



SVMs with Slack (Not Linearly Separable)

Rishabh Iyer

University of Texas at Dallas

SVMs with Slack (Remove Linear Separability)



- Allow misclassification
 - Penalize misclassification linearly (just like in the perceptron algorithm)
 - Again, easier to work with than counting misclassifications
 - Objective stays convex
 - Will let us handle data that isn't linearly separable!
 - Idea: Take the constraints into the main objective
 - The objective function then becomes exactly like what we have seen in Perceptron/Linear Regression

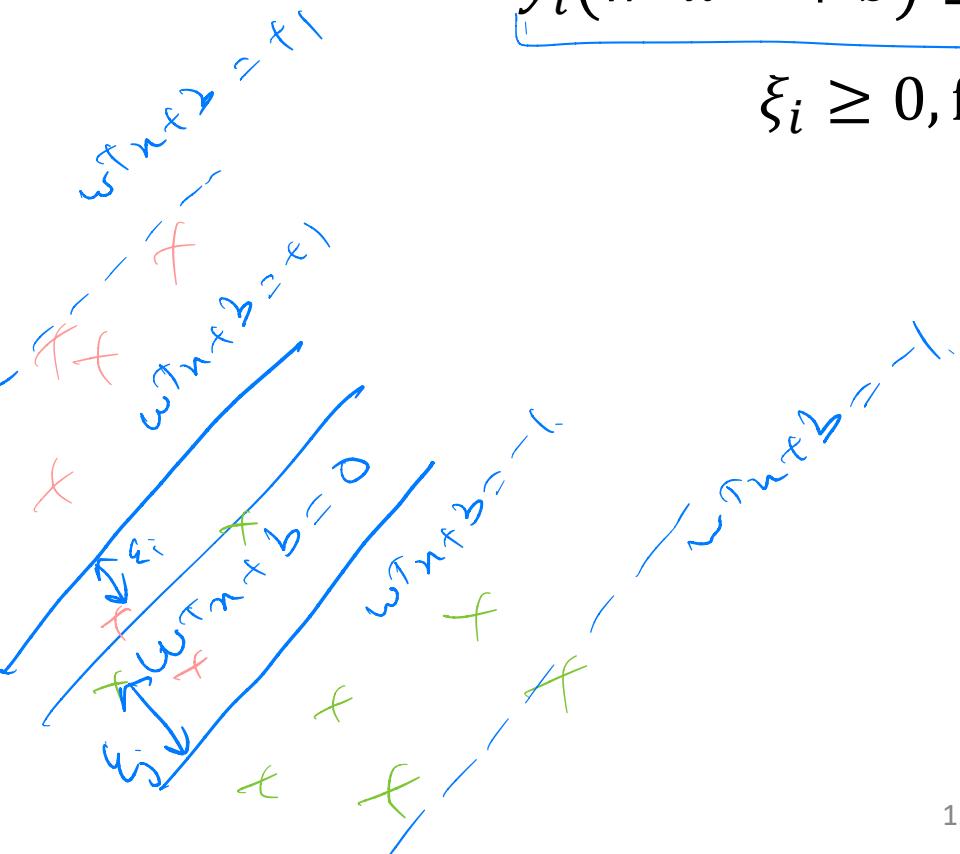
SVMs with Slack

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$$

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$



SVMs with Slack

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$$

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$

- How does this objective change with c ?

SVMs with Slack

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$$

$c \rightarrow \infty$
 $c = 10^{10}$
 $c = -\infty$
 $\xi_i = 10$

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i \text{ for all } i$$

$\xi_i \geq 0 \text{ for all } i$

- How does this objective change with c ?
 - As $c \rightarrow \infty$, requires a perfect classifier
 - As $c \rightarrow 0$, allows arbitrary classifiers (i.e., ignores the data)

SVMs with Slack

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$$

Hyper-parameters

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$

- How should we pick c ?

SVMs with Slack



$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$$

Train / Test

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$

Train → 100%
Test → 30%

- How should we pick c ? [hyperparameters → opt]
- Divide the data into three pieces training, testing, and validation
- Use the validation set to tune the value of the hyperparameter c

Evaluation Methodology

- General learning strategy
 - Build a classifier using the training data
 - Select hyperparameters using validation data →
 - Evaluate the chosen model with the selected hyperparameters on the test data

How can we tell if we overfit the training data?

ML in Practice



- Gather Data + Labels
 - Select feature vectors
 - Randomly split into three groups
 - Training set
 - Validation set
 - Test set
 - Experimentation cycle
 - Select a “good” hypothesis from the hypothesis space
 - Tune hyper-parameters using validation set
 - Compute accuracy on test set (fraction of correctly classified instances)
-

SVMs with Slack

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i (\xi_i) \leftarrow \# \text{ misClass}^{\curvearrowleft}$$

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$

- What is the optimal value of ξ for fixed w and b ?

$$\xi_i = 1 - y_i (w^T x^{(i)} + b) , \text{ if } y_i - \leq 1 \\ \text{else}$$

$$\xi_i = \max(0, 1 - y_i (w^T x^{(i)} + b))$$

SVMs with Slack

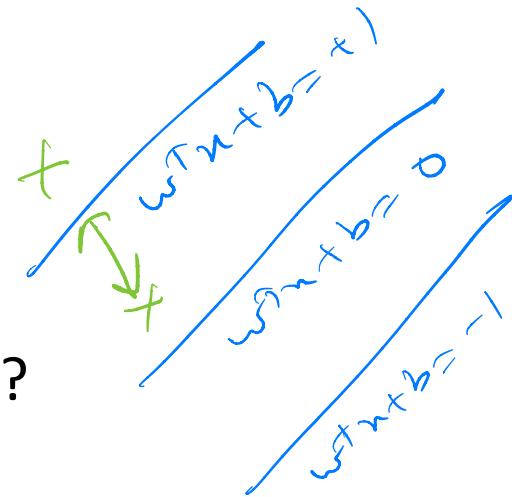
$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$$

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$

- What is the optimal value of ξ for fixed w and b ?
 - If $y_i(w^T x^{(i)} + b) \geq 1$, then $\xi_i = 0$
 - If $y_i(w^T x^{(i)} + b) < 1$, then $\xi_i = 1 - y_i(w^T x^{(i)} + b)$



SVMs with Slack

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + c \sum_i \xi_i$$

such that

$$y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$

- We can formulate this slightly differently

- $\xi_i = \max\{0, 1 - y_i(w^T x^{(i)} + b)\}$

- Does this look familiar?
- Hinge loss provides an upper bound on Hamming loss

Hinge Loss Formulation

- Obtain a new objective by substituting in for ξ

$$\min_{w,b} \frac{1}{2} \|w\|^2 + c \sum_i \max\{0, 1 - y_i(w^T x^{(i)} + b)\}$$

Can minimize with gradient descent!

$$L(w, b) + \frac{\lambda}{2} \|w\|^2$$

\uparrow
 θ

$$\lambda = \frac{1}{c}$$

Hinge Loss Formulation

- Obtain a new objective by substituting in for ξ

$$\min_{w,b} \frac{1}{2} \|w\|^2 + c \sum_i \max\{0, 1 - y_i(w^T x^{(i)} + b)\}$$

Reg
Penalty to prevent
overfitting

Loss
Hinge loss

$$\min_{w,b}$$

$$\sum_{i=1}^M L(f(w, b, x_i), y_i) + \frac{\lambda}{2} \|w\|^2$$

$$\tau = \frac{1}{c}$$

$$\text{Hinge Loss} = C \sum_{i=1}^m \max(0, 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)) + \frac{1}{2} \|\mathbf{w}\|^2$$

C is Large; focus on sep/loss
(\rightarrow train error)

\rightarrow not gen.

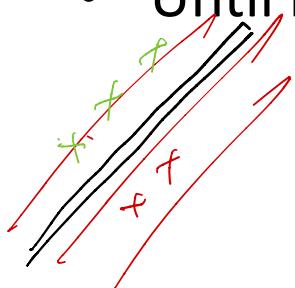
C is small/reasonable gen-

$C \rightarrow 0$ / Arb classifier
Do not care about
loss | Data

REGULARIZATION!!!!



- Until now, we have seen the following optimization problems:



$$\min_{w,b} \sum_i L(f(x^{(i)}, w, b), y_i)$$



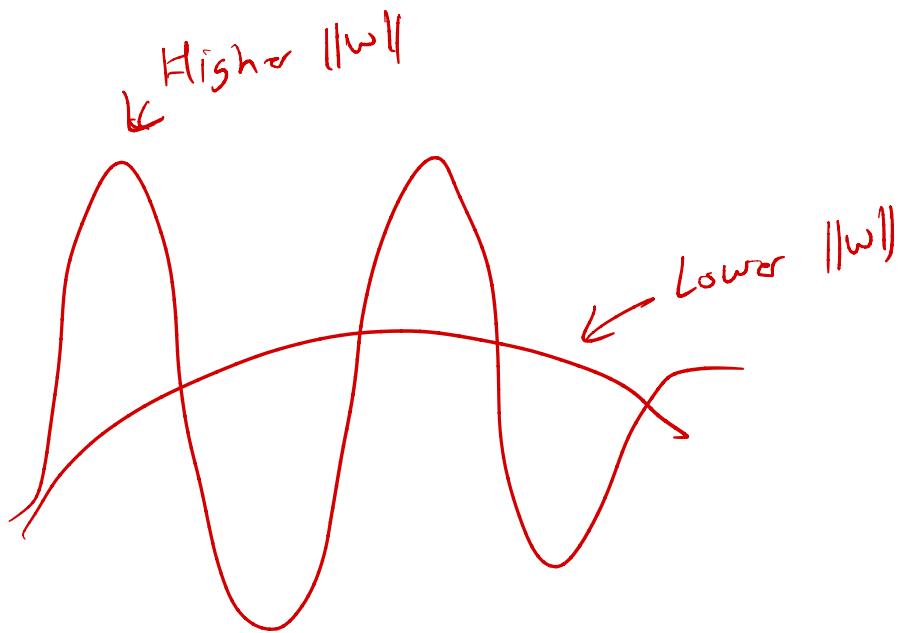
- In the case of Linear regression, L was the squared loss
- In Perceptron, L was Perceptron Loss
- The regularized version of this is:

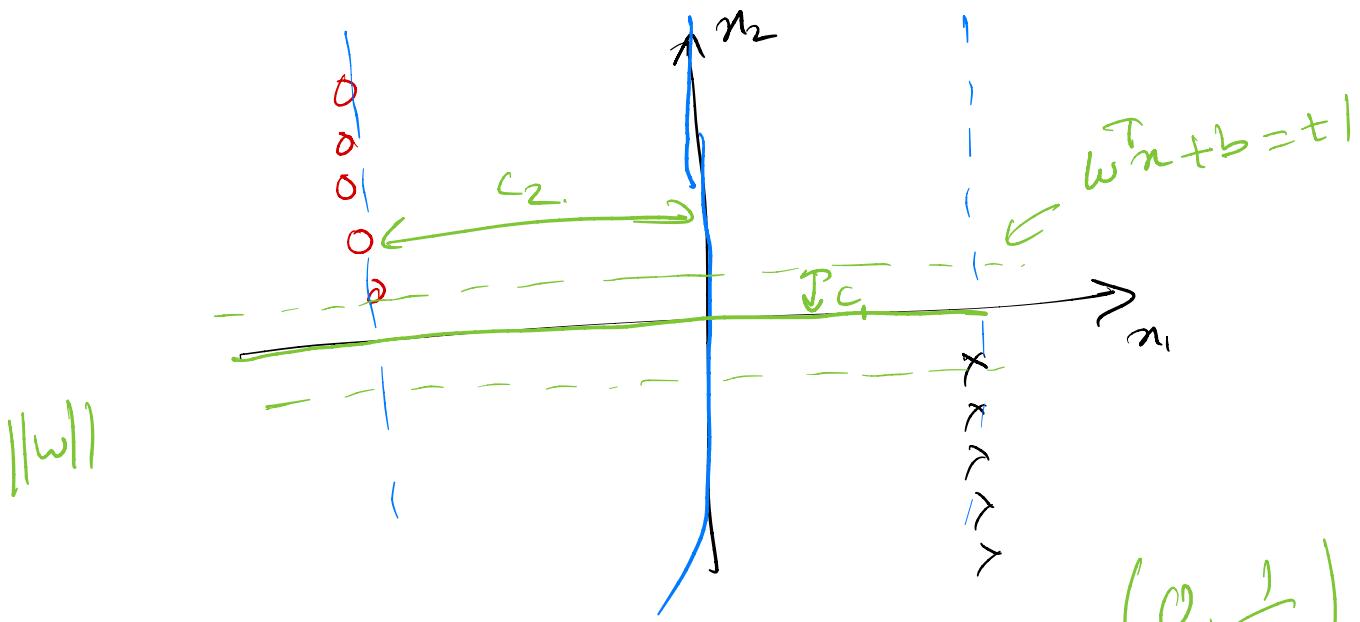
$$\min_{w,b} \frac{1}{2} \cancel{\|w\|^2} + c \sum_i L(f(x^{(i)}, w, b), y_i)$$

- c is a hyper-parameter (again, to be tuned on validation set)

$$\min_{w,b} \sum_{i=1}^m L(f(x^{(i)}, w, b), y^{(i)}) + \frac{c}{2} \|w\|^2$$

$$\min_{\omega, b} \sum_{i=1}^M [(y_i - (\omega^T x_i + b)]^2 + \lambda \|\omega\|^2$$

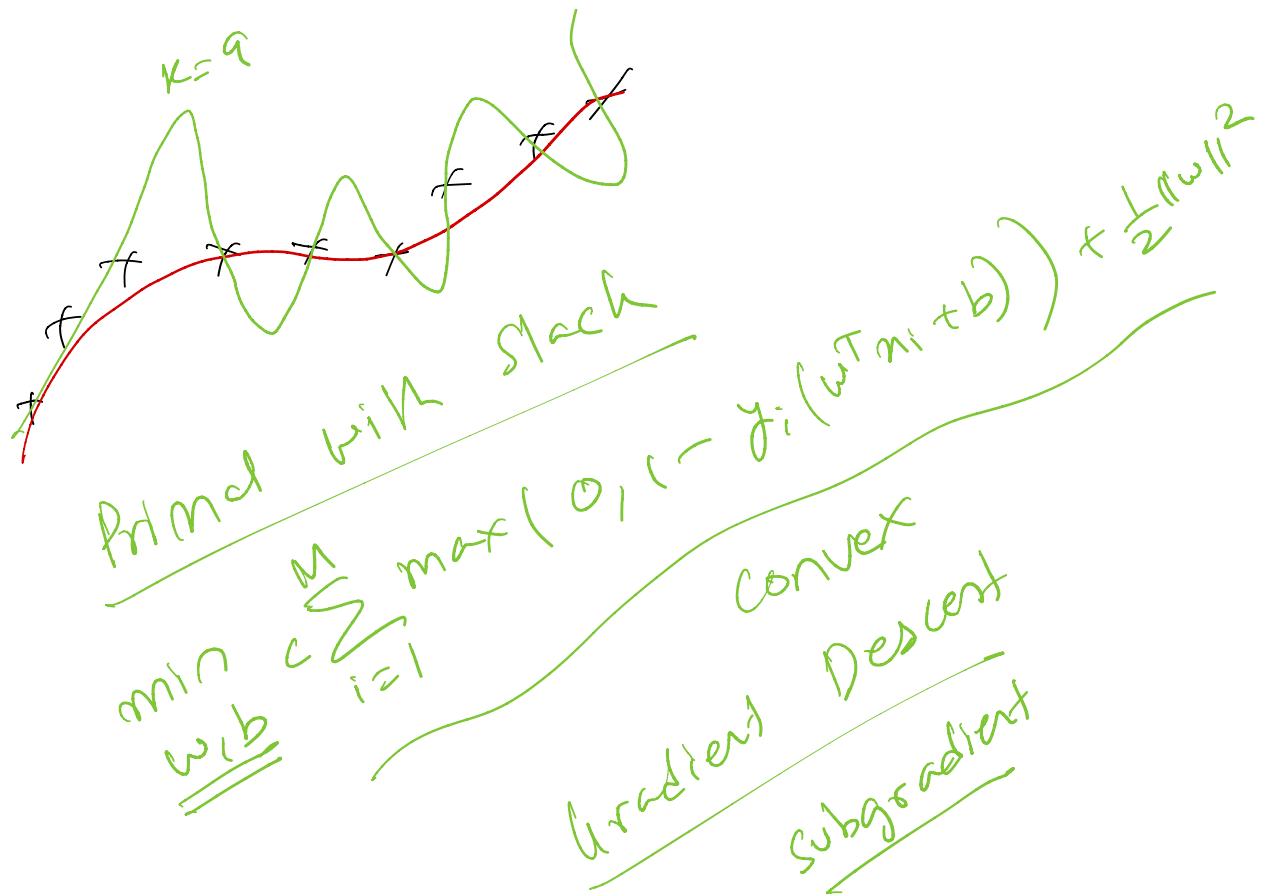




$$\min \|w\|^2$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1, \forall i$$

$$\left(\frac{1}{c_1}, 0 \right)$$



Perceptron vs Hinge vs Square vs Zero-One Loss



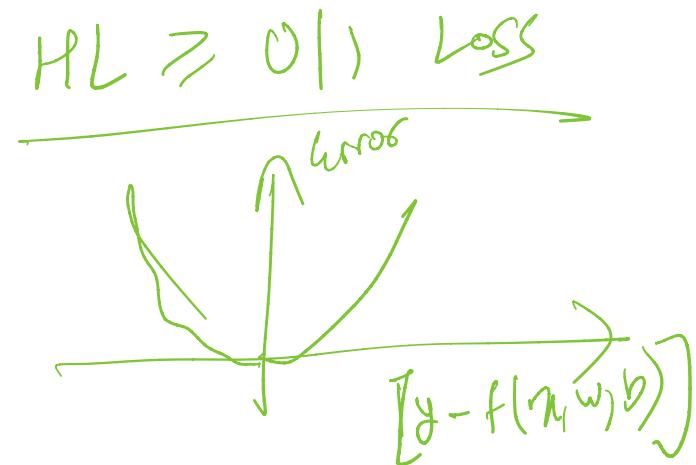
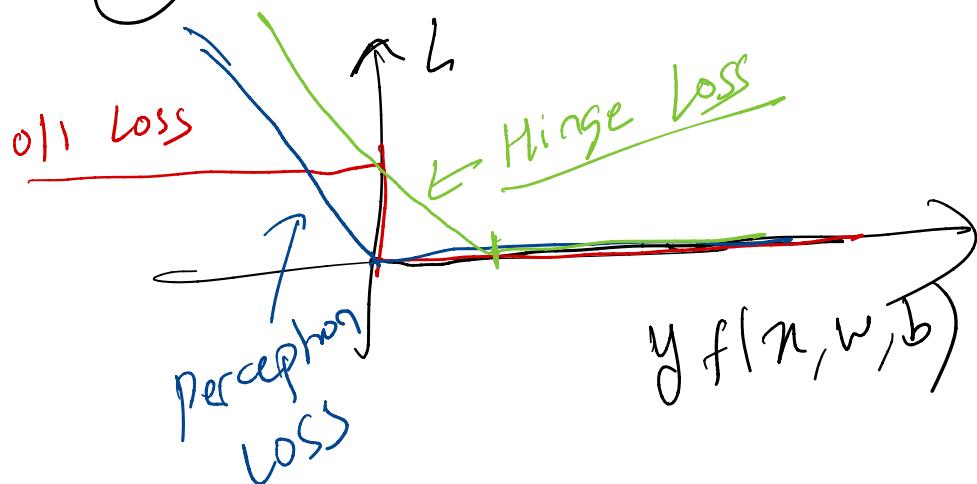
$$\textcircled{1} \quad L(f(x, w, b), y) = \begin{cases} 1 & \text{sign}(f(x, w, b)) \neq y \\ 0 & \text{otherwise} \end{cases}$$

$f(w, b, x) = w^T x + b$

$\xrightarrow{\text{0/1 loss}} = \frac{1}{2} |\text{sign}(f(x, w, b)) - y|$

$$\textcircled{2} \quad L(f(x, w, b), y) = \max(0, -y f(x, w, b))$$

$$\textcircled{3} \quad L(f(x, w, b), y) = \max(0, 1 - y f(x, w, b))$$



Imbalanced Data

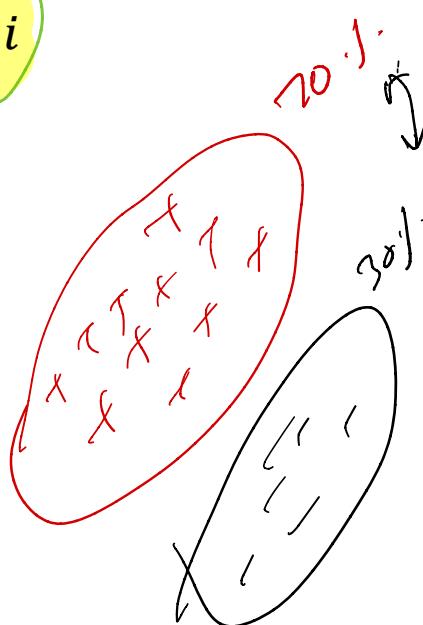
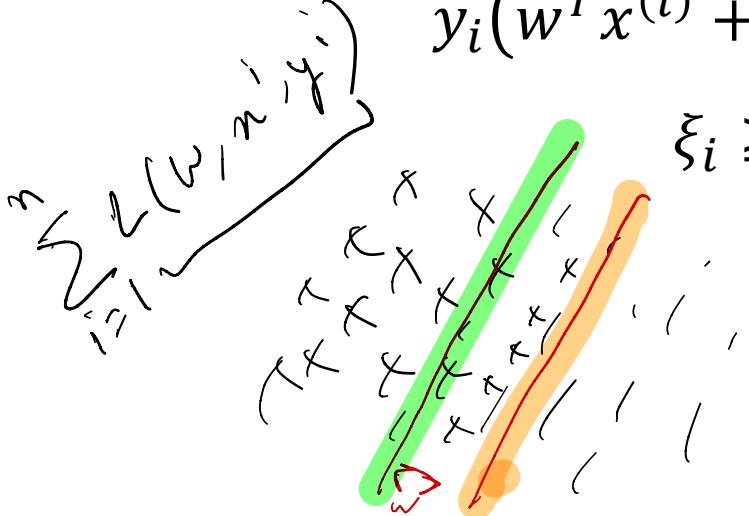


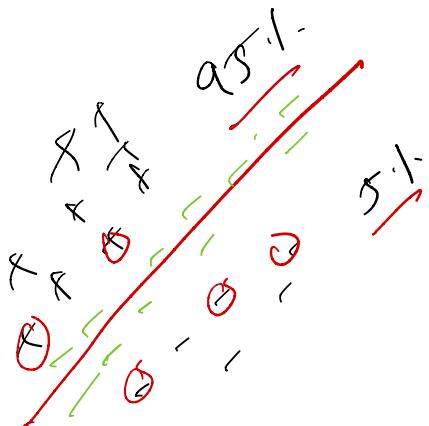
- If the data is imbalanced (i.e., more positive examples than negative examples), may want to evenly distribute the error between the two classes

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + \frac{c}{N_+} \sum_{i:y_i=1} \xi_i + \frac{c}{N_-} \sum_{i:y_i=-1} \xi_i$$

$$\text{such that } y_i(w^T x^{(i)} + b) \geq 1 - \xi_i, \text{ for all } i$$

$$\xi_i \geq 0, \text{ for all } i$$





SVM

$$\min \frac{1}{2} \|w\|^2$$

$$\text{s.t. } y_i(w^T x_i + b) \geq 1$$

$$x \sum_{i:y_i=1} \ln(f^{(m)}, w, b)$$

$$x \sum_{i:y_i=-1} \ln(f^{(m)}, w, b)$$

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i:y_i \neq 0}$$



Generalization



- We argued, intuitively, that SVMs generalize better than the perceptron algorithm → Margin Generalization through Regularization
 - How can we make this precise?

