

# CME 252: Support Vector Machines

AJ Friend  
ICME, Stanford University

# Introduction

# Outline

Introduction

Linearly Separable Problem

Which Separator?

Maximum Margin Classifier

Non-separable Linear Classification

Sparse Violation Classifier

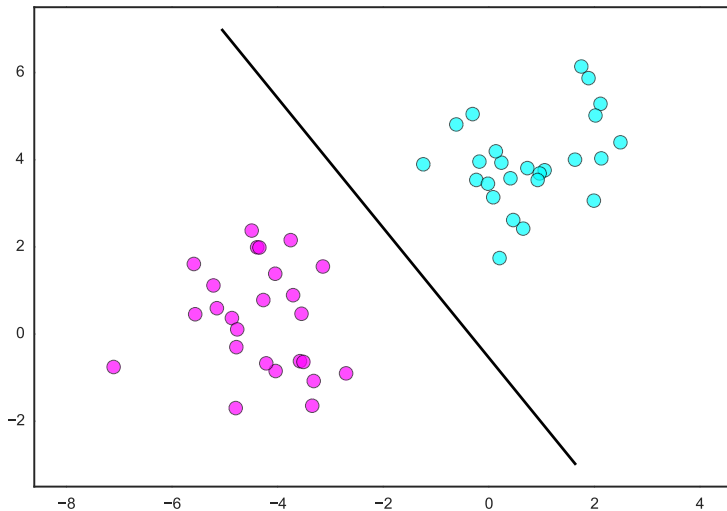
Support Vector Classifier

Loss Functions

# Support Vector Machines

- ▶ many related/overlapping names:
  - ▶ maximum margin classifier
  - ▶ support vector classifier
  - ▶ (robust) linear discrimination/classification
  - ▶ support vector machine
- ▶ I won't always use the right name
- ▶ we'll start with:
  - ▶ find a hyperplane to separate data points into two classes
  - ▶ use hyperplane to classify new (unseen) points

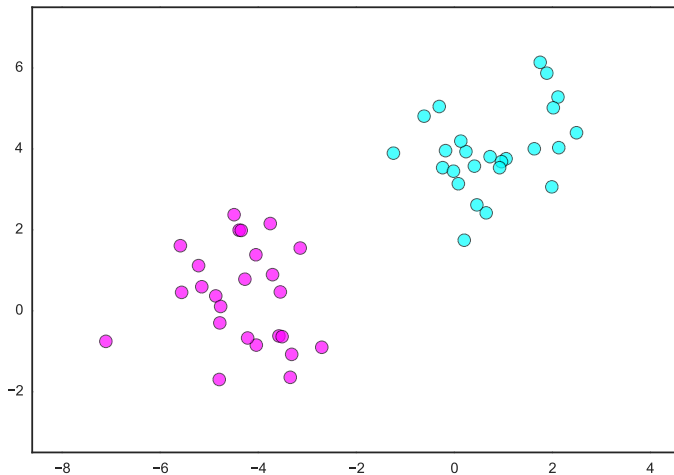
# Support Vector Machines



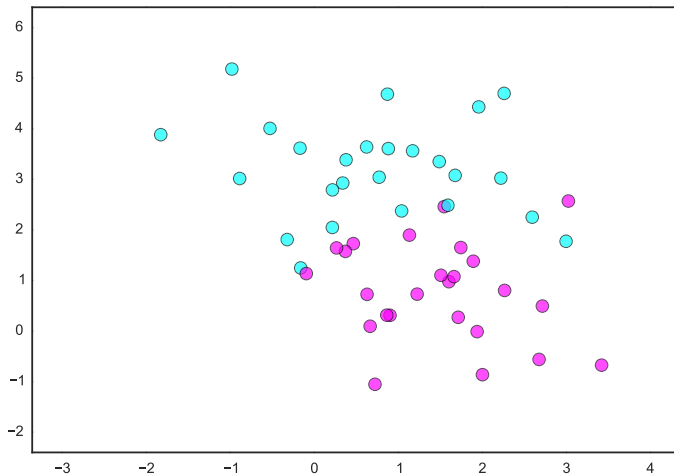
# Scenarios

- ▶ classify data in increasingly sophisticated scenarios:
  - ▶ strictly linearly separable
  - ▶ approximately (not strictly) linearly separable
  - ▶ approximately non-linearly separable (hyperplanes won't work)

# Strictly Linearly Separable Data

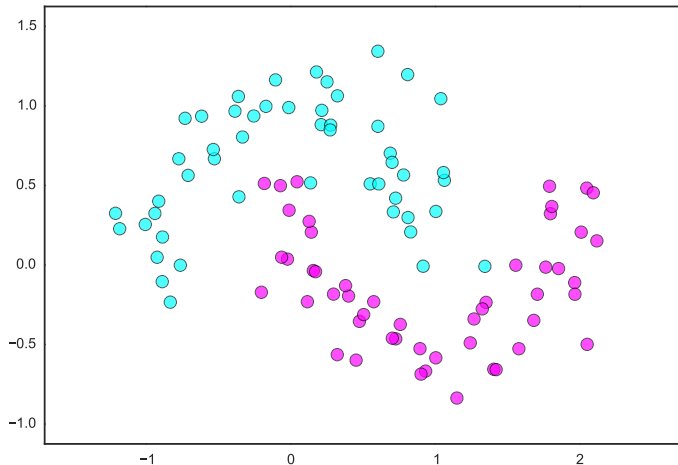


## Approximately Linearly Separable Data





## Approximately Non-linearly Separable



## Linearly Separable Problem

# Outline

Introduction

Linearly Separable Problem

Which Separator?

Maximum Margin Classifier

Non-separable Linear Classification

Sparse Violation Classifier

Support Vector Classifier

Loss Functions

## Linearly Separable Problem

- ▶ data:  $x_i \in \mathbf{R}^n$  with labels  $y_i \in \{+1, -1\}$  for  $i = 1, \dots, N$
- ▶ assume **strictly** linearly separable
- ▶ find hyperplane  $\{x \mid a^T x = b\}$  that separates points by label

$$a^T x_i - b > 0 \text{ if } y_i = +1$$

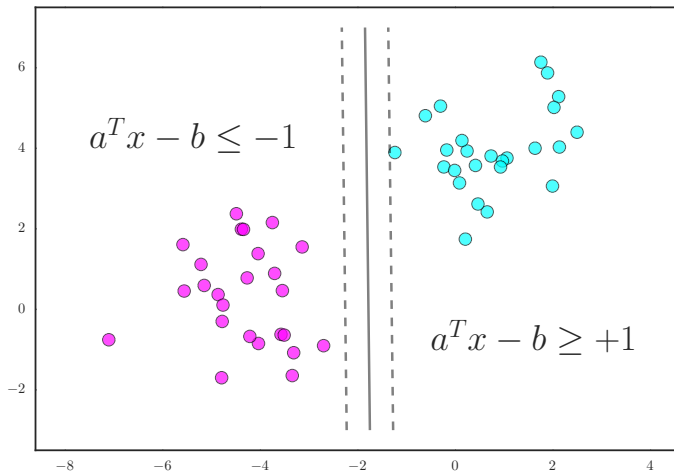
$$a^T x_i - b < 0 \text{ if } y_i = -1$$

- ▶ **rescale**  $a, b$  so that

$$a^T x_i - b \geq +1 \text{ if } y_i = +1$$

$$a^T x_i - b \leq -1 \text{ if } y_i = -1$$

## Linearly Separable Problem



## Linearly Separable Problem

- ▶ for all  $i$ , rewrite constraints as

$$y_i (a^T x_i - b) \geq 1$$

- ▶ get **feasibility** problem

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & y_i (a^T x_i - b) \geq 1 \text{ for } i = 1, \dots, N \end{array}$$

with variables  $a \in \mathbf{R}^n$ ,  $b \in \mathbf{R}$

## CVXPY for Separable Problem

```
a = Variable(n)
b = Variable()

obj = Minimize(0)
constr = [mul_elemwise(y, X*a - b) >= 1]
Problem(obj, constr).solve()
```

Which Separator?



# Outline

Introduction

Linearly Separable Problem

Which Separator?

Maximum Margin Classifier

Non-separable Linear Classification

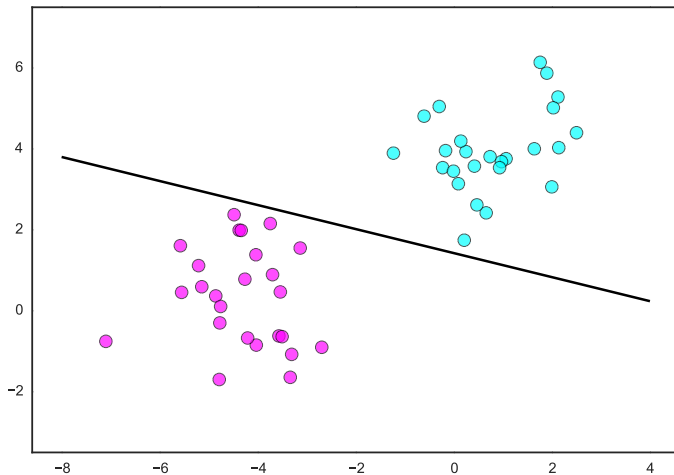
Sparse Violation Classifier

Support Vector Classifier

Loss Functions

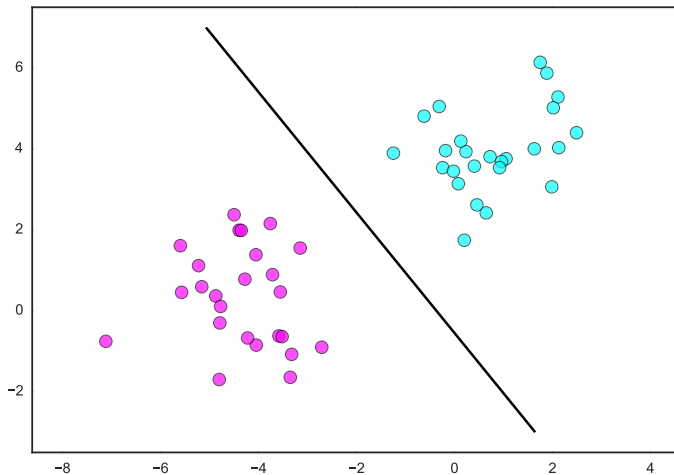
Which Separator?

## Which Separator?



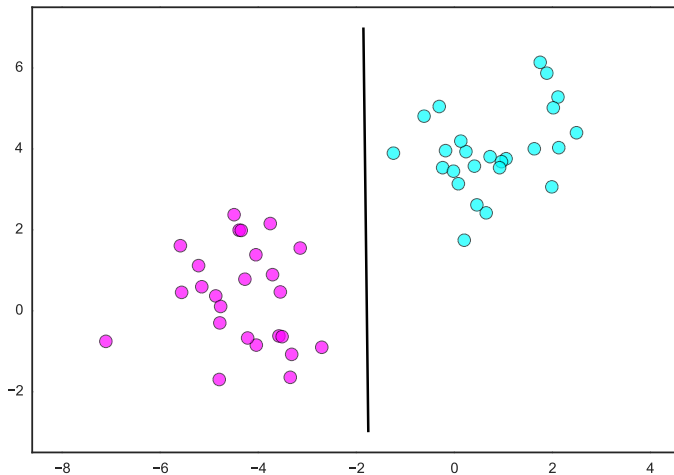
Which Separator?

## Which Separator?



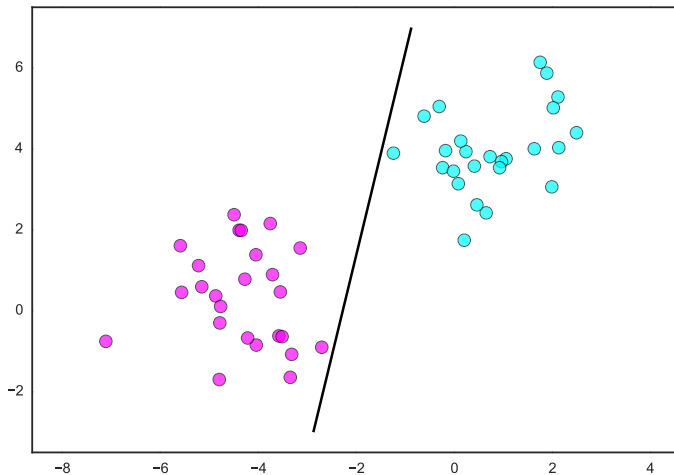
## Which Separator?

## Which Separator?



Which Separator?

## Which Separator?



Which Separator?

# Maximum Margin Classifier

# Outline

Introduction

Linearly Separable Problem

Which Separator?

**Maximum Margin Classifier**

Non-separable Linear Classification

Sparse Violation Classifier

Support Vector Classifier

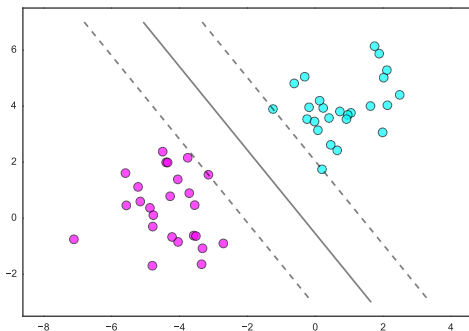
Loss Functions

# Maximum Margin Classifier

- ▶ infinitely many choices for separating hyperplane
- ▶ choose one which maximizes **width** of separating **slab**

$$\{x \mid -1 \leq a^T x - b \leq +1\}$$

- ▶ “maximum margin” or “robust linear” classifier





# Maximum Margin Classifier

- ▶ **margin**, or width of separating slab

$$\{x \mid -1 \leq a^T x - b \leq +1\}$$

is  $2/\|a\|_2$  (via linear algebra)

- ▶ suggests optimization problem

$$\begin{array}{ll} \text{maximize} & 2/\|a\|_2 \\ \text{subject to} & y_i (a^T x_i - b) \geq 1 \text{ for } i = 1, \dots, N \end{array}$$

- ▶ but not convex!

# Maximum Margin Classifier

- reformulate:

$$\text{maximize } 2/\|a\|_2 \iff \text{minimize } \|a\|_2$$

gives

$$\begin{array}{ll} \text{minimize} & \|a\|_2 \\ \text{subject to} & y_i (a^T x_i - b) \geq 1 \text{ for } i = 1, \dots, N, \end{array}$$

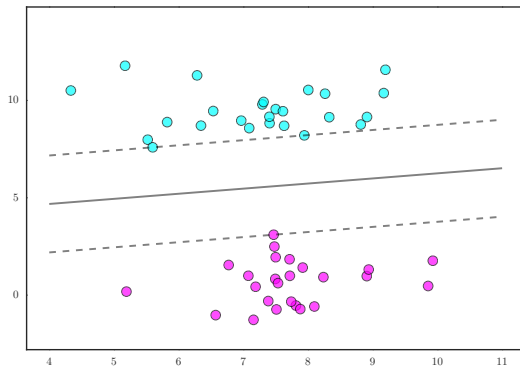
the **maximum margin classifier** (MMC) problem

# CVXPY

```
a = Variable(n)
b = Variable()

obj = Minimize(norm(a))
constr = [mul_elemwise(y, X*a - b) >= 1]
Problem(obj, constr).solve()
```

## Example



- ▶ note that max margin depends on only 3 tangent data points, called **support vectors**
- ▶ could throw away remaining data and get same solution

## Non-separable Linear Classification

# Outline

Introduction

Linearly Separable Problem

Which Separator?

Maximum Margin Classifier

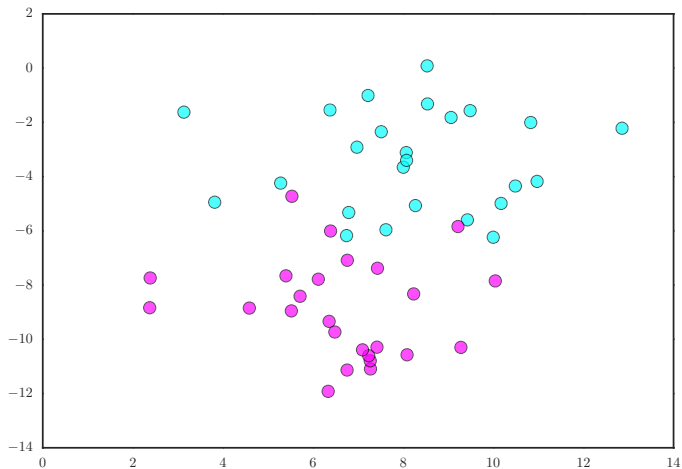
**Non-separable Linear Classification**

Sparse Violation Classifier

Support Vector Classifier

Loss Functions

## Non-separable Linear Classification



## Non-separable Linear Classification

- ▶ no separating hyperplane exists
- ▶ try finding linear separator

```
obj = Minimize(0)
constr = [mul_elemwise(y, X*a - b) >= 1]
prob = Problem(obj, constr)
prob.solve()
```

- ▶ results in `prob.status == 'infeasible'`



## Sparse Violation Classifier

# Outline

Introduction

Linearly Separable Problem

Which Separator?

Maximum Margin Classifier

Non-separable Linear Classification

Sparse Violation Classifier

Support Vector Classifier

Loss Functions

## Sparse Violation Classifier

- ▶ idea: “relax” constraints to make problem feasible
- ▶ add **slack** variables  $u \in \mathbf{R}_+^N$  to allow data points to be on “wrong side” of hyperplane

$$y_i (a^T x_i - b) \geq 1 - u_i, \quad u_i \geq 0$$

- ▶  $u_i = 0$ :  $x_i$  on **right** side of hyperplane
- ▶  $0 < u_i < 1$ :  $x_i$  on **right** side, but **inside slab**  $\{x \mid -1 \leq a^T x - b \leq +1\}$
- ▶  $u_i > 1$ :  $x_i$  on **wrong** side of hyperplane

# Sparse Violation Classifier

- ▶  $u$  gives measure of how much constraints are violated
- ▶ for large  $u$  can make **any** data feasible
- ▶ want  $u$  “small”; minimize its sum

$$\begin{array}{ll}\text{minimize} & \mathbf{1}^T u \\ \text{subject to} & y_i (a^T x_i - b) \geq 1 - u_i \text{ for } i = 1, \dots, N \\ & u \geq 0\end{array}$$

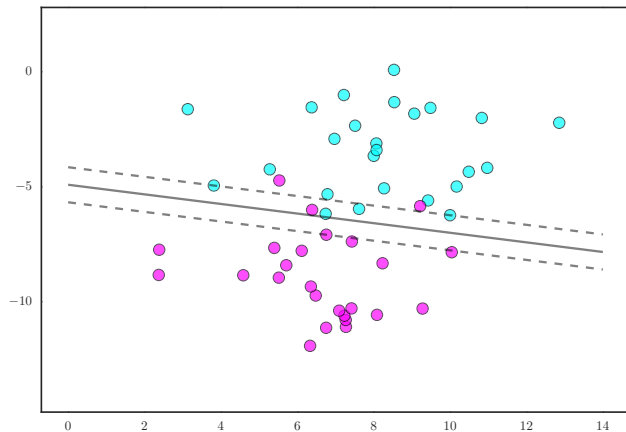
- ▶ I'll call it **sparse violation classifier** (SpVC)
- ▶  $\mathbf{1}^T u = \|u\|_1$ , since  $u \geq 0$ ; good **heuristic** for separator with few (sparse) violations

# CVXPY

```
a = Variable(n)
b = Variable()
u = Variable(N)

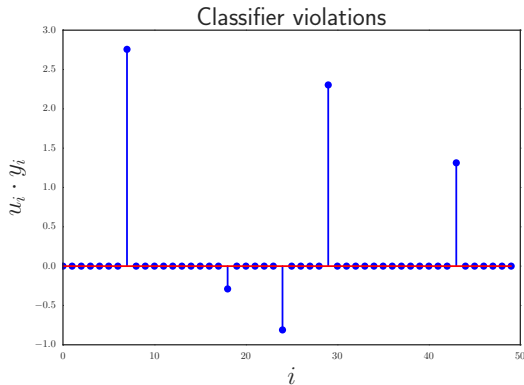
obj = Minimize(sum_entries(u))
constr = [mul_elemwise(y, X*a - b) >= 1 - u, u >= 0]
Problem(obj, constr).solve()
```

## Example



- solution depends only on points inside of, tangent to, or on wrong side of slab

## Example



- ▶ “+” class has 3 misclassified points
- ▶ “-” class has 2 correctly classified, but inside slab

# Support Vector Classifier



# Outline

Introduction

Linearly Separable Problem

Which Separator?

Maximum Margin Classifier

Non-separable Linear Classification

Sparse Violation Classifier

Support Vector Classifier

Loss Functions

# Support Vector Classifier

- ▶ idea: combine aspects of last two classifiers
  - ▶ sparse violations of SpVC
  - ▶ robustness of large separating slab in MMC
- ▶ optimize both:

$$\begin{aligned} & \text{minimize} && \|a\|_2 + \rho \mathbf{1}^T u \\ & \text{subject to} && y_i (a^T x_i - b) \geq 1 - u_i \text{ for } i = 1, \dots, N \\ & && u \geq 0 \end{aligned}$$

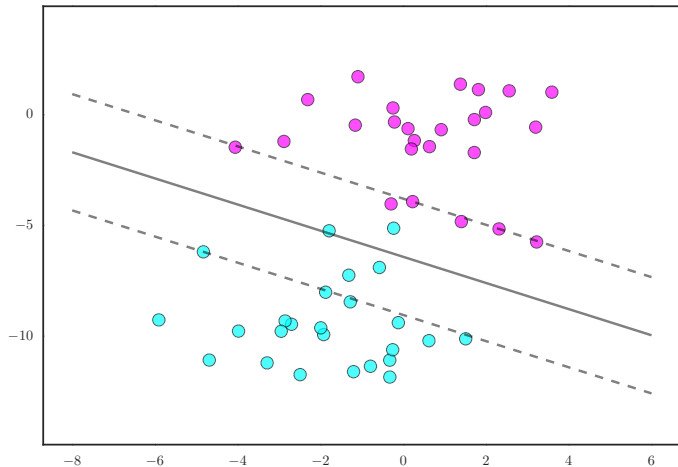
- ▶  $\rho > 0$  trades-off between margin  $2/\|a\|_2$  and classification violations  $\mathbf{1}^T u$  (multi-objective optimization)
- ▶ **support vector classifier (SVC)**

## CVXPY

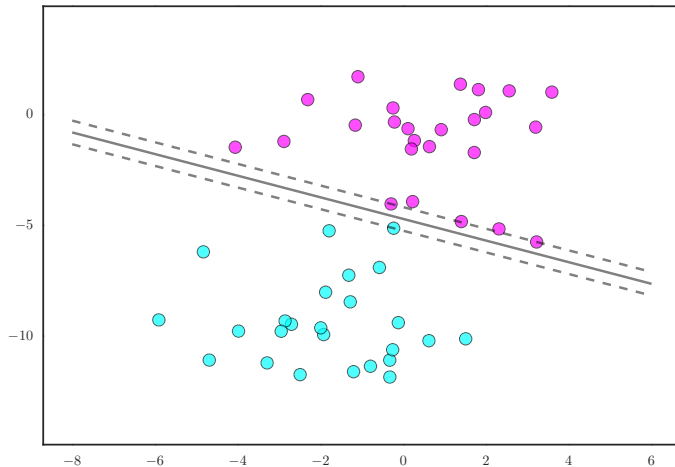
```
a = Variable(n)
b = Variable()
u = Variable(N)
rho = .1

obj = Minimize(norm(a) + rho*sum_entries(u))
constr = [mul_elemwise(y, X*a - b) >= 1 - u, u >= 0]
Problem(obj, constr).solve()
```

Example with  $\rho = .1$



Example with  $\rho = 10$



# Loss Functions

# Outline

Introduction

Linearly Separable Problem

Which Separator?

Maximum Margin Classifier

Non-separable Linear Classification

Sparse Violation Classifier

Support Vector Classifier

Loss Functions

## Hinge Loss

- ▶ in SpVC, it follows from  $y_i (a^T x_i - b) \geq 1 - u_i$ ,  $u_i \geq 0$ , that

$$u_i = \begin{cases} 0 & y_i (a^T x_i - b) \geq 1 \\ 1 - y_i (a^T x_i - b) & y_i (a^T x_i - b) < 1 \end{cases}$$

- ▶ rewrite as  $u_i = \ell_h [y_i (a^T x_i - b)]$ , where

$$\ell_h(z) = \begin{cases} 0 & z \geq 1 \\ 1 - z & z < 1 \end{cases}$$

is the **hinge loss** function, equivalently:  $\max(0, 1 - z)$  or  $(1 - z)_+$



## Hinge Loss SpVC

- ▶ note that  $\ell_h$  is convex, so we can rewrite SpVC as the **equivalent problem**

$$\text{minimize} \quad \sum_{i=1}^N \ell_h \left[ y_i \left( a^T x_i - b \right) \right]$$

- ▶ unconstrained (non-differentiable) convex problem
- ▶ in CVXPY:

```
def hinge(z):  
    return pos(1-z)  
  
r = mul_elemwise(y, X*a - b)  
obj = Minimize(sum_entries(hinge(r)))  
Problem(obj).solve()
```

# Hinge Loss SVC

- ▶ can rewrite SVC as the **unconstrained** problem

$$\text{minimize} \quad \|a\|_2 + \rho \sum_{i=1}^N \ell_h \left[ y_i \left( a^T x_i - b \right) \right]$$

- ▶ completely **equivalent** to the SVC formulation from before
- ▶ common form for classification problems:

$$\text{minimize} \quad r(a) + \rho \sum_{i=1}^N \ell \left[ y_i \left( a^T x_i - b \right) \right]$$

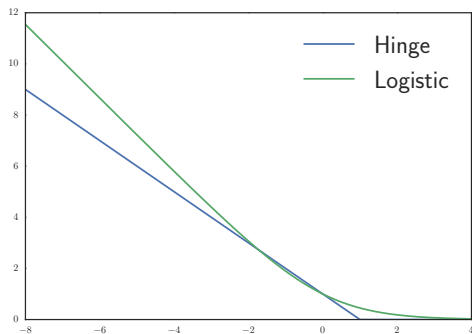
- ▶  $\ell$  is a **loss function** (fit to data)
  - ▶  $r$  is a **regularizer** (prior on parameters)
- ▶ mix and match regularizers and loss functions for different types of classification

# Logistic Loss

- ▶ **logistic loss** is an alternative to hinge loss:

$$\ell_L(z) = \log(1 + \exp(-z))$$

- ▶ convex, but not immediately obvious (2nd derivative test?)



# Logistic Regression

- ▶ get classic **logistic regression** with

$$\text{minimize } r(a) + \rho \sum_{i=1}^N \ell \left[ y_i \left( a^T x_i - b \right) \right]$$

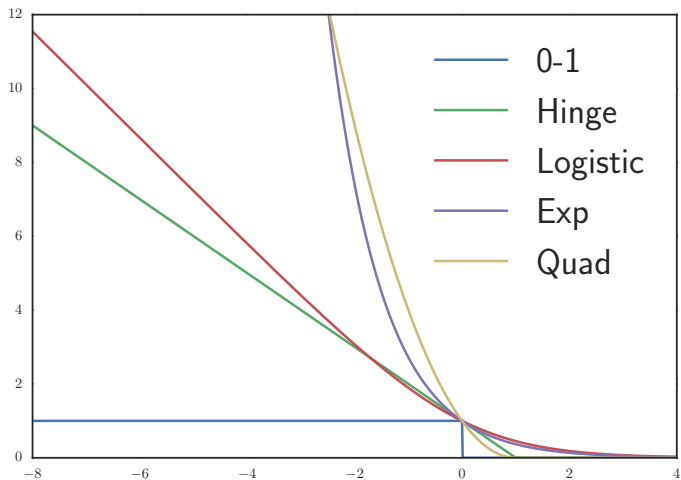
when:

- ▶  $r(a) \equiv 0$
- ▶  $\ell(z) = \ell_L(z)$
- ▶ **regularized** logistic regression when  $r(a)$  is  $\|a\|_2$  or  $\|a\|_1$  (sparsity)

## Other Loss Functions

- ▶ hard loss:  $\ell_{\text{hard}}(z) = \begin{cases} 0 & z \geq 1 \\ +\infty & z < 1 \end{cases}$
- ▶ exponential loss:  $\ell_{\text{exp}}(z) = \exp(-z)$
- ▶ quadratic loss:  $\ell_2(z) = (1 - z)_+^2$
- ▶ (nonconvex) 0-1 loss:  $\ell_{0-1}(z) = \begin{cases} 0 & z \geq 1 \\ 1 & z < 1 \end{cases}$

## Other Loss Functions



# Unified Models

- ▶ linear separator feasibility problem:
  - ▶  $\ell_{\text{hard}}, r \equiv 0$
- ▶ MMC:
  - ▶  $\ell_{\text{hard}}, r(a) = \|a\|_2$
- ▶ SpVC:
  - ▶  $\ell_h, r \equiv 0$
- ▶ SVC:
  - ▶  $\ell_h, r(a) = \|a\|_2$
- ▶ Logistic regression:
  - ▶  $\ell_L, r \equiv 0$
- ▶ many other options for modeling
  - ▶ loss for max violation instead of sum
  - ▶ sparse  $a$  for feature selection

## other

- ▶ show indicator function to show same as feasibility problem
- ▶ logistic loss is logistic regression
- ▶ other random loss functions
- ▶ infinity norm of violations for max violation



# Non-separable Linear Classification

- ▶ relaxed feasibility problem
- ▶  $l_1$  penalty to minimize misclassification: pure LP
- ▶ tradeoff between classification and width of slab: SOCP

# Hinge loss

- ▶ reformulate as hinge loss objective
- ▶ general loss function form. . .  $l(Ax + b)$

# logistic

- ▶ change loss function to get logistic loss
- ▶ other loss functions

# regularization

- ▶ regularize to get sparse classifier. . .

## nonlinear discrimination

- ▶ adding features
- ▶ polynomial discrimination any different?
- ▶ rbf kernel? radial basis function
- ▶ kernel methods and relationship with convex opt. . .

# algorithms

- ▶ note that so far, we have said **nothing** about **how** to compute a supporting vector
- ▶ we have focused on modeling
- ▶ that's OK, we're focusing on modeling
- ▶ algorithms involve duality and optimality conditions

## scikitlearn comparison

- ▶ make sure it matches up with python SVM formulation
- ▶ maybe even do a timing comparison. . .

## data science perspective

- ▶ cleaning and centering data
- ▶ sparse predictors