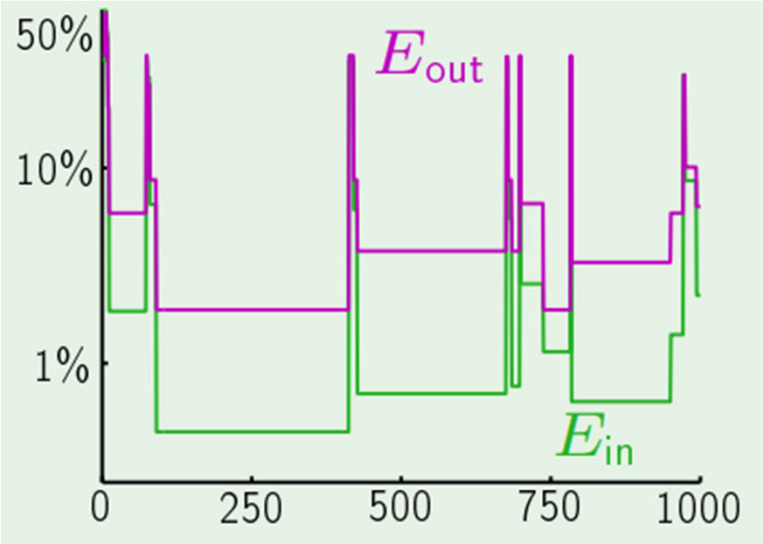
Questions

Pocket Algorithm

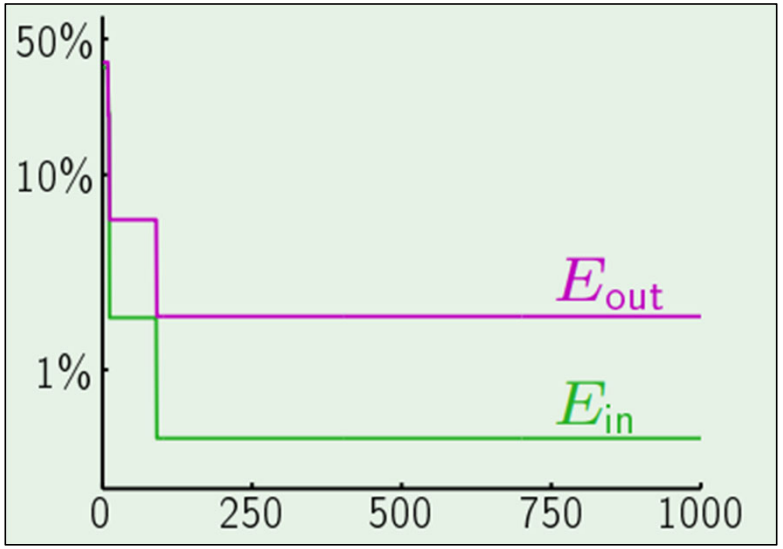
- Questions on Piazza?
- Question for You!
  - Can you think of another way to use Perceptron for non-separable data



Questions

Pocket Algorithm

- Questions on Piazza?
- Question for You!
  - Can you think of another way to use Perceptron for non-separable data



NYU Center for Data Science

DS-GA 1003
Machine Learning

Week 3: Lecture 3
Fitting Models – Gradient Descent and Ridge Regression

How can we use a guess and check method to find weights? Why would it be helpful to shrink the weights?

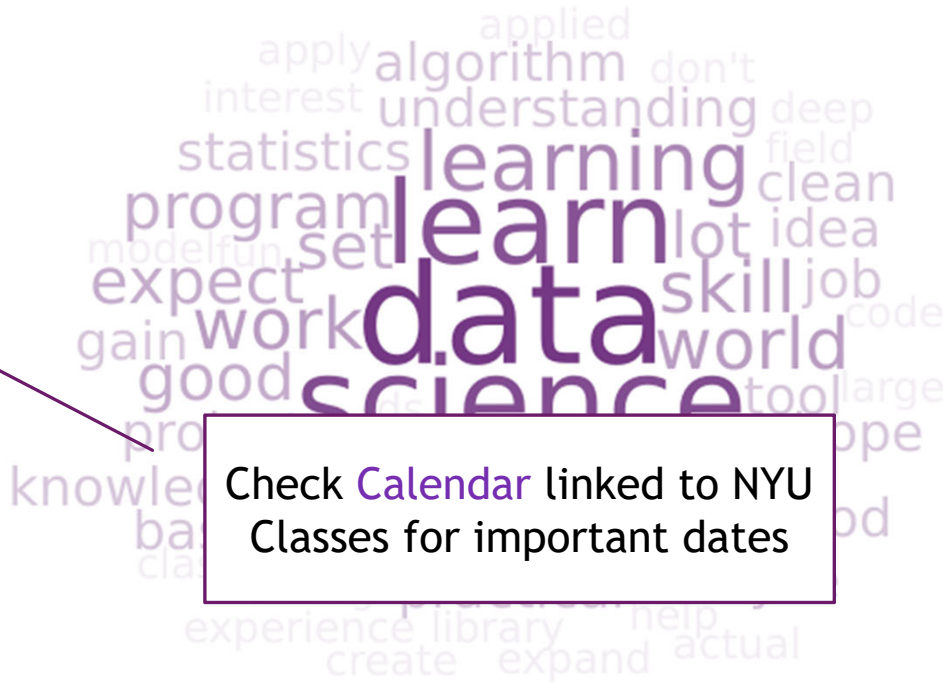
DS-GA 1003
Machine Learning

Week 3: Lecture 3
Fitting Models – Gradient Descent and Ridge Regression

Adapted from Rosenberg, Efron, Rangan, Shalev-Shwartz

Announcements

- Please check Week 2 agenda on NYU Classes
  - Homework 2
  - Homework 1
  - Tutoring Session, Office Hours
- Remember to post to Piazza



Check Calendar linked to NYU Classes for important dates

Review

Perceptron Algorithm

input: A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$

initialize: $\mathbf{w}^{(1)} = (0, \ldots, 0)$

  for $t = 1, 2, \ldots$

    if $(\exists\, i \;\text{ s.t. }\; y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle \leq 0)$ **then**

      $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$

    else

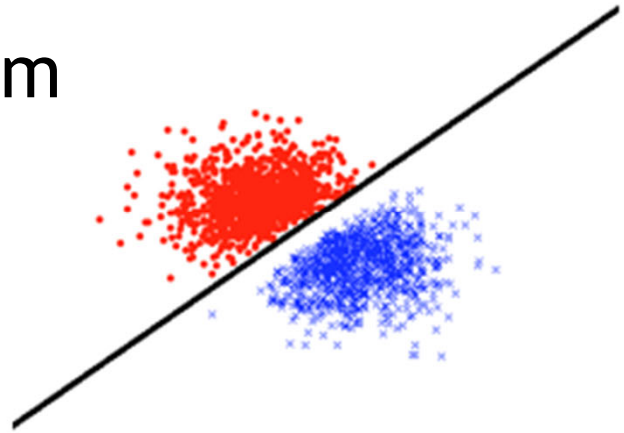
      output $\mathbf{w}^{(t)}$

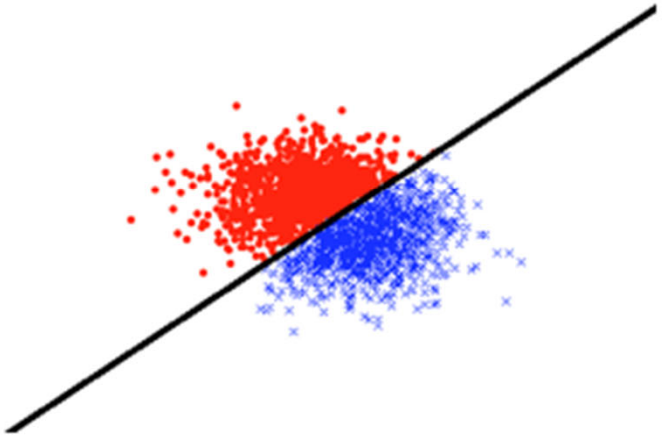
$$y \langle w_t, x \rangle$$

$$y \langle w_{t+1}, x \rangle = y \langle w_t, x \rangle + ||x||^2$$

Review

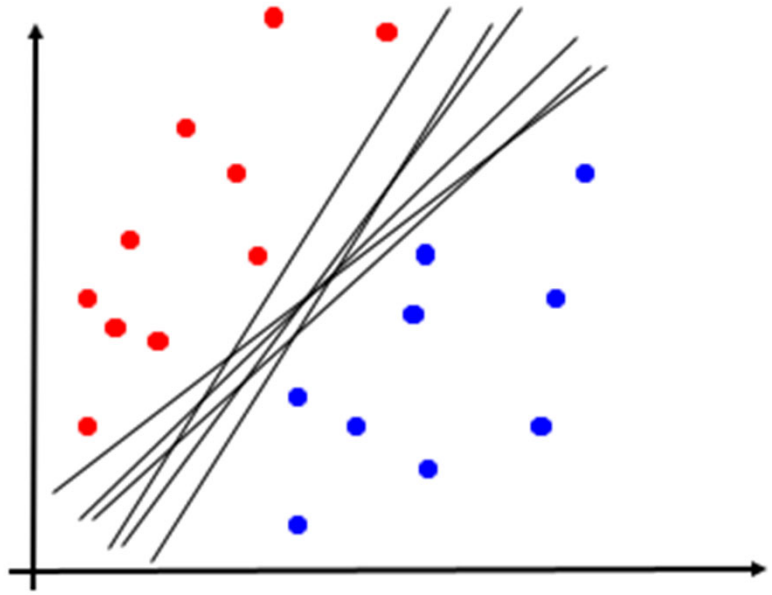
- Advantages
  - Error Bound
  - Online Algorithm
- Disadvantages
  - Many Decision Boundaries
    - Overfitting
  - Separable Data





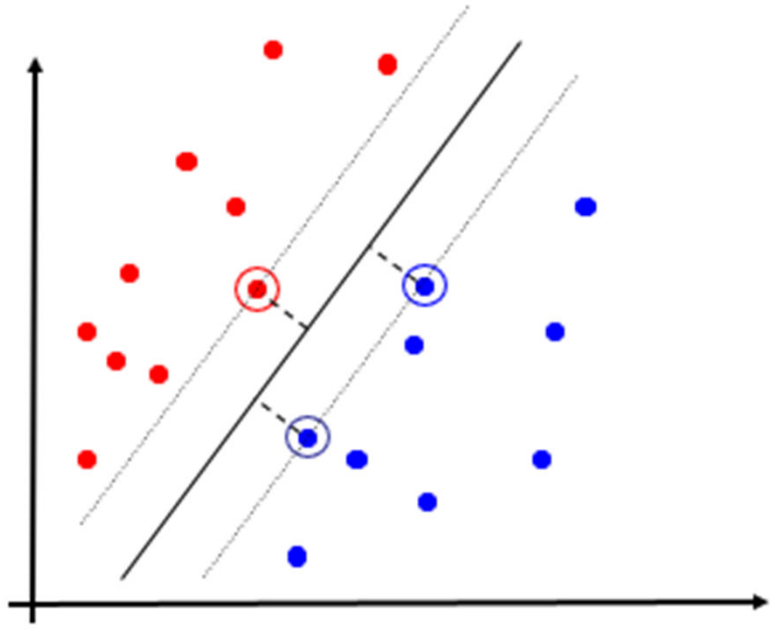
Review

- Advantages
  - Error Bound
  - Online Algorithm
- Disadvantages
  - Many Decision Boundaries
    - Overfitting
  - Separable Data



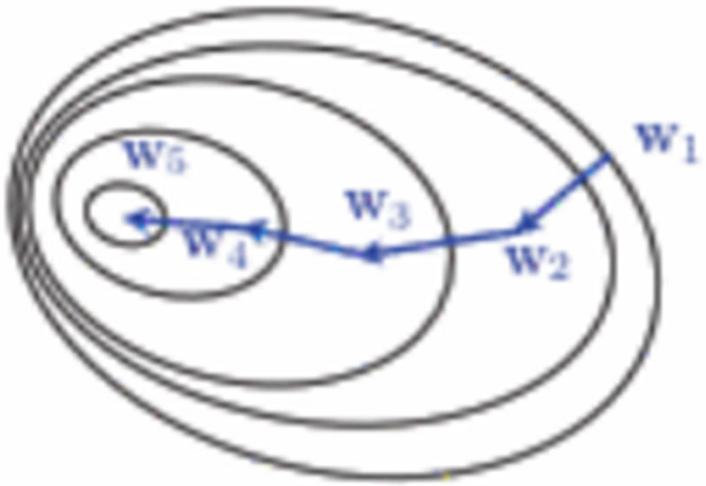
Review

- Advantages
  - Error Bound
  - Online Algorithm
- Disadvantages
  - Many Decision Boundaries
    - Overfitting
  - Separable Data



Agenda

- Statistical and Empirical Risk
  - Decomposition into Estimation, Approximation and Optimization Error
- Gradient Descent
  - Using rate of change of empirical risk to iteratively update weights
- Ridge Regression
  - Incorporating a penalty into the empirical risk to prevent against large weights.



Risk

- Steps for machine learning might have us
  - Observe input
  - Take action
  - Observe output
  - Evaluate action by some metrics

Prediction function

$$f : x \in \mathcal{X} \mapsto \mathcal{A} \ni f(x)$$

Loss function

$$\ell : (a, y) \in \mathcal{A} \times \mathcal{Y} \mapsto \mathbb{R} \ni \ell(a, y)$$

Input space: $\mathcal{X}$

Action space: $\mathcal{A}$

Outcome space: $\mathcal{Y}$

- Usually output is independent of action
  - Reinforcement Learning studies dependent output and action

Risk

- Steps for machine learning might have us
  - Observe input
  - Take action
  - Observe output
  - Evaluate action by some metrics

Feature Space $\mathcal{Z}$

Prediction function

$$f : x \in \mathcal{X} \mapsto \mathcal{A} \ni f(x)$$

Loss function

$$\ell : (a, y) \in \mathcal{A} \times \mathcal{Y} \mapsto \mathbb{R} \ni \ell(a, y)$$

Input space: $\mathcal{X}$

Action space: $\mathcal{A}$

Outcome space: $\mathcal{Y}$

- Usually output is independent of action
  - Reinforcement Learning studies dependent output and action

Risk

- Data generating distribution P_{XxY}
- Pairs of input/output (x,y) come from random samples drawn independently from P_{XxY}

$$\text{Risk of } f : \mathcal{X} \mapsto A \text{ is}$$

$$R(f) = \mathbb{E}\,\ell\left(f(x), y\right)$$

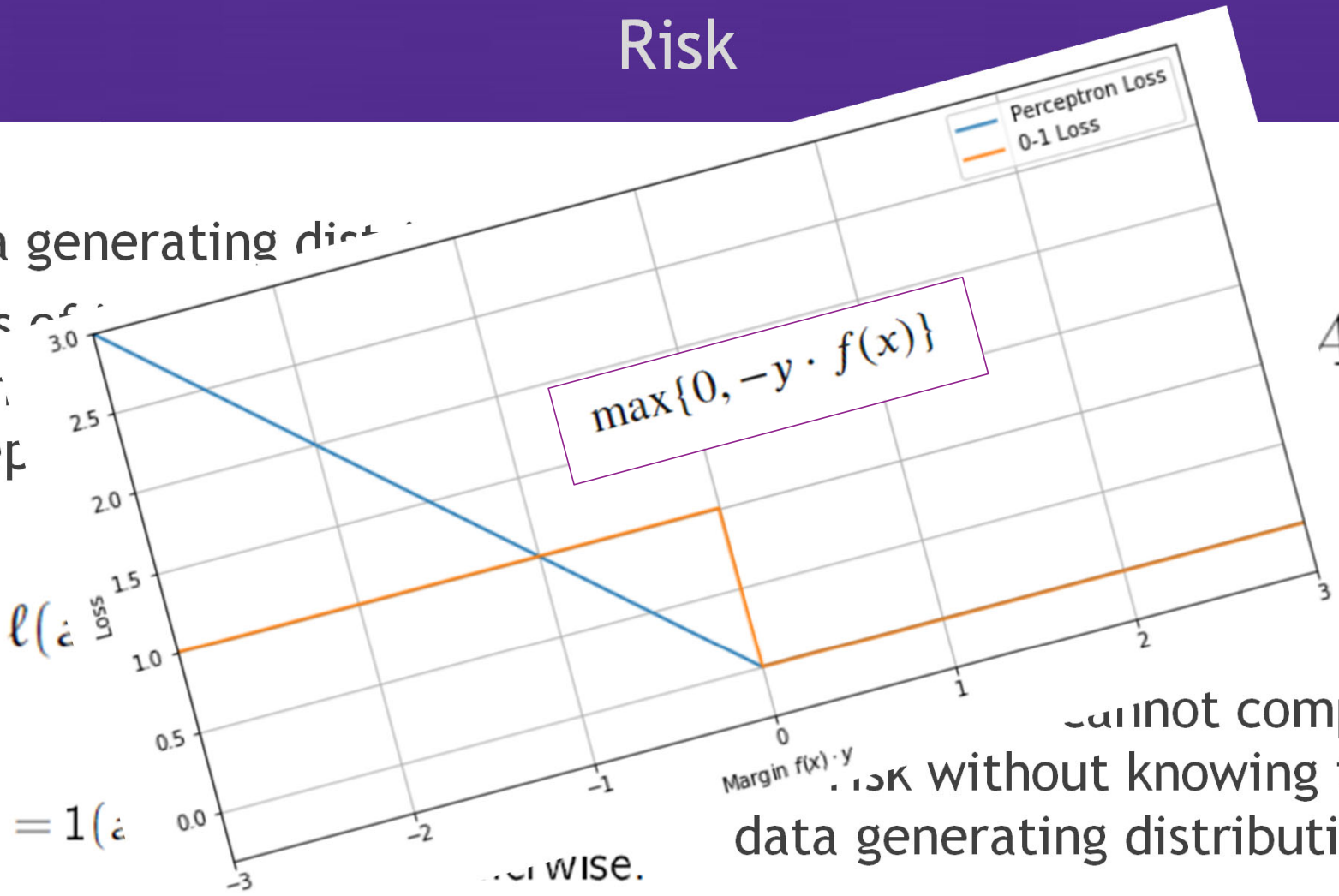
$$\ell(a, y) = (a - y)^2$$

$$\ell(a, y) = 1(a \neq y) := \begin{cases} 1 & \text{if } a \neq y \\ 0 & \text{otherwise.} \end{cases}$$

- Note that we cannot compute the risk without knowing the data generating distribution

Risk

- Data generating dist
- Pairs of
  from
  indep



4 is

$y)$

$\ell(a$

$\ell(a, y) = 1(a$

wise.

cannot compute
risk without knowing the
data generating distribution

Risk

- We can repeatedly sample to approximate the expectation through averages.
- Draw (x_1,y_1) through (x_n,y_n) for n large

Empirical Risk of $f : \mathcal{X} \mapsto A$ with respect to sample is

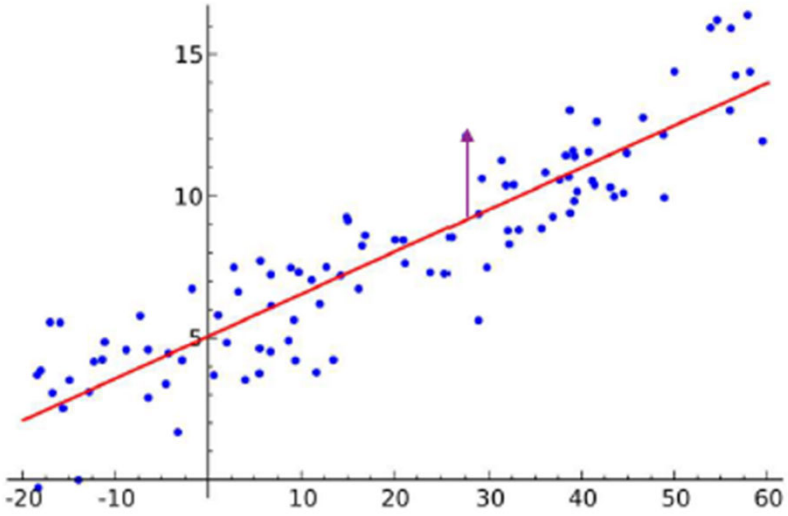
$$\widehat{R}_n(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$$

$$\lim_{n\to\infty} \hat{R}_n(f) = R(f)$$

- Sometimes risk is called statistical risk to distinguish it from empirical risk. Think population versus sample.

Risk

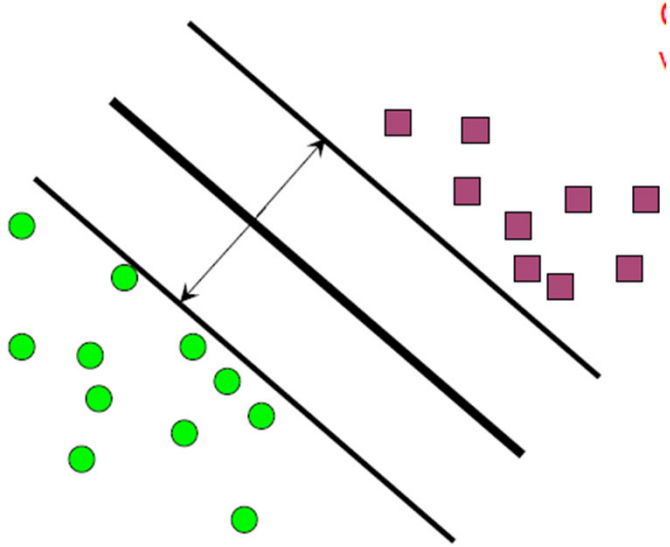
Linear Regression



$$\langle \mathbf{w}, x \rangle + b$$

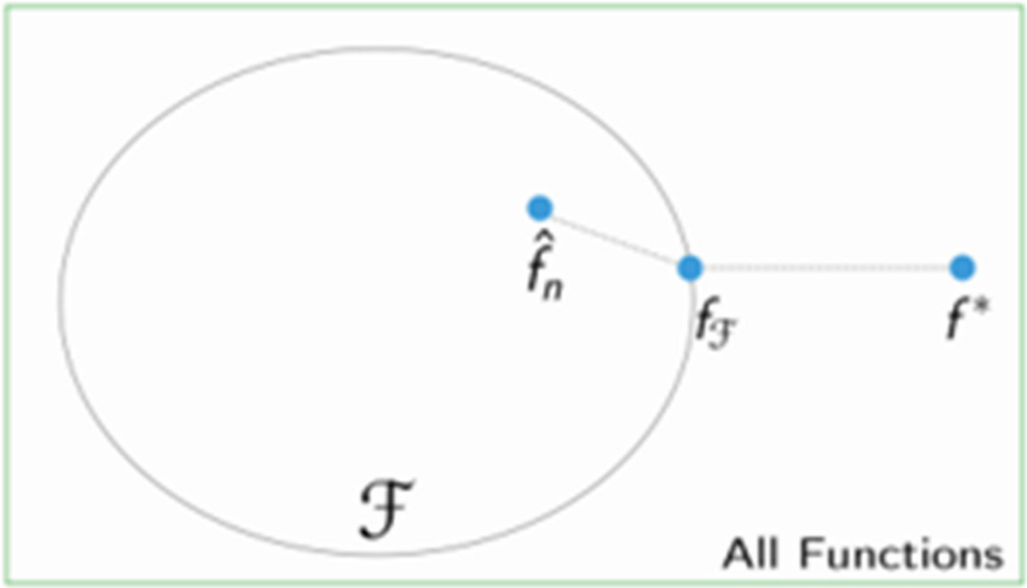
Perceptron

$$\text{sign}\left(\langle \mathbf{w}, x \rangle + b\right)$$



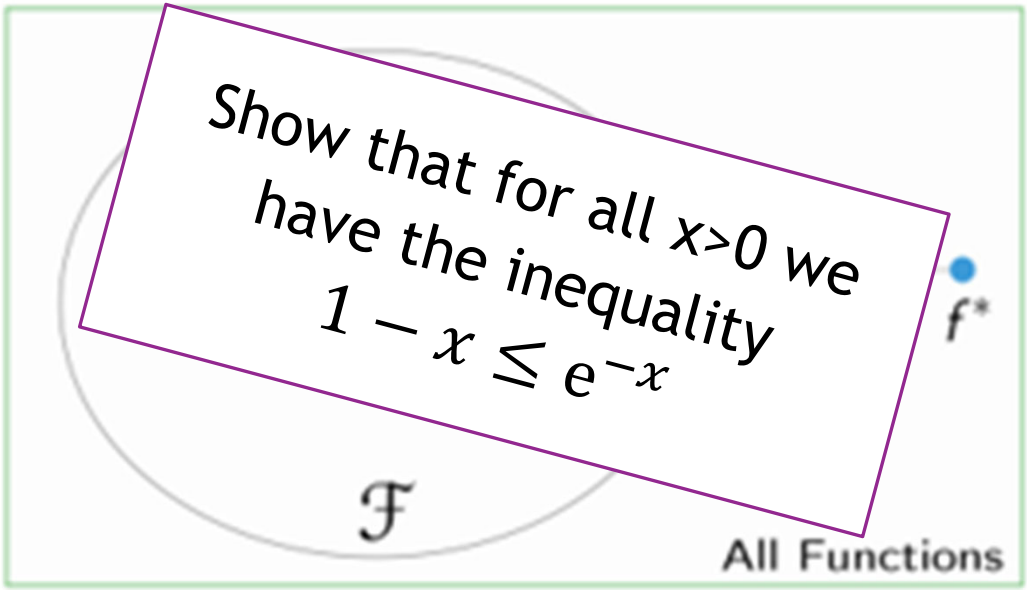
Exercise

- Consider
  - Approximation Error
  - Estimation Error
  - Optimization Error
- For each try to determine whether it's
  - Random or Not Random
  - Positive or Negative
  - Increases or Decreases with more data
  - Increases or Decreases with more Parameters
  - Can we ever compute it?



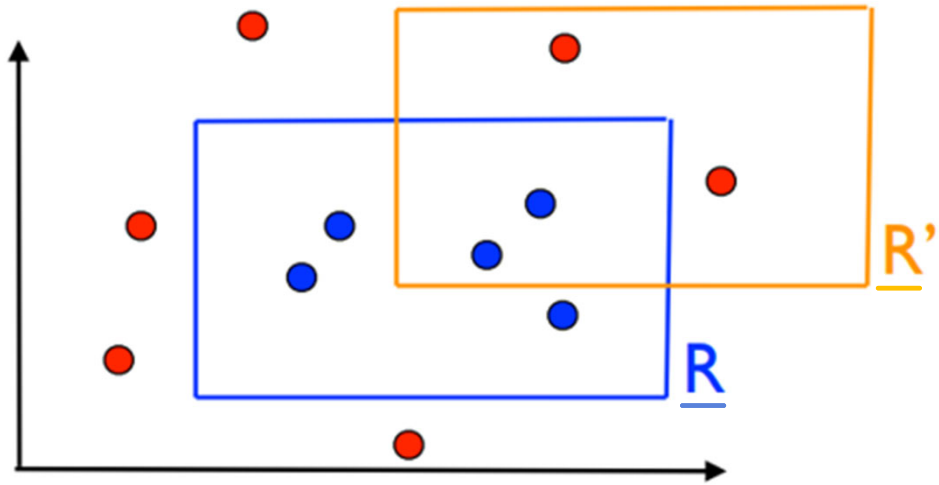
Exercise

- Consider
  - Approximation Error
  - Estimation Error
  - Optimization Error
- For each try to determine whether it's
  - Random or Not Random
  - Positive or Negative
  - Increases or Decreases with more data
  - Increases or Decreases with more Parameters
  - Can we ever compute it



PAC

- Problem: Suppose you want to predict ripeness of fruit. Does color and firmness impact ripeness?
- Input Space:
  - Fruit
- Features:
  - Encoding of color
  - Encoding of firmness
- Labels:
  - +1 for ripe
  - -1 for not ripe/spoiled



- Action Space:
  - Eat
  - Wait/Compost
- Loss Function:
  - +1 incorrect
  - 0 correct

PAC

- Proble… …pose you want to predi… …. Does colo… ripe…
- Tar…

$$f^* = \arg\min_f \ldots$$

- Statistical Risk Minimizer

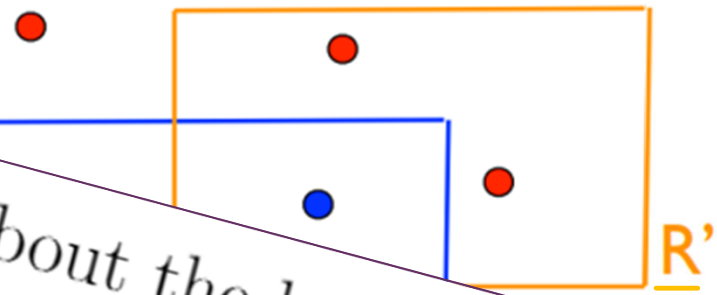
$$f_{\mathcal{F}}^* \in \arg\min_{f \in \mathcal{F}} \mathbb{E}\ell(f(x), y)$$

- Emp…

$$\hat{f}_n \in \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

What would be problematic about the hypothesis

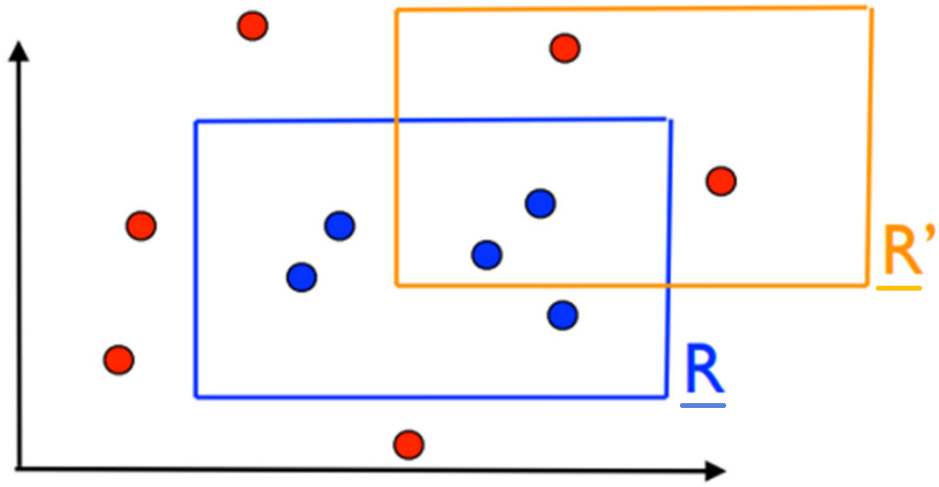
$$f(x) = \begin{cases} y & \text{if } (x, y) \text{ in training set} \\ 0 & \text{otherwise} \end{cases}$$



PAC

- Problem: Suppose you want to predict ripeness of fruit. Does color and firmness impact ripeness?
- Target Function

$$f^* = \arg\min_f \mathbb{E}\ell(f(x), y)$$



- Statistical Risk Minimizer

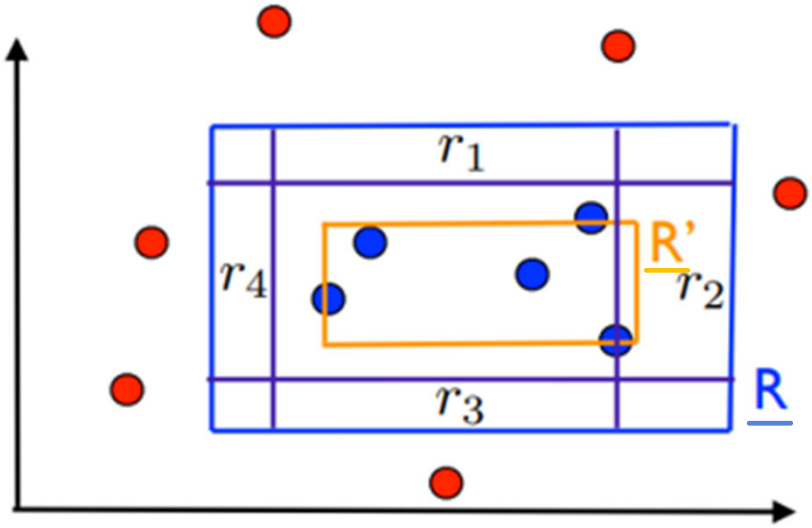
$$f^*_{\mathcal{F}} \in \arg\min_{f \in \mathcal{F}} \mathbb{E}\ell(f(x), y)$$

- Empirical Risk Minimizer

$$\hat{f}_n \in \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

PAC

- Problem: Suppose you want to predict ripeness of fruit. Does color and firmness impact ripeness?
- Hypothesis Space:
  - Assume the target function corresponds to rectangle R with sides parallel to the axes.
- Sample
  - Draw independent, identically distributed (i.i.d.) collection of fruit



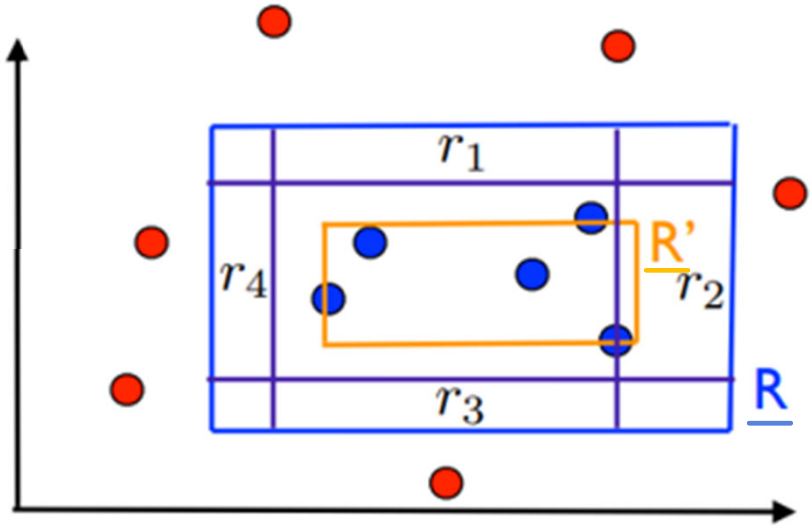
- Optimization
  - Discretization Trick
  - Search for tightest rectangle R' that separates data

PAC

- Problem: Suppose … predict … Does … ness impact … ness?

What is the probability that $R(R') > \epsilon$?

- Hypothesis Space:
  - Assume the target function corresponds to rectangle R with sides parallel to the axes.
- Sample
  - Draw independent, identically distributed (i.i.d.) collection of fruit



- Optimization
  - Discretization Trick
  - Search for tightest rectangle R' that separates data

PAC

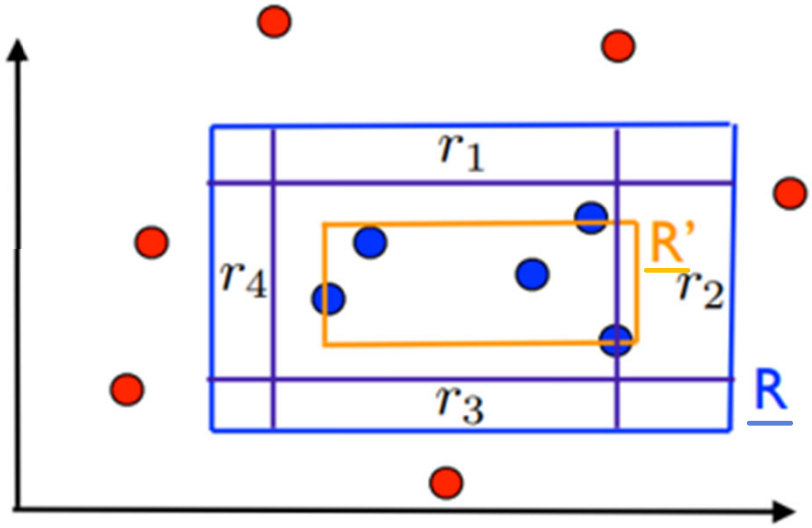
What is the probability that $R(R') > \epsilon$?

- Assume

$$\Pr_D[R] > \epsilon$$

where D is the distribution over color and firmness

- Let l,r denote the coordinates on the left, right and b,t denote the coordinates on the bottom, top

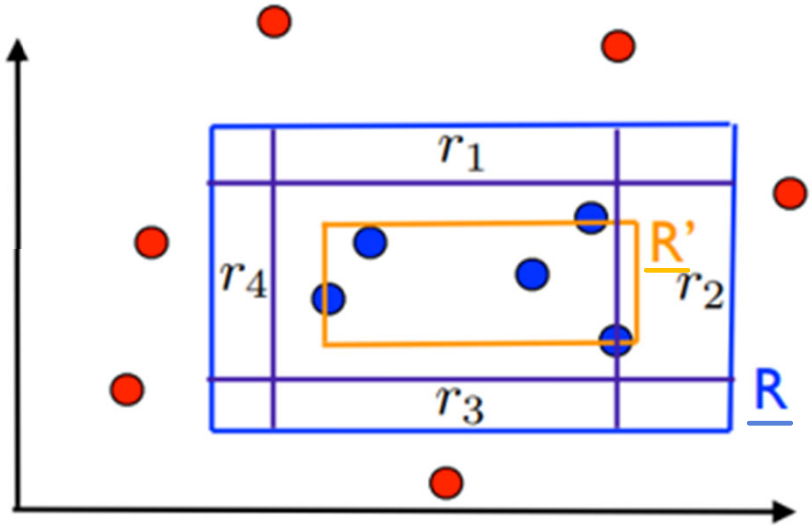


- Determine regions r_1, r_2, r_3, r_4

$$r_4 = [l, s_4] \times [b, t]$$
$$s_4 = \inf\{s : \Pr[[l, s] \times [b, t]] \geq \tfrac{\epsilon}{4}\}$$
$$\Pr_D[[l, s_4[\times [b, t]] < \tfrac{\epsilon}{4}$$

PAC

What is the probability that $R(R') > \epsilon$?



- Errors can only occur in R – R'. So

$$R(R') > \epsilon$$

implies

R′ misses at least one region r_i.

- Therefore

$$\begin{aligned}\Pr[R(R') > \epsilon] &\leq \Pr[\cup_{i=1}^{4}\{R' \text{ misses } r_i\}] \\ &\leq \sum_{i=1}^{4} \Pr[\{R' \text{ misses } r_i\}] \\ &\leq 4(1-\tfrac{\epsilon}{4})^m \leq 4e^{-\frac{m\epsilon}{4}}\end{aligned}$$

PAC

What is the probability that $R(R') > \epsilon$?

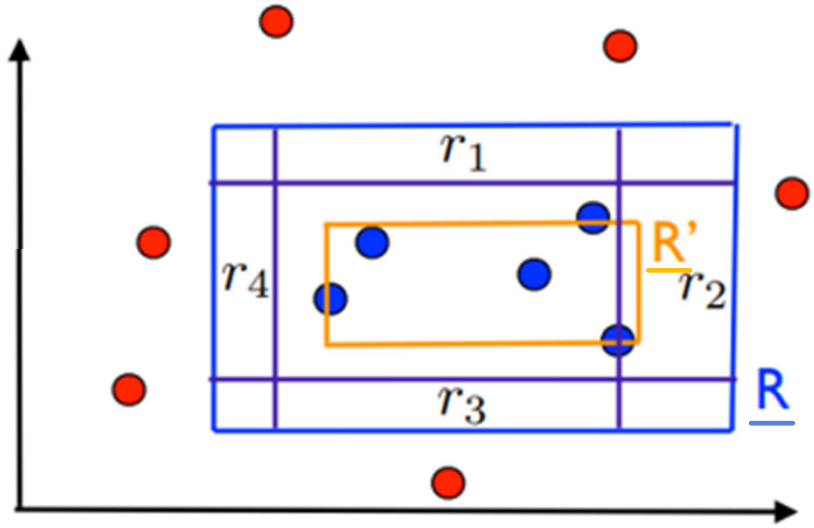
- Set $\delta > 0$ to match the bound

$$4e^{-\frac{m\epsilon}{4}} \leq \delta \Leftrightarrow m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}.$$

- For

$$m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}$$

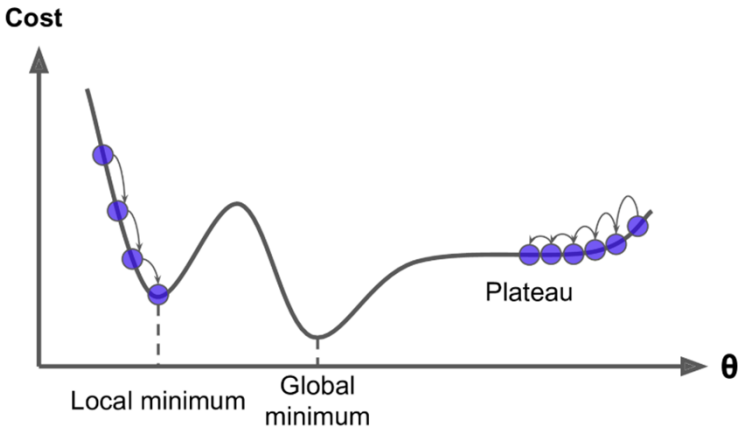
we probably (with respect to δ) have approximately (with respect to ε) have no statistical risk

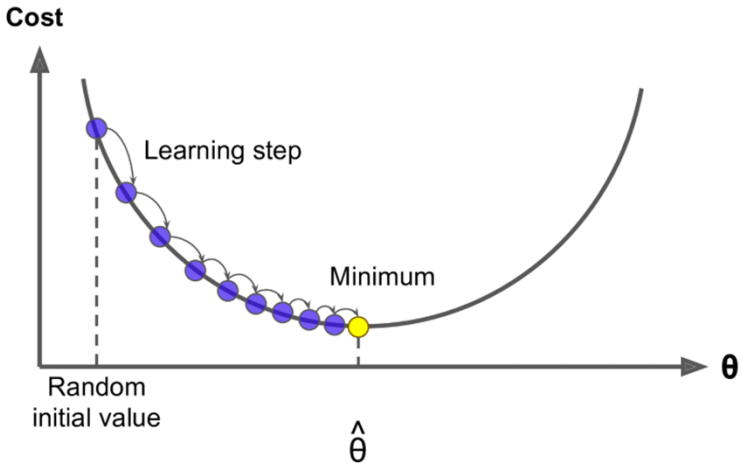


$$\Pr[R(R') > \epsilon] \leq \Pr[\cup_{i=1}^{4} \{R' \text{ misses } r_i\}]$$
$$\leq \sum_{i=1}^{4} \Pr[\{R' \text{ misses } r_i\}]$$
$$\leq 4(1 - \tfrac{\epsilon}{4})^m \leq 4e^{-\frac{m\epsilon}{4}}$$

Gradient Descent

- Gradient Descent
  - Guess and check method to iteratively find weights that minimize an objective function such as empirical risk

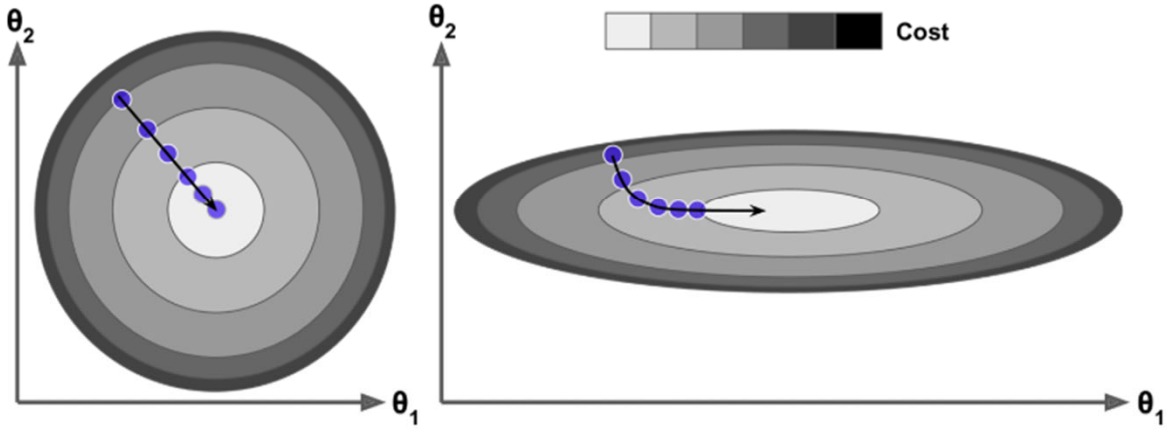




- Convexity
  - Gradient Descent works reliably with convex functions. Think second derivative is positive

Gradient Descent

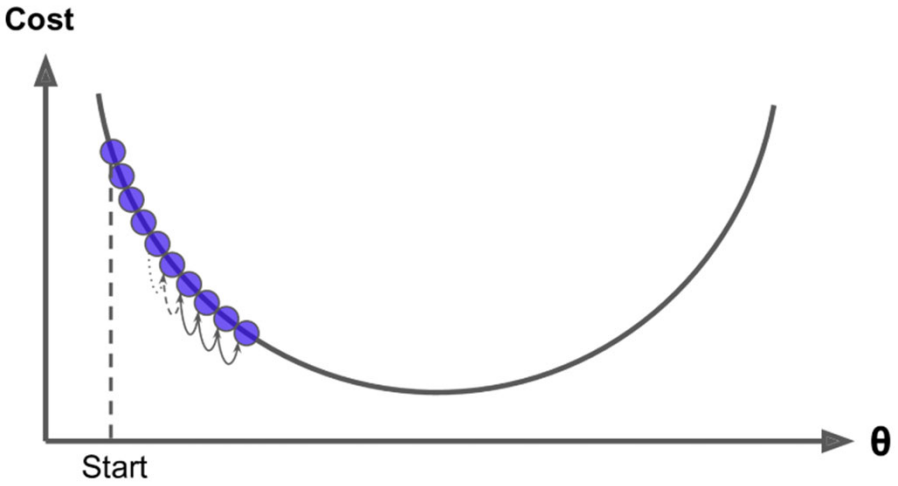
Gradient Descent

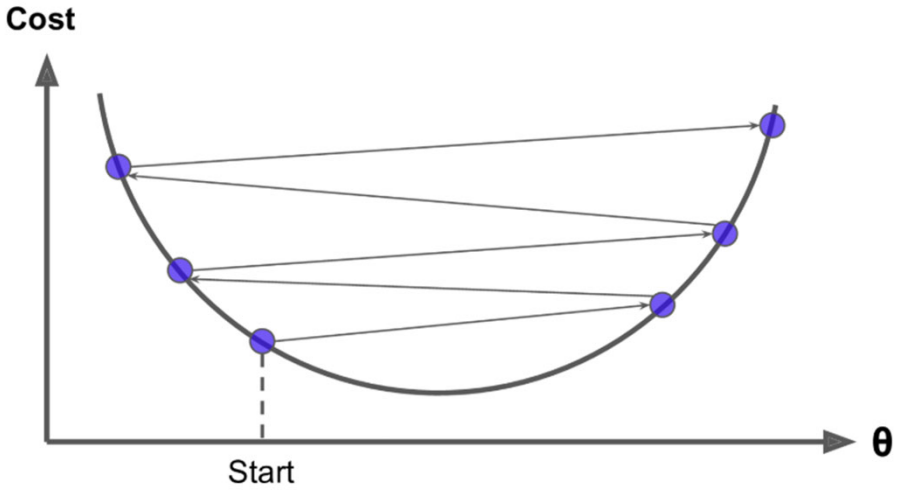


- Initialize $x = 0$
- repeat
  - $x \leftarrow x - \underbrace{\eta}_{\text{step size}} \nabla f(x)$
- until stopping criterion satisfied

Gradient Descent

- Learning Rate
  - Hyperparameter that controls the size of updates between iterations of gradient descent





- Convexity
  - Gradient Descent works reliably with convex functions. Think second derivative is positive

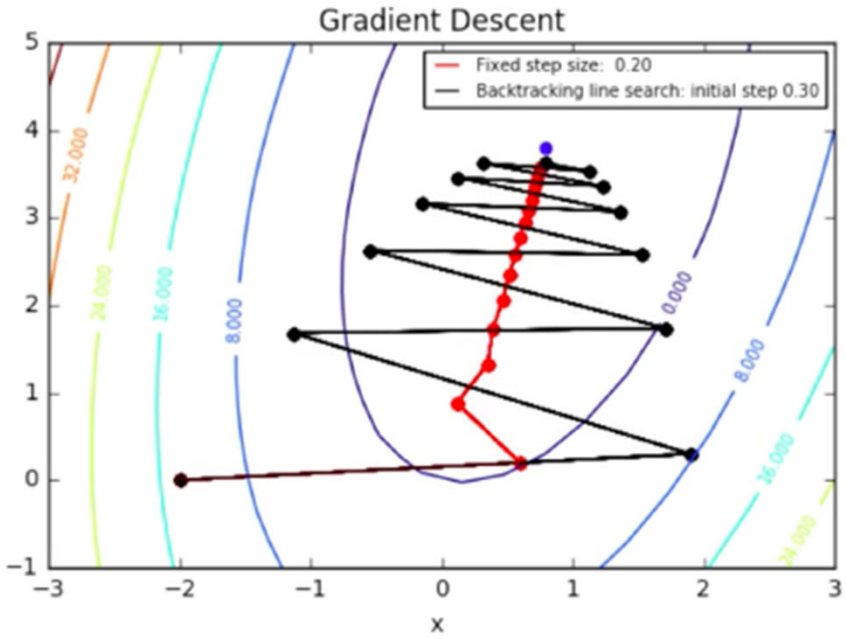
Gradient Descent

Given data set $\mathcal{D}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$,

$$\hat{R}_n(w) = \frac{1}{n} \sum_{i=1}^{n} \left(w^T x_i - y_i\right)^2,$$

giving for the derivative

$$\nabla \hat{R}_n(w) = \frac{1}{n} \sum_{i=1}^{n} \nabla_w \ell(f_w(x_i), y_i)$$



Note that the derivative is direction and rate of change from all points

Gradient Descent

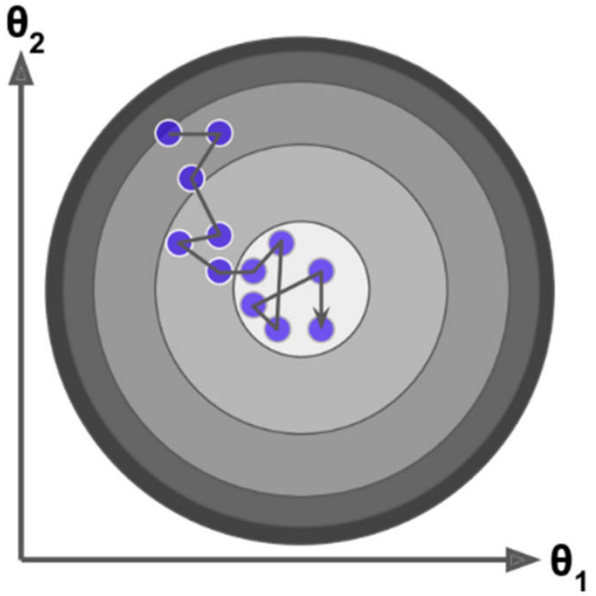
$$
\begin{aligned}
\mathbb{E}\left[\nabla \hat{R}_N(w)\right] &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\left[\nabla_w \ell(f_w(x_{m_i}), y_{m_i})\right] \\
&= \mathbb{E}\left[\nabla_w \ell(f_w(x_{m_1}), y_{m_1})\right] \\
&= \sum_{i=1}^{n} \mathbb{P}(m_1 = i)\, \nabla_w \ell(f_w(x_i), y_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} \nabla_w \ell(f_w(x_i), y_i) \\
&= \nabla \hat{R}_n(w)
\end{aligned}
$$

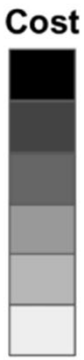
This shows that the estimate is unbiased

$$\mathbb{E}\left[\nabla \hat{R}_N(w)\right] = \nabla \hat{R}_n(w)$$

Stochastic Gradient Descent

Stochastic Gradient Descent

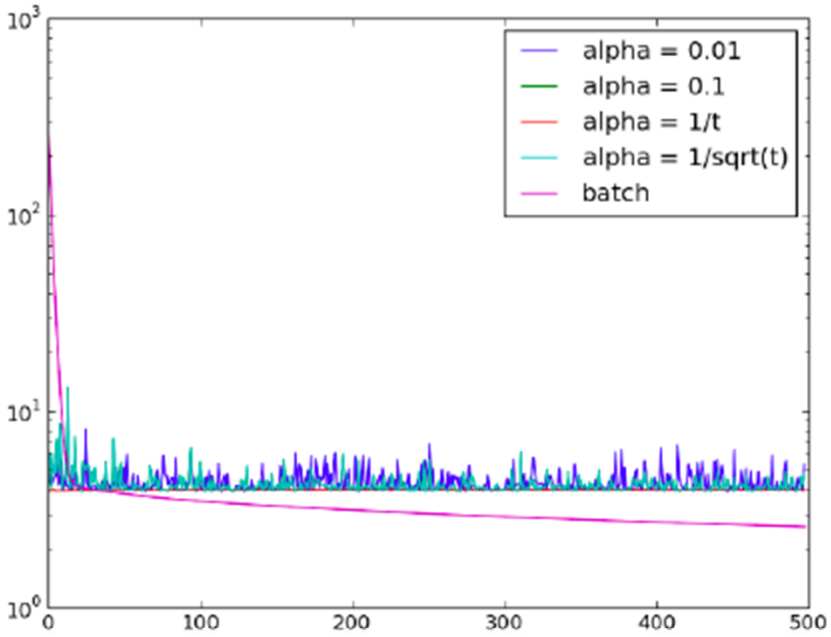




- initialize $w = 0$
- repeat
  - randomly choose training point $(x_i, y_i) \in \mathcal{D}_n$
  - $w \leftarrow w - \eta \underbrace{\nabla_w \ell(f_w(x_i), y_i)}_{\text{Grad(Loss on i'th example)}}$

Stochastic Gradient Descent

Minibatch Gradient Descent



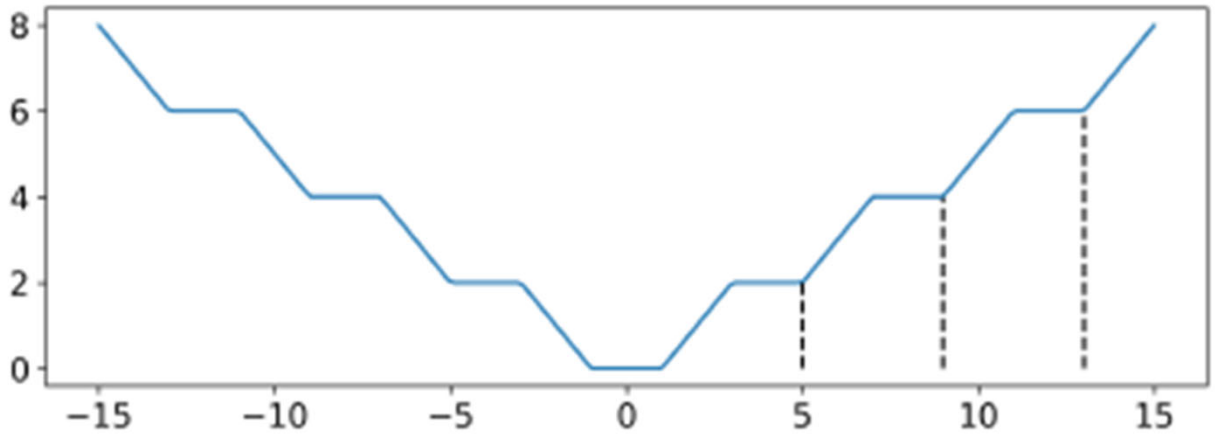
- initialize $w = 0$
- repeat
  - randomly choose N points $\{(x_i, y_i)\}_{i=1}^{N} \subset \mathcal{D}_n$
  - $w \leftarrow w - \eta \left[\frac{1}{N} \sum_{i=1}^{N} \nabla_w \ell(f_w(x_i), y_i) \right]$

Exercise

Momentum is a variation of gradient descent where we include the gradient at a previous iteration in the current iteration. The update rule is

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \alpha \frac{\partial L}{\partial \mathbf{w}} \left(\mathbf{w}^{(t)} \right) - \gamma \frac{\partial L}{\partial \mathbf{w}} \left(\mathbf{w}^{(t-1)} \right)$$

Here $\gamma > 0$ is the learning rate for the additional term. Assume for iteration $t = 0$ and $t = -1$, we set $\mathbf{w}^{(t)} = \texttt{t0}$ the initial guess.



Exercise

Assuming that $\mathbf{w}$ starts in a flat region that is not a minimum and $\alpha > 0$, will the basic gradient descent algorithm terminate at a minimum? Note that the basic gradient descent algorithm is just the same as version with momentum in the previous question, but where $\gamma = 0$.

- ☐ Yes with enough iterations
- ☐ Maybe
- ☐ Never

Assuming that $\mathbf{w}$ starts in a sloped region and $\alpha > 0$, will the basic gradient descent algorithm find the minimum?

- ☐ Yes with enoug
- ☐ Maybe
- ☐ Never

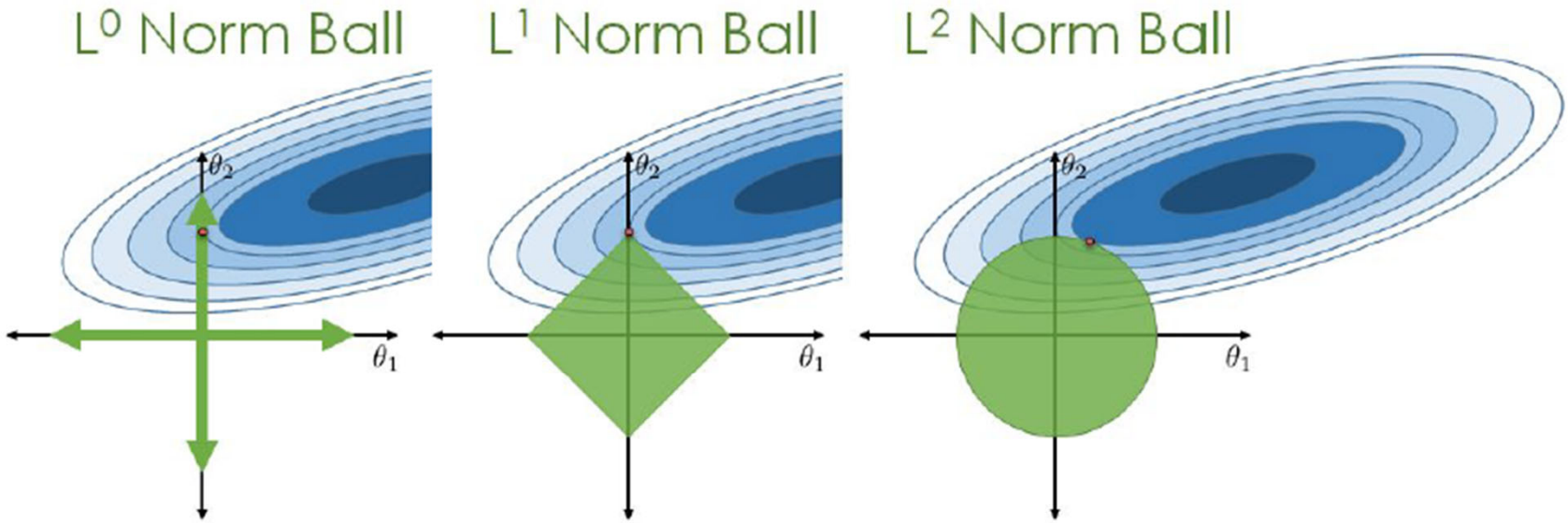
Assuming that $\mathbf{w}$ starts in a flat region that is not a minimum and $\alpha > 0$ and $\gamma > 0$, will the momentum gradient descent algorithm find the minimum?

- ☐ Yes with enough iterations
- ☐ Maybe
- ☐ Never

Assuming that $\mathbf{w}$ starts in a sloped region and $\alpha > 0$ and $\gamma > 0$, will the momentum gradient descent algorithm find the minimum?

- ☐ Yes with enough iterations
- ☐ Maybe
- ☐ Never

Regularization



Ideal for
Feature Selection
but combinatorically difficult to optimize

Encourages Sparse Solutions
Convex!

Spreads weight over features **(robust)**
does not encourage sparsity

Ridge Regression

Ridge Regression (Tikhonov Form)

The ridge regression solution for regularization parameter $\lambda \geqslant 0$ is

$$\hat{w} = \arg\min_{w \in \mathbf{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2 + \lambda \|w\|_2^2,$$

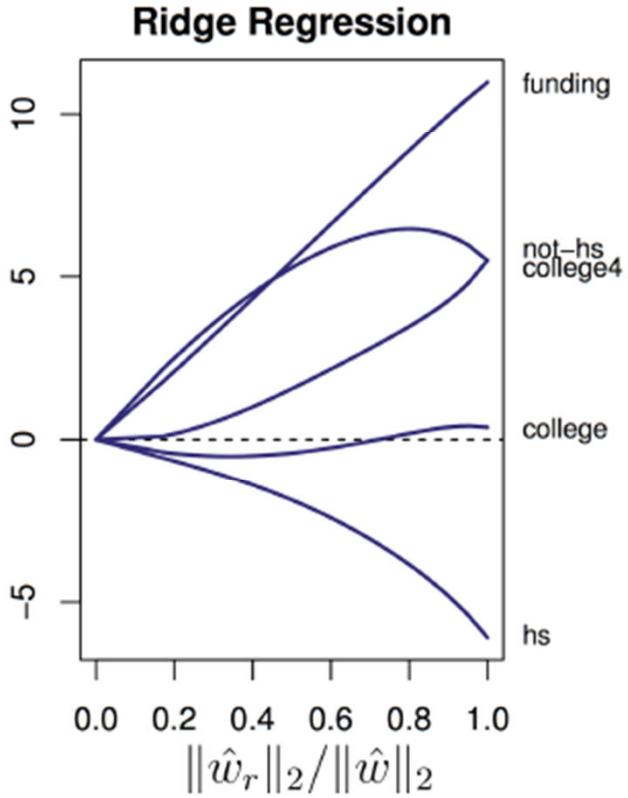
where $\|w\|_2^2 = w_1^2 + \cdots + w_d^2$ is the square of the ℓ_2-norm.

Ridge Regression (Ivanov Form)

The ridge regression solution for complexity parameter $r \geqslant 0$ is

$$\hat{w} = \arg\min_{\|w\|_2^2 \leqslant r^2} \frac{1}{n} \sum_{i=1}^{n} \left\{ w^T x_i - y_i \right\}^2.$$

Ridge Regression



For objective function

$$\underset{w\in\mathbb{R}^m}{\operatorname{argmin}} \left(\frac{1}{2m} \sum_{i=1}^{m} (x_i w - y_i)^2 + \lambda w^2 \right).$$

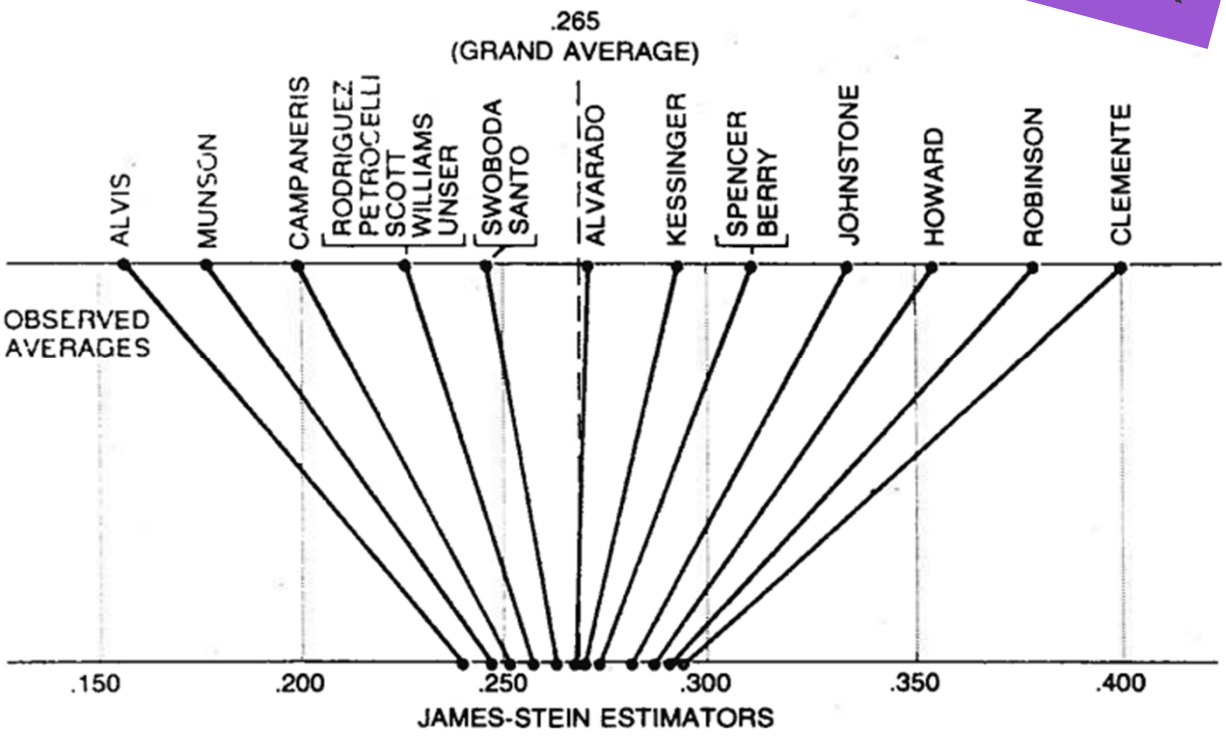
we can solve for the weights

$$w = \frac{\langle \mathbf{x}, \mathbf{y} \rangle / m}{\|\mathbf{x}\|^2/m + 2\lambda}.$$

Questions

Stein Paradox

- Questions on Piazza?
- Question for You!
  - Can you think of another way to use Perceptron for non-separable data



Questions

Stein Paradox

another
Perceptron for no
separable data