ECE595 / STAT598: Machine Learning I
Lecture 17 Perceptron 2: Algorithm and Property

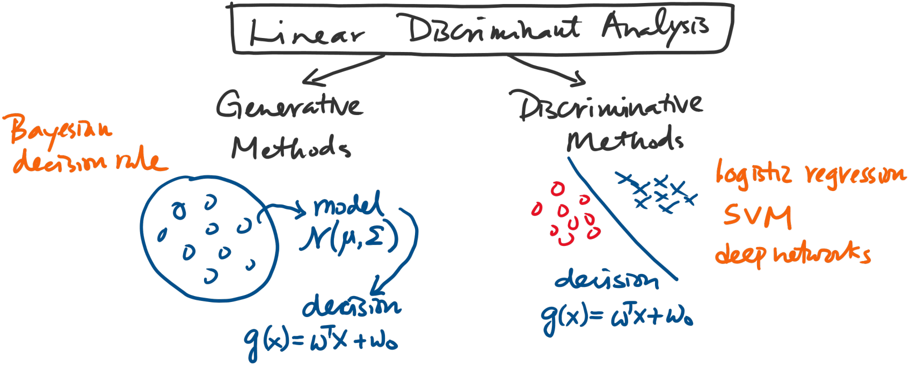
Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

© Stanley Chan 2020. All Rights Reserved.

Overview



- In linear discriminant analysis (LDA), there are generally two types of approaches
- **Generative approach**: Estimate model, then define the classifier
- **Discriminative approach**: Directly define the classifier

©Stanley Chan 2020. All Rights Reserved.

Outline

Discriminative Approaches

- Lecture 16 Perceptron 1: Definition and Basic Concepts
- Lecture 17 Perceptron 2: Algorithm and Property
- Lecture 18 Multi-Layer Perceptron: Back Propagation

This lecture: Perceptron 2

- Perceptron Algorithm
  - Loss Function
  - Algorithm
- Optimality
  - Uniqueness
  - Batch and Online Mode
- Convergence
  - Main Results
  - Implication

©Stanley Chan 2020. All Rights Reserved.

Perceptron with Hard Loss

- Historically, we have perceptron algorithm way earlier than CVX.
- Before the age of CVX, people solve perceptron using gradient descent.
- Let us be explicit about which loss:

$$J_{\text{hard}}(\boldsymbol{\theta}) = \sum_{j=1}^{N} \max\Big\{ -y_j h_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\}$$

$$J_{\text{soft}}(\boldsymbol{\theta}) = \sum_{j=1}^{N} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\}$$

- **Goal**: To get a solution for $J_{\text{hard}}(\boldsymbol{\theta})$
- **Approach**: Gradient descent on $J_{\text{soft}}(\boldsymbol{\theta})$

©Stanley Chan 2020. All Rights Reserved.

Re-defining the Loss

- **Main idea**: Use the fact that

$$J_{\text{soft}}(\boldsymbol{\theta}) = \sum_{j=1}^{N} \max\left\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \right\}$$

  is the same as this loss function

$$J(\boldsymbol{\theta}) = -\sum_{j \in \mathcal{M}(\boldsymbol{\theta})} y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j).$$

- $\mathcal{M}(\boldsymbol{\theta}) \subseteq \{1, \ldots, N\}$ is the set of misclassified samples.
- Run gradient descent on $J(\boldsymbol{\theta})$, but fixing $\mathcal{M}(\boldsymbol{\theta}) \leftarrow \mathcal{M}(\boldsymbol{\theta}^k)$ for iteration k.

©Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- We want to show that the perceptron loss function is equivalent to

$$\underbrace{\sum_{j=1}^{N} \max\Big\{ - y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\}}_{J_{\text{soft}}(\boldsymbol{\theta})} = \underbrace{- \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j)}_{J(\boldsymbol{\theta})}$$

- If $\boldsymbol{x}_j$ is misclassified ($j \in \mathcal{M}(\boldsymbol{\theta})$)
  - then by definition of $\mathcal{M}(\boldsymbol{\theta})$ we have $\text{sign}\,\{g_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\} \neq y_j$
  - So $-y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) > 0$
  - Therefore, $\max\{-y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j),\ 0\} = -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j)$.
- If $\boldsymbol{x}_j$ is correctly classified ($j \notin \mathcal{M}(\boldsymbol{\theta})$)
  - then by definition of $\mathcal{M}(\boldsymbol{\theta})$ we have $\text{sign}\,\{g_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\} = y_j$
  - So $-y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0$
  - Therefore, $\max\{-y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j),\ 0\} = 0$.

©Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- Therefore, we conclude that

$$\mathcal{M}(\boldsymbol{\theta}) = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$$

© Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- Therefore, we conclude that

$$\mathcal{M}(\boldsymbol{\theta}) = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$$

- and

$$\begin{aligned} J_{\text{soft}}(\boldsymbol{\theta}) &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} 0 \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = J(\boldsymbol{\theta}). \end{aligned}$$

©Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- Therefore, we conclude that

$$\mathcal{M}(\boldsymbol{\theta}) = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$$

- and

$$\begin{aligned}
J_{\text{soft}}(\boldsymbol{\theta}) &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ - y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ - y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} \\
&= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} 0 \\
&= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = J(\boldsymbol{\theta}).
\end{aligned}$$

©Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- Therefore, we conclude that

$$\mathcal{M}(\boldsymbol{\theta}) = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$$

- and

$$\begin{aligned} J_{\text{soft}}(\boldsymbol{\theta}) &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} 0 \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = J(\boldsymbol{\theta}). \end{aligned}$$

©Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- Therefore, we conclude that

$$\mathcal{M}(\boldsymbol{\theta}) = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$$

- and

$$\begin{aligned} J_{\text{soft}}(\boldsymbol{\theta}) &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} 0 \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = J(\boldsymbol{\theta}). \end{aligned}$$

© Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- Therefore, we conclude that

$$\mathcal{M}(\boldsymbol{\theta}) = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$$

- and

$$\begin{aligned} J_{\text{soft}}(\boldsymbol{\theta}) &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} 0 \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = J(\boldsymbol{\theta}). \end{aligned}$$

- Minimizing $J(\boldsymbol{\theta})$ is less obvious because $\mathcal{M}(\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$.

©Stanley Chan 2020. All Rights Reserved.

Equivalent Perceptron Loss

- Therefore, we conclude that

$$\mathcal{M}(\boldsymbol{\theta}) = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$$

- and

$$\begin{aligned} J_{\text{soft}}(\boldsymbol{\theta}) &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} \max\Big\{ -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j), 0 \Big\} \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) + \sum_{j \notin \mathcal{M}(\boldsymbol{\theta})} 0 \\ &= \sum_{j \in \mathcal{M}(\boldsymbol{\theta})} -y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = J(\boldsymbol{\theta}). \end{aligned}$$

- Minimizing $J(\boldsymbol{\theta})$ is less obvious because $\mathcal{M}(\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$.
- But it gives a very easy algorithm.

©Stanley Chan 2020. All Rights Reserved.

Perceptron Algorithm

- The loss is

$$J(\boldsymbol{\theta}) = -\sum_{j \in \mathcal{M}(\boldsymbol{\theta})} y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j),$$

- At iteration k, fix $\mathcal{M}_k = \mathcal{M}(\boldsymbol{\theta}^{(k)})$
- Then, update via gradient descent

$$\begin{aligned} \boldsymbol{\theta}^{(k+1)} &= \boldsymbol{\theta}^{(k)} - \alpha_k \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}^{(k)}) \\ &= \boldsymbol{\theta}^{(k)} - \alpha_k \sum_{j \in \mathcal{M}_k} \nabla_{\boldsymbol{\theta}} \Big(- y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) \Big). \end{aligned}$$

©Stanley Chan 2020. All Rights Reserved.

Perceptron Algorithm

- We can show that

$$\nabla_{\boldsymbol{\theta}}\Big(-y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\Big) = \begin{cases} -y_j \nabla_{\boldsymbol{\theta}}\Big(\boldsymbol{w}^T \boldsymbol{x}_j + w_0\Big) & , \\ 0, & , \end{cases}$$

$$= \begin{cases} = -y_j \begin{bmatrix} \boldsymbol{x}_j \\ 1 \end{bmatrix} & \text{if} \quad j \in \mathcal{M}_k, \\ 0, & \text{if} \quad j \notin \mathcal{M}_k. \end{cases}$$

- Thus, the update is

$$\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ w_0^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}^{(k)} \\ w_0^{(k)} \end{bmatrix} + \alpha_k \sum_{j \in \mathcal{M}_k} \begin{bmatrix} y_j \boldsymbol{x}_j \\ y_j \end{bmatrix}.$$

©Stanley Chan 2020. All Rights Reserved.

Perceptron Algorithm

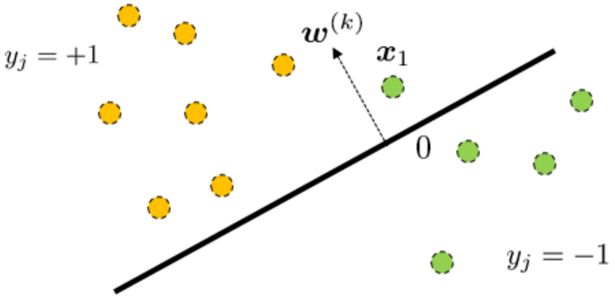
- The algorithm is
- For $k = 1, 2, \ldots,$
- Update $\mathcal{M}_k = \{j \mid y_j g_{\boldsymbol{\theta}}(\boldsymbol{x}_j) < 0\}$ for $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$.
- Gradient descent

$$\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ w_0^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}^{(k)} \\ w_0^{(k)} \end{bmatrix} + \alpha_k \sum_{j \in \mathcal{M}_k} \begin{bmatrix} y_j \boldsymbol{x}_j \\ y_j \end{bmatrix}.$$

- End For
- The set $\mathcal{M}_k$ can grow or can shrink from $\mathcal{M}_{k-1}$.
- If training samples are linearly separable, then converge. Zero training loss.
- If training samples are not linearly separable, then oscillates.

©Stanley Chan 2020. All Rights Reserved.

Updating One Sample



©Stanley Chan 2020. All Rights Reserved.

Outline

Discriminative Approaches

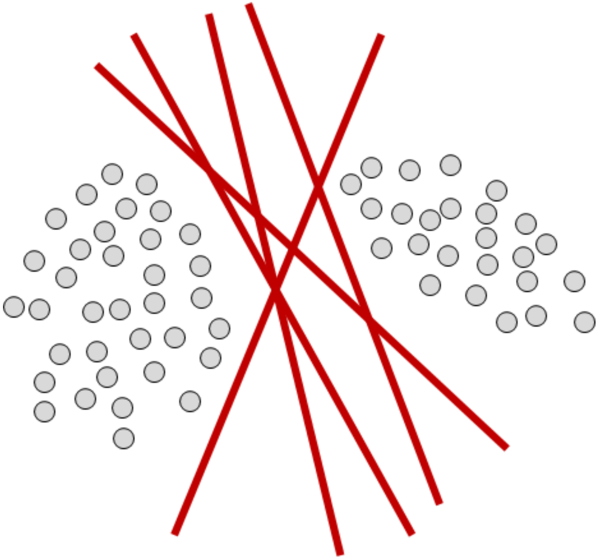
- Lecture 16 Perceptron 1: Definition and Basic Concepts
- Lecture 17 Perceptron 2: Algorithm and Property
- Lecture 18 Multi-Layer Perceptron: Back Propagation

This lecture: Perceptron 2

- Perceptron Algorithm
  - Loss Function
  - Algorithm
- Optimality
  - Uniqueness
  - Batch and Online Mode
- Convergence
  - Main Results
  - Implication

©Stanley Chan 2020. All Rights Reserved.

Non-uniqueness of Global Minimizer



©Stanley Chan 2020. All Rights Reserved.

Optimality of Perceptron Algorithm

- Let **perceptron algorithm** output

$$\boldsymbol{\theta}^*_{\text{perceptron}} = \text{Perceptron Algorithm}(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}).$$

- Let **ideal** solution

$$\boldsymbol{\theta}^*_{\text{hard}} = \underset{\boldsymbol{\theta}}{\text{argmin}} \;\; J_{\text{hard}}(\boldsymbol{\theta}).$$

  That means

$$J_{\text{hard}}(\boldsymbol{\theta}^*_{\text{hard}}) \leq J_{\text{hard}}(\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta}.$$

- If the two classes are linearly separable, then $\boldsymbol{\theta}^*_{\text{perceptron}}$ is a global minimizer:

$$J_{\text{hard}}(\boldsymbol{\theta}^*_{\text{perceptron}}) \leq J_{\text{hard}}(\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta}.$$

  and

$$J_{\text{hard}}(\boldsymbol{\theta}^*_{\text{perceptron}}) = J_{\text{hard}}(\boldsymbol{\theta}^*_{\text{hard}}) = 0.$$

©Stanley Chan 2020. All Rights Reserved.

Uniqueness of Perceptron Solution

- If $\boldsymbol{\theta}^*$ minimizes $J_{\text{hard}}(\boldsymbol{\theta}^*)$, then $\alpha\boldsymbol{\theta}^*$ for some constant $\alpha > 0$ also minimizes $J_{\text{hard}}(\boldsymbol{\theta}^*)$.
- This is because

$$
\begin{aligned}
g_{\alpha\boldsymbol{\theta}}(\boldsymbol{x}) &= (\alpha\boldsymbol{w})^T\boldsymbol{x} + (\alpha w_0) \\
&= \alpha(\boldsymbol{w}^T\boldsymbol{x} + w_0).
\end{aligned}
$$

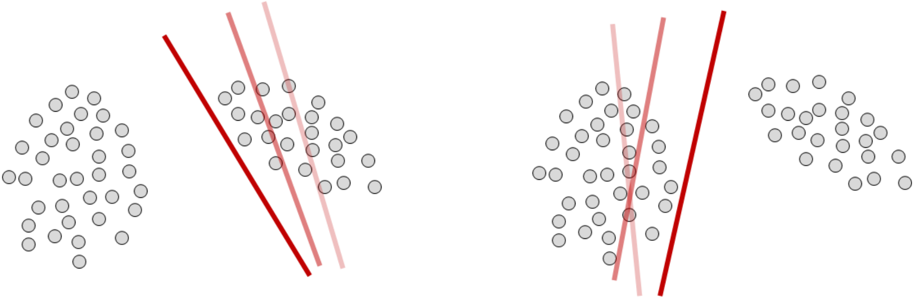
- If $g_{\boldsymbol{\theta}}(\boldsymbol{x}) > 0$, then $g_{\alpha\boldsymbol{\theta}}(\boldsymbol{x}) > 0$. So if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = +1$, then $h_{\alpha\boldsymbol{\theta}}(\boldsymbol{x}) = +1$.
- If $g_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0$, then $g_{\alpha\boldsymbol{\theta}}(\boldsymbol{x}) < 0$. So if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = -1$, then $h_{\alpha\boldsymbol{\theta}}(\boldsymbol{x}) = -1$.
- The sign of $\boldsymbol{w}^T\boldsymbol{x} + w_0$ is unchanged as long as $\alpha > 0$.

$$
\begin{aligned}
J_{\text{hard}}(\boldsymbol{\theta}^*) &= \sum_{j=1}^{N} \max\left\{ -y_j h_{\boldsymbol{\theta}^*}(\boldsymbol{x}_j), 0 \right\} \\
&= \sum_{j=1}^{N} \max\left\{ -y_j h_{\alpha\boldsymbol{\theta}^*}(\boldsymbol{x}_j), 0 \right\} = J_{\text{hard}}(\alpha\boldsymbol{\theta}^*)
\end{aligned}
$$

©Stanley Chan 2020. All Rights Reserved.

Factors for Uniqueness

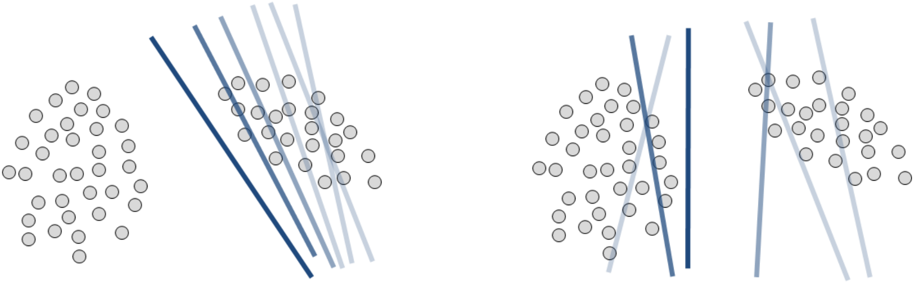
- **Initialization**
- Start at a different location, end on a different location
- You still converge, but no longer unique solution
- $\mathcal{M}_k$ changes



© Stanley Chan 2020. All Rights Reserved.

Factors for Uniqueness

- **Step Size**
- Too large step: oscillate
- Too small step: slow movement
- Terminates as long as no misclassification



©Stanley Chan 2020. All Rights Reserved.

Batch vs Online Mode

- **Batch mode**

$$\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ w_0^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}^{(k)} \\ w_0^{(k)} \end{bmatrix} + \alpha_k \sum_{j \in \mathcal{M}_k} \begin{bmatrix} y_j \boldsymbol{x}_j \\ y_j \end{bmatrix}.$$

  Update via the average of misclassified samples

- **Online mode**

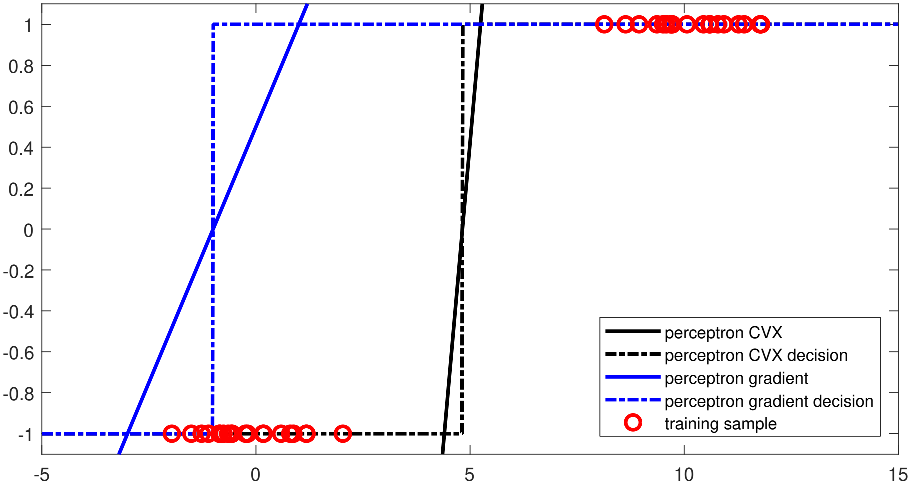
$$\begin{bmatrix} \boldsymbol{w}^{(k+1)} \\ w_0^{(k+1)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}^{(k)} \\ w_0^{(k)} \end{bmatrix} + \alpha_k \begin{bmatrix} y_j \boldsymbol{x}_j \\ y_j \end{bmatrix},$$

  Update via a single misclassified sample

- j is a sample randomly picked from $\mathcal{M}_k$.
- Stochastic gradient descent.

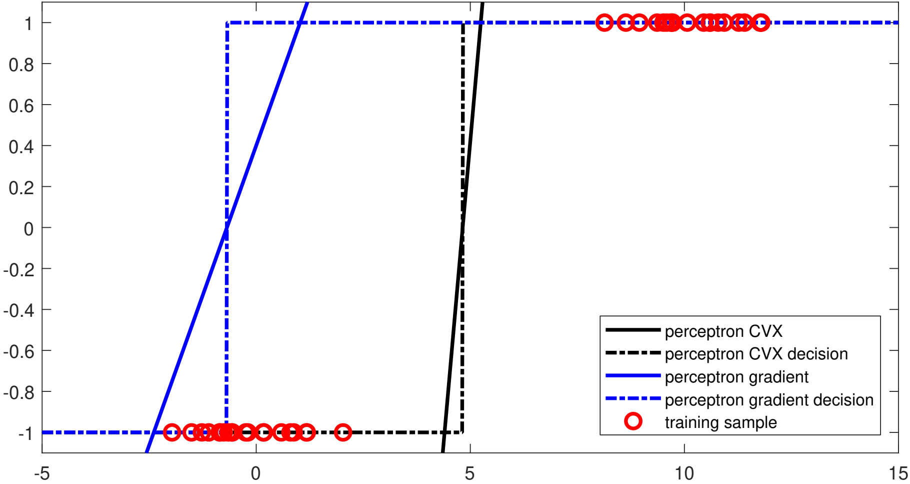
©Stanley Chan 2020. All Rights Reserved.

Online Mode



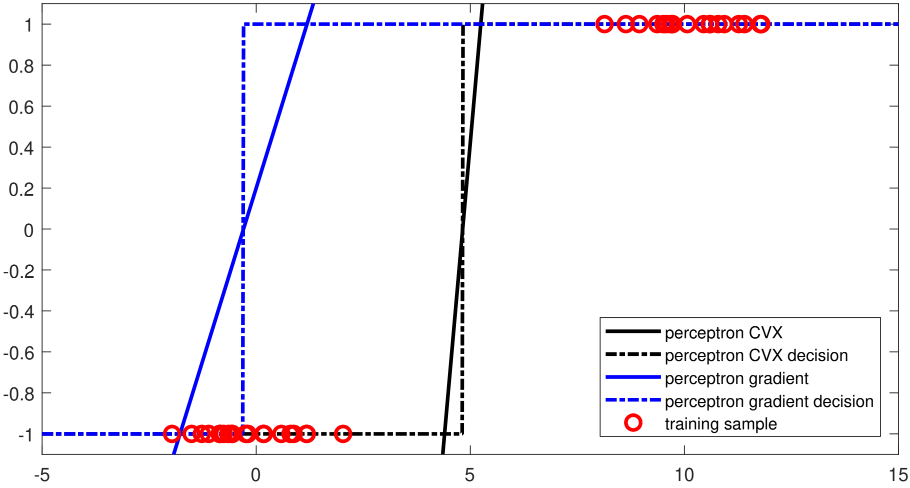
©Stanley Chan 2020. All Rights Reserved.

Online Mode



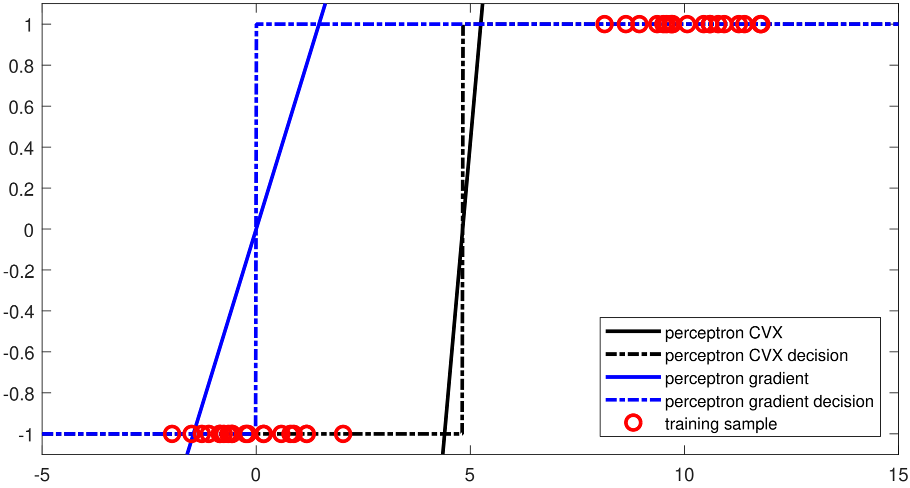
©Stanley Chan 2020. All Rights Reserved.

Online Mode



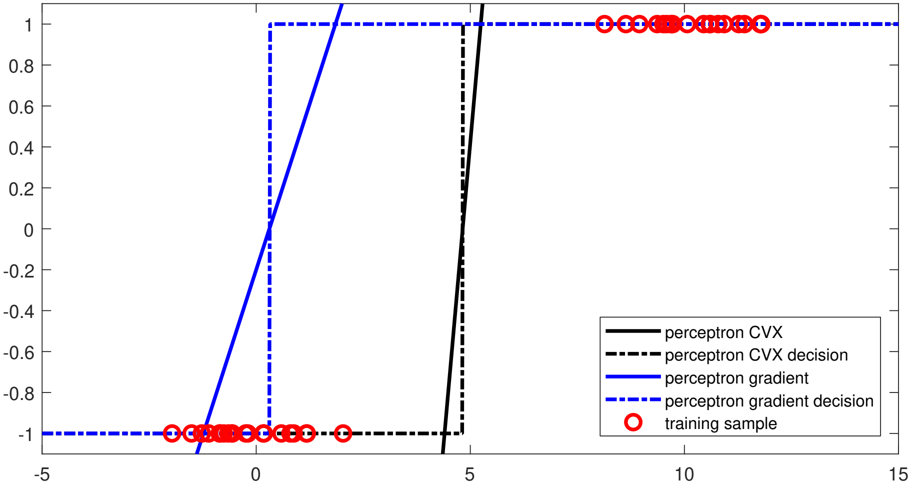
©Stanley Chan 2020. All Rights Reserved.

Online Mode



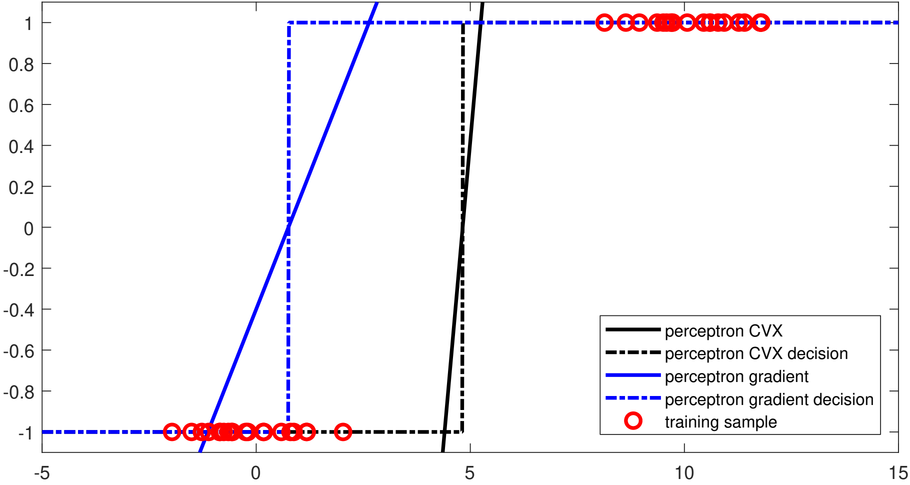
©Stanley Chan 2020. All Rights Reserved.

Online Mode



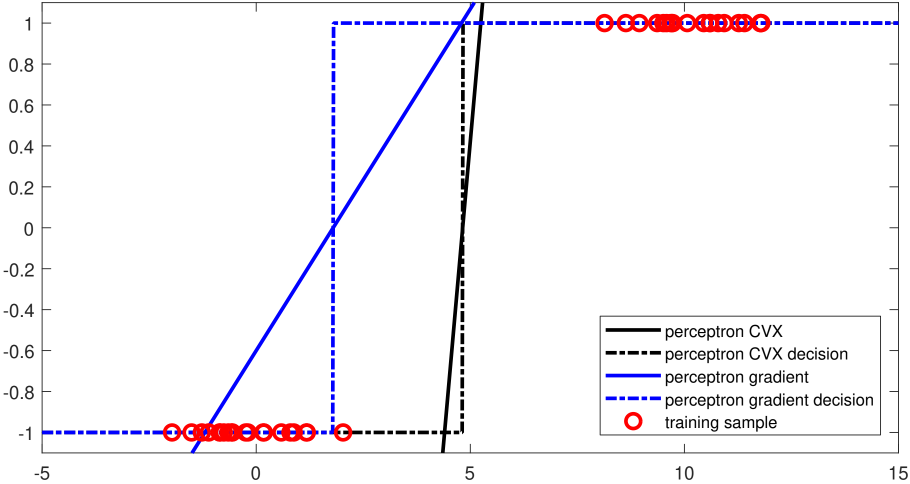
©Stanley Chan 2020. All Rights Reserved.

Online Mode



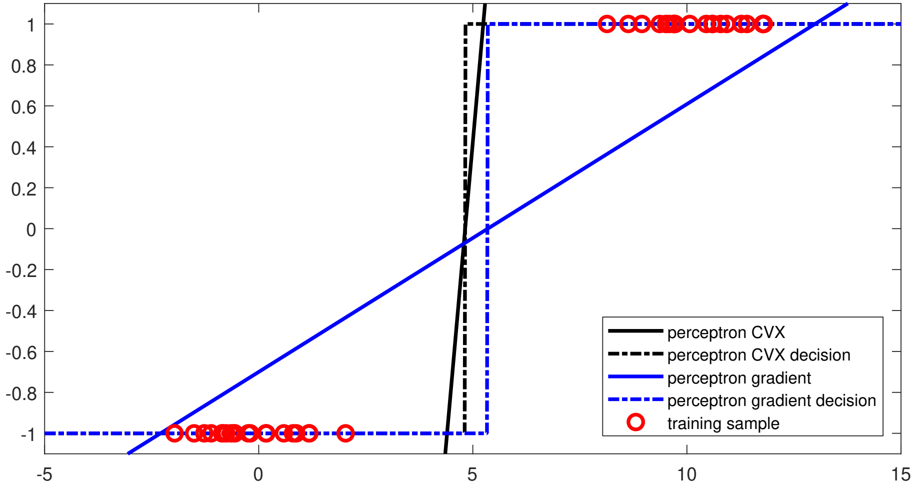
©Stanley Chan 2020. All Rights Reserved.

Online Mode



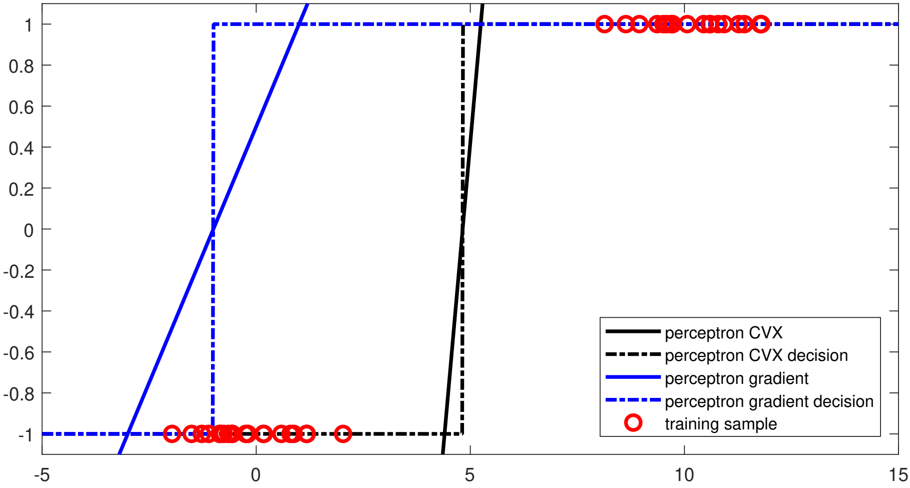
©Stanley Chan 2020. All Rights Reserved.

Online Mode



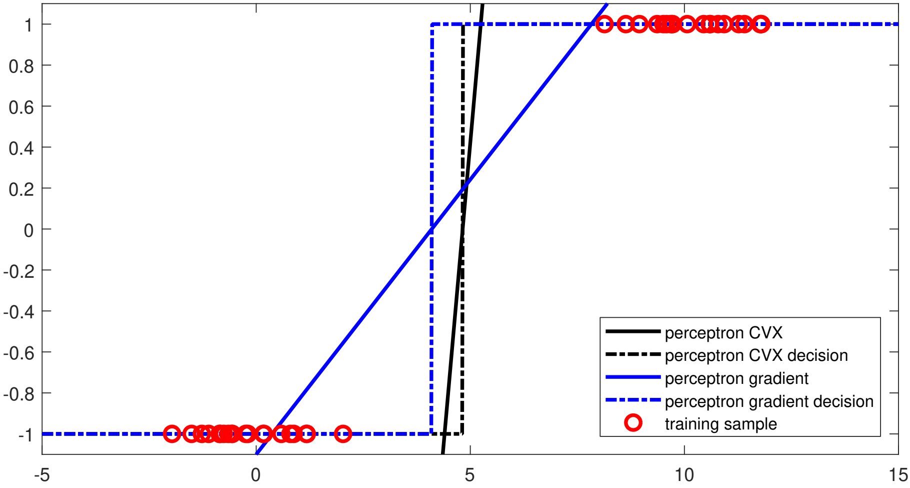
©Stanley Chan 2020. All Rights Reserved.

Batch Mode



© Stanley Chan 2020. All Rights Reserved.

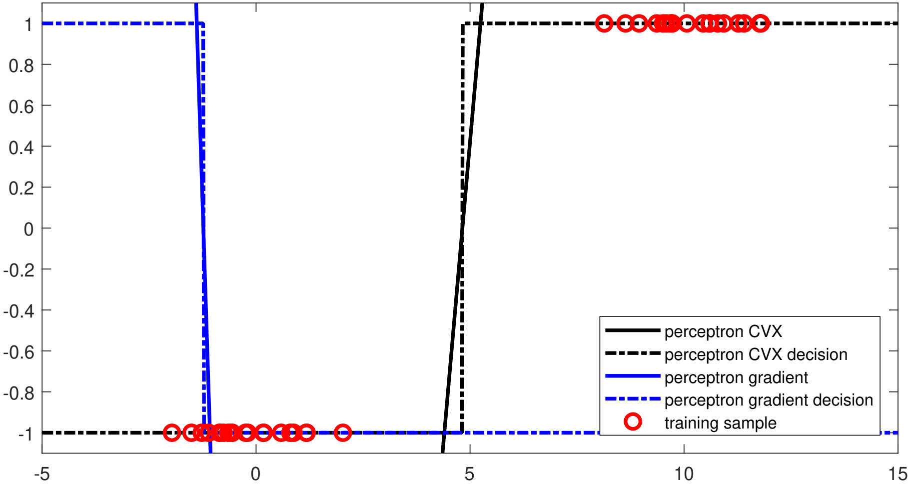
Batch Mode



© Stanley Chan 2020. All Rights Reserved.

Step Size

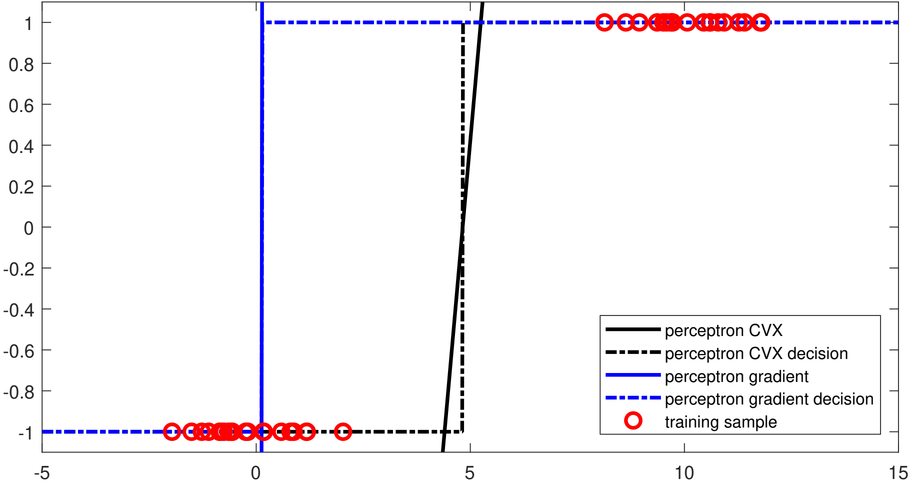
Batch mode: Step size too large.



ⓒStanley Chan 2020. All Rights Reserved.

Step Size

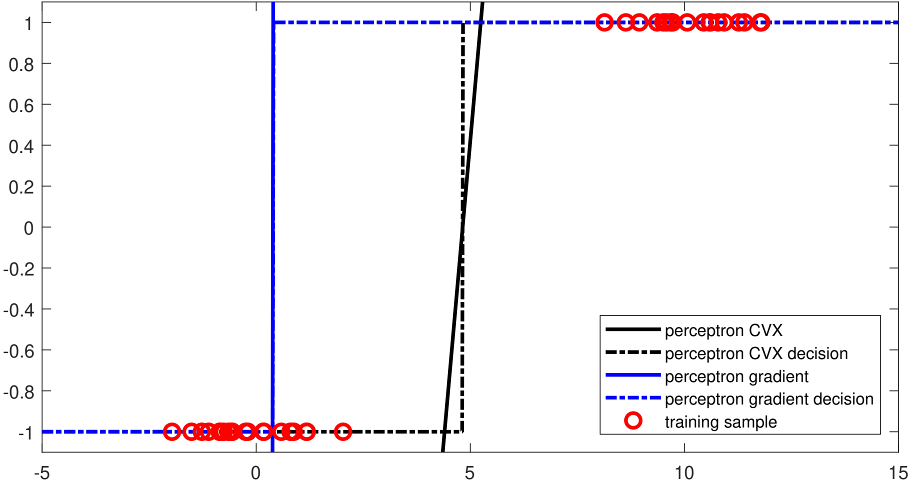
Batch mode: Step size too large.



©Stanley Chan 2020. All Rights Reserved.

Step Size

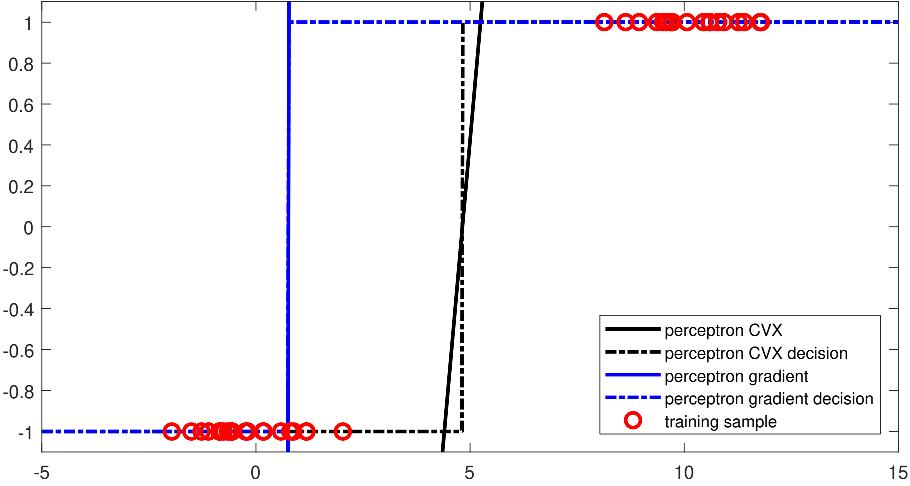
Batch mode: Step size too large.



©Stanley Chan 2020. All Rights Reserved.

Step Size

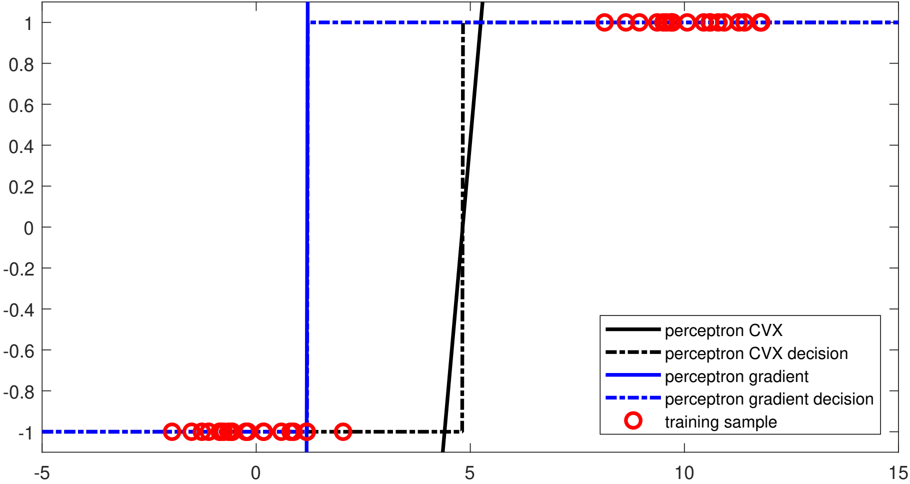
Batch mode: Step size too large.



©Stanley Chan 2020. All Rights Reserved.

Step Size

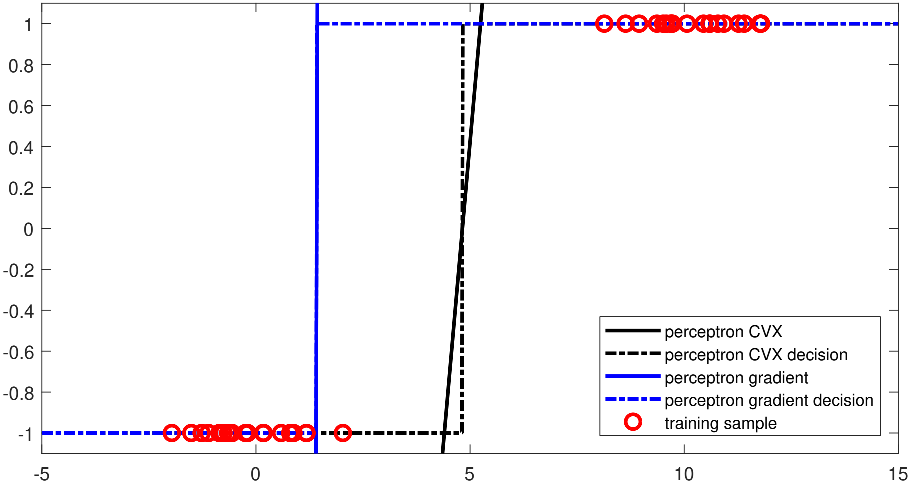
Batch mode: Step size too large.



©Stanley Chan 2020. All Rights Reserved.

Step Size

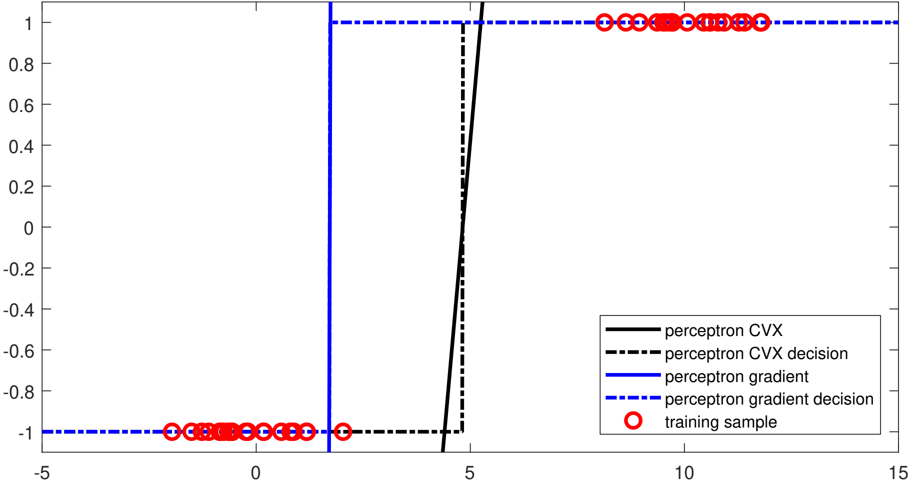
Batch mode: Step size too large.



ⓒStanley Chan 2020. All Rights Reserved.

Step Size

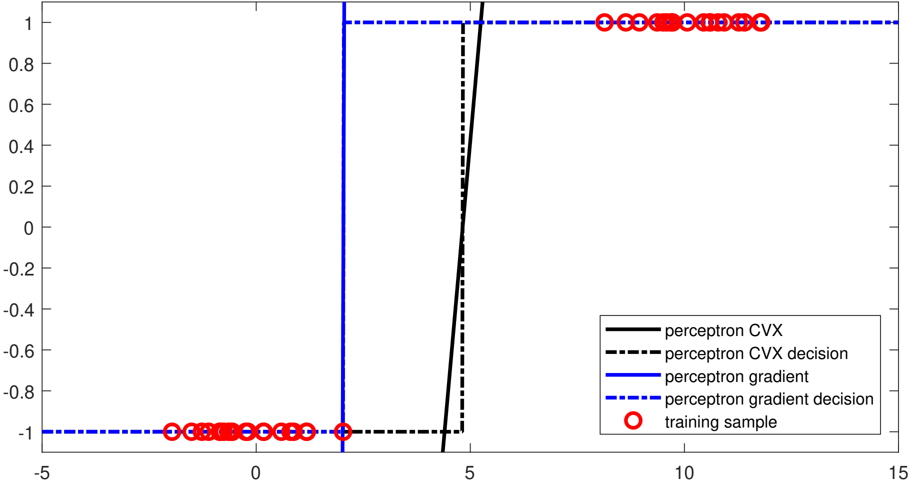
Batch mode: Step size too large.



©Stanley Chan 2020. All Rights Reserved.

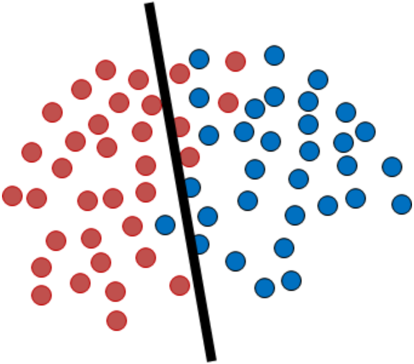
Step Size

Batch mode: Step size too large.



© Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

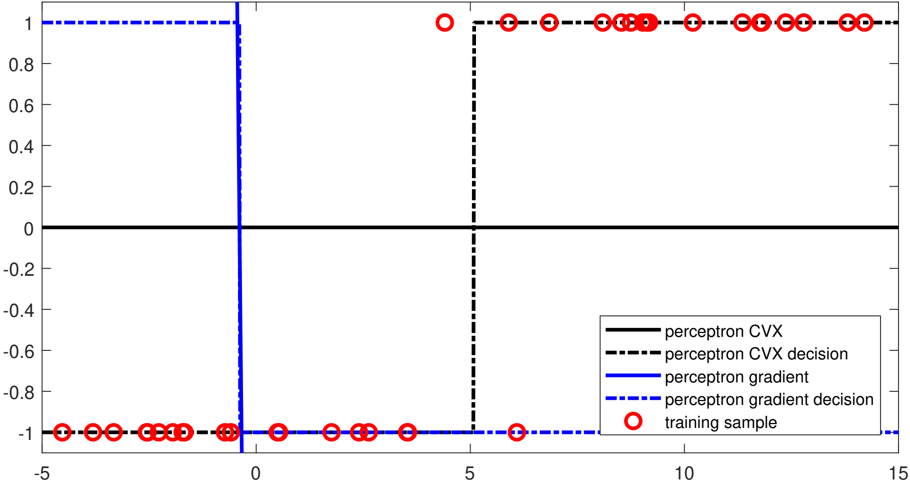


- No separating hyperplane
- CVX will still find you a solution
- But loss is no longer zero
- Perceptron algorithm will not converge

©Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

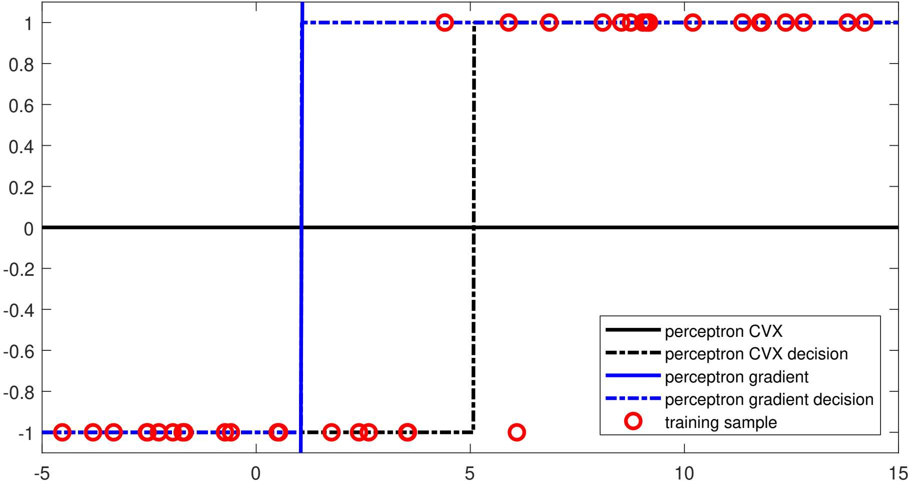
If the two classes are overlapping



©Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

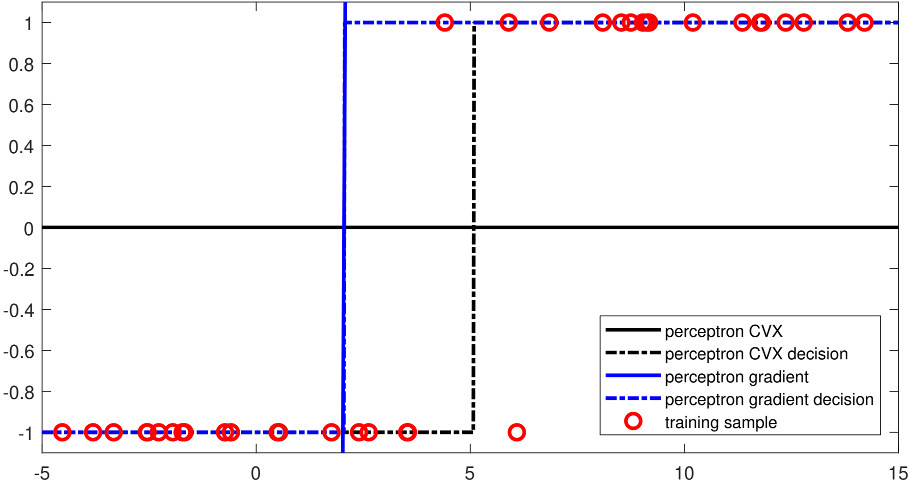
If the two classes are overlapping



©Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

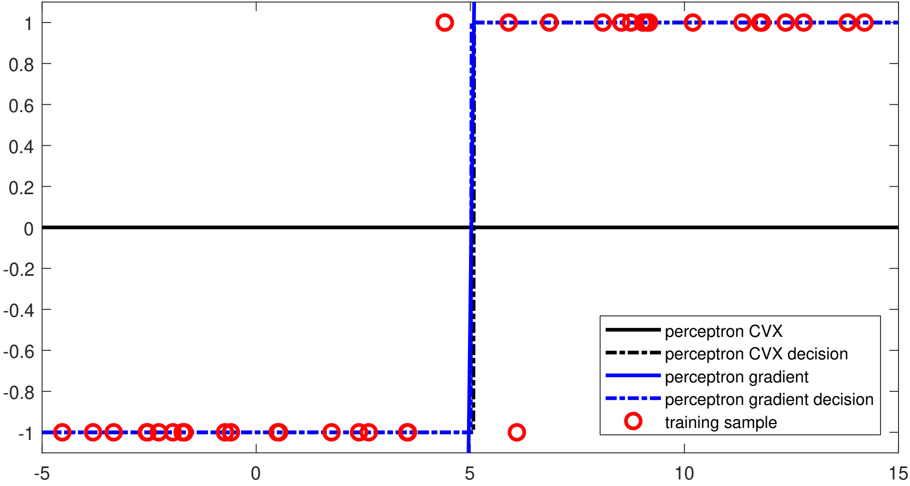
If the two classes are overlapping



©Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

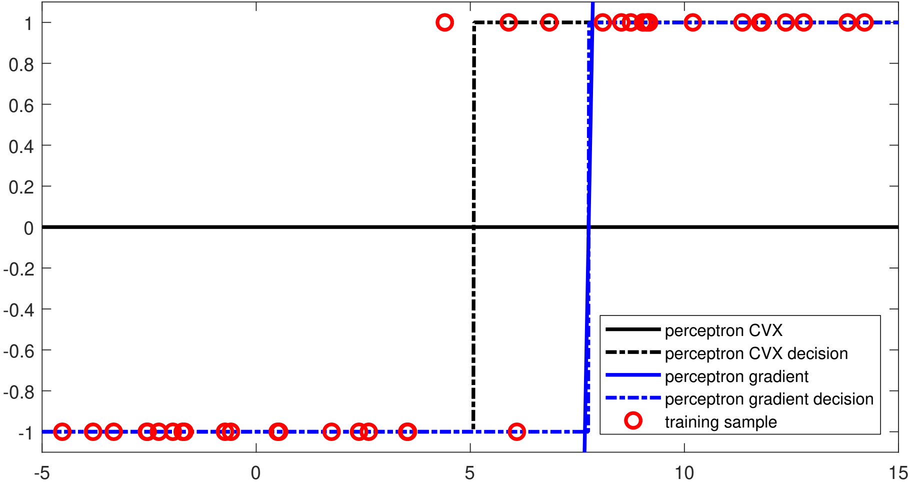
If the two classes are overlapping



©Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

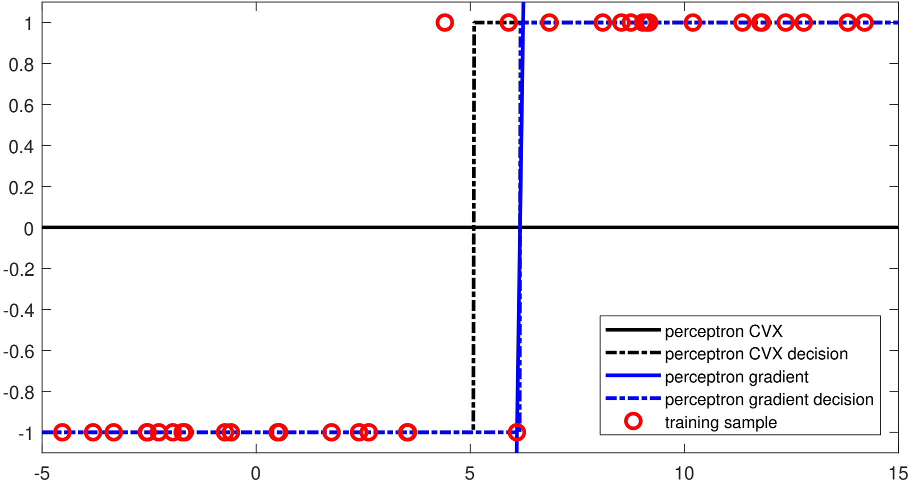
If the two classes are overlapping



©Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

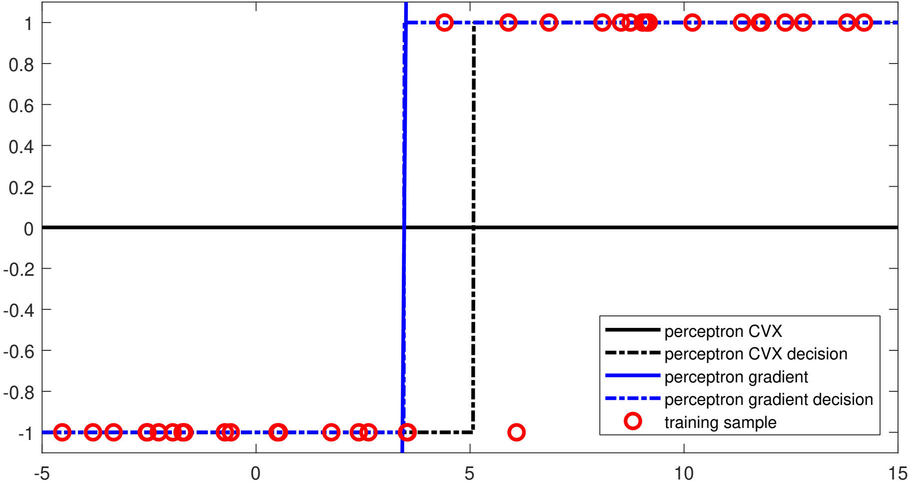
If the two classes are overlapping



©Stanley Chan 2020. All Rights Reserved.

Linearly Not Separable

If the two classes are overlapping



©Stanley Chan 2020. All Rights Reserved.

Outline

Discriminative Approaches

- Lecture 16 Perceptron 1: Definition and Basic Concepts
- Lecture 17 Perceptron 2: Algorithm and Property
- Lecture 18 Multi-Layer Perceptron: Back Propagation

This lecture: Perceptron 2

- Perceptron Algorithm
  - Loss Function
  - Algorithm
- Optimality
  - Uniqueness
  - Batch and Online Mode
- Convergence
  - Main Results
  - Implication

© Stanley Chan 2020. All Rights Reserved.

Convergence of Perceptron Algorithm

Theorem. Assume the following things:

- The two classes are linearly separable
- This means: $(\boldsymbol{\theta}^*)^T(y_j \boldsymbol{x}_j) = y_j((\boldsymbol{w}^*)^T \boldsymbol{x}_j + w_0^*) \geq \gamma$ for some $\gamma > 0$
- $\|\boldsymbol{x}_j\|_2 \leq R$ for some constant
- Initialize $\boldsymbol{\theta}^{(0)} = \mathbf{0}$

Then, batch mode perceptron algorithm converges to the true solution $\boldsymbol{\theta}^*$

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 = 0,$$

when the number of iterations k exceeds

$$k \geq \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}.$$

©Stanley Chan 2020. All Rights Reserved.

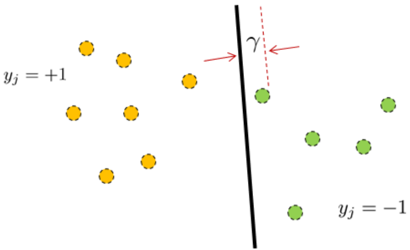
Interpreting the Perceptron Convergence

Theorem. Assume the following things:

- The two classes are linearly separable
- This means: $(\boldsymbol{\theta}^*)^T(y_j\boldsymbol{x}_j) = y_j((\boldsymbol{w}^*)^T\boldsymbol{x}_j + w_0^*) \geq \gamma$ for some $\gamma > 0$
- $\|\boldsymbol{x}_j\|_2 \leq R$ for some constant
- Initialize $\boldsymbol{\theta}^{(0)} = \mathbf{0}$

Comment.

- γ is the margin
- $\boldsymbol{\theta}^*$ is ONE solution such that the margin is at least γ



© Stanley Chan 2020. All Rights Reserved.

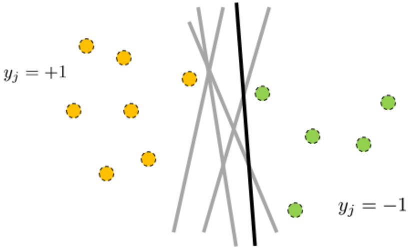
Interpreting the Perceptron Convergence

Theorem. Assume the following things:

- The two classes are linearly separable
- This means: $(\boldsymbol{\theta}^*)^T(y_j\boldsymbol{x}_j) = y_j((\boldsymbol{w}^*)^T\boldsymbol{x}_j + w_0^*) \geq \gamma$ for some $\gamma > 0$
- $\|\boldsymbol{x}_j\|_2 \leq R$ for some constant
- Initialize $\boldsymbol{\theta}^{(0)} = \mathbf{0}$

Comment.

- If you do not initialize at $\mathbf{0}$, still converge.
- The solution $\boldsymbol{\theta}^*$ might be different.



©Stanley Chan 2020. All Rights Reserved.

Interpreting the Perceptron Convergence

Then, batch mode perceptron algorithm converges to the true solution $\boldsymbol{\theta}^*$

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 = 0$$

when the number of iterations k exceeds

$$k \geq \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}.$$

Comment:

- You can turn batch mode to online mode by picking only one $j \in \mathcal{M}_k$
- You will do slower, but you can still converge
- $\boldsymbol{\theta}^*$ is the converging point of *this* particular sequence $\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots \boldsymbol{\theta}^\infty\}$
- Not an arbitrary separating hyperplane

©Stanley Chan 2020. All Rights Reserved.

Interpreting the Perceptron Convergence

Then, batch mode perceptron algorithm converges to the true solution $\boldsymbol{\theta}^*$

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 = 0,$$

when the number of iterations k exceeds

$$k \geq \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}.$$

Comment:

- R controls the radius of the class.
- Large R: Wide spread. Difficult. Need large k.
- γ controls the margin.
- Large γ: Big margin. Easy. Need small k.

©Stanley Chan 2020. All Rights Reserved.

Summary of the Convergence Theorem

- **Algorithm**: You use gradient descent on $J_{\text{soft}}(\boldsymbol{\theta})$
- **Solution**: You get a global minimizer for $J_{\text{hard}}(\boldsymbol{\theta})$
- But this is just one of the many global minimizers
- **Assumption**: Linearly separable
- If not linearly separable, then will oscillate
- **Margin**: At optimal solution there is a margin because separable
- Applications: Not quite; There are many better methods
- Theoretical usage: Good for analyzing linear models. Very simple algorithm.

©Stanley Chan 2020. All Rights Reserved.

Reading List

Perceptron Algorithm

- Abu-Mostafa, Learning from Data, Chapter 1.2
- Duda, Hart, Stork, Pattern Classification, Chapter 5.5
- Cornell CS 4780 Lecture https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote03.html
- UCSD ECE 271B Lecture http://www.svcl.ucsd.edu/courses/ece271B-F09/handouts/perceptron.pdf

©Stanley Chan 2020. All Rights Reserved.

Appendix

©Stanley Chan 2020. All Rights Reserved.

Proof Part 1

- Define

$$\overline{\boldsymbol{x}}^{(k)} = \sum_{j \in \mathcal{M}_k} y_j \boldsymbol{x}_j.$$

- Let $\boldsymbol{\theta}^*$ be the minimizer. Then,

$$\begin{aligned}
\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 &= \|\boldsymbol{\theta}^{(k)} + \alpha_k \overline{\boldsymbol{x}}^{(k)} - \boldsymbol{\theta}^*\|^2 \\
&= \|(\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*) + \alpha_k \overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 + 2\alpha_k (\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*)^T \overline{\boldsymbol{x}}^{(k)} + \alpha_k^2 \|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 + 2\alpha_k \left(\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\right)^T \left(\sum_{j \in \mathcal{M}_k} y_j \boldsymbol{x}_j\right) \\
&\qquad + \alpha_k^2 \left\| \sum_{j \in \mathcal{M}_k} y_j \boldsymbol{x}_j \right\|^2.
\end{aligned}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 2

- By construction, $\boldsymbol{\theta}^{(k)}$ updates only the misclassified samples (during the k-th iteration)
- So for any $j \in \mathcal{M}_k$ we must have $(\boldsymbol{\theta}^{(k)})^T(y_j \boldsymbol{x}_j) \leq 0$.
- This implies that

$$(\boldsymbol{\theta}^{(k)})^T \overline{\boldsymbol{x}}^{(k)} = \sum_{j \in \mathcal{M}_k} (\boldsymbol{\theta}^{(k)})^T y_j \boldsymbol{x}_j \leq 0.$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 2

- By construction, $\boldsymbol{\theta}^{(k)}$ updates only the misclassified samples (during the k-th iteration)
- So for any $j \in \mathcal{M}_k$ we must have $(\boldsymbol{\theta}^{(k)})^T(y_j\boldsymbol{x}_j) \leq 0$.
- This implies that

$$(\boldsymbol{\theta}^{(k)})^T\overline{\boldsymbol{x}}^{(k)} = \sum_{j\in\mathcal{M}_k}(\boldsymbol{\theta}^{(k)})^T y_j\boldsymbol{x}_j \leq 0.$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 2

- By construction, $\boldsymbol{\theta}^{(k)}$ updates only the misclassified samples (during the k-th iteration)
- So for any $j \in \mathcal{M}_k$ we must have $(\boldsymbol{\theta}^{(k)})^T(y_j\boldsymbol{x}_j) \leq 0$.
- This implies that

$$(\boldsymbol{\theta}^{(k)})^T\overline{\boldsymbol{x}}^{(k)} = \sum_{j\in\mathcal{M}_k} (\boldsymbol{\theta}^{(k)})^T y_j \boldsymbol{x}_j \leq 0.$$

© Stanley Chan 2020. All Rights Reserved.

Proof Part 2

- By construction, $\boldsymbol{\theta}^{(k)}$ updates only the misclassified samples (during the k-th iteration)
- So for any $j \in \mathcal{M}_k$ we must have $(\boldsymbol{\theta}^{(k)})^T(y_j\boldsymbol{x}_j) \leq 0$.
- This implies that

$$(\boldsymbol{\theta}^{(k)})^T\overline{\boldsymbol{x}}^{(k)} = \sum_{j \in \mathcal{M}_k} (\boldsymbol{\theta}^{(k)})^T y_j \boldsymbol{x}_j \leq 0.$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 2

- By construction, $\boldsymbol{\theta}^{(k)}$ updates only the misclassified samples (during the k-th iteration)
- So for any $j \in \mathcal{M}_k$ we must have $(\boldsymbol{\theta}^{(k)})^T(y_j\boldsymbol{x}_j) \leq 0$.
- This implies that

$$(\boldsymbol{\theta}^{(k)})^T\overline{\boldsymbol{x}}^{(k)} = \sum_{j\in\mathcal{M}_k} (\boldsymbol{\theta}^{(k)})^T y_j\boldsymbol{x}_j \leq 0.$$

- Therefore, we can show that

$$\begin{aligned}
&\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 \\
&\leq \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 + 2\alpha_k \left(\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\right)^T \overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 + \cancel{2\alpha_k \left(\boldsymbol{\theta}^{(k)}\right)^T \overline{\boldsymbol{x}}^{(k)}} - 2\alpha_k \left(\boldsymbol{\theta}^*\right)^T \overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&\leq \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 - 2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2.
\end{aligned}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 3

- So we have

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 \le \|\boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^*\|^2 \underbrace{-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2}.$$

- The sum of the last two terms is

$$-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 = \alpha_k\left(-2(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k\|\overline{\boldsymbol{x}}^{(k)}\|^2\right),$$

- Negative if and only if $\alpha_k < \frac{2(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}}{\|\overline{\boldsymbol{x}}^{(k)}\|^2}$
- Thus, we choose

$$\alpha_k = \frac{(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}}{\|\overline{\boldsymbol{x}}^{(k)}\|^2},$$

© Stanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 = -2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2$$

- By assumption $\|\boldsymbol{x}_j\|^2 \leq R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \geq \gamma$ for any j
- So

$$\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} = \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2}$$

$$\geq \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k} R^2}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 = -2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2$$

- By assumption $\|\boldsymbol{x}_j\|^2 \leq R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \geq \gamma$ for any j
- So

$$\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} = \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2}$$

$$\geq \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k} R^2}$$

ⓒStanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$
\begin{aligned}
-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 &= -2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= -\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2}.
\end{aligned}
$$

- By assumption $\|\boldsymbol{x}_j\|^2 \leq R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \geq \gamma$ for any j
- So

$$
\begin{aligned}
\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} &= \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2} \\
&\geq \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k}R^2}
\end{aligned}
$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$
\begin{aligned}
-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 &= -2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= -\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2}.
\end{aligned}
$$

- By assumption $\|\boldsymbol{x}_j\|^2 \le R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \ge \gamma$ for any j
- So

$$
\begin{aligned}
\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} &= \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2} \\
&\ge \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k} R^2}
\end{aligned}
$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$
\begin{aligned}
-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 &= -2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= -\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2}.
\end{aligned}
$$

- By assumption $\|\boldsymbol{x}_j\|^2 \leq R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \geq \gamma$ for any j
- So

$$
\begin{aligned}
\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} &= \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2} \\
&\geq \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k} R^2}
\end{aligned}
$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$
\begin{aligned}
-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 &= -2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= -\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2}.
\end{aligned}
$$

- By assumption $\|\boldsymbol{x}_j\|^2 \le R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \ge \gamma$ for any j
- So

$$
\begin{aligned}
\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} &= \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2} \\
&\ge \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k} R^2}
\end{aligned}
$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$
\begin{aligned}
-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 &= -2{\color{red}\alpha_k}(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + {\color{red}\alpha_k^2}\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= -\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2}.
\end{aligned}
$$

- By assumption $\|\boldsymbol{x}_j\|^2 \le R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \ge \gamma$ for any j
- So

$$
\begin{aligned}
\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} &= \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2} \\
&\ge \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k} R^2}
\end{aligned}
$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 4

- Then, we can have

$$
\begin{aligned}
-2\alpha_k(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + \alpha_k^2\|\overline{\boldsymbol{x}}^{(k)}\|^2 &= -2{\color{red}\alpha_k}(\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)} + {\color{red}\alpha_k^2}\|\overline{\boldsymbol{x}}^{(k)}\|^2 \\
&= -\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2}.
\end{aligned}
$$

- By assumption $\|\boldsymbol{x}_j\|^2 \leq R$ for any j, and $y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j \geq \gamma$ for any j
- So

$$
\begin{aligned}
\frac{\left((\boldsymbol{\theta}^*)^T\overline{\boldsymbol{x}}^{(k)}\right)^2}{\|\overline{\boldsymbol{x}}^{(k)}\|^2} &= \frac{\left(\sum_{j\in\mathcal{M}_k} y_j(\boldsymbol{\theta}^*)^T\boldsymbol{x}_j\right)^2}{\sum_{j\in\mathcal{M}_k}\|\boldsymbol{x}_j\|^2} \\
&\geq \frac{\left(\sum_{j\in\mathcal{M}_k}\gamma\right)^2}{\sum_{j\in\mathcal{M}_k} R^2} = |\mathcal{M}_k|\frac{\gamma^2}{R^2}
\end{aligned}
$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}$$

© Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}$$

© Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2}$$

©Stanley Chan 2020. All Rights Reserved.

Proof Part 5

- Then by induction we can show that

$$\|\boldsymbol{\theta}^{(k+1)} - \boldsymbol{\theta}^*\|^2 < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 - \sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2}.$$

- We can conclude that

$$\sum_{i=1}^{k} |\mathcal{M}_i| \frac{\gamma^2}{R^2} < \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}^*\|^2,$$

- Therefore,

$$\underbrace{\sum_{i=1}^{k} |\mathcal{M}_i|}_{k \leq (\cdot)} < \frac{\|\boldsymbol{\theta}^*\|^2 R^2}{\gamma^2} = \frac{\max_j \|\boldsymbol{\theta}^*\|^2 \|\boldsymbol{x}_j\|^2}{(\min_j (\boldsymbol{\theta}^*)^T \boldsymbol{x}_j)^2}$$

©Stanley Chan 2020. All Rights Reserved.