

ETC3250: Classification with Support Vector Machines

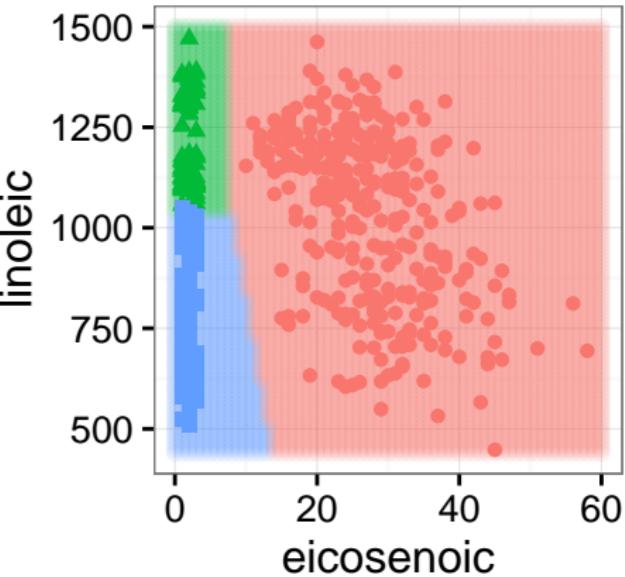
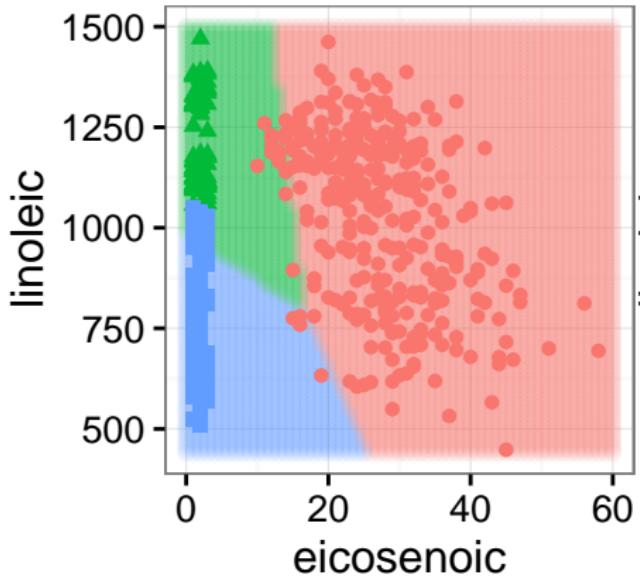
Week 8, class 2

Professor Di Cook, Econometrics and
Business Statistics

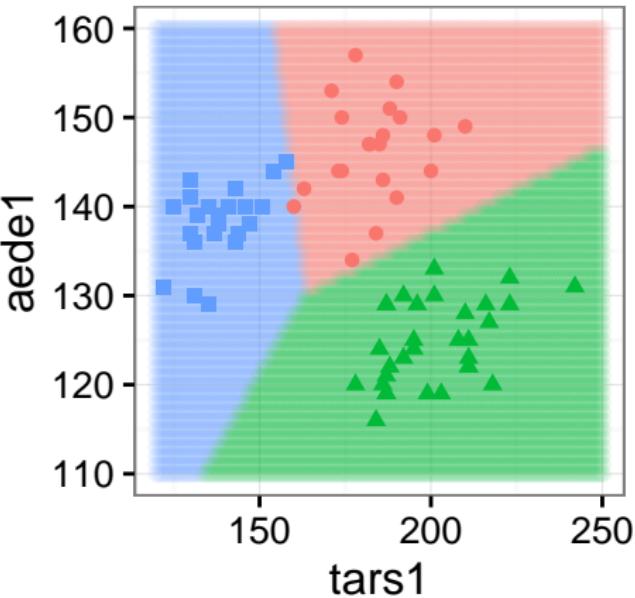
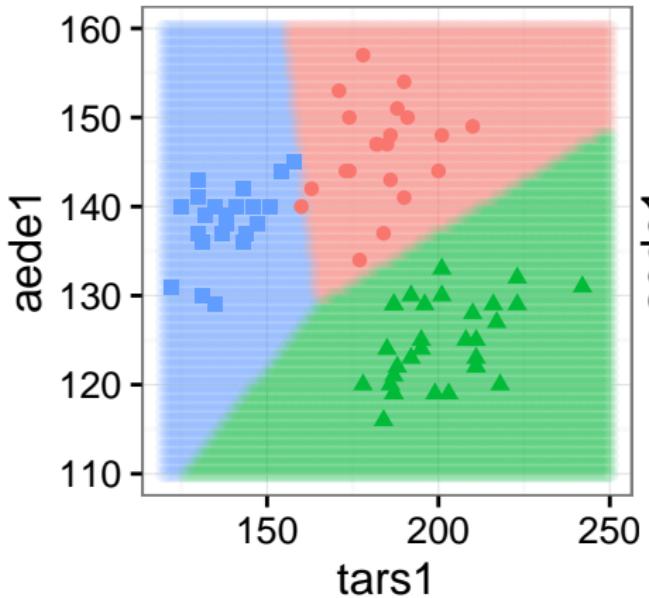


- Support vector machines build a classifier by finding **gaps** between clusters
- Primarily only work on two classes at a time, so multiclass problems need some tweaking of approach
- Let's look at how it works on the same three examples as used for LDA

Boundaries for Olive oils: LDA, SVM

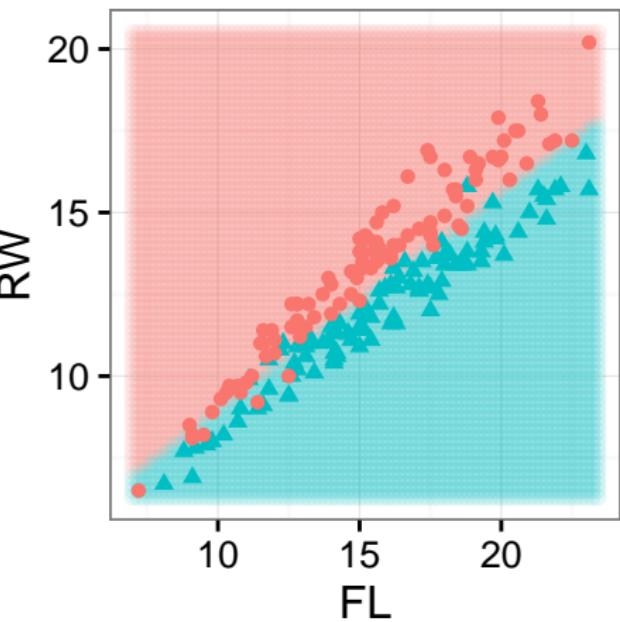
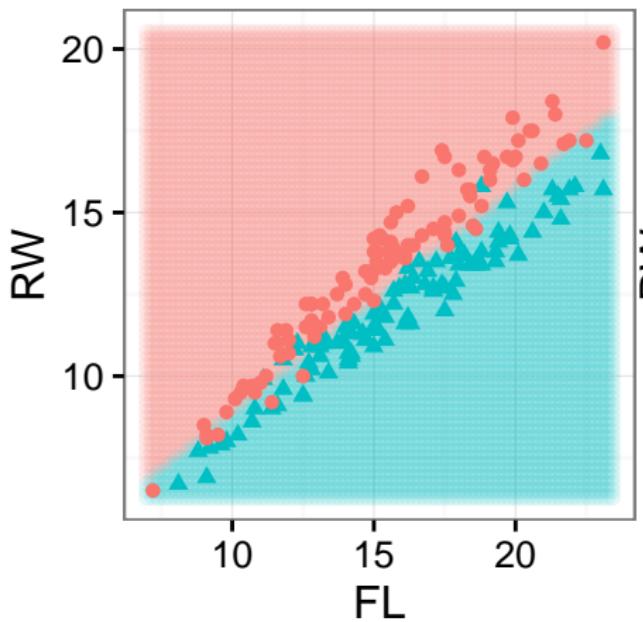


Boundaries for Beetles: LDA, SVM



Boundaries for Crabs: LDA, SVM

M



- Which boundaries look better?
- Why?

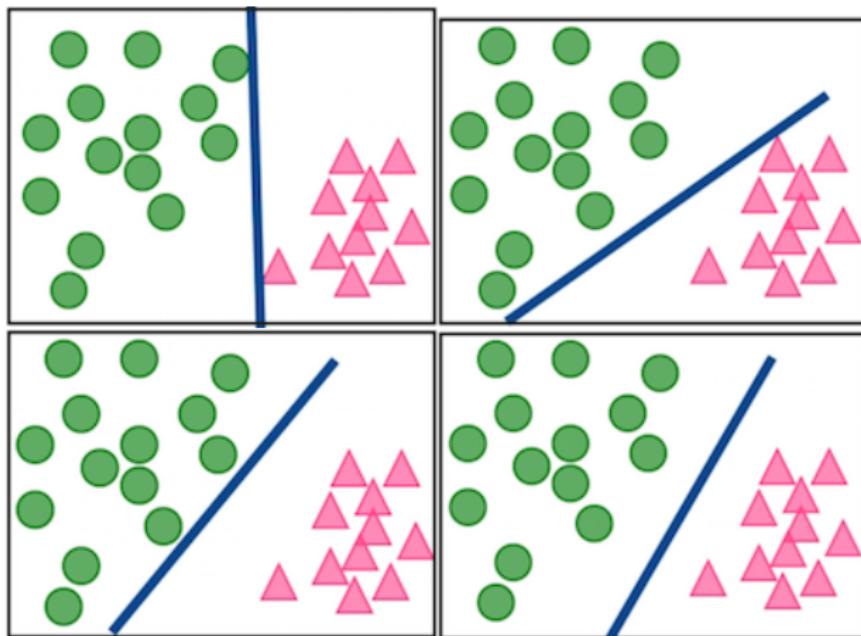
How does SVM work?

- Variables (x_1, \dots, x_p) need to be standardized
- Class (y) is coded as ± 1
- Separating hyperplane defined to be $\{x : x^T b + b_o = 0\}$ (x, b are p -dimensional vectors)
- where $b = \sum_{i=1}^s (\alpha_i y_i) x_i$
- s is the number of support vectors
- estimated by maximizing margin $M = 2/\|b\|$ subject to $\sum_1^p b_i^2 = 1$,
 $y_i(x_i^T b + b_o) \geq 1, i = 1, \dots, n$

Best separating hyperplane

M

- All are separating hyperplanes. Which is best?



Maximum margin

M

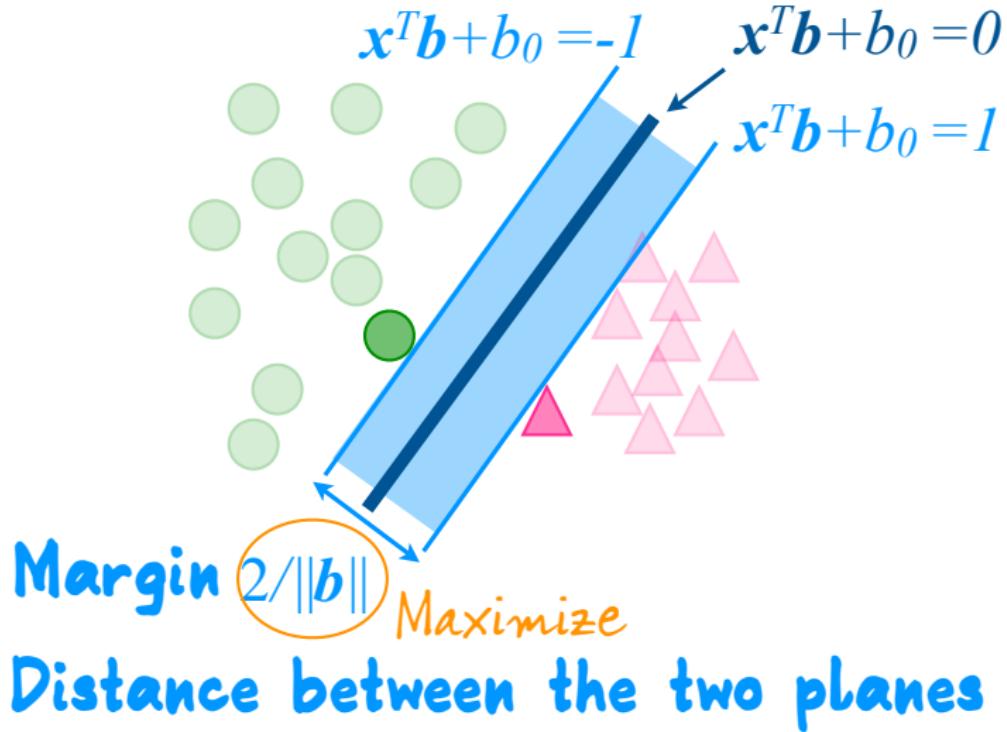
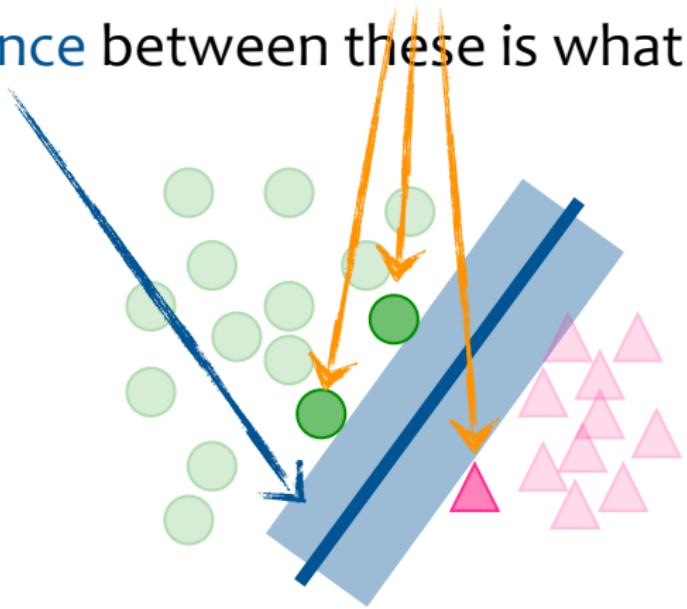


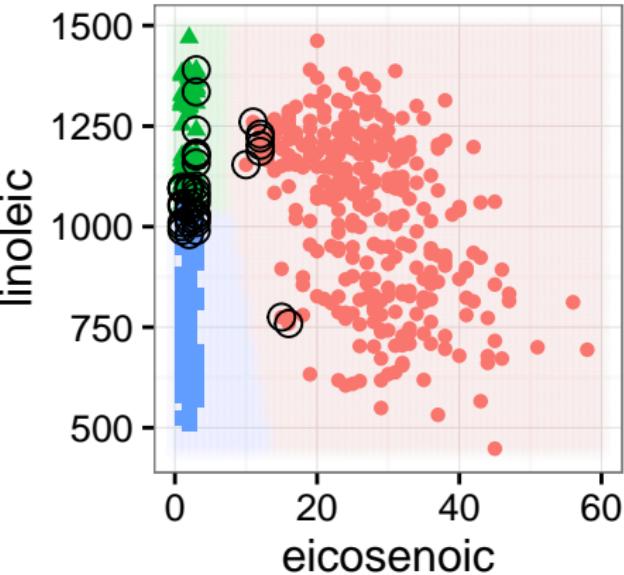
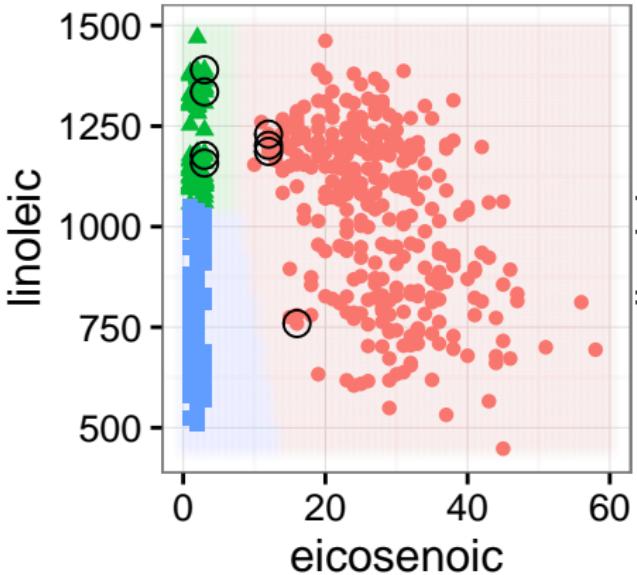
Figure 1: maximum margin

- Hyperplane is defined by a subset of the points, the **support vectors**
- Distance between these is what is maximized



Support vectors for the olive oil classification

M



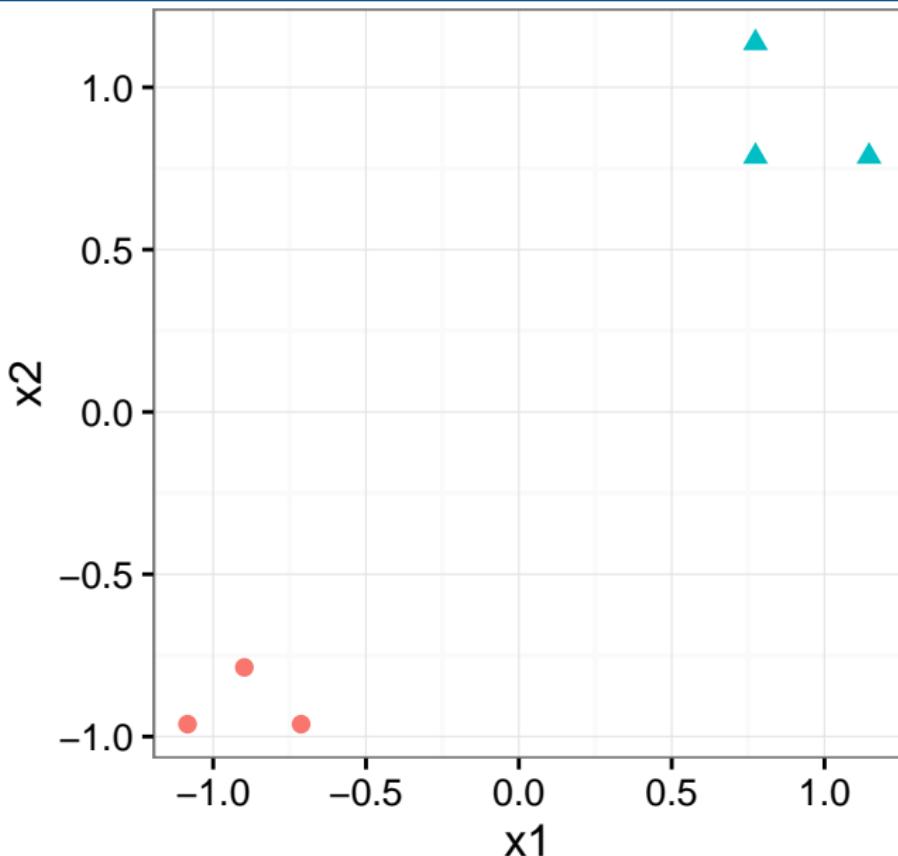
Simulation example

M

```
##           x1          x2  y
## 1  0.7741756  0.7869346  1
## 2  1.1457799  0.7869346  1
## 3  0.7741756  1.1366833  1
## 4 -0.8980437 -0.7869346 -1
## 5 -1.0838459 -0.9618089 -1
## 6 -0.7122416 -0.9618089 -1
```

Simulation example

M



Fit the classifier



```
df.svm <- svm(y~., data=df, kernel="linear")
df.svm$SV
df.svm$index
df.svm$coefs

##           x1           x2
## 1  0.7741756  0.7869346
## 4 -0.8980437 -0.7869346
## 6 -0.7122416 -0.9618089

## [1] 1 4 6

##          [,1]
## [1,]  0.3094284
## [2,] -0.1404977
## [3,] -0.1689307
```

Calculate

M

$$b = \sum_{i=1}^s (\alpha_i y_i) x_i$$

```
t(as.matrix(df.svm$coefs))%*%df.svm$SV
```

```
##           x1          x2  
## [1,] 0.4860445 0.5165415
```

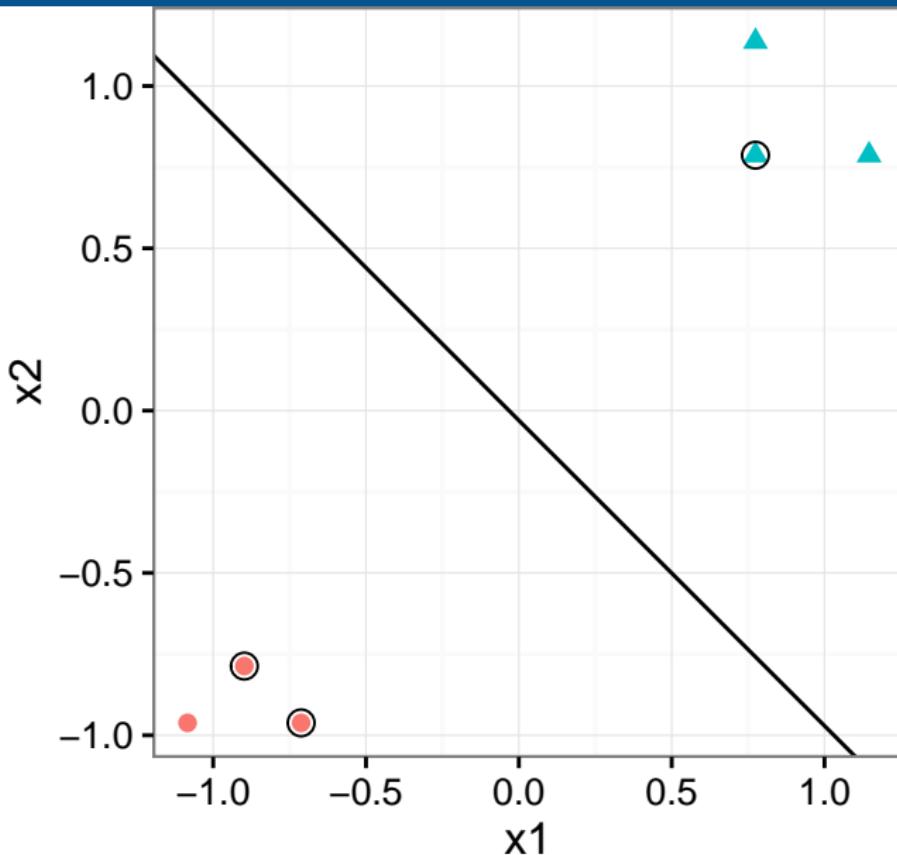
$$\{x : 0.486x_1 + 0.517x_2 + b_o = 0\}$$

```
df.svm$rho
```

```
## [1] -0.03010995
```

Look at it

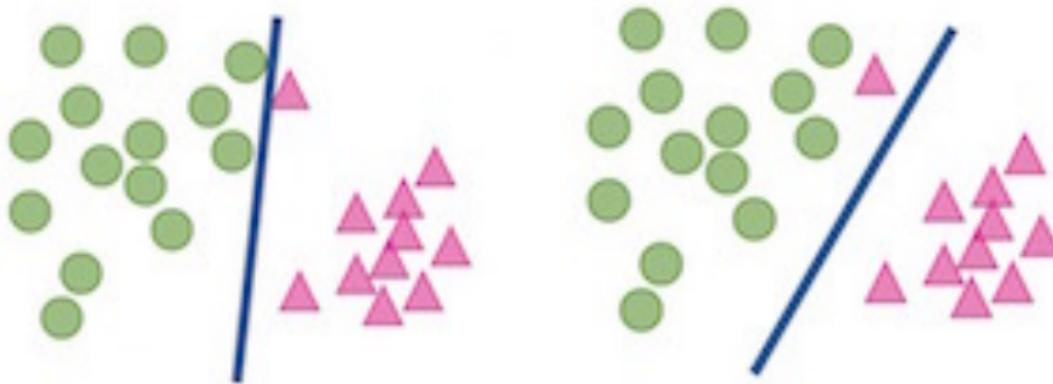
M



Non-separable, and outliers

M

- Outliers can overly influence a strict boundary



Non-separable

- estimated by maximizing margin $M = 2/||b||$
- subject to $\sum_1^p b_i^2 = 1$, $y_i(x_i^T b + b_o) \geq M(1 - \epsilon_i)$, $i = 1, \dots, n$, where $b = \sum_{i=1}^s (\alpha_i y_i) x_i$
- $\epsilon_i \geq 0$, and $\sum_1^n \epsilon_i < C$ where C is a non-negative tuning parameter

Nonlinear separability

- Variables (x_1, \dots, x_p) could be expanded to include (x_1^2, \dots, x_p^2)
- then proceed with building classifier in the expanded space
- maximize margin $M = 2/\|b\|$ subject to $\sum_1^p b_{1i}^2 + \sum_1^p b_{2i}^2 = 1$,
 $y_i((x_i^2)^T b_2 + x_i^T b_1 + b_o) \geq M(1 - \epsilon_i)$

- Because $b = \sum_{i=1}^s (\alpha_i y_i) x_i$
- $y_i(x_i^T b + b_o)$ can be written as
- $y_i(\alpha_i x_i^T x_i + b_o)$
-
- $x_i^T x_i$ can be wrapped into a kernel function $K(x_i^T x_i)$ which enables building nonlinear boundaries

Common kernels

M

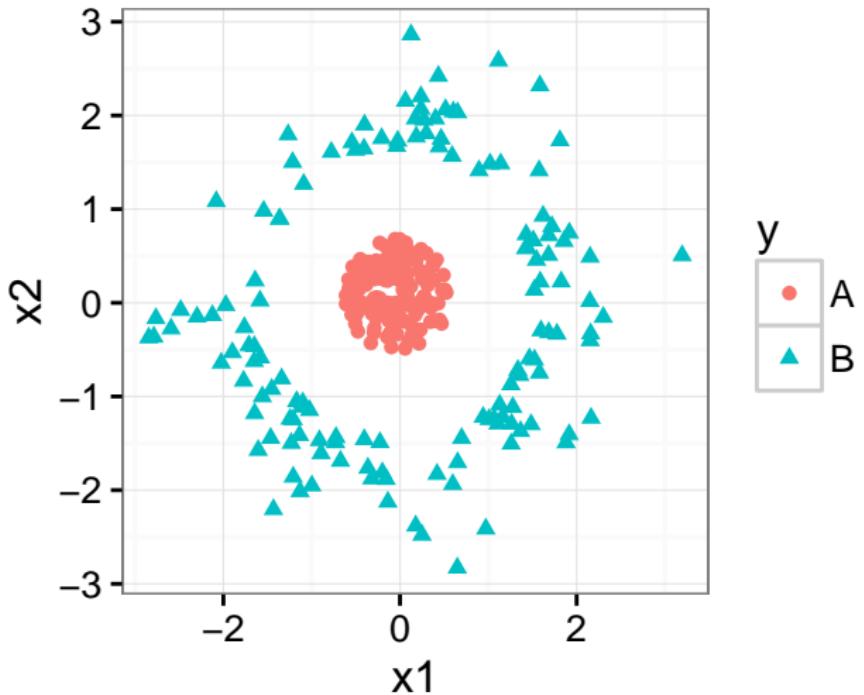
$$K(\mathbf{x}_i, \mathbf{x}_j)$$

Name	Function
Polynomial	$(\ \mathbf{x}_i^T \mathbf{x}_j\ + d)^p$
Gaussian radial basis	$\exp(-\ \mathbf{x}_i - \mathbf{x}_j\ ^2 / 2\sigma^2)$
Sigmoid	$\tanh(a \ \mathbf{x}_i^T \mathbf{x}_j\ + d)$

Figure 3: kernels

Examples

M



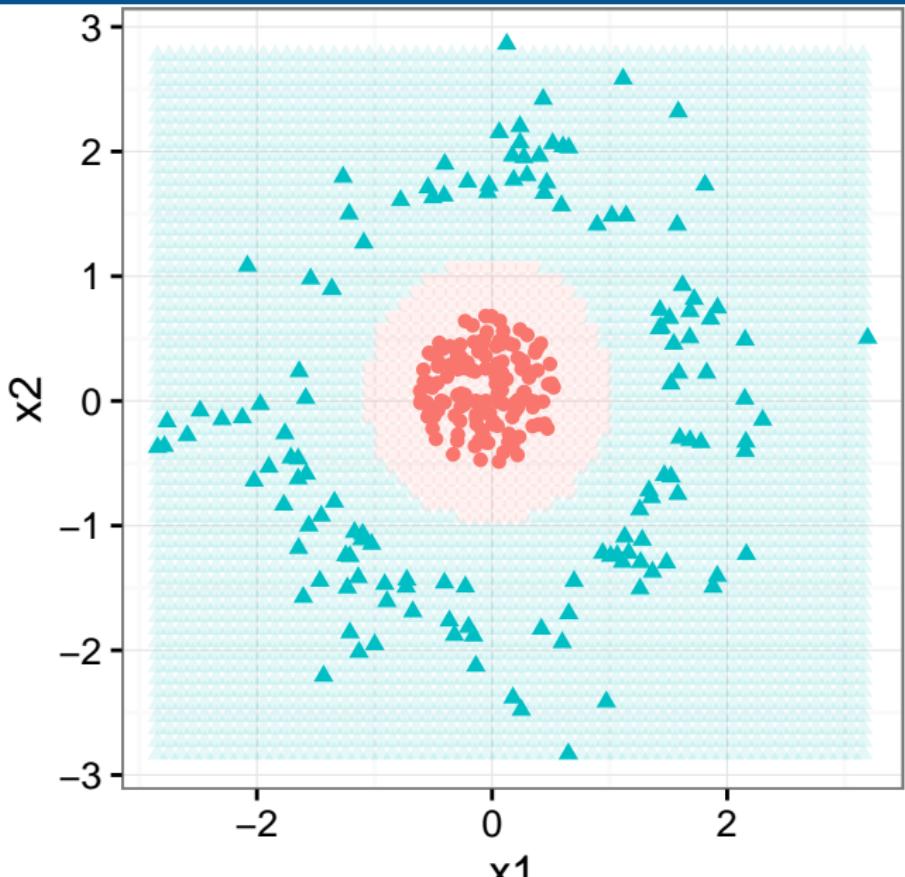
Examples



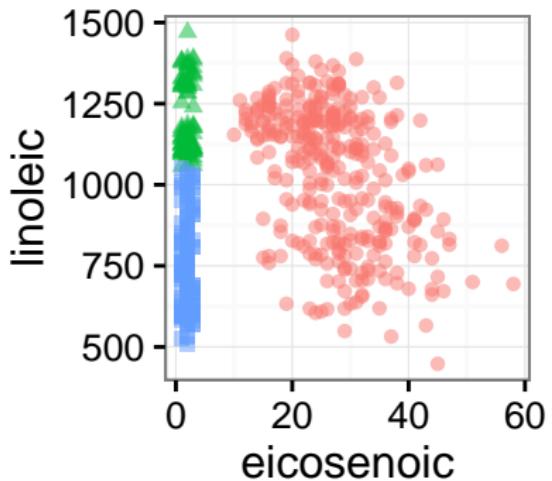
```
##  
## Call:  
## svm(formula = y ~ ., data = df)  
##  
##  
## Parameters:  
##      SVM-Type: C-classification  
##      SVM-Kernel: radial  
##      cost: 1  
##      gamma: 0.5  
##  
## Number of Support Vectors: 15  
##          x1          x2  
## 138 -0.33053299 -0.42878464  
## 139 -0.47702990 -0.30511245  
## 141  0.41465396  0.46079848
```

Examples

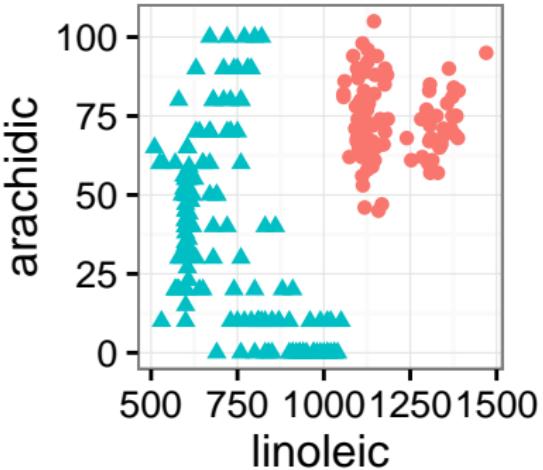
M



Examples: olive oils



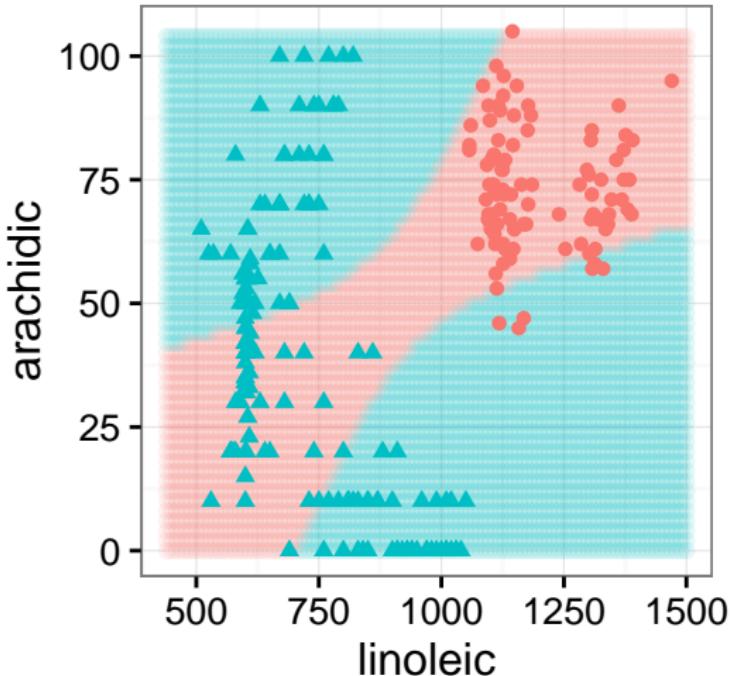
● south ▲ sardinia ■ north



● sardinia ▲ north

Examples: olive oils

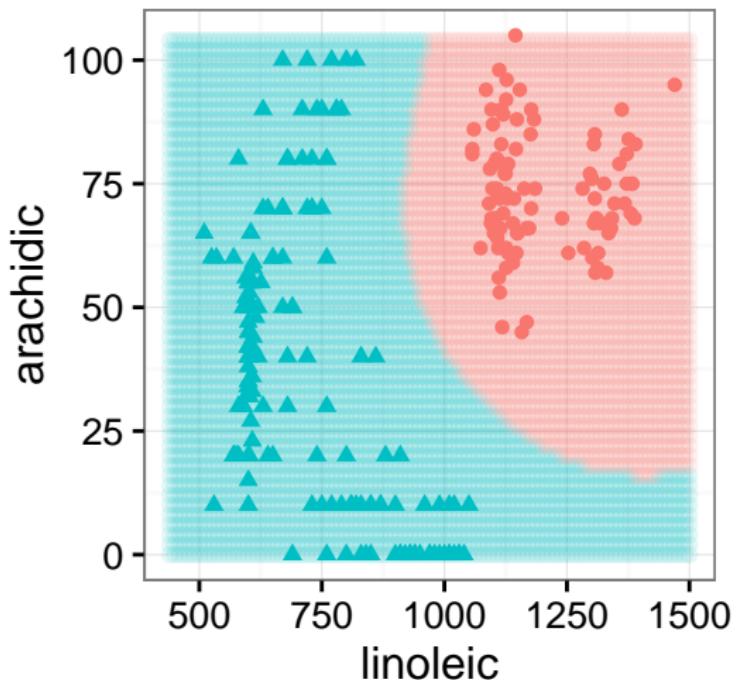
```
olive.svm <- svm(region~linoleic + arachidic, data=olive.sub,  
kernel="polynomial", degree=2)
```



Examples: olive oils

M

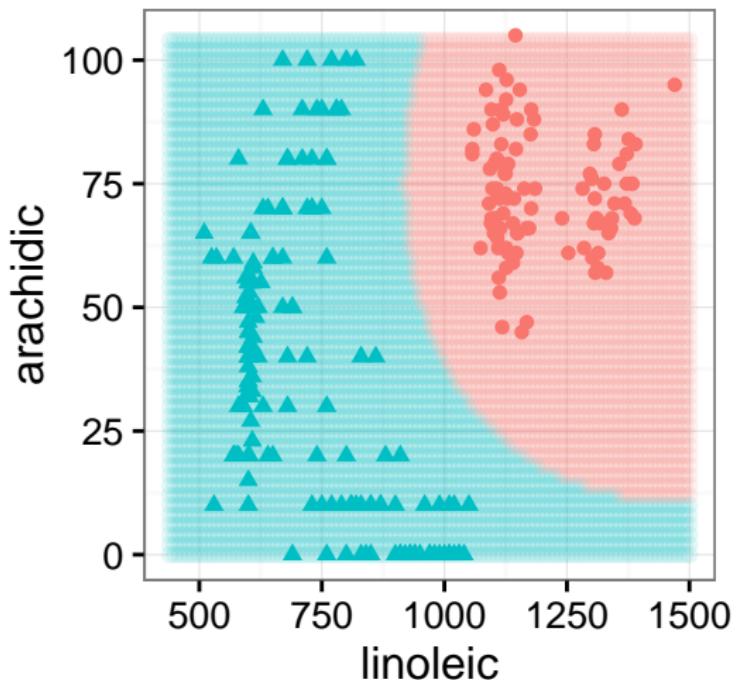
```
olive.svm <- svm(region~linoleic + arachidic, data=olive.sub,  
kernel="radial")
```



Examples: olive oils

M

```
olive.svm <- svm(region~., data=olive.sub[,-c(2,10)],  
kernel="radial")
```



- For high-dimension low sample size problems (more variables than samples) SVM cannot properly estimate the coefficients for the separating hyperplane
- Even fitting a linear kernel is a problem
- The same is true for LDA
- Dimension reduction, or penalisation, needs to be used in association with the classifiers

- How do boundaries look in high dimensions?
- This video is a basic intro to visualising the SVM model
- This video shows boundaries for a radial kernel fitted to 3D data
- This video shows boundaries for a polynomial kernel fitted to 5D data
- This video another video illustrating looking at boundaries for an SVM model

The available procedures are:

- One-vs-one (also called all-vs-all) or one-vs-all.
- One-vs-one, makes all pairwise classifiers. Predictions are made by a voting scheme.
- One-vs-all does A vs not A, B vs not B, ... Predictions are made by picking the best “positive” class (A, B, ...).

Share and share alike



This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 United States License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.