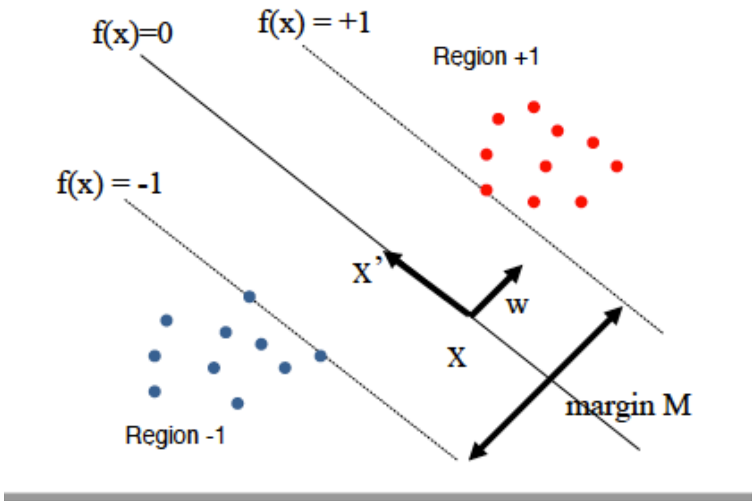
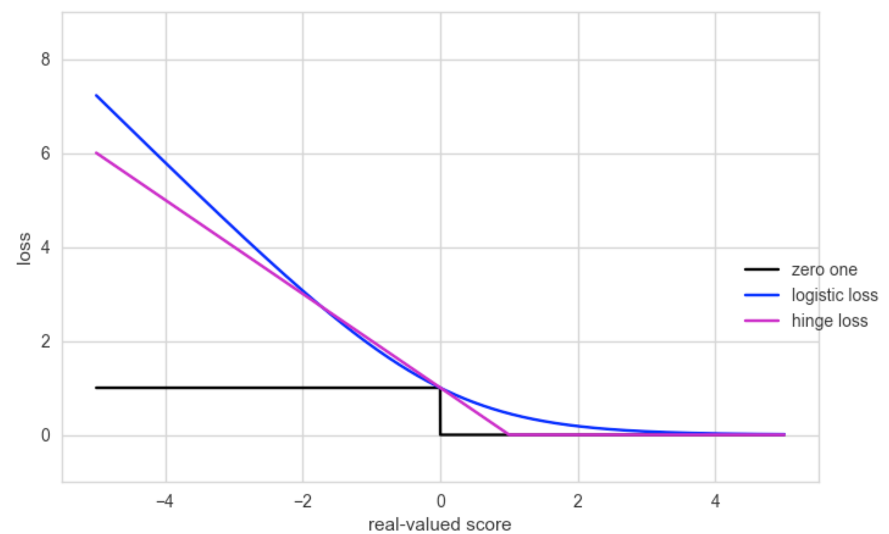
Tufts COMP 135: Introduction to Machine Learning
https://www.cs.tufts.edu/comp/135/2020f/

Support Vector Machines





Prof. Mike Hughes

Many ideas/slides attributable to:
Dan Sheldon (U.Mass.), Erik Sudderth (UCI), Liping Liu (Tufts)
James, Witten, Hastie, Tibshirani (ISL/ESL books)

SVM Objectives (day 17)

Support Vector Machine classifier

- Why maximize margin?
- What is a support vector?
- What is hinge loss?

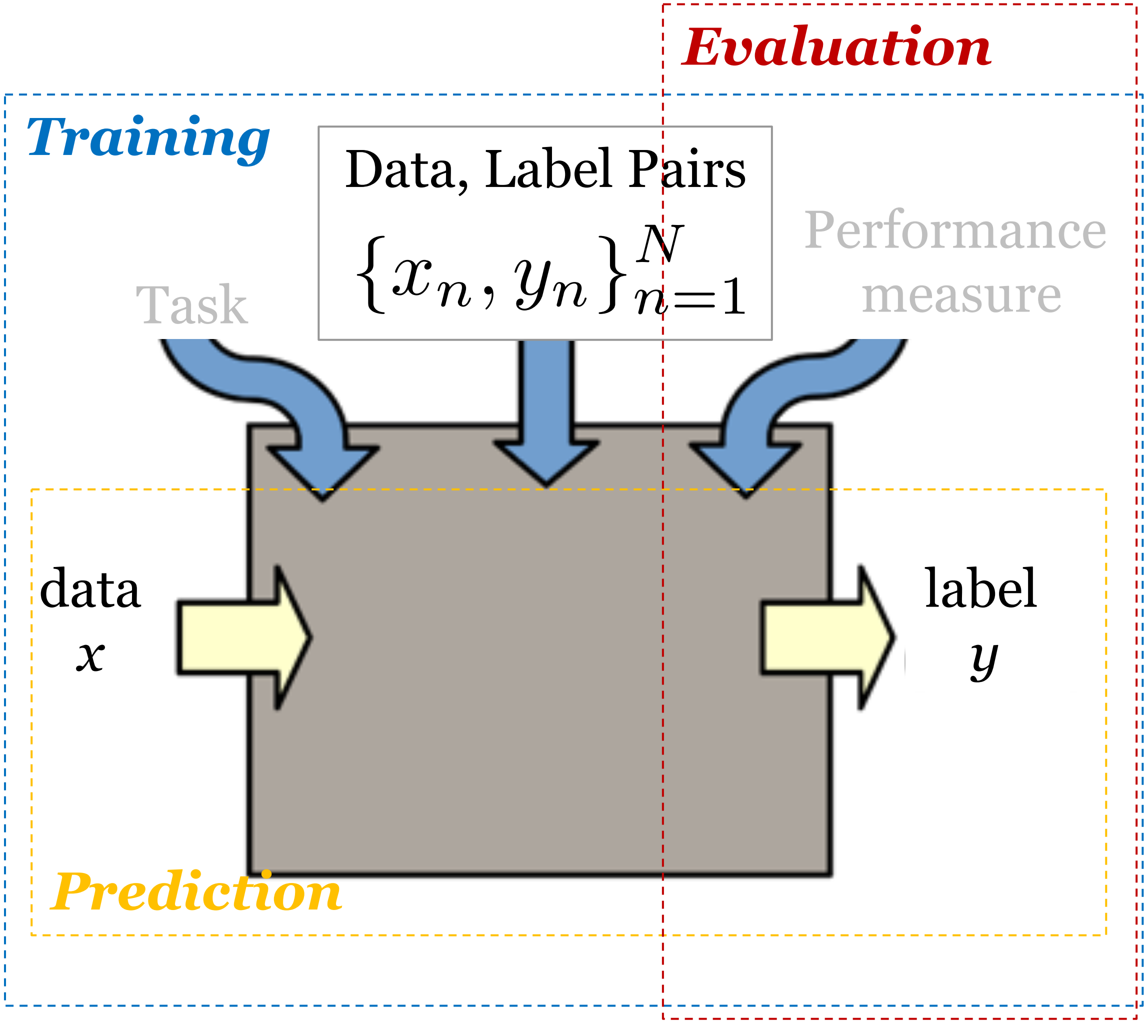
- Advantages over logistic regression
  - Less sensitive to outliers
  - Advantages from sparsity in when using kernels
- Disadvantages
  - Not probabilistic
  - Less elegant to do multi-class

What will we learn?

Supervised Learning

Unsupervised Learning

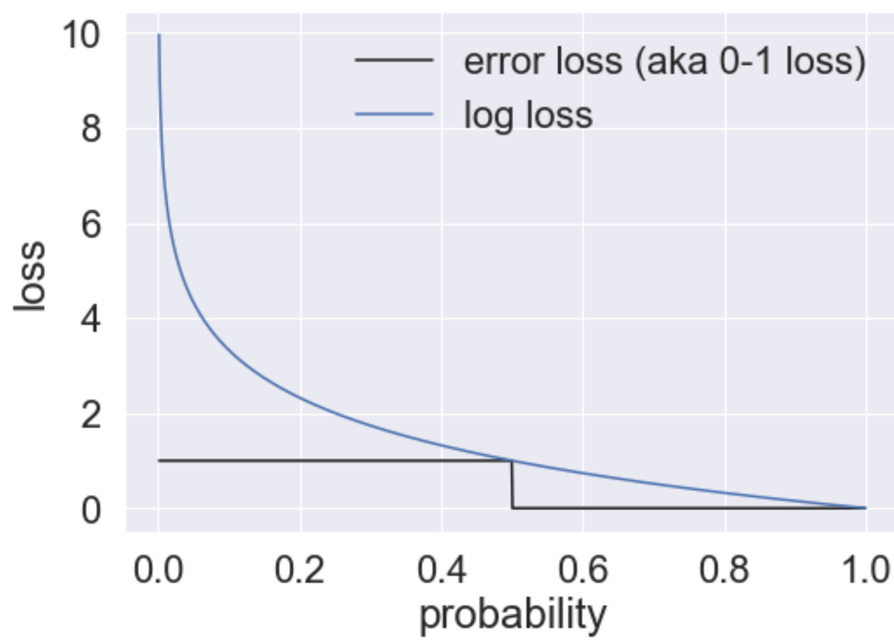
Reinforcement Learning



Downsides of Logistic Regression

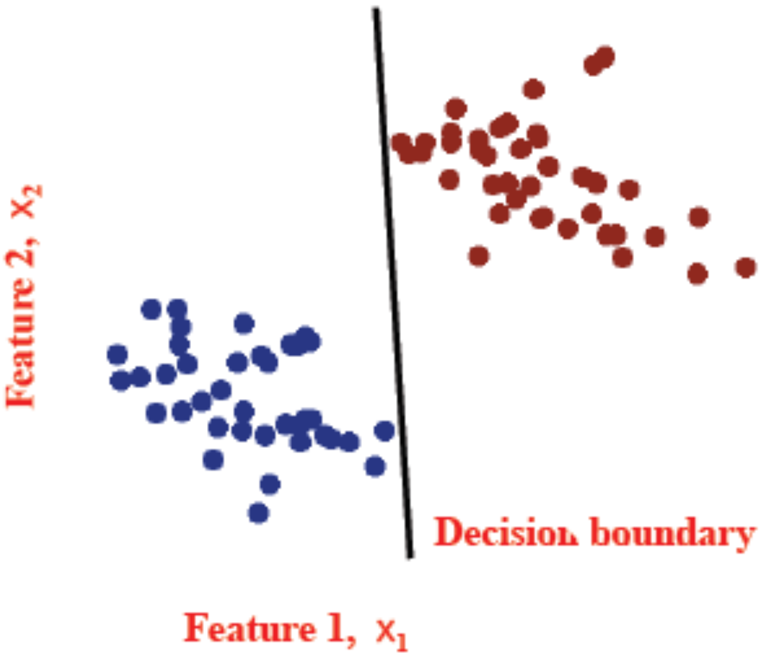
Logistic regression minimizes log loss, where any example that is *misclassified* pays a *steep cost*.

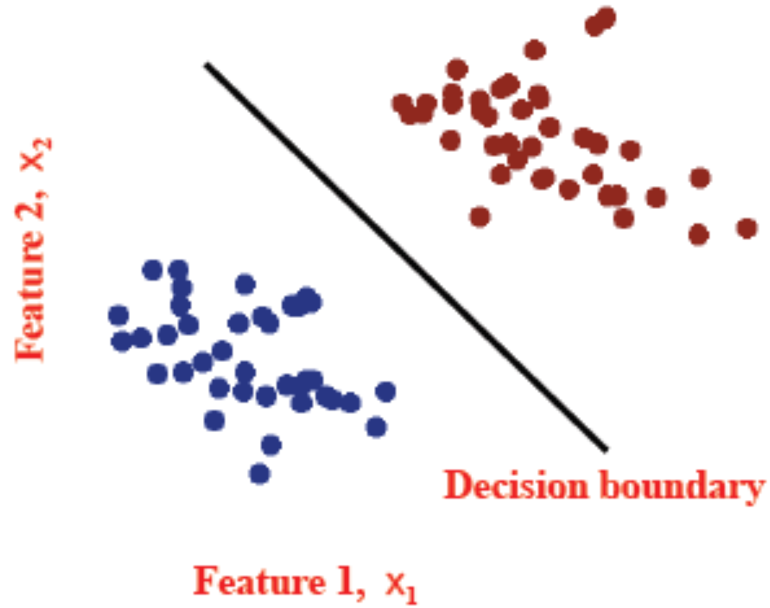
Thus, this loss function is **sensitive to outliers.**
One training example (x, y) can impact optimal weights a lot.



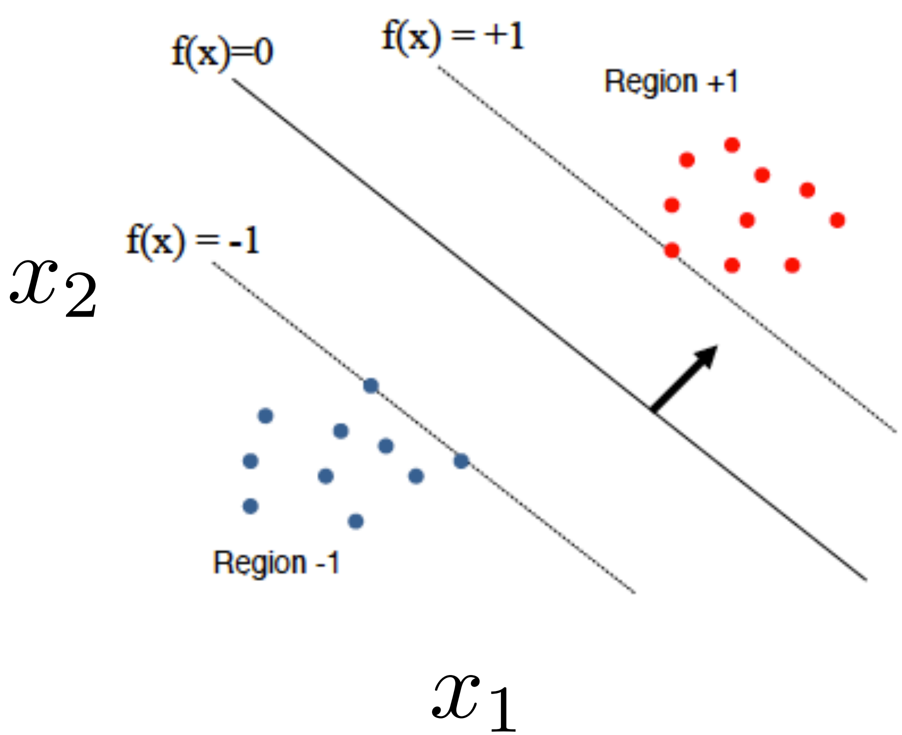
Stepping back

Which do we prefer? Why?





Idea: Define binary *regions* separated by wide *margin*



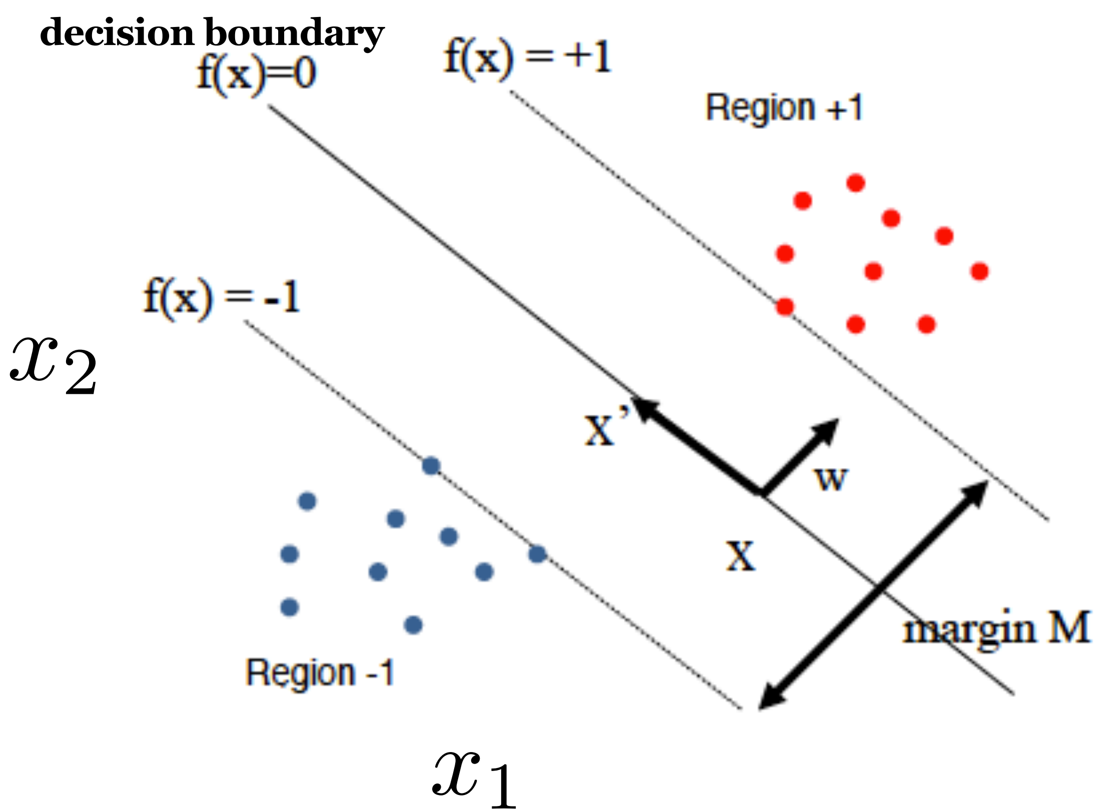
We could define such a function:

$$f(x) = w_1 x_1 + w_2 x_2 + b$$

f(x) > +1 in region +1
f(x) < –1 in region –1

Passes through zero in center…

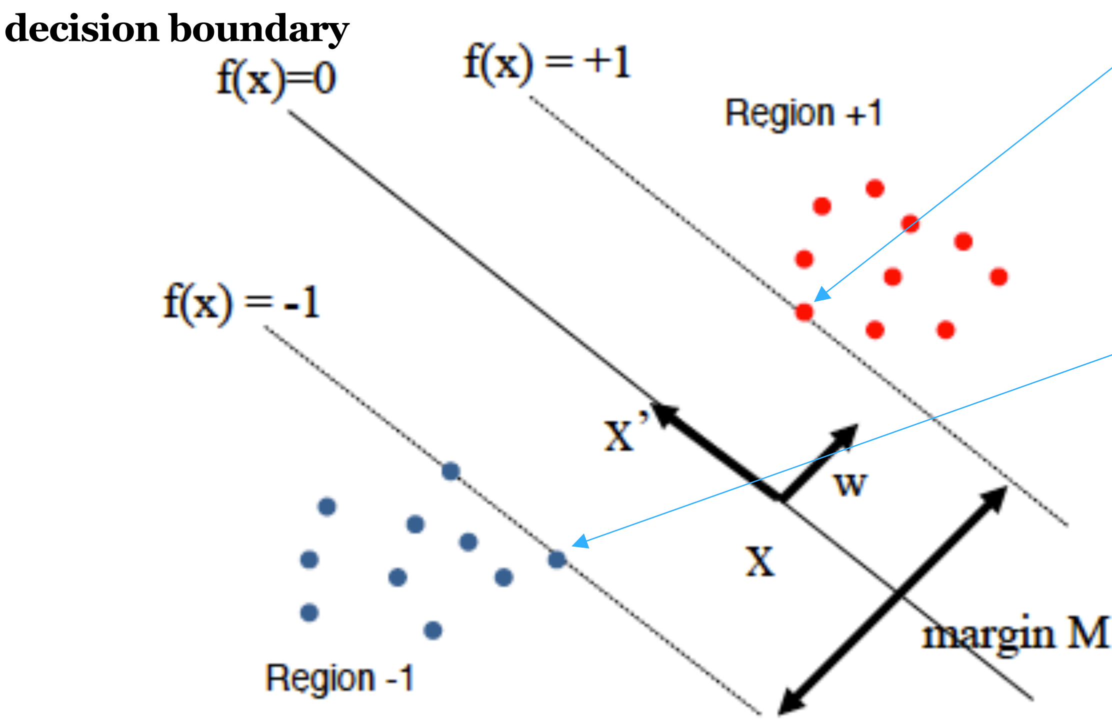
Weight vector w is <u>perpendicular</u> to boundary



$$f(x) = w_1 x_1 + w_2 x_2 + b$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

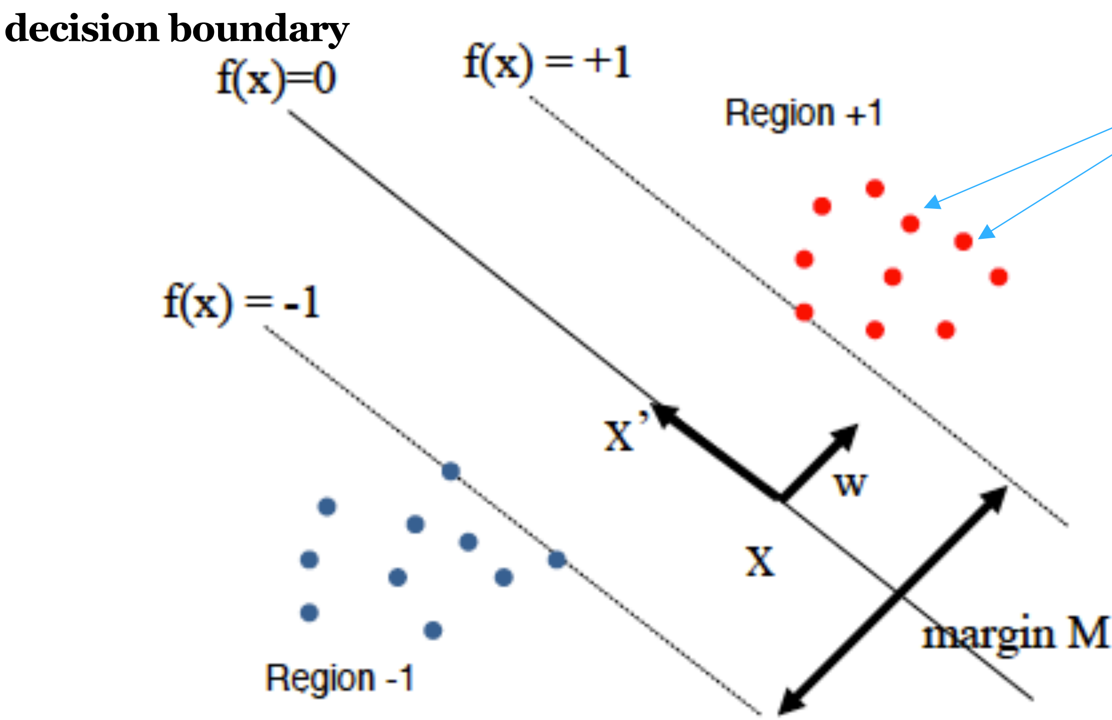
Examples that define the margin are called **support** (feature) **vectors**



Nearest positive example
x_+

Nearest negative example
x_-

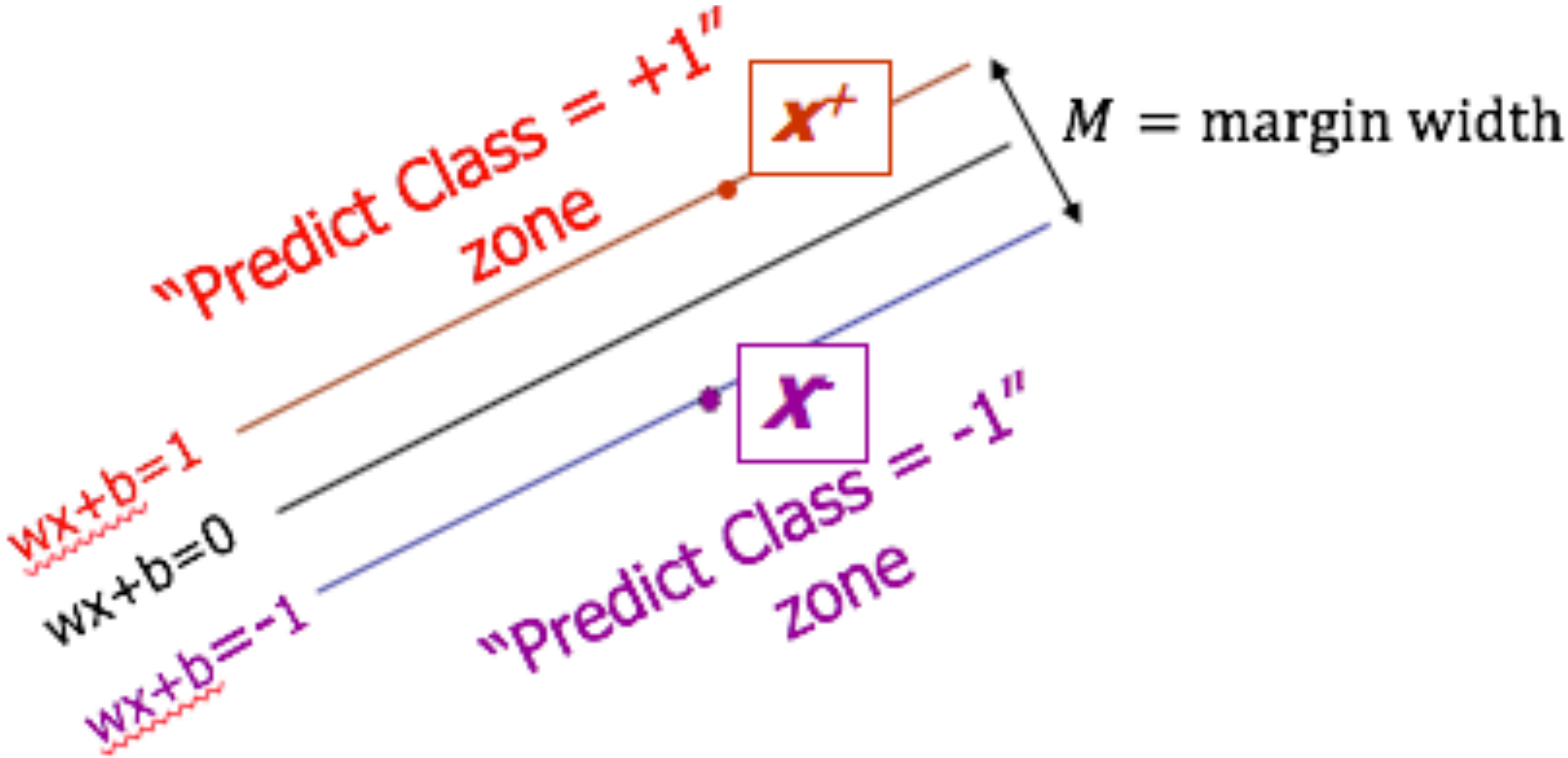
Observation: Non-support training examples do not influence margin ***at all***



Could perturb these examples slightly without impacting boundary

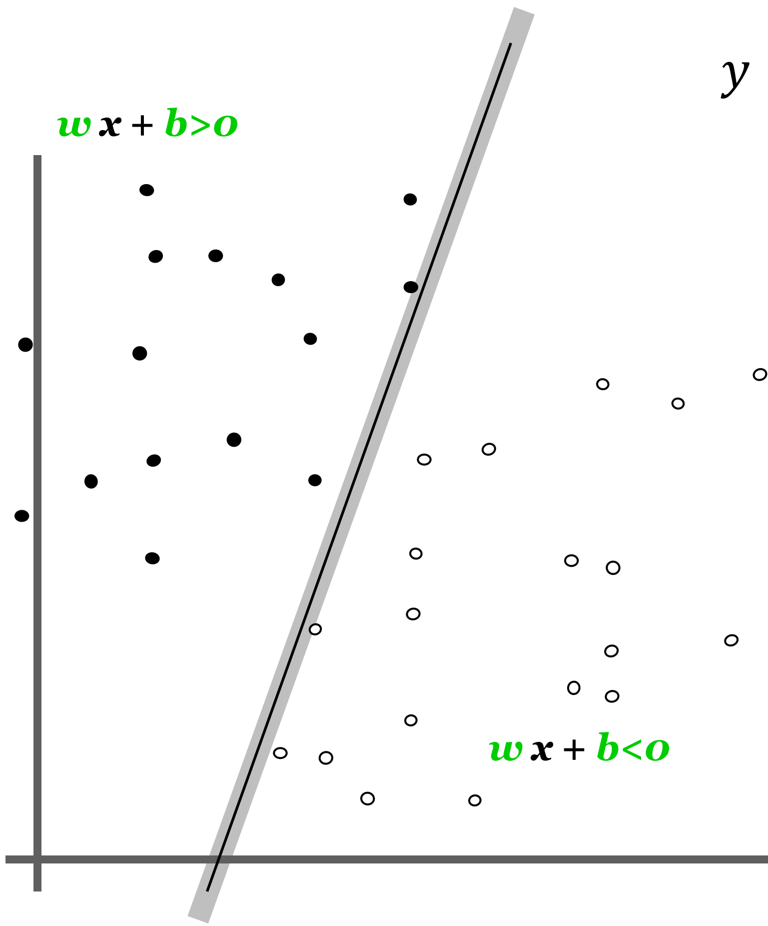
Only a **small** fraction of all training examples are support vectors.
If we can efficiently identify these vectors, model training (finding weights) might be very fast.

How wide is the margin?



Small margin

- y positive
- ◦ y negative

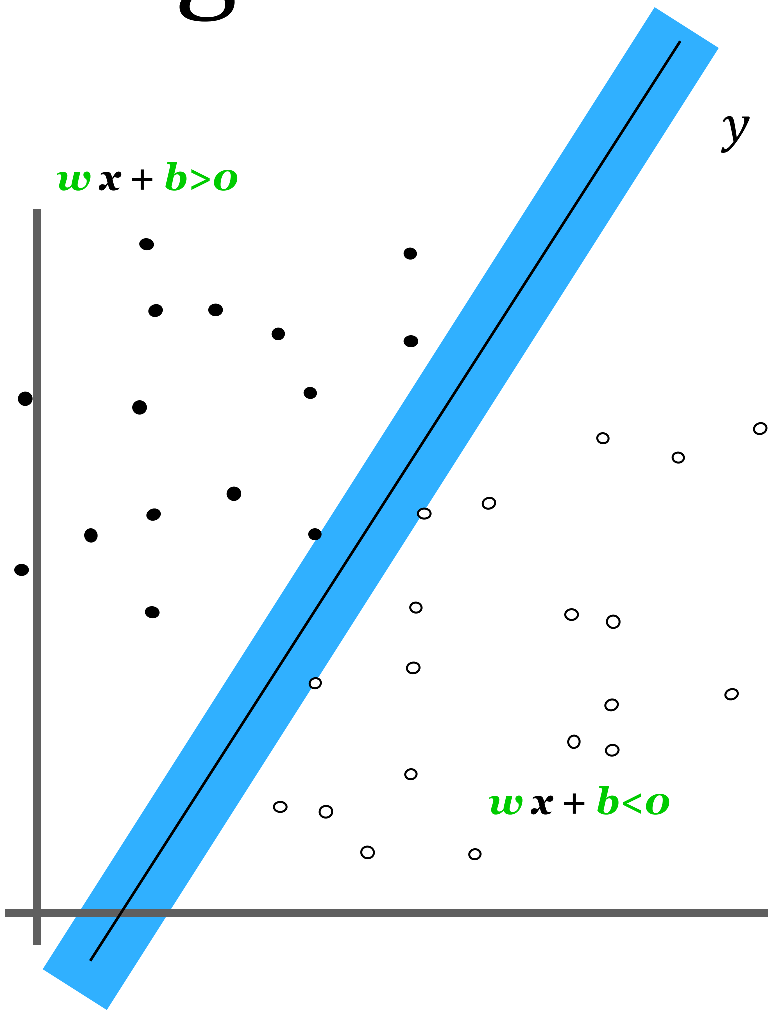


$$y = \begin{cases} +1 & if\ \boldsymbol{w}\,\boldsymbol{x} + b \geq 0 \\ -1 & if\ \boldsymbol{w}\,\boldsymbol{x} + b < 0 \end{cases}$$

Margin: distance to the boundary

Large margin

- y positive
- ◦ y negative



$$y = \begin{cases} +1 & if\ \boldsymbol{w}\,\boldsymbol{x} + b \geq 0 \\ -1 & if\ \boldsymbol{w}\,\boldsymbol{x} + b < 0 \end{cases}$$

Margin: distance to the boundary

How wide is the margin?

Distance from nearest positive example to nearest negative example along vector w:

$$M(w) = \frac{(x_+ - x_-)^T w}{||w||_2} = \frac{(x_+ - x_-)^T w}{\sqrt{w_1^2 + \ldots w_F^2}}$$

The scalar projection of $\overline{a}$ on $\overline{b}$ is the magnitude of the vector projection of $\overline{a}$ on $\overline{b}$.

$$|proj_{\overline{b}}\overline{a}| = \frac{\overline{a} \cdot \overline{b}}{|\overline{b}|}$$

How wide is the margin?

Distance from nearest positive example to nearest negative example along vector w:

$$M(w) = \frac{(x_+ - x_-)^T w}{||w||_2} = \frac{(x_+ - x_-)^T w}{\sqrt{w_1^2 + \dots w_F^2}}$$

By construction, we assume

$$w^T x_+ + b = +1$$

$$w^T x_- + b = -1$$

$$w^T (x_+ - x_-) = 2$$

$$= \frac{2}{||w||_2}$$

Remember that the L2 norm is shorthand for: $\sqrt{w_1^2 + \dots w_F^2}$

SVM Training Problem
Version 1: Hard margin

$$\max_{w,b} \frac{2}{||w||_2}$$

$$\text{subject to} \begin{cases} w^T x_n + b \geq +1 & \text{if } y_n = 1 \\ w^T x_n + b \leq -1 & \text{if } y_n = 0 \end{cases}$$

For each n = 1, 2, ... N

This is a constrained quadratic optimization problem.
There are standard methods to solve this, as well methods specially designed for SVM.

Limitation: Requires **all** training examples to be correctly classified.
Otherwise, no solution exists (at least one constraint violated).
Thus, *hard margin SVM should never be used in practice*.

SVM Training Problem Version 1: Hard margin

$$\min_{w,b} \frac{1}{2} ||w||_2$$

Minimizing the L2 norm ||w|| equivalent to maximizing the margin width (1 / ||w||)

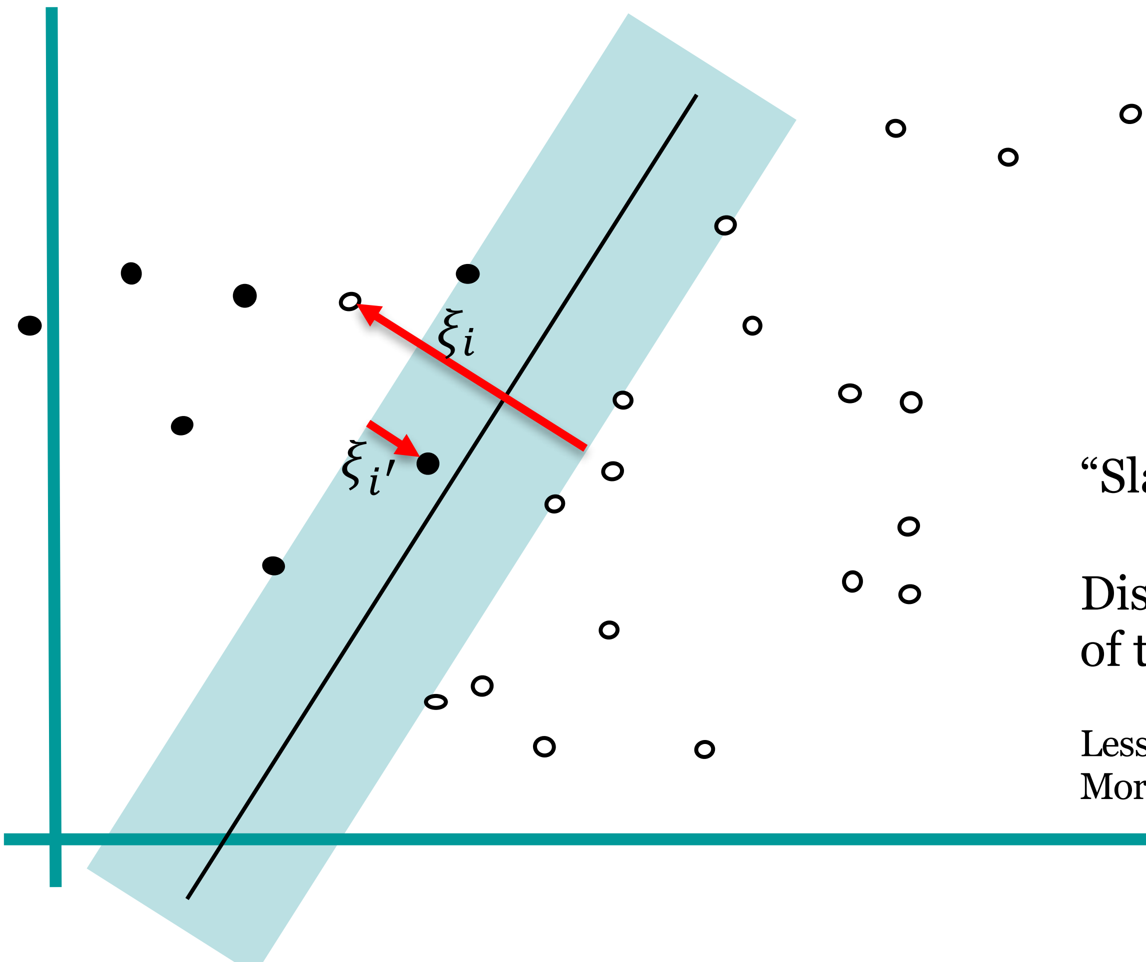
$$\text{subject to} \begin{cases} w^T x_n + b \geq +1 & \text{if } y_n = 1 \\ w^T x_n + b \leq -1 & \text{if } y_n = 0 \end{cases}$$

For each n = 1, 2, ... N

This is a constrained quadratic optimization problem.
There are standard methods to solve this, as well methods specially designed for SVM.

Limitation: Requires **all** training examples to be correctly classified.
Otherwise, no solution exists (at least one constraint violated).
Thus, *hard margin SVM should never be used in practice.*

Soft margin: Allow *some* misclassifications



$$\xi_i \geq 0$$

"Slack" at example *i*

Distance on wrong side of the margin

Less than 1.0: still **correctly classified**
More than 1.0: **misclassified**

Hard vs. soft constraints

HARD: All positive examples must satisfy

$$w^T x_n + b \geq +1$$

SOFT: Want each positive examples to satisfy

$$w^T x_n + b \geq +1 \boxed{-\xi_n} \qquad \xi_i \geq 0$$

with slack as small as possible
(minimize **absolute value**)

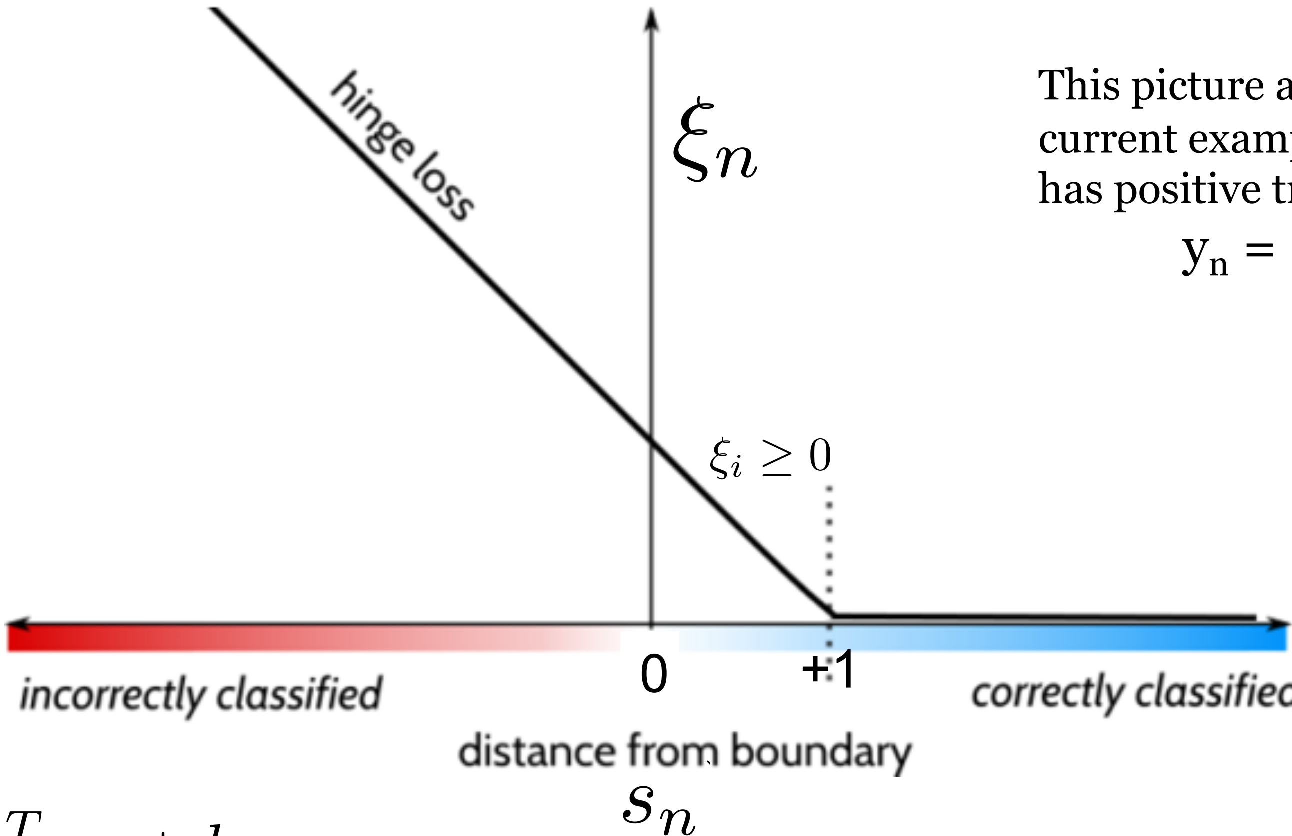
Soft constraint leads to **hinge loss**

Want positive examples to satisfy

$$w^T x_n + b \geq +1 \boxed{-\xi_n}$$

$$\xi_i \geq 0$$

$$\text{hinge\_loss}(y_n, s_n) = \begin{cases} \max(1 - s_n, 0) & \text{if } y_n = 1 \\ \max(1 + s_n, 0) & \text{if } y_n = 0 \end{cases}$$



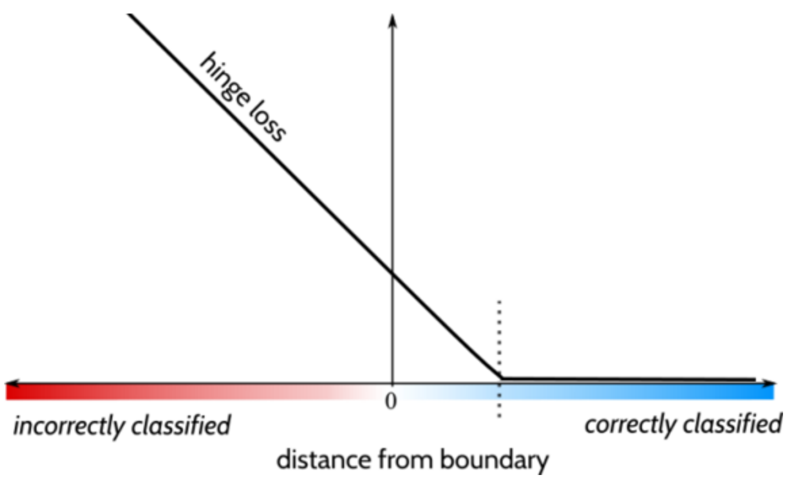
This picture assumes current example has positive true label

$y_n = +1$

$$s_n = w^T x_n + b$$

SVM Training Problem
Version 2: Soft margin

$$\min_{w,b} \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \text{hinge\_loss}(y_n, w^T x_n + b)$$



Tradeoff parameter C
controls model complexity

Smaller C: Simpler model, encourage large margin even if we make lots of mistakes

Bigger C: Avoid mistakes

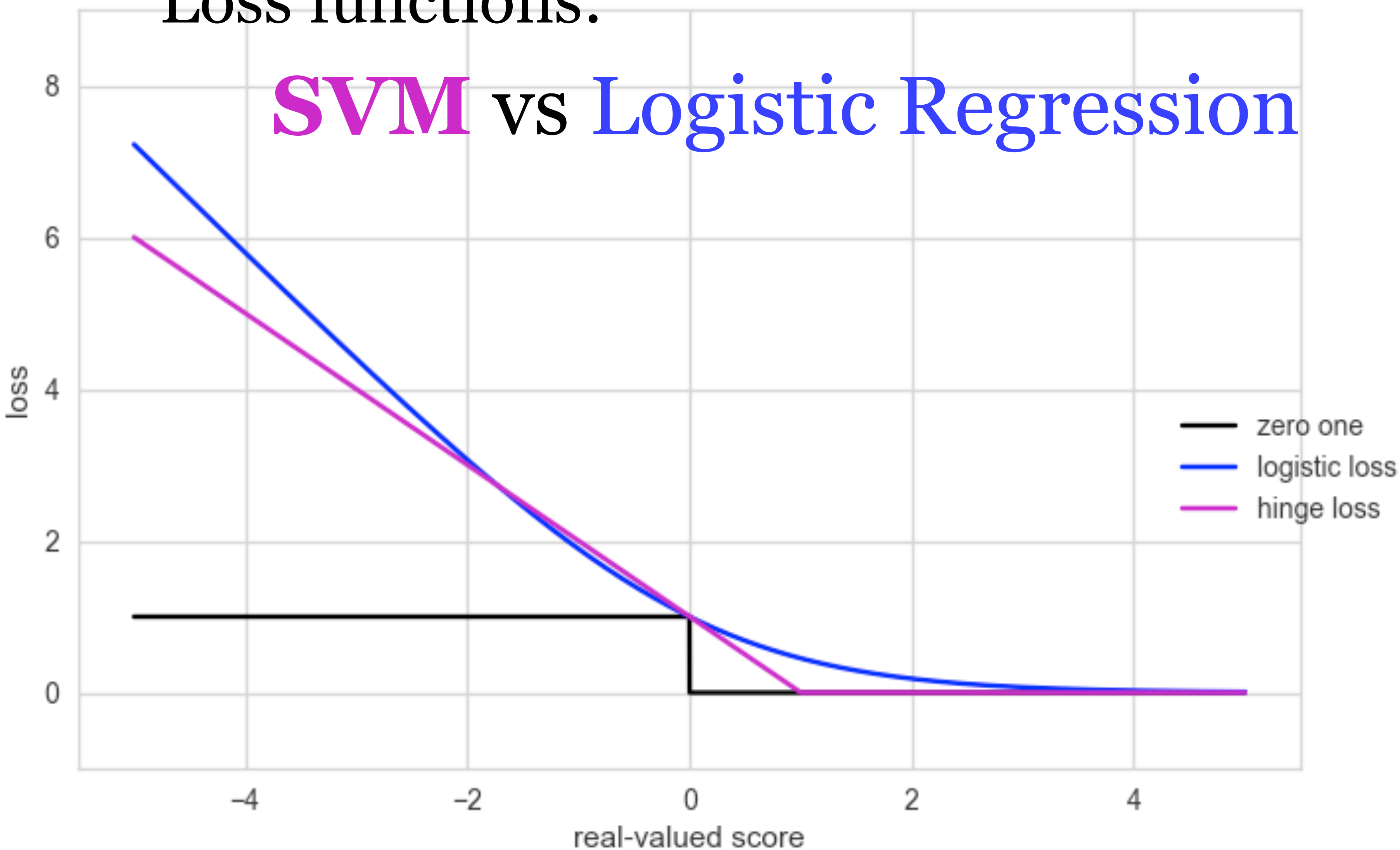
SVM vs Logistic Regression: Compare training objectives

$$\min_{w,b} \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \text{hinge\_loss}(y_n, w^T x_n + b)$$

$$\min_{w,b} \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \text{log\_loss}(y_n, \sigma(w^T x_n + b))$$

Both require tuning complexity hyperparameter C > 0 to avoid overfitting

Loss functions: SVM vs Logistic Regression



SVMs: Prediction

$$\hat{y}(x_i) = w^T x_i + b$$

Make binary prediction via hard threshold

$$\begin{cases} 1 & \text{if } \hat{y}(x_i) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Does not use any notion of probability. Immediately jumps to a hard binary decision.

	SVM	Logistic Regression
Loss	hinge	cross entropy (log loss)
Sensitive to outliers	**Less**	More sensitive
Probabilistic?	No	**Yes**
Multi-class?	Only via separate model for each class (one-vs-all)	**Easy**, using softmax
Kernelizable? **(cover next class)**	Yes, with speed benefits from **sparsity**	Yes

Lab Activity

- Open Day18 Lab Notebook

- Key idea:
  - What happens to decision boundary of SVM when outliers are added?
  - How does that compare to Logistic Regression?