

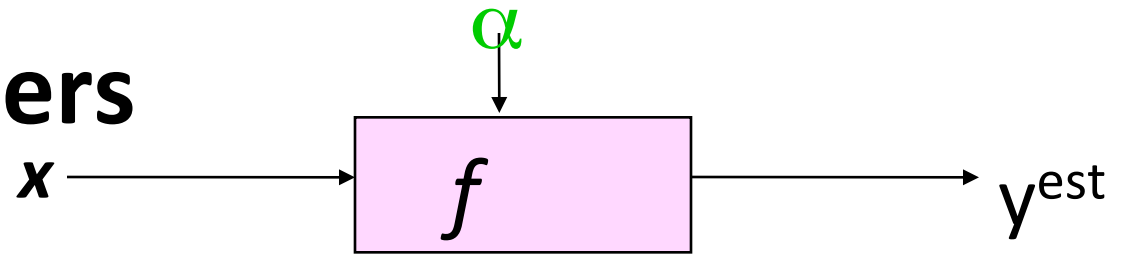
Support Vector Machines

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

Support Vector Machines

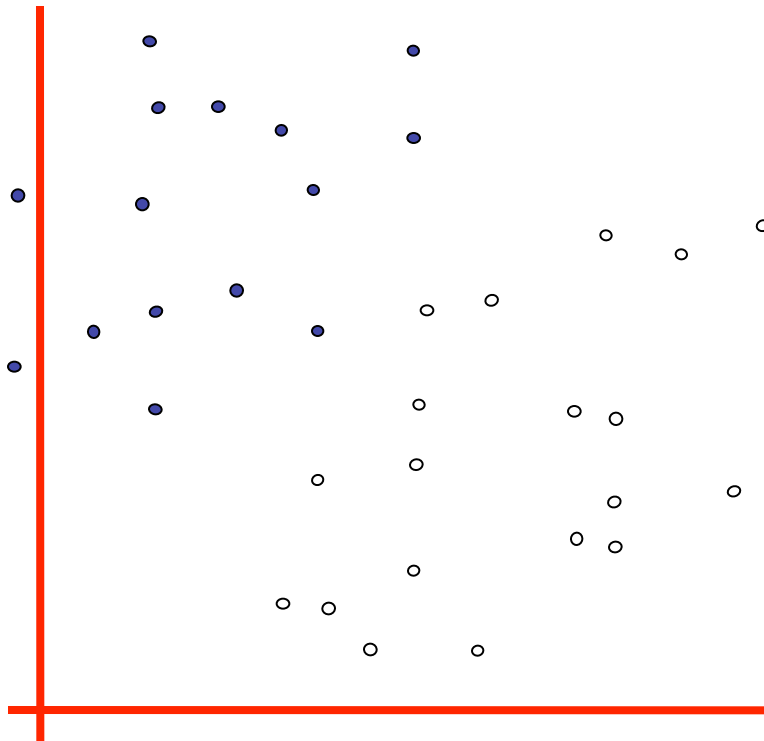
- Very popular ML technique
 - Became popular in the late 90s (Vapnik 1995; 1998)
 - Invented in the late 70s (Vapnik, 1979)
- Controls complexity and overfitting, so works well on a wide range of practical problems
- Can handle high dimensional vector spaces, which makes feature selection less critical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

Linear Classifiers



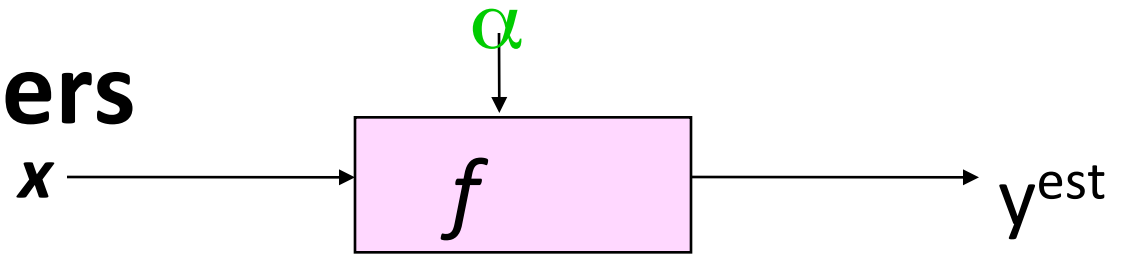
- denotes +1
- denotes -1

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

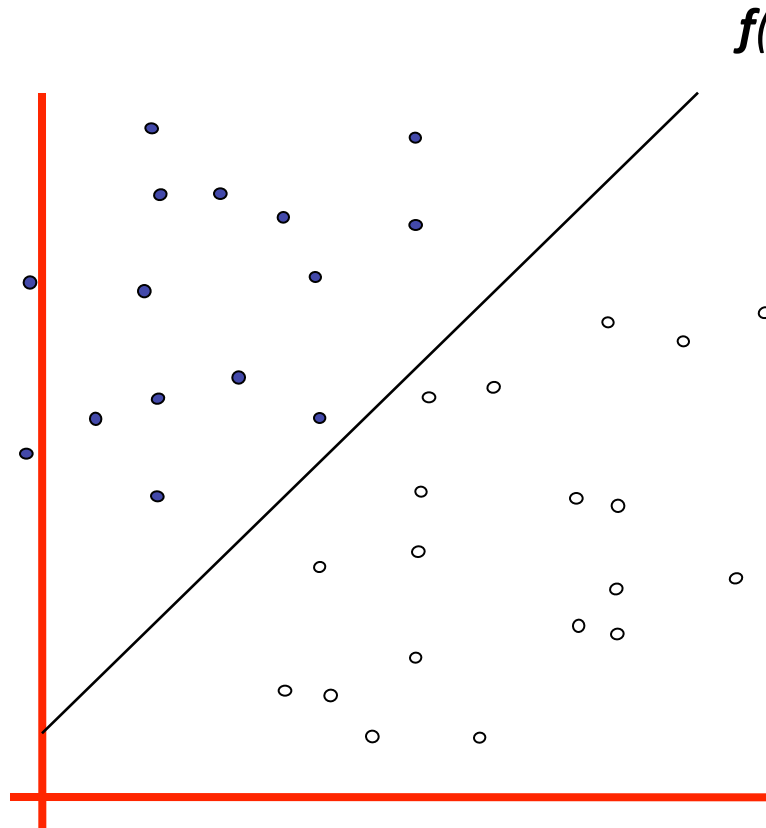


How would you classify this data?

Linear Classifiers

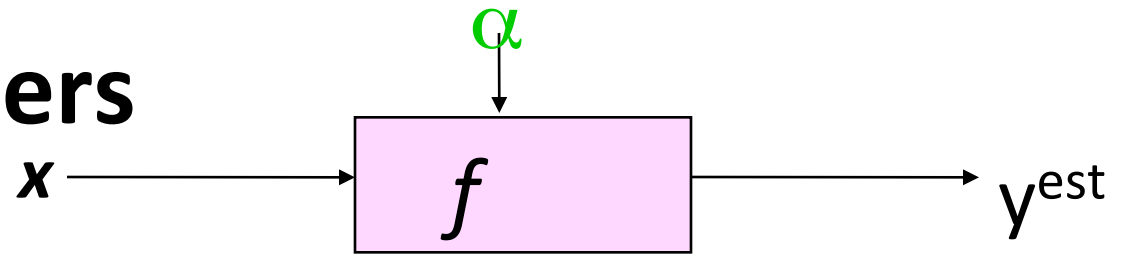


- denotes +1
- denotes -1



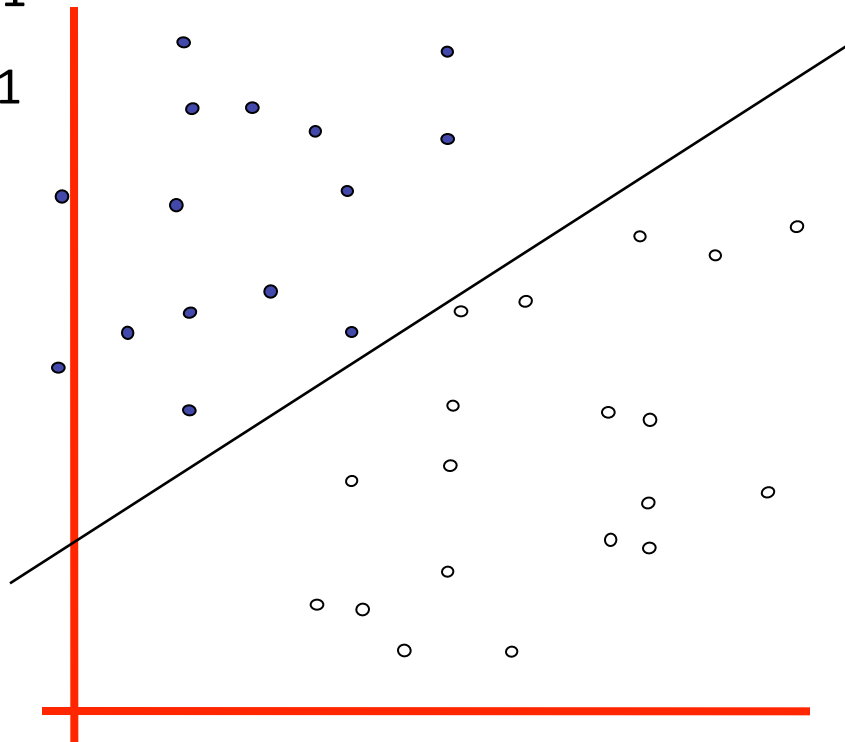
How would you classify this data?

Linear Classifiers



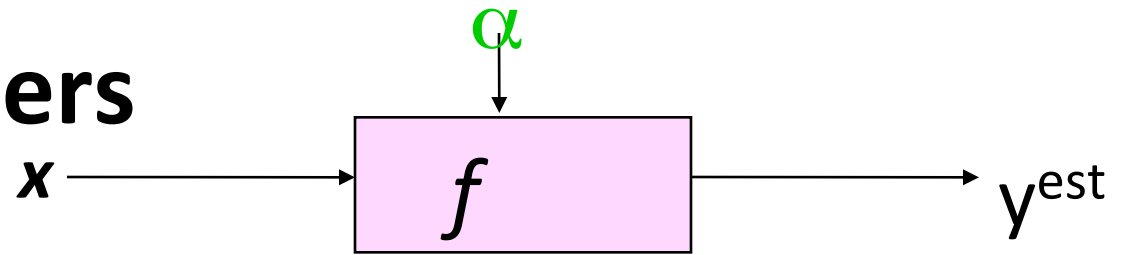
$$f(x, w, b) = \text{sign}(w \cdot x - b)$$

- denotes +1
- denotes -1

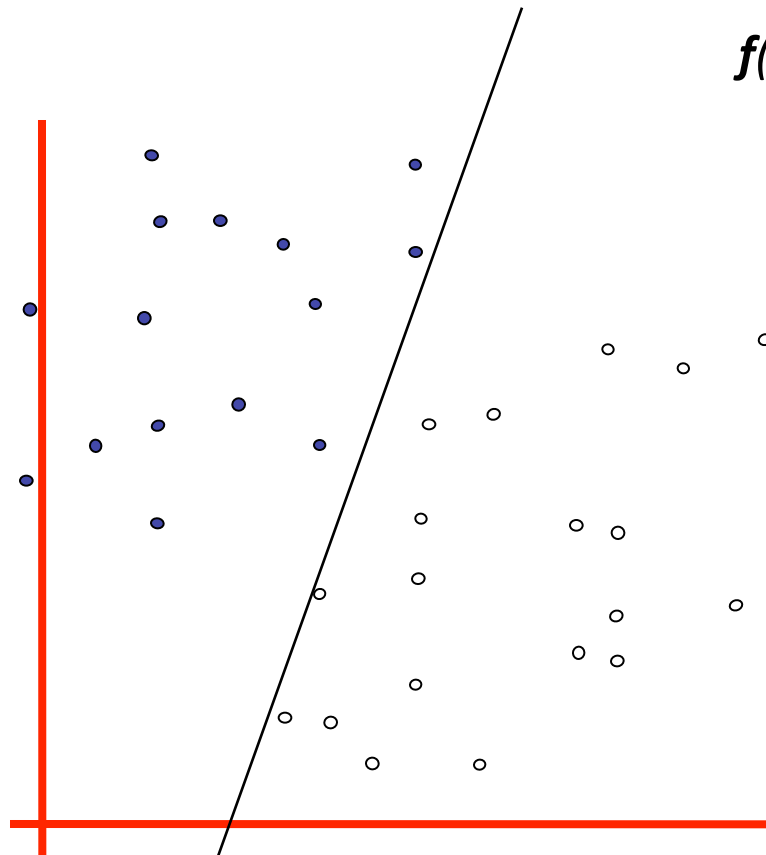


How would you classify this data?

Linear Classifiers



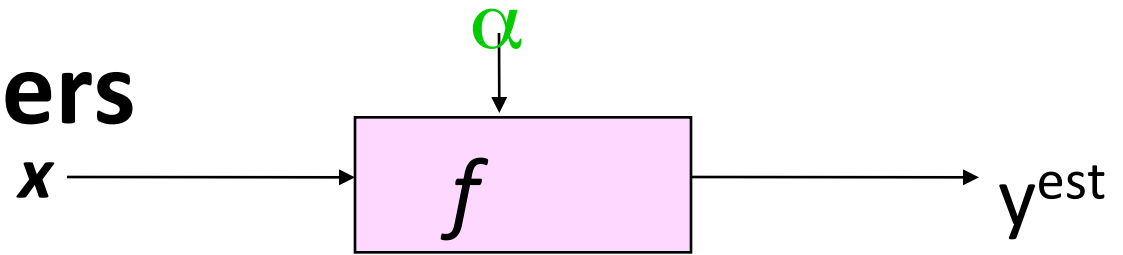
- denotes +1
- denotes -1



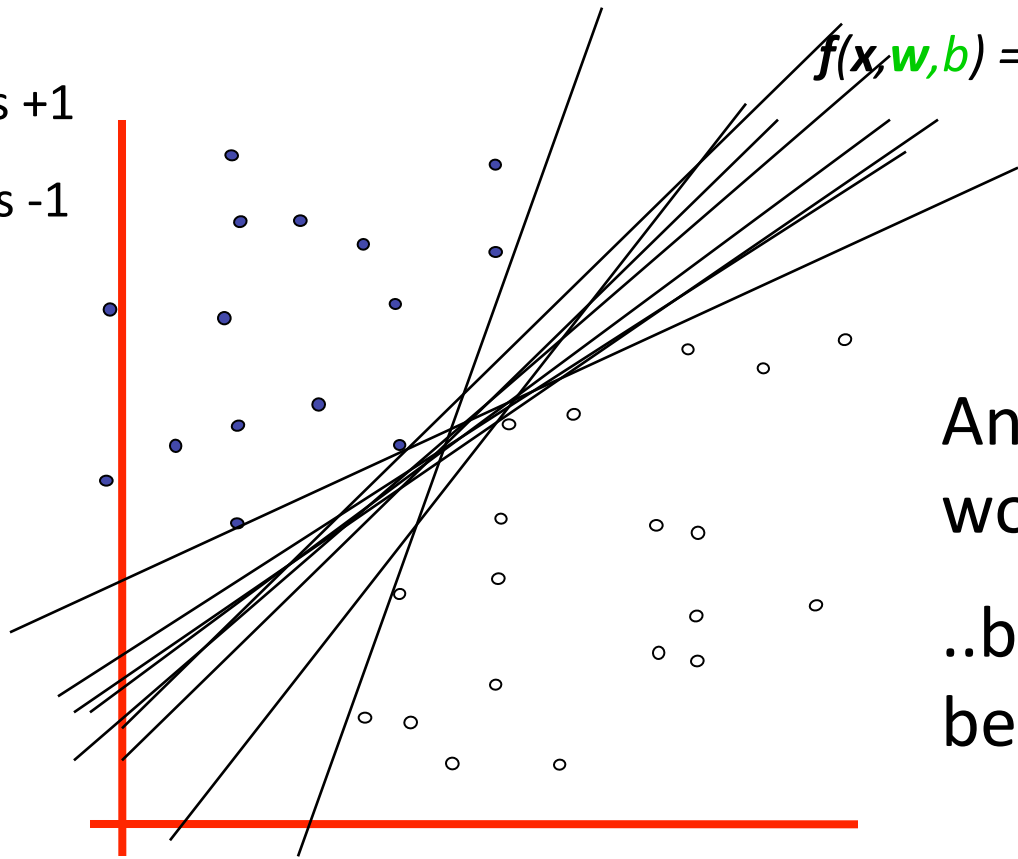
$$f(x, w, b) = \text{sign}(w \cdot x - b)$$

How would you classify this data?

Linear Classifiers

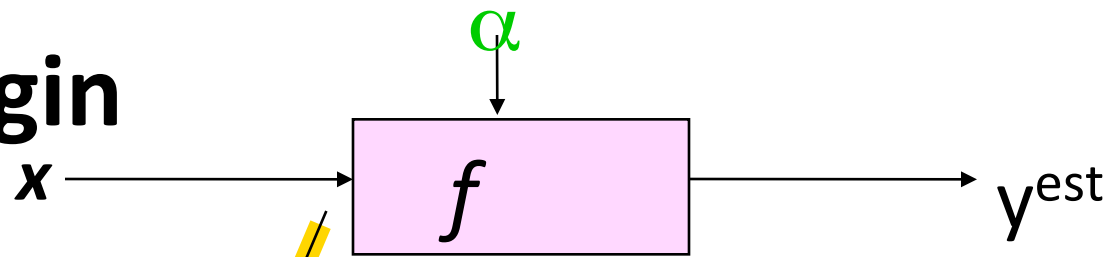


- denotes +1
- denotes -1



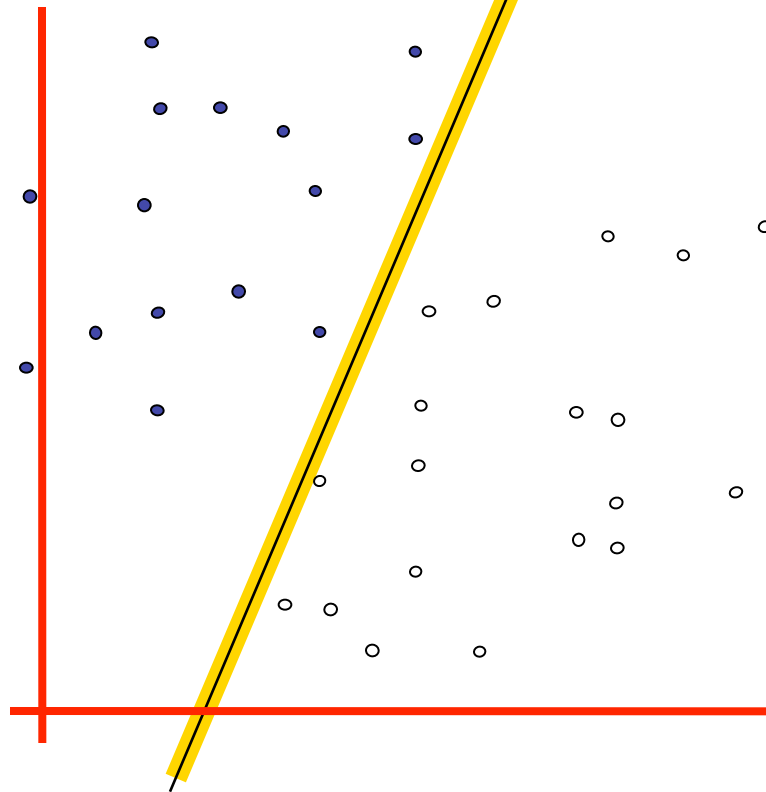
Any of these
would be fine..
..but which is
best?

Classifier Margin



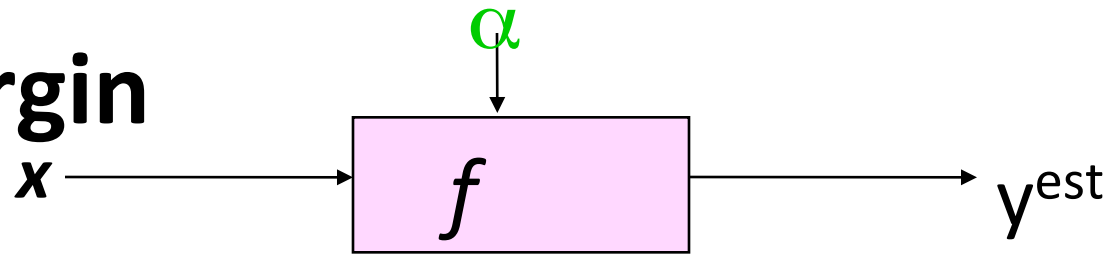
$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

- denotes +1
- denotes -1

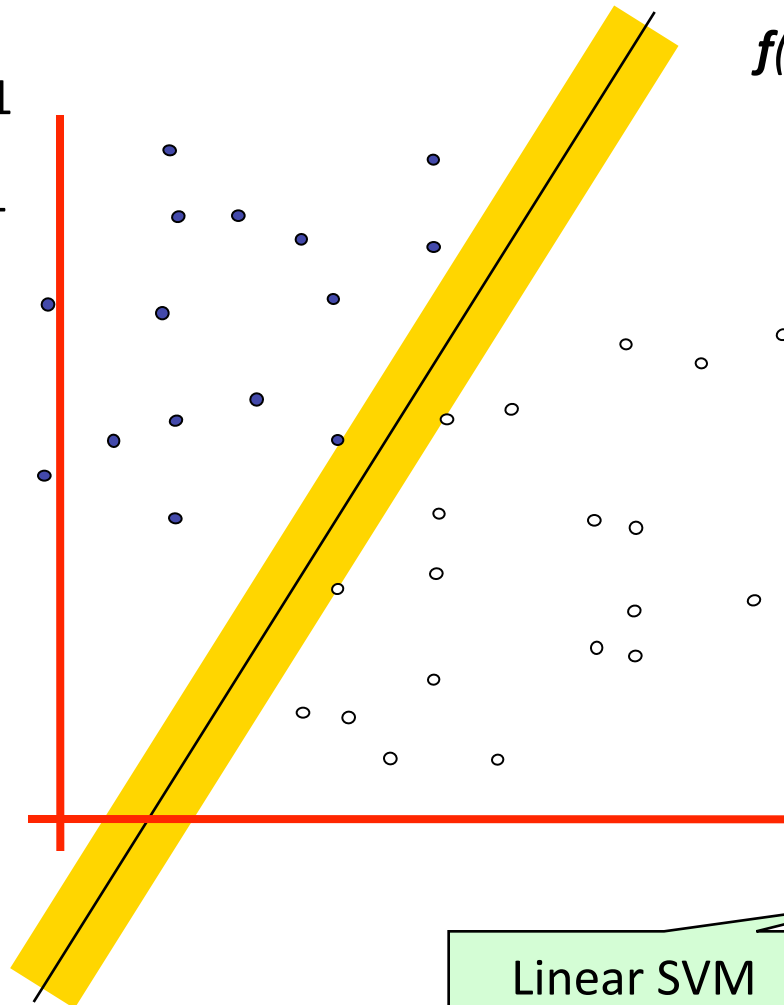


Define **margin** of linear classifier as width that boundary could be increased by before hitting a datapoint

Maximum Margin



- denotes +1
- denotes -1

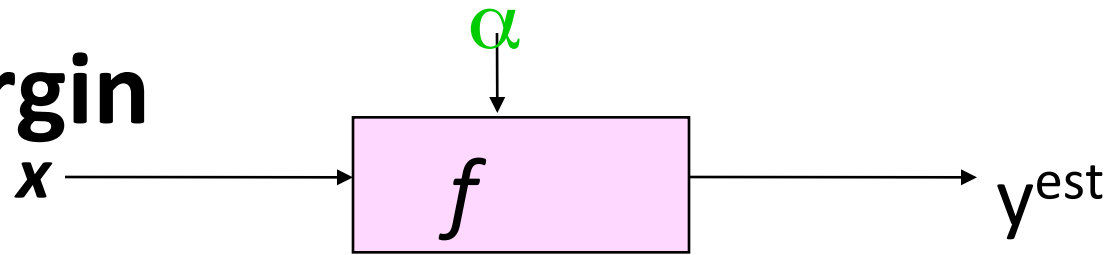


$$f(x, \mathbf{w}, b) = \text{sign}(\mathbf{w} \cdot \mathbf{x} - b)$$

The **maximum margin linear classifier** is the linear classifier with the, um, maximum margin

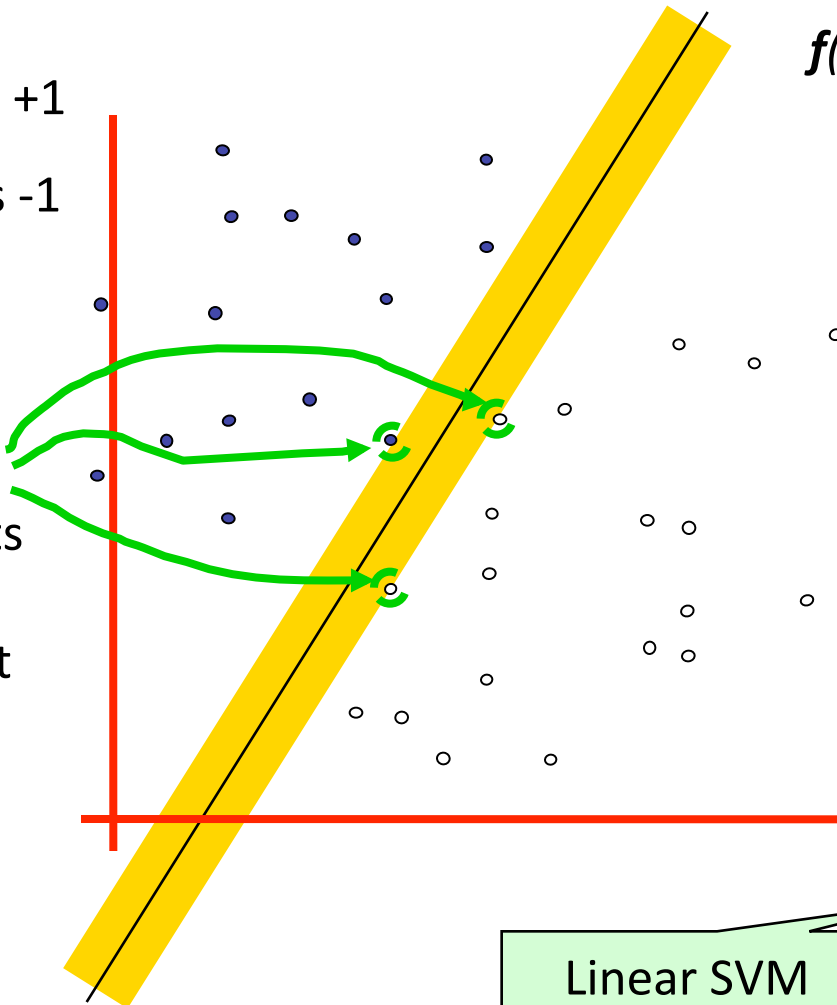
This is the simplest kind of SVM (Called an LSVM)

Maximum Margin



- denotes +1
- denotes -1

Support Vectors
are the datapoints
that margin
pushes up against



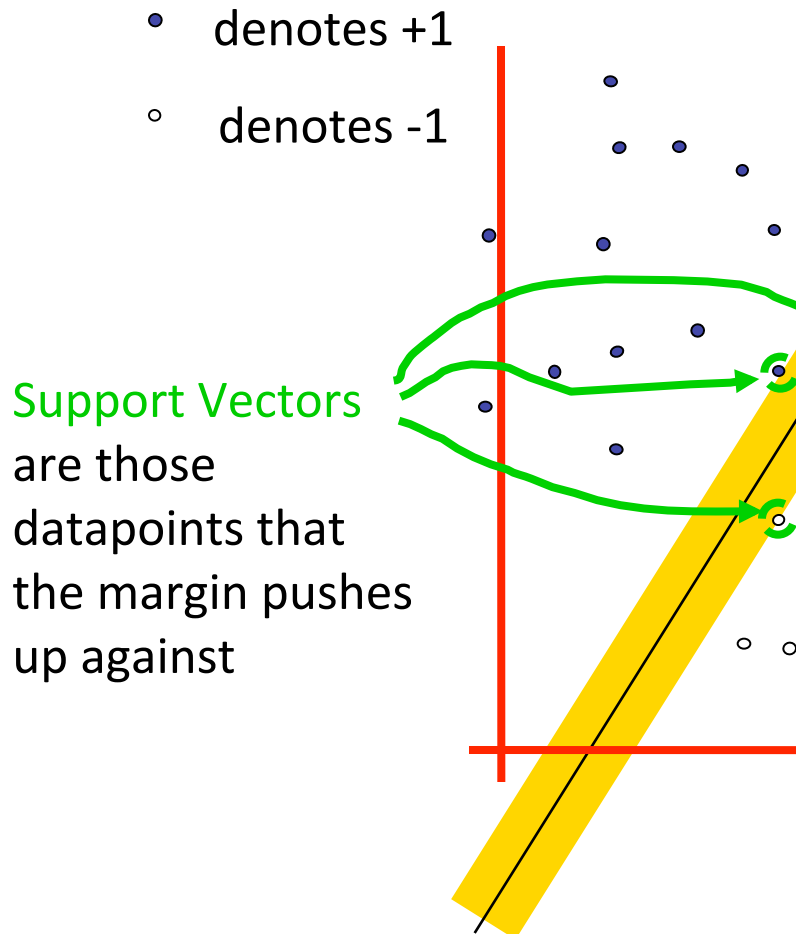
$$f(x, w, b) = \text{sign}(w \cdot x - b)$$

The **maximum margin linear classifier** is the linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear SVM

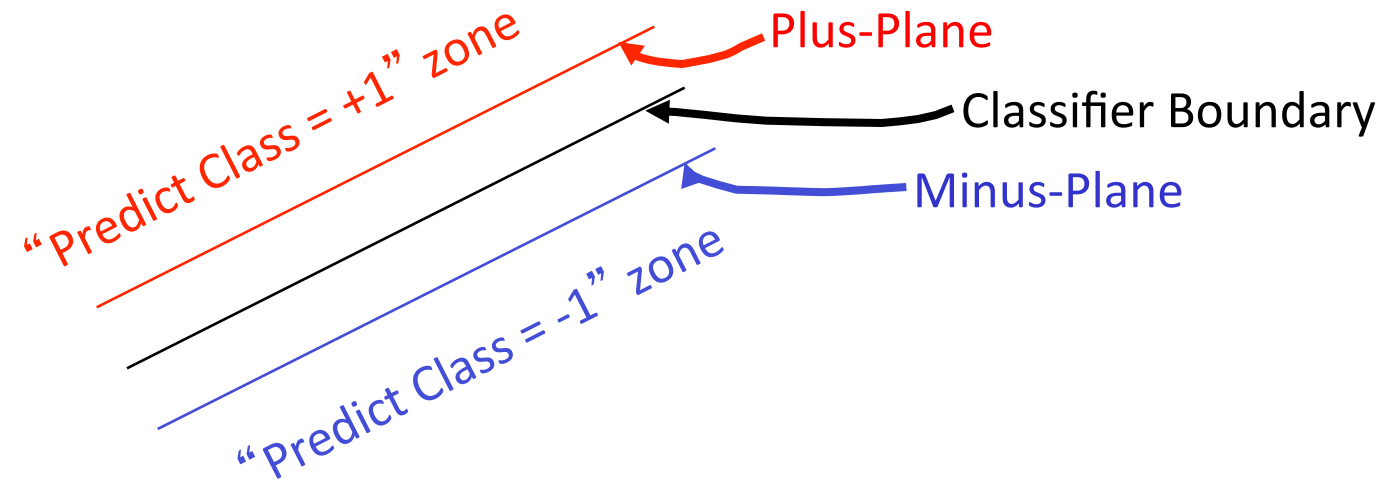
Why Maximum Margin?



1. Intuitively this feels safest
2. If we've made a small error in the location of the boundary (it's been jolted in its perpendicular direction) this gives us least chance of causing a misclassification
3. LOOCV is easy since the model is immune to removal of any 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

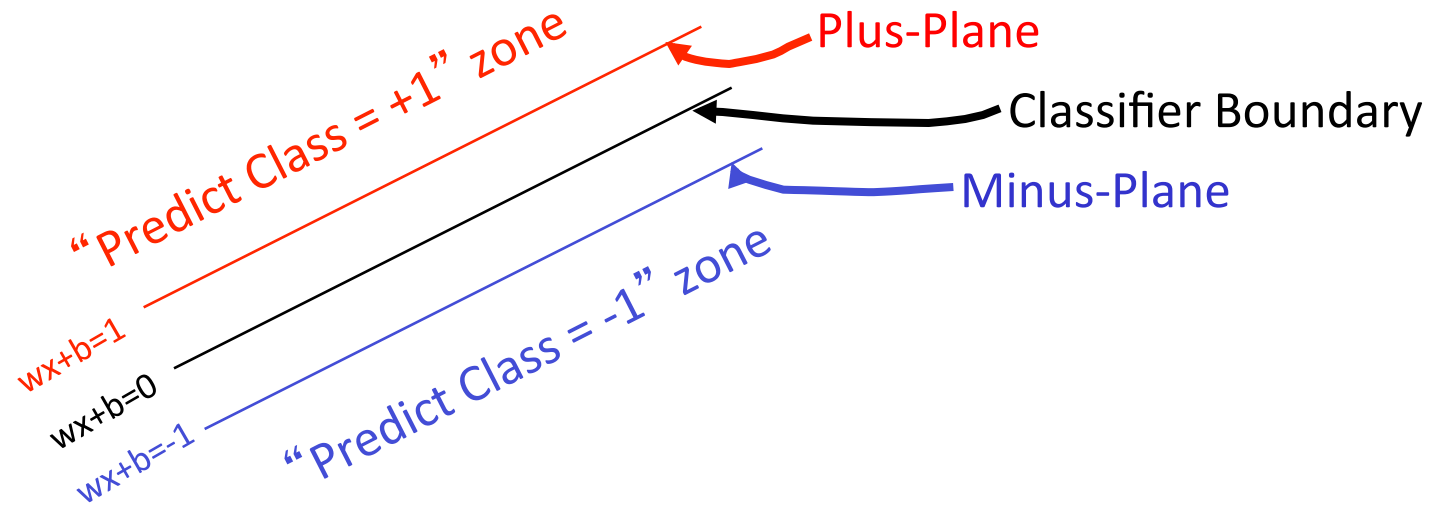
LOOCV = leave one out cross validation

Specifying a line and margin



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

Specifying a line and margin



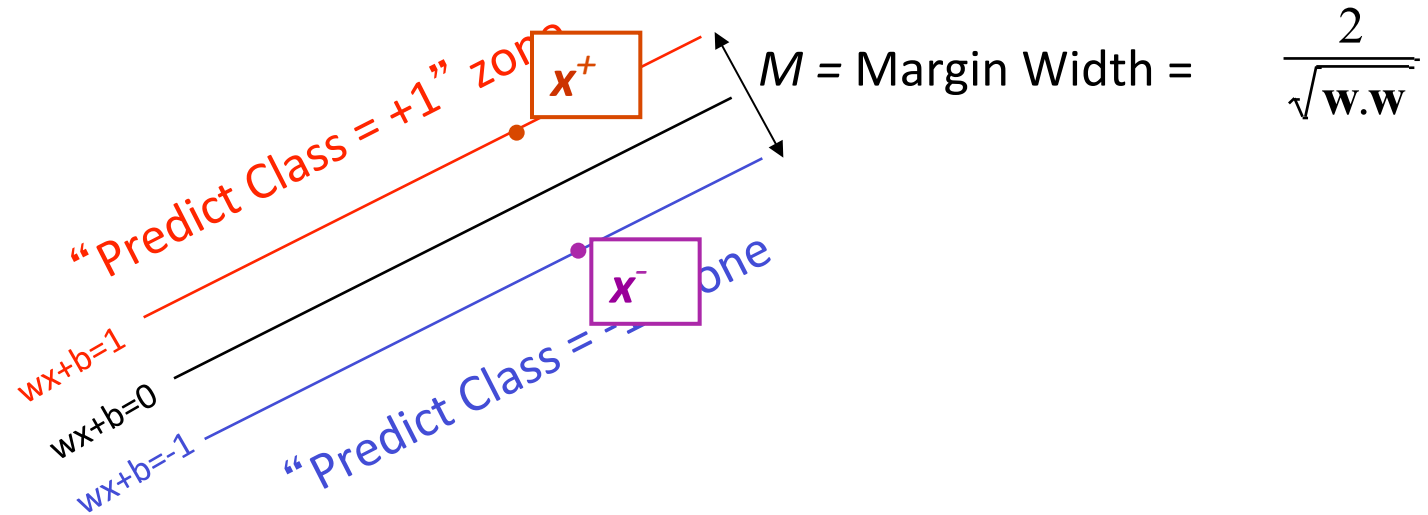
- Plus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = +1 \}$
- Minus-plane = $\{ \mathbf{x} : \mathbf{w} \cdot \mathbf{x} + b = -1 \}$

Classify as.. **+1** if **$\mathbf{w} \cdot \mathbf{x} + b \geq 1$**

-1 if **$\mathbf{w} \cdot \mathbf{x} + b \leq -1$**

Universe if **$-1 < \mathbf{w} \cdot \mathbf{x} + b < 1$**
explodes

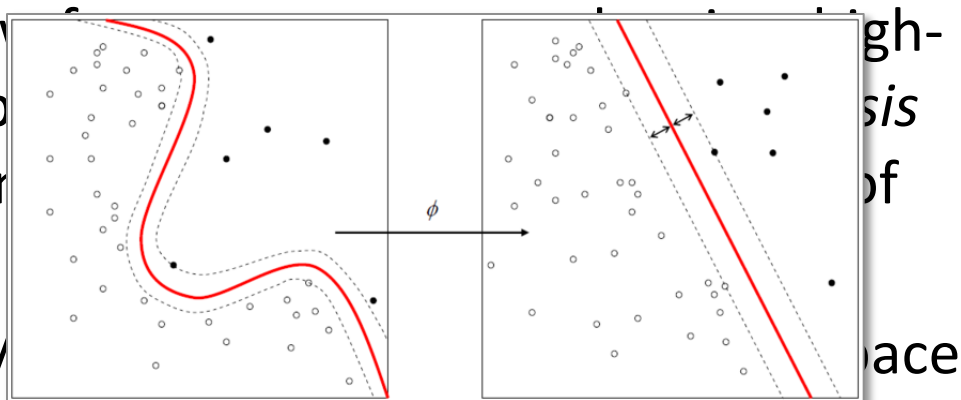
Learning the Maximum Margin Classifier



- Given a guess of w and b we can
 - Compute whether all data points in the correct half-planes
 - Compute the width of the margin
- Write a program to search the space of w s and b s to find widest margin matching all the datapoints.
- *How?* -- Gradient descent? Simulated Annealing? Matrix Inversion? EM? Newton's Method?

Learning SVMs

- Trick #1: Just find the points that would be closest to the optimal separating plane (“support vectors”) and work directly from those instances
- Trick #2: Represent as a **quadratic optimization problem**, and use quadratic programming techniques
- Trick #3 (“kernel trick”):
 - Instead of using the raw n -dimensional feature space, use *kernel functions* (e.g., polynomial) to map the base features
 - Find separating plane /
 - Voila: A nonlinear classifier!



SVM Performance

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

Binary vs. multi classification

- SVMs can only do binary classification
- Two approaches to multi classification:
 - One-vs-all: can turn an n-way classification into n binary classification tasks
 - E.g., for zoo problem, do mammal vs. not-mammal, fish vs. not-fish, ...
 - Pick one that results in the highest score
 - $N*(N-1)/2$ One-vs-one classifiers that vote on results
 - Mammal vs. fish, mammal vs. reptile, etc...

Feature Engineering

- Finding features for data instances that make machine learning algorithms more effective
- Usually a combination of domain knowledge and experimentation
- For example, consider the problem of classifying an email message as spam or not
- What are good features?

Spam Features

- Finding features for data instances that make machine learning algorithms more effective
- Usually a combination of domain knowledge and experimentation
- For example, consider the problem of classifying an email message as spam or not
- What are good features?

Example of a Spam Message

Received: from nm37-vm8.bullet.mail.gq1.yahoo.com (nm37-vm8.bullet.mail.gq1.yahoo.com [98.136.217.44])

Date: Mon, 2 May 2016 15:26:36 +0000 (UTC)

From: "Mrs.Sabeen Gharam" <SabeenGharam1@outlook.com>

Reply-To: "Mrs.Sabeen Gharam" <aramsabeen1@gmail.com>

Subject: CAN YOU HANDLE INVESTMENT?

Â CAN YOU HANDLE INVESTMENT?

My name is Mrs. Sabeen Gharam Aram widow of late Mohamed Assad Aram from idlib in syria arab republic, am in an urgent need of an individual or organization abroad that is willing to help to secure, move and invest my late husband's fund which he has offshore; the investment will depend on the agreement with me.

Mrs.Sabeen Gharam Aram

Possible Features

- Each of the words in the message
- Is there a TO: header?
- Does FROM: match REPLY-TO:?
- Is FROM: address a free email service?
- Is the subject in all caps?
- Length of message in $\log(\#bytes)$
- Are there attachments?
- Extension of any attached file (e.g., zip, pdf, doc, ..)
- TLD of FROM: address (e.g., .com, .edu, .org, ...)
- ... dozens more ...

Evaluating Features

- Choose some features, generate training data, train system, evaluation using 10-fold cross validation
- Some systems can show you the features that contribute the most to classification
- If not, perform *ablation* experiments by dropping a feature and re-training
- And try adding a new feature and training
- You can also automate this to try adding/removing various subsets of features
- Rely on your own knowledge and intuition also

Feature Engineering for Text Classification

- Typical features: words and/or phrases along with term frequency or (better) TF-IDF scores
- Δ TFIDF amplifies the training set signals by using the ratio of the IDF for the negative and positive collections
- Results in a significant boost in accuracy



Text: The quick brown fox jumped over the lazy white dog.

Features: the 2, quick 1, brown 1, fox 1, jumped 1, over 1, lazy 1, white 1, dog 1, the quick 1, quick brown 1, brown fox 1, fox jumped 1, jumped over 1, over the 1, lazy white 1, white dog 1

Δ TFIDF BoW Feature Set

- Value of feature t in document d is $C_{t,d} * \log_2\left(\frac{N_t}{P_t}\right)$
- Where
 - $C_{t,d}$ = count of term t in document d
 - N_t = number of negative labeled training docs with term t
 - P_t = number of positive labeled training docs with term t
- Normalize to avoid bias towards longer documents
- Gives greater weight to rare (significant) words
- Downplays very common words
- Similar to Unigram + Bigram BoW in other aspects

Example: Δ TFIDF vs TFIDF vs TF



15 features with highest values for a review of *City of Angels*

Δ tfidf

, city
cage is
mediocrity
criticized
exhilarating
well worth
out well
should know
really enjoyed
maggie ,
it's nice
is beautifully
wonderfully
of angels
Underneath the

tfidf

angels
angels is
, city
of angels
maggie ,
city of
maggie
angel who
movie goes
cage is
seth ,
goers
angels ,
us with
city

tf

,
the
.
to
of
a
and
is
that
it
who
in
more
you
but

Improvement over TFIDF (Uni- + Bi-grams)

- **Movie Reviews:** 88.1% accuracy vs. 84.6%
- **Subjectivity Detection** (Opinionated or not): 91.3% vs. 89.4%
- **Congressional Support for Bill** (Voted for/against): 72.5% vs. 66.8%
- **Enron Email Spam Detection:** (Spam or not): 98.9% vs. 96.6%
- All tests used 10 fold cross validation

Justin Martineau and Tim Finin, [Delta TFIDF: An Improved Feature Space for Sentiment Analysis](#), 3rd Conf. on Weblogs and Social Media, AAAI Press, May 2009.