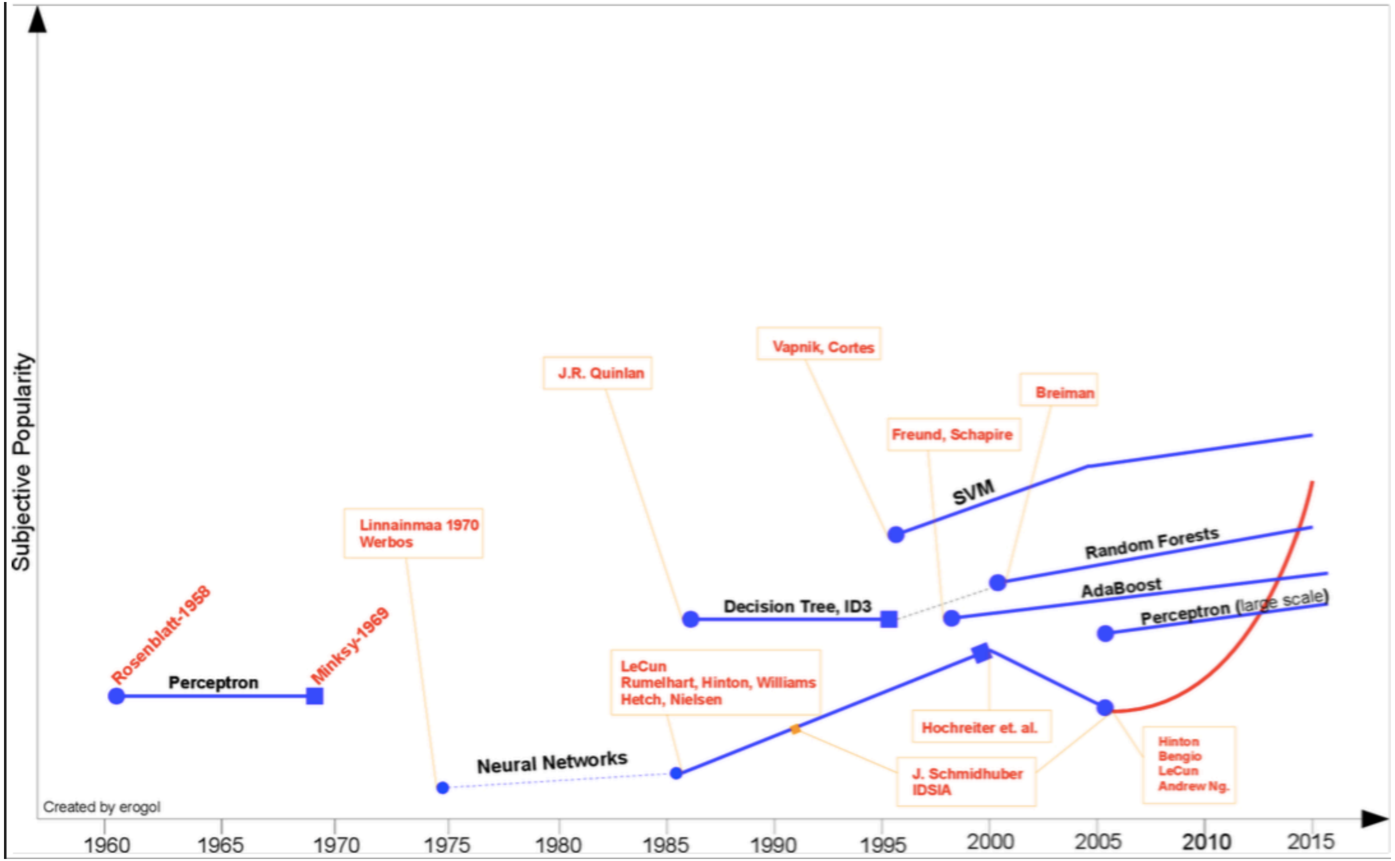
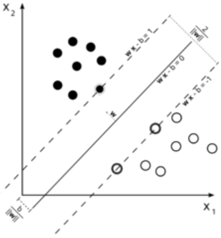

Subjective Popularity
Rosenblatt-1958
Perceptron
Minksy-1969
Linnainmaa 1970
Werbos
Neural Networks
J.R. Quinlan
Decision Tree, ID3
LeCun
Rumelhart, Hinton, Williams
Hetch, Nielsen
Vapnik, Cortes
SVM
Freund, Schapire
Breiman
Random Forests
AdaBoost
Perceptron (large scale)
Hochreiter et. al.
J. Schmidhuber
IDSIA
Hinton
Bengio
LeCun
Andrew Ng.
Created by erogol
1960
1965
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015



- The present form of support vector machine (SVM) was largely developed at AT&T Bell Laboratories by Vapnik and co-workers.
- Known as a maximum margin classifier.
- Originally proposed for classification and soon applied to regression and time series prediction.
- One of the most efficient supervised learning methods.

Problem

- Given a set of training samples

$$\left(x_1, y_1\right), \left(x_2, y_2\right), \cdots, \left(x_N, y_N\right), x_i \in R^n, y_i \in \{-1, 1\},$$

find a function $f(x, \alpha)$ to classify the samples, such that

$$f(x_i, \alpha) \begin{cases} > 0, & \forall y_i = +1; \\ < 0, & \forall y_i = -1, \end{cases}$$

where α denotes the parameters.

Problem

- Given a set of training samples

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N), x_i \in R^n, y_i \in \{-1, 1\},$$

find a function $f(x, \alpha)$ to classify the samples, such that

$$f(x_i, \alpha) \begin{cases} > 0, & \forall y_i = +1; \\ < 0, & \forall y_i = -1, \end{cases}$$

where α denotes the parameters.

- For a testing sample x, we can predict its label by $\text{sign}[f(x, \alpha)]$.

Problem

- Given a set of training samples

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N), x_i \in R^n, y_i \in \{-1, 1\},$$

find a function $f(x, \alpha)$ to classify the samples, such that

$$f(x_i, \alpha) \begin{cases} > 0, & \forall y_i = +1; \\ < 0, & \forall y_i = -1, \end{cases}$$

where α denotes the parameters.

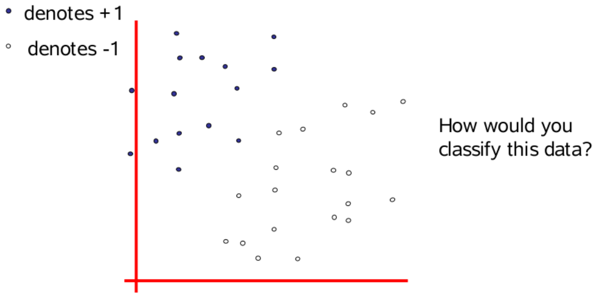
- For a testing sample x, we can predict its label by sign$[f(x, \alpha)]$.
- $f(x, \alpha) = 0$ is called the separation hyperplane.

Linear classifiers

Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

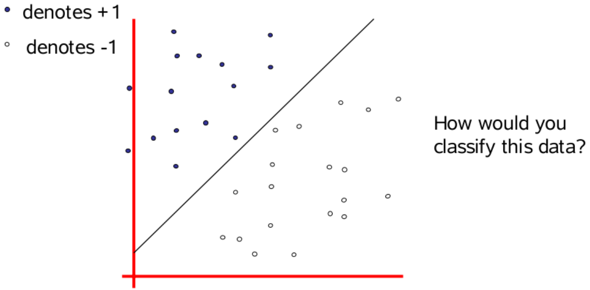


Linear classifiers

Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

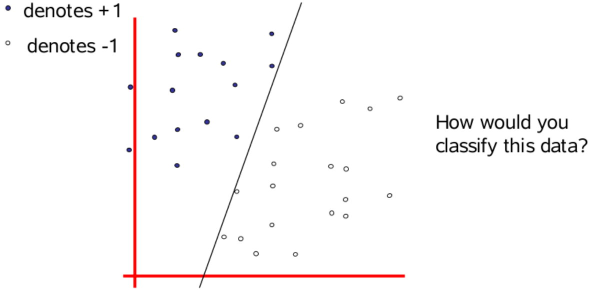


Linear classifiers

Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

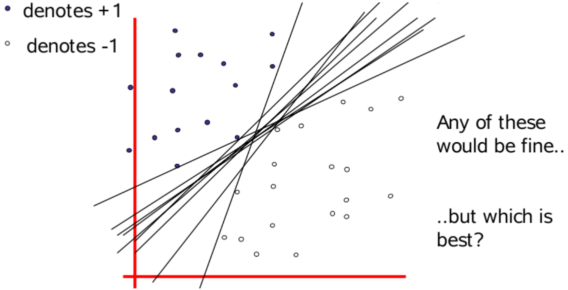


Linear classifiers

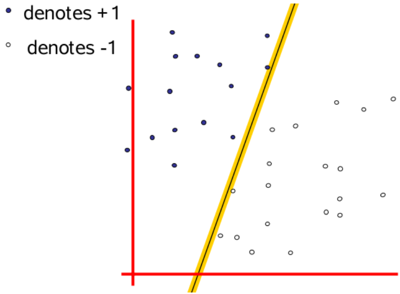
Linear hyperplane

$$f(x, w, b) = \langle x, w \rangle + b = 0$$

Consider the linearly separable case, there are infinite number of hyperplanes that can do the job.

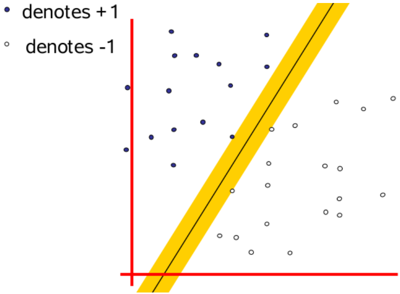


Margin of a linear classifier



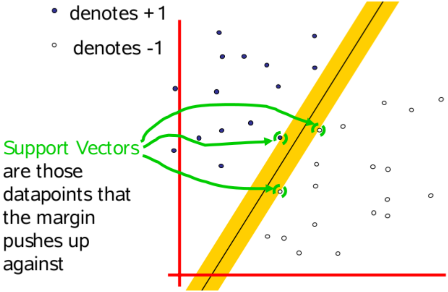
Definition: the width that the boundary could be increased by before hitting a data point.

Maximum margin linear classifier



Definition: the linear classifier with the maximum margin.

Support vectors



Problem formulation

To formulate the margin, we further requires that for all samples

$$f(x_i, \alpha) = \langle x_i, w \rangle + b \begin{cases} \geq +1, & \forall y_i = +1; \\ \leq -1, & \forall y_i = -1. \end{cases}$$

or

$$y_i(\langle x_i, w \rangle + b) \geq 1, \quad i = 1, \ldots, N.$$

Problem formulation

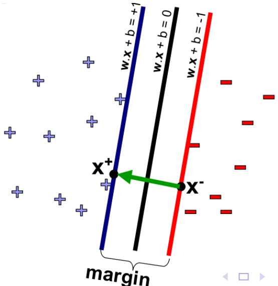
To formulate the margin, we further requires that for all samples

$$f(x_i, \alpha) = \langle x_i, w \rangle + b \begin{cases} \geq +1, & \forall y_i = +1; \\ \leq -1, & \forall y_i = -1. \end{cases}$$

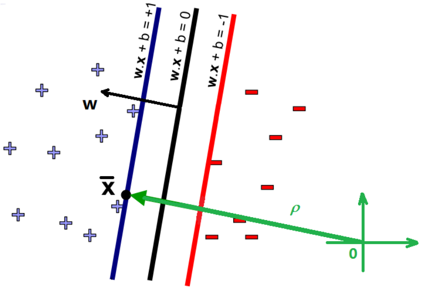
or

$$y_i(\langle x_i, w \rangle + b) \geq 1, \quad i = 1, \ldots, N.$$

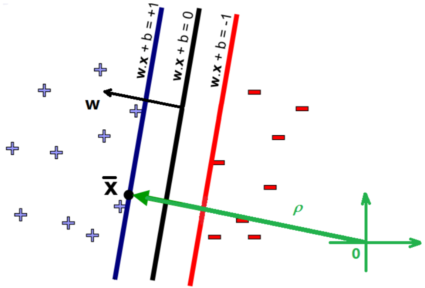
- We have introduced two additional hyperplanes $\langle x, w \rangle + b = \pm 1$ parallel to the separation hyperplane $\langle x, w \rangle + b = 0$



What is the margin? The distance between the two new hyperplanes.

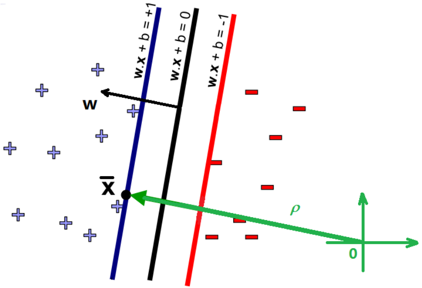


What is the margin? The distance between the two new hyperplanes.



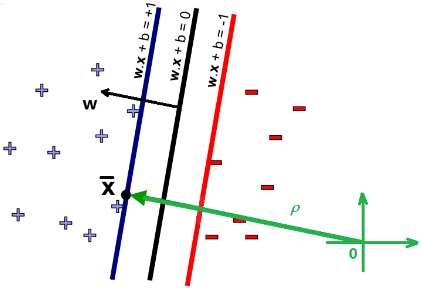
- The minimum distance between the hyperplane $\langle x, w \rangle + b = 1$ and the origin is $\rho_1 = \frac{1-b}{\|w\|}$. (why?)

What is the margin? The distance between the two new hyperplanes.



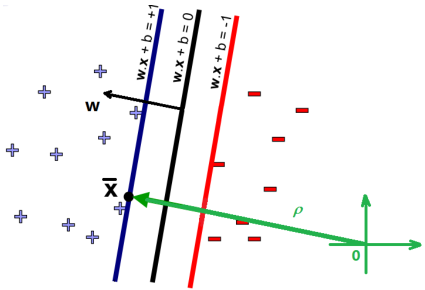
- The minimum distance between the hyperplane $\langle x, w \rangle + b = 1$ and the origin is $\rho_1 = \frac{1-b}{\|w\|}$. (why?)
- The minimum distance between the hyperplane $\langle x, w \rangle + b = -1$ and the origin is $\rho_2 = \frac{-1-b}{\|w\|}$.

What is the margin? The distance between the two new hyperplanes.



- The minimum distance between the hyperplane $\langle x, w \rangle + b = 1$ and the origin is $\rho_1 = \frac{1-b}{\|w\|}$. (why?)
- The minimum distance between the hyperplane $\langle x, w \rangle + b = -1$ and the origin is $\rho_2 = \frac{-1-b}{\|w\|}$.
- The margin is $|\rho_1 - \rho_2| = 2/\|w\|$.

How to calculate ρ_1 and ρ_2?

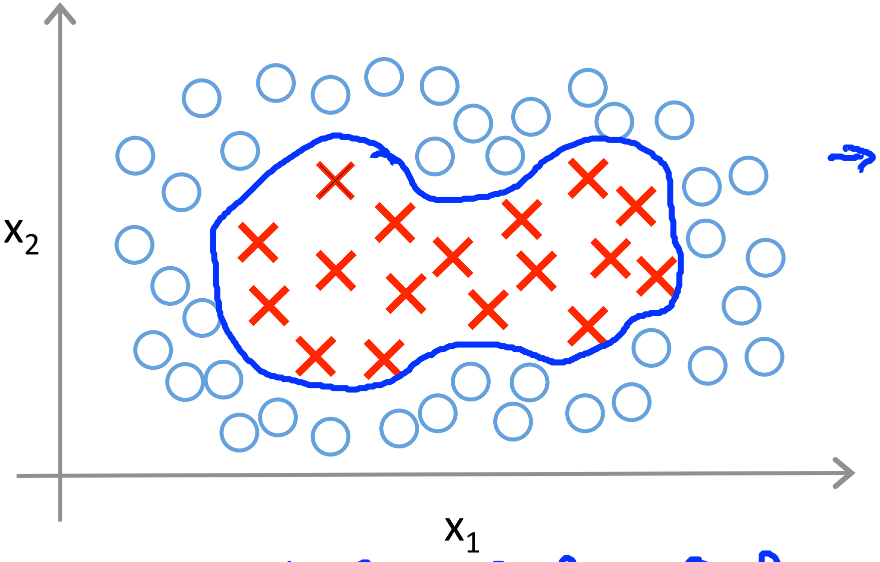


Note $\bar{x} = \rho_1 w/\|w\|$, where $w/\|w\|$ is the unit vector along the direction w. Since $\bar{x}$ is on the blue hyperplane, then

$$\langle \rho_1 w/\|w\|, w \rangle + b = 1$$

which follows $\rho_1 = \frac{1-b}{\|w\|}$. Similarly, we obtain $\rho_2 = \frac{-1-b}{\|w\|}$.

Non-linear Decision Boundary



Predict $y = 1$ if

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots \geq 0$$

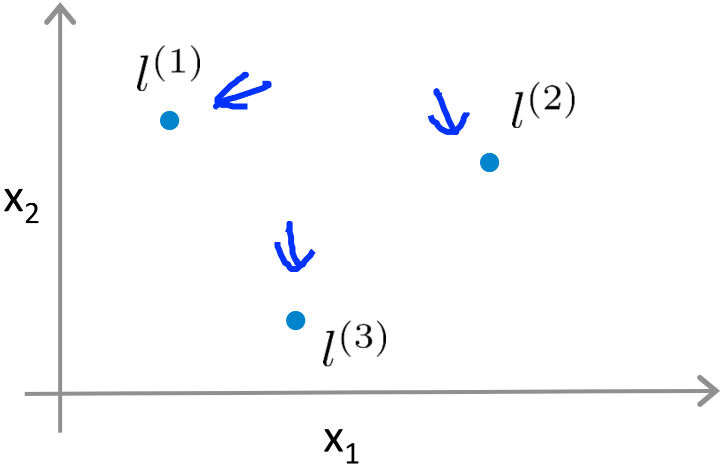
$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \cdots \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$$

$f_1 = x_1, \quad f_2 = x_2, \quad f_3 = x_1 x_2, \quad f_4 = x_1^2, \quad f_5 = x_2^2, \ldots$

Is there a different / better choice of the features $f_1, f_2, f_3, \ldots$?

Choosing the landmarks

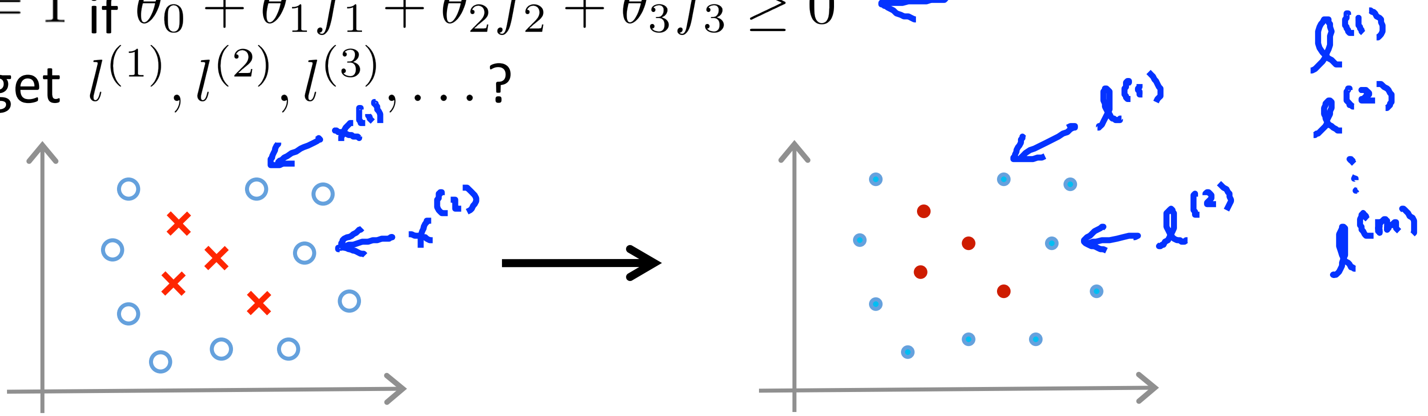


Given x:

$$f_i = \text{similarity}(x, l^{(i)})$$
$$= \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right)$$

Predict $y = 1$ if $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

Where to get $l^{(1)}, l^{(2)}, l^{(3)}, \ldots$?



The kernel-based function is exactly equivalent to preprocessing the data by applying similarity function to all inputs, then learning a linear model in the new transformed space.

SVM with Kernels

Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$,

choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \dots, l^{(m)} = x^{(m)}$.

Given example x:

$f_1 = \text{similarity}(x, l^{(1)})$ $\leftarrow x^{(i)}$

$f_2 = \text{similarity}(x, l^{(2)})$

$\dots$

$$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} \qquad f_0 = 1$$

For training example $(x^{(i)}, y^{(i)})$:

$x^{(i)} \rightarrow$

$f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)})$

$f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)})$

$\vdots$

$f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp\left(-\frac{0}{2\sigma^2}\right) = 1$ $\leftarrow x^{(i)}$

$\vdots$

$f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)})$

$x^{(i)} \in \mathbb{R}^{n+1}$ (or $\mathbb{R}^n$)

$$f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix}$$

$f_0^{(i)} = 1$

Andrew Ng

Commonly used kernels

- Homogeneous polynomials

$$k(x, y) = (\langle x, y \rangle)^d$$

- Inhomogeneous polynomials

$$k(x, y) = (\langle x, y \rangle + 1)^d$$

- Gaussian Kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

- Sigmoid Kernel

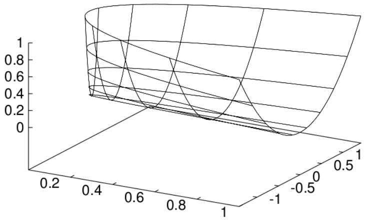
$$k(x, y) = \tanh(\eta\langle x, y \rangle + v)$$

Polynomial kernel

$$k(x, y) = (\langle x, y \rangle)^d$$

Example: $n = 2, d = 2, x = (x_1, x_2)$

- $\Phi(x) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$

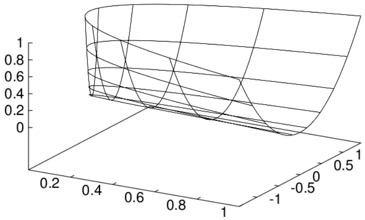


Polynomial kernel

$$k(x, y) = (\langle x, y \rangle)^d$$

Example: $n = 2, d = 2, x = (x_1, x_2)$

- $\Phi(x) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$



- Neither the mapping Φ nor the feature space is unique
  - $\Phi(x) = \left(x_1^2, x_1x_2, x_1x_2, x_2^2\right)$
  - $\Phi(x) = \frac{1}{\sqrt{2}}\left(x_1^2 - x_2^2, 2x_1x_2, x_1^2 + x_2^2\right)$

Logistic regression vs. SVMs

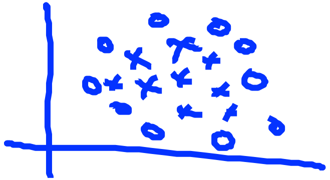
$n =$ number of features ($x \in \mathbb{R}^{n+1}$), $m =$ number of training examples

If n is large (relative to m): (e.g. $n \geq m$, $n = 10{,}000$, $m = 10 \cdots 1000$)

Use logistic regression, or SVM without a kernel ("linear kernel")

If n is small, m is intermediate: ($n = 1 - 1000$, $m = 10 - 10{,}000$)

Use SVM with Gaussian kernel



If n is small, m is large: ($n = 1 - 1000$, $m = 50{,}000+$)

Create/add more features, then use logistic regression or SVM without a kernel

Neural network likely to work well for most of these settings, but may be slower to train.

Andrew Ng