

# Support Vector Machines

Lecture 10

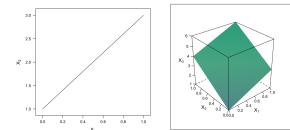
Termeh Shafee

Support Vector Machine (SVM) is a supervised learning algorithm used to learn a **hyperplane** that can solve the **binary classification** problem

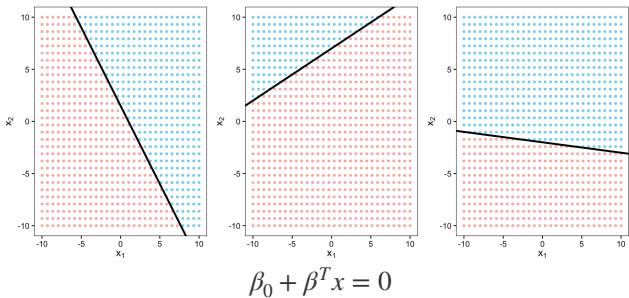
## Hyperplanes

"a flat affine subspace" 😊

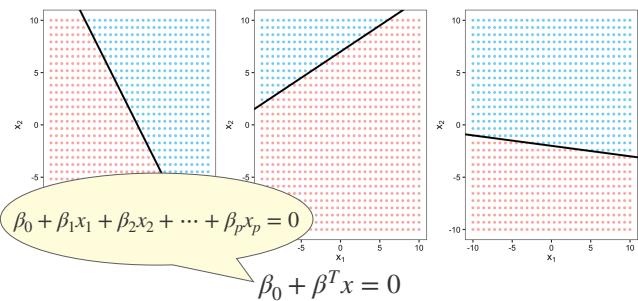
- **Flat:**
  - hyperplane is not curved, it increases/decreases constantly in each direction
- **Affine:**
  - the hyperplane doesn't need to pass through the origin
  - it can have an "offset" or be shifted (may have intercept)
- **Subspace:**
  - a subset of vectors in a larger vector space
  - in a  $d$ -dimensional space, a hyperplane has dimension  $d - 1$
  - in 2D it is a **line**, in 3D it is a **plane**



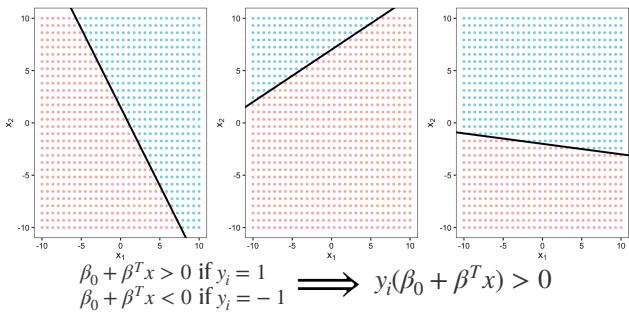
### Hyperplanes Divide the Space in Half



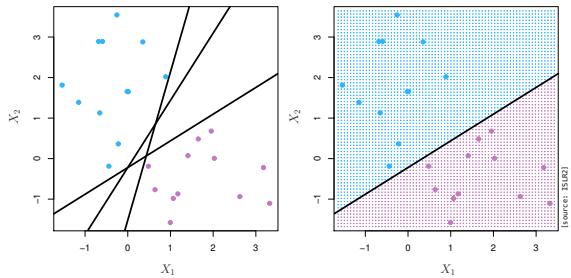
### Hyperplanes Divide the Space in Half



### Hyperplanes Divide Spaces in Half

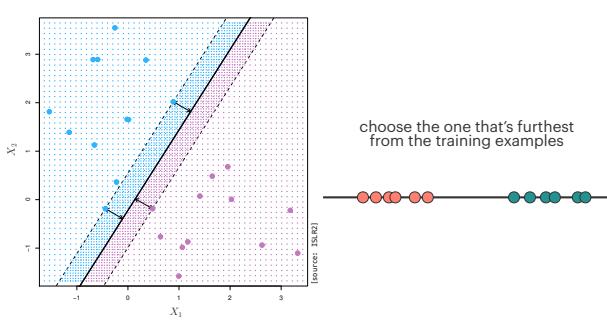


## Hyperplanes Divide Spaces in Half

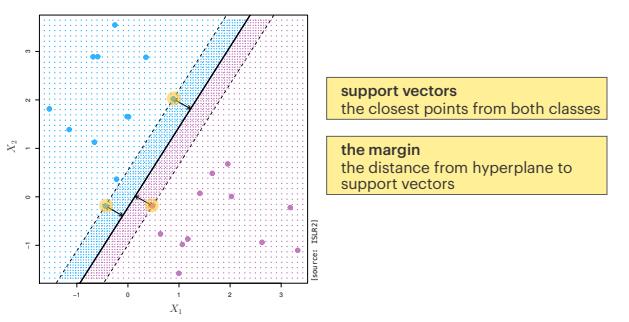


if a separating hyperplane exists, there will be  $\infty$  many of them  
how do we choose just one?

## Maximal Margin Classifier



## Maximal Margin Classifier



## Maximal Margin Classifier: The Math

The maximal margin classifier solves a constrained optimization problem:

$$\max_{\beta_0, \beta_1, \dots, \beta_p} M$$

subject to:

$$\|\beta\| = 1$$

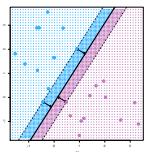
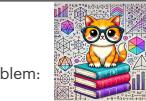
$$y_i(\beta_0 + \beta^T x_i) \geq M, \quad \forall i = 1, \dots, n$$

ensured each observation is on the correct side of the hyperplane and at least a distance  $M$  from the hyperplane,

i.e.,  $M$  is the margin of the hyperplane

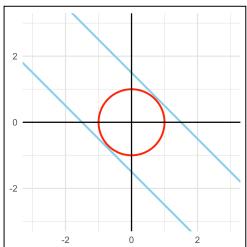
distance between  $x_i$  and line where

$$\|\beta\| = \sqrt{\sum_{j=1}^p \beta_j^2}$$
 is the Euclidean norm of  $\beta$



## What is a Constrained Optimization Problem?

Optimize  $f(x, y)$  subject to  $g(x, y) = k$



$$f(x, y) = 2x + y$$

$$g(x, y) = x^2 + y^2 = 1$$

$$\max_{\beta_0, \beta_1, \dots, \beta_p} M$$

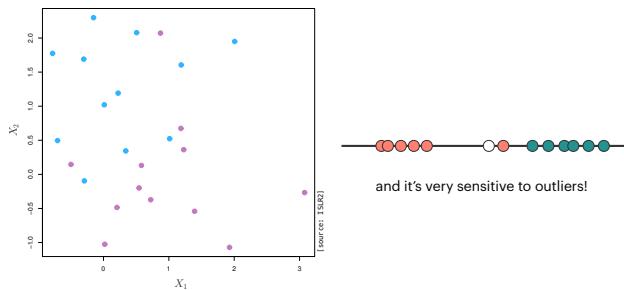
subject to:

$$\|\beta\| = 1$$

$$y_i(\beta_0 + \beta^T x_i) \geq M$$

## The Non-Separable Case

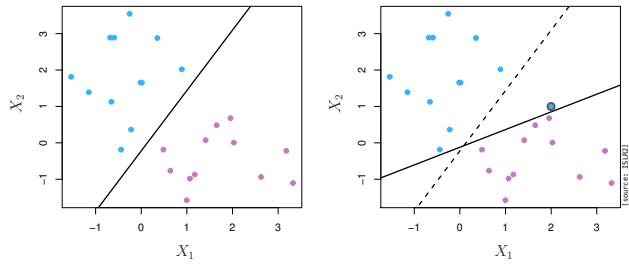
the optimization problem for the maximal margin classifier often has no solution with  $M > 0$



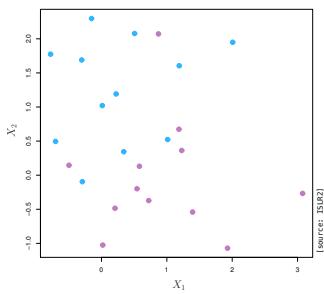
and it's very sensitive to outliers!

## The Non-Separable Case

even if the data are separable, they are sometimes noisy  
⇒ poor solution for the maximum margin classifier



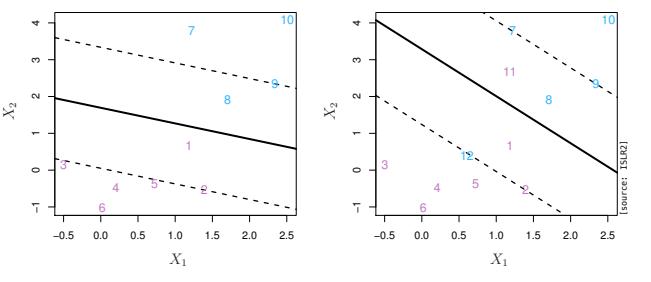
## The Non-Separable Case



support vector classifier:  
using a so-called soft margin  
to almost separate the classes

## Support Vector Classifier

allows us to classify data that is not linearly separable



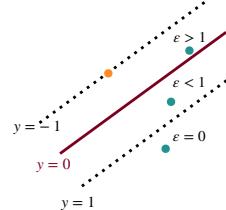
## Support Vector Classifier

$$\max_{\beta_0, \beta_1, \dots, \beta_p, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n} M$$

subject to:

$$\begin{aligned} & \|\beta\| = 1 \\ & y_i(\beta_0 + \beta^T x_i) \geq M(1 - \varepsilon_i) \\ & \varepsilon_i \geq 0, \sum_{i=1}^n \varepsilon_i \leq C \end{aligned}$$

$\varepsilon_1, \dots, \varepsilon_n$  are slack variables where  $\varepsilon_i = 0$  means  $i^{\text{th}}$  observation is on correct side of margin  
 $< 1$  means  $i^{\text{th}}$  observation is on wrong side of margin  
 $> 1$  means  $i^{\text{th}}$  observation is on wrong side of hyperplane



## Support Vector Classifier

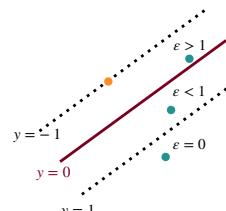
$$\max_{\beta_0, \beta_1, \dots, \beta_p, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_n} M$$

subject to:

$$\begin{aligned} & \|\beta\| = 1 \\ & y_i(\beta_0 + \beta^T x_i) \geq M(1 - \varepsilon_i) \\ & \varepsilon_i \geq 0, \sum_{i=1}^n \varepsilon_i \leq C \end{aligned}$$

$C$  is the tuning parameter/penalty on error:

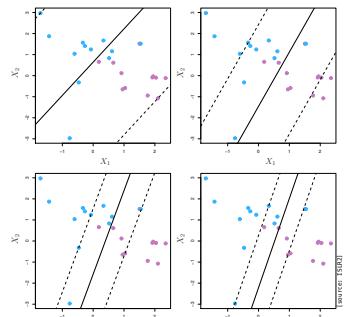
- $C = 0$  implies maximal margin hyperplane (superposed if exists)
- $C > 0$  is the total violations to the margin that we can tolerate
- $\Rightarrow \max C$  observations can be on the wrong side of hyperplane



## Support Vector Classifier

$C$ : penalty on error

	Regularization	Margins	Bias/Variance
Small $C$	more	wider	prone to underfitting
Large $C$	less	narrower	prone to overfitting

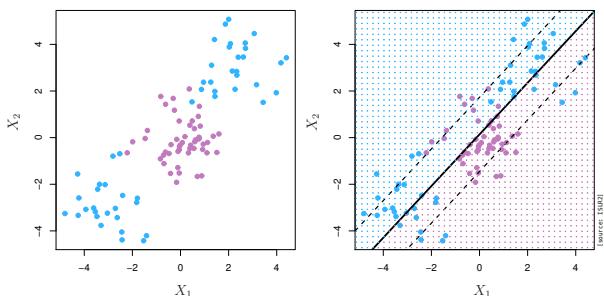


## Support Vector Machines

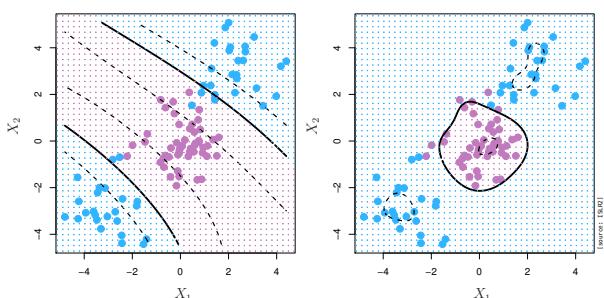


Support Vector Machine (SVM) use Kernel Functions to systematically find Support Vector Classifiers in higher dimensions

## Not Linearly Separable Even With Error

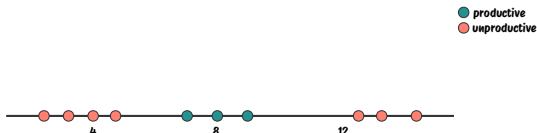


## The Kernel Trick



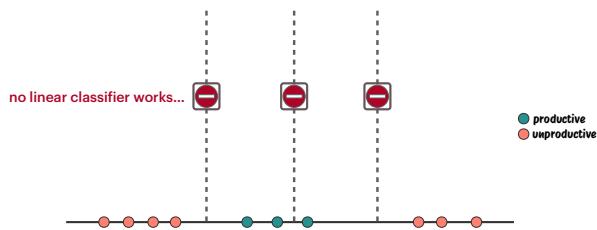
## The Kernel Trick

my productivity based on hours of sleep



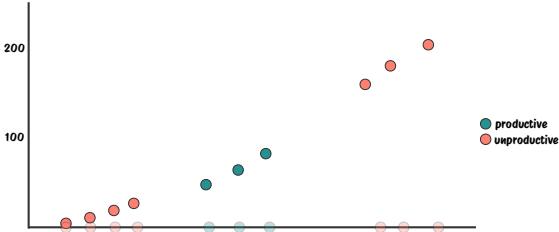
## The Kernel Trick

my productivity based on hours of sleep



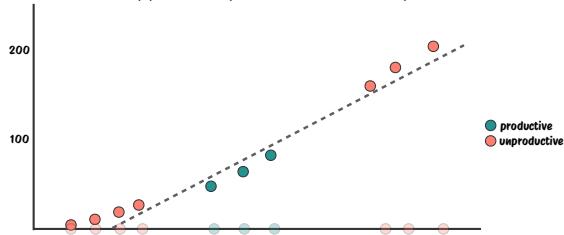
## The Kernel Trick

my productivity based on hours of sleep



## The Kernel Trick

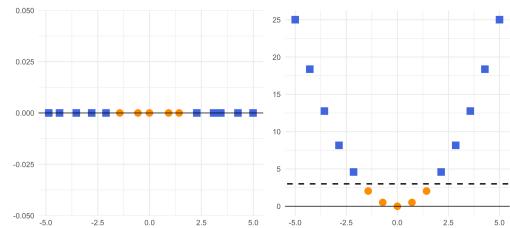
my productivity based on hours of sleep



## The Kernel Trick

### what is an SVM Kernel?

A function that computes the relationship between vectors in multiple dimensions (without actually having to calculate the coordinates for those dimensions)



## The Polynomial Kernel

The **Polynomial Kernel** in the previous sleep vs. happiness example

$$\underbrace{K(a, b)}_{\text{Kernel}} = (a \cdot b + r)^d \quad \text{where } r \text{ is the coefficients and } d \text{ the degree (determined by cross validation)}$$

Kernel  
different observations

## The Polynomial Kernel

The **Polynomial Kernel** in the previous sleep vs. happiness example

$$K(a, b) = (a \cdot b + r)^d \quad \text{where } r \text{ is the coefficients and } d \text{ the degree}$$

we set  $r = \frac{1}{2}$  and  $d = 2$ :

$$\begin{aligned} (a \cdot b + \frac{1}{2})^2 &= (a \cdot b + \frac{1}{2})(a \cdot b + \frac{1}{2}) \\ &+ a^2b^2 + \frac{1}{2}ab + \frac{1}{2}ab + \frac{1}{4} \\ &= ab + a^2b^2 + \frac{1}{4} \end{aligned}$$



## The Polynomial Kernel

The **Polynomial Kernel** in the previous sleep vs. happiness example

$$K(a, b) = (a \cdot b + r)^d \quad \text{where } r \text{ is the coefficients and } d \text{ the degree}$$

we set  $r = \frac{1}{2}$  and  $d = 2$ :

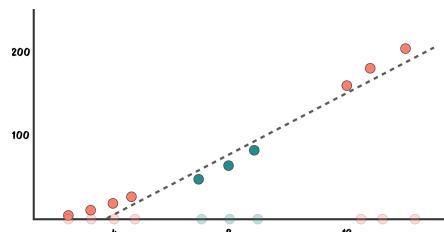
$$\begin{aligned} (a \cdot b + \frac{1}{2})^2 &= (a \cdot b + \frac{1}{2})(a \cdot b + \frac{1}{2}) \\ &+ a^2b^2 + \frac{1}{2}ab + \frac{1}{2}ab + \frac{1}{4} \\ &= ab + a^2b^2 + \frac{1}{4} = \underbrace{(a, a^2, \frac{1}{2}) \cdot (b, b^2, \frac{1}{2})}_{\text{dot product}} \end{aligned}$$

gives us the high dimensional coordinates for the data



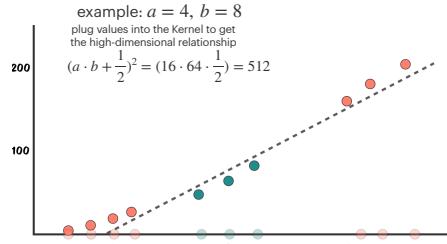
## The Polynomial Kernel

$$(a, a^2, \frac{1}{2}) \cdot (b, b^2, \frac{1}{2})$$



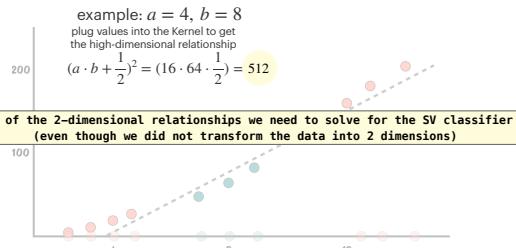
## The Polynomial Kernel

A function that computes the relationship between vectors in multiple dimensions  
(without actually having to calculate the coordinates for those dimensions)



## The Polynomial Kernel

A function that computes the relationship between vectors in multiple dimensions  
(without actually having to calculate the coordinates for those dimensions)



## The Radial Kernel (RBF)

The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

## The Radial Kernel (RBF)

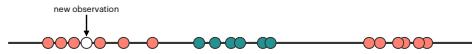
The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

the amount of influence one observation has on another is a function of the squared distance

$\gamma$  scales the squared distance to determine the strength of influence (determined by cross validation)



## The Radial Kernel (RBF)

The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

we can use the Polynomial Kernel to get the intuition behind how Radial Kernel works in infinite dimensions

$$K(a, b) = (a \cdot b + r)^d$$

set  $r = 0 \implies (a \cdot b)^d = a^d \cdot b^d$

set  $d = 1 \implies (a) \cdot (b)$



## The Radial Kernel (RBF)

The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

we can use the Polynomial Kernel to get the intuition behind how Radial Kernel works in infinite dimensions

$$K(a, b) = (a \cdot b + r)^d$$

set  $r = 0 \implies (a \cdot b)^d = a^d \cdot b^d$

set  $d = 1 \implies (a) \cdot (b)$

set  $d = 2 \implies (a^2) \cdot (b^2)$



## The Radial Kernel (RBF)

The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

we can use the Polynomial Kernel to get the intuition  
behind how Radial Kernel works in infinite dimensions

$$K(a, b) = (a \cdot b + r)^d$$

$$\text{set } r = 0 \implies (a \cdot b)^d = a^d \cdot b^d$$

$$\text{set } d = 1 \implies (a) \cdot (b)$$

$$\text{set } d = 2 \implies (a^2) \cdot (b^2)$$

$$\text{set } d = 3 \implies (a^3) \cdot (b^3)$$



## The Radial Kernel (RBF)

The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

we can use the Polynomial Kernel to get the intuition  
behind how Radial Kernel works in infinite dimensions

$$K(a, b) = (a \cdot b + r)^d$$

$$\text{set } r = 0 \implies (a \cdot b)^d = a^d \cdot b^d$$

$$\text{set } d = 1 \implies (a) \cdot (b)$$

$$\text{set } d = 2 \implies (a^2) \cdot (b^2)$$

$$\text{set } d = 3 \implies (a^3) \cdot (b^3)$$

we stay in same dimension  
but what if we took these polynomials as a sum?



## The Radial Kernel (RBF)

The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

we can use the Polynomial Kernel to get the intuition  
behind how Radial Kernel works in infinite dimensions

$$K(a, b) = (a \cdot b)^d$$

$$ab + a^2b^2 = (a, a^2)(b, b^2)$$



## The Radial Kernel (RBF)

The Radial Kernel

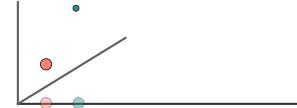
$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

we can use the Polynomial Kernel to get the intuition  
behind how Radial Kernel works in infinite dimensions

$$K(a, b) = (a \cdot b)^d$$

$$ab + a^2b^2 + a^3b^3 = (a, a^2, a^3)(b, b^2, b^3)$$



## The Radial Kernel (RBF)

The Radial Kernel

$$K(a, b) = e^{-\gamma}(a - b)^2$$

projects to infinite dimensional space  
works similar to nearest neighbors classifier

we can use the Polynomial Kernel to get the intuition  
behind how Radial Kernel works in infinite dimensions

$$K(a, b) = (a \cdot b)^d$$

$$ab + a^2b^2 + a^3b^3 + \dots + a^\infty b^\infty = (a, a^2, a^3, \dots, a^\infty)(b, b^2, b^3, \dots, b^\infty)$$

take sum for infinite terms gives dot product with infinite dimensions!

## The Radial Kernel: Taylor Series Expansion

$$K(a, b) = e^{-\gamma}(a - b)^2 = e^{-\gamma(a^2+b^2-2ab)} = e^{-\gamma(a^2+b^2)}e^{\gamma 2ab}$$

$$\text{set } \gamma = \frac{1}{2} \implies e^{-\frac{1}{2}\gamma(a^2+b^2)} e^{ab}$$



$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots + \frac{f^{(\infty)}(a)}{\infty!}(x - a)^\infty$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^\infty}{\infty!}$$

$$e^{ab} = 1 + (ab) + \frac{(ab)^2}{2!} + \frac{(ab)^3}{3!} + \dots + \frac{(ab)^\infty}{\infty!}$$

each term contains Polynomial Kernel with  $r = 0$  and  $d$  from 0 to  $\infty$

## The Radial Kernel: Taylor Series Expansion

$$K(a, b) = e^{-\gamma}(a - b)^2 = e^{-\gamma(a^2 + b^2 - 2ab)} = e^{-\gamma(a^2 + b^2)}e^{\gamma 2ab}$$

set  $\gamma = \frac{1}{2} \implies e^{-\frac{1}{2}\gamma(a^2 + b^2)} e^{ab}$  Taylor expansion of this term

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots + \frac{f^{(\infty)}(a)}{\infty!}(x - a)^\infty$$

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^\infty}{\infty!}$$

$$e^{ab} = 1 + ab + \frac{1}{2!}(ab)^2 + \frac{1}{3!}(ab)^3 + \dots + \frac{1}{\infty!}(ab)^\infty$$



Radial Kernels have coordinates for infinite dimensions!

$$a^0b^0 + a^1b^1 + a^2b^2 + a^3b^3 + \dots + a^\infty b^\infty = (a, a^2, a^3, \dots, a^\infty)(b, b^2, b^3, \dots, b^\infty)$$

## This Week's Practical

Support Vector Machines

