Learning as Loss Minimization

Machine Learning

THE UNIVERSITY OF UTAH

Learning as loss minimization

- The setup
  - Examples x drawn from a fixed, unknown distribution D
  - Hidden oracle classifier f labels examples
  - We wish to find a hypothesis h that mimics f

Learning as loss minimization

- The setup
  - Examples x drawn from a fixed, unknown distribution D
  - Hidden oracle classifier f labels examples
  - We wish to find a hypothesis h that mimics f
- The ideal situation
  - Define a function L that penalizes bad hypotheses
  - **Learning:** Pick a function $h \in H$ to minimize expected loss

Learning as loss minimization

- The setup
  - Examples x drawn from a fixed, unknown distribution D
  - Hidden oracle classifier f labels examples
  - We wish to find a hypothesis h that mimics f
- The ideal situation
  - Define a function L that penalizes bad hypotheses
  - **Learning:** Pick a function $h \in H$ to minimize expected loss

$$\min_{h \in H} E_{x \sim D}\left[L\big(h(x), f(x)\big)\right]$$

Learning as loss minimization

- The setup
  - Examples x drawn from a fixed, unknown distribution D
  - Hidden oracle classifier f labels examples
  - We wish to find a hypothesis h that mimics f
- The ideal situation
  - Define a function L that penalizes bad hypotheses
  - **Learning:** Pick a function $h \in H$ to minimize expected loss

$$\min_{h \in H} E_{x \sim D}\left[L\big(h(x), f(x)\big)\right]$$

But distribution D is unknown

Learning as loss minimization

- The setup
  - Examples x drawn from a fixed, unknown distribution D
  - Hidden oracle classifier f labels examples
  - We wish to find a hypothesis h that mimics f
- The ideal situation
  - Define a function L that penalizes bad hypotheses
  - **Learning:** Pick a function $h \in H$ to minimize expected loss

$$\min_{h \in H} E_{x \sim D}\left[L\big(h(x), f(x)\big)\right]$$

But distribution D is unknown

- Instead, minimize *empirical loss* on the training set

$$\min_{h \in H} \frac{1}{m} \sum_{i} L\big(h(x_i), f(x_i)\big)$$

Empirical loss minimization

Learning = minimize *empirical loss* on the training set

$$\min_{h \in H} \frac{1}{m} \sum_{i} L\big(h(x_i), f(x_i)\big)$$

Is there a problem here?

Empirical loss minimization

Learning = minimize *empirical loss* on the training set

$$\min_{h \in H} \frac{1}{m} \sum_i L\big(h(x_i), f(x_i)\big)$$

Is there a problem here? Overfitting!

We need something that biases the learner towards simpler hypotheses

- Achieved using a regularizer, which penalizes complex hypotheses

Regularized loss minimization

- Learning:

$$\min_{h \in H} \left(\text{regularizer}(h) + C \frac{1}{m} \sum_i L\big(h(x_i), f(x_i)\big) \right)$$

Regularized loss minimization

- Learning:

$$\min_{h \in H} \left(\text{regularizer}(h) + C \frac{1}{m} \sum_i L\big(h(x_i), f(x_i)\big) \right)$$

- With linear classifiers:

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_i L(y_i, \mathbf{x}_i, \mathbf{w})$$

Regularized loss minimization

- Learning:

$$\min_{h \in H} \left(\text{regularizer}(h) + C \frac{1}{m} \sum_i L\big(h(x_i), f(x_i)\big) \right)$$

- With linear classifiers:

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{C}{m} \sum_i L(y_i, \mathbf{x}_i, \mathbf{w})$$

- What is a loss function?
  - Loss functions should penalize mistakes
  - We are minimizing average loss over the training data

Regularized loss minimization

- Learning:

$$\min_{h \in H} \left(\text{regularizer}(h) + C \frac{1}{m} \sum_i L\big(h(x_i), f(x_i)\big) \right)$$

- With linear classifiers:

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{C}{m} \sum_i L(y_i, \mathbf{x}_i, \mathbf{w})$$

- What is a loss function?
  - Loss functions should penalize mistakes
  - We are minimizing average loss over the training data

- What is the ideal loss function for classification?

The 0-1 loss

Penalize classification mistakes between true label y and prediction y'

$$L_{0-1}(y, y') = \begin{cases} 1 & \text{if } y \neq y' \\ 0 & \text{if } y = y' \end{cases}$$

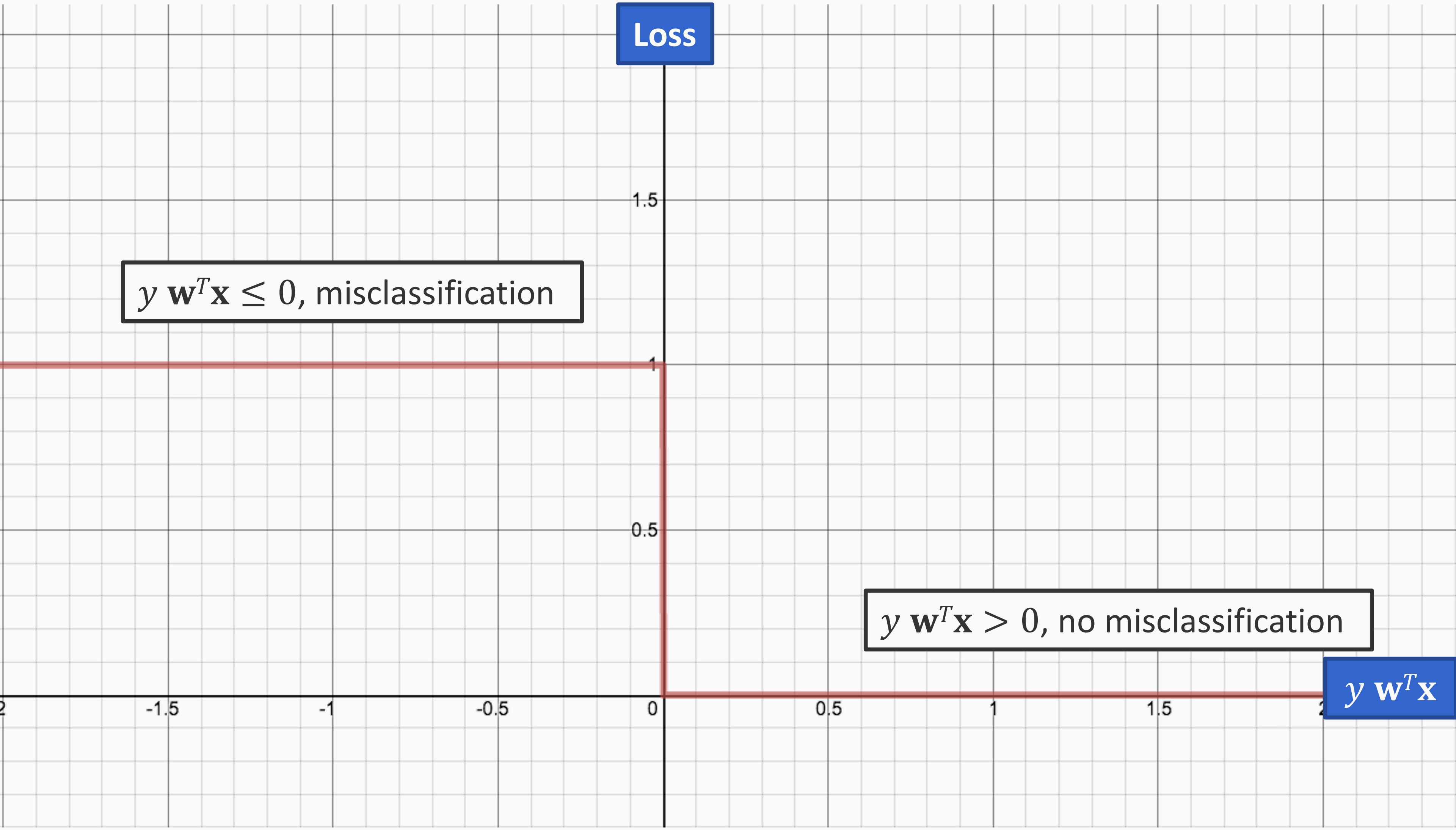
For linear classifiers, the prediction $y' = \text{sgn}(\mathbf{w}^T\mathbf{x})$

– Mistake if $y\ \mathbf{w}^T\mathbf{x} \leq\ 0$

$$L_{0-1}(y, y') = \begin{cases} 1 & \text{if } y\mathbf{w}^T\mathbf{x} \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

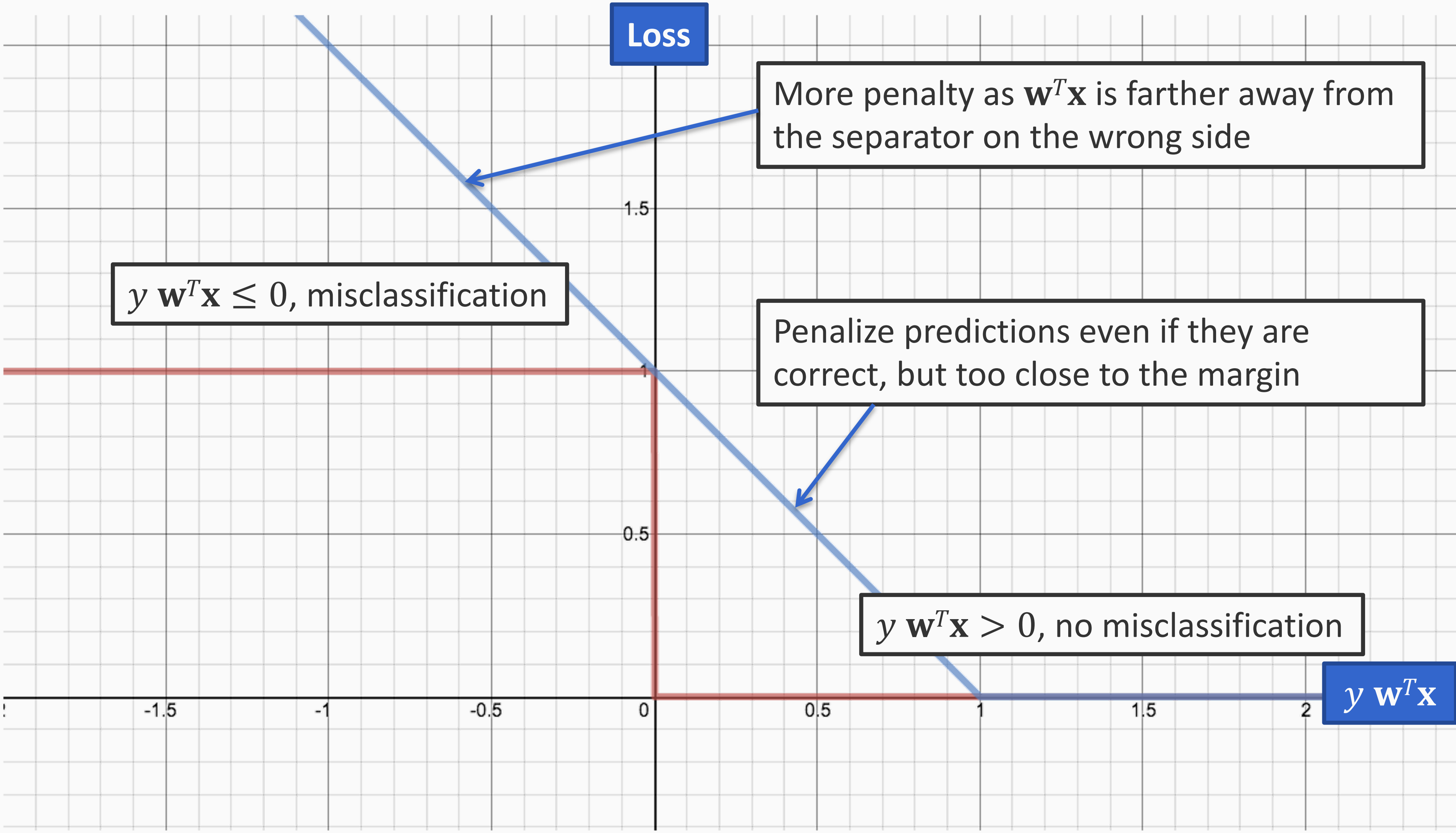
Minimizing 0-1 loss is intractable. Need surrogates

The 0-1 loss



Loss
$y\ \mathbf{w}^T\mathbf{x} \leq 0$, misclassification
$y\ \mathbf{w}^T\mathbf{x} > 0$, no misclassification
$y\ \mathbf{w}^T\mathbf{x}$
1.5
1
0.5
-1.5
-1
-0.5
0
0.5
1
1.5

Compare to the hinge loss



Support Vector Machines

- SVM = linear classifier combined with regularization

- Ideally, we would like to minimize 0-1 loss,
  – But we can’t for computational reasons

- SVM minimizes hinge loss

$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}) = \max(0, 1 - y\mathbf{w}^T\mathbf{x})$$

  – Variants exist

SVM objective function

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_i \max(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i)$$

Regularization term:
- Maximize the margin
- Imposes a preference over the hypothesis space and pushes for better generalization
- Can be replaced with other regularization terms which impose other preferences

Empirical Loss:
- Hinge loss
- Penalizes weight vectors that make mistakes
- Can be replaced with other loss functions which impose other preferences

SVM objective function

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_i \max(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i)$$

Regularization term:
- Maximize the margin
- Imposes a preference over the hypothesis space and pushes for better generalization
- Can be replaced with other regularization terms which impose other preferences

Empirical Loss:
- Hinge loss
- Penalizes weight vectors that make mistakes
- Can be replaced with other loss functions which impose other preferences

A hyper-parameter that controls the tradeoff between a large margin and a small hinge-loss

$$\min_{h \in H} \text{regularizer}(h) + C\frac{1}{m}\sum_i L(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

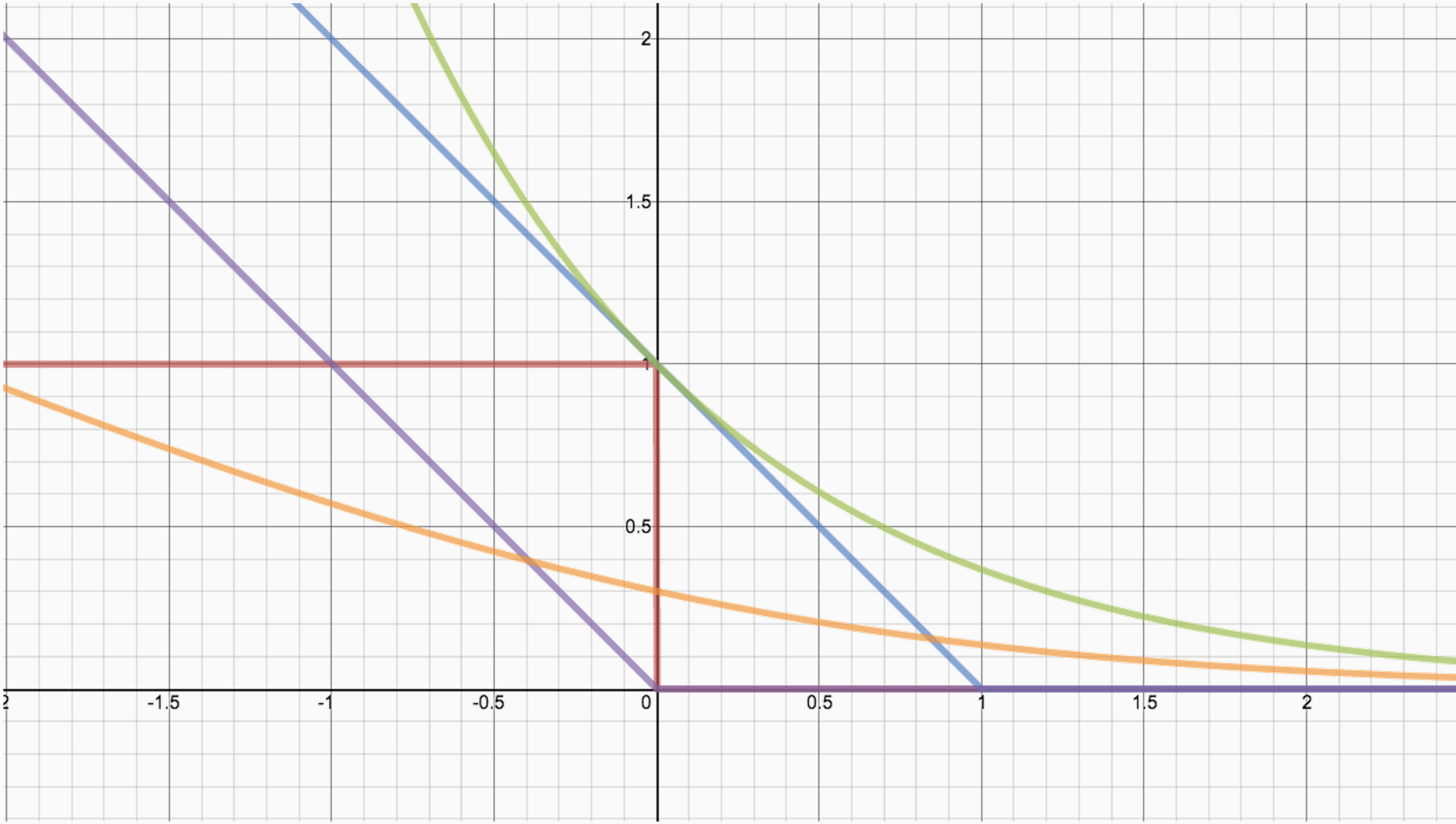
The loss function zoo

Many loss functions exist

- Perceptron loss $$L_{Perceptron}(y, \mathbf{x}, \mathbf{w}) = \max(0, -y\mathbf{w}^T\mathbf{x})$$
- Hinge loss (SVM) $$L_{Hinge}(y, \mathbf{x}, \mathbf{w}) = \max(0, 1 - y\mathbf{w}^T\mathbf{x})$$
- Exponential loss (AdaBoost) $$L_{Exponential}(y, \mathbf{x}, \mathbf{w}) = e^{-y\mathbf{w}^T\mathbf{x}}$$
- Logistic loss (logistic regression) $$L_{Logistic}(y, \mathbf{x}, \mathbf{w}) = \log(1 + e^{-y\mathbf{w}^T\mathbf{x}})$$

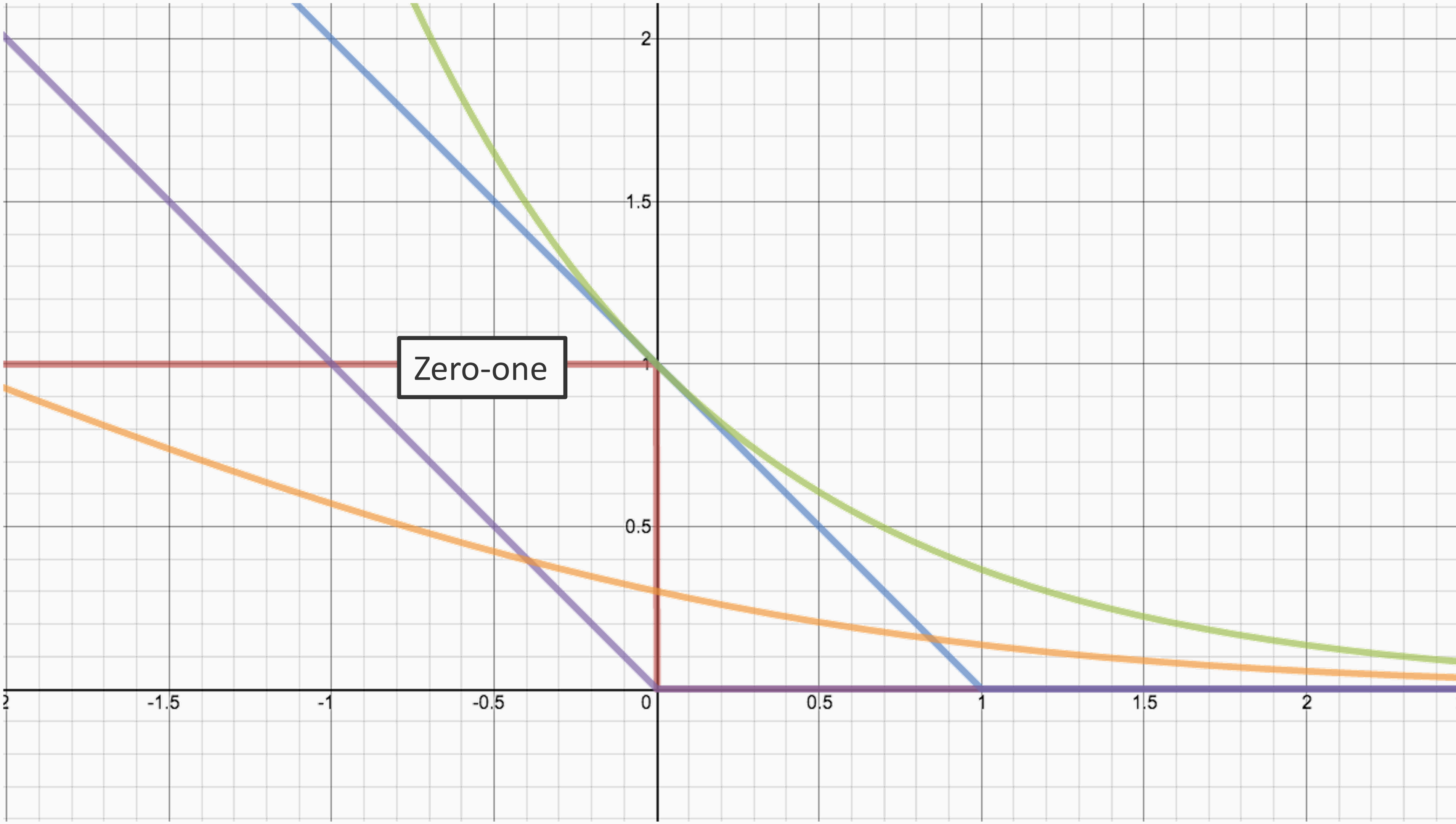
$$\min_{h \in H} \text{regularizer}(h) + C\frac{1}{m}\sum_i L(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

The loss function zoo



$$\min_{h \in H} \text{regularizer}(h) + C \frac{1}{m} \sum_i L(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

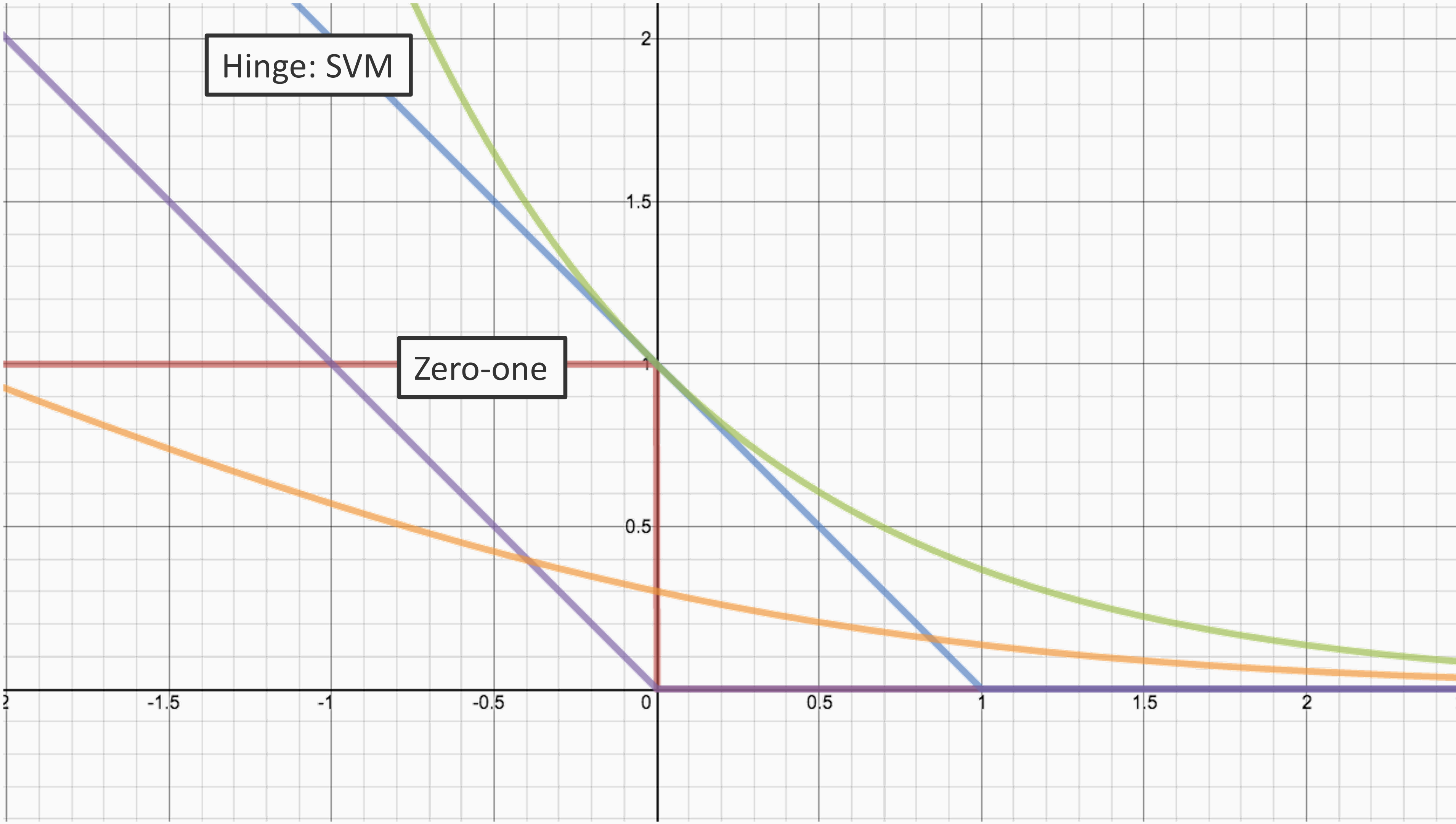
The loss function zoo



$$\min_{h \in H} \text{regularizer}(h) + C\frac{1}{m}\sum_i L(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

The loss function zoo

$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}) = \max(0, 1 - y\mathbf{w}^T\mathbf{x})$$

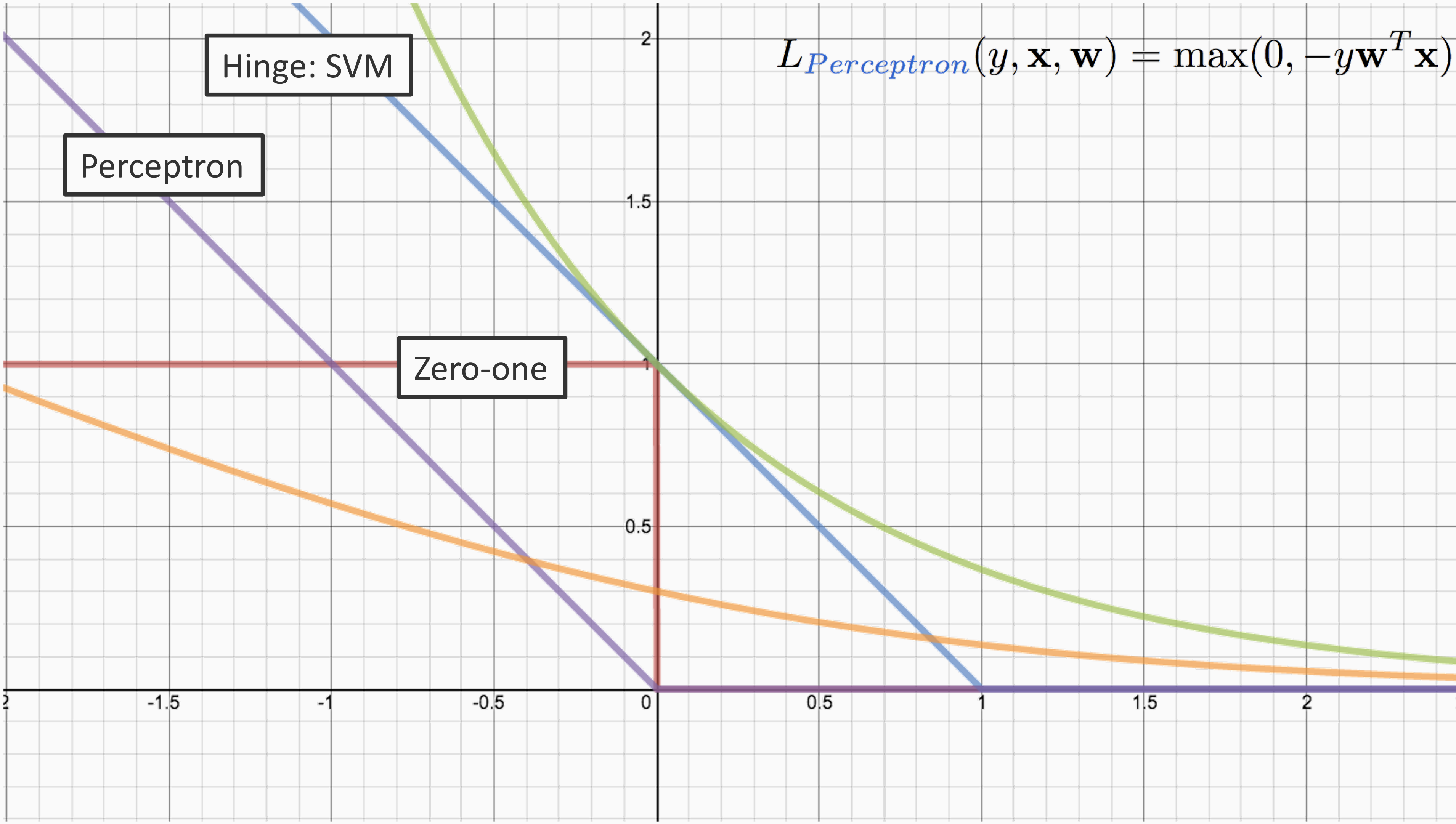


$$\min_{h \in H} \text{regularizer}(h) + C\frac{1}{m}\sum_i L(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

The loss function zoo

$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}) = \max(0, 1 - y\mathbf{w}^T\mathbf{x})$$

$$L_{Perceptron}(y, \mathbf{x}, \mathbf{w}) = \max(0, -y\mathbf{w}^T\mathbf{x})$$



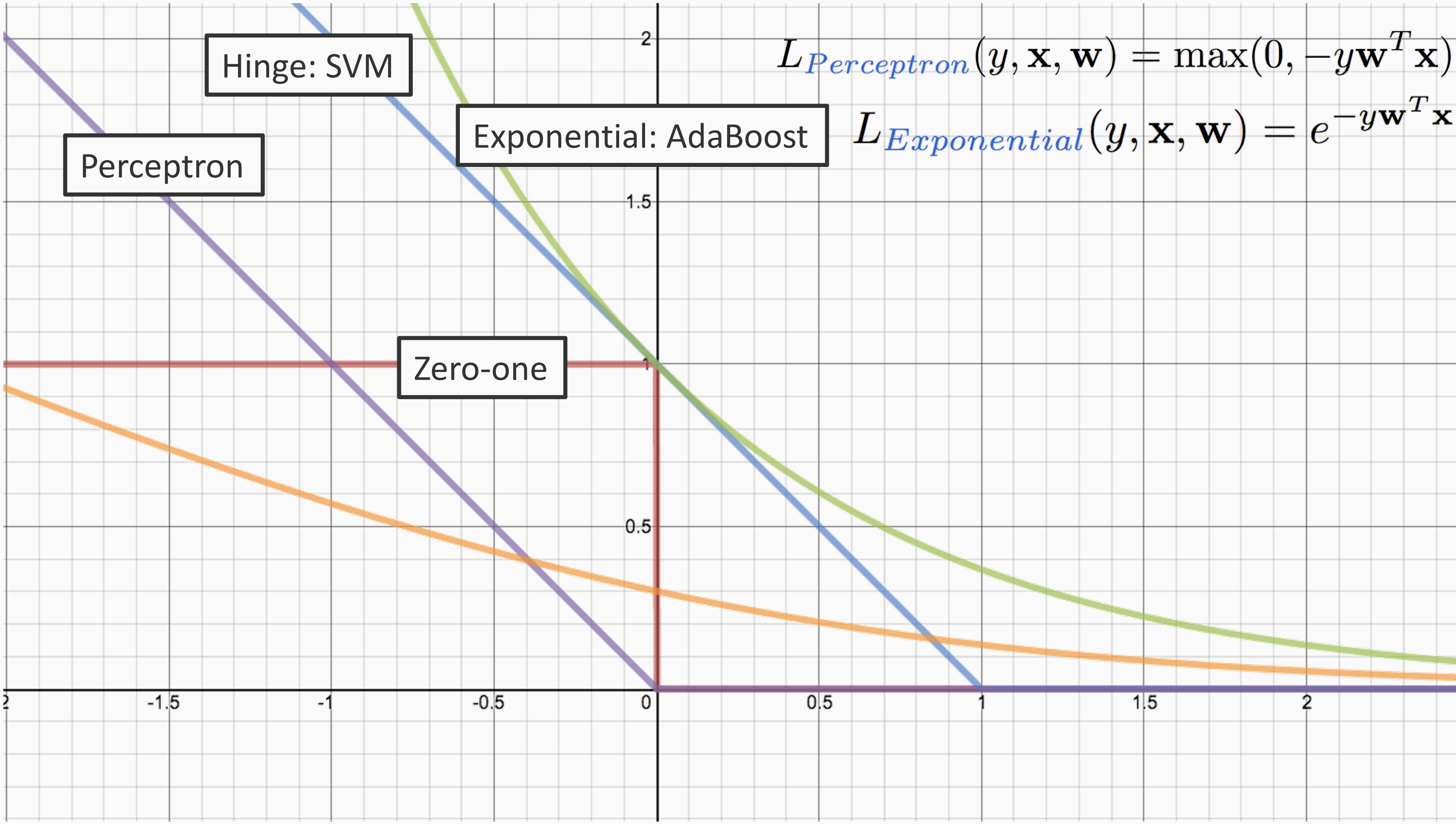
$$\min_{h \in H} \text{regularizer}(h) + C\frac{1}{m}\sum_i L(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

The loss function zoo

$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}) = \max(0, 1 - y\mathbf{w}^T\mathbf{x})$$

$$L_{Perceptron}(y, \mathbf{x}, \mathbf{w}) = \max(0, -y\mathbf{w}^T\mathbf{x})$$

$$L_{Exponential}(y, \mathbf{x}, \mathbf{w}) = e^{-y\mathbf{w}^T\mathbf{x}}$$



$$\min_{h \in H} \text{regularizer}(h) + C\frac{1}{m}\sum_i L(h(\mathbf{x}_i), f(\mathbf{x}_i))$$

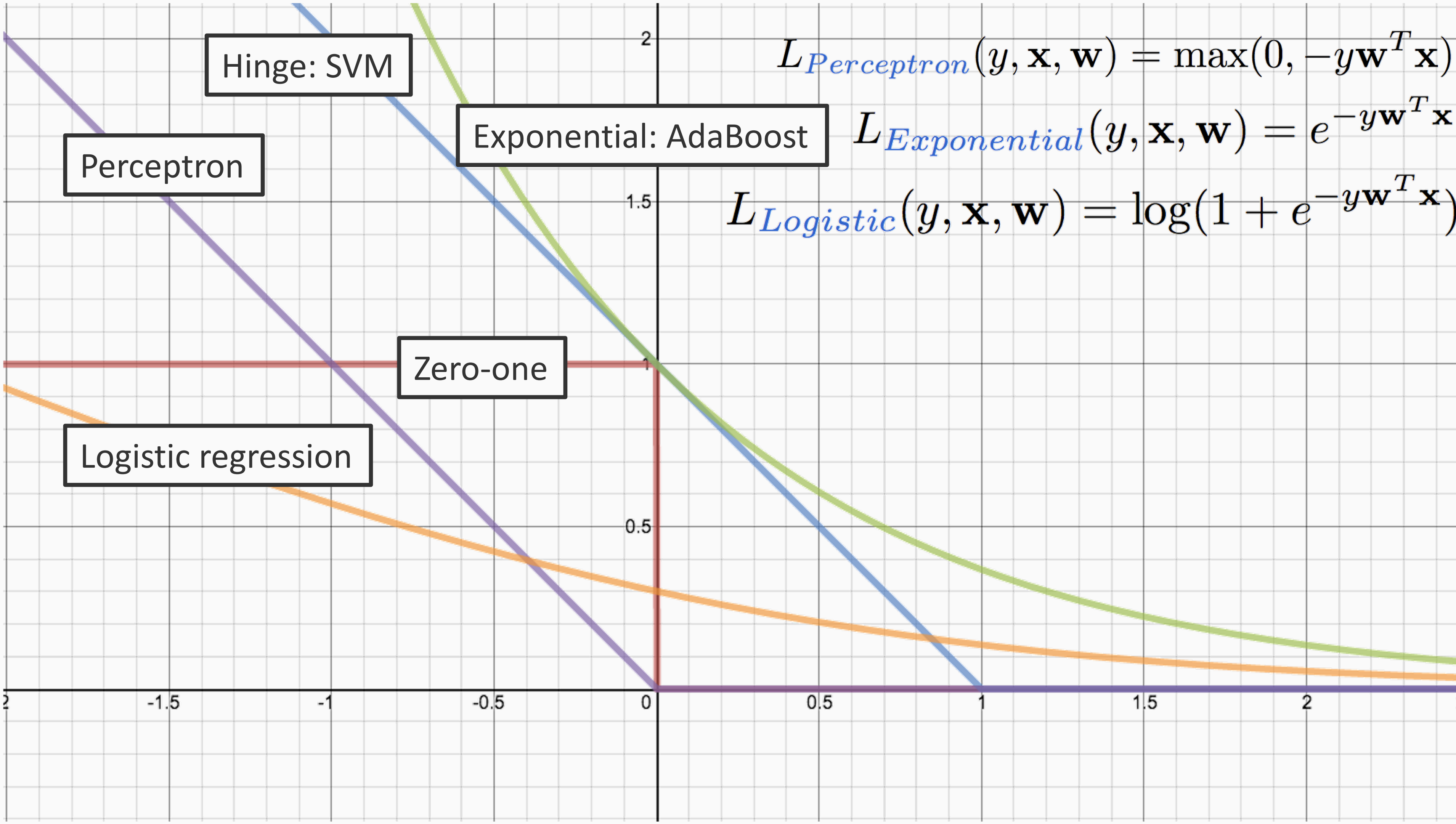
The loss function zoo

$$L_{Hinge}(y, \mathbf{x}, \mathbf{w}) = \max(0, 1 - y\mathbf{w}^T\mathbf{x})$$

$$L_{Perceptron}(y, \mathbf{x}, \mathbf{w}) = \max(0, -y\mathbf{w}^T\mathbf{x})$$

$$L_{Exponential}(y, \mathbf{x}, \mathbf{w}) = e^{-y\mathbf{w}^T\mathbf{x}}$$

$$L_{Logistic}(y, \mathbf{x}, \mathbf{w}) = \log(1 + e^{-y\mathbf{w}^T\mathbf{x}})$$



Learning via Loss Minimization: Summary

- Learning via Loss Minimization
  - Write down a loss function
  - Minimize empirical loss
- Regularize to avoid overfitting
  - Neural networks use other strategies such as dropout
- Widely applicable, different loss functions and regularizers