CME 252: Support Vector Machines

AJ Friend
ICME, Stanford University

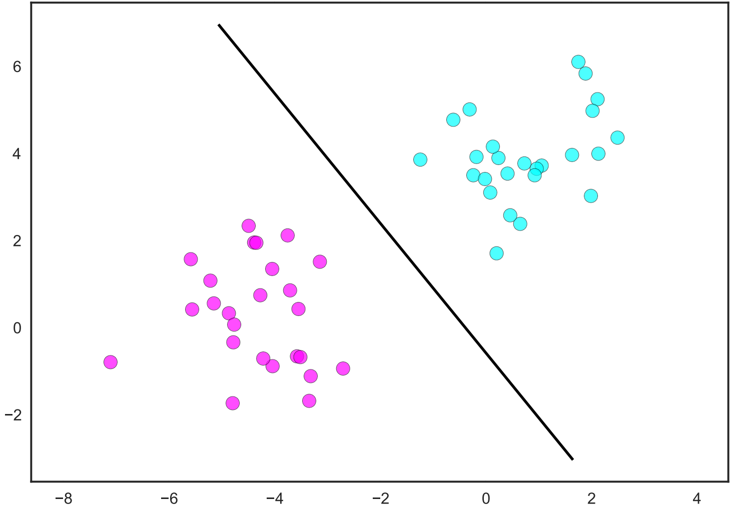
Introduction

Outline

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

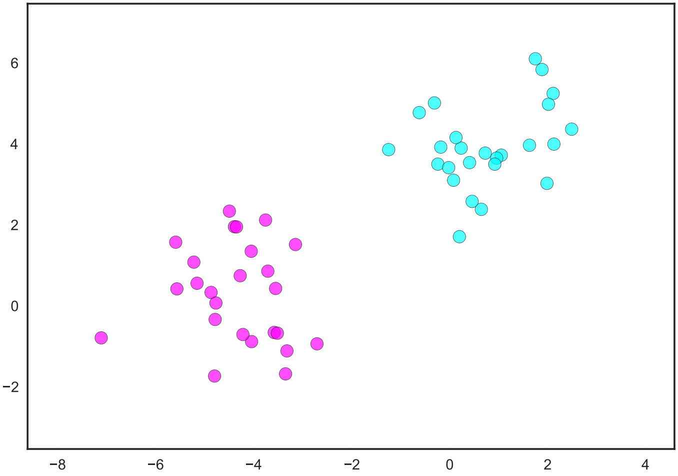
Support Vector Machines



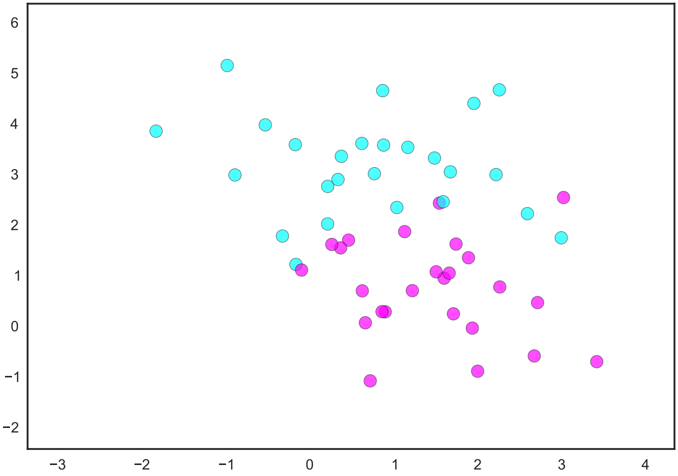
Scenarios

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

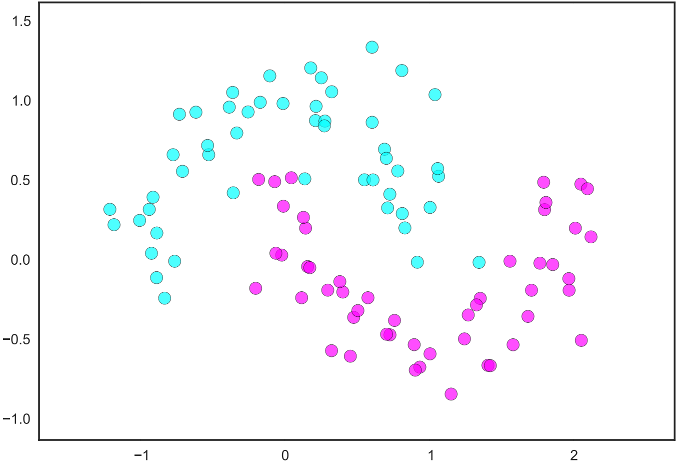
Strictly Linearly Separable Data



Approximately Linearly Separable Data



Approximately Non-linearly Separable



Linearly Separable Problem

Outline

Linearly Separable Problem

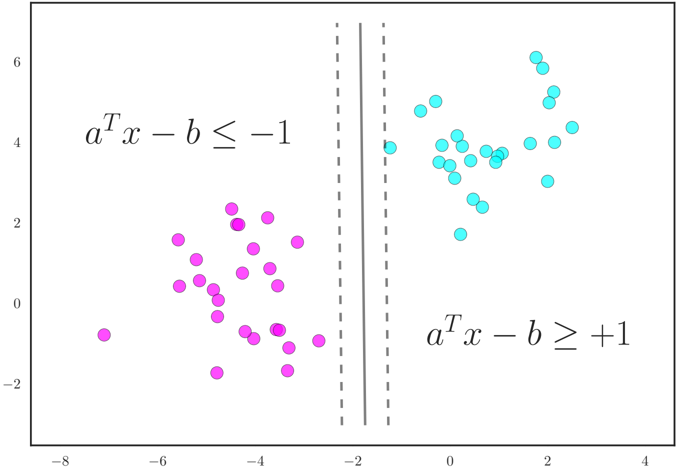
- data: $x_i \in \mathbf{R}^n$ with labels $y_i \in \{+1, -1\}$ for $i = 1, \ldots, N$
- assume **strictly** linearly separable
- find hyperplane $\{x \mid a^T x = b\}$ that separates points by label

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

- **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

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

- get **feasibility** problem

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

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

CVXPY for Separable Problem

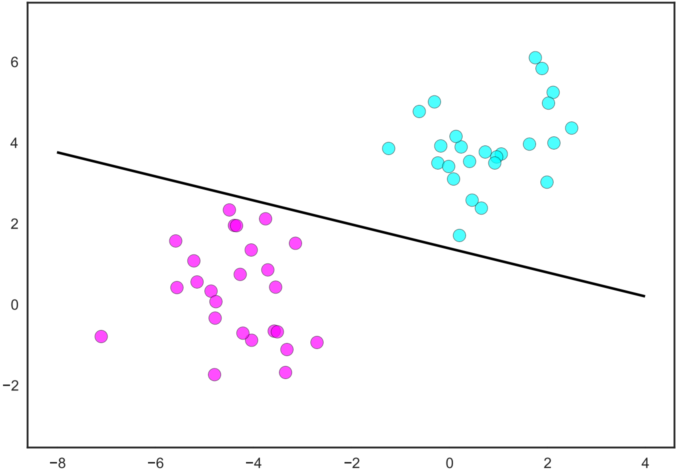
```
a = Variable(n)
b = Variable()

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

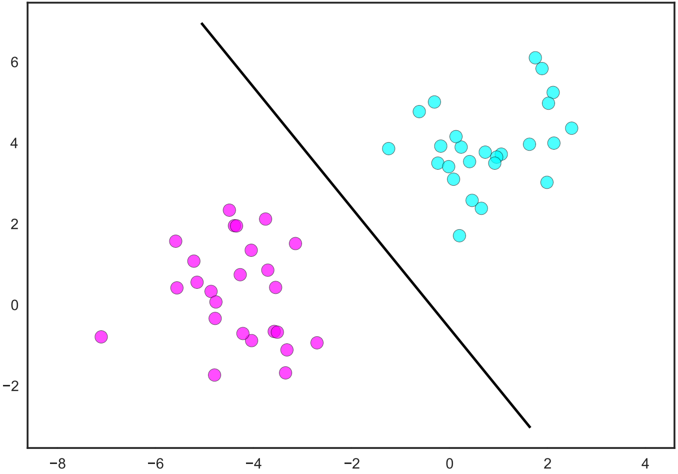
Which Separator?

Outline

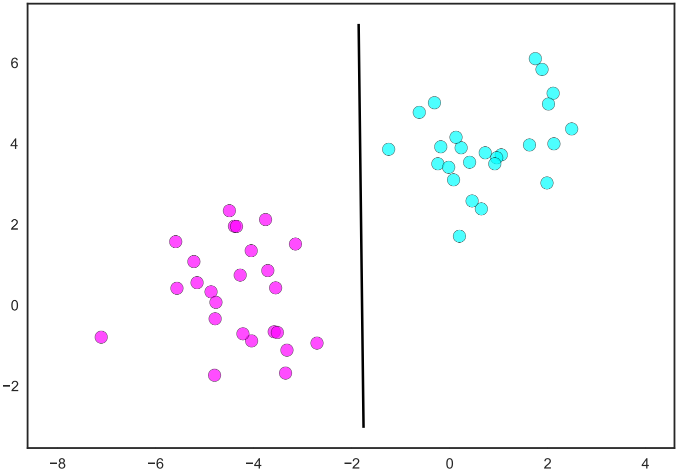
Which Separator?



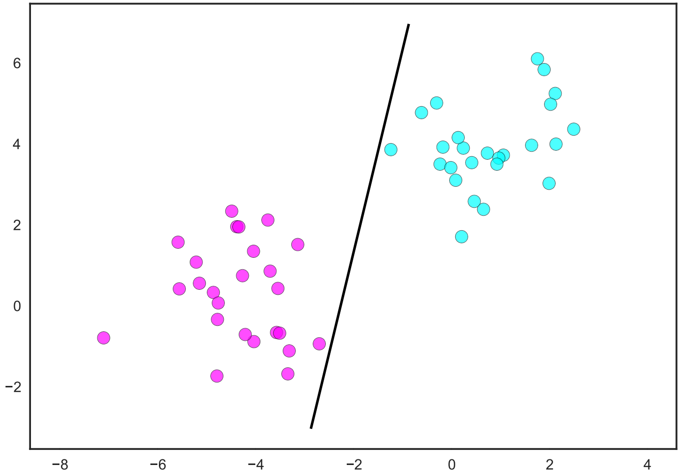
Which Separator?



Which Separator?



Which Separator?



Maximum Margin Classifier

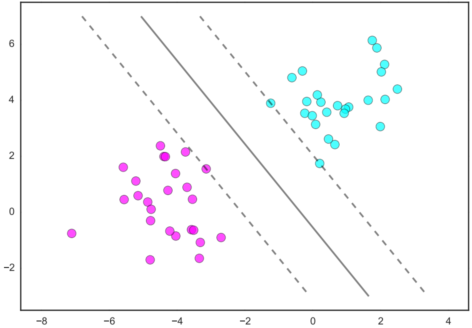
Outline

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

- width of separating slab

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

  is $2/\|a\|_2$ (via linear algebra)
- suggests optimization problem

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

- but not convex!

Maximum Margin Classifier

- reformulate:

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

gives

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

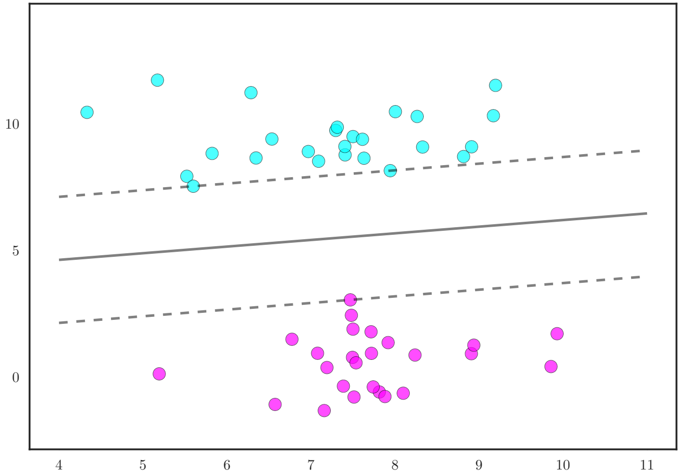
the **maximum margin classifier** problem

Maximum Margin Classifier in CVXPY

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

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

Maximum Margin Classifier



Non-separable Linear Classification

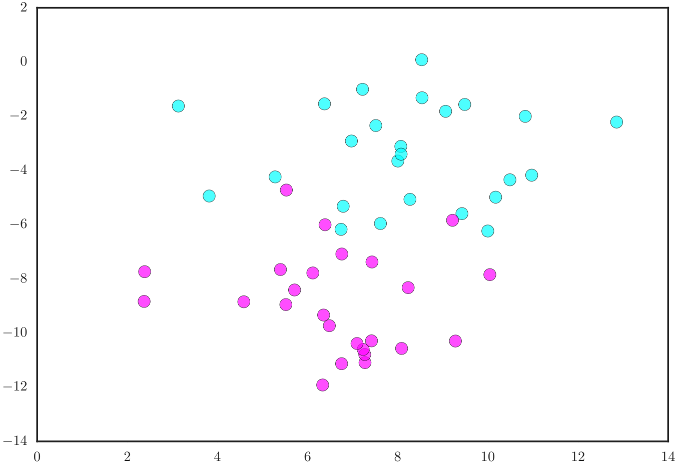
Outline

Non-separable Linear Classification

- reformulate as hinge loss function
- show indicator function to show same as feasibility problem
- logistic loss is logistic regression
- other random loss functions

"violation of margins/constraints" - combine relaxation with width of slab for "support vector classifier"

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'`

Non-separable Linear Classification

- 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

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

  - $u_i = 0$: 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

Non-separable Linear Classification

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

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

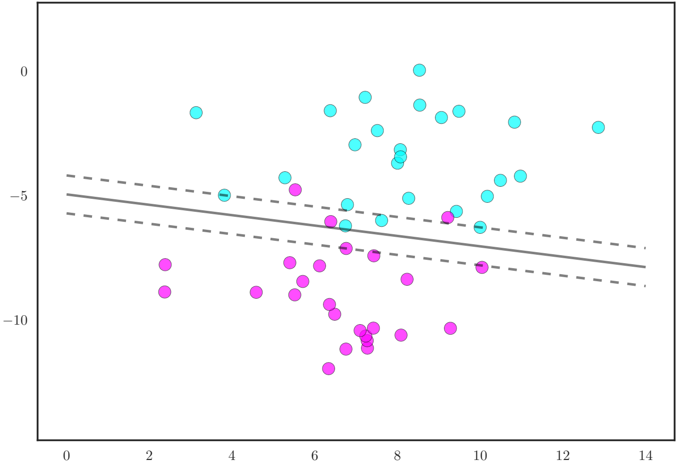
- called **support vector classifier**
- $\mathbf{1}^T u = \|u\|_1$, since $u \geq 0$; good **heuristic** for separator with few (sparse) violations

CVXPY

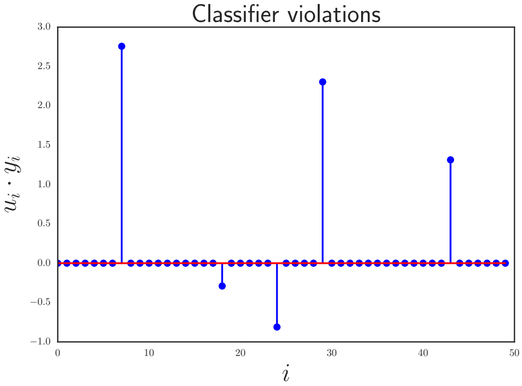
```
a = Variable(n)
b = Variable()
u = Variable(N)

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

Example



Example



- "$+$" class has 3 misclassified points
- "$-$" class has 2 correctly classified, but inside slab

Hinge loss

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

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

- rewrite as $u_i = \ell_h\left[y_i\left(a^T x_i - b\right)\right]$, 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 SVC

- note that ℓ_h is convex, so we can rewrite SVC as the **equivalent problem**

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

- 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()
```

Non-separable Linear Classification

- relaxed feasibility problem
- l1 penality to minimize misclassificaiton: 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