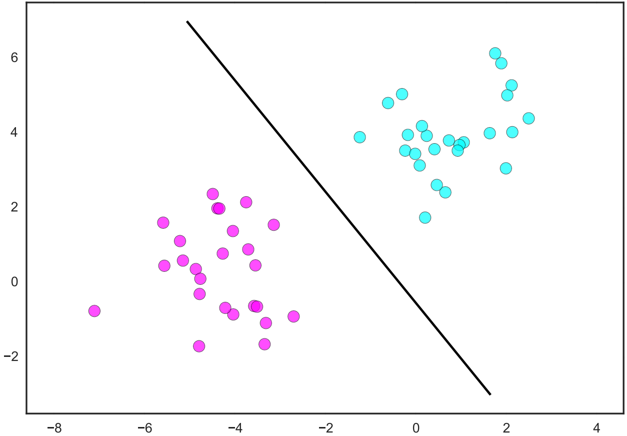
CME 252: Support Vector Machines

AJ Friend
ICME, Stanford University

Support Vector Machines

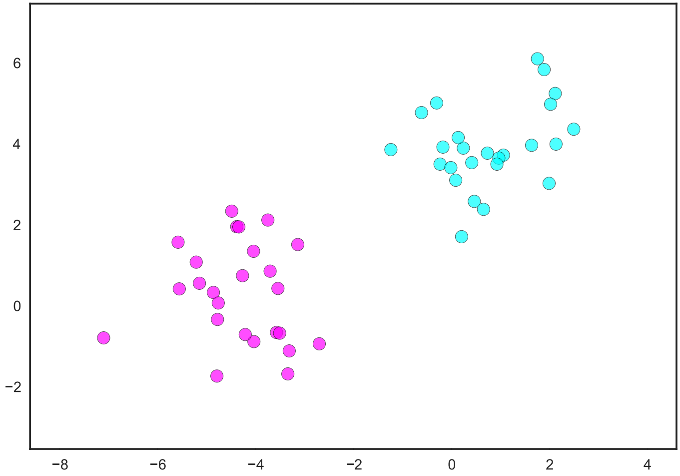
- find a hyperplane to separate data points into two classes
- use hyperplane to classify new (unseen) points



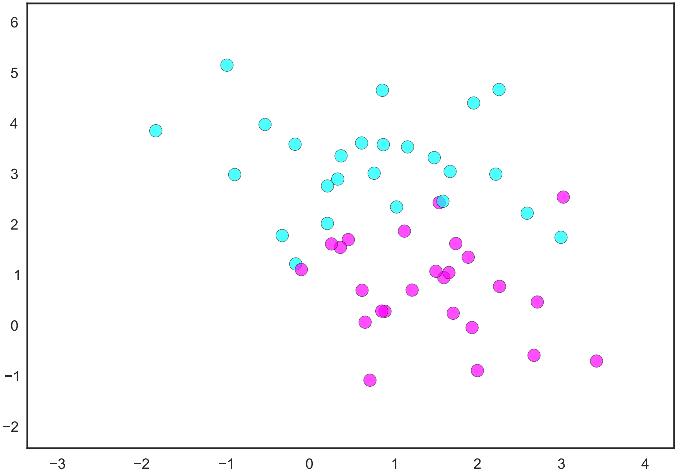
Scenarios

- assume data falls into one category:
  - strictly linearly separable
  - approximately (not strictly) linearly separable
  - approximately non-linearly separable (hyperplanes won't work)

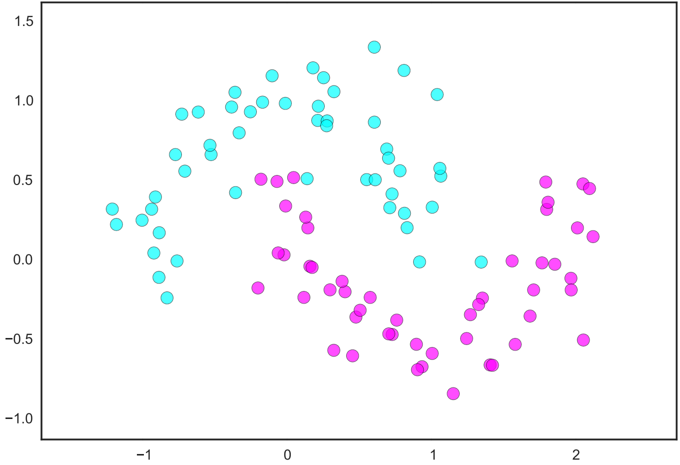
Strictly Linearly Separable Data



Approximately Linearly Separable Data



Approximately Non-linearly Separable



Linearly Separable Problem

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

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

- **rescale** a, b so that

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

more problem

- for all i, rewrite as

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

- get **feasibility** problem

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

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

Separable linear classification/discrimination

- many hyperplanes
- maximum margin classifier and robustness

Nonseparable 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