



CS 4375
SVMs with Slack
(Not Linearly Separable)

Rishabh Iyer
University of Texas at Dallas

- 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

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

such that

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

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

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

such that

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

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

Potentially allows some points to be misclassified/inside the margin

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

such that

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

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

Constant c determines
degree to which slack is
penalized

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

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 ?

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

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

for $\xi > 1$ point is **misclassified**



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

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 should we pick c ?

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

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 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

- 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?

- 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)

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

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 ?

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

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

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

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!

Hinge Loss Formulation



- Obtain a new objective by substituting in for ξ

$$\min_{w,b} \underbrace{\frac{1}{2} \|w\|^2}_{\text{Penalty to prevent overfitting}} + c \underbrace{\sum_i \max\{0, 1 - y_i(w^T x^{(i)} + b)\}}_{\text{Hinge loss}}$$

Penalty to prevent
overfitting

Hinge loss

- 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:

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

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

Perceptron vs Hinge vs Square vs Zero-One Loss



- 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

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

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

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

- Where are we headed?
 - Other simple hypothesis spaces for supervised learning
 - k nearest neighbor
 - Decision trees
 - Probabilistic Methods
 - Bayesian Methods
 - Naïve Bayes
 - Logistic Regression