SVMs with Slack
(Not Linearly Separable)

Rishabh Iyer

University of Texas at Dallas

Slides adapted from Nick Rouzzi

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

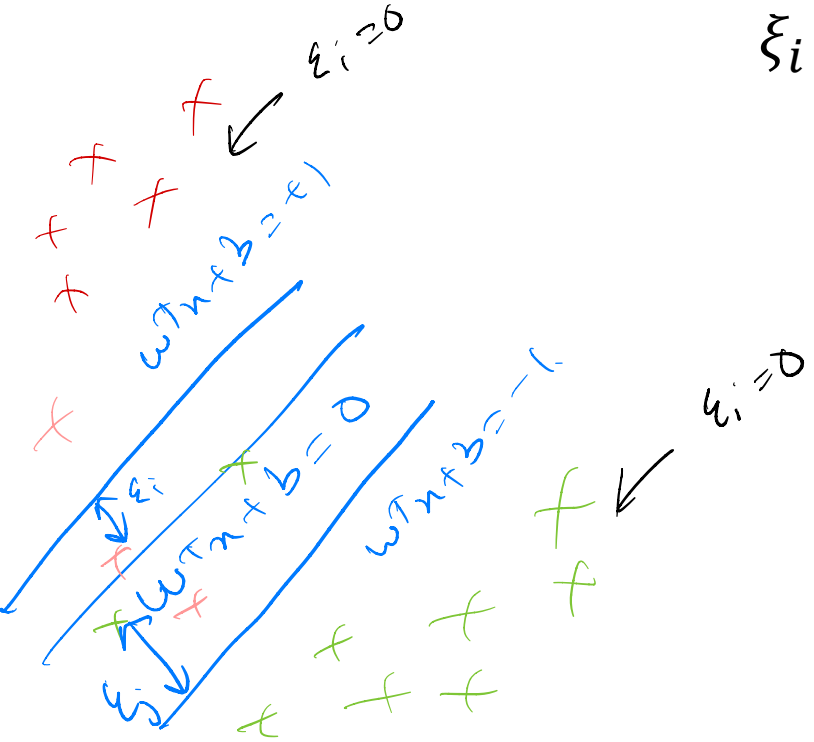
SVMs with Slack

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

such that

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

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



SVMs with Slack

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

such that

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

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

- How does this objective change with c?

SVMs with Slack

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

such that

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

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

- How does this objective change with c?
  - As $c \to \infty$, requires a perfect classifier
  - As $c \to 0$, 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-parameter

such that

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

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

- How should we pick c?

SVMs with Slack

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

such that

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

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

- How should we pick c?
  - 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?

1. Train set.
   - Optimize Model params (w, b)

2. Val set
   - Select Hyper-params (C)
   - Compare multiple trained models for diff C on Val set

3. Test set
   - Evaluate best Hyper-params & model params.

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

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

such that

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

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

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

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

$= 0$ if $y_i(w^T x^{(i)} + b) \geq 1$

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

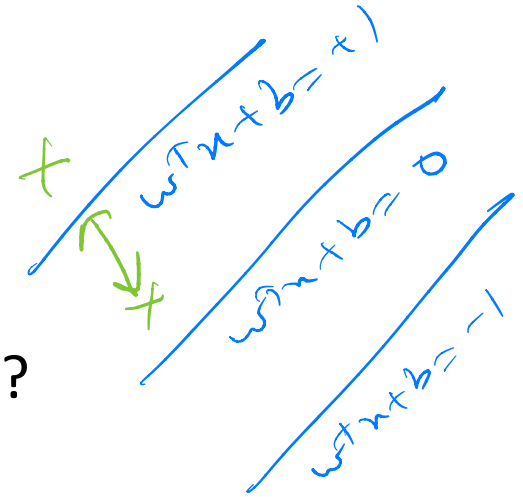
SVMs with Slack

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

such that

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

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



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

SVMs with Slack

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

such that

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

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

- We can formulate this slightly differently
  - $\xi_i = \max\{0, 1 - y_i\left(w^T x^{(i)} + b\right)\}$
  - 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

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

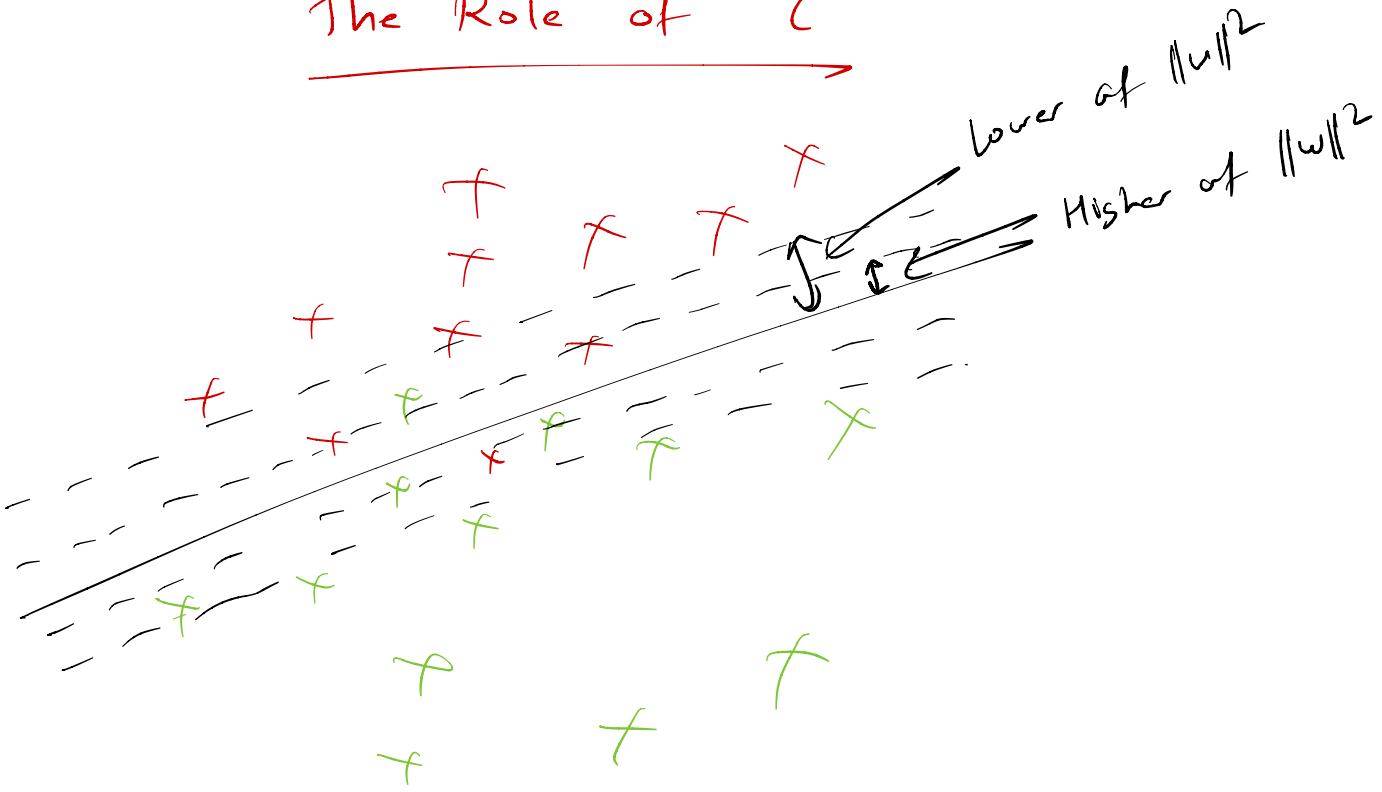
Can minimize with gradient descent!

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

θ

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

The Role of "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

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

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

Hinge Loss: $C \sum_{i=1}^{m} \max(0, 1 - y_i(w^T x_i + b)) + \frac{1}{2}\|w\|^2$

C is Large: Focus on sep / loss
(0 train error)

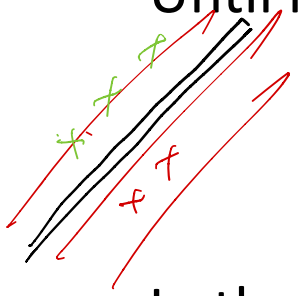
→ not gen.

C is small / reasonable gen.

$C \to 0$, Arb Classifier

Do not care about loss / Data

REGULARIZATION!!!!!



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

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

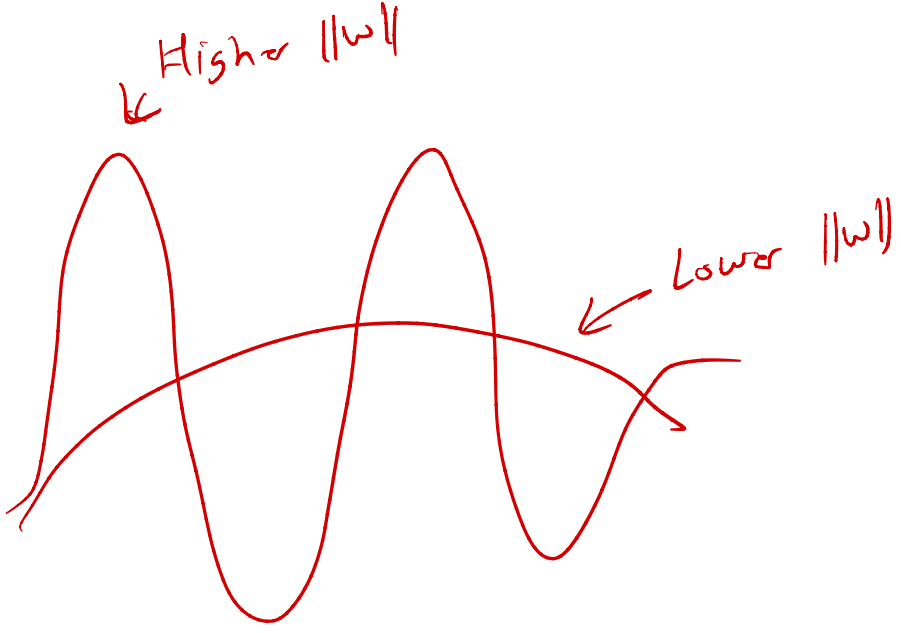
- In the case of Linear regression, L was the squared loss
- In Perceptron, L was Perceptron Loss
- The regularized version of this is:

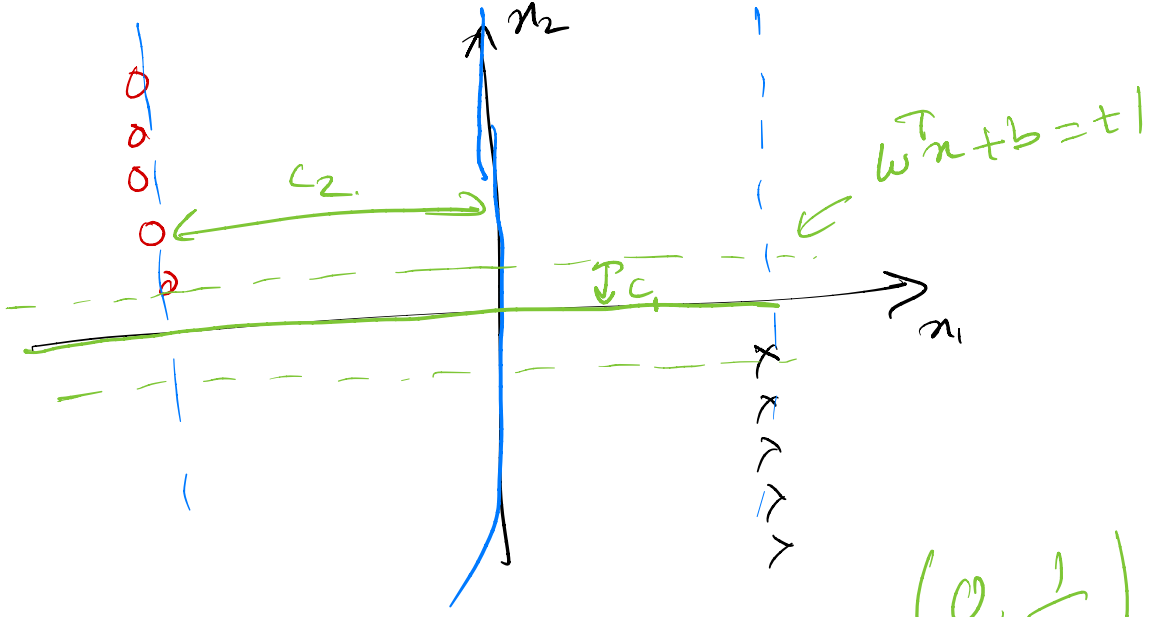
$$\min_{w,b} \frac{1}{2}\|w\|^2 + c\sum_i L\left(f\left(x^{(i)}, w, b\right), y_i\right)$$

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

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

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





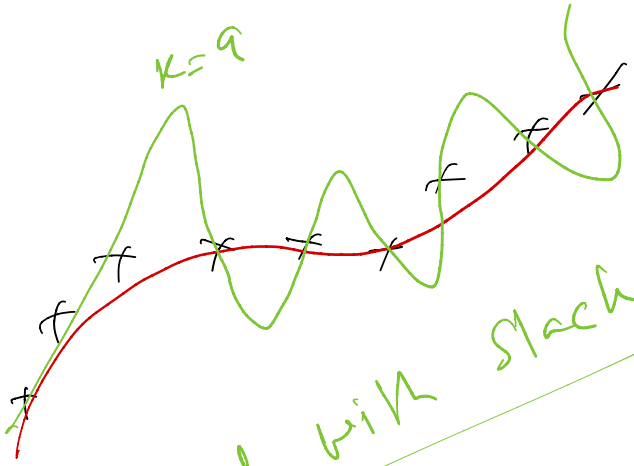
$\|w\|$

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

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

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

$\left(\frac{1}{c_2}, 0\right)$



Primal with Slack

$$\min_{w,b} C \sum_{i=1}^{M} \max(0, 1 - y_i(w^T x_i + b)) + \frac{1}{2}\|w\|^2$$

Convex

Gradient Descent

Subgradient

Perceptron vs Hinge vs Square vs Zero-One Loss

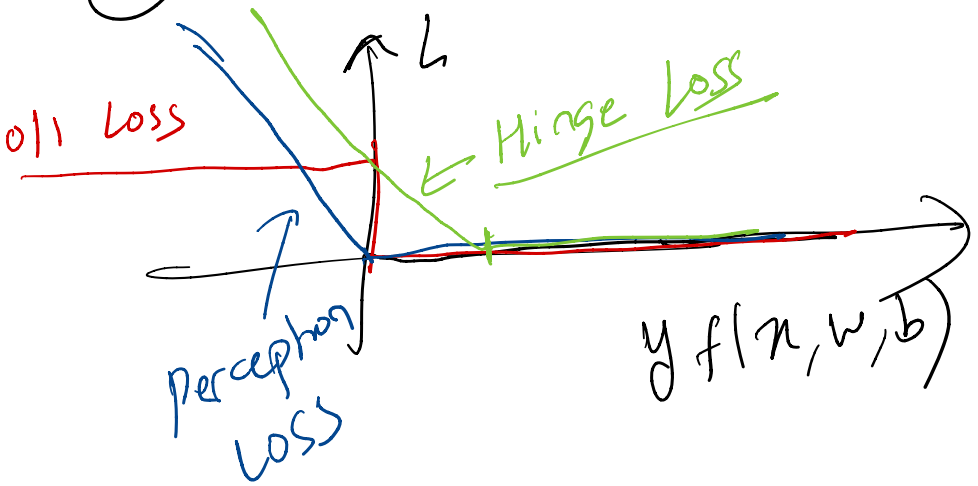
① $L(f(x,w,b),y) = \mathbb{1}_{sign(f(x,w,b)) \neq y}$

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

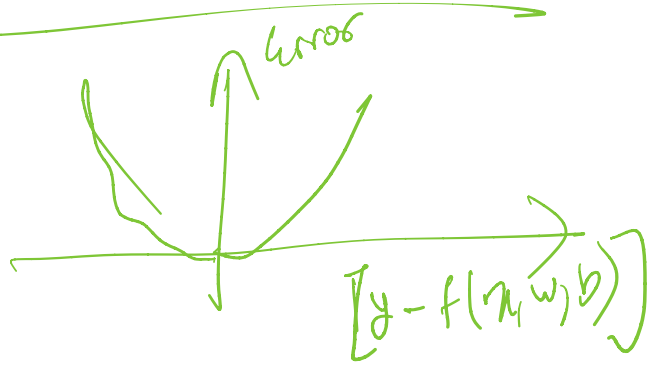
0/1 Loss → $= \frac{1}{2} |sign(f(x,w,b) - y|$

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

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



HL ≥ 0/1 Loss



Imbalanced Data

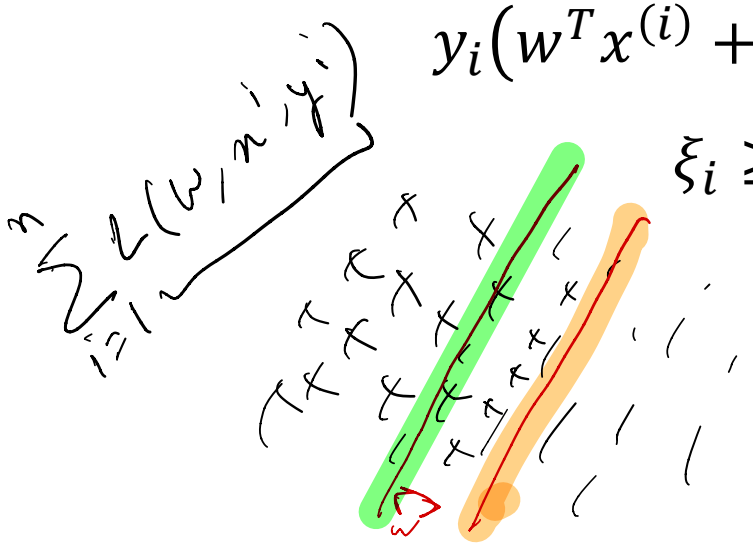
- If the data is imbalanced (i.e., more positive examples than negative examples), may want to evenly distribute the error between the two classes

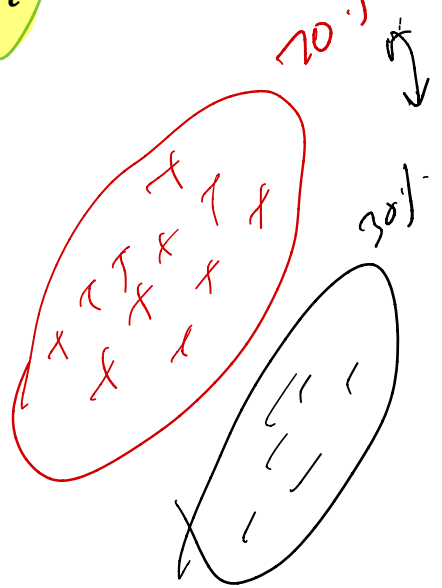
$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + \frac{C}{N_+}\sum_{i:y_i=1}\xi_i + \frac{C}{N_-}\sum_{i:y_i=-1}\xi_i$$

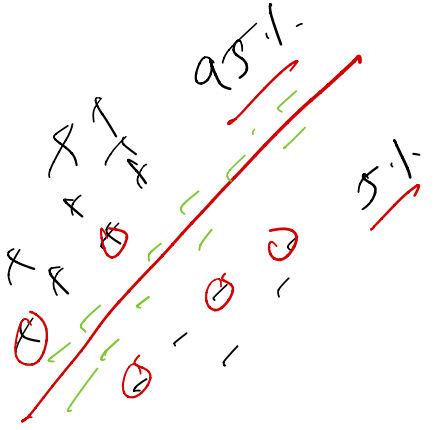
such that

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

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







SVM

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

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

$$\min_{w,b} \frac{1}{2}\|w\|^2 + \frac{C}{N_+}\sum_{i:y_i=1} L_H\left(f(x^{(i)}, w, b), 1\right) + \frac{C}{N_-}\sum_{i:y_i=-1} L_H\left(f(x^{(i)}, w, b), -1\right)$$

Generalization

- We argued, intuitively, that SVMs generalize better than the perceptron algorithm → margin generalization through Regularizn
  - How can we make this precise?

[ML Theory]