Administrivia, Introduction to Online Learning

3/29/2016

CS 159: Advanced Topics in Machine Learning

Class Details

- **Instructor:** Yisong Yue

- **TAs:**

Hoang Le

Stephan Zheng

- **Course Website:** http://www.yisongyue.com/courses/cs159/

Style of Course

- Graduate level course
- Give students an overview of topics
- Dig deep into one topic for final project
- Assume students are mathematically mature
  - Goal is to understand basic concepts
  - Understand specific mathematical details depending on your interest

Grading Breakdown

- Participation (20%)

- Mini-quizzes (10%)

- Final Project (70%)

Paper Reading & Discussion

- Paper Reading Course
  - Reading assignments for each lecture
  - Lectures more like discussion

- Student Presentations
  - Presentation schedule signup soon
  - Present in groups
  - Can choose which paper(s) to present

Mini-quizzes

- Evening after every lecture
  - Very short
  - Easy if you read material & attended lecture

- **Released via Piazza**
  - Also use Piazza for Q&A

Final Project

- Can be on any topic related to the course
- Work in groups
- Will release timeline of progress reports soon
- Peer review (?)

Topics

- Online Learning
- Multi-armed Bandits
- Active Learning
- Crowdsourcing
- Reinforcement Learning
- Models of Human Decision making

Focus of Course

- Rigorous algorithm design
  - Math intensive, but nothing too hard
  - Will walk through relevant math in class
- Apply to interesting applications
  - What are the right ways to model a problem?

What Does Rigorous Mean?

- Formal model
  - Explicitly state your assumptions
- Rigorously reason about how your algorithm solves the model
  - Sometimes with provable guarantees
- Argue that your model is a reasonable one

What Makes a Good Final Project?

- Pure Theory
  - Study proof techniques, try to extend proof, or apply to new setting
- Algorithms
  - Extend algorithms, design new ones, for new settings
- Modeling
  - Model new setting, what are the right assumptions?

Outline

- First 3-5 lectures
  - Review basic algorithms
  - Somewhat dry, but necessary

- Topics/readings chosen by students
  - With curating from Instructor & Tas
  - List of papers already on website
    - But is negotiable

Rest of Today

- Introduction to Online Learning
  - Follow the Leader
  - Perceptron

- Brief Overview of Other Topics in Course

Introduction to Online Learning

(Most Basic) Online Learning

- For t = 1….T (sometimes T is unknown)
  - Algorithm chooses p_t
  - World reveals loss function L_t
  - Algorithm suffers loss $L_t(p_t)$

What are the semantics of p_t?

What is the loss?

How is the loss chosen?

- **Goal:** minimize total loss $\sum_{t=1}^{T} L_t(p_t)$

Recall: Supervised Learning

$$\operatorname*{argmin}_{w} \sum_{i=1}^{N} L\big(y_i, f(x_i \mid w)\big) \qquad\qquad S = \{(x_i, y_i)\}_{i=1}^{N}$$

- Optimize via Stochastic Gradient Descent
  - Maintain a w_t
  - Each iteration receive: $L_t(w_t) = L\big(y_i, f(x_i \mid w_t)\big)$
  - Assume sampled randomly from S
  - Choose w_{t+1} based on w_t and L_t

(Most Basic) Online Learning

- For $t = 1....T$ (sometimes T is unknown)
  - Algorithm chooses p_t
  - World reveals loss function L_t
  - Algorithm suffers loss $L_t(p_t)$

$p_t = w_t$

$L_t(w_t) = L(y_t, f(x_t | w_t))$

L_t chosen randomly

- **Goal:** minimize total loss $\sum_{t=1}^{T} L_t(p_t)$

What if…

- We receive a constant stream of data?
  - Don't know T a priori

- We receive data in some arbitrary way?
  - Not sampled independently from some distribution

- **Can we still (provably) achieve good performance?**

Quantifying Performance

- In supervised learning we care about:

$$\sum_{i=1}^{N} L\left(y_i, f(x_i \mid w)\right) = \sum_{i=1}^{N} L_i(w)$$

a single w

- In online learning, we care about:

$$\sum_{t=1}^{T} L\left(y_t, f(x_t \mid w_t)\right) = \sum_{t=1}^{T} L_t(w_t)$$

a sequence of w_t

Quantifying Performance

- Compete against single best w in hindsight:

$$\mathrm{R}(T) = \sum_{t=1}^{T} L_t(w_t) - \sum_{t=1}^{T} L_t(w^*)$$

"Regret"

$$\sum_{t=1}^{T} L_t(w^*) = \min_{w} \sum_{t=1}^{T} L_t(w)$$

Interpretation: best possible loss w.r.t. supervised learning

Interpreting Regret

- Expected Training Error is: $\frac{1}{T}\sum_{t=1}^{T} L_t(w_t)$

- Want expected training error to (quickly) converge to optimal
  - **Equivalent to average regret (quickly) converging to 0:**

$$\frac{1}{T}\mathrm{R}(T) = \frac{1}{T}\left(\sum_{t=1}^{T} L_t(w_t) - \sum_{t=1}^{T} L_t(w^*)\right) \to 0$$

- **Satisfied when regret grows sublinearly w.r.t. T!**

Summary of Regret

- Generic way to quantify performance
  - Characterizes speed of convergence for SGD
- Applies to many online learning settings
- We'll see other ways to quantify performance later in course

Follow the Leader

Basic Online Convex Optimization

- For $t = 1 \ldots T$ (T unknown)
  - Algorithm chooses p_t in R^D
  - World reveals loss function $L_t(p_t) = |y_t - p_t|^2$
  - Algorithm suffers loss $L_t(p_t)$

Squared Distance to y_t
In general, convex loss

- **Goal:** minimize total loss $\sum_{t=1}^{T} L_t(p_t)$

Follow the Leader Algorithm

- The "leader" is the best point given what we know so far:

$$p_t = \underset{p}{\operatorname{argmin}} \sum_{t'=1}^{t-1} L_{t'}(p) = \underset{p}{\operatorname{argmin}} \sum_{t'=1}^{t-1} \left\| y_{t'} - p \right\|^2 = \frac{1}{t-1} \sum_{t'=1}^{t-1} y_{t'}$$

This is the entire algorithm!

Benefits and Drawbacks

- **Benefits:**
  - Efficient regret bounds (will see next slide)
  - Conceptually very simple
    - Can be applied to many settings

- **Drawbacks:**
  - Can be computationally very expensive
    - For arbitrary loss functions
      - (can't use average all the time)

Definitions

- Best hindsight choice of first t time steps:

$$p_t^* = \operatorname*{argmin}_p \sum_{t'=1}^{t} L_{t'}(p) = \operatorname*{argmin}_p \sum_{t'=1}^{t} \left\| y_{t'} - p \right\|^2 = \frac{1}{t} \sum_{t'=1}^{t} y_{t'}$$

- Follow the Leader plays: $p_t = p_{t-1}^*$

$$p_t = \operatorname*{argmin}_p \sum_{t'=1}^{t-1} L_{t'}(p) = \operatorname*{argmin}_p \sum_{t'=1}^{t-1} \left\| y_{t'} - p \right\|^2 = \frac{1}{t-1} \sum_{t'=1}^{t-1} y_{t'}$$

Goal

- Minimize Regret:

$$\mathrm{R}(T) = \sum_{t=1}^{T} L_t(p_t) - \sum_{t=1}^{T} L_t(p_T^*)$$

$$p_T^* = \underset{p}{\operatorname{argmin}} \sum_{t=1}^{T} L_t\left(p\right) = \underset{p}{\operatorname{argmin}} \sum_{t=1}^{T} \left\| y_t - p \right\|^2 = \frac{1}{T} \sum_{t=1}^{T} y_t$$

Lemma 1

$$\sum_{t=1}^{T} L_t(p_t^*) \leq \sum_{t=1}^{T} L_t(p_T^*)$$

- **Interpretation:**
  – the moving best hindsight is at least as good as the final best hindsight

- Proof by Induction
  – Base case (T=1): $L_1(p_1^*) = L_1(p_1^*)$

Proof Continued

- Inductive Case (T>1):
  - Remove last term because it’s equivalent

$$\sum_{t=1}^{T} L_t(p_t^*) \le \sum_{t=1}^{T} L_t(p_T^*) \Rightarrow \sum_{t=1}^{T-1} L_t(p_t^*) \le \sum_{t=1}^{T-1} L_t(p_T^*)$$

  - Observe:

$$\underbrace{\sum_{t=1}^{T-1} L_t(p_t^*) \le \sum_{t=1}^{T-1} L_t(p_{T-1}^*)}_{\text{Inductive Hypothesis}} \overbrace{\le \sum_{t=1}^{T-1} L_t(p_T^*)}^{\text{Definition of p*}}$$

Regret Bound

$$\mathrm{R}(T) = \sum_{t=1}^{T} L_t(p_t) - \sum_{t=1}^{T} L_t(p_T^*)$$

$$= \sum_{t=1}^{T} L_t(p_{t-1}^*) - \sum_{t=1}^{T} L_t(p_T^*)$$

Definition of Follow the Leader

$$\leq \sum_{t=1}^{T} L_t(p_{t-1}^*) - \sum_{t=1}^{T} L_t(p_t^*)$$

Lemma 1

Regret Bound (continued)

$$\sum_{t=1}^{T} L_t(p_{t-1}^*) - \sum_{t=1}^{T} L_t(p_t^*) = \sum_{t=1}^{T} \left\| p_{t-1}^* - y_t \right\|^2 - \sum_{t=1}^{T} \left\| p_t^* - y_t \right\|^2$$

$$= \sum_{t=1}^{T} \left\langle p_{t-1}^* - p_t^*, p_{t-1}^* + p_t^* - 2y_t \right\rangle$$

Cauchy-Schwarz

$$\le \sum_{t=1}^{T} \left\| p_{t-1}^* - p_t^* \right\| \cdot \left\| p_{t-1}^* + p_t^* - 2y_t \right\|$$

Triangle Inequality

$$\le \sum_{t=1}^{T} \left\| p_{t-1}^* - p_t^* \right\| \cdot \left(\left\| p_{t-1}^* \right\| + \left\| p_t^* \right\| + \left\| 2y_t \right\| \right)$$

Regret Bound (continued)

Assume each y_t has norm bounded by B:

$$\sum_{t=1}^{T}\left\|p_{t-1}^{*}-p_{t}^{*}\right\|\cdot\left(\left\|p_{t-1}^{*}\right\|+\left\|p_{t}^{*}\right\|+\left\|2y_{t}\right\|\right)\leq 4B\sum_{t=1}^{T}\left\|p_{t-1}^{*}-p_{t}^{*}\right\|$$

Note that each p* also has norm bounded by B

Regret Bound (continued)

Use the fact that:

$$p_t^* = \frac{(t-1)p_{t-1}^* + y_t}{t}$$

$$\left\| p_{t-1}^* - p_t^* \right\| = \left\| p_{t-1}^* - \frac{(t-1)p_{t-1}^* + y_t}{t} \right\|$$

$$= \frac{1}{t}\left\| p_{t-1}^* - y_t \right\|$$

Triangle Inequality

$$\leq \frac{1}{t}\left(\left\| p_{t-1}^* \right\| + \left\| y_t \right\| \right)$$

Each has norm B

$$\leq \frac{2B}{t}$$

Regret Bound (complete)

$$
\begin{aligned}
\mathrm{R}(T) &= \sum_{t=1}^{T} L_t(p_t) - \sum_{t=1}^{T} L_t(p_T^*) \\
&\leq \sum_{t=1}^{T} L_t(p_{t-1}^*) - \sum_{t=1}^{T} L_t(p_t^*) \\
&\leq 4B \sum_{t=1}^{T} \left\| p_{t-1}^* - p_t^* \right\| \\
&\leq 8B^2 \sum_{t=1}^{T} \frac{1}{t} = O\left(B^2 \ln T\right)
\end{aligned}
$$

Logarithmic Regret!

Independent of how each y_t is chosen!

Recall: Interpreting Regret

- Expected Training Error is: $\frac{1}{T}\sum_{t=1}^{T} L_t(w_t)$

- Want expected training error to (quickly) converge to optimal
  - **Equivalent to average regret (quickly) converging to 0:**

$$\frac{1}{T}\mathrm{R}(T) = \frac{1}{T}\left(\sum_{t=1}^{T} L_t(w_t) - \sum_{t=1}^{T} L_t(w^*)\right) \longrightarrow 0$$

- **Satisfied when regret grows sublinearly w.r.t. T!**

When Should You Use FTL in Practice?

- When solving each optimization problem is not the bottleneck
  - For simple squared distance, it is trivial
  - For more complex loss functions, might require expensive optimization

- We will see an analysis of SGD-style algorithms next Tuesday
  - Make small updates to p_t using only L_t

Perceptron

Binary Classification Online Learning

- For t = 1....T (sometimes T is unknown)
  – Algorithm chooses w_t in R^D
  – World reveals loss function:

$$L_t(w_t) = 1_{\left[y_t \neq sign\left(\langle w_t, x_t \rangle\right)\right]}$$

0/1 Loss

  – Algorithm suffers loss $L_t(w_t)$

- **Goal:** minimize total loss $\sum_{t=1}^{T} L_t(p_t)$

Perceptron Learning Algorithm

If $L_t(w_t)$=1 : $$w_{t+1} = w_t + y_t x_t$$

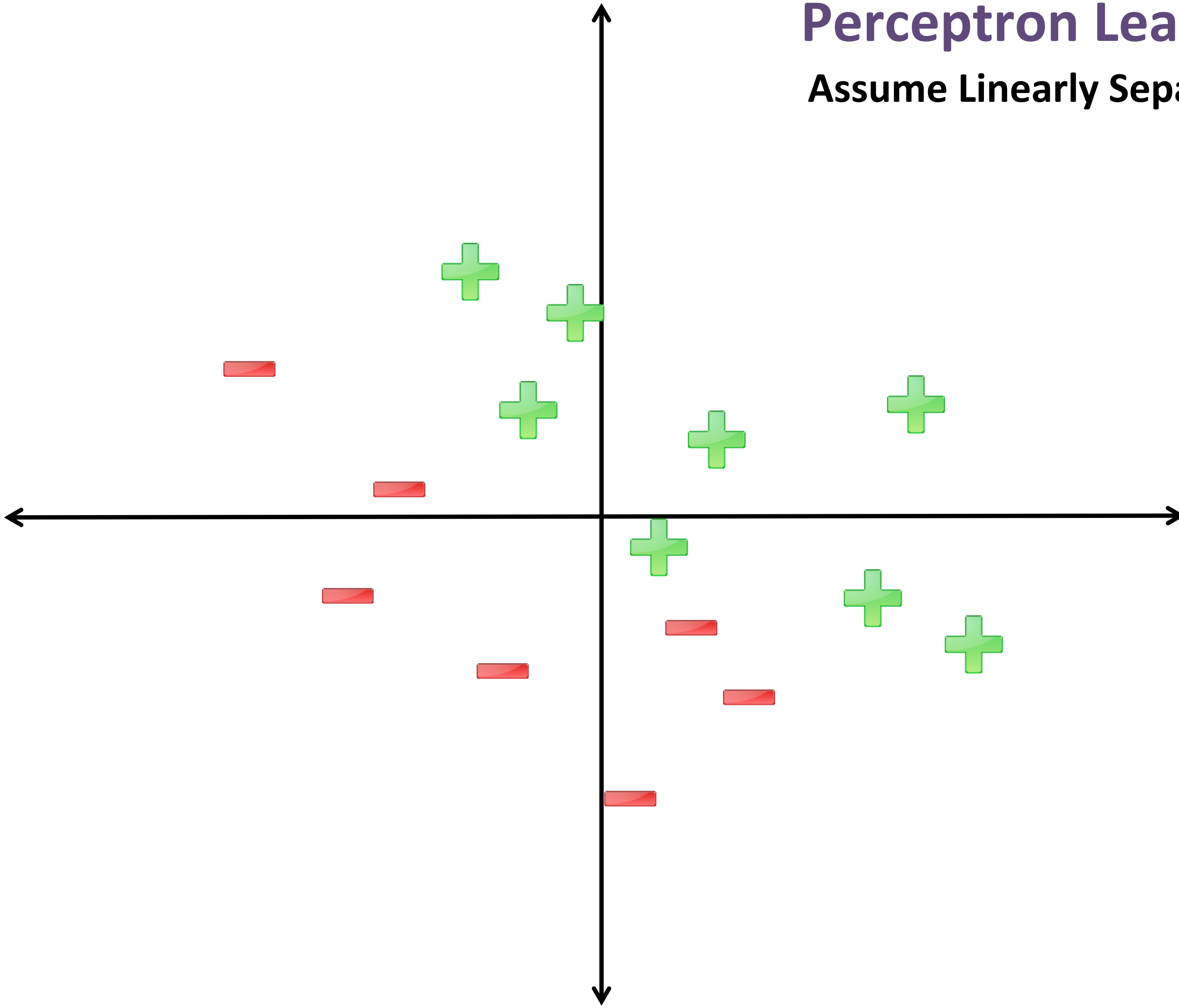
Else : $$w_{t+1} = w_t$$

$y \in \{-1, +1\}$

$x \in R^D$

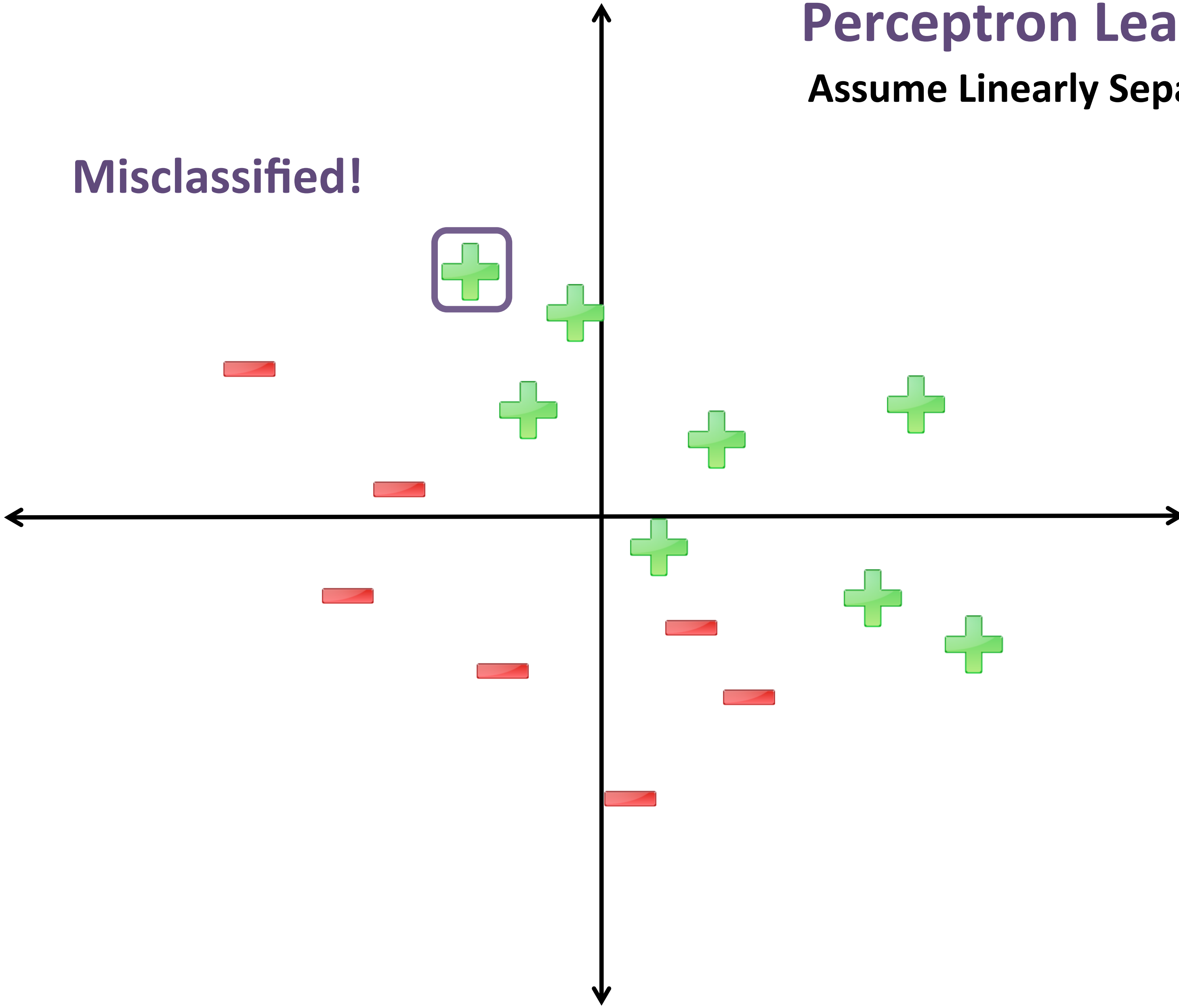
Perceptron Learning

Assume Linearly Separable



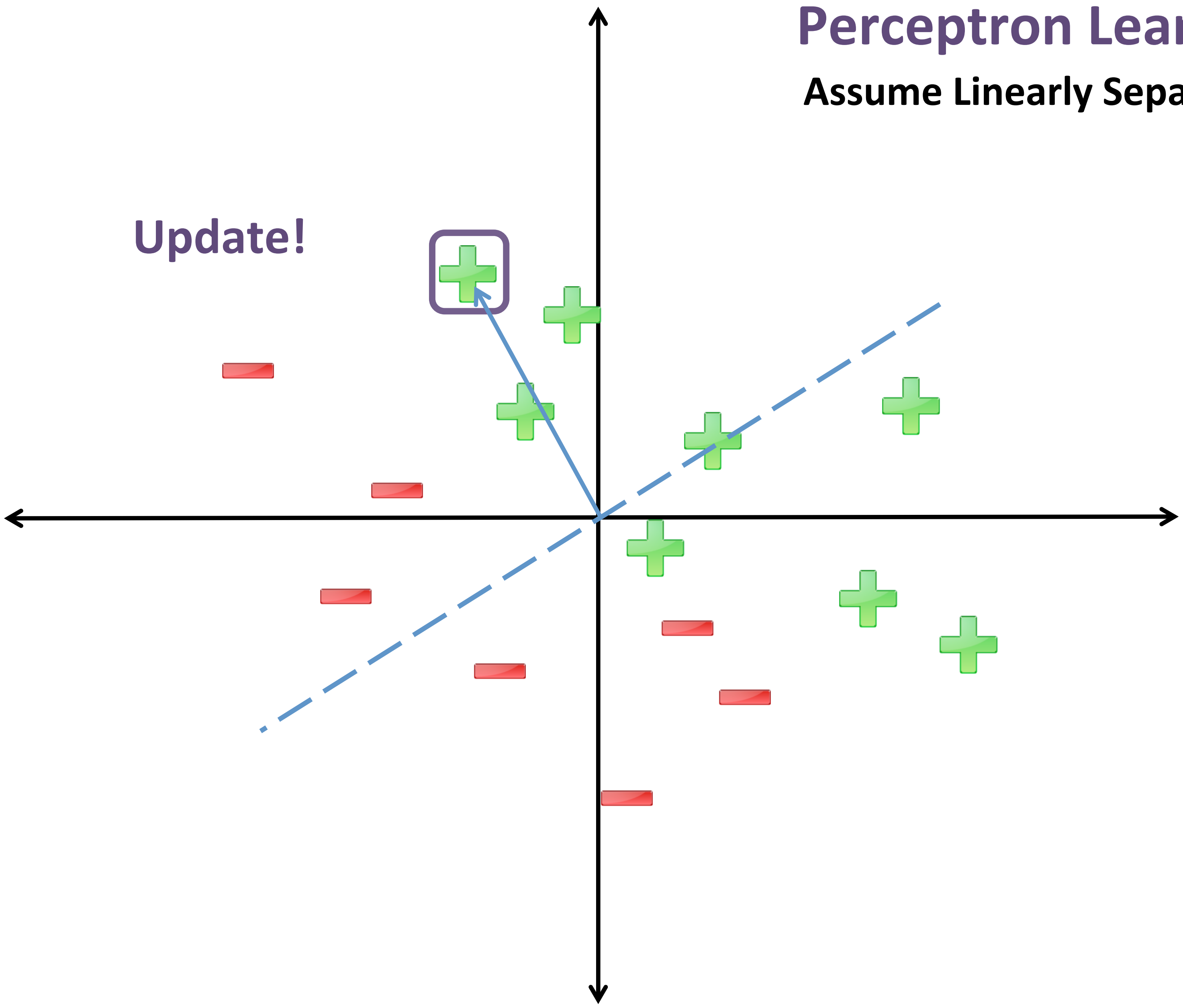
Perceptron Learning

Assume Linearly Separable



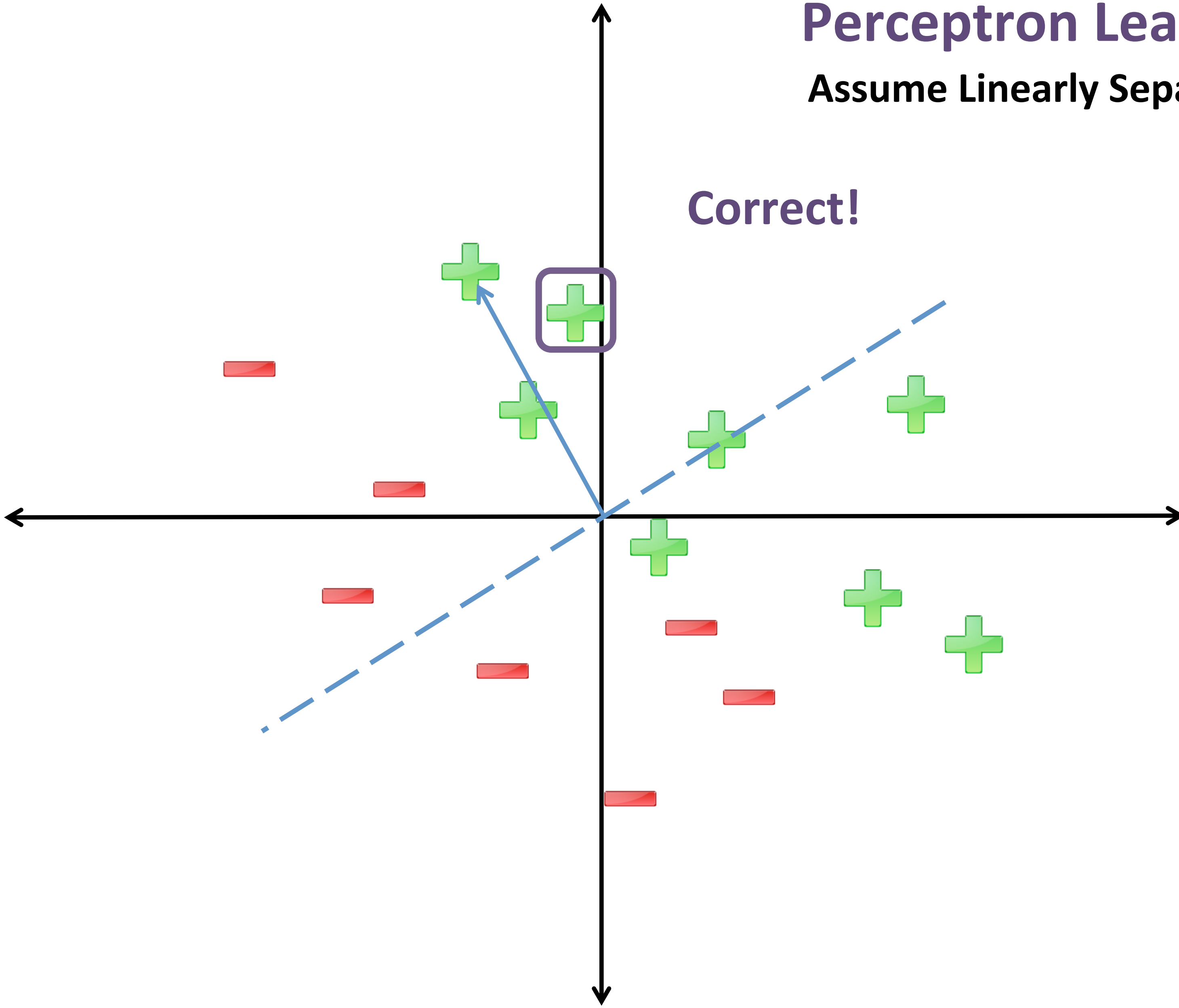
Perceptron Learning

Assume Linearly Separable



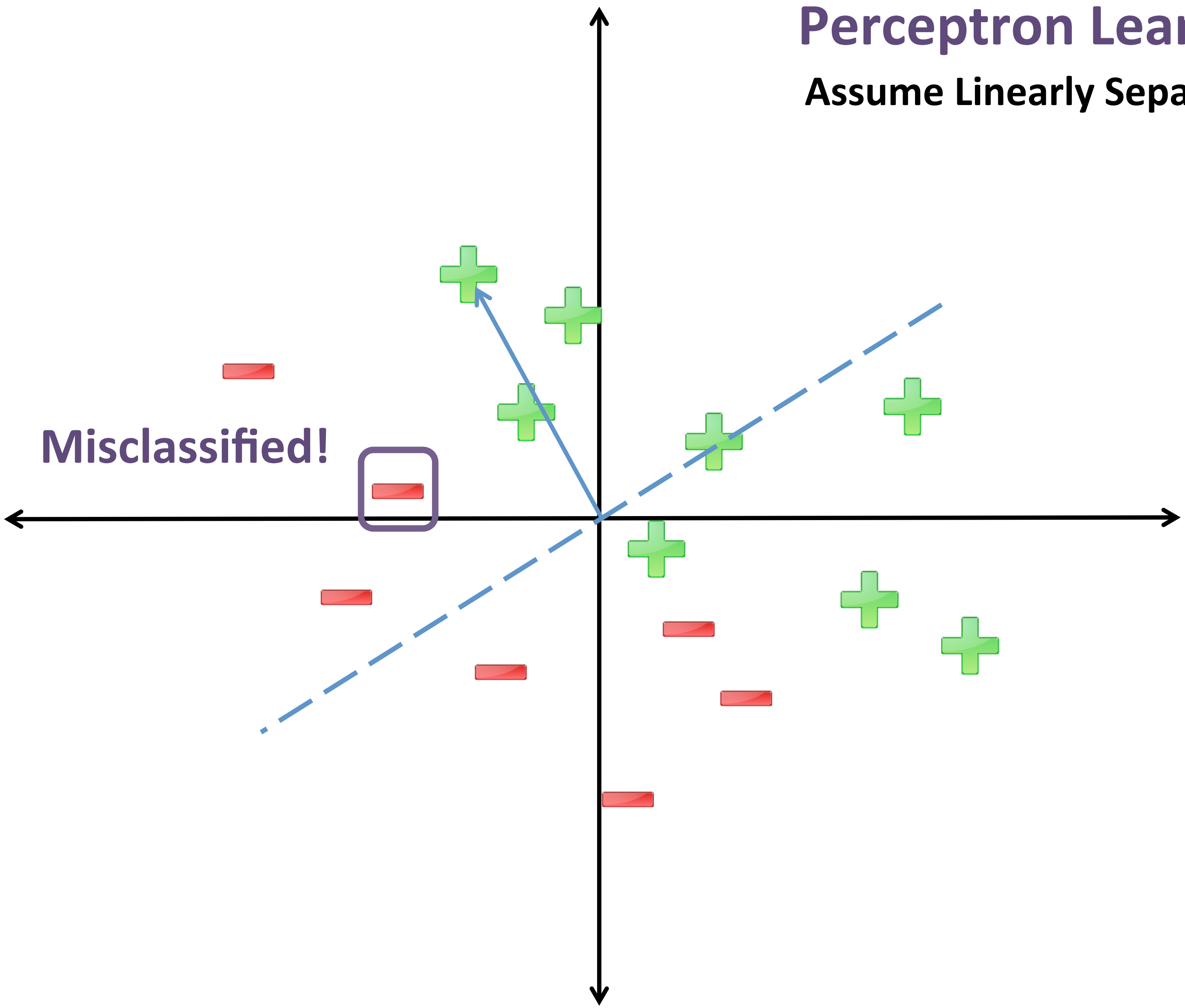
Perceptron Learning

Assume Linearly Separable



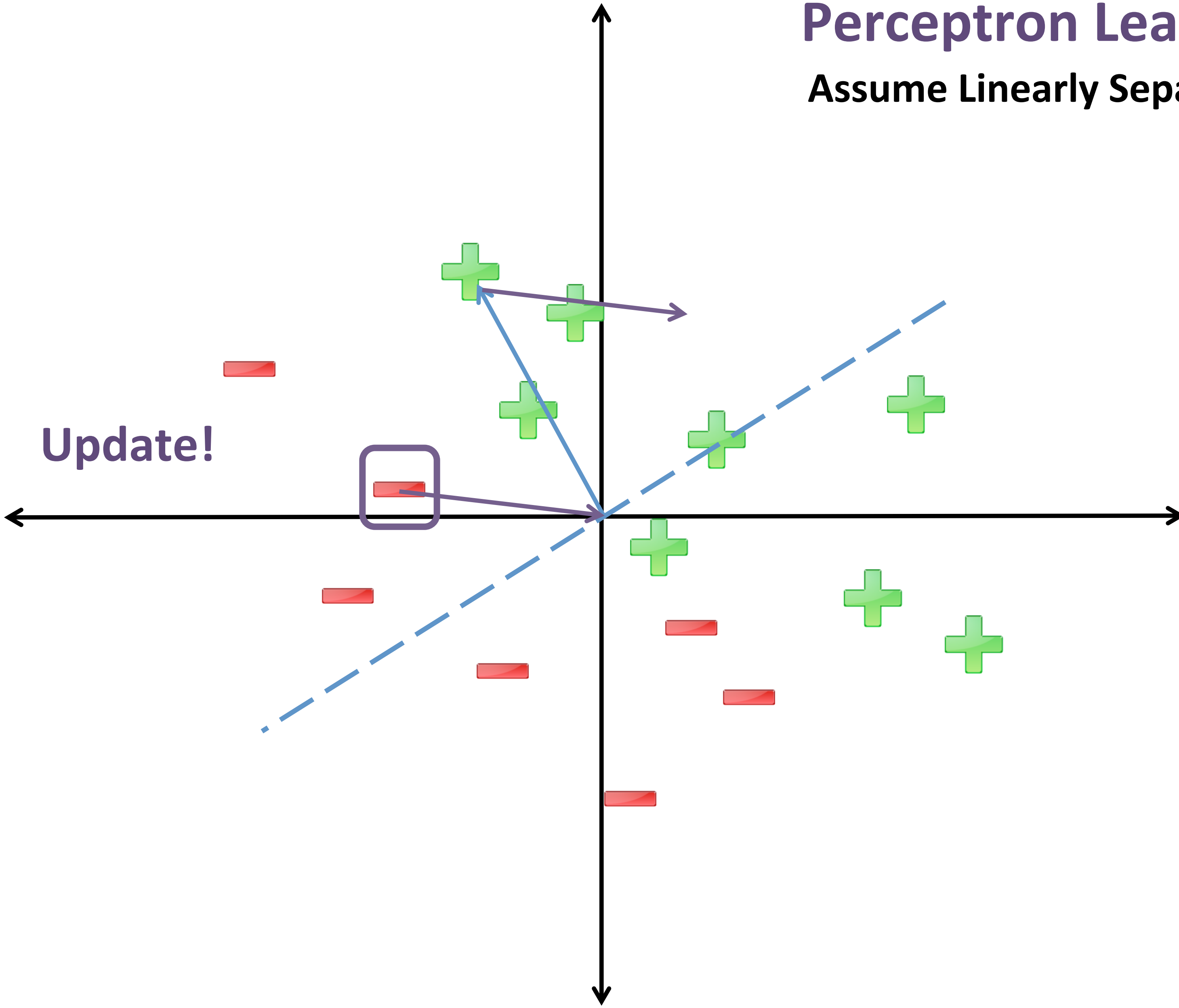
Perceptron Learning

Assume Linearly Separable



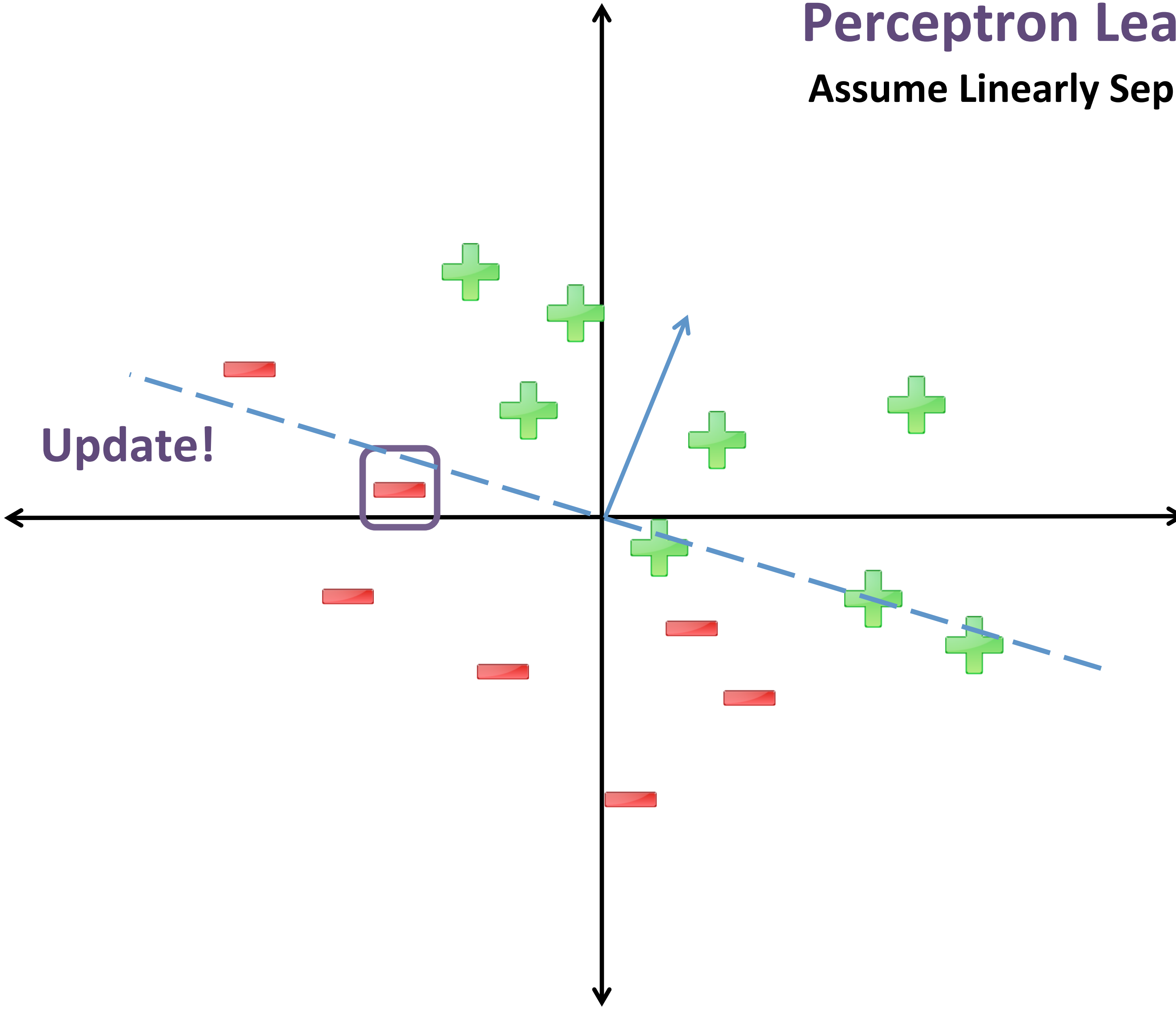
Perceptron Learning

Assume Linearly Separable



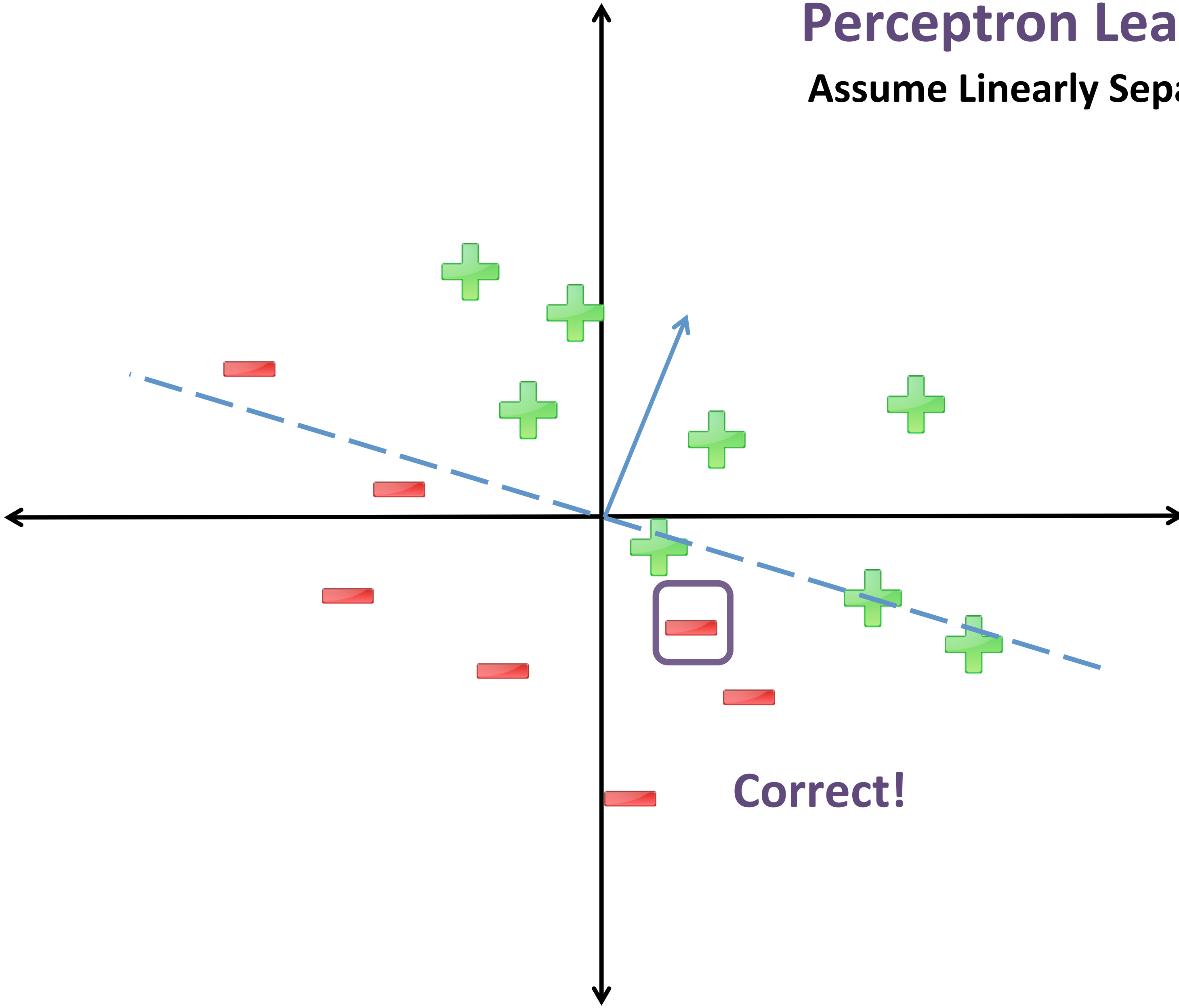
Perceptron Learning

Assume Linearly Separable



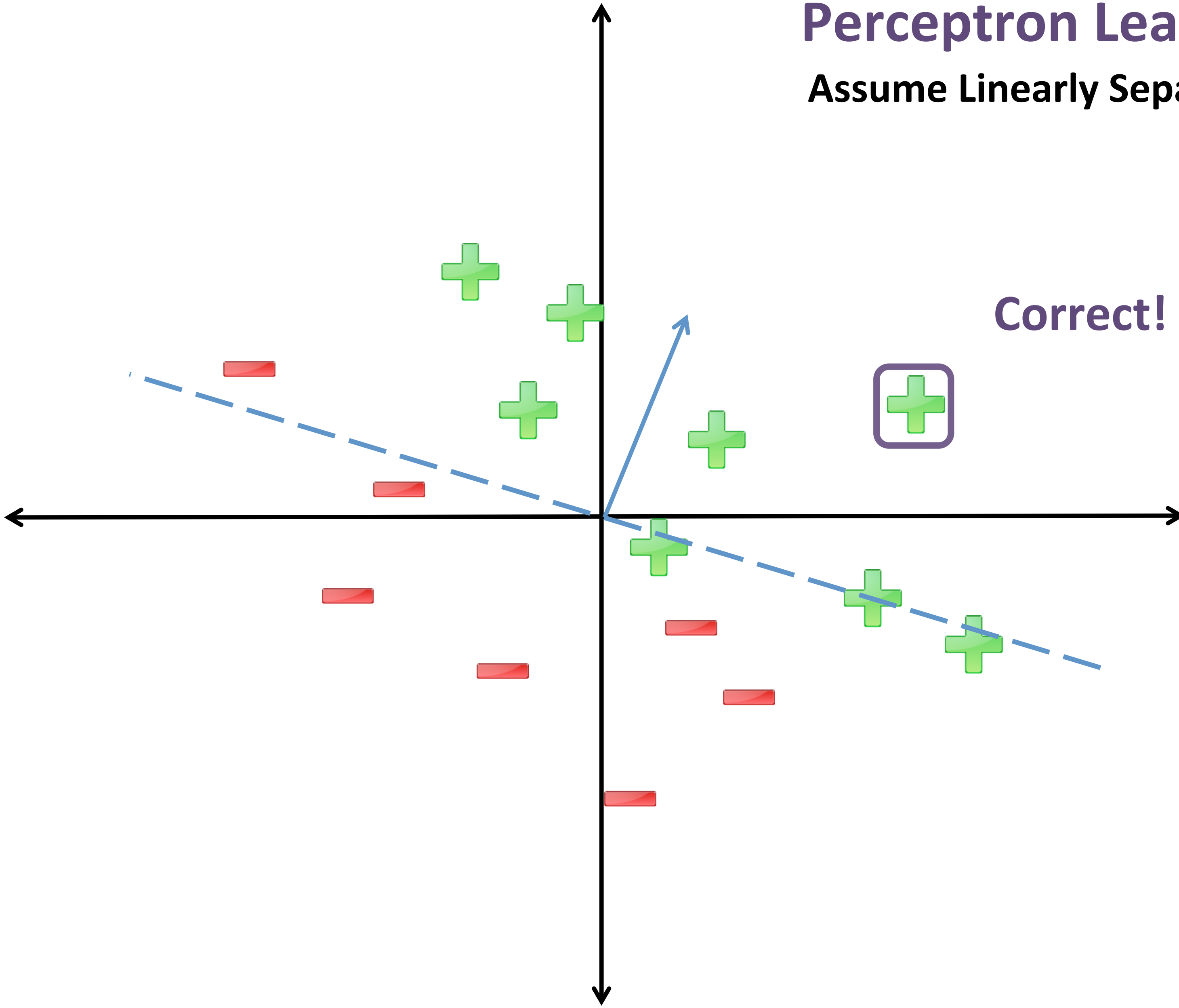
Perceptron Learning

Assume Linearly Separable



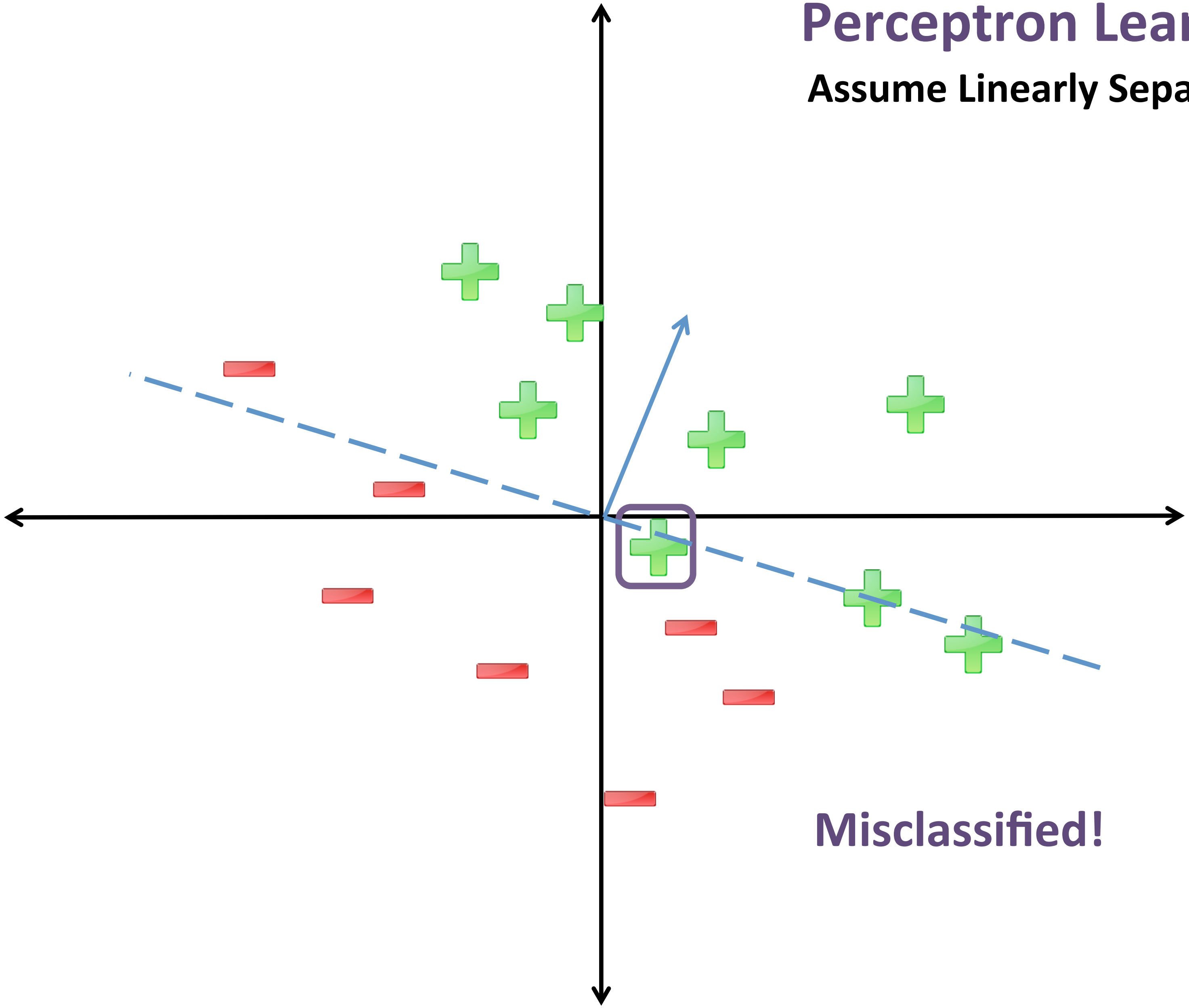
Perceptron Learning

Assume Linearly Separable



Perceptron Learning

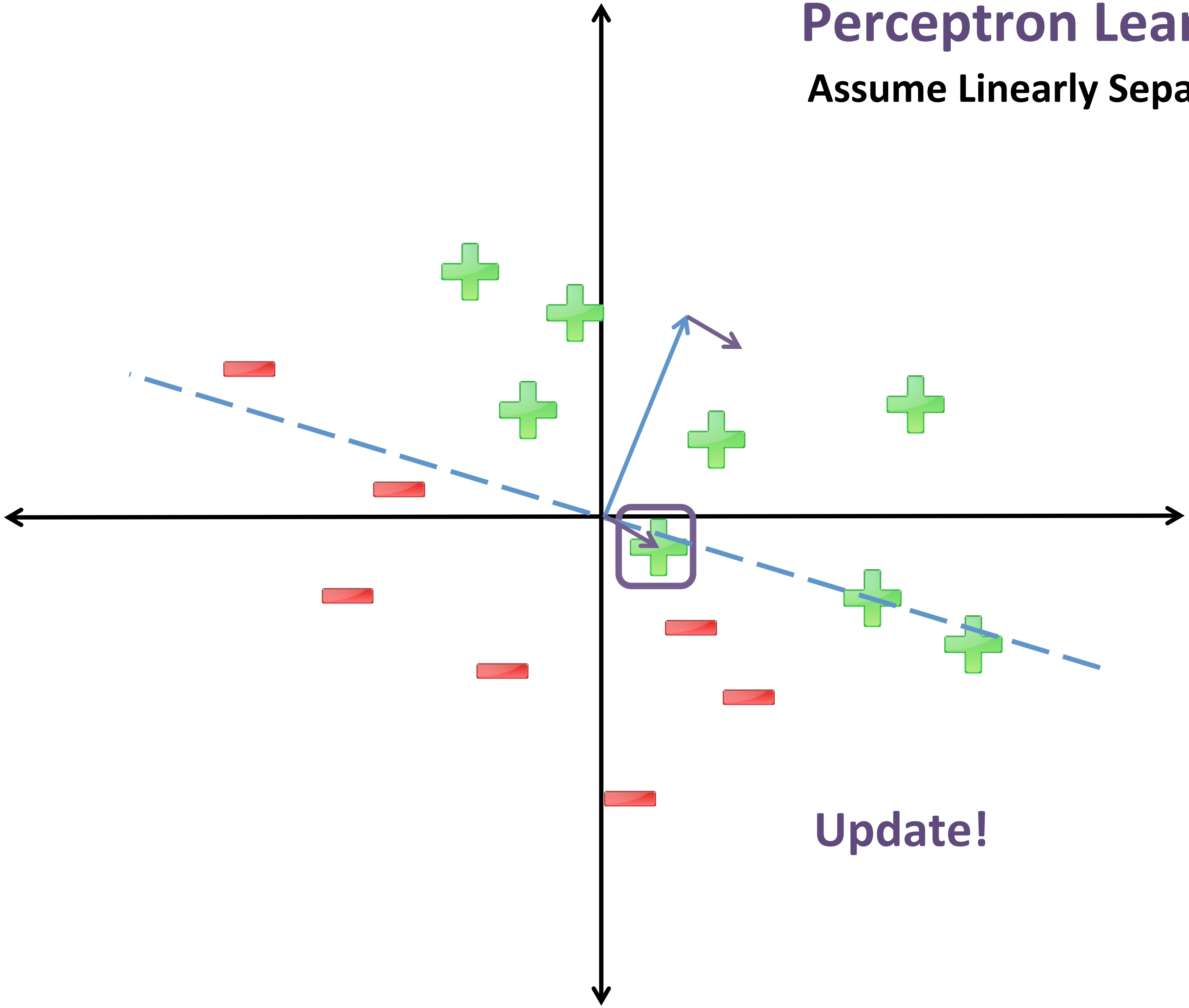
Assume Linearly Separable



Misclassified!

Perceptron Learning

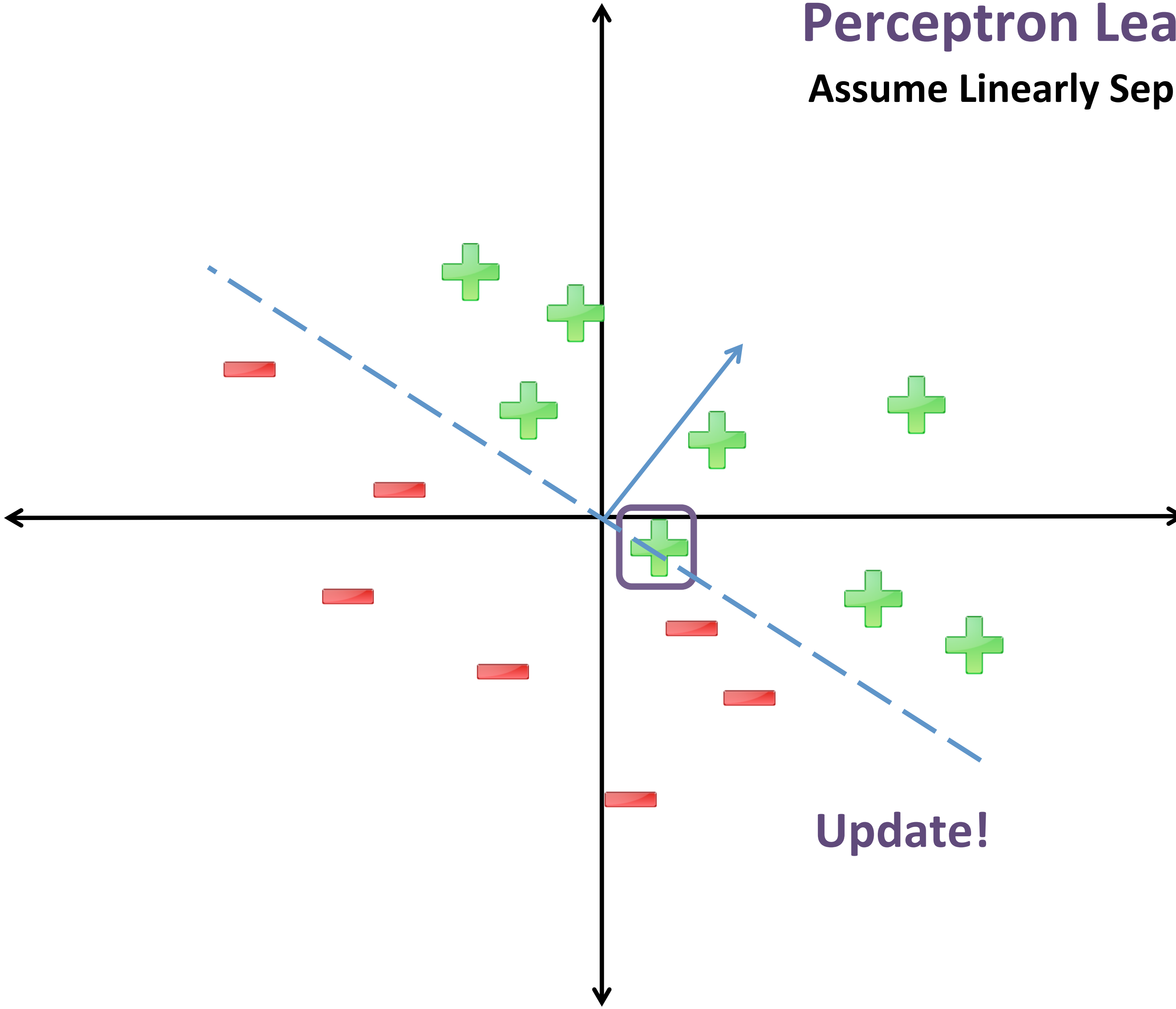
Assume Linearly Separable



Update!

Perceptron Learning

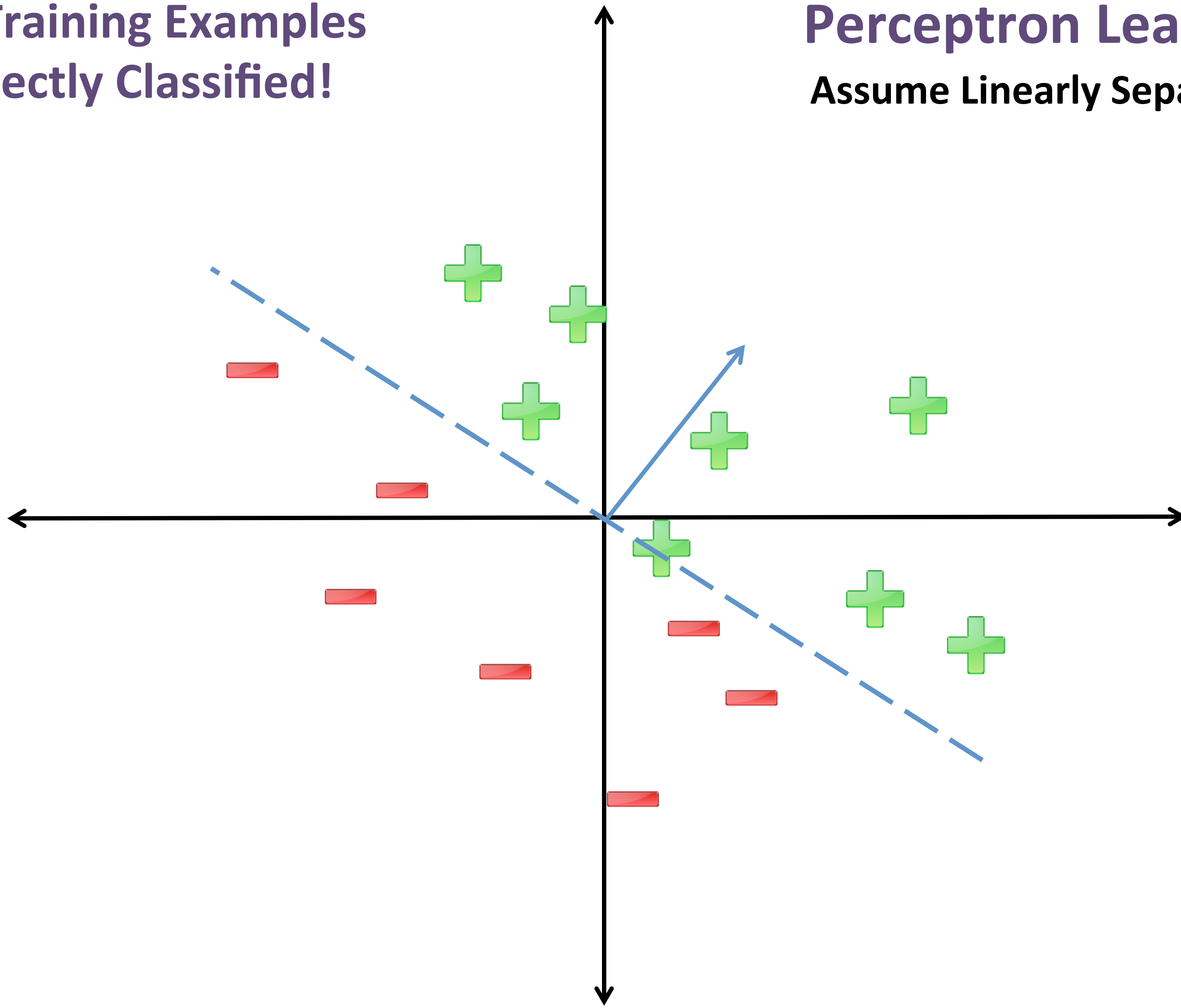
Assume Linearly Separable



All Training Examples Correctly Classified!

Perceptron Learning

Assume Linearly Separable



Regret Bound = Mistake Bound
(for Separable Case)

$$R(T) = \sum_{t=1}^{T} L_t(w_t) - \sum_{t=1}^{T} L_t(w^*)$$

- For separable case:

$$\sum_{t=1}^{T} L_t(w^*) = 0$$

- Regret = #Mistakes Perceptron makes

Lemma 2

$$\left\|\sum_{t\in I} y_t x_t\right\| \le \sqrt{\sum_{t\in I} \|x_t\|^2}$$

Mistake Iterations

Proof:

$$\left\|\sum_{t\in I} y_t x_t\right\| = \left\|\sum_{t\in I} (w_{t+1} - w_t)\right\| = \|w_{T+1}\|$$

Telescoping Sum

$$= \sqrt{\sum_{t\in I}\left(\|w_{t+1}\|^2 - \|w_t\|^2\right)}$$

Update Definition

$$= \sqrt{\sum_{t\in I}\left(\|w_t + y_t x_t\|^2 - \|w_t\|^2\right)}$$

$$= \sqrt{\sum_{t\in I}\left(\underbrace{2y_t\langle w_t, x_t\rangle}_{\le 0} + \|x_t\|^2\right)}$$

$$\le \sqrt{\sum_{t\in I}\|x_t\|^2}$$

Perceptron Mistake Bound

Holds for any ordering of training examples!

"Radius" of Feature Space

$$B = \max_x \|x\|$$

#Mistakes Bounded By: $\dfrac{B^2}{\gamma^2}$

Margin

**If Linearly Separable

Proof

- Margin: $$\gamma = \max_{w} \min_{(x_t, y_t)} \left\{ \frac{y_t \langle w, x_t \rangle}{\|w\|} \right\}$$

Must be positive due to linear separability

$$|I|\gamma \leq \frac{\left\langle w, \sum_{t \in I} y_t x_t \right\rangle}{\|w\|} \leq \left\| \sum_{t \in I} y_t x_t \right\| \leq \sqrt{\sum_{t \in I} \|x_t\|^2} \leq \sqrt{|I| B^2}$$

$$|I|\gamma \leq \sqrt{|I| B^2} \Rightarrow |I| \leq \frac{B^2}{\gamma^2}$$

Interpretation

- If the data is linearly separable
- Then ANY ordering of (x,y) will cause perceptron to converge with finite mistakes
- No dependence on IID sampling from true distribution

Brief Overview of Other Topics

Contextual Online Learning
(aka Online Learning with Experts)

- Given: Set of experts $\{f_k\}$
- For t = 1....T (sometimes T is unknown)
  - Each expert predicts $f_{k,t}$
  - Algorithm chooses p_t
  - World reveals loss function L_t
  - Algorithm suffers loss $L_t(p_t)$

Generalizes Boosting

- **Goal:** minimize total loss $\sum_{t=1}^{T} L_t(p_t)$

Partial Information Online Learning

- For $t = 1....T$ (sometimes T is unknown)
  - Algorithm chooses p_t
  - World reveals loss $L_t(p_t)$
  - Algorithm suffers loss $L_t(p_t)$

We don't know loss of other choices

Need to "explore" to measure loss of alternatives

- **Goal:** minimize total loss $\sum_{t=1}^{T} L_t(p_t)$

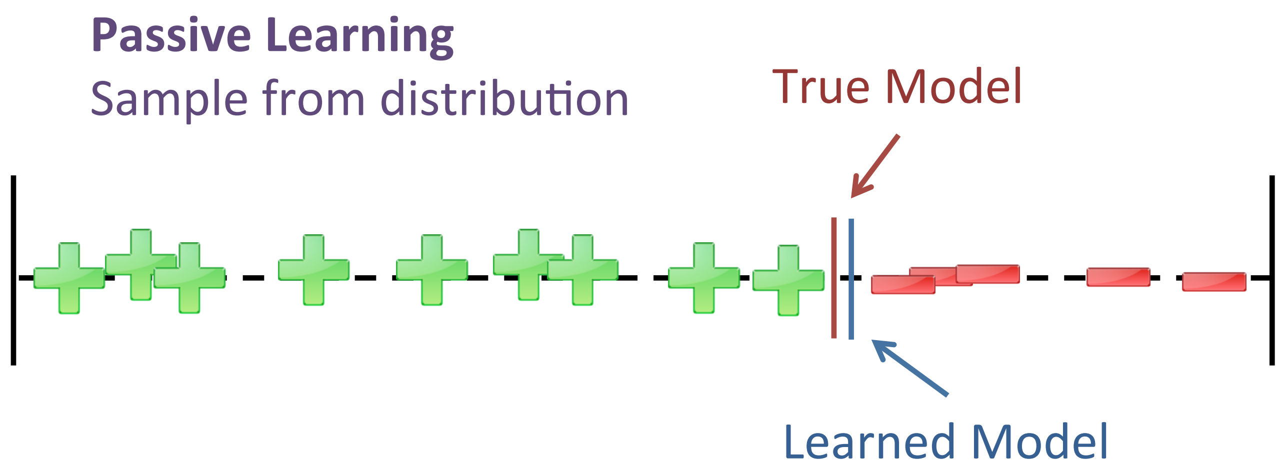
Basic Active Learning
(for supervised learning)

- For t = 1….
  - Algorithm chooses x
  - World reveals associated label y
  - Add (x,y) to training set

- Terminate when sufficiently confident of best model

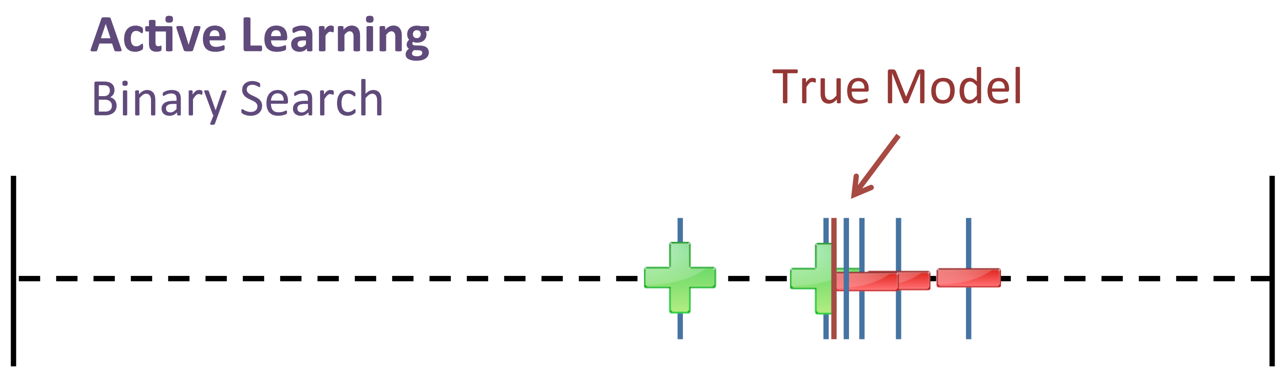
Simple Example

- 1 feature
- Learn threshold function



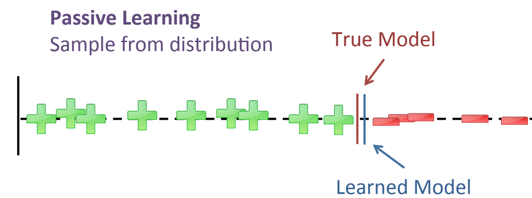
Simple Example

- 1 feature
- Learn threshold function

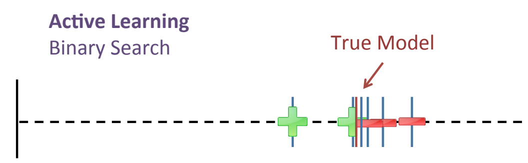


Comparison with Passive Learning

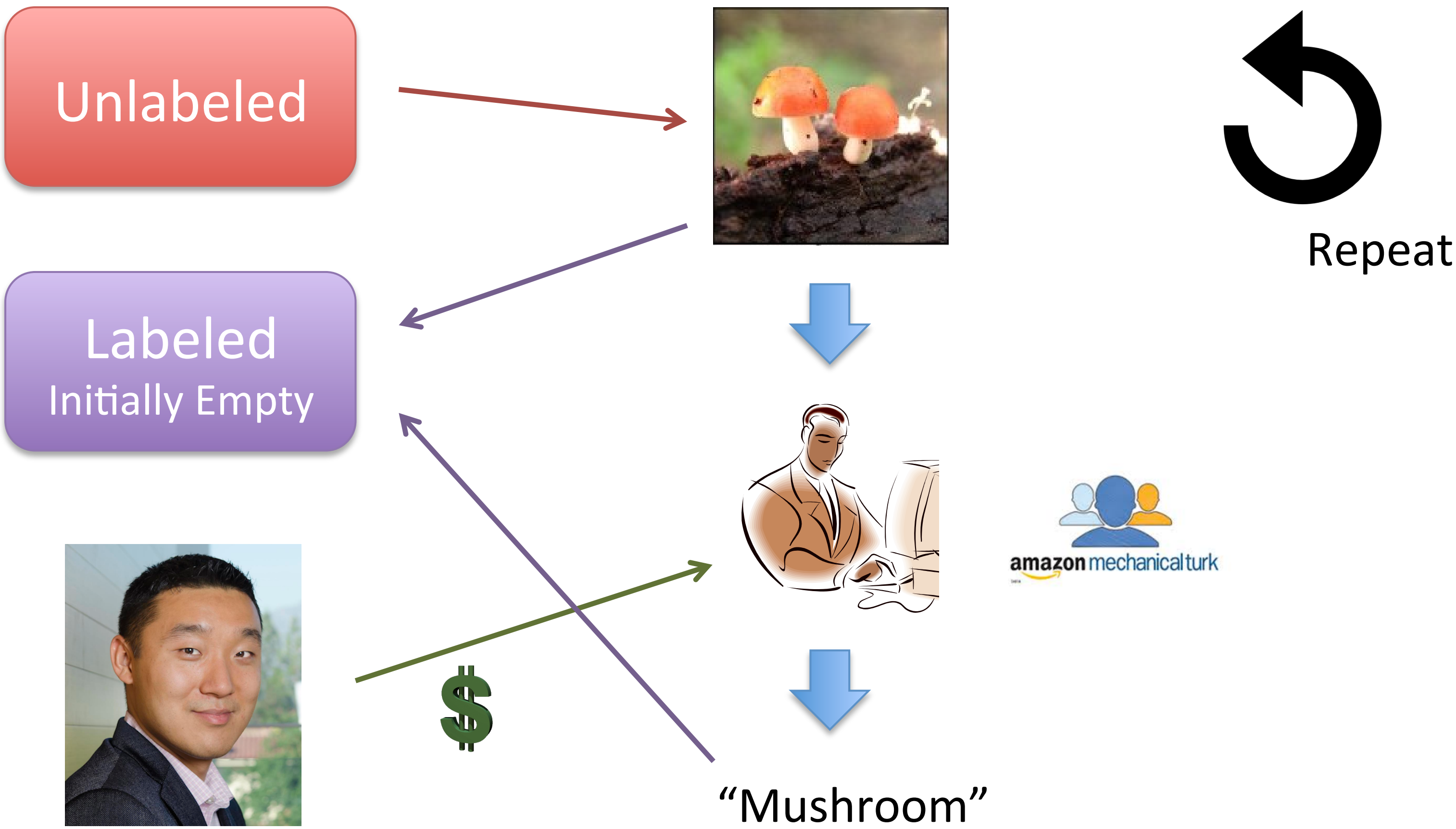
- # samples to be within ε of true model
- Passive Learning: $O\left(\frac{1}{\varepsilon}\right)$



- Active Learning: $O\left(\log\frac{1}{\varepsilon}\right)$



Crowdsourcing



How Reliable are Annotators?

- If we knew what the labels were
  - Can judge workers on label quality
- If we knew who the good workers were
  - Can create labels from their annotations
- **Chicken and egg problem!**

Reinforcement Learning

- In previous settings:
  - Actions do not impact state
  - "Stateless"

- Reinforcement Learning
  - Actions effect state you're in
  - Reward function depends on state
  - Example: Playing Go

Off-Policy Evaluation

- Example: We have hospital logs of pneumonia deaths under various conditions.
  - Want to train model predict who is most at risk
  - Model predicts that asthma patients have LOWER risk for pneumonia death….
  - Because doctors pay closer attention to asthma patients!

Modeling Human Decision Making

- How do humans react in sequential decision making processes?
  - Do they behave like follow the leader?
  - Do they behave like a perceptron?