Linear Classification

Anita Faul

Laboratory for Scientific Computing, University of Cambridge

What are the distinguishing features?

Distinguishing between objects is called *classification*

Cat or Dog?

Features

- A child is not told what is different between a cat and a dog.
- It is told that is a cat and this is a dog.
- At most it is told: That goes meow and this goes woof.
- It has to work out itself how to distinguish them.
- What are cat features and what are dog features?
- We humans do that easily in the first years of our life.

Classification - Feature Categories

Categories of features are:

- Boolean (binary, answerable by yes or no)
  - Is it red?
  - Is the person tall?
  - Do we have a storm?
- Discrete with multiple values (categorical, ordinal)
  - What colour is it?
  - Is the person short, medium built or tall?
  - What number is the wind on the Beaufort scale?
- Continuous (real-valued)
  - What is the wavelength?
  - What is the person's height?
  - What is the wind speed?

Examples of Classification

Examples of classification problems:

- text categorization (e.g., spam filtering),
- fraud detection,
- optical character recognition,
- machine vision (e.g., face detection),
- natural-language processing (e.g., spoken language understanding),
- market segmentation (e.g., predict if customer will respond to promotion),
- bio informatics (e.g., classify proteins according to their function).
- Any suggestion?

Fisher's iris data

Iris setosa

Iris versicolor

Iris virginica

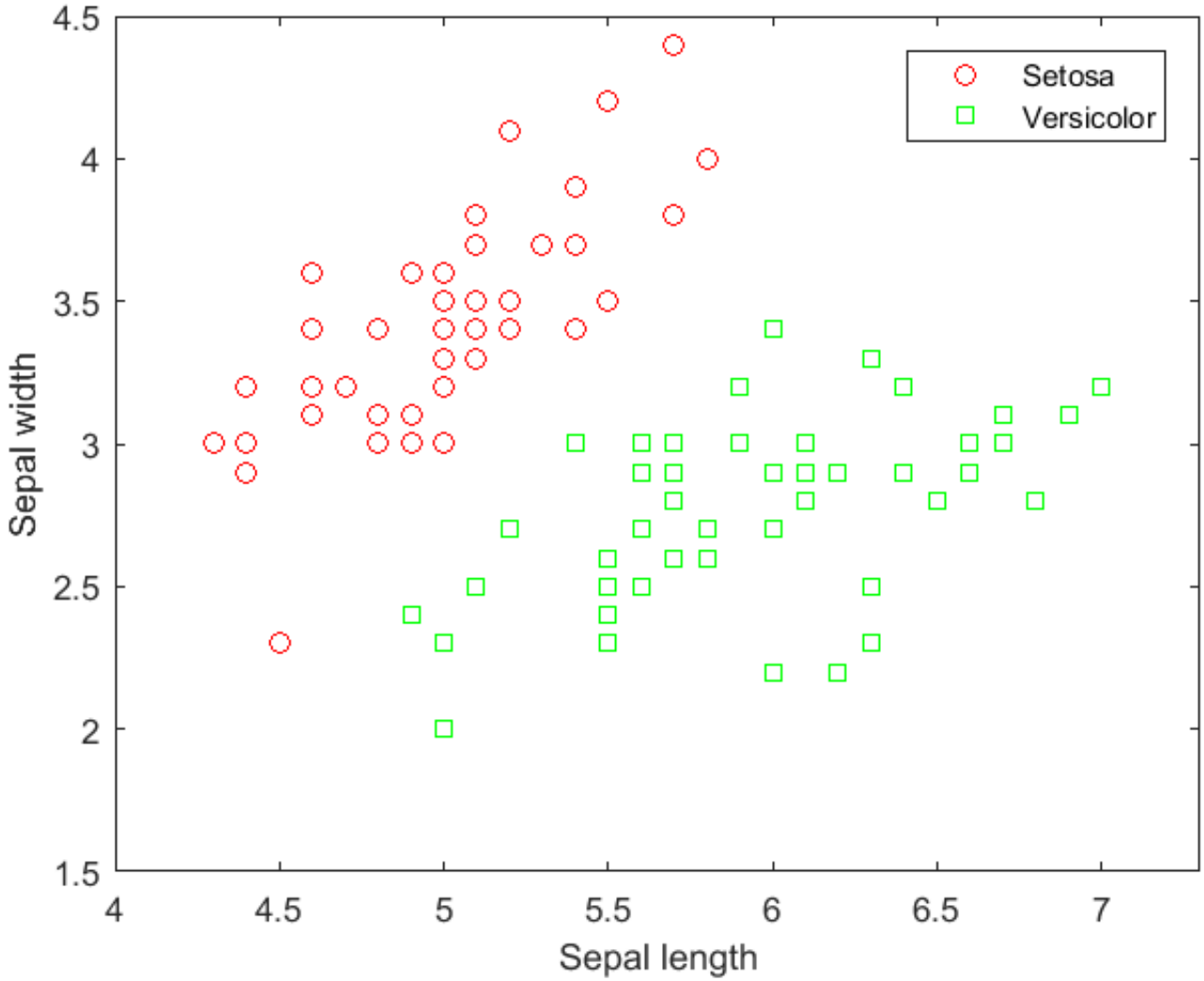
- used by Fisher in 1936,
- 50 samples of each species, which are in classes C_0, C_1 and C_2,
- 4 measured features: sepal length, sepal width, petal length, petal width.

Binary Classification

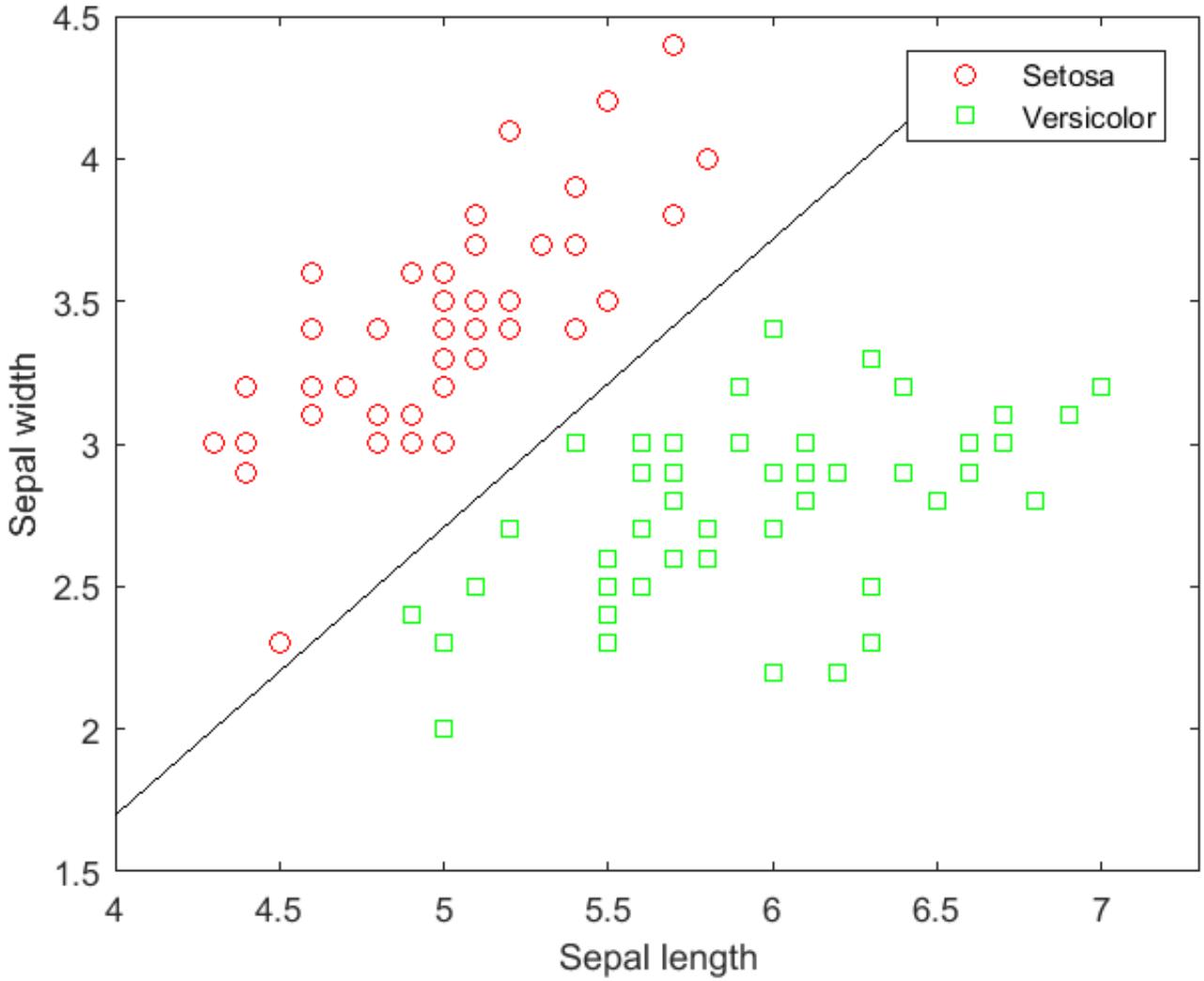
Binary Classification:

- N_0 samples belonging to class C_0,
- N_1 samples belonging to class C_1,
- altogether $N = N_0 + N_1$ samples.

Visual representation of the data



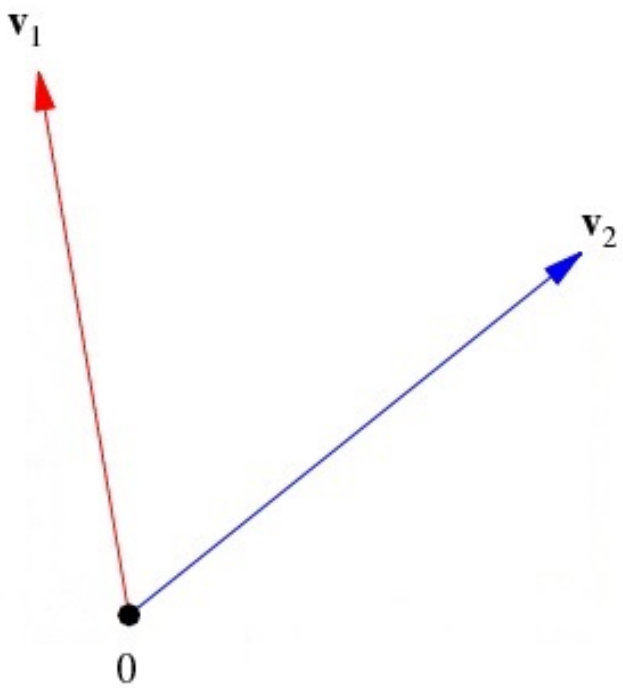
Line


Setosa
Versicolor
Sepal width
Sepal length
4.5
4
3.5
3
2.5
2
1.5
4
4.5
5
5.5
6
6.5
7

Data as vectors

Each sample has two features written as vector

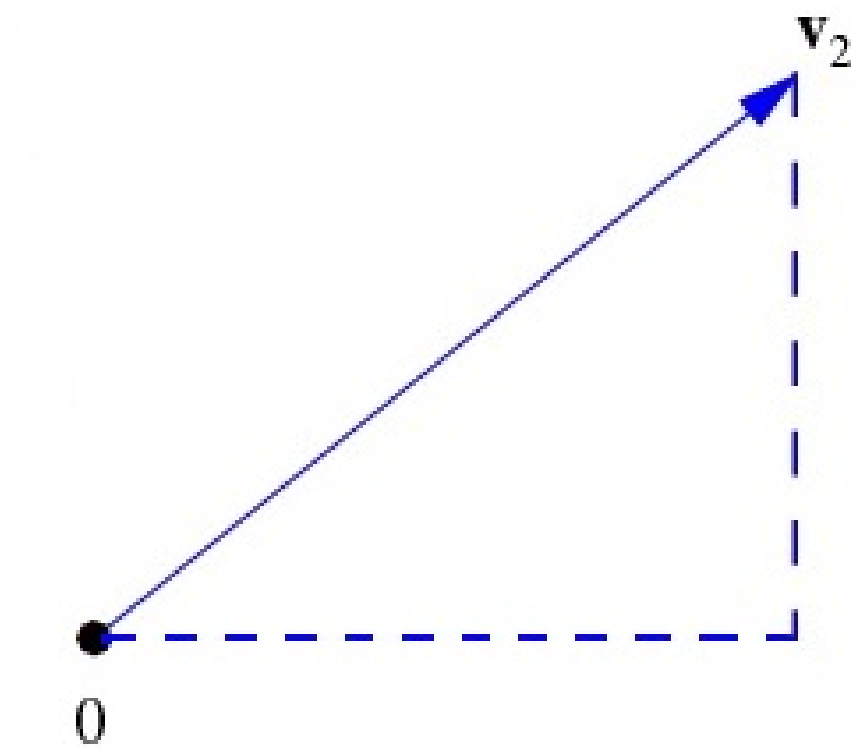
$$\mathbf{v}_1 = \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} \quad \text{and} \quad \mathbf{v}_2 = \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix}$$



Vector Length

The length of a vector is the square root of the sum of the squares of the coordinates.

$$\|\mathbf{v}_2\| = \sqrt{v_{21}^2 + v_{22}^2}.$$



Inner product

- Inner product, scalar product, dot product:

$$\mathbf{v}_1 \bullet \mathbf{v}_2 = \langle \mathbf{v}_1, \mathbf{v}_2 \rangle = v_{11}v_{21} + v_{12}v_{22}.$$

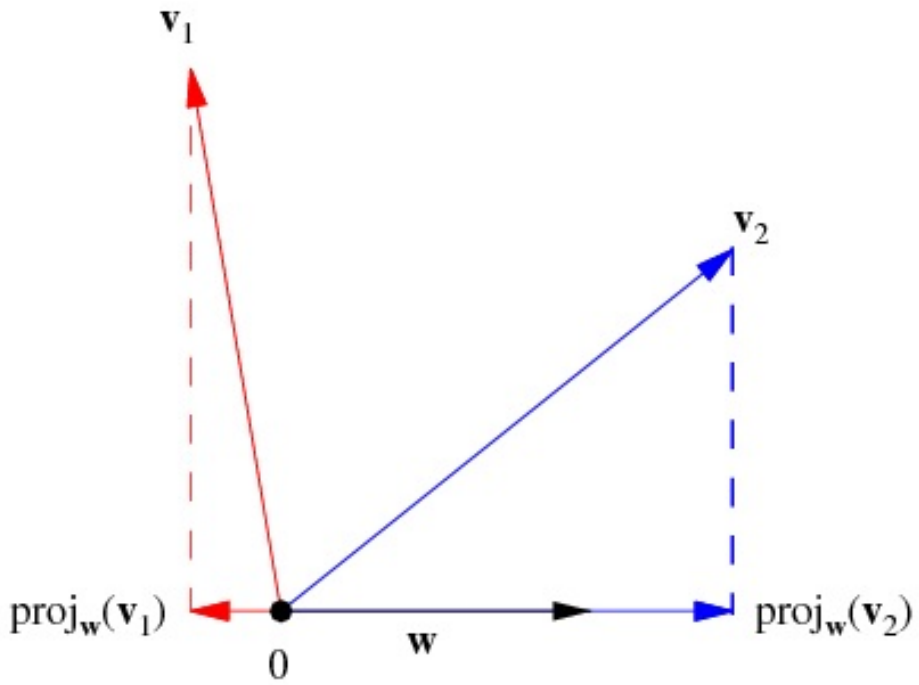
- It can be viewed as matrix product:

$$\mathbf{v}_1 \bullet \mathbf{v}_2 = \mathbf{v}_1^T \mathbf{v}_2 = \begin{pmatrix} v_{11} & v_{12} \end{pmatrix} \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix} = v_{11}v_{21} + v_{12}v_{22}.$$

- A 1×2 matrix times a 2×1 matrix is a 1×1 matrix.

Projection onto line

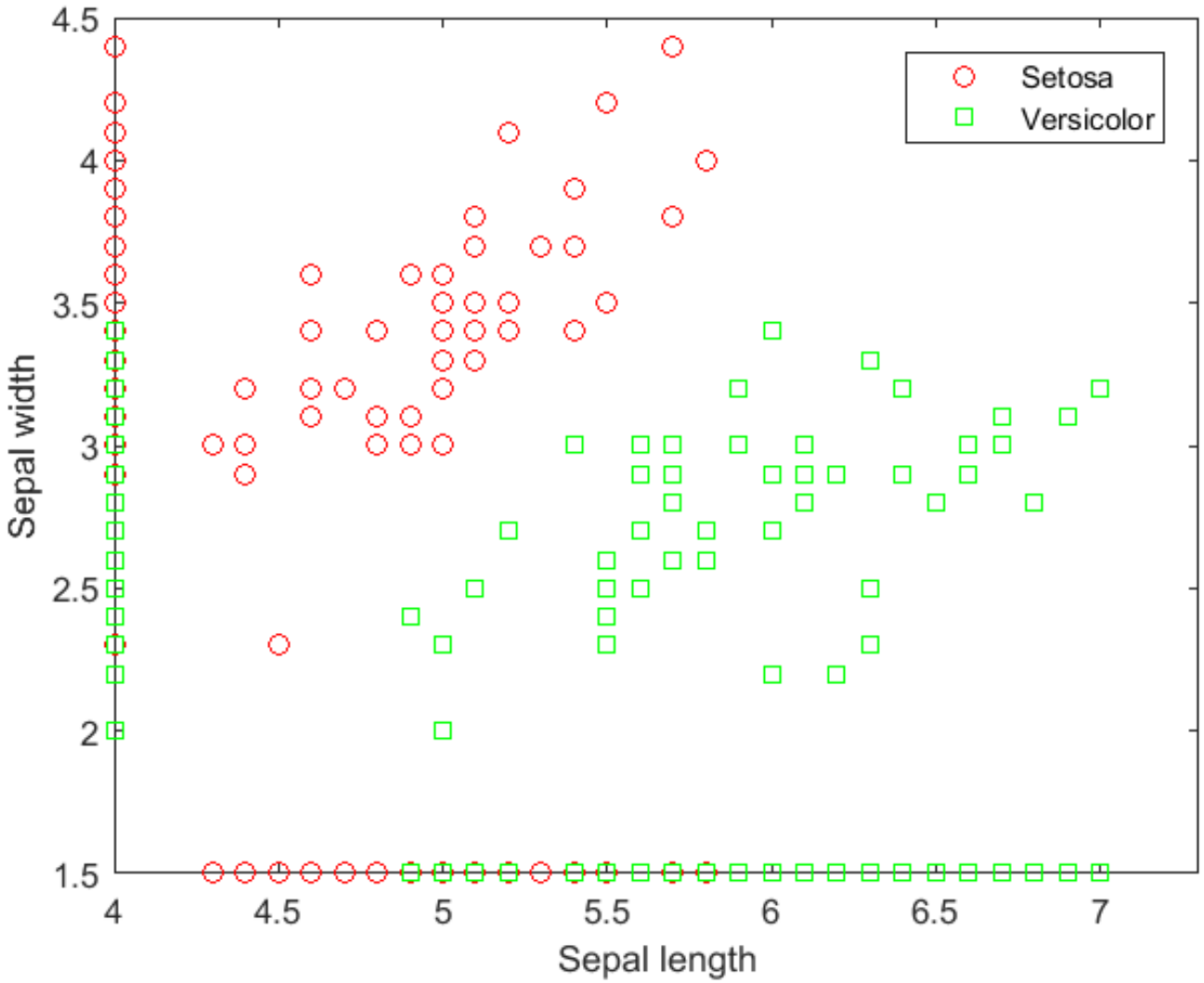
- Reduce 2D problem to 1D by projecting onto a line



$$\text{proj}_{\mathbf{w}}(\mathbf{v}) = \frac{\mathbf{w}^T\mathbf{v}}{\|\mathbf{w}\|^2}\mathbf{w}$$

- Seek *separation threshold* b (also known as *bias*) such that for $\mathbf{v}$
  - $\mathbf{w}^T\mathbf{v} < b \quad \Rightarrow \quad \mathbf{v} \in C_0$,
  - $\mathbf{w}^T\mathbf{v} > b \quad \Rightarrow \quad \mathbf{v} \in C_1$.

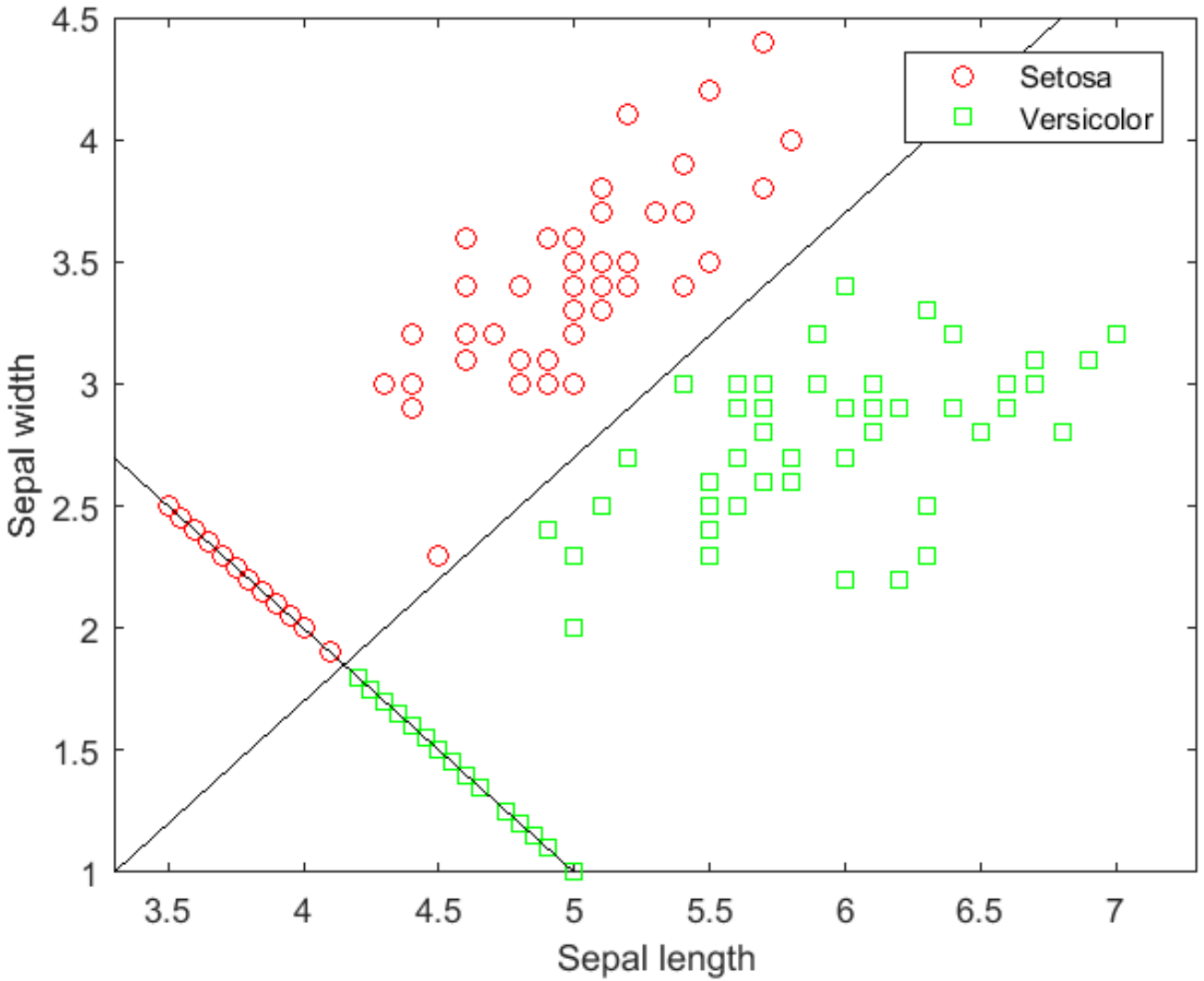
Projection onto Axes



Projection onto Line

- Line vertical to the projection line and through b should separate the classes.
- Guessed $\mathbf{w}$, the direction of the line of projection.
- Guessed separation threshold b.

Projection onto Line



Projection onto Line

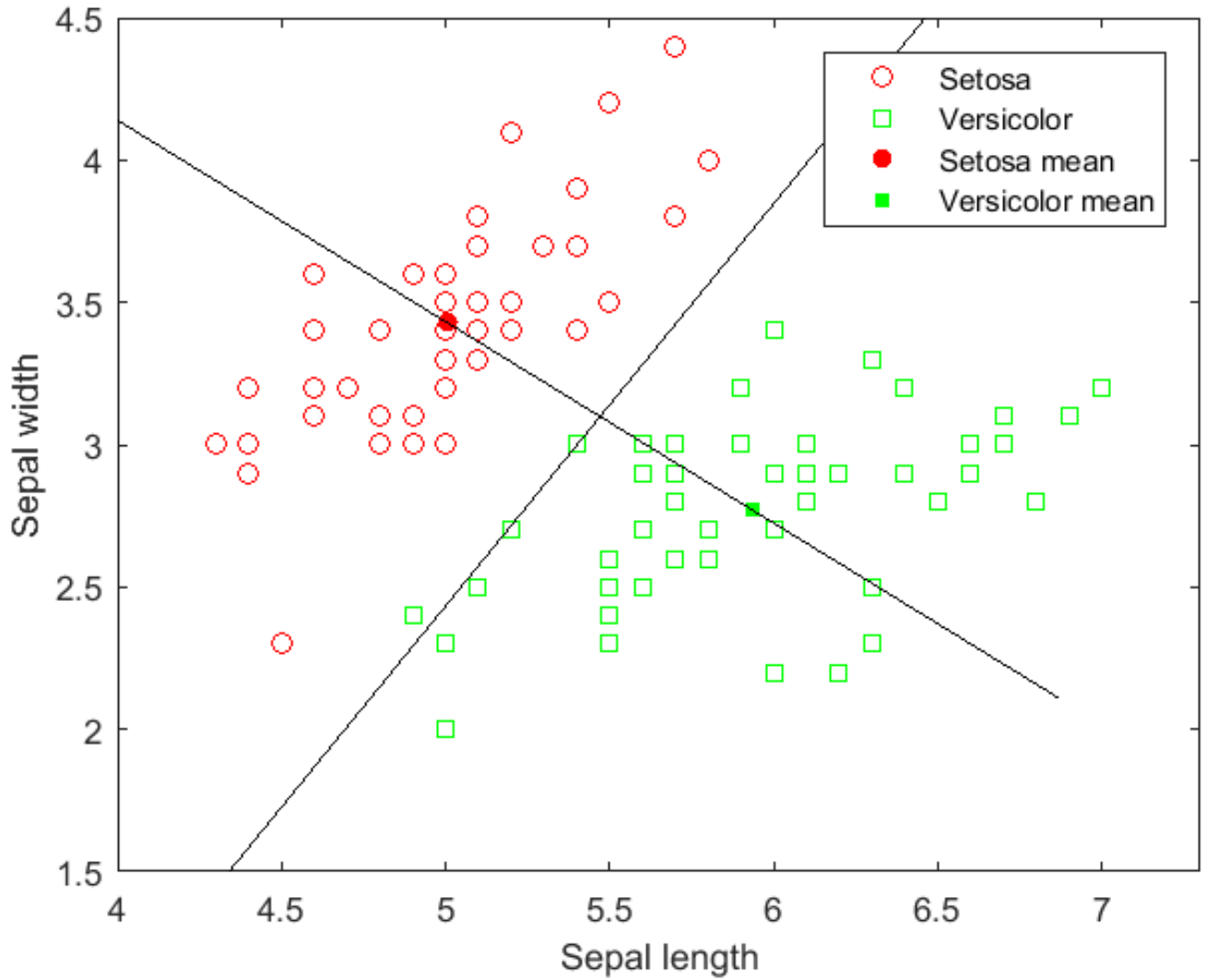
- Use line between sample means for projection
- and midpoint between sample means as separation threshold.

Projection onto Line

- Sample mean of class C_i is $\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{\mathbf{v} \in C_i} \mathbf{v}$, $i = 0, 1$.
- Line through means is parametrized by $\mathbf{l}(a) = \boldsymbol{\mu}_0 + a(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$.
- Midpoint given by $a = 1/2$:

$$\frac{1}{2}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1).$$

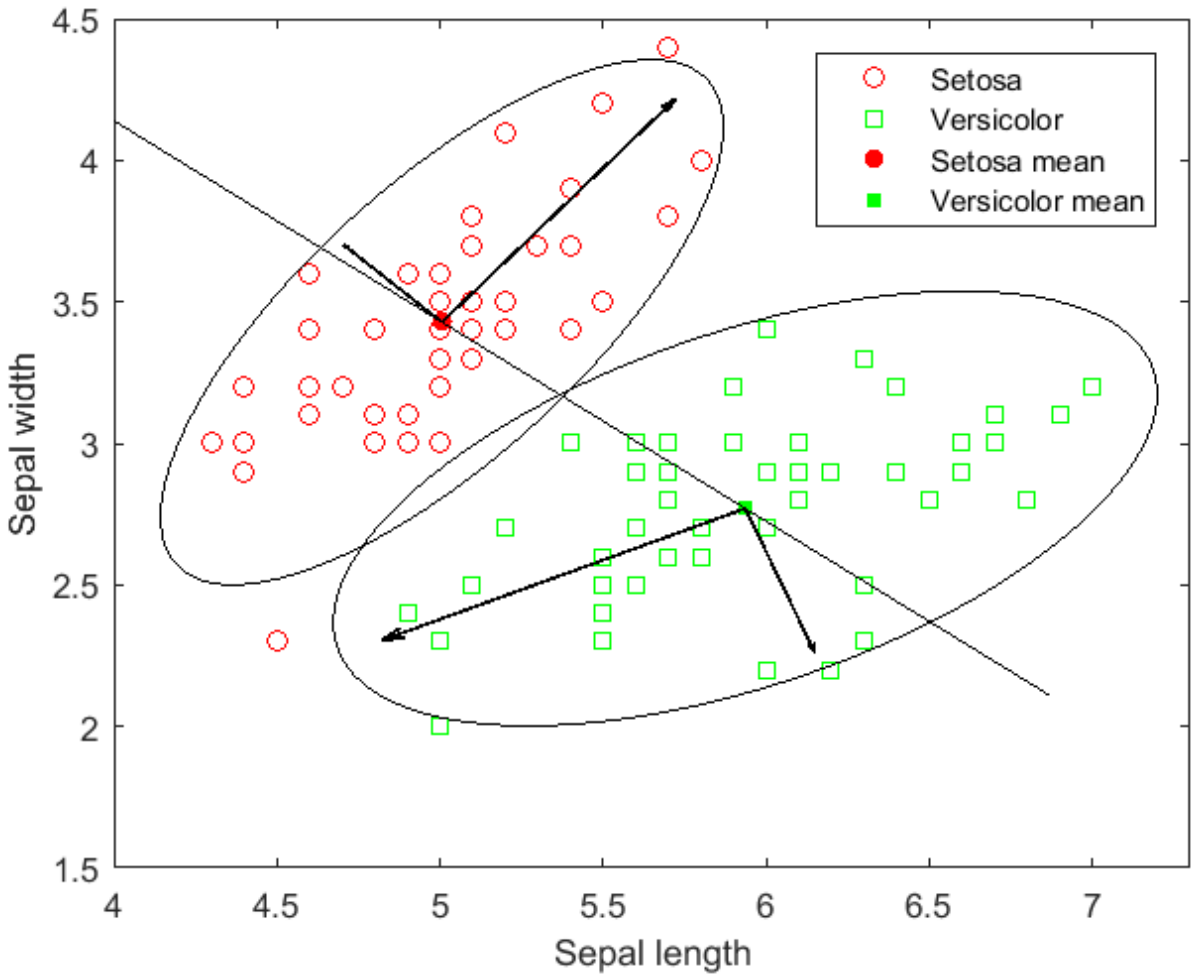
Projection onto Line



Projection onto Line

This fails to take into account the variance within classes, that is how much the samples differ from the mean.

Projection onto Line

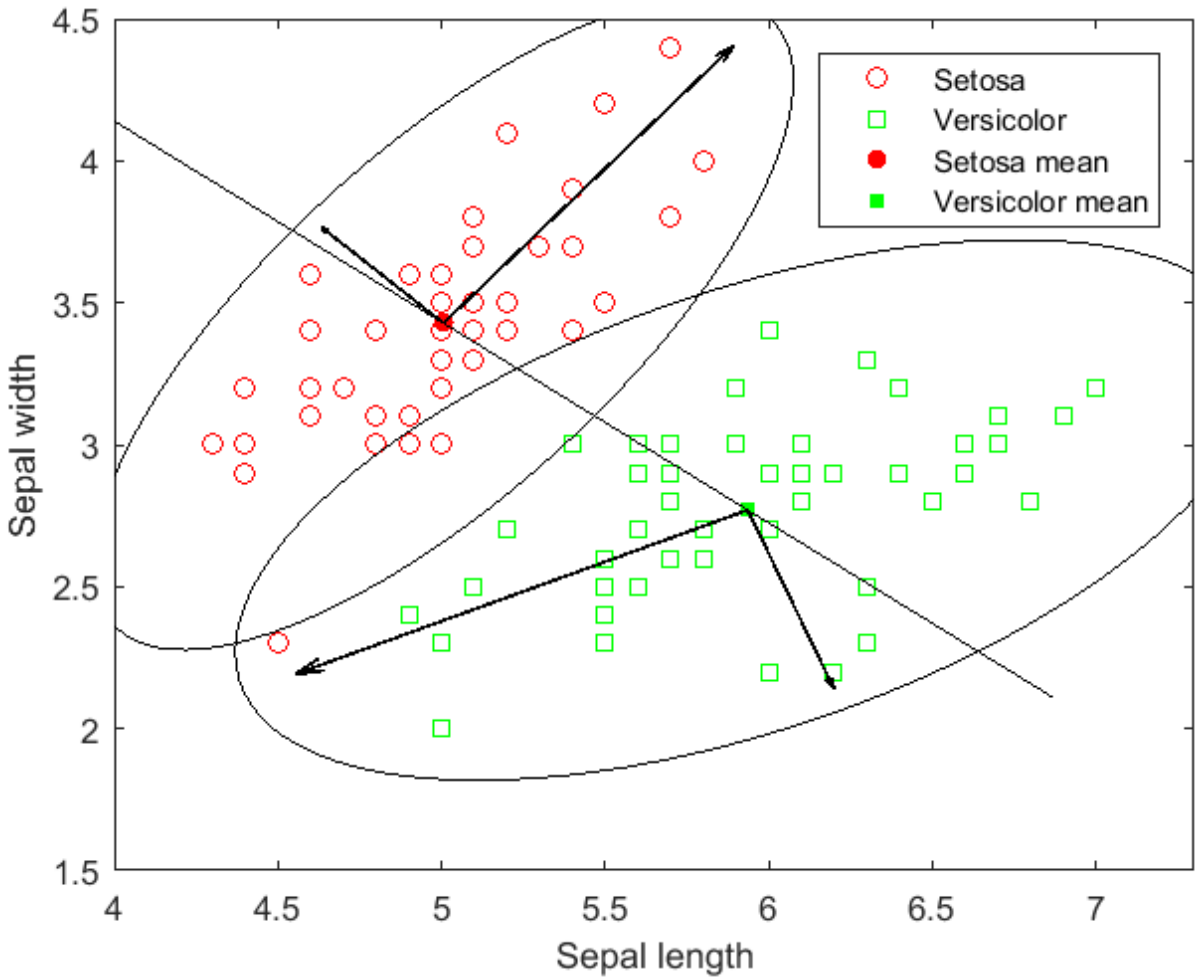


95% confidence ellipses

Projection onto Line

Choose value where the confidence ellipses intersect - you may suggest.

Projection onto Line



99% confidence ellipses

Projection onto Line

- The ellipses have different axes.
- This has the effect that the projections of the data of one class are more spread out than the data of the other class.

Projection onto Line

Goal: Find line of projection such that

- The projected means of each class are as far apart as possible,
- The projected samples of each class are as close together as possible.

Projection onto Line

Recall:

- Sample mean of C_i is $\boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{\mathbf{v} \in C_i} \mathbf{v}$.

We define:

- Sample covariance of C_i is $\boldsymbol{\Sigma}_i = \frac{1}{N_i} \sum_{\mathbf{v} \in C_i} (\mathbf{v} - \boldsymbol{\mu}_i)(\mathbf{v} - \boldsymbol{\mu}_i)^T$.

Outer Product

- Outer product:

$$\mathbf{v}_1\mathbf{v}_2^T = \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix} \begin{pmatrix} v_{21} & v_{22} \end{pmatrix} = \begin{pmatrix} v_{11}v_{21} & v_{11}v_{22} \\ v_{12}v_{21} & v_{12}v_{22} \end{pmatrix}.$$

- It can be viewed as matrix product.
- A 2×1 matrix times a 1×2 matrix is a 2×2 matrix.

Projection onto Line

Note:

- If M is the number of features ($M = 2$),
- $\boldsymbol{\mu}_i$ is a vector with M elements.
- The j-th entry is the mean of the j-th feature of samples in class C_i.

Projection onto Line

Note:

- $\mathbf{\Sigma}_i$ is an $M \times M$ matrix.
- The (j, j) diagonal entry is the variance of the j-th feature of samples in class C_i.
- The variance describes how spread the values of a feature are.
- The (j, k) off-diagonal entry is the covariance between the j-th and k-th feature of samples in class C_i.
- The covariance describes how much different features influence each other.

Projection onto Line

- Let $\mathbf{w}$ be the direction of the line of projection.
- (The location of the line of projection can be neglected, since it cancels in the calculations.)
- The mean of the projected samples in class C_i is given by $\mathbf{w}^T \boldsymbol{\mu_i}$.
- The variance of the projected samples in class C_i is given by $\mathbf{w}^T \boldsymbol{\Sigma}_i \mathbf{w}$.

Projection onto Line

We seek $\mathbf{w}$ such that

- $\|\mathbf{w}^T\boldsymbol{\mu}_0 - \mathbf{w}^T\boldsymbol{\mu}_1\|^2$ is as large as possible, while
- $\mathbf{w}^T\boldsymbol{\Sigma}_0\mathbf{w} + \mathbf{w}^T\boldsymbol{\Sigma}_1\mathbf{w}$ is as small as possible.

Fisher's discriminant analysis

Fisher defined the separation between classes achieved by projecting onto the line given by $\mathbf{w}$ as

$$s(\mathbf{w}) = \frac{\|\mathbf{w}^T\boldsymbol{\mu}_0 - \mathbf{w}^T\boldsymbol{\mu}_1\|^2}{\mathbf{w}^T\boldsymbol{\Sigma}_0\mathbf{w} + \mathbf{w}^T\boldsymbol{\Sigma}_1\mathbf{w}}.$$

This function needs to be maximized.

Fisher's discriminant analysis

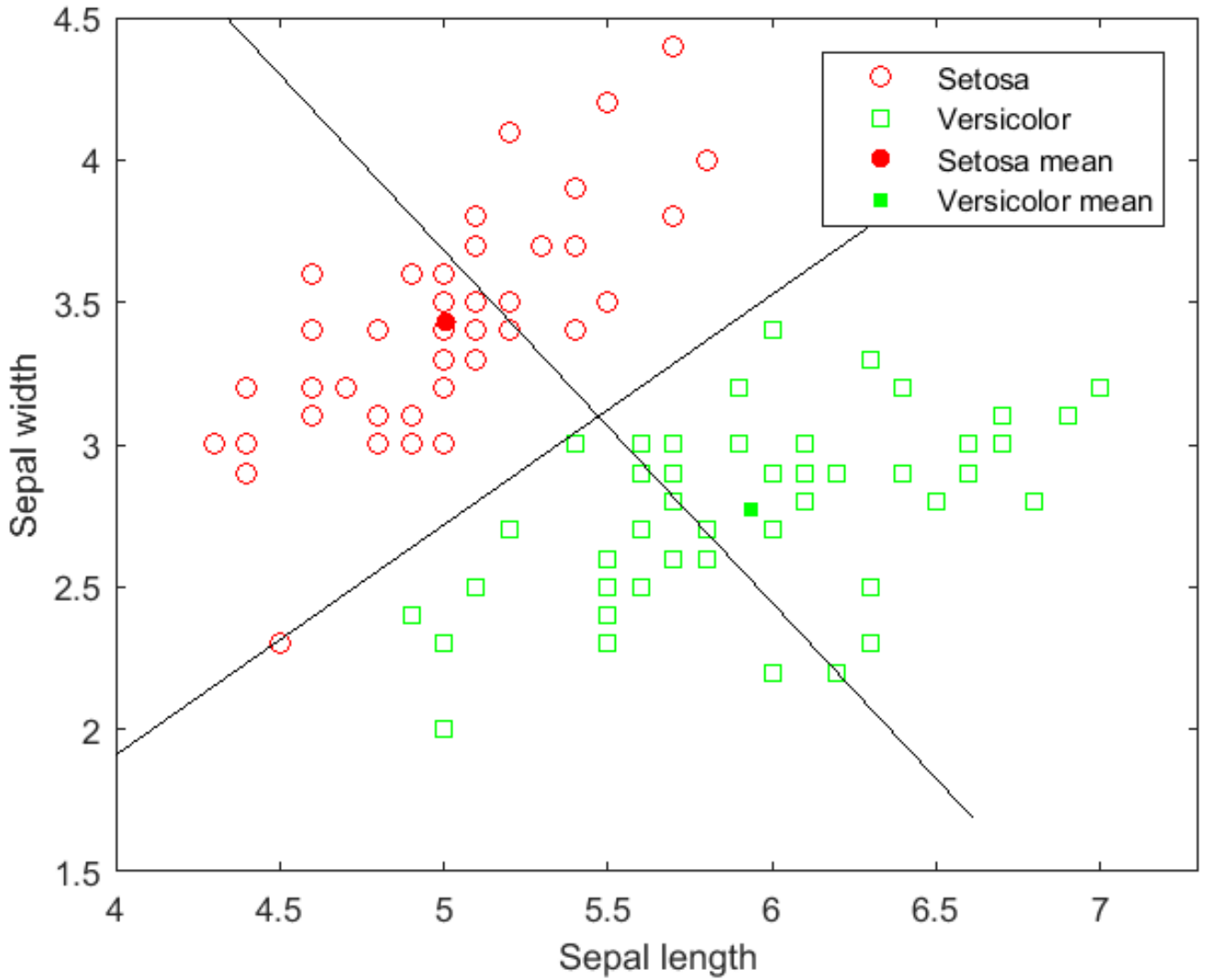
- $\|\mathbf{w}^T\boldsymbol{\mu}_0 - \mathbf{w}^T\boldsymbol{\mu}_1\|^2 = \mathbf{w}^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T\mathbf{w}$ is called the variance between projected classes or the between-class scatter of the projected samples.
- $\mathbf{w}^T\boldsymbol{\Sigma}_0\mathbf{w} + \mathbf{w}^T\boldsymbol{\Sigma}_1\mathbf{w}$ is called the variance within projected classes or the within-class scatter of the projected samples.
- Thus we are maximizing the ratio of the variance between projected classes to the variance within projected classes.

Fisher's discriminant analysis

It can be shown (differentiating with respect to each feature and setting to zero) that the maximum separation occurs when $\mathbf{w}$ is a multiple of

$$(\boldsymbol{\Sigma}_0 + \boldsymbol{\Sigma}_1)^{-1} (\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) .$$

Fisher's discriminant analysis



Linear Discriminant Analysis (LDA)

The assumption $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1$ simplifies the calculations. It is known as *Linear Discriminant Analysis (LDA)*.

- $\mathbf{w}$ is set to $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)$ (since scaling can be neglected) and
- the separation threshold is the projection of the midpoint between the means:

$$b = \mathbf{w}^T \frac{1}{2}(\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1) = \frac{1}{2}(\boldsymbol{\mu}_0^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_0 - \boldsymbol{\mu}_1^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_1).$$

- (In the iris example the classes do not have the same covariance.)

Multiple classes

LDA generalizes to more than two classes. Assume there are K classes. Previously we defined:

- $\mathbf{w}^T(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1)^T\mathbf{w}$ as the variance between projected classes or the between-class scatter of the projected samples.
- $\mathbf{w}^T\boldsymbol{\Sigma}\mathbf{w}$ as the variance within projected classes or the within-class scatter of the projected samples.

Multiple classes

- We now define the between-class scatter

$$\boldsymbol{\Sigma}_b = \frac{1}{K} \sum_{i=1}^{K} (\boldsymbol{\mu}_i - \boldsymbol{\mu}) (\boldsymbol{\mu}_i - \boldsymbol{\mu})^T ,$$

  where $\boldsymbol{\mu}$ is the mean of the class means $\boldsymbol{\mu} = \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{\mu}_i$,

- while $\boldsymbol{\Sigma}$ remains the within class scatter.

Multiple classes

The separation between classes in the direction of $\mathbf{w}$ is defined as

$$s(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{\Sigma}_b \mathbf{w}}{\mathbf{w}^T \mathbf{\Sigma} \mathbf{w}}.$$

To separate K classes we need to find $K-1$ directions $\mathbf{w}_1, \ldots, \mathbf{w}_{K-1}$ for which $s(\mathbf{w})$ is maximal. This is equivalent to maximizing

$$S(\mathbf{W}) = \frac{|\mathbf{W}^T \mathbf{\Sigma}_b \mathbf{W}|}{|\mathbf{W}^T \mathbf{\Sigma} \mathbf{W}|},$$

where $\mathbf{W}$ denotes the matrix whose columns are the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_{K-1}$.
($|\mathbf{A}|$ denotes the determinant of the matrix.)

Multiple classes

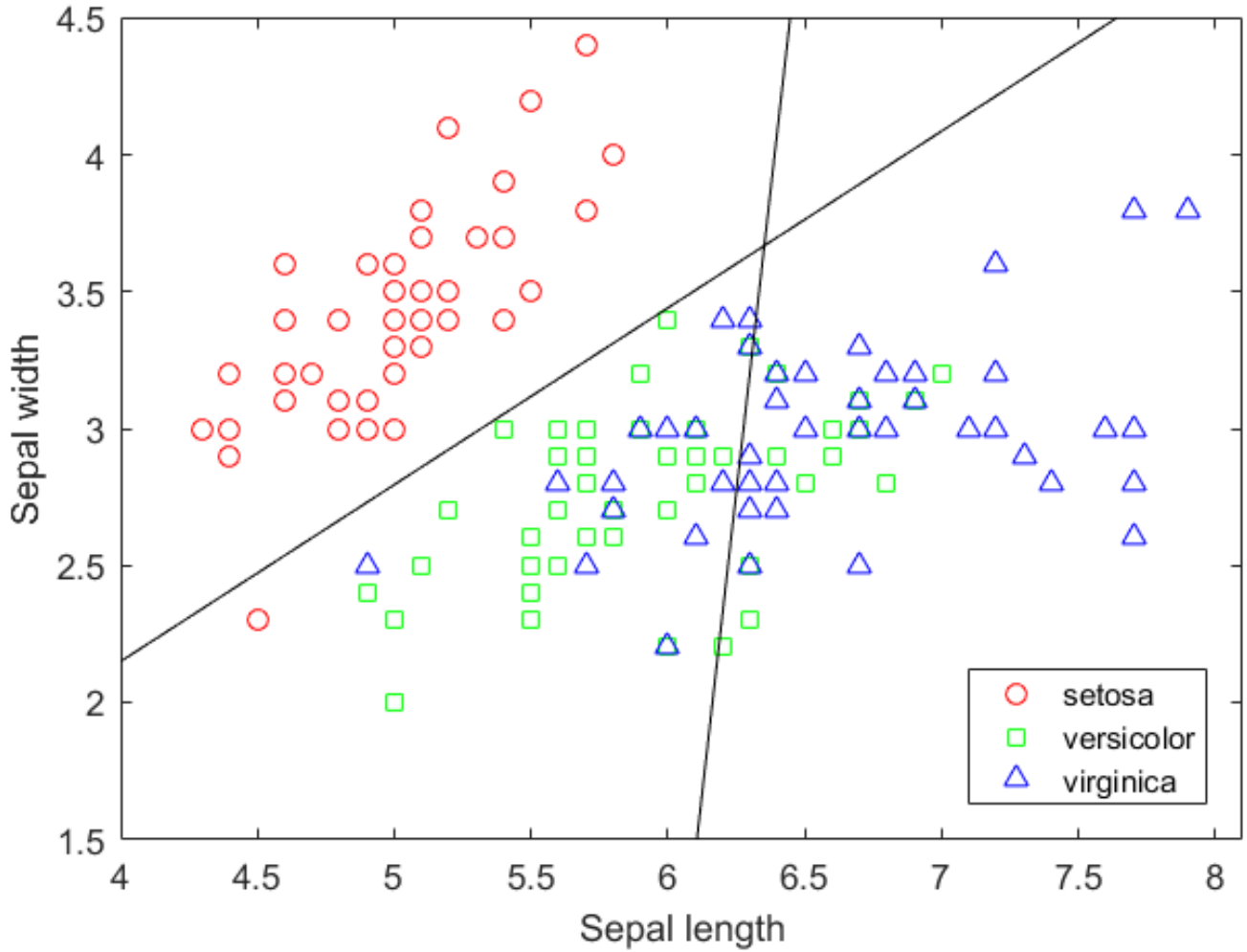
If the number of features M is greater or equal to the number of classes K, it can be shown that $S(\mathbf{W})$ is maximal if $\mathbf{w}_1, \ldots, \mathbf{w}_{K-1}$ are the generalized eigenvectors corresponding to the $K-1$ largest generalized eigenvalues of the generalized eigenvalue problem

$$\boldsymbol{\Sigma}_B \mathbf{w} = \lambda \boldsymbol{\Sigma} \mathbf{w}.$$

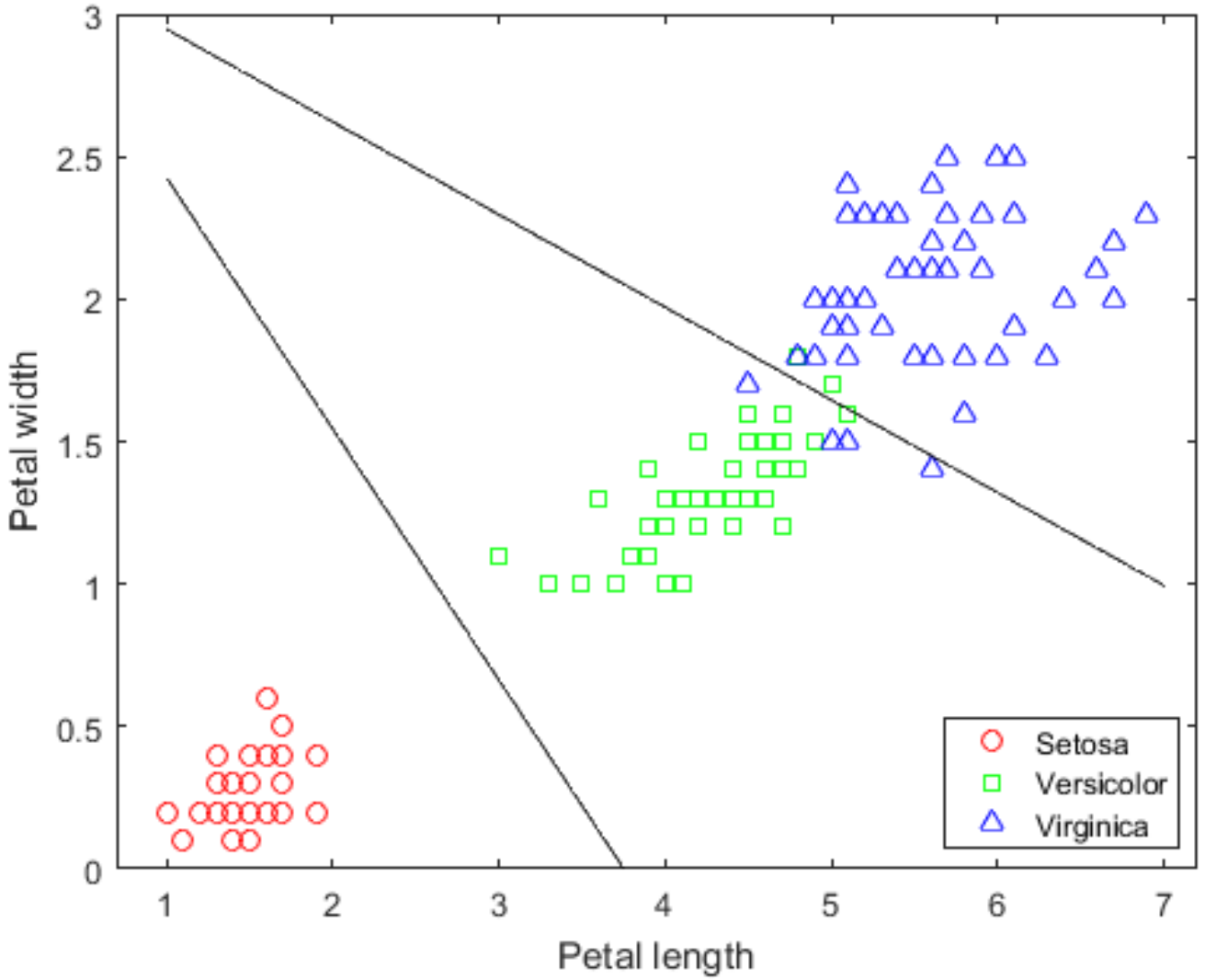
Implemented in Matlab as

```
ClassificationDiscriminant.fit
```

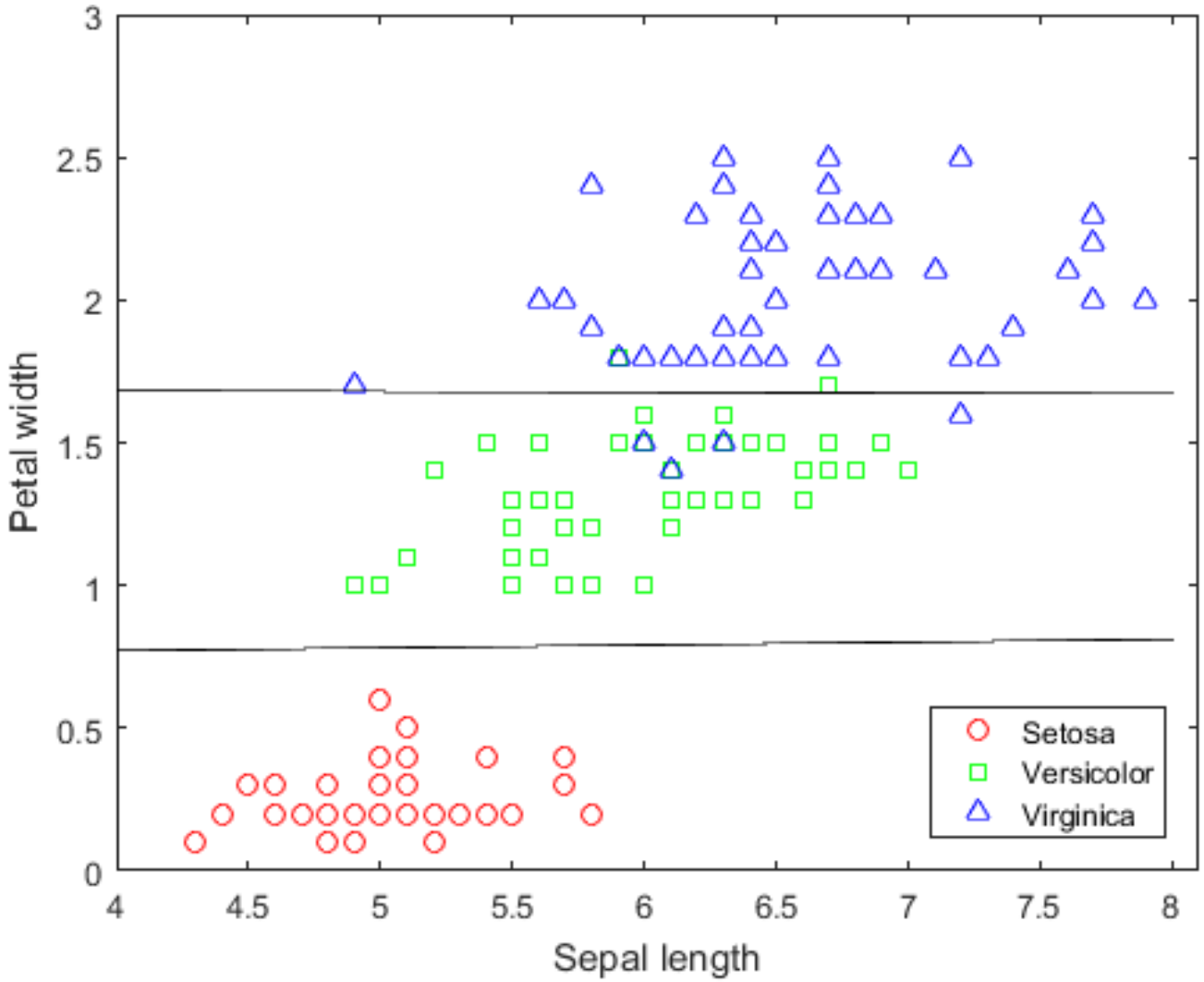
LDA - multiple classes



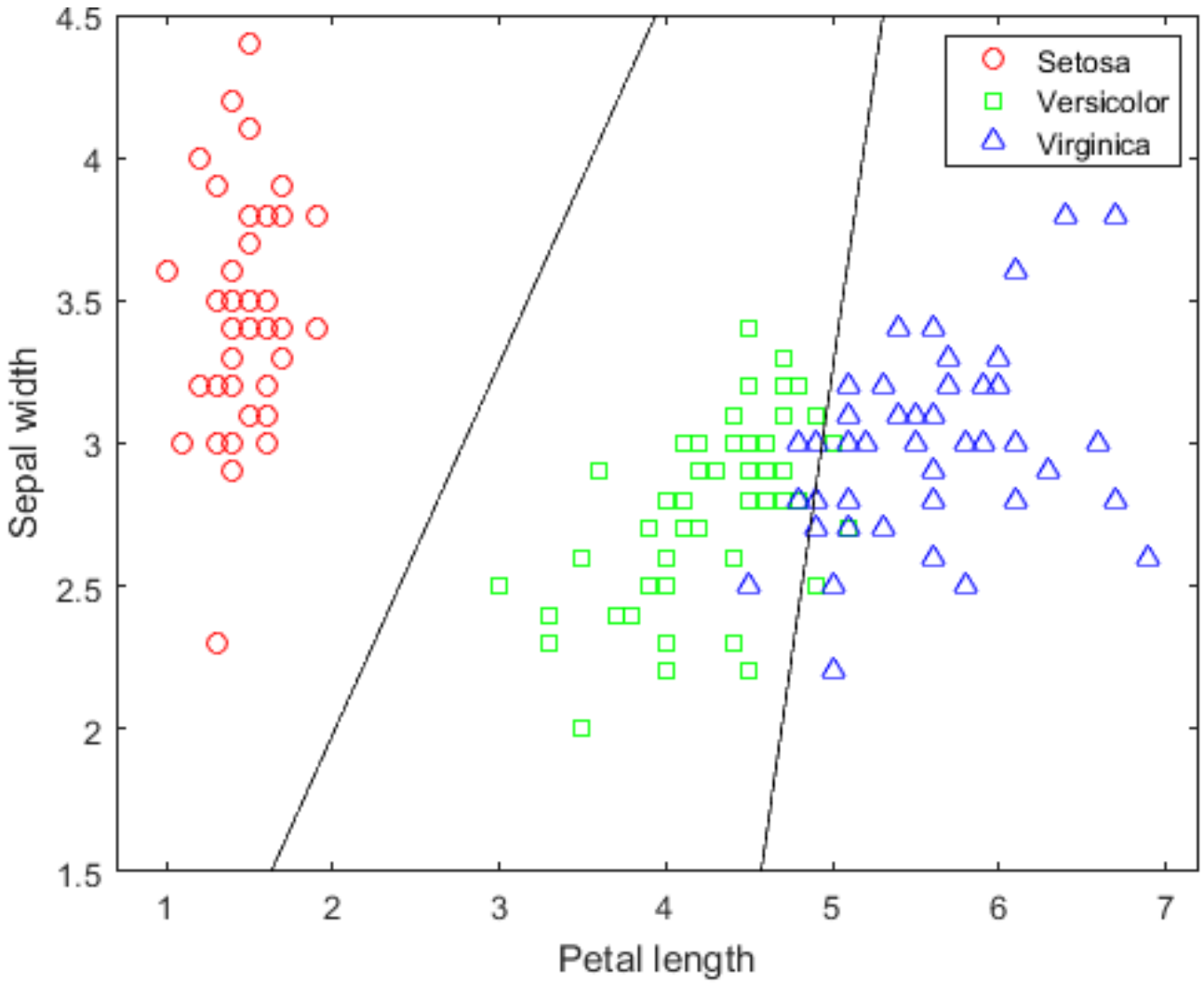
LDA - multiple classes



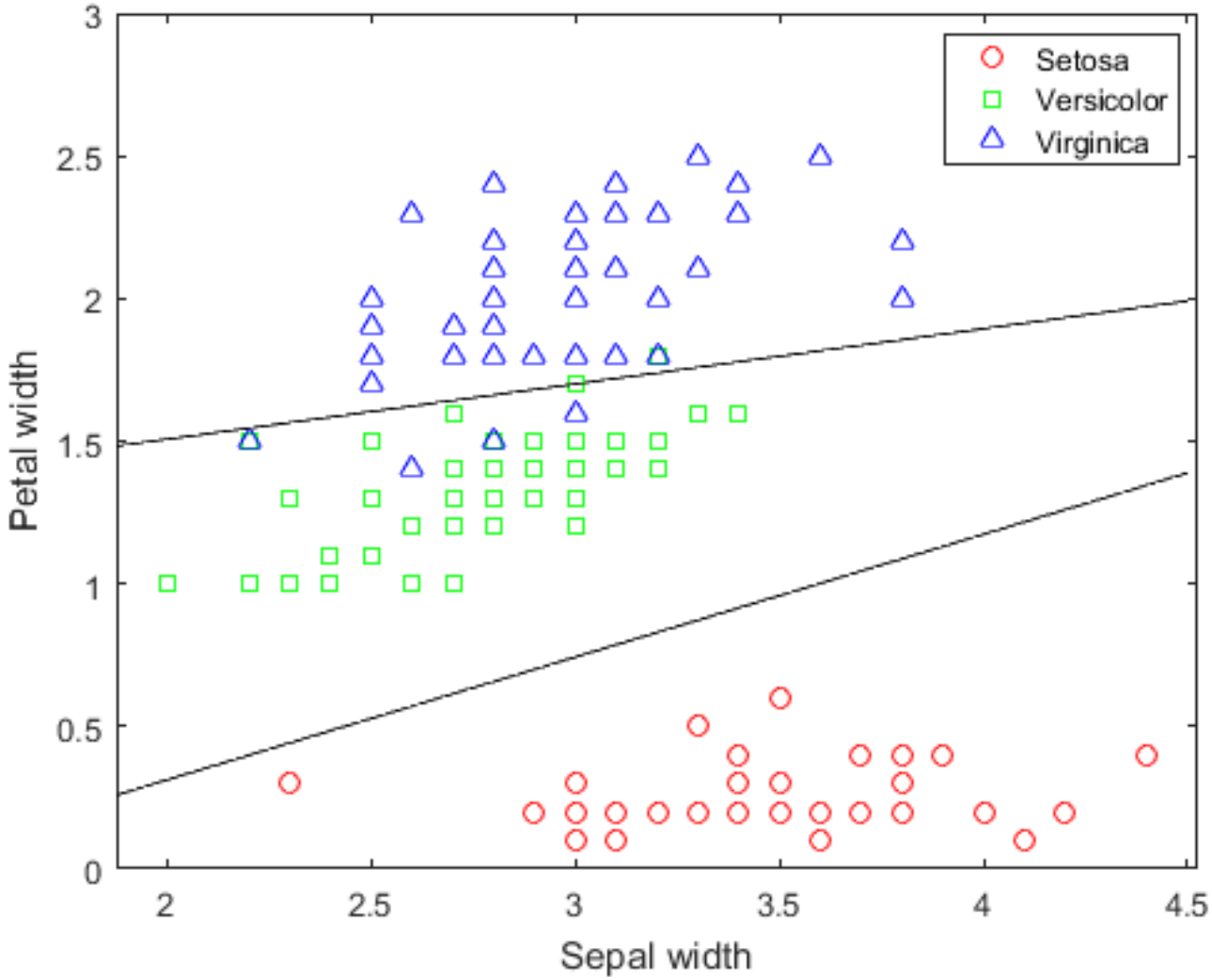
LDA - multiple classes



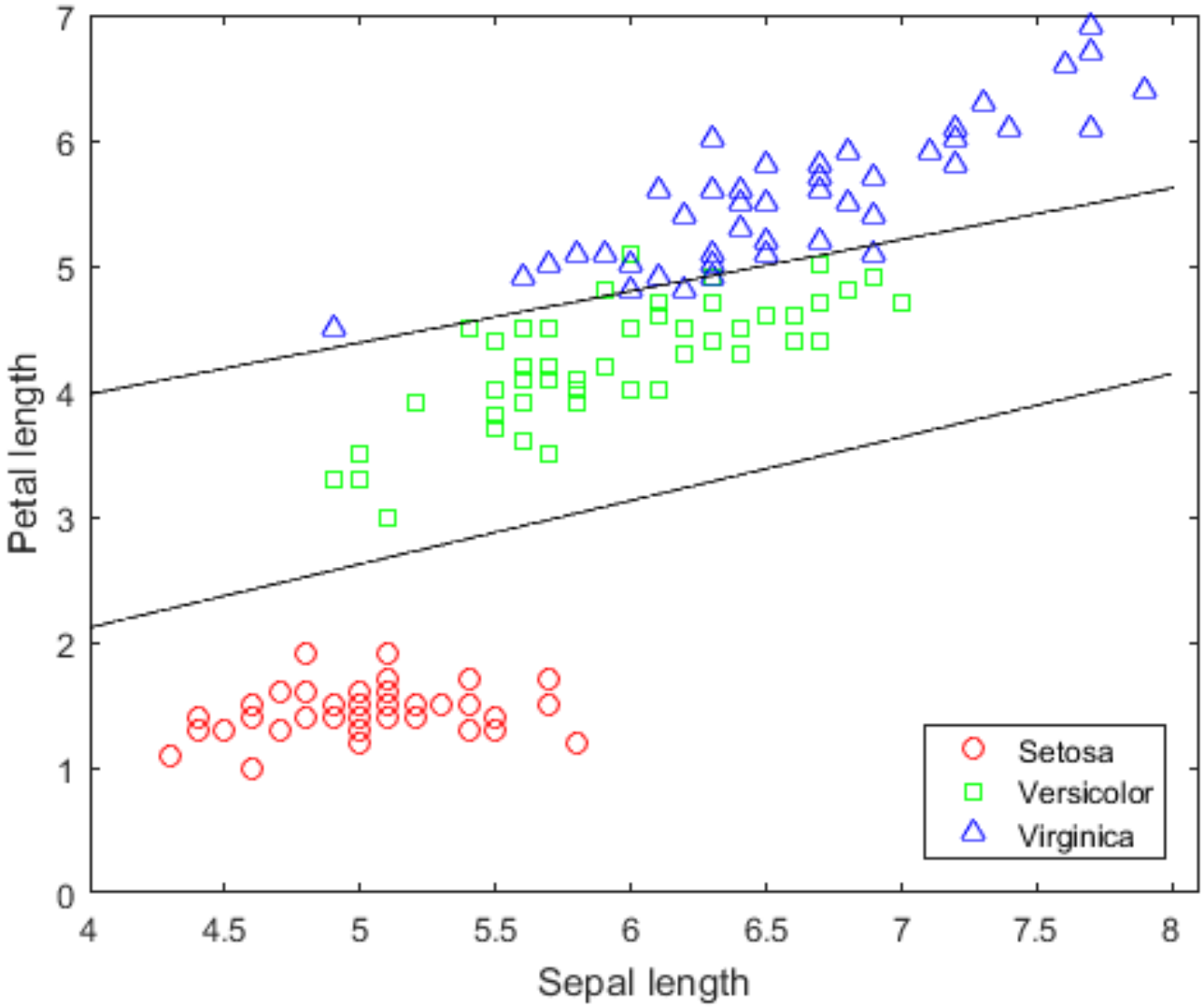
LDA - multiple classes



LDA - multiple classes



LDA - multiple classes



Feature Selection

- Some features help more in classification than others.
- Only spend resources on the important features.
- What are the important features?
- We will revisit the topic when covering feature selection.

Multiple classes - other strategies

One vs Rest (OvR) (One vs All (OvA))

- K binary classifiers giving a confidence score for a sample belonging to the k-th class.
- A sample is labeled with the class with the highest confidence score.
- Disadvantages:
  - The confidence scores have to be calibrated between each other.
  - The binary classifiers see unbalanced distributions, since the number of samples in the k-th class is much smaller than the number of samples NOT in the k-th class.

Multiple classes - other strategies

One vs One (OvO)

- $K(K-1)/2$ classifiers for each pair of classes.
- A sample is labeled with the class for which the most classifiers vote.
- Disadvantage: What if two classes receive the same number of votes?

Online learning

So far we considered learning tasks where the training data is available upfront. This is known as *batch learning* or *offline learning*.

In contrast, in *online learning* the training data becomes available sequentially.

Perceptron

Recall that we seek a direction vector $\mathbf{w}$ and separation threshold b such that for sample $\mathbf{v}$

- $\mathbf{w}^T\mathbf{v} < b \quad \Rightarrow \quad \mathbf{v} \in C_0$,
- $\mathbf{w}^T\mathbf{v} > b \quad \Rightarrow \quad \mathbf{v} \in C_1$.

$\mathbf{w}$ is also known as the vector of *weights*, while b is known as the *bias*.

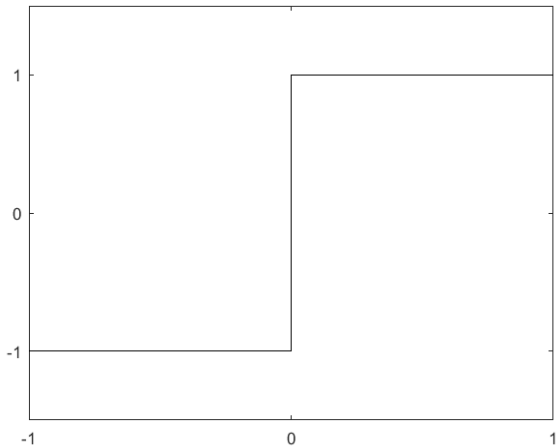
By extending the vectors $\mathbf{w}$ to $\hat{\mathbf{w}} = (-b, \mathbf{w})$ and $\mathbf{v}$ to $\hat{\mathbf{v}} = (1, \mathbf{v})$ this can be rephrased to

- $\hat{\mathbf{w}}^T\hat{\mathbf{v}} = \mathbf{w}^T\mathbf{v} - b < 0 \quad \Rightarrow \quad \mathbf{v} \in C_0$
- $\hat{\mathbf{w}}^T\hat{\mathbf{v}} = \mathbf{w}^T\mathbf{v} - b > 0 \quad \Rightarrow \quad \mathbf{v} \in C_1$

Perceptron

Define the step function

$$\text{sgn}(x) = \begin{cases} -1 & \text{if} \quad x < 0 \\ 0 & \text{if} \quad x = 0 \\ 1 & \text{if} \quad x > 0 \end{cases}$$



Then for sample $\mathbf{v}$

- $\text{sgn}(\hat{\mathbf{w}}^T \hat{\mathbf{v}}) = -1 \quad \Rightarrow \quad \mathbf{v} \in C_0$
- $\text{sgn}(\hat{\mathbf{w}}^T \hat{\mathbf{v}}) = 1 \quad \Rightarrow \quad \mathbf{v} \in C_1$

Perceptron

- The Perceptron initializes $\hat{\mathbf{w}}_0 = 0$ and updates the vector $\hat{\mathbf{w}}_{i-1}$ with each new training sample $\mathbf{v}_i$ by

$$\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \frac{\alpha}{2}\left(c_i - \text{sgn}(\hat{\mathbf{w}}_{i-1}^T \hat{\mathbf{v}}_i)\right)\hat{\mathbf{v}}_i,$$

  where c_i is the class label of $\mathbf{v}_i$, that is $c_i = -1$ if $\mathbf{v}_i \in C_0$ and $c_i = 1$ if $\mathbf{v}_i \in C_1$.
- $0 < \alpha \leq 1$ is the learning rate. If the learning rate is chosen too big any changes are too radical and oscillations occur (In the following examples α was set to 1).

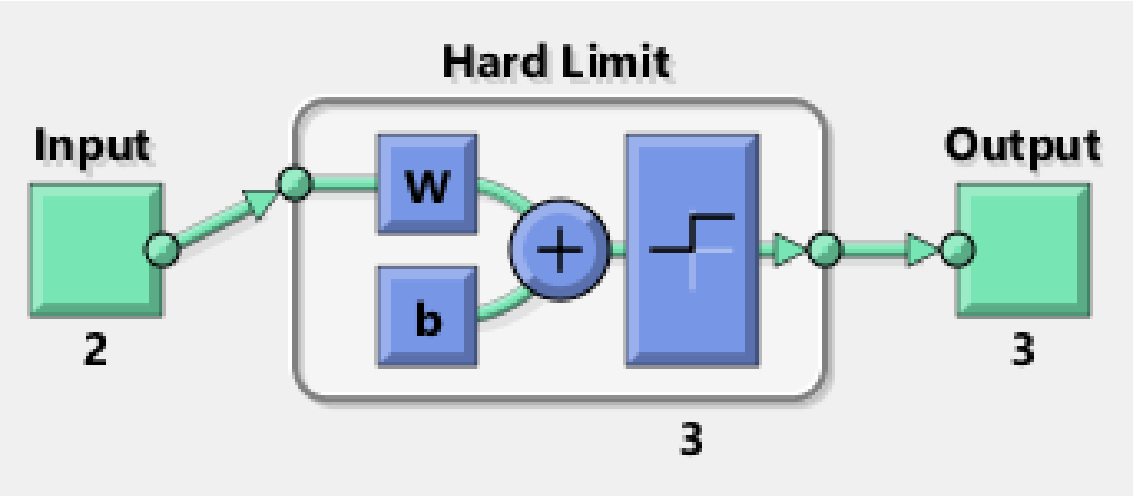
Perceptron

$$\hat{\mathbf{w}}_i = \hat{\mathbf{w}}_{i-1} + \frac{\alpha}{2}\left(c_i - \text{sgn}(\hat{\mathbf{w}}_{i-1}^T \hat{\mathbf{v}}_i)\right)\hat{\mathbf{v}}_i,$$

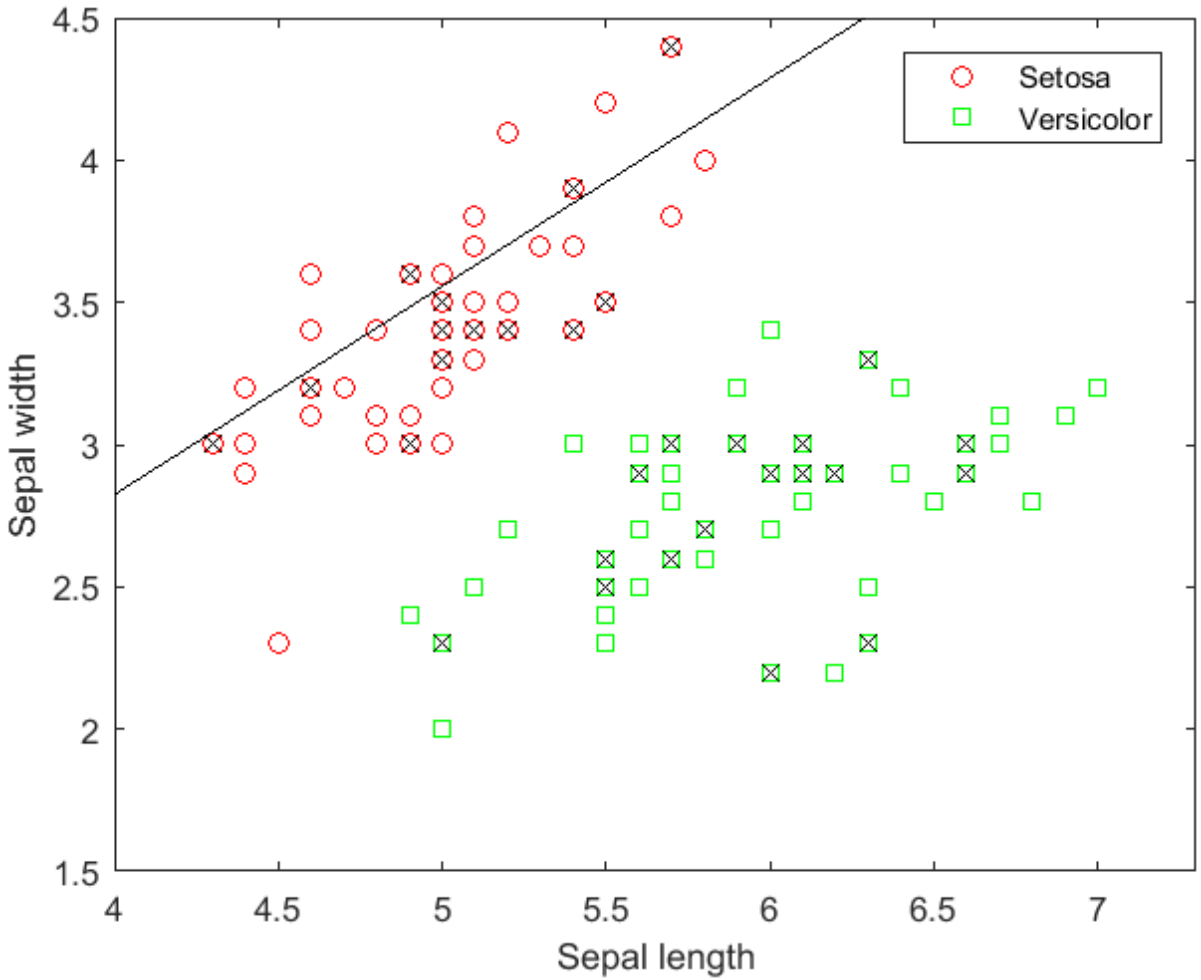
- Note that $\hat{\mathbf{w}}_i$ is the same as $\hat{\mathbf{w}}_{i-1}$ if the class label was determined correctly.
- If the class label was determined incorrectly the separation line is pulled in the direction such that it is more likely that the sample is labeled correctly in future.
- The sign of $c_i - \text{sgn}(\hat{\mathbf{w}}_{i-1}^T \hat{\mathbf{v}}_i)$ determines in which direction the separation line is pulled.
- Note that $\hat{\mathbf{w}}_i$ is a linear combination of all samples which caused a change to the separation line.

Perceptron

The Perceptron is a single layer neural network and implemented within the neural network framework.

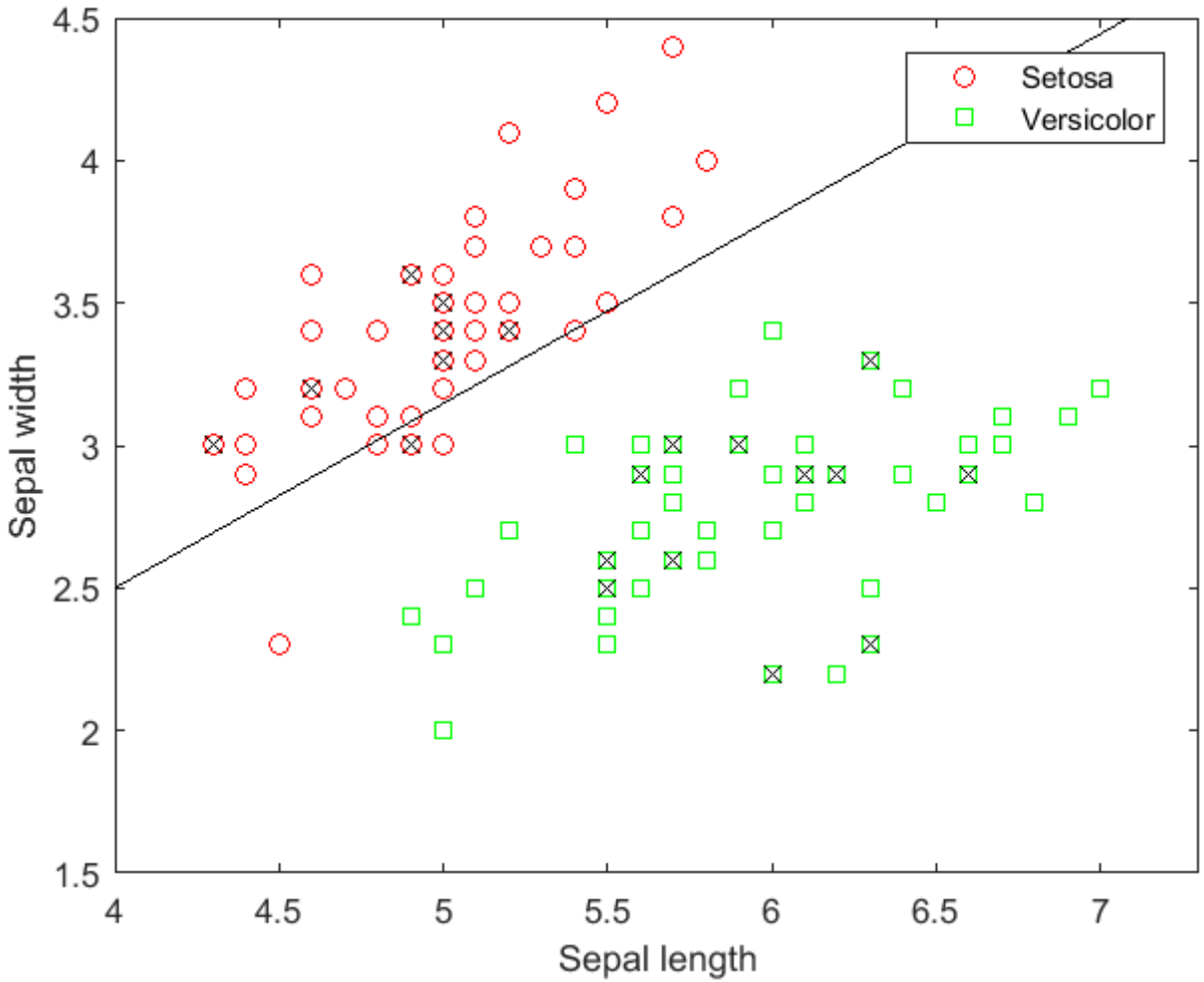


Perceptron



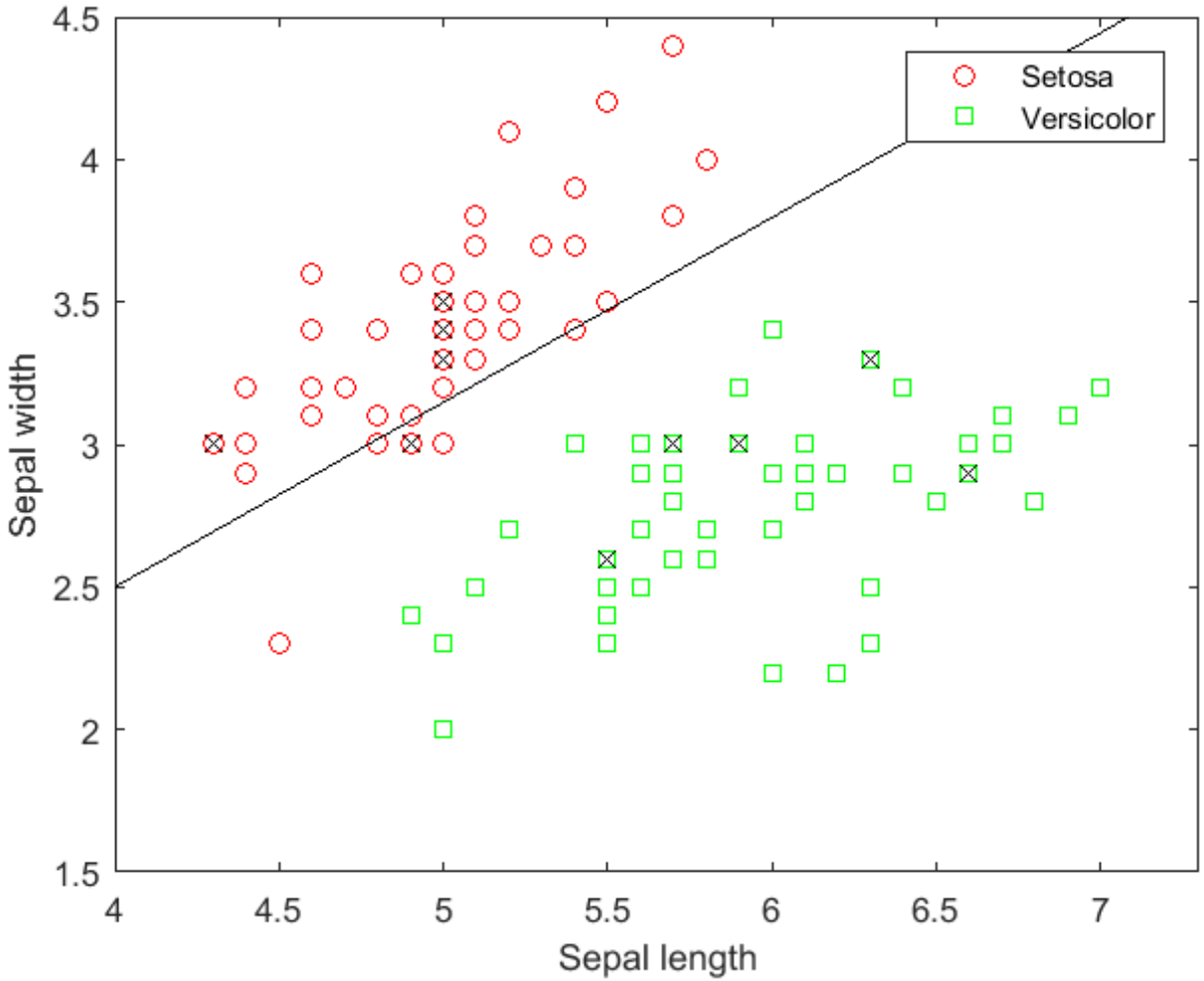
70% of data seen.

Perceptron



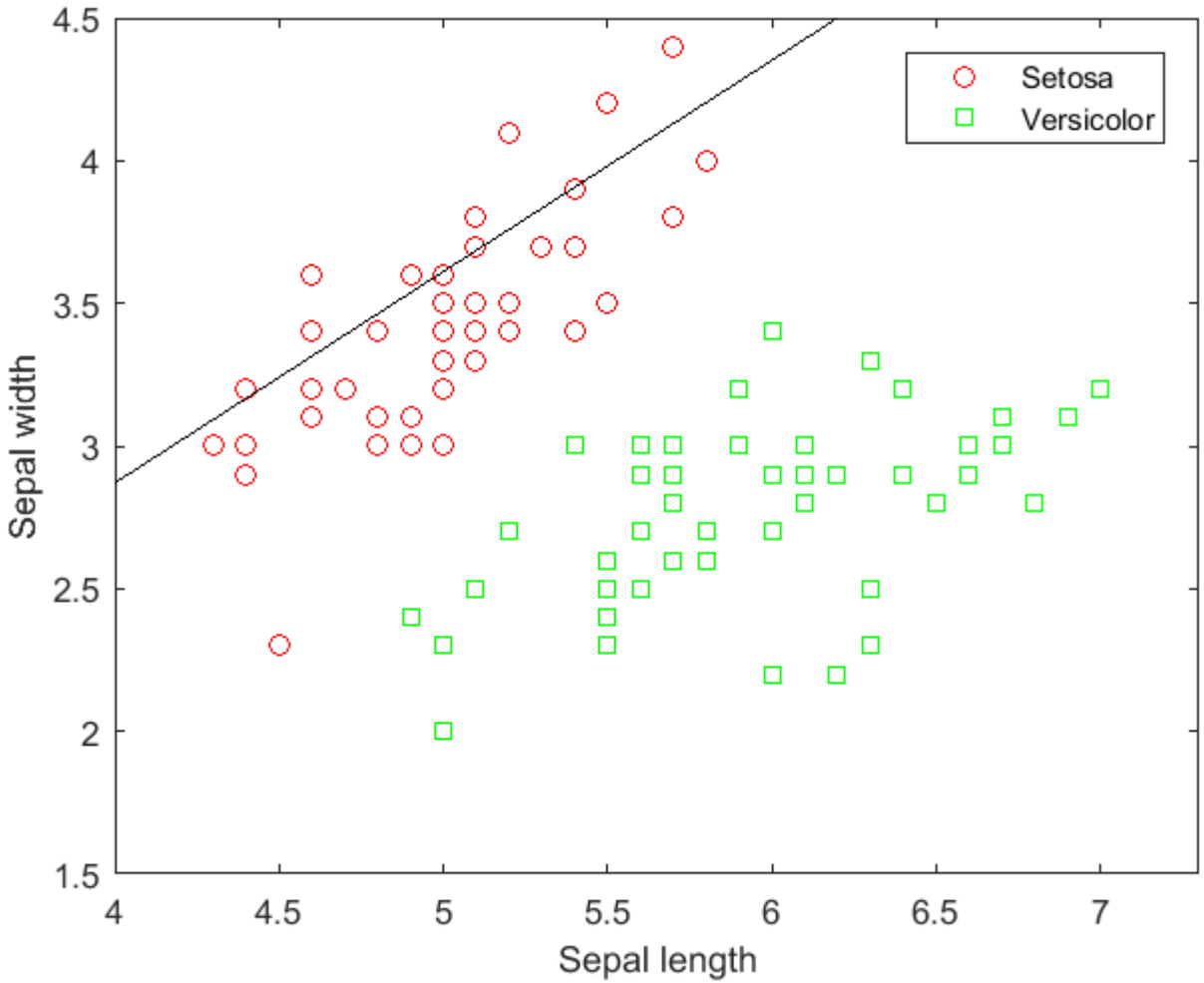
80% of data seen.

Perceptron



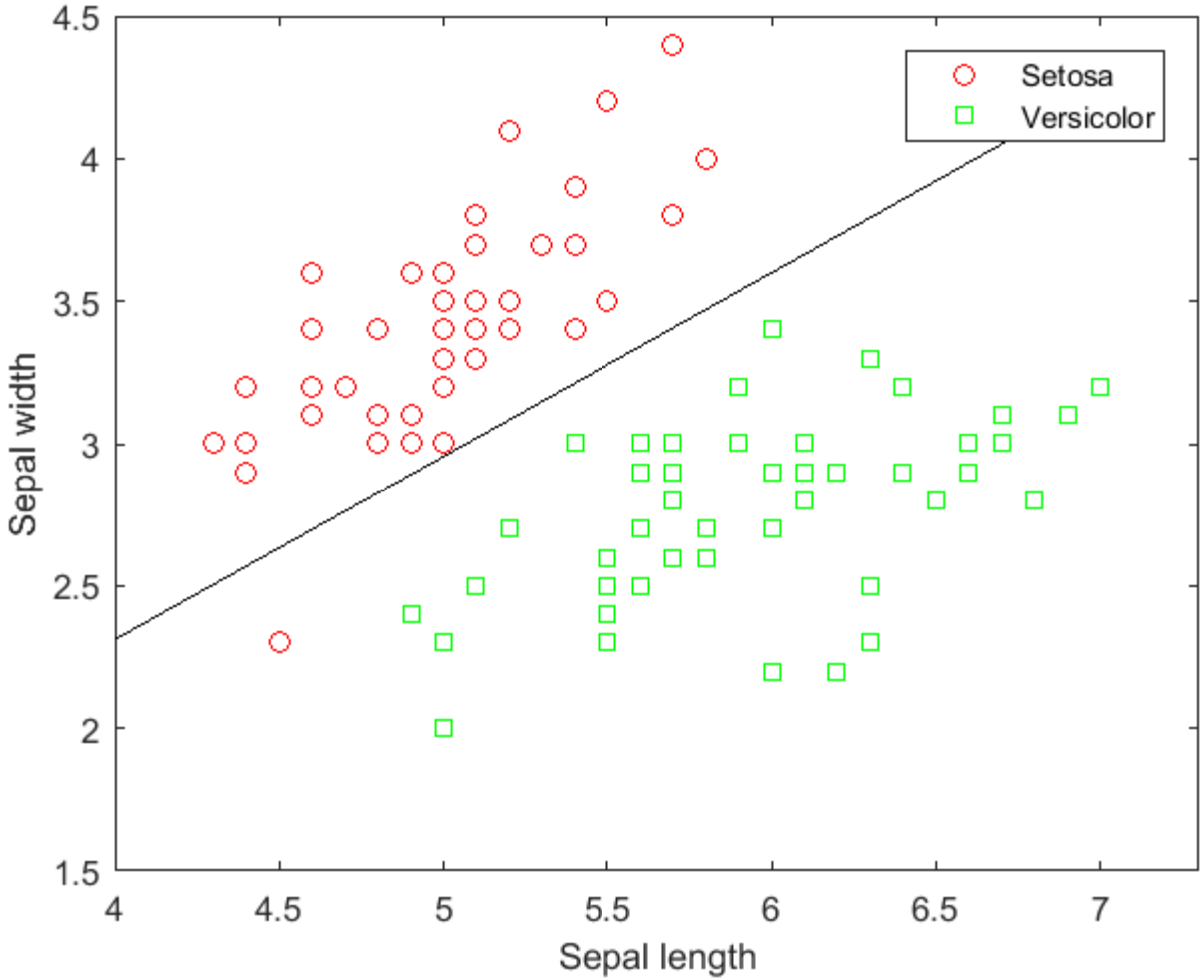
90% of data seen.

Perceptron



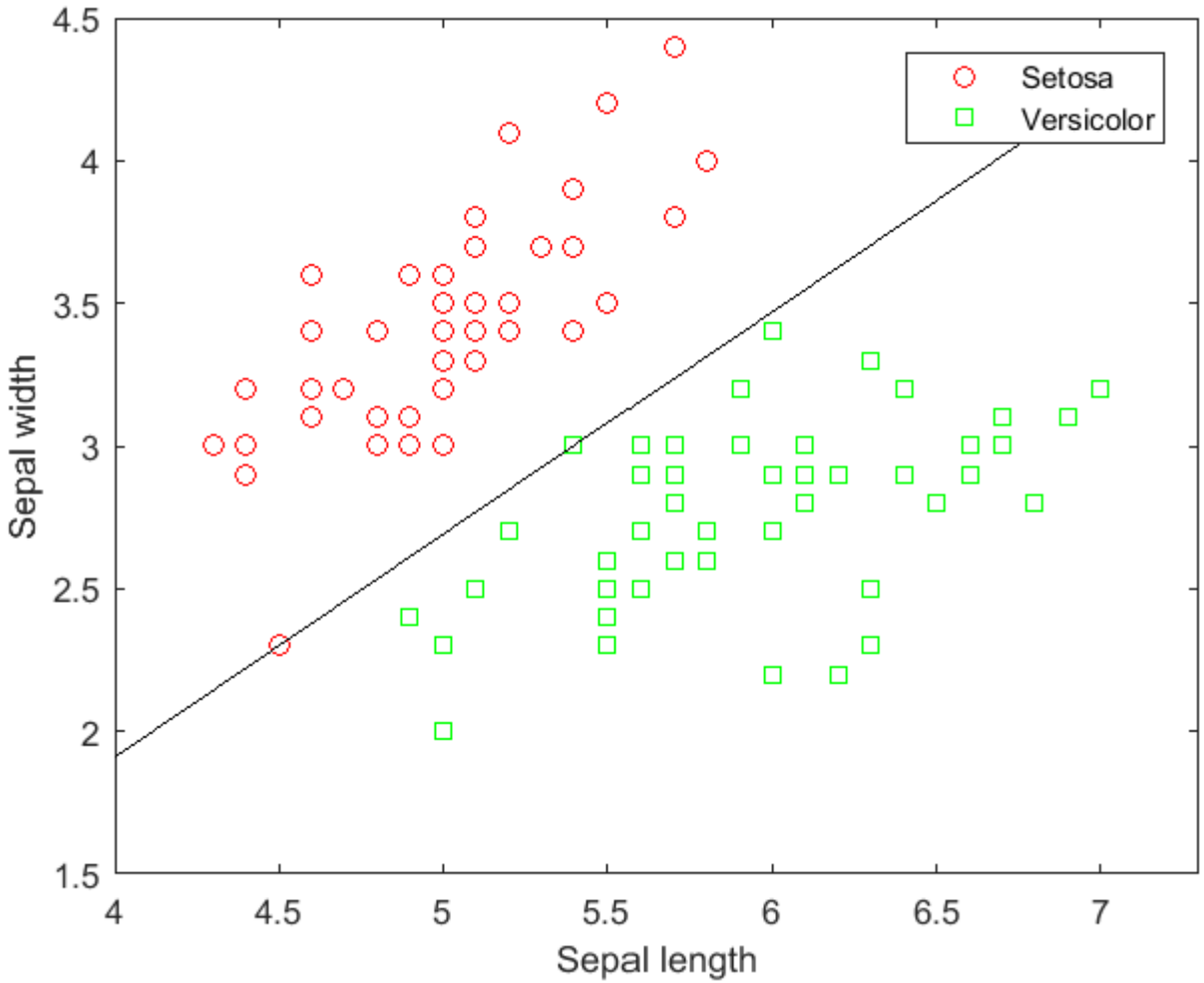
Complete data seen.

Perceptron



Complete data seen 100 times.

Perceptron

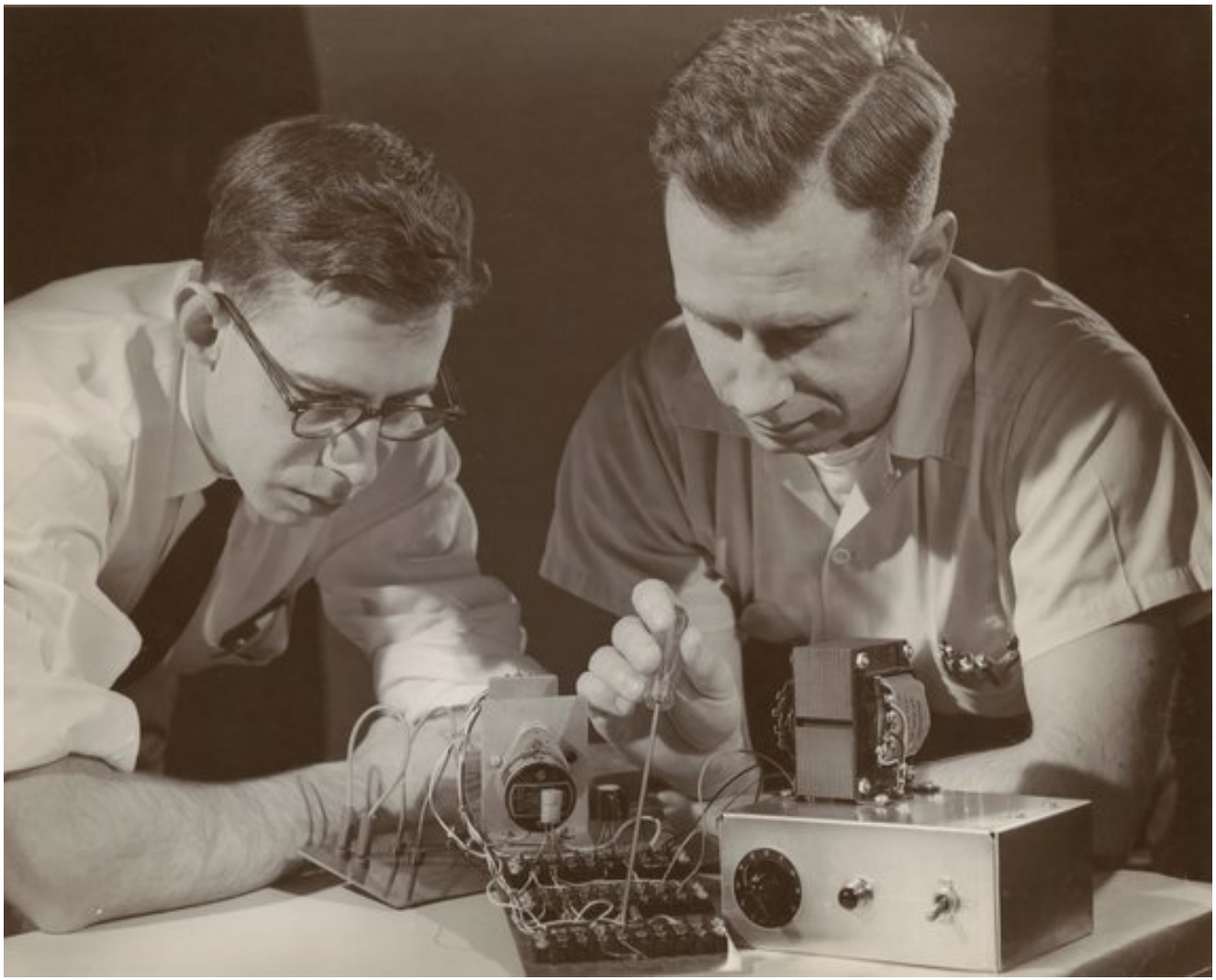


Convergence.

Perceptron

- invented 1957 by Rosenblatt,
- built in 1958 as a physical machine for image recognition,
- 400 photo cells,
- weights as potentiometers which were updated by electric motors.

Perceptron



Support Vector Machine (SVM)

- Letting the learning rate α be a choice, is crude.
- All should be determined by the data.
- The *support vector machine* tries to maximize the margin between the two classes.
- When all samples are correctly classified, that is $c_n = \text{sgn}(\hat{\mathbf{w}}^T \hat{\mathbf{v}}_n)$, the *margin* is defined as

$$\min_{n=1,\ldots,N} c_n \frac{\hat{\mathbf{w}}^T \hat{\mathbf{v}}_n}{\|\hat{\mathbf{w}}\|} > 0.$$

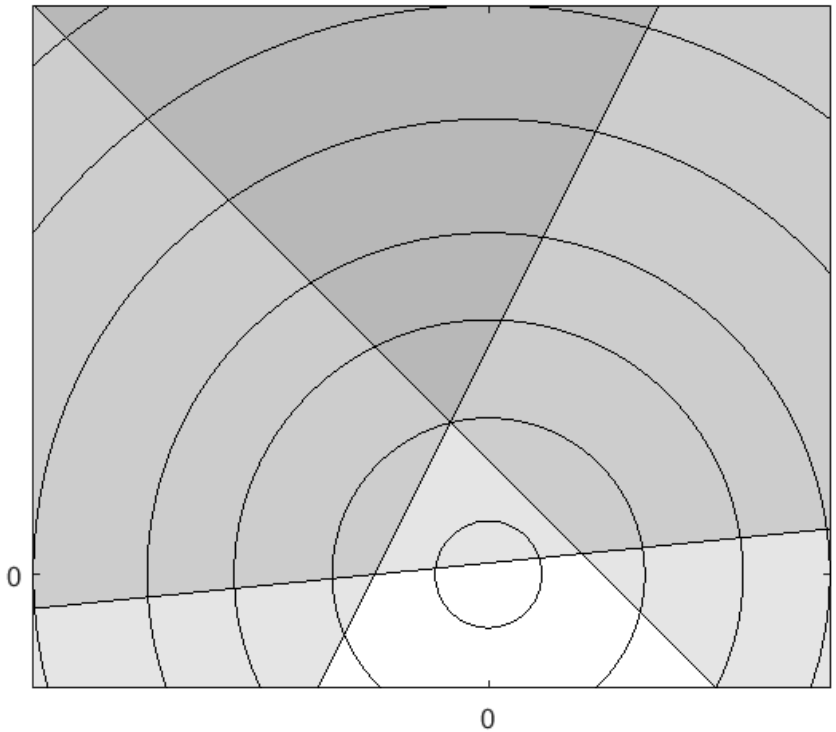
- Note that the margin is independent of any rescaling of $\hat{\mathbf{w}}$.

SVM

- We use the freedom to rescale $\hat{\mathbf{w}}$ such that $c_n \hat{\mathbf{w}}^T \hat{\mathbf{v}}_n \geq 1$ for all $n = 1, \ldots N$ with equality for at least one sample.
- In the case of samples, where we have equality, the constraints are said to be *active*. For the other samples, they are *inactive*.
- Subject to these constraints, maximizing the margin is equivalent to maximizing $\|\hat{\mathbf{w}}\|^{-1}$.
- Equivalent problem: Minimize the *objective function* $\|\hat{\mathbf{w}}\|^2/2$ subject to the constraint $c_n \hat{\mathbf{w}}^T \hat{\mathbf{v}}_n \geq 1$ for all $n = 1, \ldots N$.
- This is a *quadratic programming problem*.

Constraints

- isolines of $\|\hat{\mathbf{w}}\|^2/2$,
- white: no constraints satisfied,
- light grey: one constraint satisfied,
- medium grey: two constraints satisfied,
- dark grey: all constraints satisfied, known as the *feasible set*.

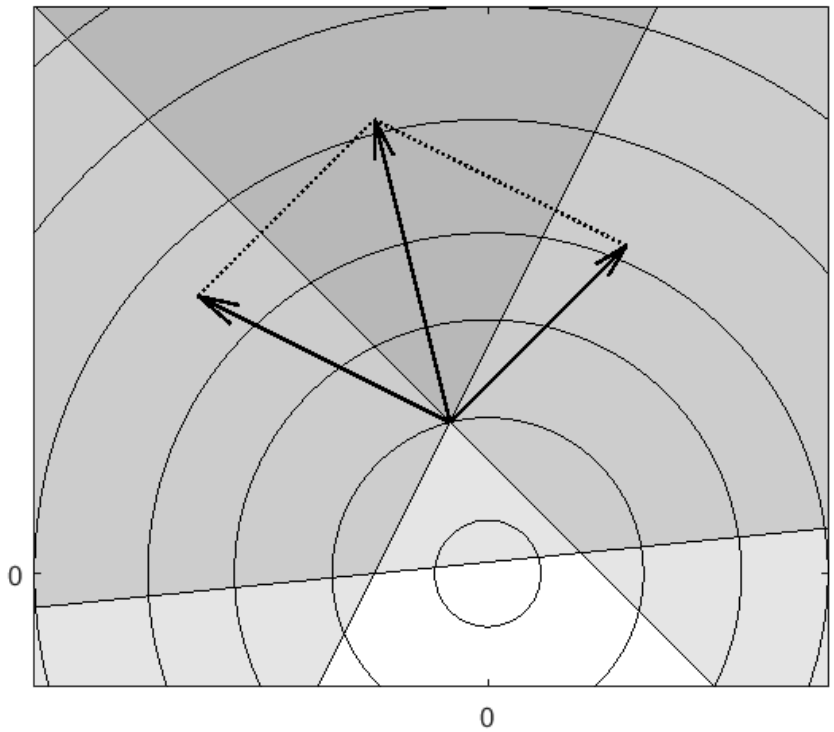


Gradients

At the minimum subject to the constraints the gradient of $\|\hat{\mathbf{w}}\|^2/2$ is a linear combination of the gradients of the constraints.

$$\nabla(\|\hat{\mathbf{w}}\|^2/2) = \sum_{n=1}^{N} \alpha_n \nabla \left(c_n \hat{\mathbf{w}}^T \hat{\mathbf{v}}_n\right),$$

$$\hat{\mathbf{w}} = \sum_{n=1}^{N} \alpha_n c_n \hat{\mathbf{v}}_n.$$



Lagrangian multipliers

- The *Lagrangian function* is given by

$$L(\hat{\mathbf{w}}, \boldsymbol{\alpha}) = \frac{1}{2}\|\hat{\mathbf{w}}\|^2 - \sum_{n=1}^{N} \alpha_n \left[c_n \hat{\mathbf{w}}^T \hat{\mathbf{v}}_n - 1\right],$$

  where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)^T$ are *Lagrange multipliers*.
- $\alpha_n = 0$ if the constraint is inactive, positive otherwise. This property is known as *complementarity*.
- This strategy combines the objective function and the constraints into one function.
- The solution to the constrained optimization is a stationary point of $L(\hat{\mathbf{w}}, \boldsymbol{\alpha})$.

SVM

- Inserting $\hat{\mathbf{w}} = \sum_{n=1}^{N} \alpha_n c_n \hat{\mathbf{v}}_n$, into the Lagrangian function, $\boldsymbol{\alpha}$ maximizes the function

$$L(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{i=1}^{N} \sum_{n=1}^{N} \alpha_i \alpha_n c_i c_n \hat{\mathbf{v}}_i^T \hat{\mathbf{v}}_n.$$

  subject to the constraints $\alpha_n \geq 0$, $n = 1, \ldots, N$.
- This is the *dual representation* of the maximum margin problem.

SVM

- Recall that $\hat{\mathbf{w}} = (-b, \mathbf{w})$ and $\mathbf{v}$ to $\hat{\mathbf{v}} = (1, \mathbf{v})$.
- The bias b is given by $-\sum_{n=1}^{N} \alpha_n c_n$.
- The dual maximization problem becomes

$$L(\boldsymbol{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{i=1}^{N} \sum_{n=1}^{N} \alpha_i \alpha_n c_i c_n \left(1 + \mathbf{v}_i^T \mathbf{v}_n\right).$$

- The solution depends solely on the inner product between samples.

SVM

- The classification is according to the sign of

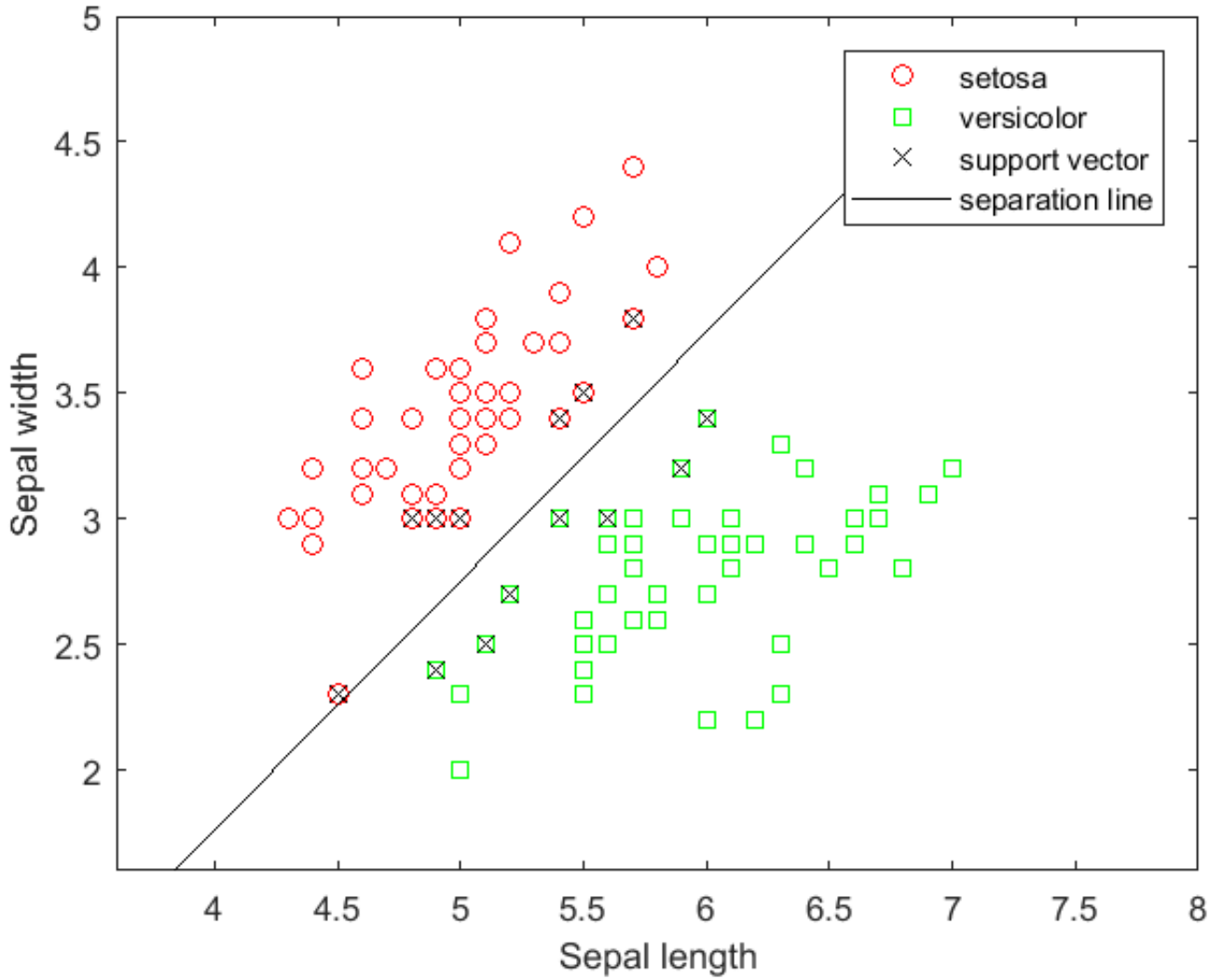
$$\hat{\mathbf{w}}^T\hat{\mathbf{v}} = \sum_{n=1}^{N} \alpha_n c_n \hat{\mathbf{v}}_n^T \hat{\mathbf{v}} = \sum_{n=1}^{N} \alpha_n c_n \left(1 + \mathbf{v}_n^T \mathbf{v}\right) = -b + \sum_{n=1}^{N} \alpha_n c_n \mathbf{v}_n^T \mathbf{v}.$$

- We will see the advantages of recasting the problem when considering non-linear classification and the kernel trick.

SVM

- The samples for which $\alpha_i \neq 0$ in the linear combination to form the weight vector are called the *Support Vectors*.
- Invented in 1964 by Vapnik and Lerner as the Generalized Portrait Method.
- The term generalized portrait refers to the centre of a sphere which contains patterns belonging to a certain class, but no other class.

SVM


setosa
versicolor
support vector
separation line
Sepal width
Sepal length
5
4.5
4
3.5
3
2.5
2
4
4.5
5
5.5
6
6.5
7
7.5
8