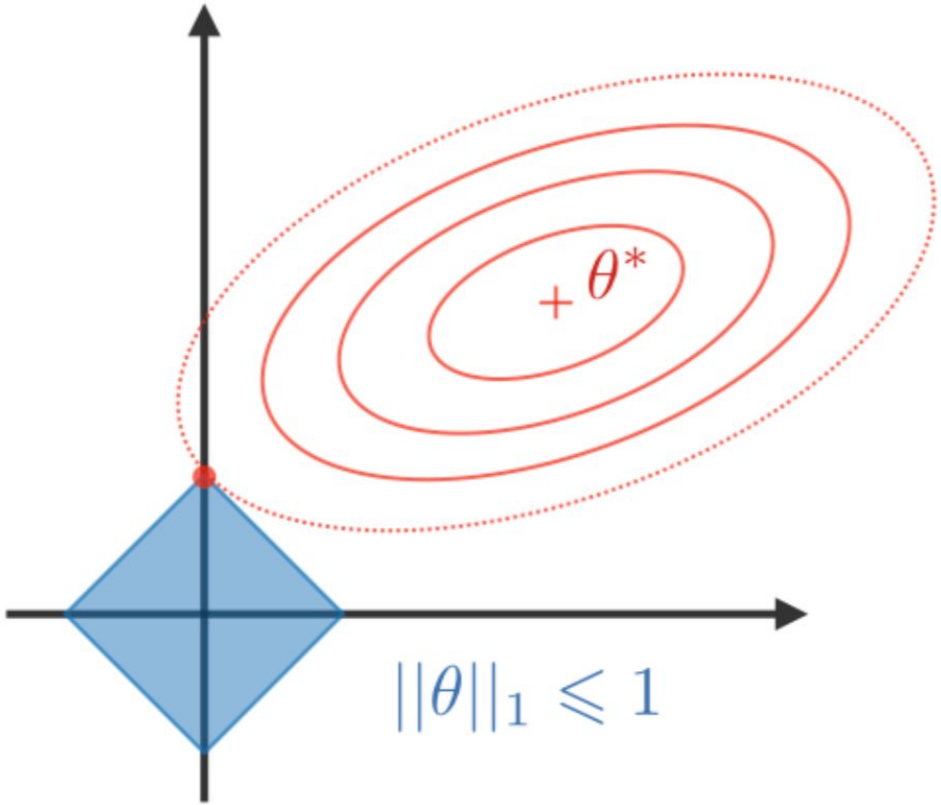


$$\sigma(z) = g(z) = \frac{1}{1+e^{-z}}$$

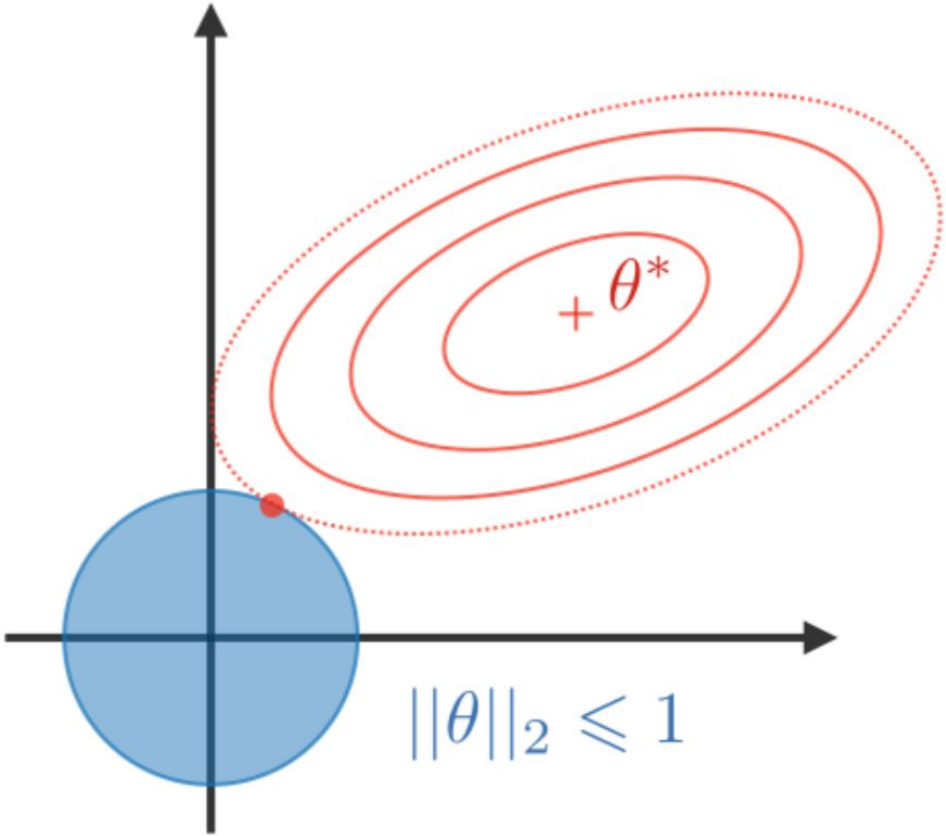
Hyperbolic Tangent



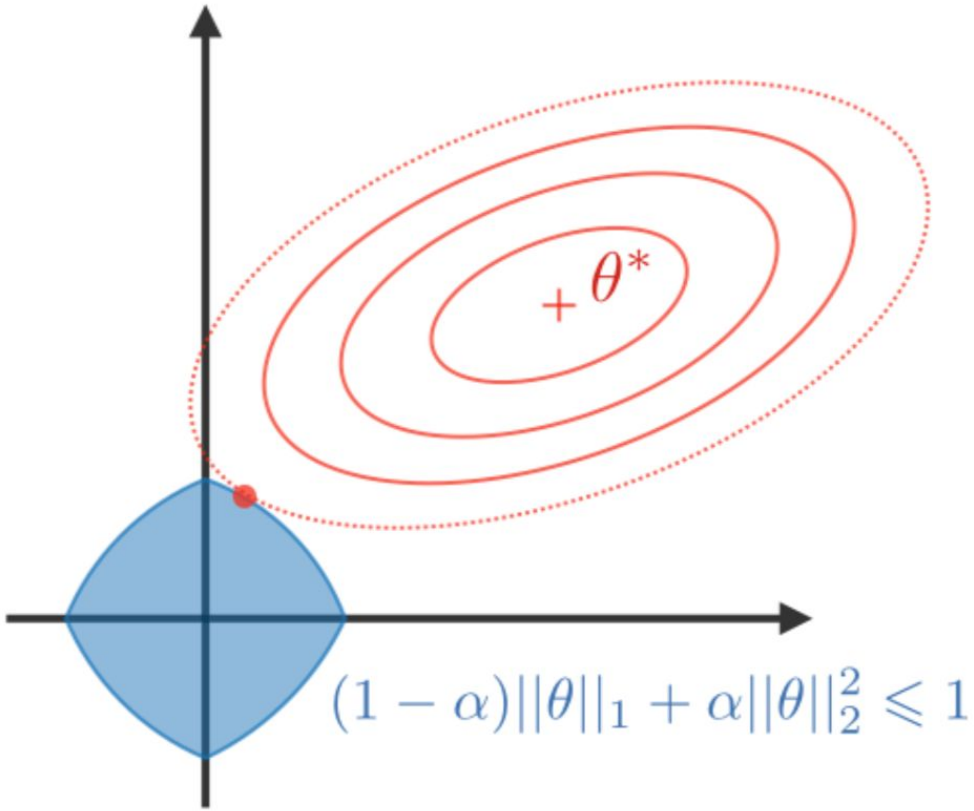
Regularization: Lasso



Regularization: Ridge



Regularization: Elastic



Regularization

$$W^{\ell} = W^{\ell} - \alpha dW^{\ell}$$

without regularization

$$W^{\ell} = W^{\ell}\left(1 - \frac{\lambda}{m}\right) - \alpha dW^{\ell}$$

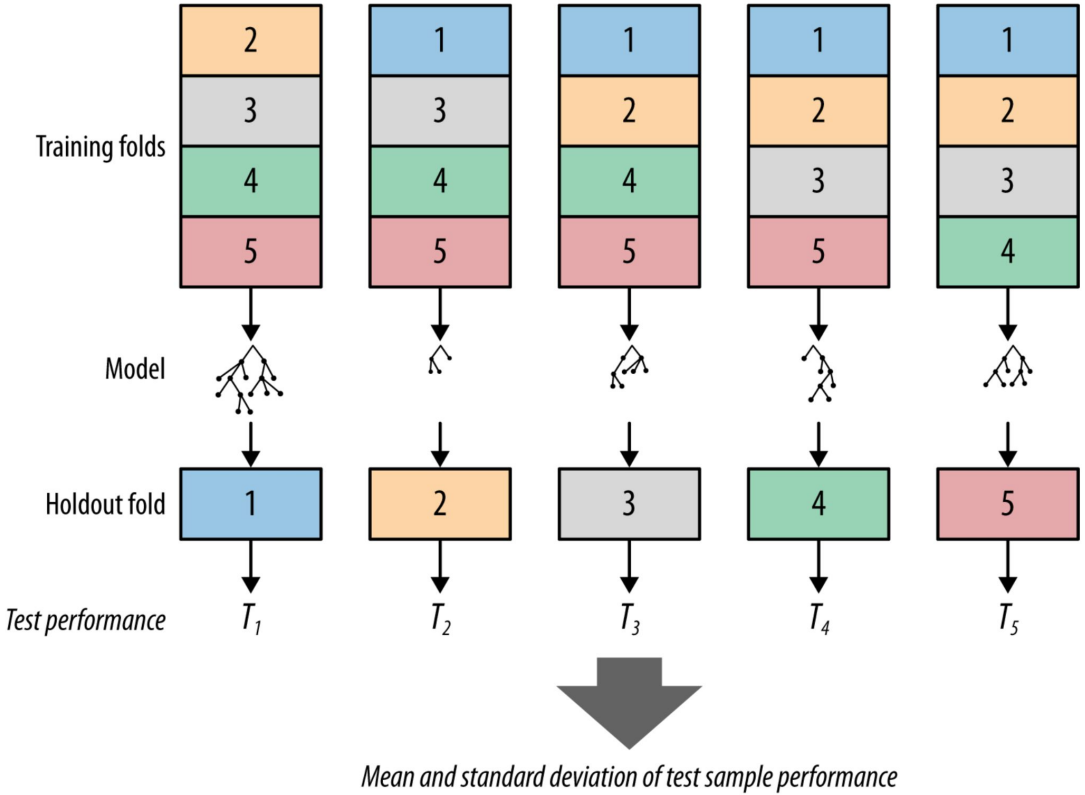
with regularization

shrinks weights

NYU

Cross Validation

Cross Validation



Source: DSB

Cross Validation

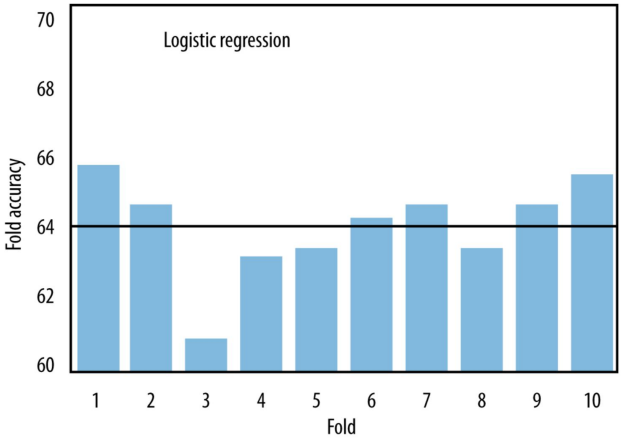
Randomly split data into k folds and iteratively

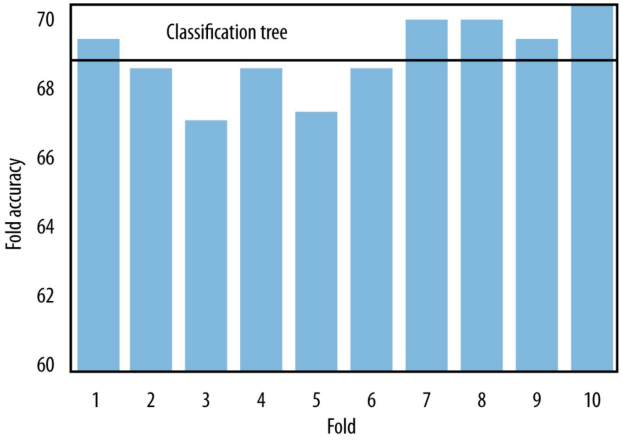
take training data to be k−1 folds for building model which is tested on remaining fold.

After testing each model we compute mean and variance of generalization error over all k models.

$$G(f(X,W)) = \frac{1}{m}\sum_i L(y^i, f(x^i, W)) - E_{(X,Y)\sim P}(L(y^i, f(x^i, W)))$$

Cross Validation





Source: DSB

NYU

Ensemble of Models

Ensemble of Models

- Taking the argmax over the average of probabilities over the models for each class
- Models m, classes j

$$y = \arg\max_j \frac{1}{m}\left(\sum_{i=1}^{m} p_i^{(j)}\right)$$

Source: DSB

NYU

Optimization

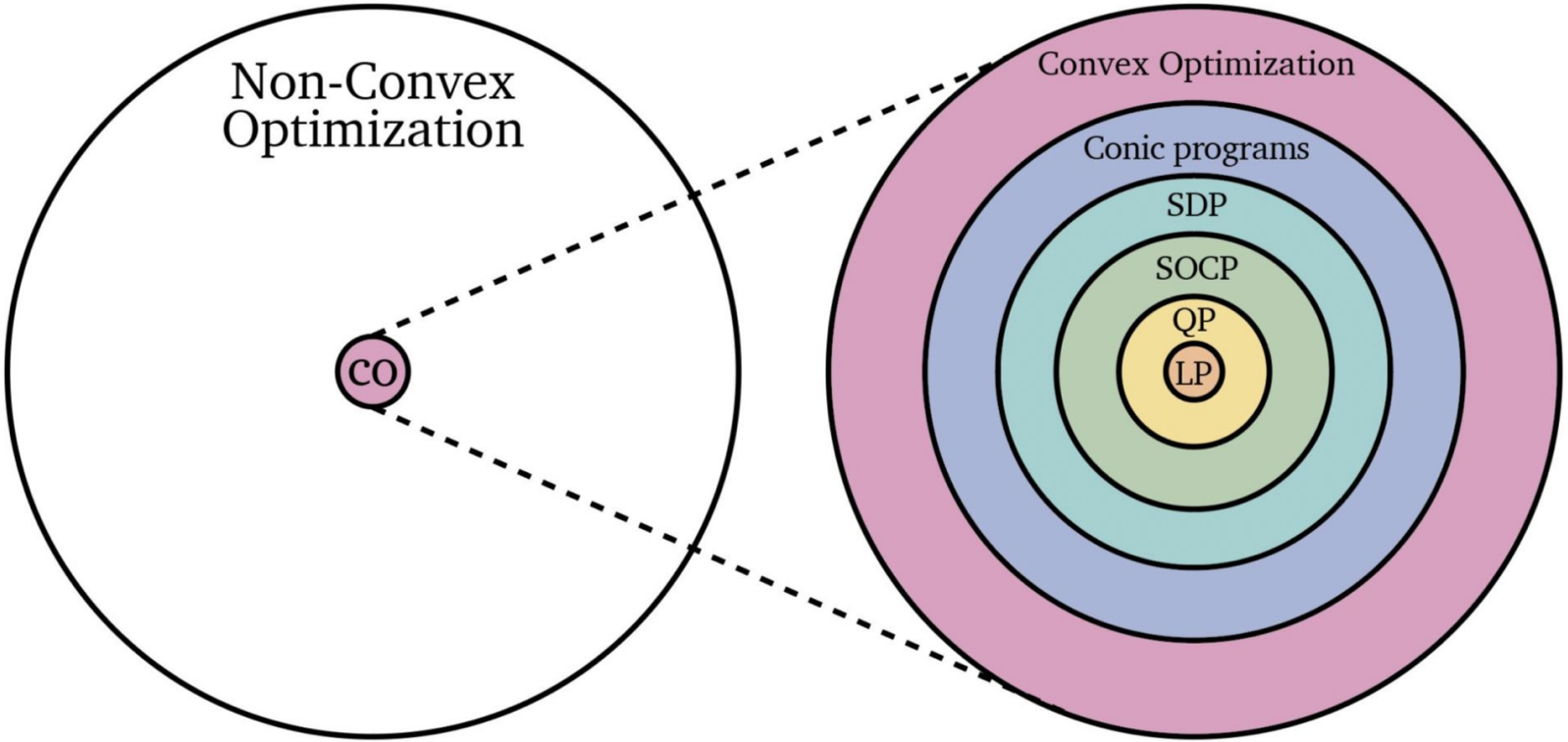
Optimization

Translate problem into optimization problem

Solve optimization problem

$$\underset{x \in D}{\text{minimize}}\, f(x)$$

Optimization Problems



Convex Optimization

$$\underset{x \in D}{\text{minimize}}\ f(x)$$

subject to

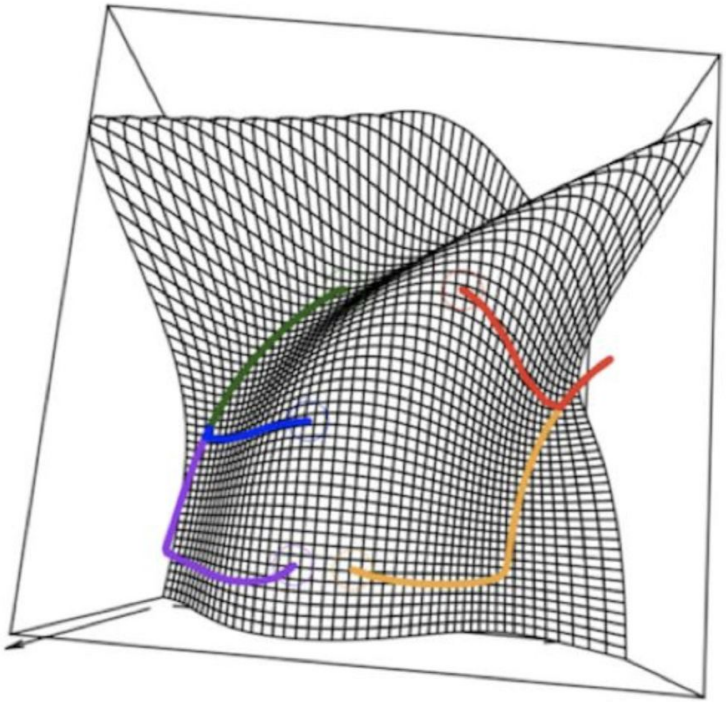
$$g_i(x) \leq 0 \qquad i = 1, \ldots, m$$

$$h_j(x) = 0 \qquad j = 1, \ldots, r$$

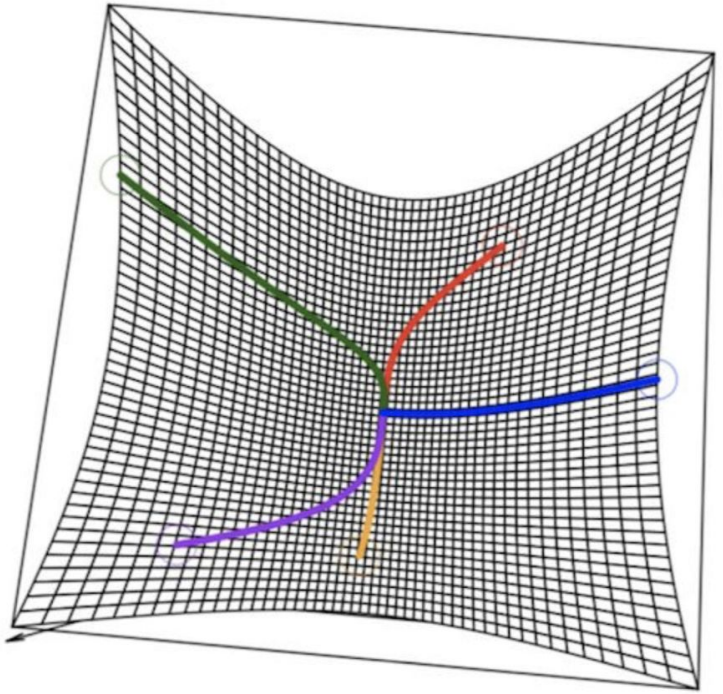
convex functions  affine functions $ax+b$

Any local minimum is a global minimum

Convex vs. Non-Convex Optimization



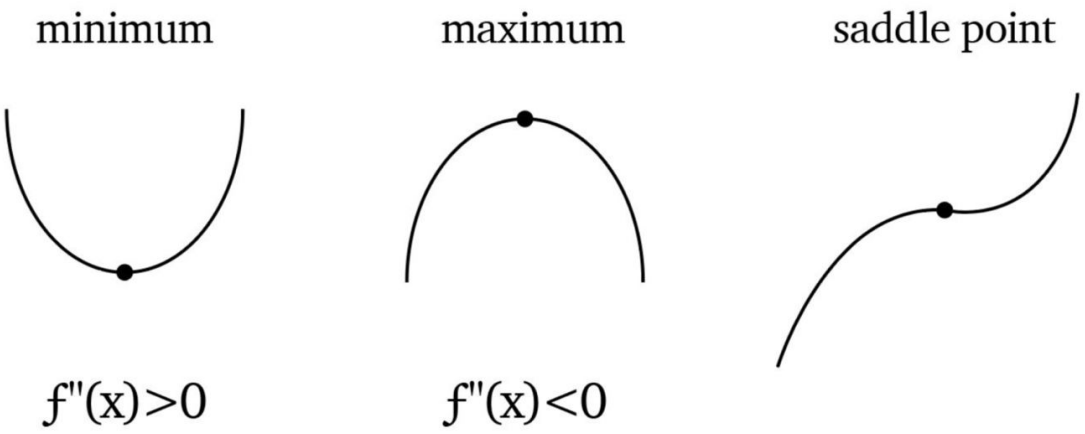
non-convex



convex

Extreme Points

A function for which $f'(x)=0$ has one of three extreme points:



Optimization Algorithms

- **Gradient Descent**: simple, fast, commonly used, local minimum for convex problems.
- **Newton's method**: impractical for neural networks.
- **Quasi-Newton methods**: practical, less frequently used.

Gradient Descent

$$\underset{x \in D}{\text{minimize}}\, f(x)$$

Algorithm 1 Gradient descent

given a starting point $x \in \mathbf{dom} f$

repeat

determine descent direction $\Delta x = -\nabla f(x)$

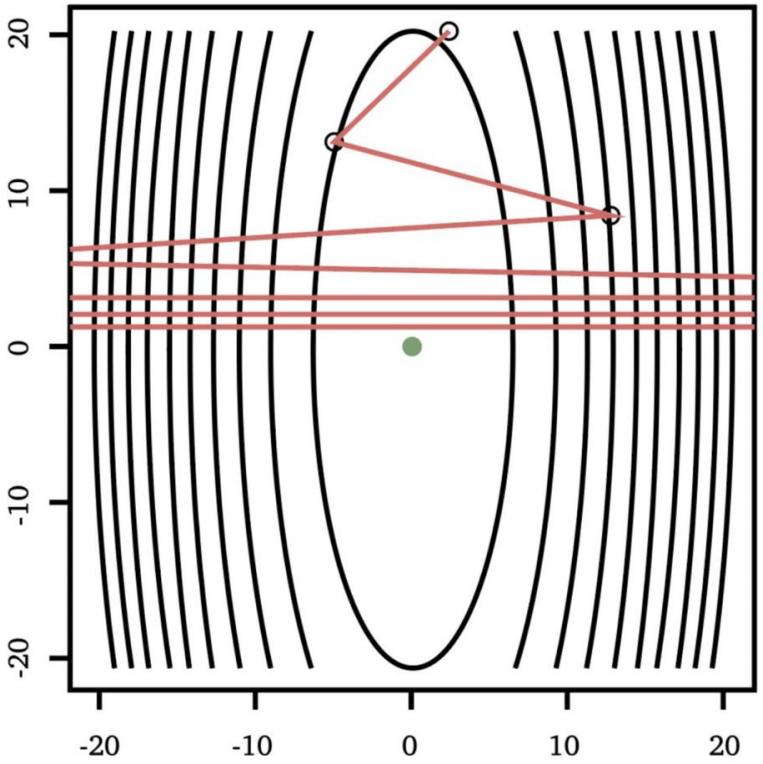
choose step size α

update $x = x + \alpha \Delta x$

until stopping criterion is satisfied.

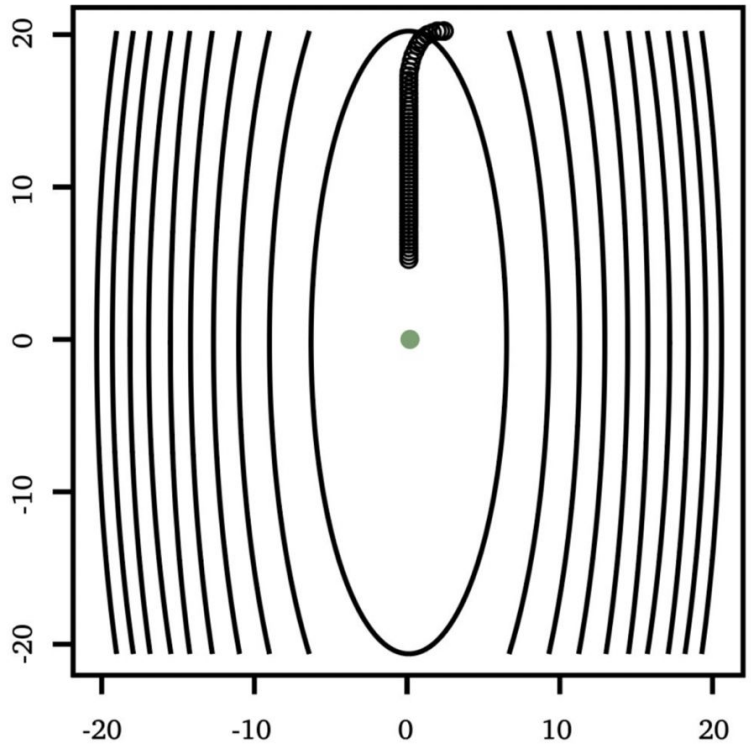
Gradient Descent Step Size

too big



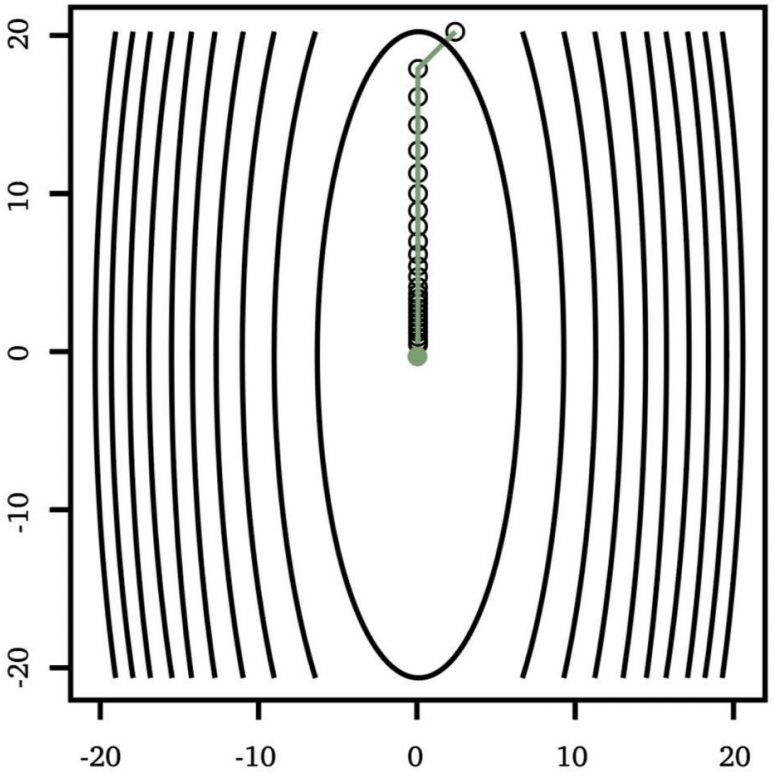
Gradient Descent Step Size

too small



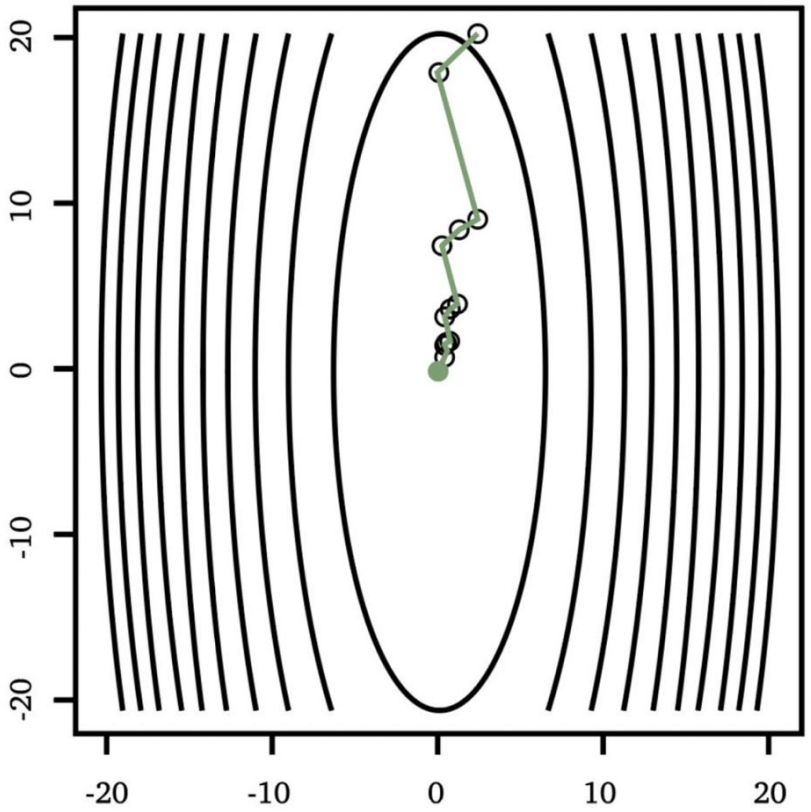
Gradient Descent Step Size

just right



Gradient Descent Step Size

adaptive
line search
optimize for size
expensive



Mini-Batch Gradient Descent

Split data into n/k disjoint sets of size k

Instead of process entire dataset, iteratively process each of the subsets

Example:
estimating gradient from 100 samples instead of 10000 samples

+ Reducing computation time and memory by a factor of **100**
- Reduces standard error of the mean only by a factor of **10**:
  - Standard error of sample mean is $s/sqrt(n)$ where s is the standard deviation of the population and n in the number of observations of the samples.
  - If $x_1,...,x_n$ are n independent sample then the variance of their sum $T=sum(xi)$ is ns^2
  - The variance of the sample mean T/n is s^2/n and the standard deviation is $s/sqrt(n)$

Stochastic Gradient Descent

Special case in which the mini-batch is of size 1

Using a single random sample at a time

Suitable as online method handling data stream, single example at a time.