# Support Vector Machine

Source:
Hastie, Tibshirani and Friedman, 2009
C. Manning et al, Introduction to Information Retrieval, ch. 15
Slides from A. Moore, K. McKeown, T. Lane, C. Manning.

# Brief introduction of linear algebra concepts: Dot product

A *dot product* (inner product): is the p*rojection of one vector onto another*

Calculation: Multiplying together the corresponding components and adding up the products:

$(1, 2, 3, 4) \cdot (5, 6, 7, 8) = 1\times5 + 2\times6 + 3\times7 + 4\times8 = 5 + 12 + 21 + 32 = 70.$

$\mathbf{x} \cdot \mathbf{y} = (x_1, x_2, x_3, x_4) \,.\, (y_1, y_2, y_3, y_4) = x_1y_2 + x_2y_2 + x_3y_3 + x_4y_4$

Two vectors are said to be *orthogonal* if their inner product is 0:

$(1, 1, 1, 1) \cdot (1, –1, 1, –1) = 1 + –1 + 1 + –1 = 0$

*Length* of a vector **x:** the square root of its inner product with itself:

$||\mathbf{x}|| = \sqrt{(\mathbf{x} \cdot \mathbf{x})}$.

Example: $||(1, –1, 1, –1)|| = \sqrt{(1\times1 + (–1\times–1) + 1\times1 + (–1\times–1))} = \sqrt{4} = 2$.

*Normal vector:* if it has length 1.

Example: (0.5, –0.5, 0.5, –0.5) is normal.

If **x** is any vector, then $\mathbf{x}/||\mathbf{x}||$ is normal, and the process of converting **x** into $\mathbf{x}/||\mathbf{x}||$ is called normalization. A normalized vector $\mathbf{x}/||\mathbf{x}||$ is the unit vector (length 1) codirectional with **x.**

# Hyperplanes

In an *n*-dimensional space, (i.e. a space of vectors with *n* components), a hyperplane is a (*n*–1)-dimensional flat subset that separates the space into two half spaces.

$$\mathbf{a} \bullet \mathbf{x} = b$$

where $\mathbf{a} = (a_1, a_2, ..., a_n)$ and $\mathbf{x} = (x_1, x_2, ..., x_n)$.

In 3-space, the hyperplane is the set of all points $(x, y, z)$ such that $ax + cy + dz = b$ holds.

where $x$, $y$, and $z$ are variables and $a$, $b$, $c$ and $d$ are constants.

or $(a, c, d) \bullet (x, y, z) = b$

If we write **a** for (a, c, d) and **x** for (x, y, z), the equation becomes $\mathbf{a} \bullet \mathbf{x} = b$

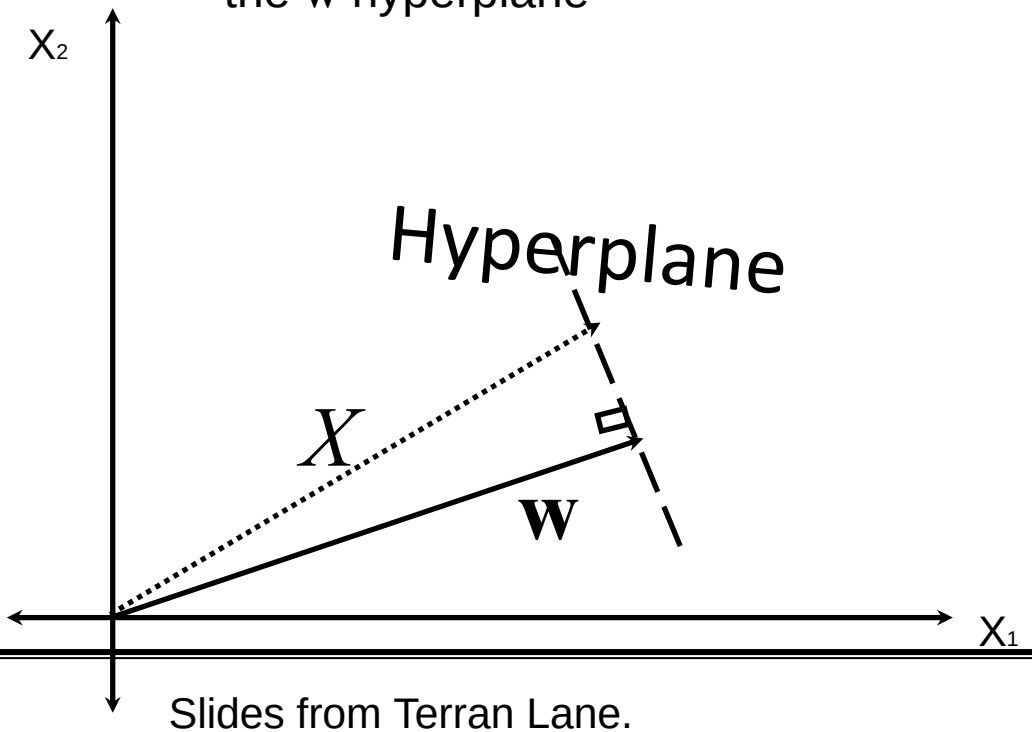
For a point **x** in *n*-space, the point lies on one "side" of the hyperplane if $\mathbf{a} \bullet \mathbf{x} - b$ is negative, and on the other side if $\mathbf{a} \bullet \mathbf{x} - b$ is positive

# Hyperplanes

Given a hyperplane defined by a weight vector

**W =** [b**,** $w_1$, $w_2$,…, $w_d$]$^T$
where b = $w_0$

- A *dot product* (inner product) is a *projection of one vector onto another*
  - When the projection of **x** onto **w** is equal to $w_0$, then **x** falls exactly onto the **w** hyperplane




$$b + \sum_{i=1}^{d} w_i x_i = 0$$

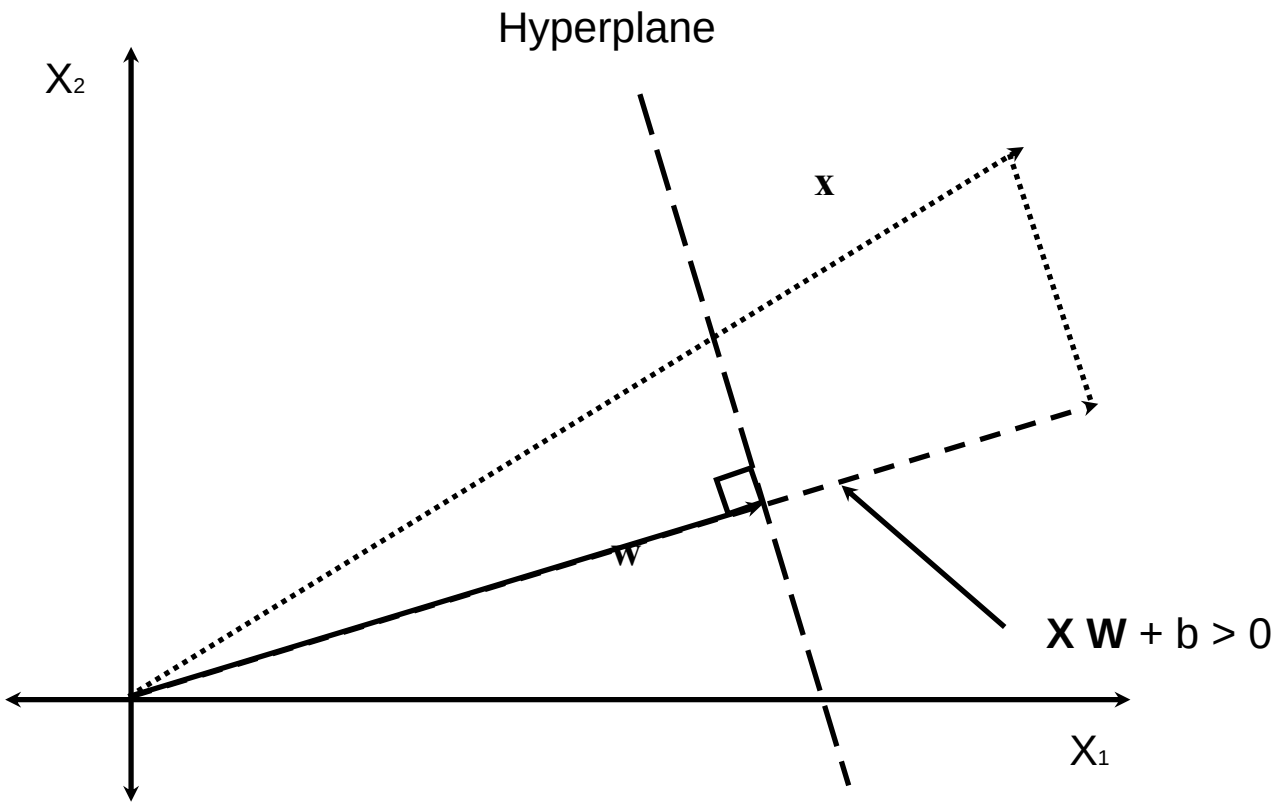
**X W** + b =0

**X W = -b**

Slides from Terran Lane.

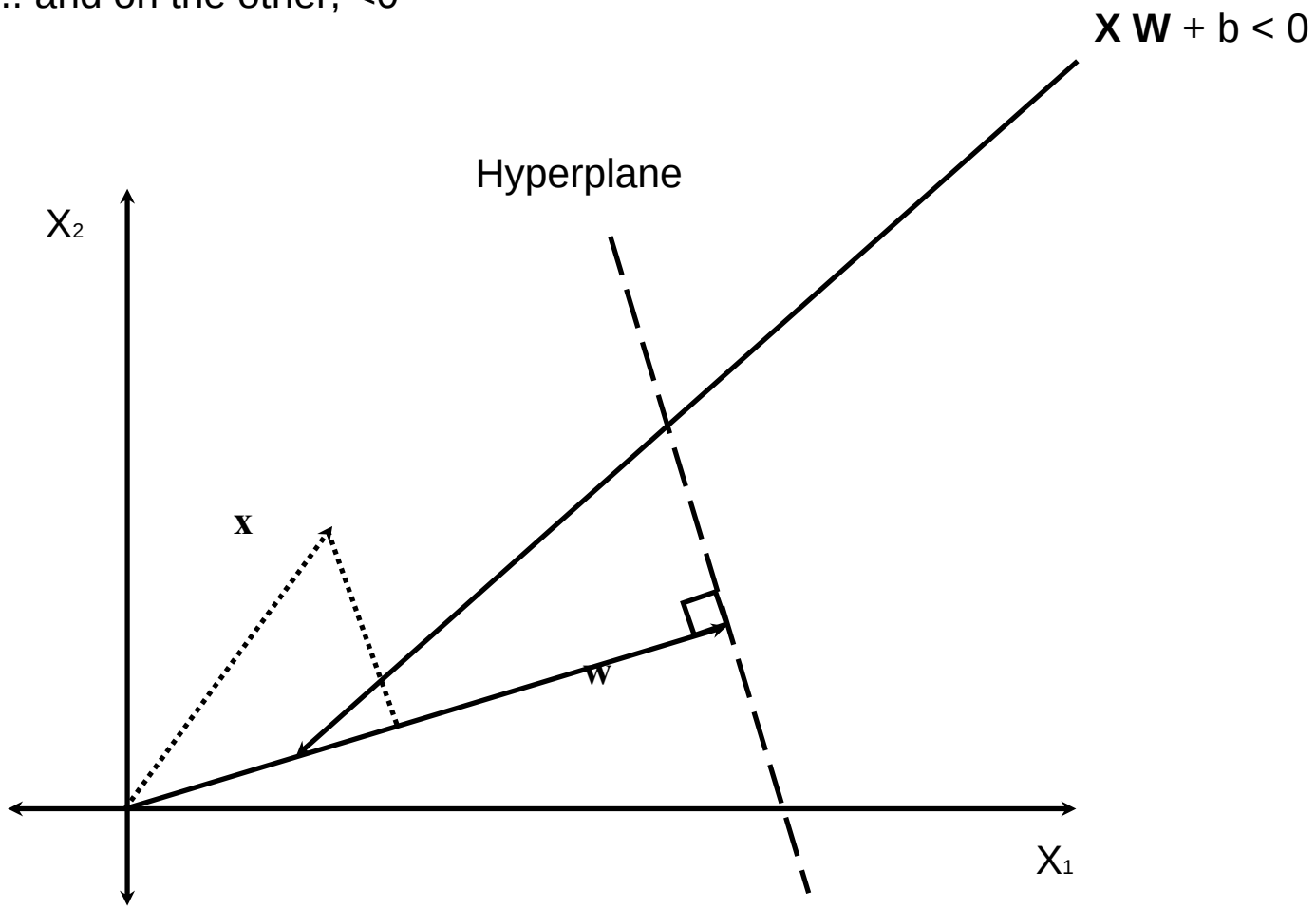
# Hyperplanes

- Projections on one side of the line (hyperplane) have dot products >0...



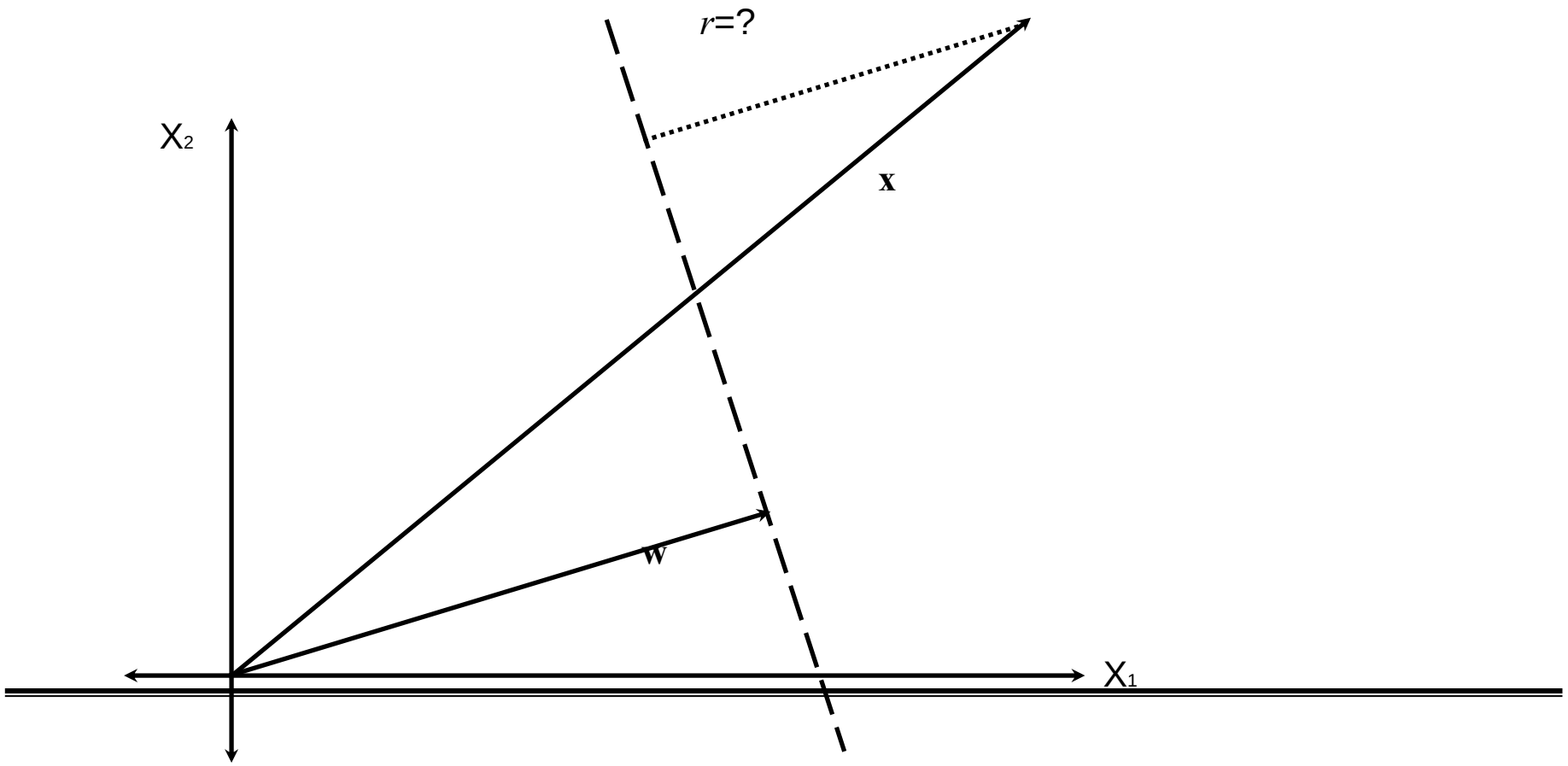
# Hyperplanes

- Projections on one side of the line have dot products >0...
- ... and on the other, <0



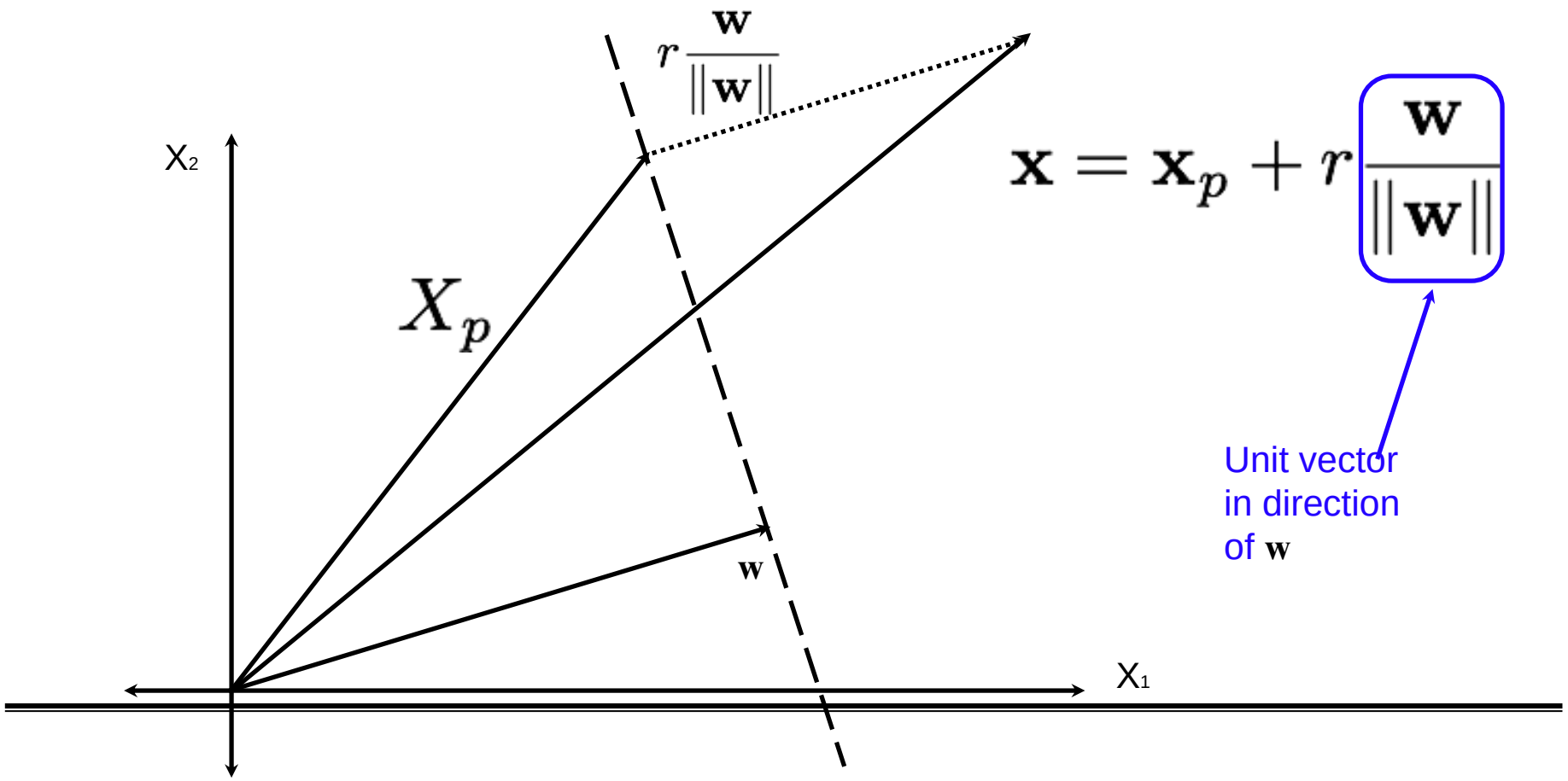
# Hyperplanes

- What is the distance from any vector **x** to the hyperplane?



# Hyperplanes

- What is the distance from any vector $\mathbf{x}$ to the hyperplane?
- Write $\mathbf{x}$ as a point on plane + offset from plane



# Hyperplanes

- Now:

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

$$\Rightarrow \quad \mathbf{w}\cdot\mathbf{x} = \mathbf{w}\cdot\mathbf{x}_p + \mathbf{w}\cdot\left(r\frac{\mathbf{w}}{\|\mathbf{w}\|}\right)$$

Hint: $||\mathbf{x}||=\sqrt{(\mathbf{x}\bullet\mathbf{x})}$.

$\mathbf{X}\,\mathbf{W} = -b$

$= -b + r||\mathbf{W}||$

$$r = \frac{\mathbf{W}\,\mathbf{X} + b}{||\mathbf{W}||}$$

# Hyperplanes

- *Theorem*: The distance, $r$, from any point $\mathbf{x}$ to the hyperplane defined by $\mathbf{w}$ and b is given by:

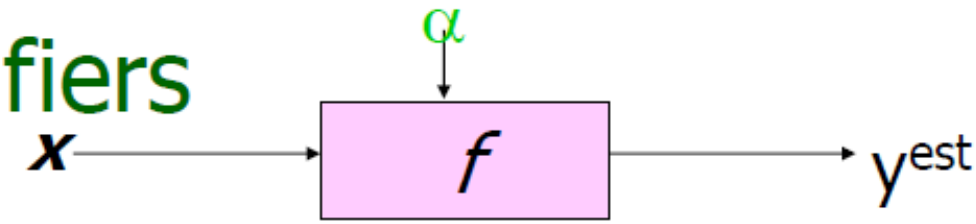
$$r = \frac{\mathbf{W}\,\mathbf{X} + b}{||\mathbf{W}||}$$

- *Lemma*: The distance from the origin to the hyperplane is given by:

$$r = \frac{b}{||\mathbf{W}||}$$

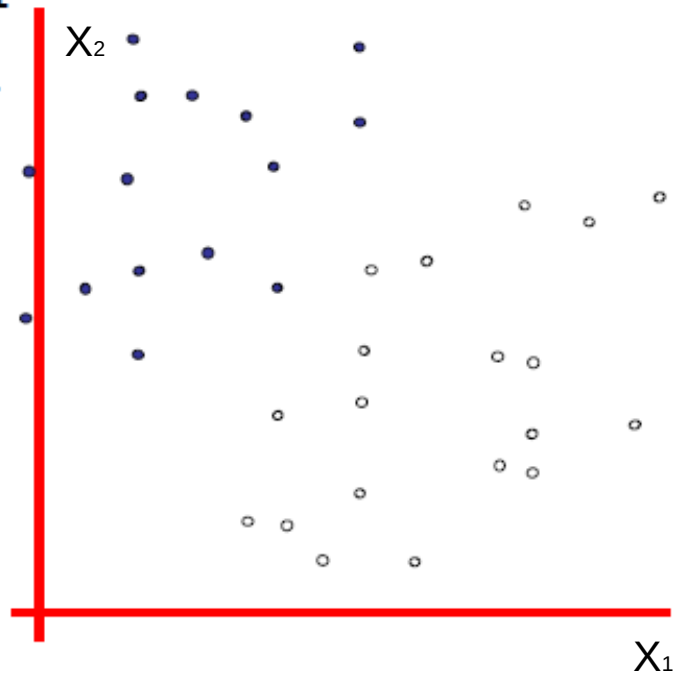
- Also: $r>0$ for points on one side of the hyperplane; $r<0$ for points on the other

# Linear Classifiers



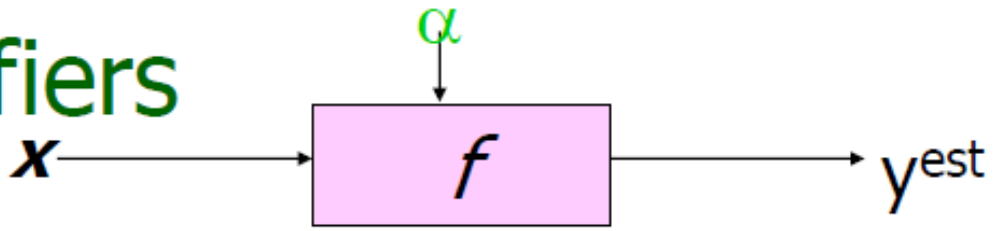

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\mathbf{x} - b)$

- denotes +1
- denotes -1



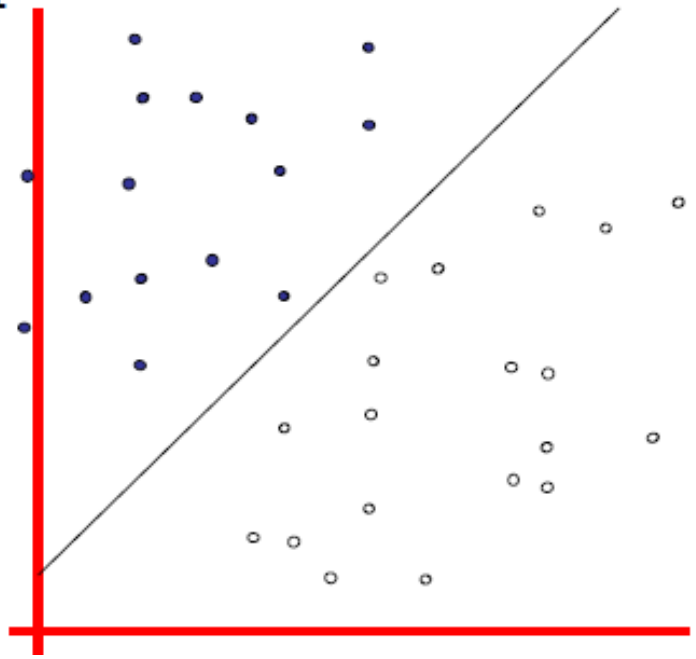

How would you classify this data?

# Linear Classifiers



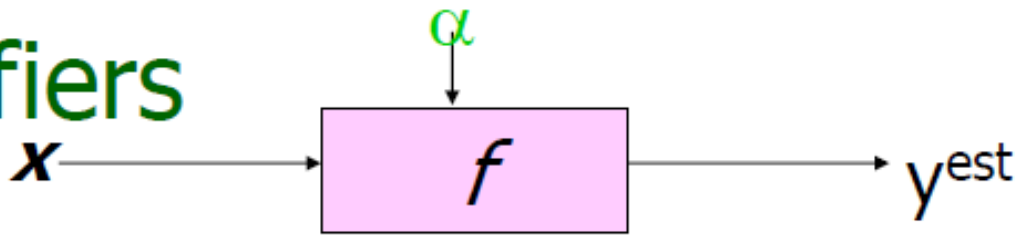

$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}. \mathbf{x} - b)$

- denotes +1
- denotes -1



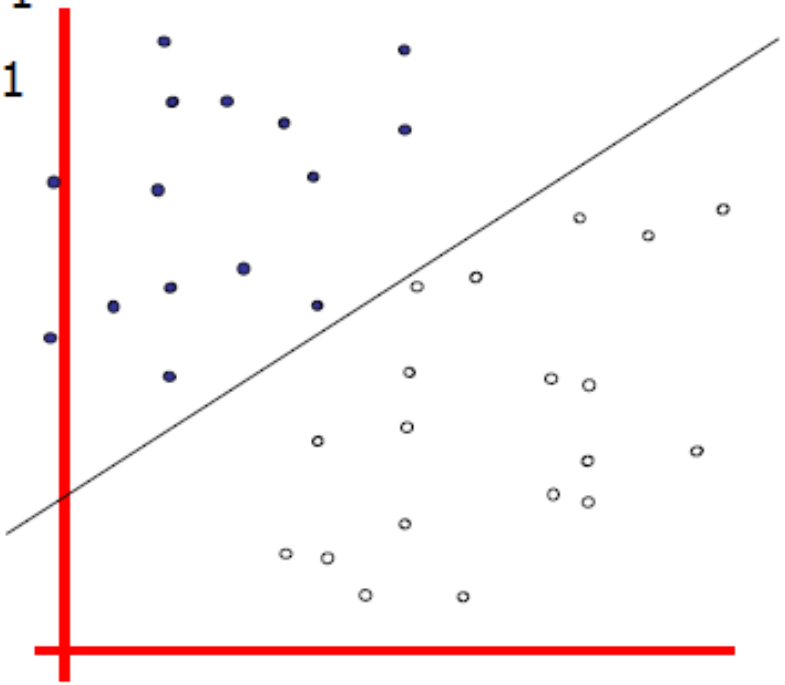
How would you classify this data?

# Linear Classifiers



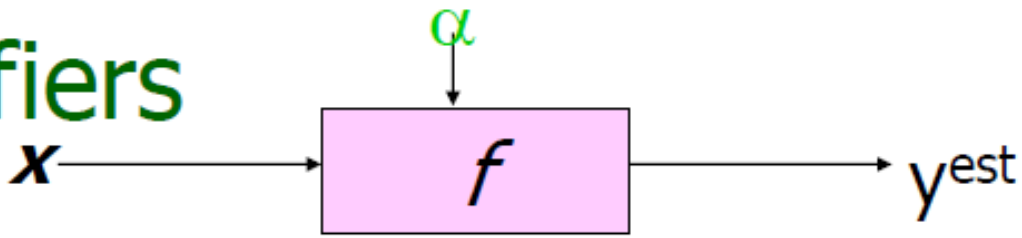

$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}. \mathbf{x} - b)$

- denotes +1
- denotes -1



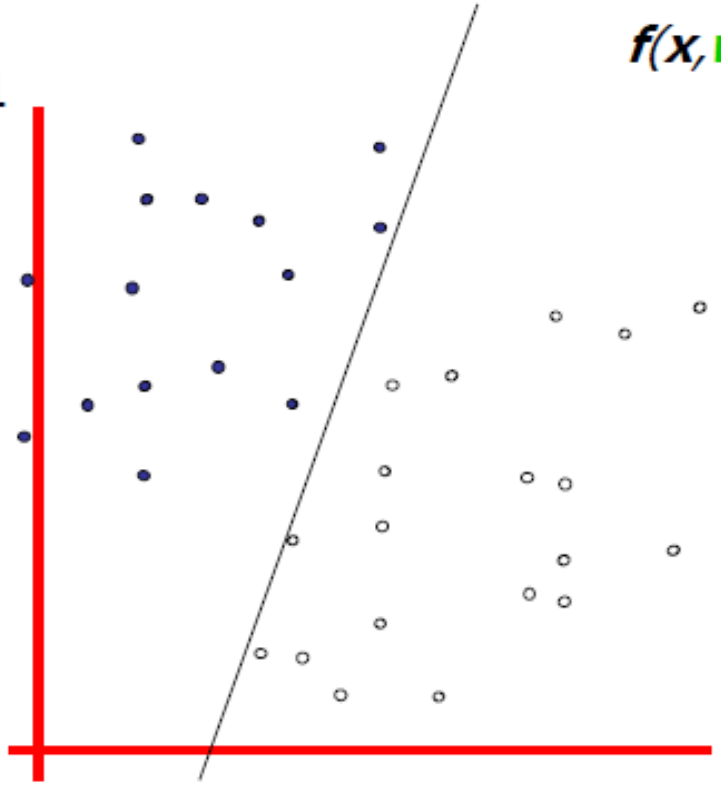
How would you classify this data?

# Linear Classifiers



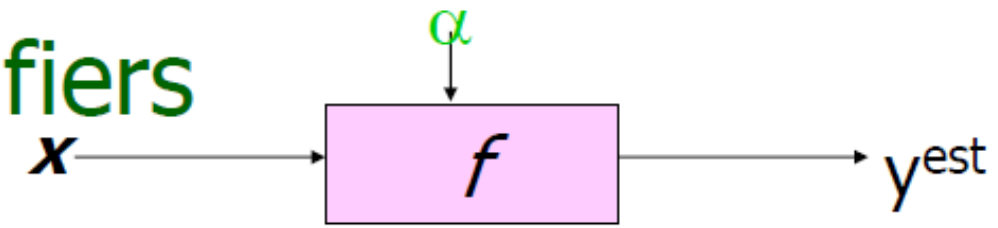

$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

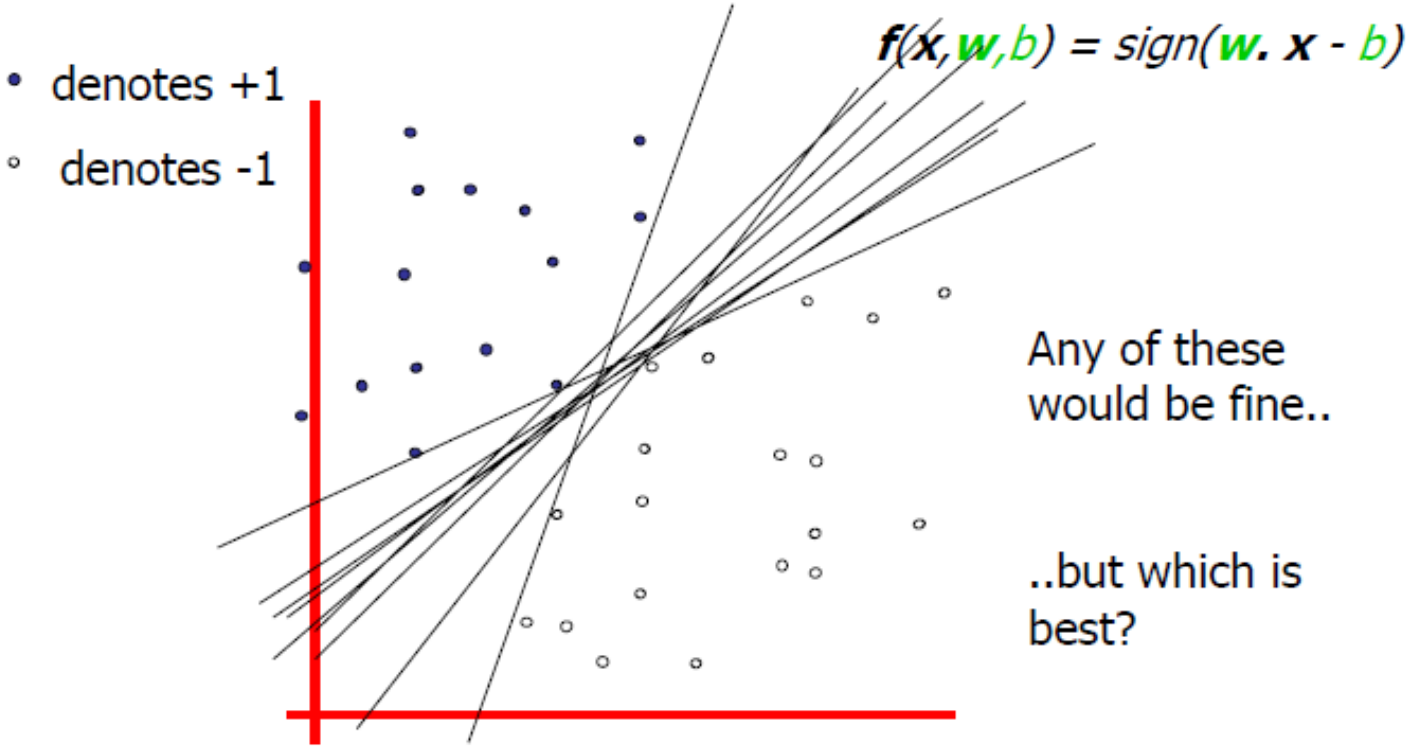
- denotes +1
- denotes -1



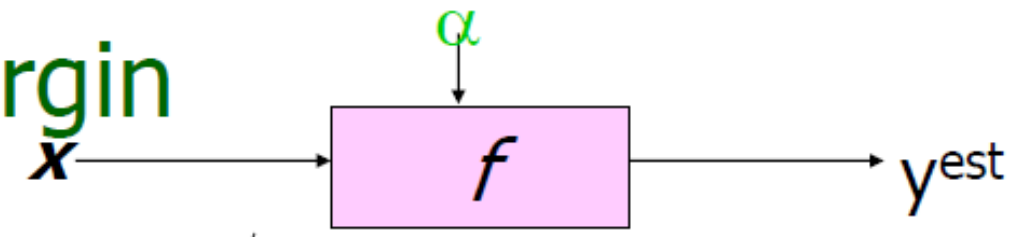
How would you classify this data?

# Linear Classifiers

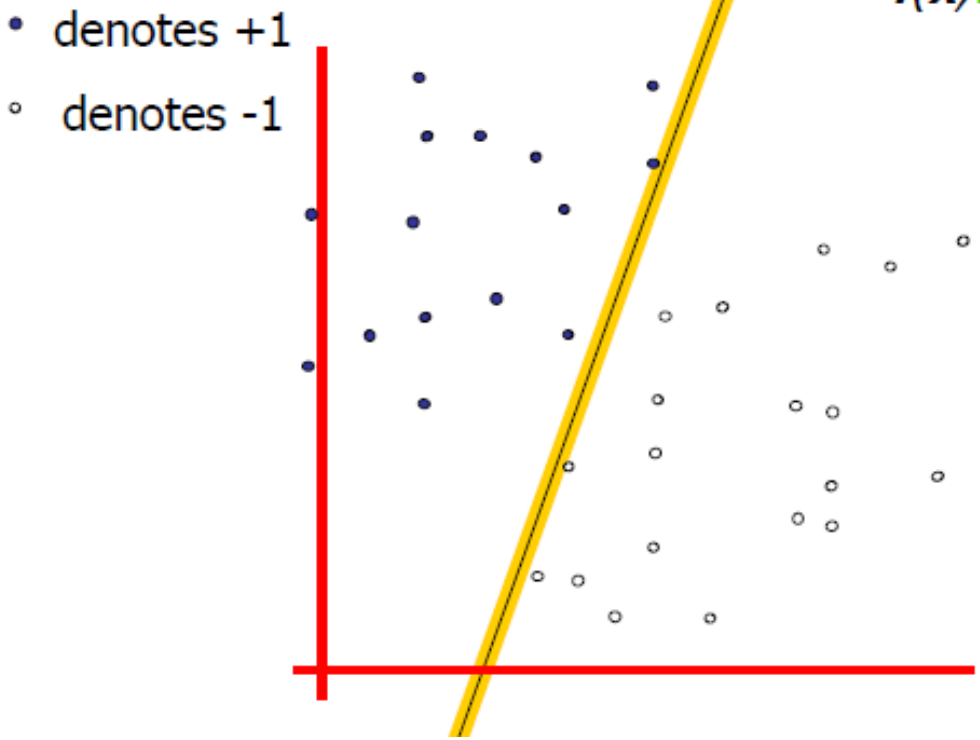


# Classifier Margin

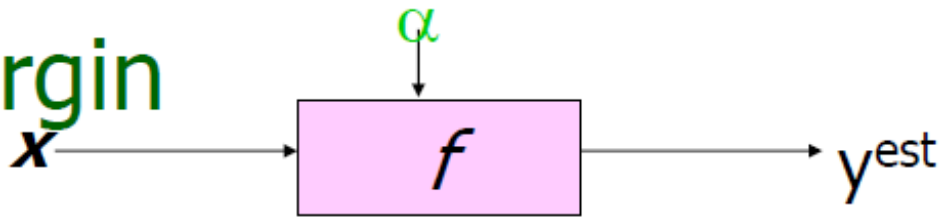



$f(x,w,b) = sign(w.x - b)$



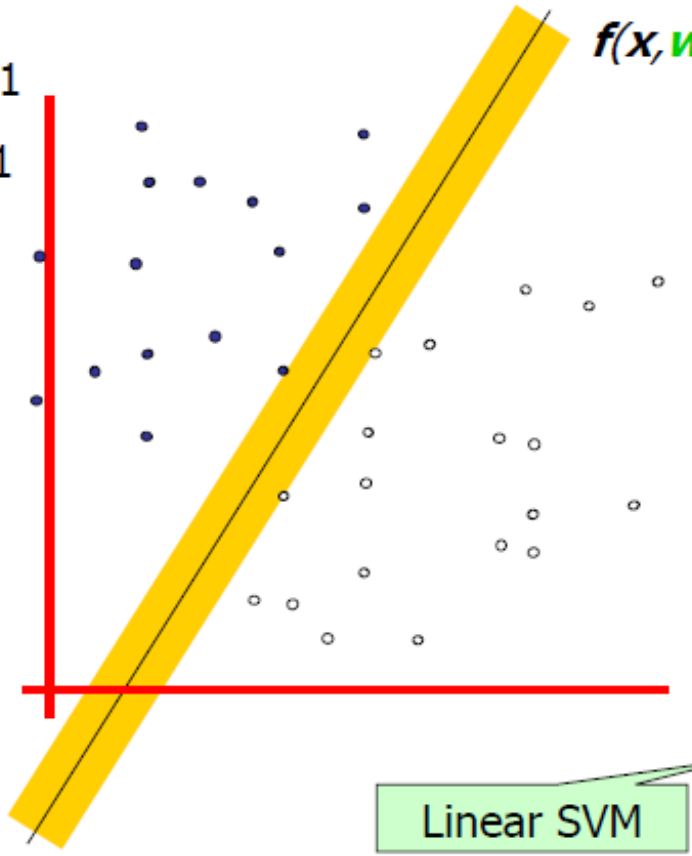

Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

# Maximum Margin




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

- denotes +1
- denotes -1

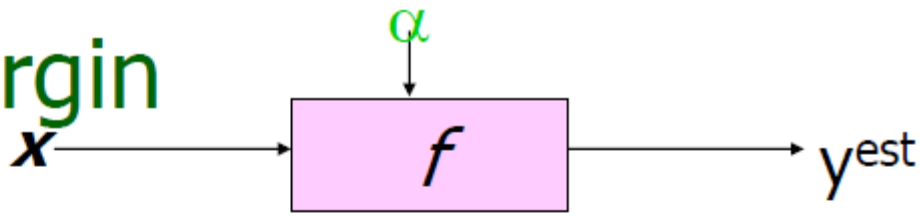


The maximum margin linear classifier is the linear classifier with the, um, maximum margin.

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

Linear SVM

# Maximum Margin




$f(x,w,b) = sign(w.x - b)$

- denotes +1
- denotes -1

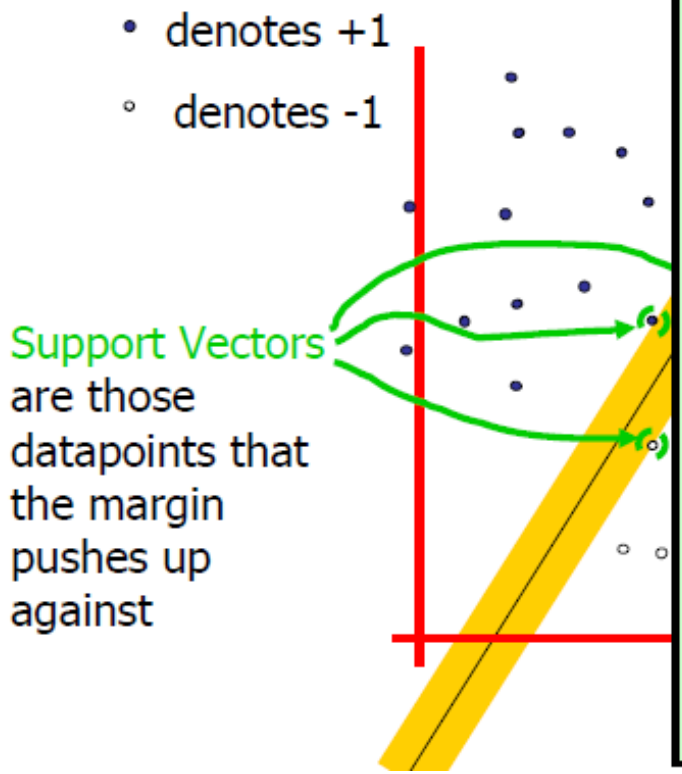
Support Vectors are those datapoints that the margin pushes up against

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

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

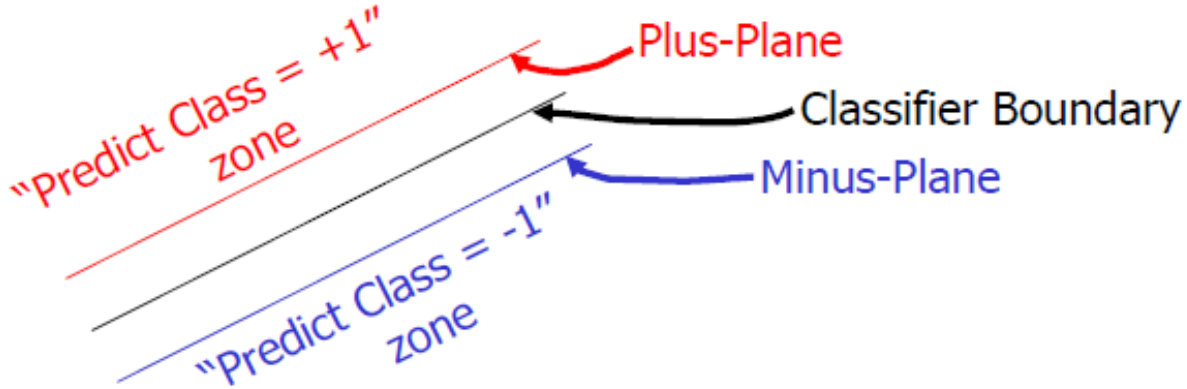
Linear SVM

# 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.

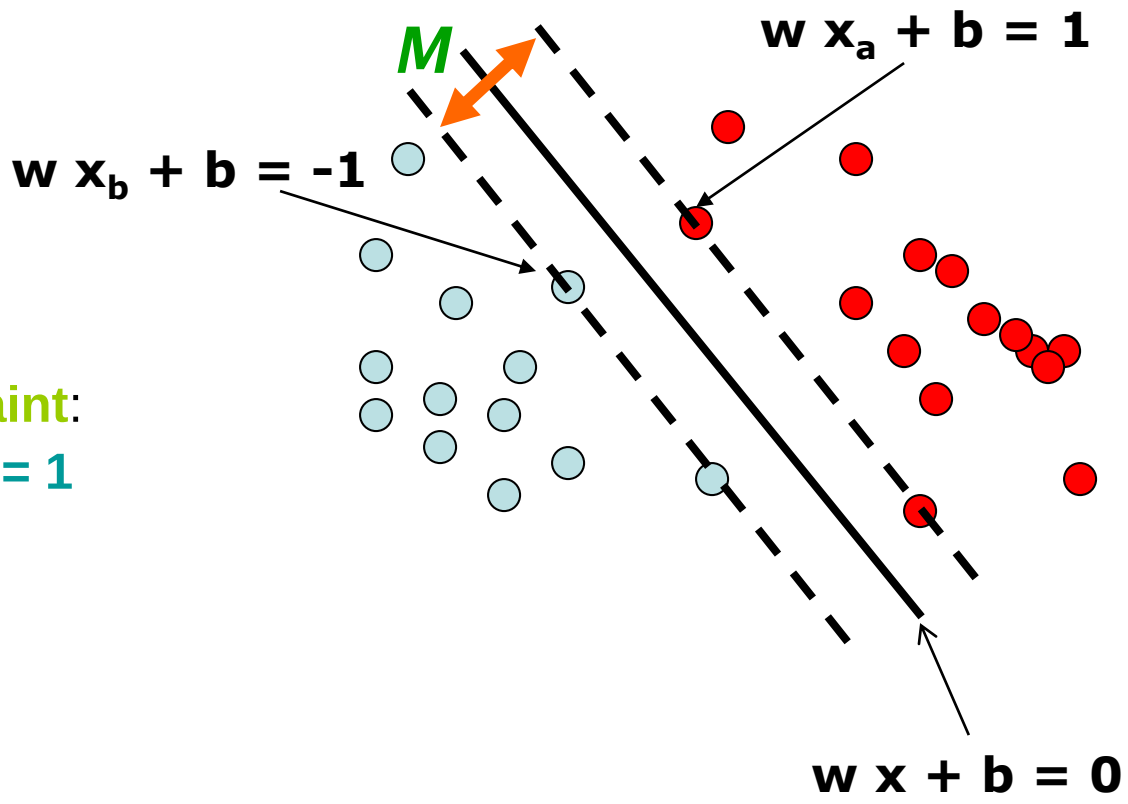
# Specifying a line and margin




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

# Linear Support Vector Machine (SVM)

- **Hyperplane**
  $\mathbf{w}\ \mathbf{x_b} + b = 0$
- **Extra scale constraint**:
  $\min_{i=1,\ldots,n} |\mathbf{w}\ \mathbf{x}_i + b| = 1$



# Linear SVM Mathematically: The linearly separable case

- Assume that all data is at least distance 1 from the hyperplane, then the following two constraints follow for a training set $\{(\mathbf{x_i}, y_i)\}$

$$\mathbf{y(w\ x_i} + b) \geq 1 \quad \text{if } y_i = 1$$

$$\mathbf{y(w\ x_i} + b) \leq -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality
- Then, since each example's distance from the hyperplane is

$$r = y\frac{\mathbf{wx} + b}{\| \mathrm{w} \|}$$

- As the absolute value of numerator of r is 1 ($|\mathbf{y(w\ x_i} + b)| = 1$ ) and margin includes the distance to the + and – support vectors (r twice), then:

$$M = \frac{2}{\| \mathrm{w} \|}$$

# Linear SVMs Mathematically (cont.)

- Then we can formulate the *quadratic optimization problem:*

Find $\mathbf{w}$ and $b$ such that

$M = \frac{2}{\|\mathbf{w}\|}$ is maximized; and for all $\{(\mathbf{x_i}, y_i)\}$

$\mathbf{w}\,\mathbf{x_i} + b \geq 1$ if $y_i = 1$; $\mathbf{w}\,\mathbf{x_i} + b \leq -1$ if $y_i = -1$

- A better formulation (min $\|\mathbf{w}\|$ = max $1/\|\mathbf{w}\|$ ):

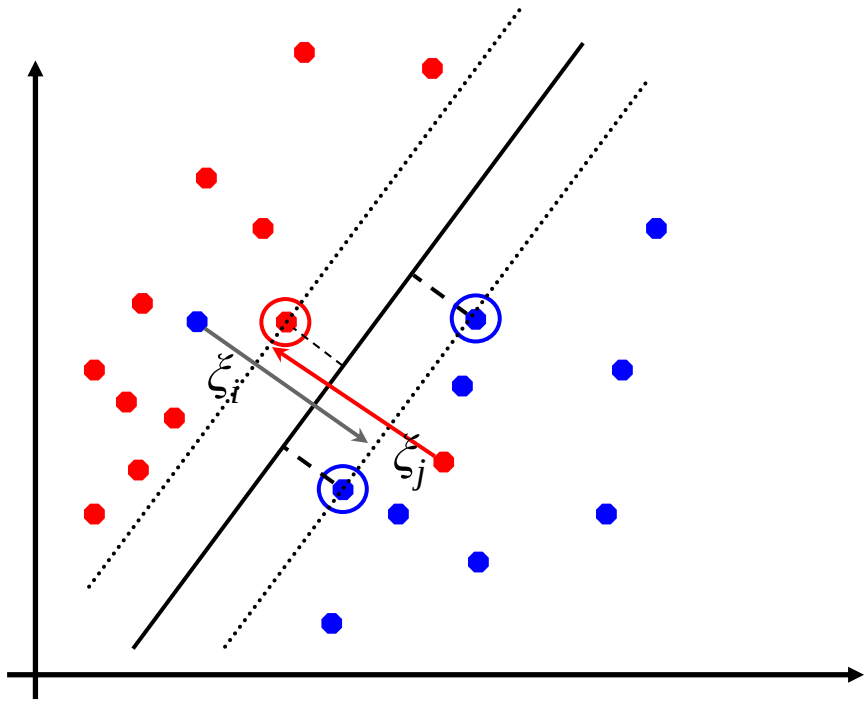
Find $\mathbf{w}$ and $b$ such that

$\Phi(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ is minimized;

and for all $\{(\mathbf{x_i}, y_i)\}$: $y_i\,(\mathbf{w}\,\mathbf{x_i} + b) \geq 1$

# Soft Margin Classification

- If the training data is not linearly separable, *slack variables* $\xi_i$ can be added to allow misclassification of difficult or noisy examples.
- Allow some errors
  - Let some points be moved to where they belong, at a cost
- Still, try to minimize training set errors, and to place hyperplane “far” from each class (large margin)



# Soft Margin Classification Mathematically

- The old formulation:

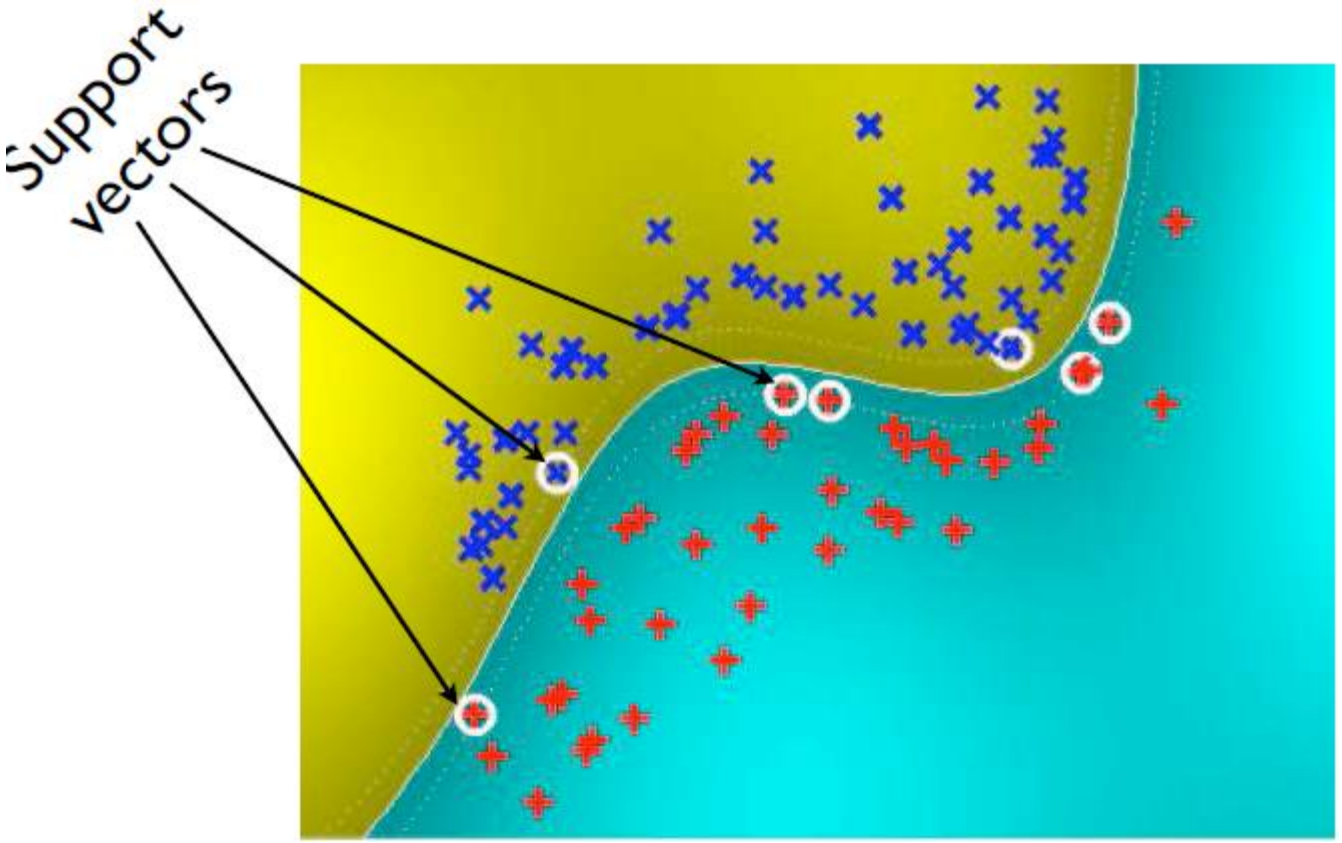
Find $\mathbf{w}$ and $b$ such that
$\mathbf{\Phi}(\mathbf{w}) = \frac{1}{2}\, \mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i\,(\mathbf{w}\,\mathbf{x_i} + b) \geq 1$

- The new formulation incorporating slack variables:

Find $\mathbf{w}$ and $b$ such that
$\mathbf{\Phi}(\mathbf{w}) = \frac{1}{2}\, \mathbf{w}^T\mathbf{w} + C\sum\xi_i$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$
$y_i\,(\mathbf{w}\,\mathbf{x_i} + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$ for all $i$

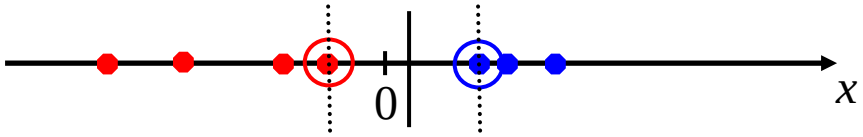
- Parameter $C$ can be viewed as a way to control overfitting
  - A regularization term

# Non-linear SVM

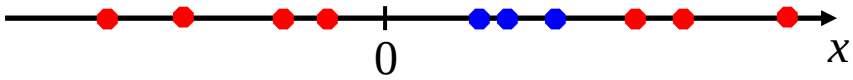


# Non-linear SVMs

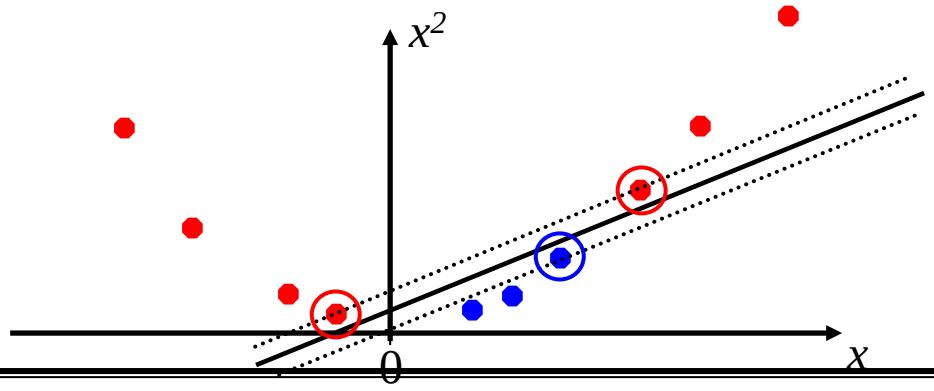
- Datasets that are linearly separable (with some noise) work out great:




- But what are we going to do if the dataset is just too hard?

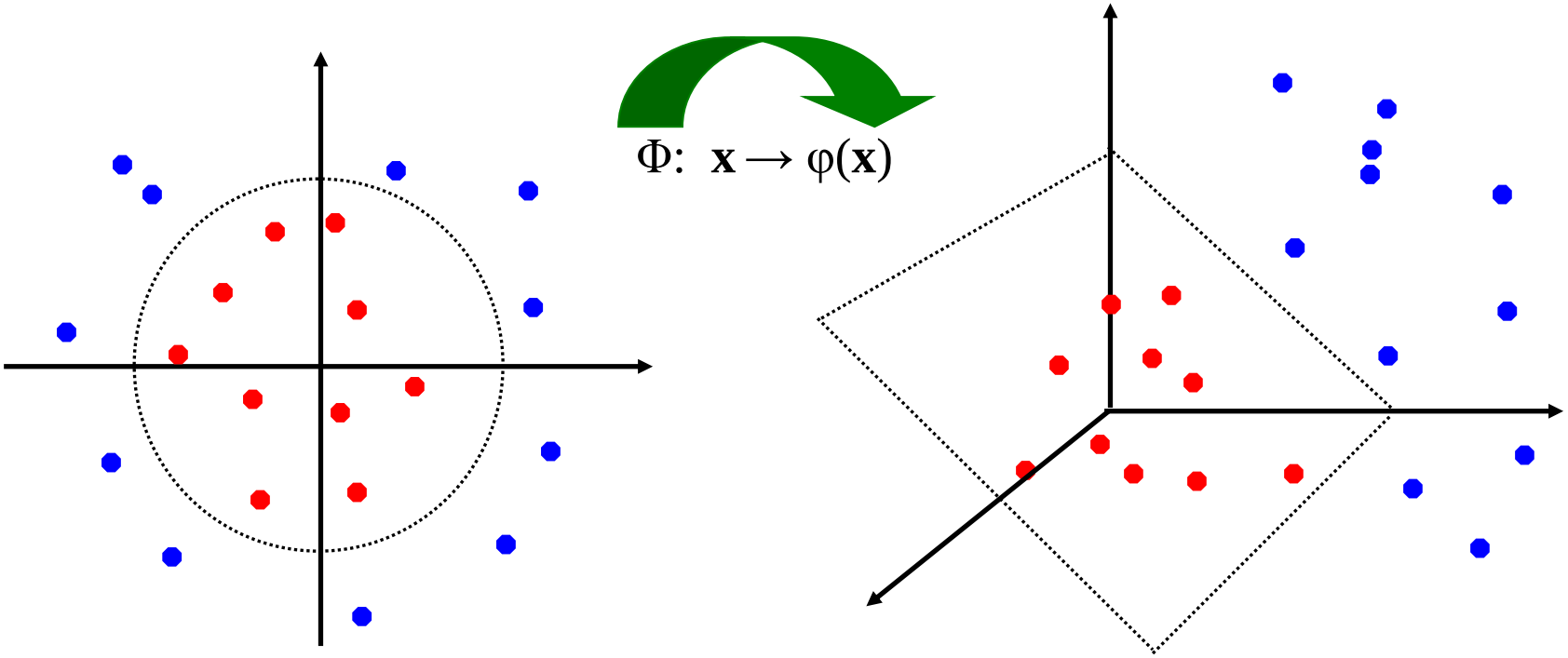



- How about … mapping data to a higher-dimensional space:



# Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



# The "Kernel Trick"

- The linear classifier relies on an inner product between vectors $K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j$
- If every datapoint is mapped into high-dimensional space via some transformation $\Phi$: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:

$$K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$$

- A *kernel function* is some function that corresponds to an inner product in some expanded feature space.
- Example:

  2-dimensional vectors $\mathbf{x}=[x_1 \;\; x_2]$; let $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2$,

  Need to show that $K(\mathbf{x}_i,\mathbf{x}_j)= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$:

  $K(\mathbf{x}_i,\mathbf{x}_j)=(1 + \mathbf{x}_i^T\mathbf{x}_j)^2 = 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}=$

  $= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T\, [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}]$

  $= \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$ where $\varphi(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$

# Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space
- Common kernels
  - Linear
  - Polynomial $K(x,z) = (1+x^Tz)^d$
    - Gives feature conjunctions.
      - Quadratic kernel: Model occurrences of pairs of words, which give distinctive information about topic classification, not given by the words alone. (i.e. operating AND system, ethnic AND cleansing)
      - Cubic kernel: Triples of words.
  - Radial basis function (infinite dimensional space): the most common form of radial basis function is a Gaussian distribution calculated as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}$$

- Haven't been very useful in text classification

# Evaluation: Classic Reuters-21578 Data Set

- Most (over)used data set
- 21578 documents
- 9603 training, 3299 test articles (ModApte/Lewis split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Average document: about 90 types, 200 tokens
- Average number of classes assigned
  - 1.24 for docs with at least one category
- Only about 10 out of 118 categories are large

Common categories (#train, #test)

- Earn (2877, 1087)
- Acquisitions (1650, 179)
- Money-fx (538, 179)
- Grain (433, 149)
- Crude (389, 189)
- Trade (369,119)
- Interest (347, 131)
- Ship (197, 89)
- Wheat (212, 71)
- Corn (182, 56)

# Reuters Text Categorization data set (**Reuters-21578)** document

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981" NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

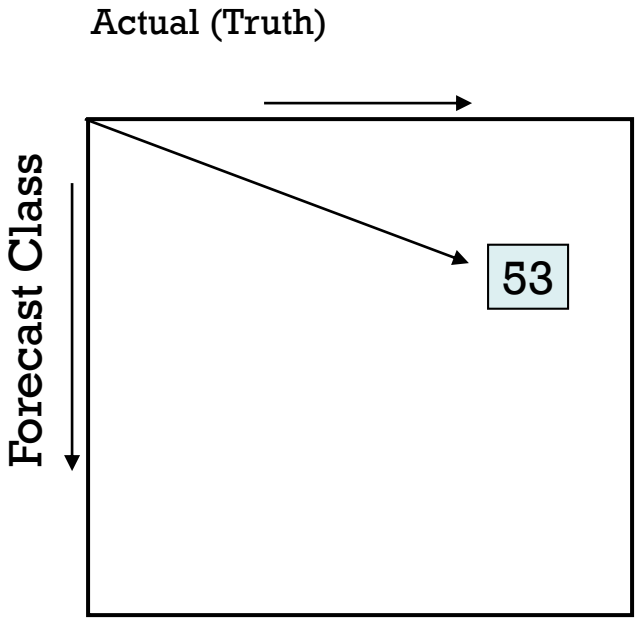
Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter

&#3;</BODY></TEXT></REUTERS>

# Good practice: confusion matrix

This (*i*, *j*) entry means 53 of the docs actually in class *i* were put in class *j* by the classifier.




| | Actual: yes | Actual: no |
|---|---|---|
| Forecast: yes | 10 | 10 |
| Forecast: no | 10 | 970 |

- In a perfect classification, only the diagonal has non-zero entries
- Look at common confusions and how they might be addressed

# Per class evaluation measures

- Recall: True positive rate: TP/(TP+FN)
  - Fraction of docs in class *i* classified correctly:
  - Fraction of relevant instances that are retrieved

$$\frac{c_{ii}}{\sum_j c_{ij}}$$

- Precision: Fraction of docs assigned class *i* that are actually about class *i*: TP/(TP+FP)
  - Fraction of retrieved instances that are relevant

$$\frac{c_{ii}}{\sum_j c_{ji}}$$

- Accuracy: (1 - error rate) Fraction of docs classified correctly:
  (TP+TN)/(TP+FP+TN+FN)

$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$$

| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| **Test Outcome** | Test Outcome Positive | **True Positive (TP)** | **False Posit. (FP)** (Type I error) |
| | Test Outcome Negative | **False Negat. (FN)** (Type II error) | **True Negative (TN)** |

# Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- Macroaveraging: Compute performance for each class, then average.
- Microaveraging: Collect decisions for all classes, compute contingency table, evaluate.

# Micro- vs. Macro-Averaging: Example

Class 1

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 10 | 10 |
| Classifier: no | 10 | 970 |

Class 2

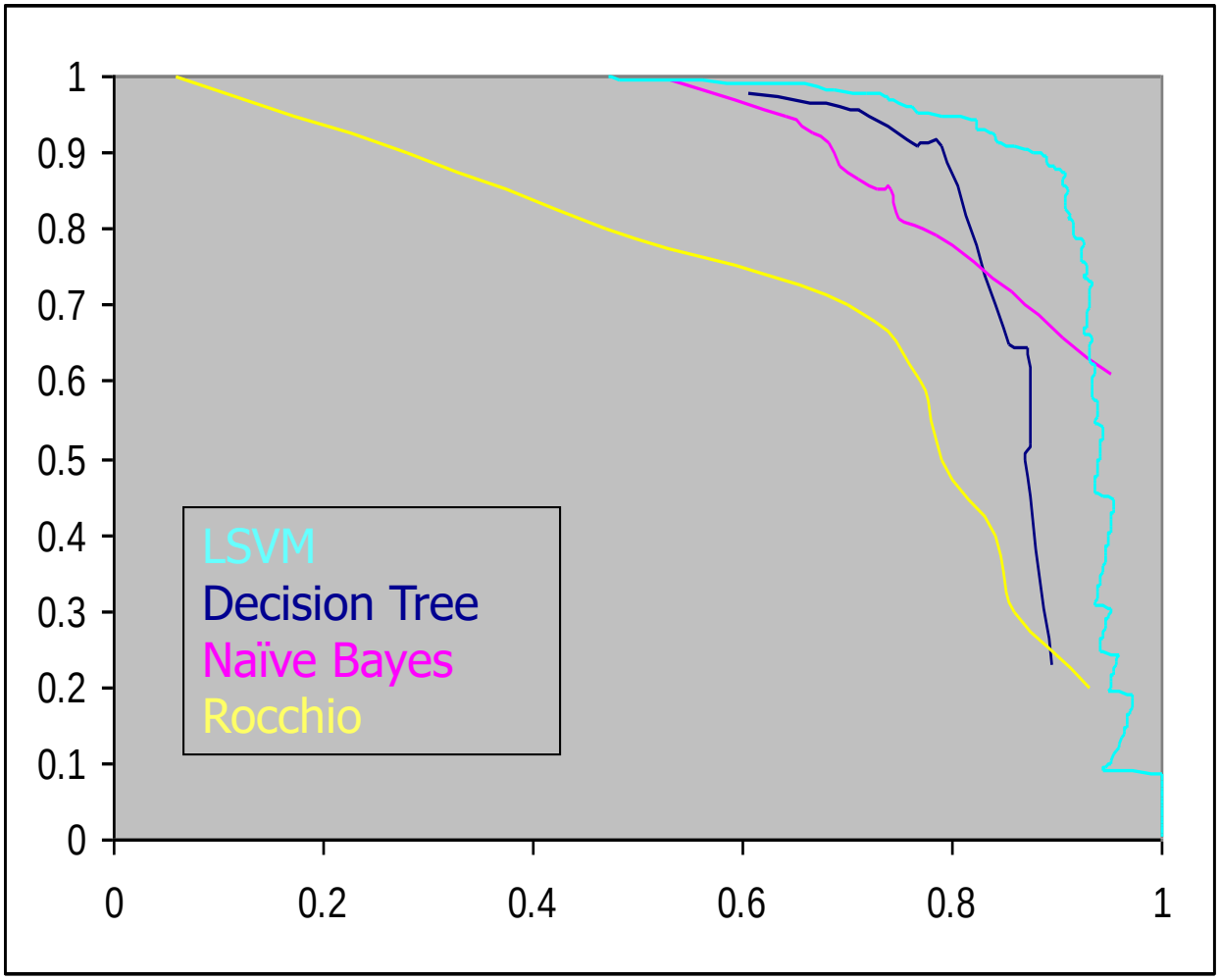
| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 90 | 10 |
| Classifier: no | 10 | 890 |

Micro Ave. Table

| | Truth: yes | Truth: no |
|---|---|---|
| Classifier: yes | 100 | 20 |
| Classifier: no | 20 | 1860 |

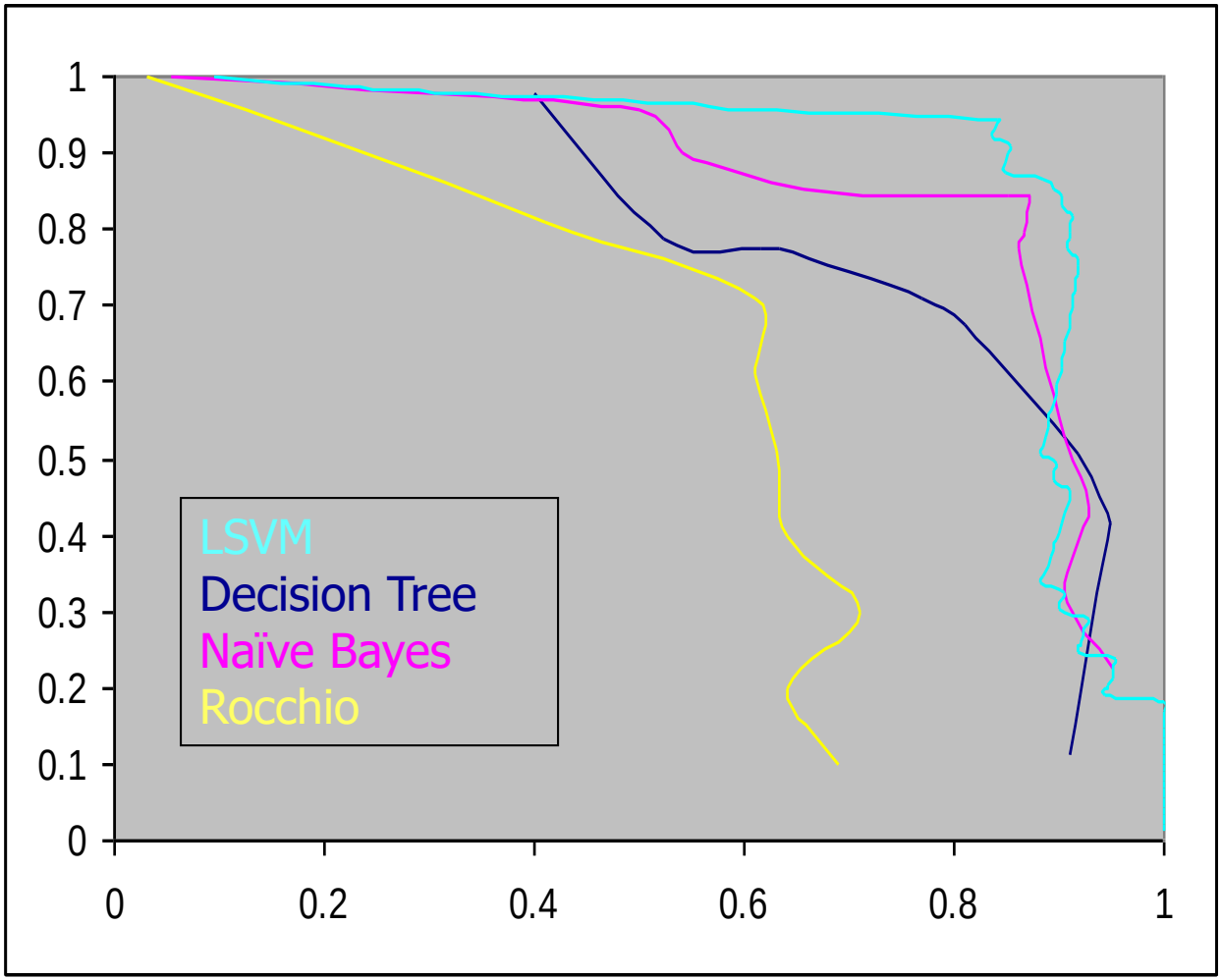
- Macroaveraged precision: (0.5 + 0.9)/2 = 0.7
- Microaveraged precision: 100/120 = .83
- Microaveraged score is dominated by score on common classes

# Precision-recall for category: Crude




Dumais (1998)

# Precision-recall for category: Ship




Dumais (1998)

# The Real World

P. Jackson and I. Moulinier. 2002. *Natural Language Processing for Online Applications*

- "There is no question concerning the commercial value of being able to classify documents automatically by content. There are myriad potential applications of such a capability for corporate intranets, government departments, and Internet publishers"

- "Understanding the data is one of the keys to successful categorization, yet this is an area in which most categorization tool vendors are extremely weak. Many of the 'one size fits all' tools on the market have not been tested on a wide range of content types."

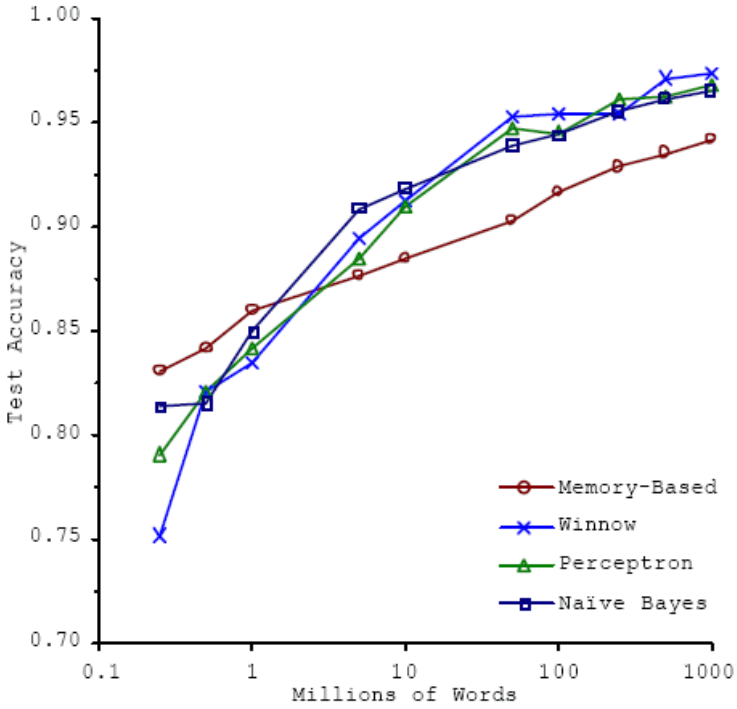
# The Real World

- Gee, I'm building a text classifier for real, now!
- What should I do?

- How much training data do you have?
  - None
  - Very little
  - Quite a lot
  - A huge amount and its growing

# Accuracy as a function of data size (learning curve)

- With enough data the choice of classifier may not matter much, and the best choice may be unclear
  - Data: Brill and Banko on context-sensitive spelling correction
- But the fact that you have to keep doubling your data to improve performance is a little unpleasant



# Summary

- Support vector machines (SVM)
  - Choose hyperplane based on support vectors
    - Support vector = “critical” point close to decision boundary
  - (Degree-1) SVMs are linear classifiers.
  - Kernels: powerful and elegant way to define similarity metric
  - Perhaps best performing text classifier
    - But there are other methods that perform about as well as SVM, such as regularized logistic regression (Zhang & Oles 2001)
  - Partly popular due to availability of good software
    - SVMlight is accurate and fast – and free (for research)
    - Now lots of good software: libsvm, TinySVM, ….
- Comparative evaluation of methods
- Real world: exploit domain specific structure!

# References


- Christopher J. C. Burges. 1998. A Tutorial on Support Vector Machines for Pattern Recognition
- S. T. Dumais. 1998. Using SVMs for text categorization, IEEE Intelligent Systems, 13(4)
- S. T. Dumais, J. Platt, D. Heckerman and M. Sahami. 1998. Inductive learning algorithms and representations for text categorization. *CIKM '98*, pp. 148-155.
- Yiming Yang, Xin Liu. 1999. A re-examination of text categorization methods. 22nd Annual International SIGIR
- Tong Zhang, Frank J. Oles. 2001. Text Categorization Based on Regularized Linear Classification Methods. Information Retrieval 4(1): 5-31
- Trevor Hastie, Robert Tibshirani and Jerome Friedman. *Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer-Verlag, New York.
- T. Joachims, *Learning to Classify Text using Support Vector Machines*. Kluwer, 2002.
- Fan Li, Yiming Yang. 2003. A Loss Function Analysis for Classification Methods in Text Categorization. ICML 2003: 472-479.
- Tie-Yan Liu, Yiming Yang, Hao Wan, et al. 2005. Support Vector Machines Classification with Very Large Scale Taxonomy, SIGKDD Explorations, 7(1): 36-43.
- 'Classic' Reuters-21578 data set: http://www.daviddlewis.com /resources /testcollections/reuters21578/