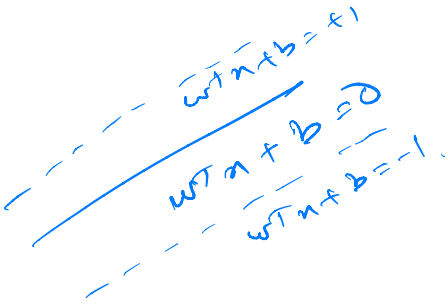
SVMs with Slack & kernels
(Not Linearly Separable)

Rishabh Iyer

University of Texas at Dallas

Slides adapted from Nick Rouzzi

Recap of SVM



$$\min_{w,b} \|w\|^2$$

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

$\Rightarrow \underbrace{1 - y_i(w^T x_i + b)}_{} \leq 0$ ← constant

Lagrangian: $L(w, b, \lambda) = \|w\|^2 + \sum_{i=1}^{M} \lambda_i (1 - y_i(w^T x_i + b))$

$$g(\lambda) = \min_{w,b} L(w, b, \lambda)$$

Dual: $\max_{\lambda : \lambda \geq 0} g(\lambda)$

Dual SVM

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j x^{(i)T} x^{(j)} + \sum_i \lambda_i$$

such that

$$\lambda \geq 0 \quad -(1)$$

$$\sum_i \lambda_i y_i = 0 \quad -(2)$$

- The dual formulation only depends on inner products between the data points
  - Same thing is true if we use feature vectors instead

Primal

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

subject to $y_i(w^T x^{(i)} - b) \geq 1$

Equivalently

Dual

$$\max_{\lambda} \ -\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M} \lambda_i \lambda_j y_i y_j x^{(i)T} x^{(j)} + \sum_i \lambda_i$$

s.t $\lambda \geq 0$

$\sum_i \lambda_i y_i = 0$

} constraints

$w = \sum_{i=1}^{M} \lambda_i y_i x^{(i)}$

b ← complementary slackness

Dual SVM

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j \Phi\left(x^{(i)}\right)^T \Phi\left(x^{(j)}\right) + \sum_i \lambda_i$$

such that

$$\sum_i \lambda_i y_i = 0$$

Primal:

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

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

- The dual formulation only depends on inner products between the data points
  - Same thing is true if we use feature vectors instead

$$\phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix} \qquad \phi(x) = \begin{bmatrix} \tilde{x} \\ x \\ 1 \end{bmatrix}$$

The Kernel Trick

- For some feature vectors, we can compute the inner products quickly, even if the feature vectors are very large
- This is best illustrated by example
  - Let $\phi(x_1, x_2) = \begin{bmatrix} x_1 x_2 \\ x_2 x_1 \\ x_1^2 \\ x_2^2 \end{bmatrix}$
  - $\phi(x_1, x_2)^T \phi(z_1, z_2) = x_1^2 z_1^2 + 2x_1 x_2 z_1 z_2 + x_2^2 z_2^2$

$$= (x_1 z_1 + x_2 z_2)^2$$

$$= \left(x^T z\right)^2$$

$x = [x_1, x_2 \cdots x_n] \in \mathbb{R}^n$

$\phi(x) = [\underbrace{x_1, x_2, \cdots x_n}_{n}, \underbrace{x_1 x_2, \cdots}_{\binom{n}{2}}, \underbrace{x_1^2, \cdots x_n^2}_{n}]$

$2n + \binom{n}{2} \simeq n^2$

$\phi(x_i)^T \phi(x_j)$

$O(n^2) \rightarrow O(n)$

$\phi(x_i)^T \phi(x_j) = x_i^T x_j + (x_i^T x_j)^2$

The Kernel Trick

- The same idea can be applied for the feature vector ϕ of all polynomials of degree (exactly) d
  - $\phi(x)^T \phi(z) = (x^T z)^d$
- More generally, a kernel is a function $k(x, z) = \phi(x)^T \phi(z)$ for some feature map ϕ
- Rewrite the dual objective

$$\max_{\lambda \geq 0, \sum_i \lambda_i y_i = 0} -\frac{1}{2} \sum_i \sum_j \lambda_i \lambda_j y_i y_j k(x^{(i)}, x^{(j)}) + \sum_i \lambda_i$$

↑ Kernel SVM

Examples of Kernels

- Polynomial kernel of degree exactly d
  - $k(x, z) = (x^T z)^d$
- General polynomial kernel of degree d for some c
  - $k(x, z) = (x^T z + c)^d$
- Gaussian kernel for some σ
  - $k(x, z) = \exp\left(\frac{-\|x-z\|^2}{2\sigma^2}\right)$
  - The corresponding ϕ is infinite dimensional!
- So many more…

Gaussian Kernels

- Consider the Gaussian kernel

$$\exp\left(\frac{-\|x-z\|^2}{2\sigma^2}\right) = \exp\left(\frac{-(x-z)^T(x-z)}{2\sigma^2}\right)$$

$$= \exp\left(\frac{-\|x\|^2 + 2x^Tz - \|z\|^2}{2\sigma^2}\right)$$

$$= \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)\exp\left(-\frac{\|z\|^2}{2\sigma^2}\right)\exp\left(\frac{x^Tz}{\sigma^2}\right)$$

- Use the Taylor expansion for exp()

$$\exp\left(\frac{x^Tz}{\sigma^2}\right) = \sum_{n=0}^{\infty}\frac{(x^Tz)^n}{\sigma^{2n}n!}$$

Kernels

- Bigger feature space increases the possibility of overfitting
  - Large margin solutions may still generalize reasonably well
- Alternative: add "penalties" to the objective to disincentivize complicated solutions

Kernel Properties

① If K_1 & K_2 are kernels, is $K = K_1 + K_2$ also a kernel?

$K_1(x,y) = \phi_1(x)^T \phi_1(y)$

$K_2(x,y) = \phi_2(x)^T \phi_2(y)$

$K(x,y) = \phi(x)^T \phi(y)$

$\phi_3(x) = [\phi_1(x)\ \phi_2(x)]$

$\phi_3(y) = [\phi_1(y)\ \phi_2(y)]$

② If K is a kernel, is $K' = \alpha K$, for $\alpha \geq 0$ also a kernel?

$$\phi_3(x)^T \phi_3(y) = \begin{bmatrix} \phi_1(x)^T & \phi_2(x)^T \end{bmatrix} \begin{bmatrix} \phi_1(y) \\ \phi_2(y) \end{bmatrix} = \phi_1(x)^T \phi_1(y) + \phi_2(x)^T \phi_2(y) = K_1 + K_2$$

Kernel SVM

Dual: $\max_{\lambda} \; -\frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \, k(x^{(i)}, x^{(j)}) + \sum_i \lambda_i$

$$\text{s.t} \quad \sum_i \lambda_i y_i = 0 \;, \quad \lambda \geq 0$$

Prediction problem?

$$y_t = \text{sign}\left(\sum_i \lambda_i y_i \, k(x^{(i)}, x_t) + b \right)$$

where $b = y_i - \sum_j \lambda_j y_j \, k(x^{(i)}, x^{(j)})$ for $i : \lambda_i \geq 0$

w, b

$$y_t = \text{sign}(w^T \phi(x_t) + b)$$

$$w = \sum_{i=1}^{m} \lambda_i y_i \phi(x^{(i)})$$

$$y_t = \text{sign}\left(\sum_{i=1}^{m} \lambda_i y_i \underbrace{\phi(x^{(i)})\, \phi(x_t)}_{K(x^{(i)}, x_t)} + b \right)$$

$$= \text{sign}\left(\sum_{i=1}^{m} \lambda_i y_i K(x^{(i)}, x_t) + b \right)$$

Comp. slackness: $b = y_i - w \cdot x_i$, for any $i : \lambda_i \geq 0$

$$= y_i - \sum_{j=1}^{m} \lambda_j y_j \underbrace{\phi(x^{(j)})\, \phi(x^{(i)})}_{K(x^{(j)}, x^{(i)})}$$

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

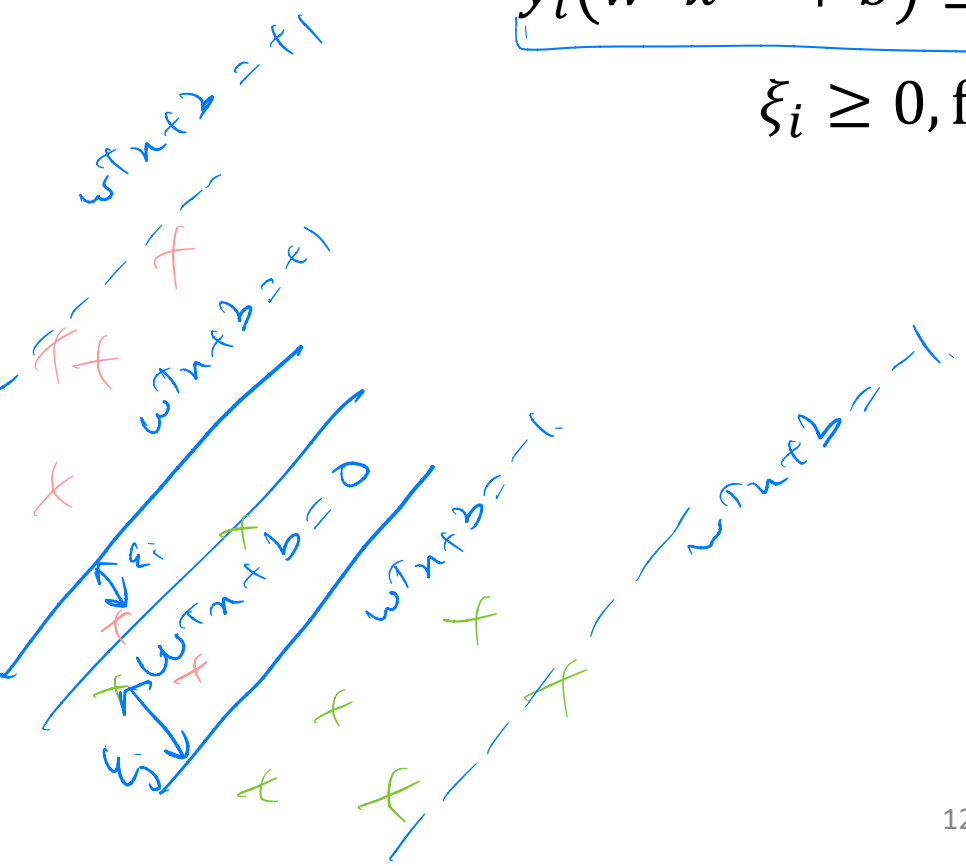
SVMs with Slack

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

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?

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 $y_i \ldots \leq 1$

else

$= 0$

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

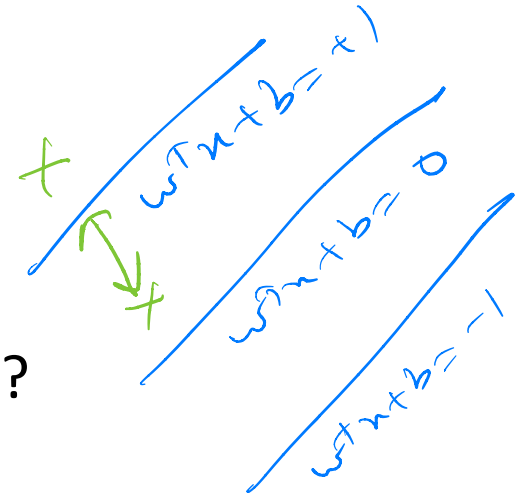
SVMs with Slack

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

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!

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)

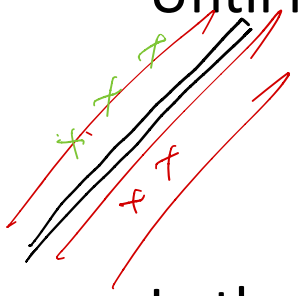
→ not gen.

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:

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

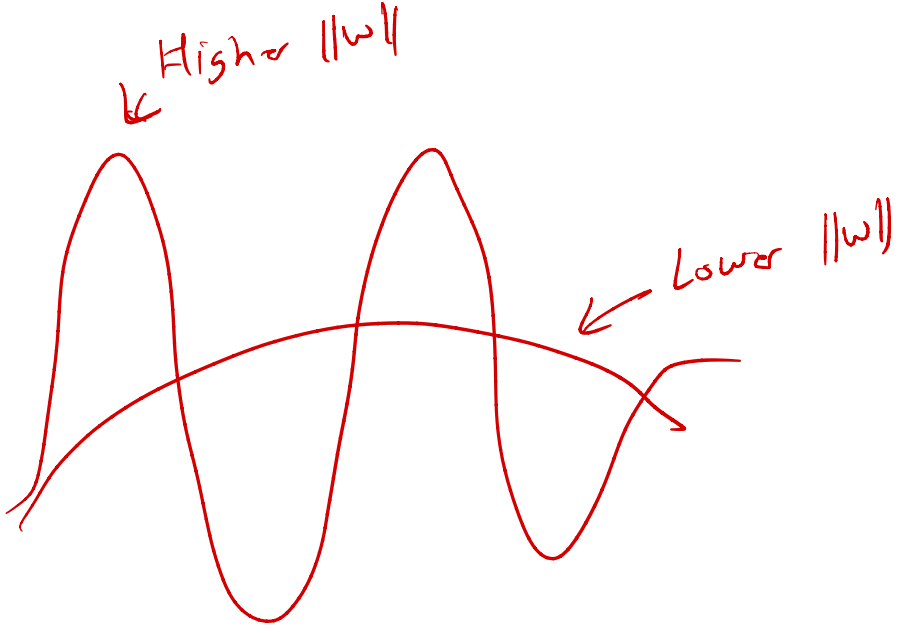
- In the case of Linear regression, L was the squared loss
- In Perceptron, L was Perceptron Loss
- The regularized version of this is:

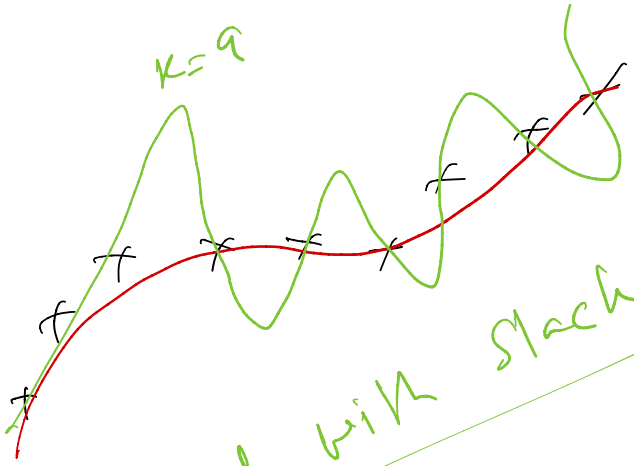
$$\min_{w,b} \frac{1}{2}\|w\|^2 + c\sum_i L\left(f\left(x^{(i)}, w, b\right), y_i\right)$$

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

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

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





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

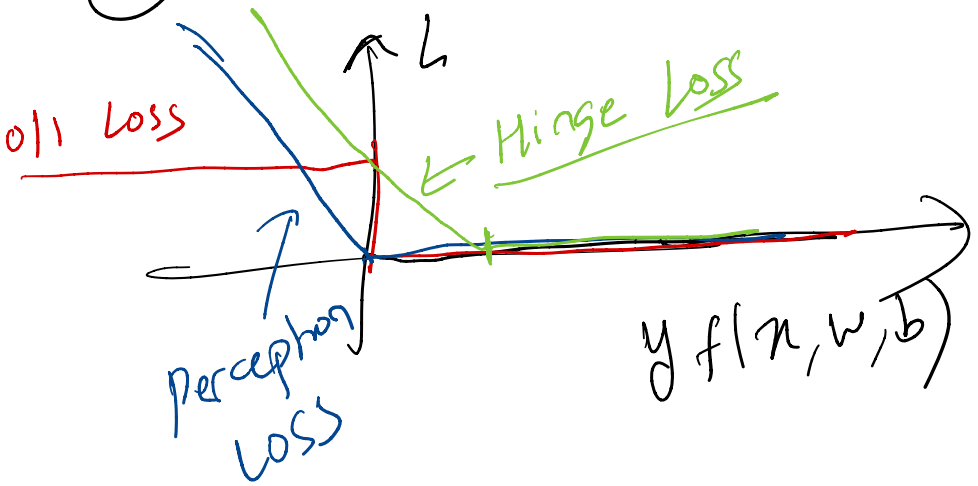
① $L(f(x,w,b),y) = \mathbb{1}_{sign(f(x,w,b)) \neq y}$

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

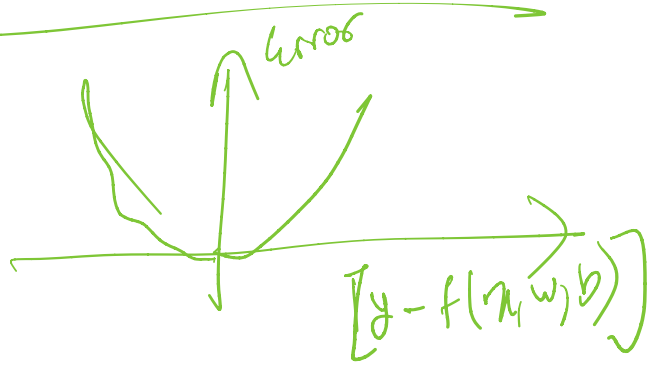
0/1 Loss → $= \frac{1}{2} |sign(f(x,w,b) - y|$

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

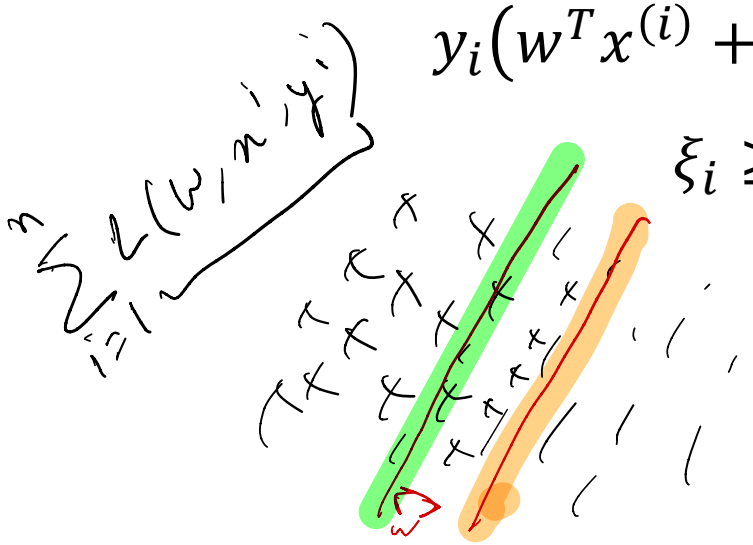
- If the data is imbalanced (i.e., more positive examples than negative examples), may want to evenly distribute the error between the two classes

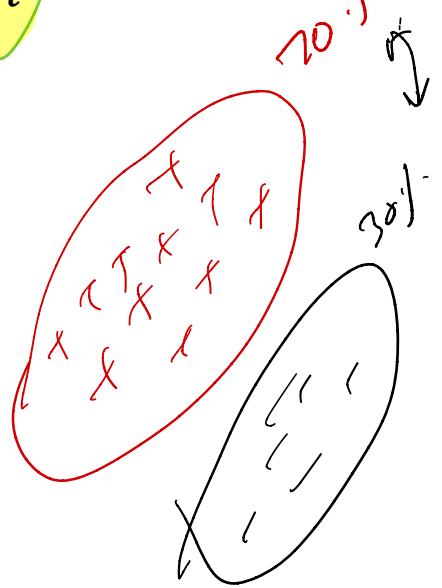
$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + \frac{C}{N_+}\sum_{i:y_i=1}\xi_i + \frac{C}{N_-}\sum_{i:y_i=-1}\xi_i$$

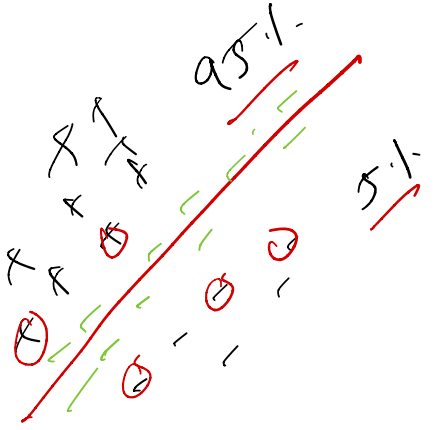
such that

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

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







SVM

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

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

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

Dual of Slack Formulation

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

such that

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

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

Dual of Slack Formulation

$$L(w,b,\xi,\lambda,\mu) = \frac{1}{2}w^T w + c\sum_i \xi_i + \sum_i \lambda_i(1-\xi_i - y_i(w^T x^{(i)} + b)) + \sum_i -\mu_i \xi_i$$

$$\sum_i \xi_i [c - \lambda_i - \mu_i] = 0$$

Convex in w, b, ξ, so take derivatives to form the dual

$w \in \mathbb{R}^n$
$x^{(i)} \in \mathbb{R}^n$

$$\frac{\partial L}{\partial w_k} = w_k + \sum_i -\lambda_i y_i x_k^{(i)} = 0 \quad \Rightarrow \quad w = \sum_i \lambda_i y_i x^{(i)}$$

$$\frac{\partial L}{\partial b} = \sum_i -\lambda_i y_i = 0 \quad \Rightarrow \quad \sum_i \lambda_i y_i = 0$$

$$\frac{\partial L}{\partial \xi_k} = c - \lambda_k - \mu_k = 0 \quad \Rightarrow \quad c = \lambda_k + \mu_k$$

$\lambda_k \geq 0$
$\mu_k \geq 0$
$\lambda_k \leq C$

Dual of Slack Formulation

$$\max_{\lambda \geq 0} -\frac{1}{2}\sum_i \sum_j \lambda_i \lambda_j y_i y_j x^{(i)T} x^{(j)} + \sum_i \lambda_i$$

such that

$$\sum_i \lambda_i y_i = 0$$

$$c \geq \lambda_i \geq 0, \text{for all } i$$

$\frac{1}{2}\|w\|^2 + C\sum_i \max(0, 1 - y_i(w^T x_i + b))$

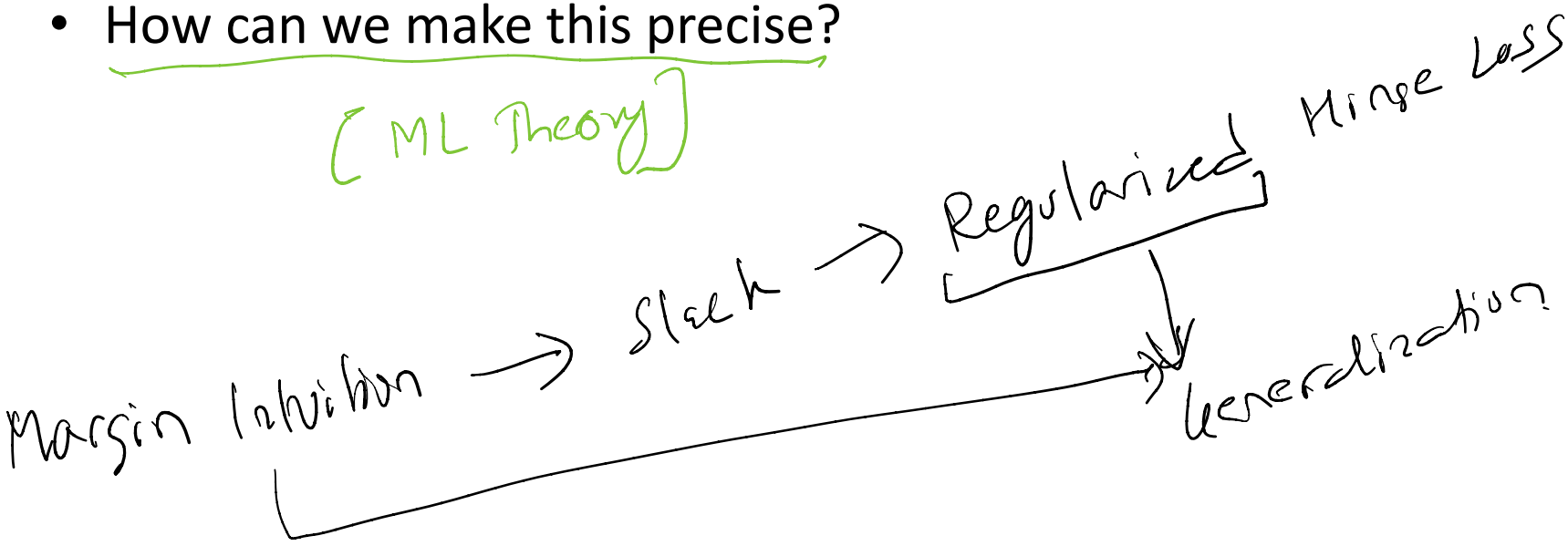
Regularization in Dual

$\lambda_i \leq C, \forall i$

Generalization

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

[ML Theory]



Roadmap

- Where are we headed?
  - Other simple hypothesis spaces for supervised learning
    - k nearest neighbor
    - Decision trees
  - Learning theory
    - Generalization and PAC bounds
    - VC dimension
    - Bias/variance tradeoff

Naive Bayes

Logistic Regression