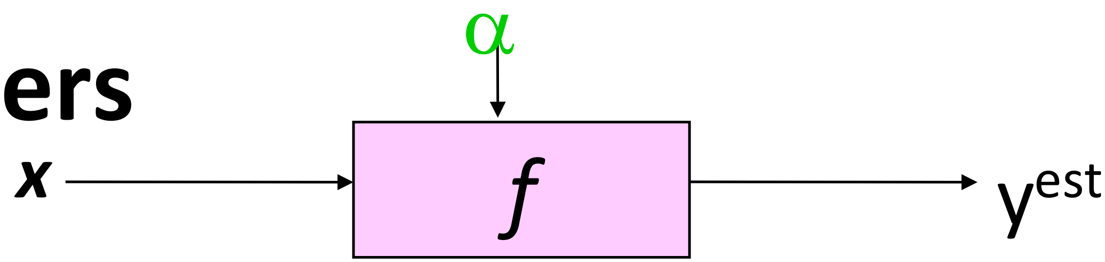
Support Vector Machines

Some slides borrowed from Andrew Moore's slides on SVMs.
His repository is here: http://www.cs.cmu.edu/~awm/tutorials .

Support Vector Machines

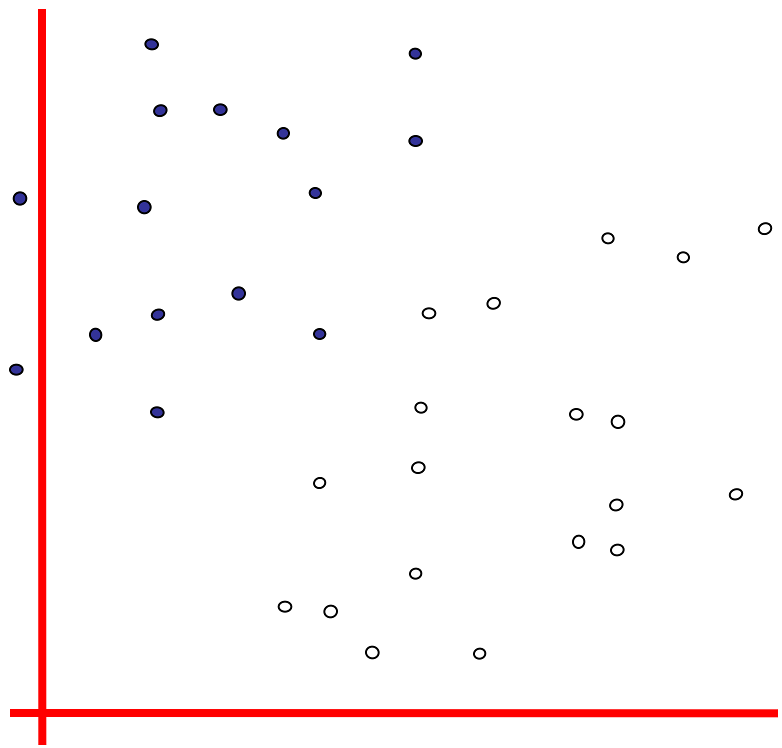
- Very popular ML technique
  - Became popular in the late 90s (Vapnik 1995; 1998)
  - Invented in the late 70s (Vapnik, 1979)
- Controls complexity and overfitting, so works well on a wide range of practical problems
- Can handle high dimensional vector spaces, which makes feature selection less critical
- Very fast and memory efficient implementations, e.g., [svm_light]
- Not always best solution, especially for problems with small vector spaces

Linear Classifiers



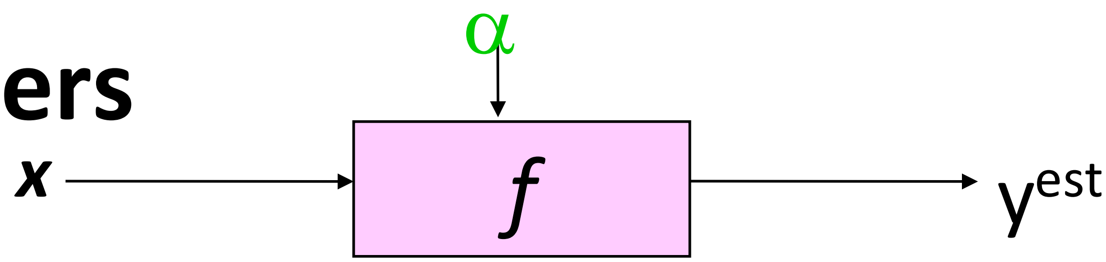
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

- • denotes +1
- ∘ denotes -1

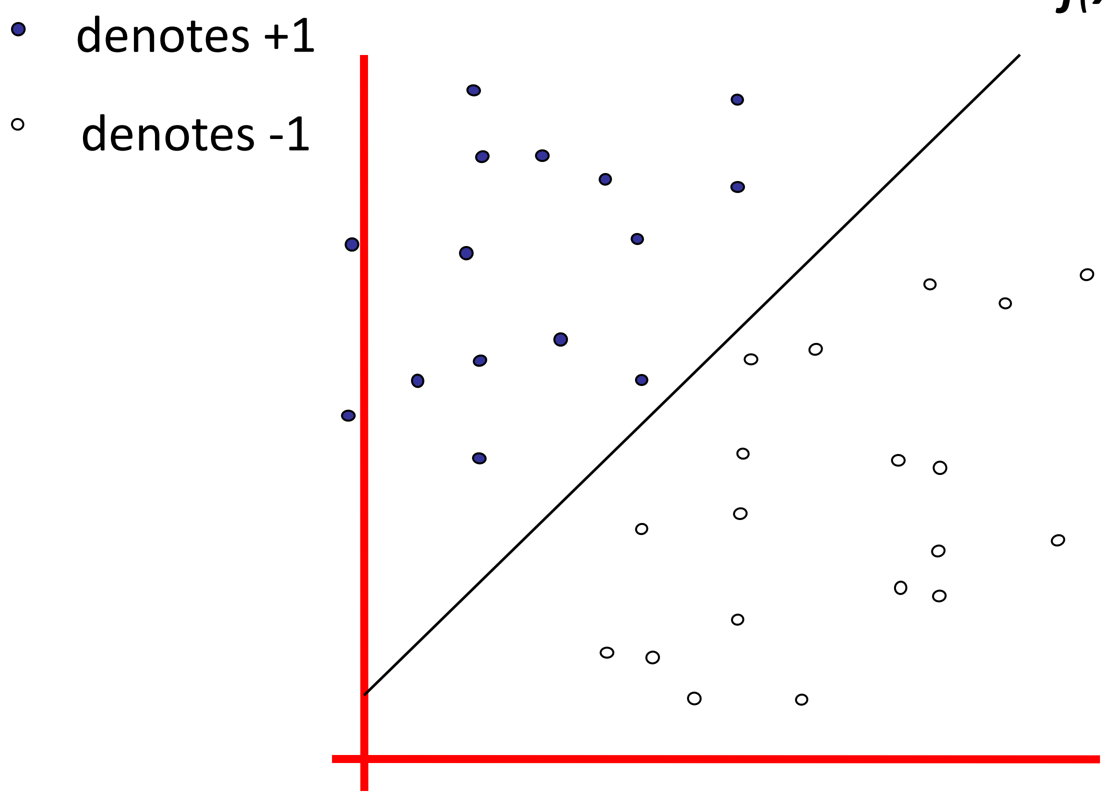


How would you classify this data?

Linear Classifiers

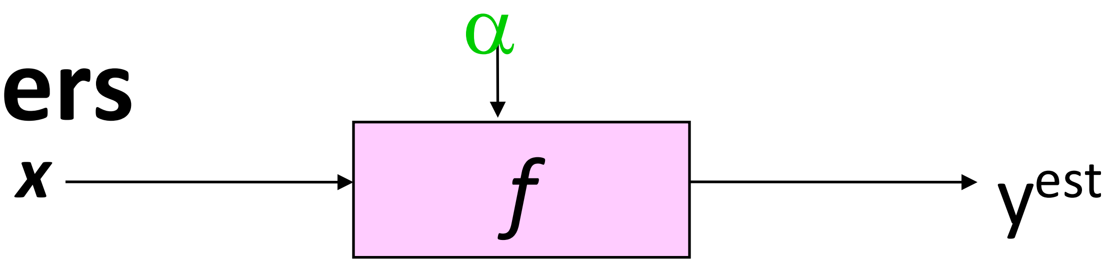


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$



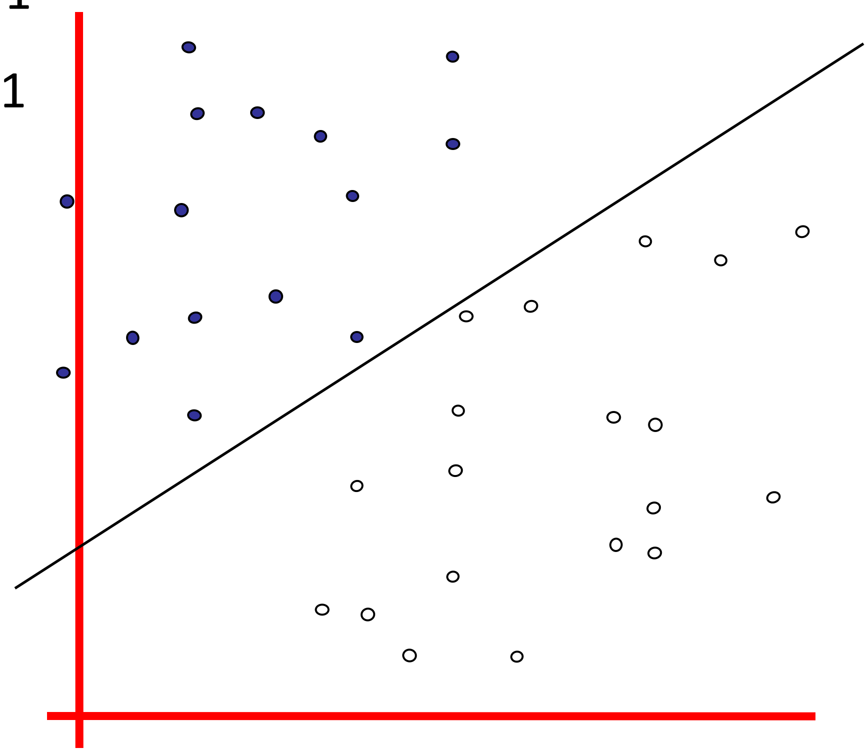
How would you classify this data?

Linear Classifiers



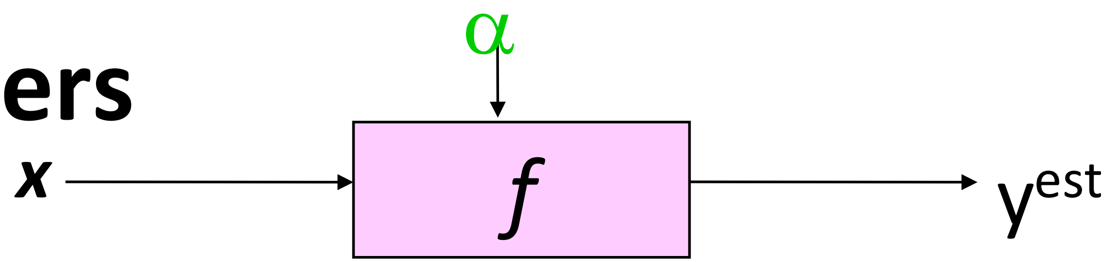
$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$$

- denotes +1
- denotes -1



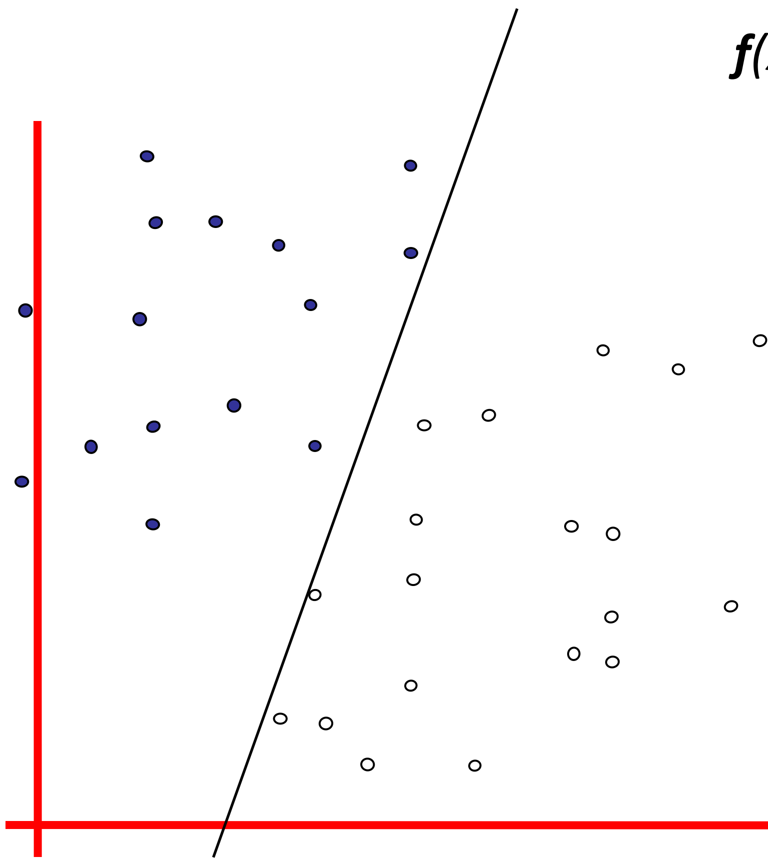
How would you classify this data?

Linear Classifiers



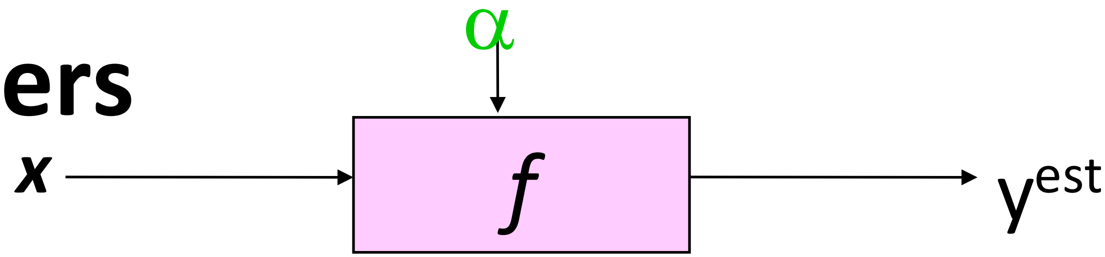
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w} \cdot \mathbf{x} - b)$

- denotes +1
- denotes -1



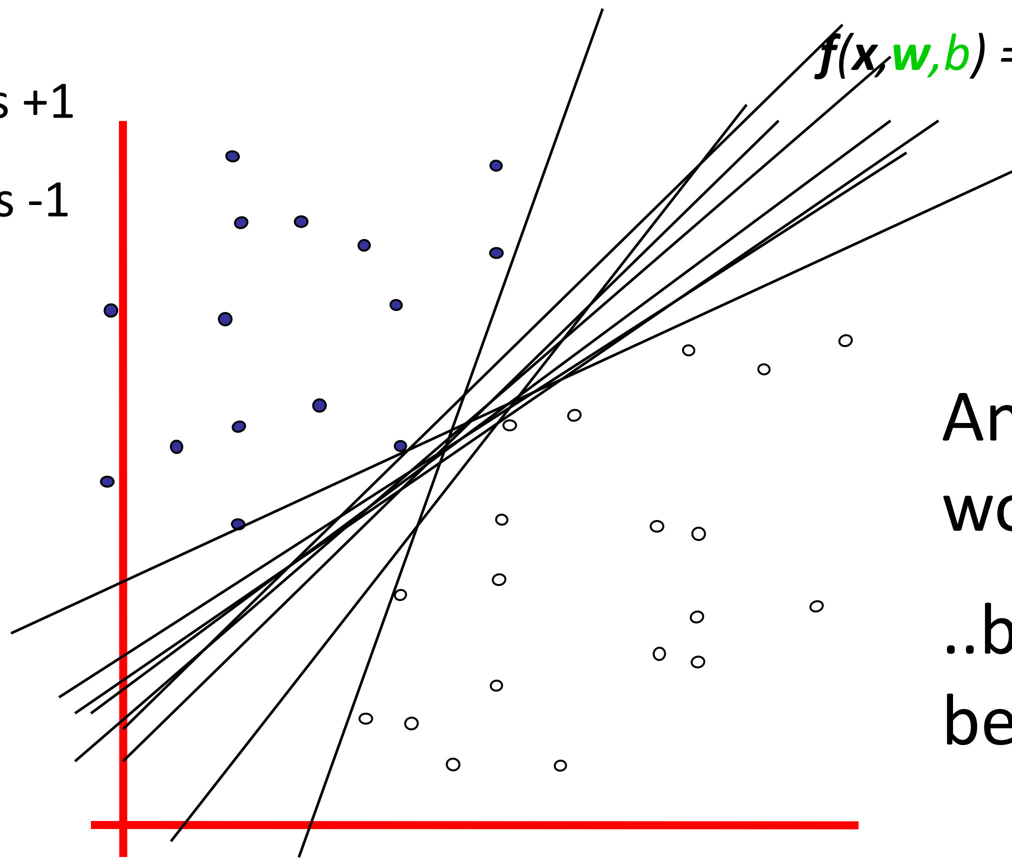
How would you classify this data?

Linear Classifiers



$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}.\ \boldsymbol{x} - b)$

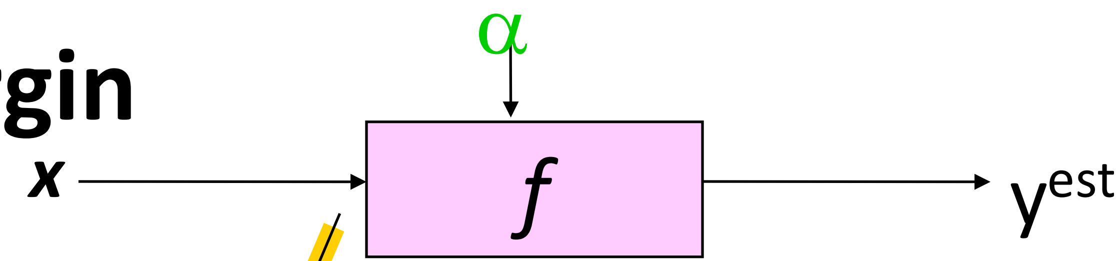
- denotes +1
- denotes -1



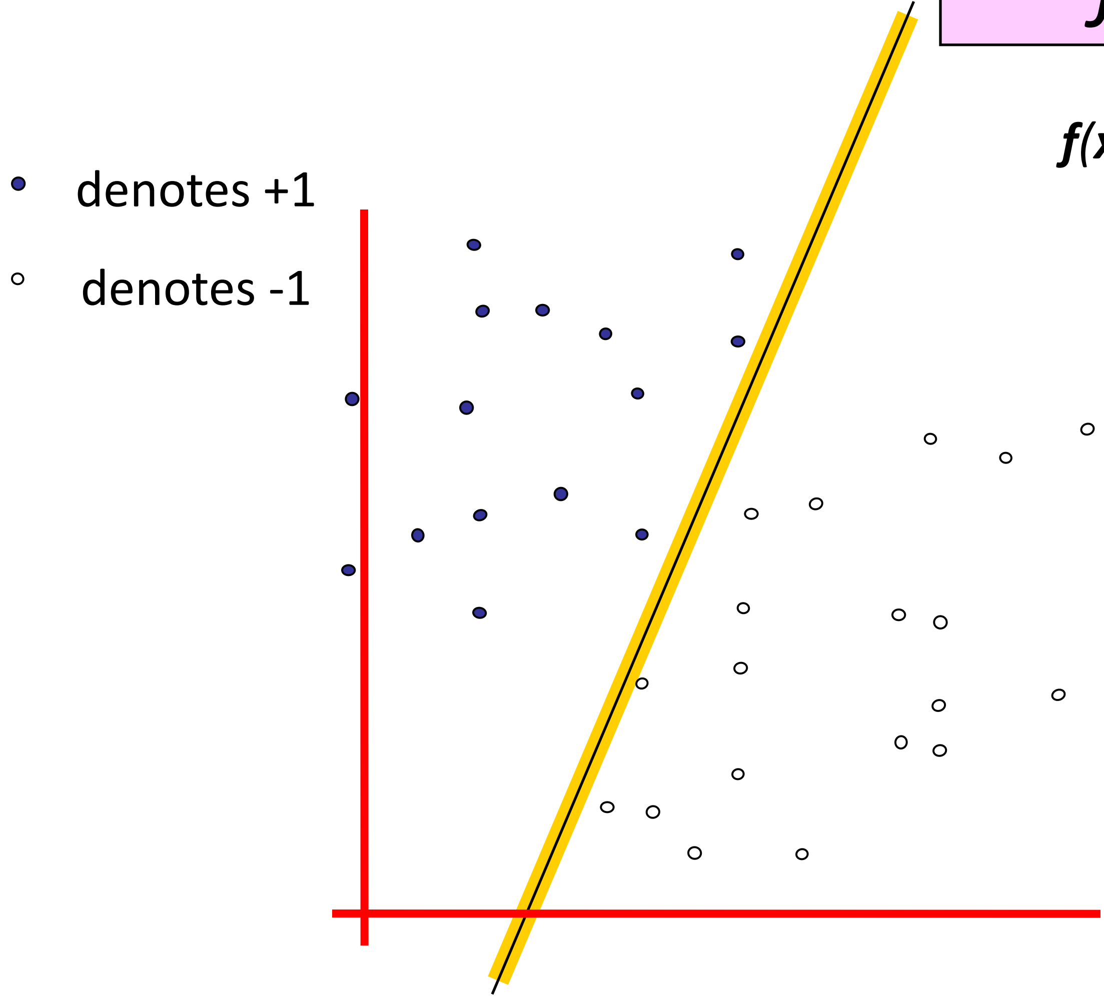
Any of these would be fine..

..but which is best?

Classifier Margin

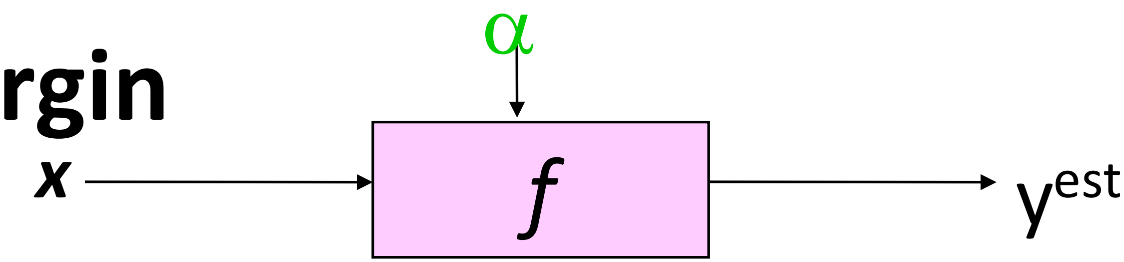


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$



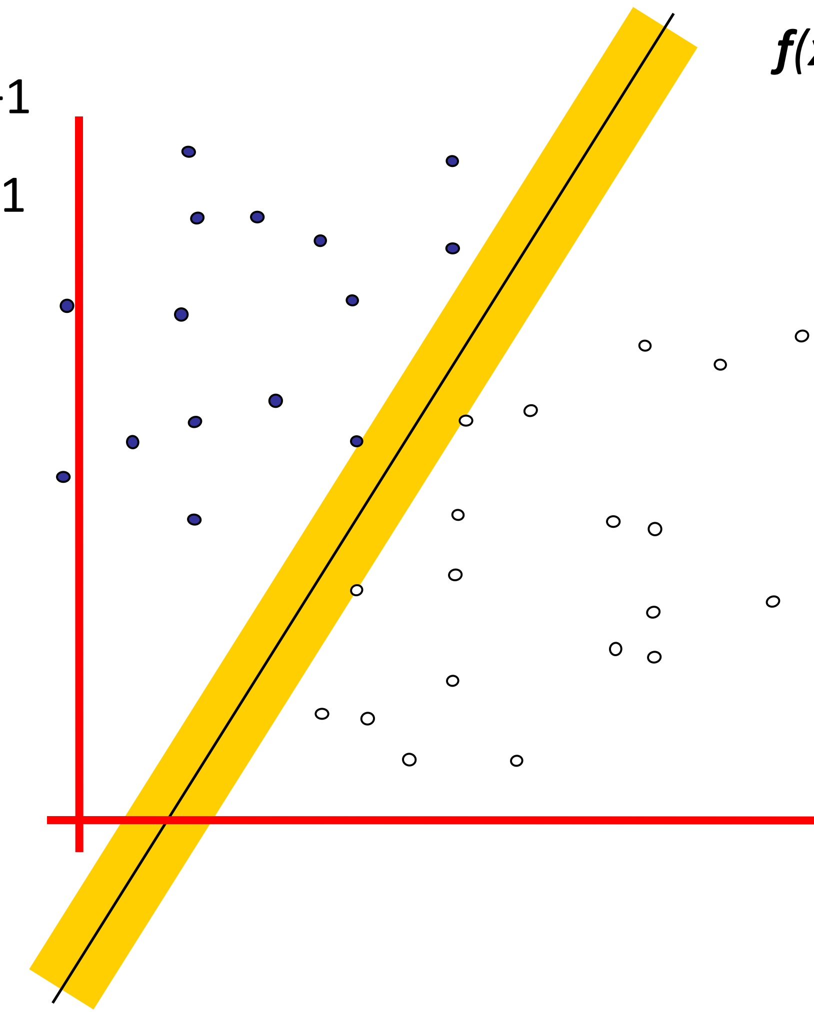
A linear classifier's margin is width that boundary could be increased by before hitting a datapoint

Maximum Margin



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

- denotes +1
- denotes -1

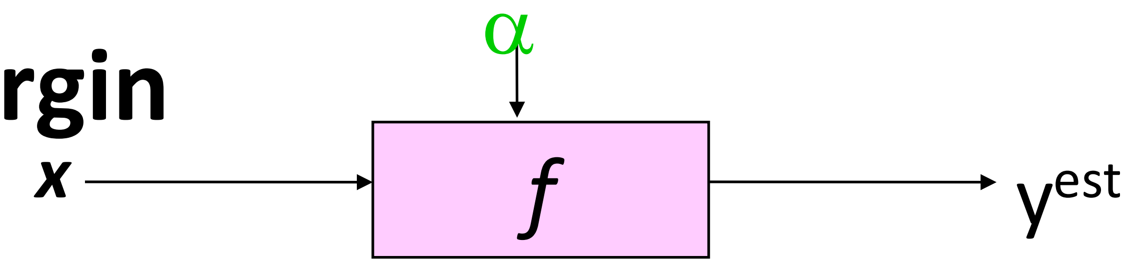


Maximum margin linear classifier is the linear classifier with the, um, maximum margin

The simplest kind of SVM, called an LSVM

Linea SVM

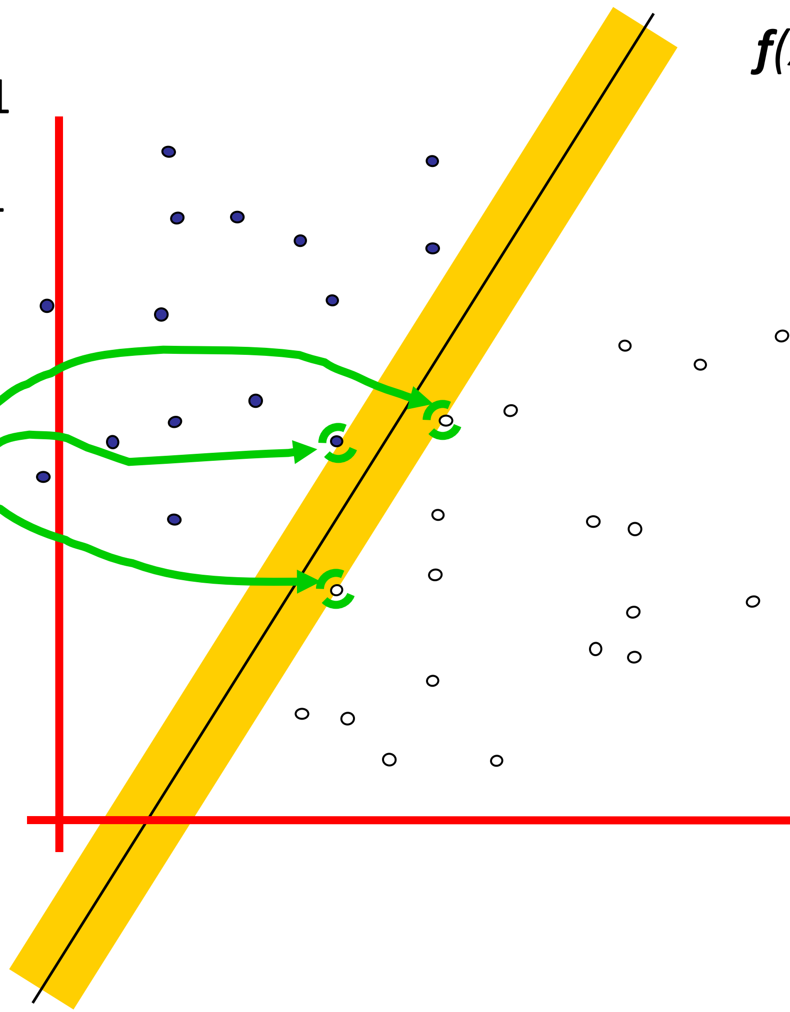
Maximum Margin



$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$$

• denotes +1

° denotes -1



Support Vectors are the datapoints that margin pushes up against

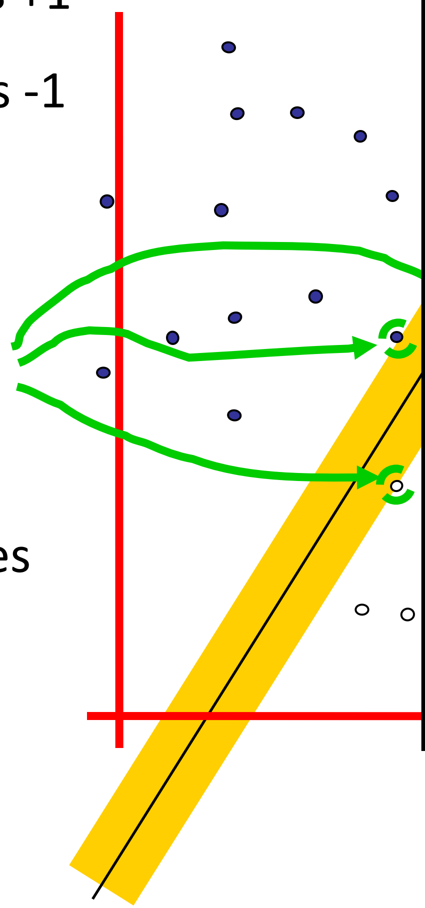
Maximum margin linear classifier is the linear classifier with the, um, maximum margin

The simplest kind of SVM, called an LSVM

Linea SVM

Why Maximum Margin?

- denotes +1
- denotes -1

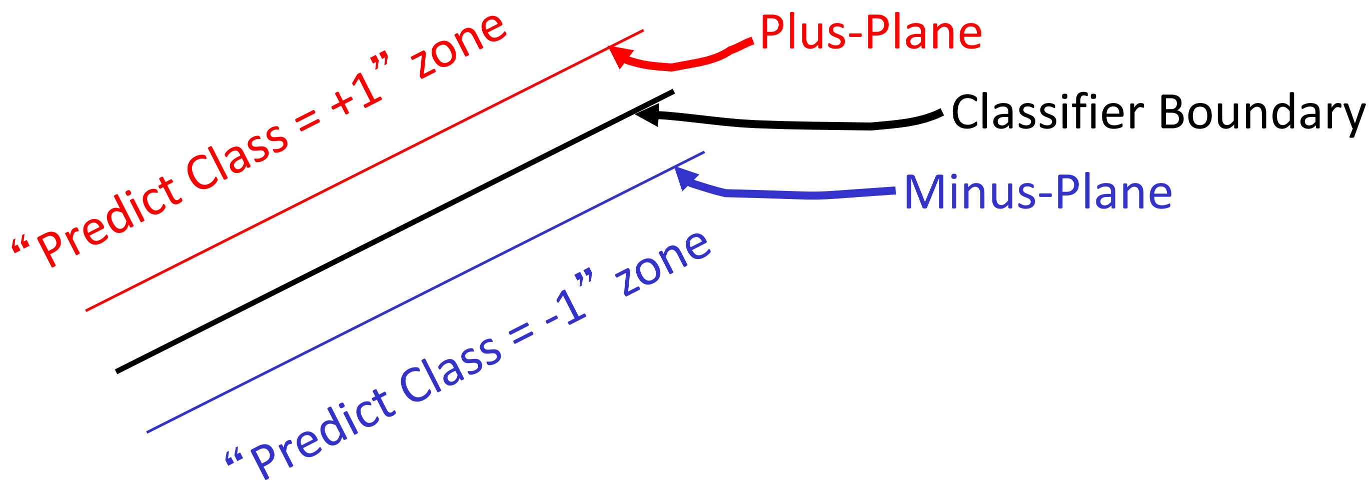


Support Vectors are those datapoints that the margin pushes up against

1. Intuitively this feels safest
2. Small errors in boundary location unlikely to cause misclassification
3. LOOCV is easy since model is immune to removal of non-support-vector datapoints
4. There’s some theory (using VC dimension) that is related to (but not the same as) the proposition that this is a good thing
5. Empirically it works very very well

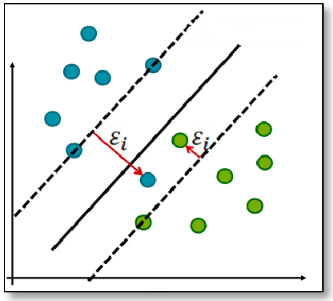
LOOCV = leave one out cross validation

Specifying a line and margin



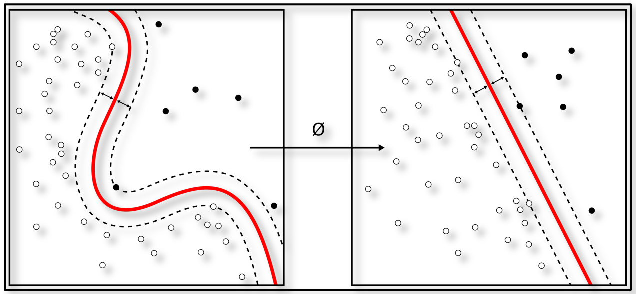
- How do we represent this mathematically?
- ...in *m* input dimensions?

Soft margin classification



- What if data from two classes not linearly separable?
- Allow a fat decision margin to make a few mistakes
- Some points, outliers or noisy examples, are inside or on wrong side of the margin
- Each outlier incurs a cost based on distance to hyperplane

Kernel trick



- What if data from two classes not linearly separable?
- Project data onto a higher dimensional space where it becomes linearly separable
- Many SVMs can take an argument, a kernel, that does the transformation of the data
- Deciding what kernel function to use is done through eperimentation

SVM Performance

- Can handle very large features spaces (e.g., 100K features)
- Relatively fast
- Anecdotally they work very, very well indeed
- Example: They are among the best-known classifier on a well-studied hand-written-character recognition benchmark

Binary vs. multi classification (1)

- SVMs can only do **binary** classification
- Two approaches to multi classification: OVA and OVO
- OVA or **one-vs-all**: turns n-way classification into n binary classification tasks:
  - E.g., for zoo problem, do mammal vs. not-mammal, fish vs. not-fish, … bird vs. not-bird, …
  - Pick one that results in the highest score (e.g., widest margin)

Binary vs. multi classification (2)

- OVO for n classes builds N*(N-1)/2 one-vs-one classifiers
    - Mammal vs. fish, mammal vs. reptile, etc…
- Choose the on that wins the most 1x1 pairings