UNIVERSITY OF BERGEN

INFO 284 - Machine Learning

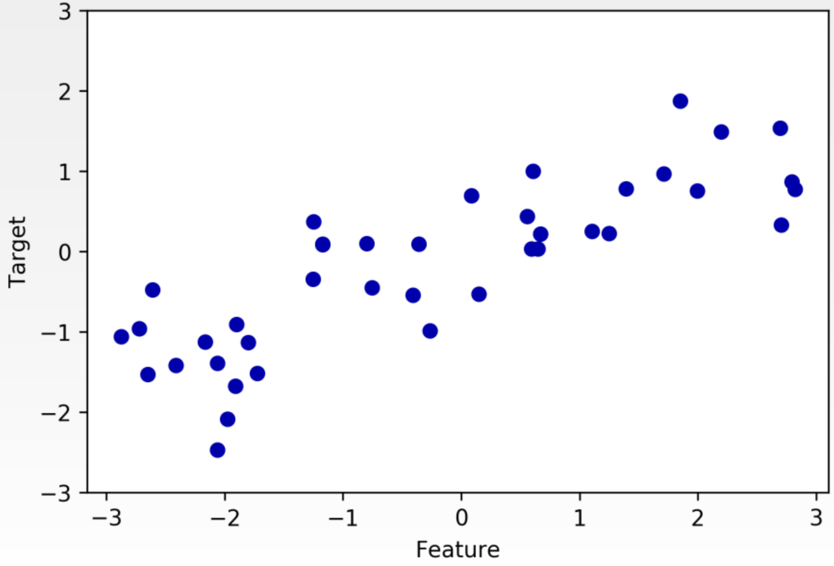
Linear Classifiers

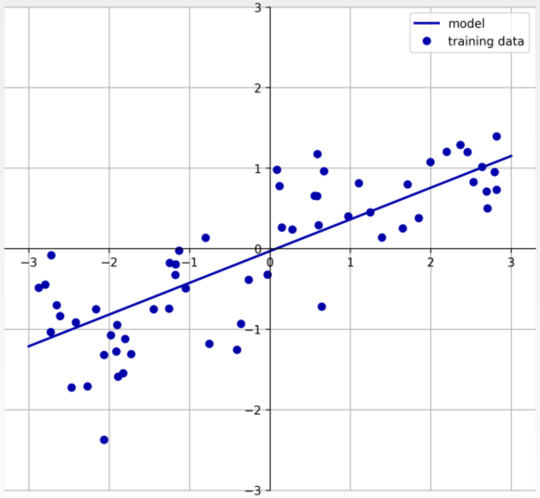
Bjørnar Tessem

UNIVERSITY OF BERGEN

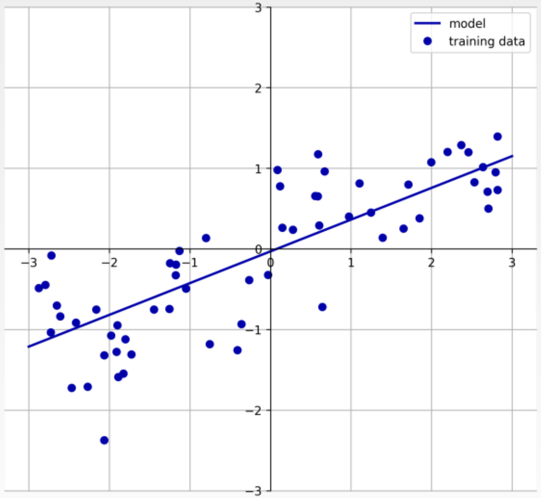
Univariate linear regression

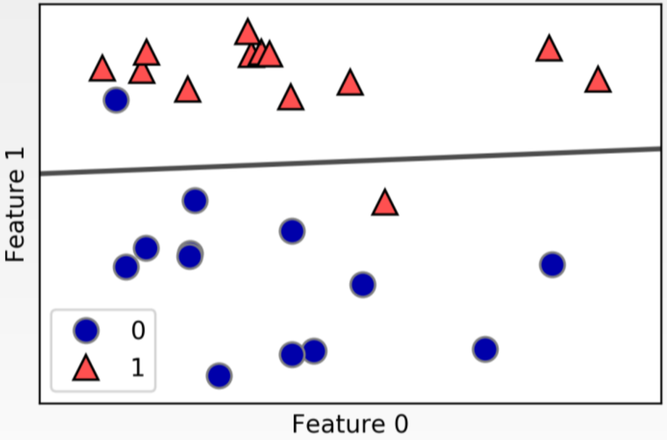
$$\hat{y} = \boxed{w[0]} * x[0] + \boxed{b}$$





Linear Binary Classifiers





Linear Binary Classifiers

- The hypothesis is a **linear function** of the input features.

$$\hat{y} = \boxed{w[0] * x[0] + w[1] * x[1] + \cdots + w[p] * x[p] + b > 0}$$

- If the expression is <= 0, then one category, else the other
- The decision boundary is a linear function of the input
- A linear binary classifier separates two classes using a line, or a plane or a hyper-plane

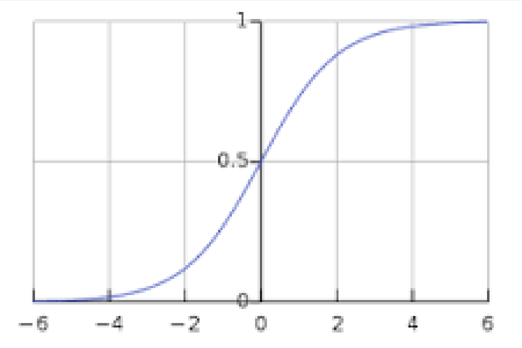
Linear Binary Classification Algorithms

The algorithms differ on:

- the way in which they measure how well a particular combination of w and b fits the training data
- if and what kind of regularisation they use
- Here: Logistic Regression and Linear Support Vector Machines

Logistic Regression

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$



- Named for the function used in the core of the method - logistic function (also called sigmoid).
- Logistic regression models the probability of the first class
- The first class is often called the default

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \cdots + w[p] * x[p] + b$$

Assumptions

- Dichotomous targets (e.g., presence vs. absent).

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

- No outliers in the data
  - assessed by converting the continuous features to standardised scores, and removing values below -3.29 or greater than 3.29.
- No high correlations (multicollinearity) among the features.
  - compute correlation matrix among all features.
  - correlation coefficients in [-0.9, 0.9] → good enough

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Logistic Regression

$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

- Assume there is class A and class B. Default is A.
- Logistic regression is a linear method, but the predictions are transformed to a probability using the logistic function

$$p(\hat{y} \text{ is } A) = \frac{e^{w[0]*x[0]+b}}{e^{w[0]*x[0]+b}+1}$$

Logistic Regression

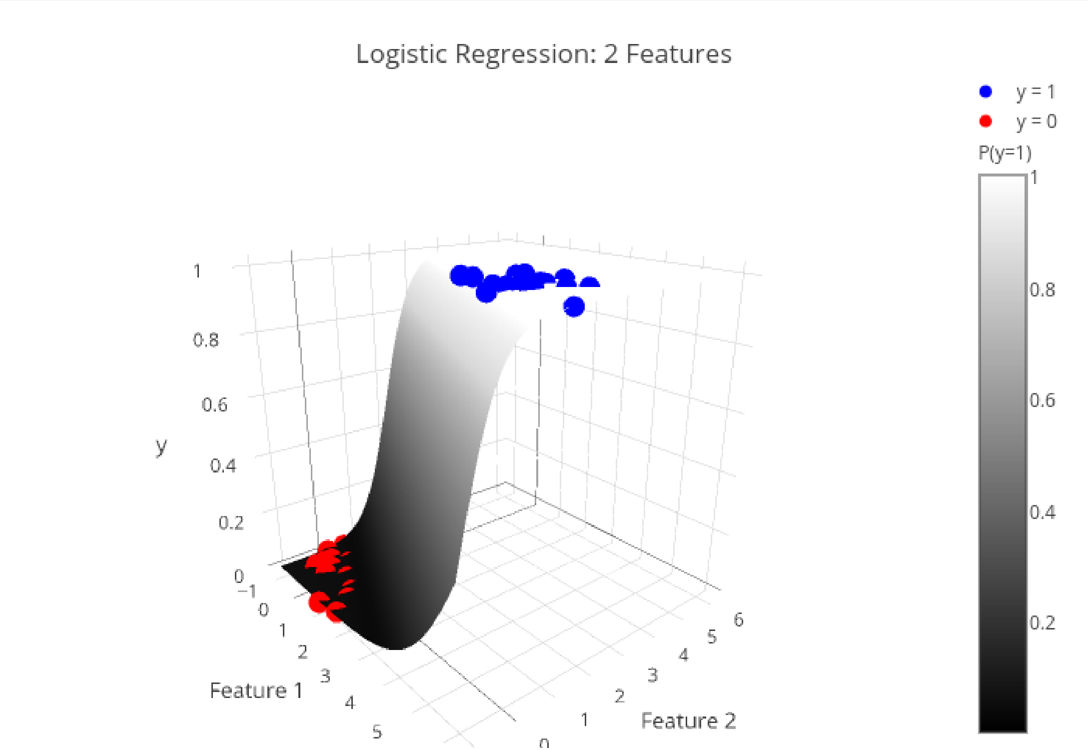
$$S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$

$$p(\hat{y}) = S(\sum_{i=1}^{n} w[i] * x[i] + b)$$

$$loss(\hat{y}) = |y_i - p(\hat{y})|$$

- Optimize to mimimize total Loss
  1. Compute loss of each data point in training set
  2. Sum up for whole set
  3. Adjust weights a little bit in the direction that reduce total loss
  4. Repeat from 1 until no changes
- Gradient descent optimization process

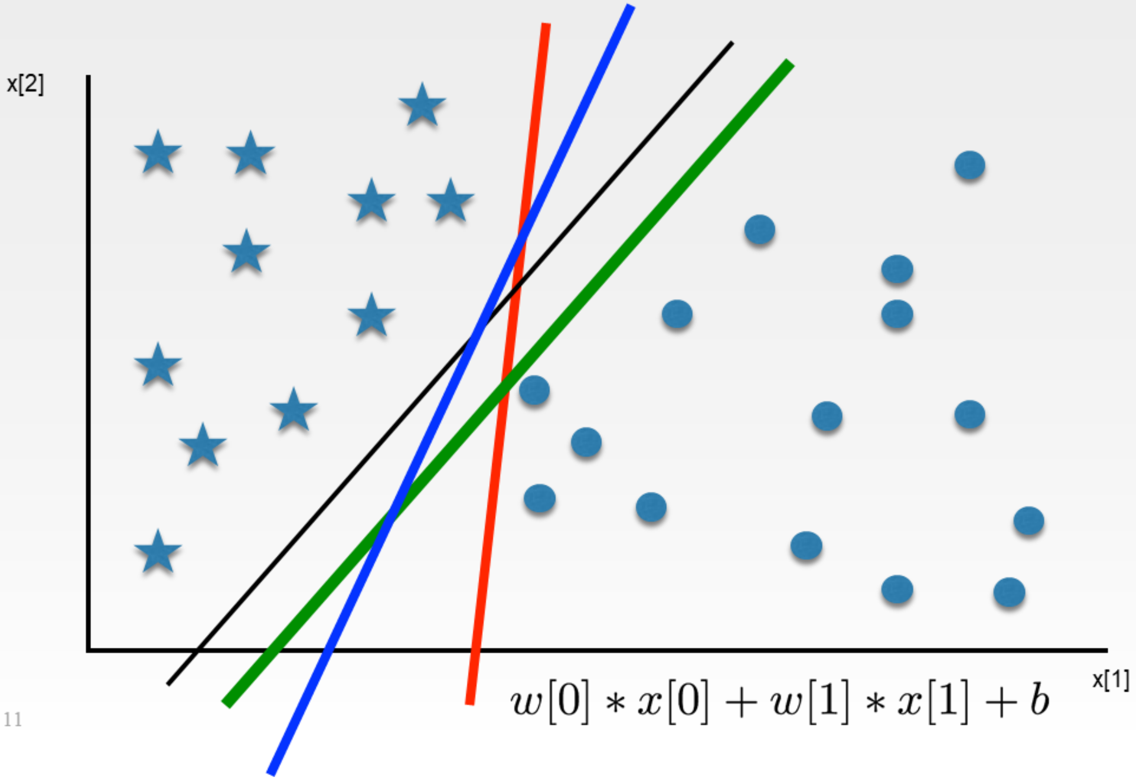
The logistic function



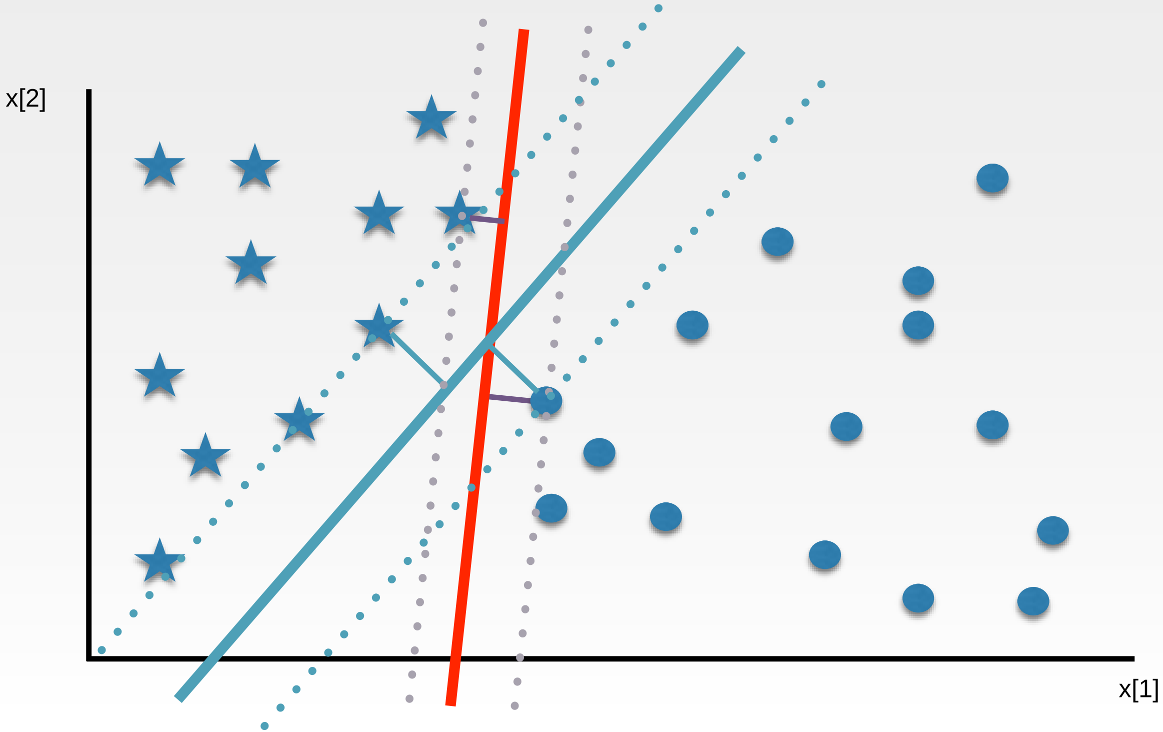
Properties of logistic regression

- L1 and L2 regularization is possible – similar to linear regression
    - Add to total Loss function: $\alpha * \|\boldsymbol{w}\|_2$
    - Choice of alpha <==> choice of C since alpha = 1/C
    - large alpha/ small C => simple models
- L1 when you assume that not all your features are actually relevant
- Logistic regression is fast to train and make fast predictions. They **scale** to very large data sets
- Relatively easy to understand how they work → use logistic function on linear equation
- They are not transparent
    - not clear why some coefficients are high, especially if the features are highly correlated

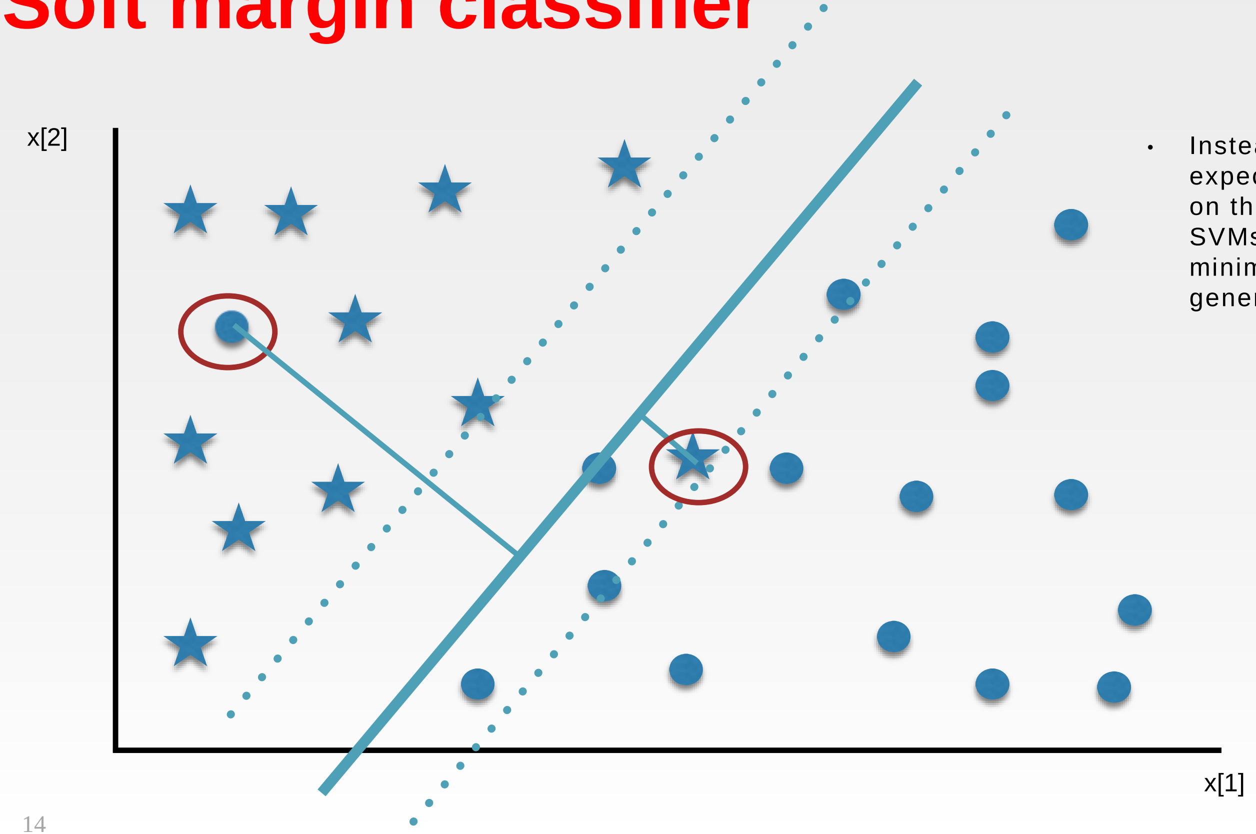
Linear Support Vector Machines (SVM)



Maximal margin classifier

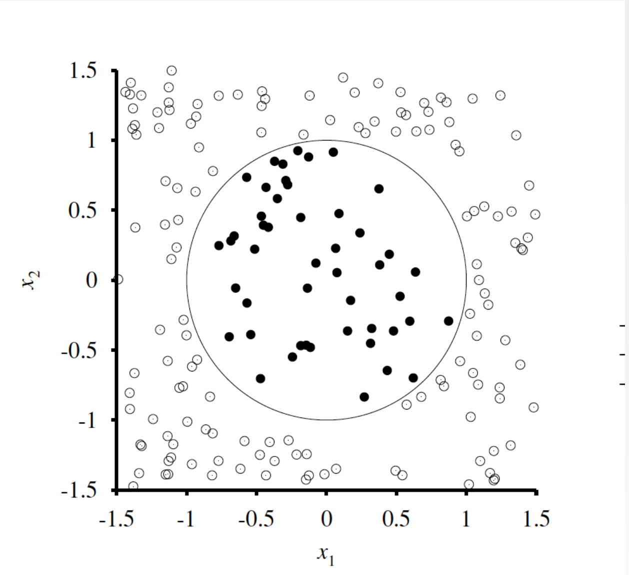


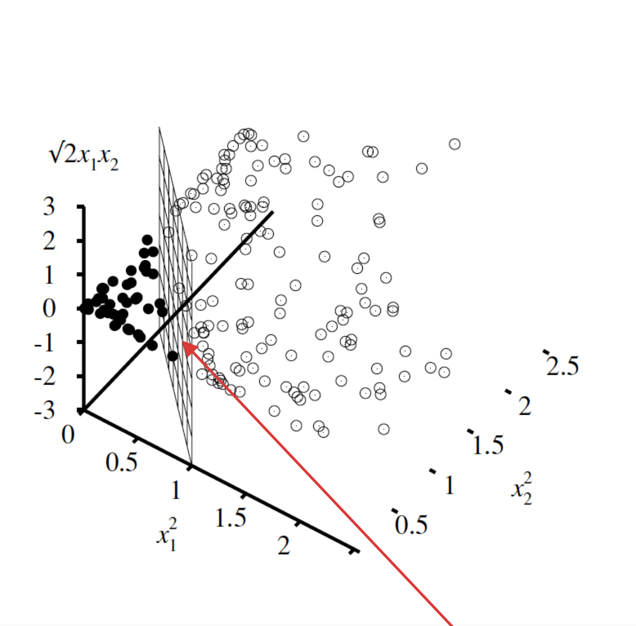
Soft margin classifier



- Instead of minimizing expected empirical loss on the training data, SVMs attempt to minimize expected generalization loss

Non-linear data



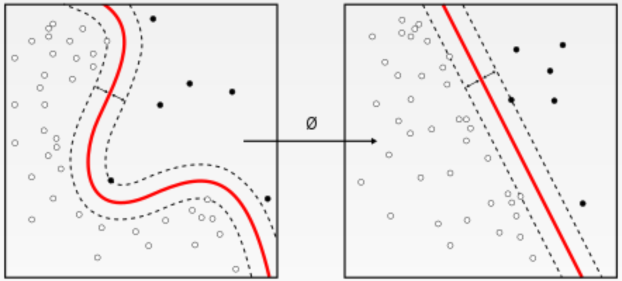


$$f_1 = {x_1}^2 \qquad f_2 = {x_2}^2 \qquad f_3 = \sqrt{2}x_1x_2$$

Separating plane

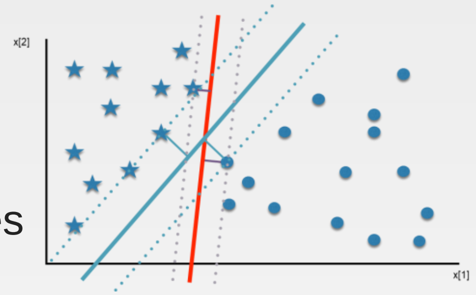
Making data linearly separable

- If data are mapped into a space of sufficiently high dimension, then they will almost always be linearly separable = if you look at a set of points from enough directions, you'll find a way to make them line up.



- Traditionally SVMs use the convention that class labels are +1 and -1

Linearly separable data



- The separator is defined as the set of points **x** such that $\mathbf{w} \cdot \mathbf{x} + b = 0$
- Find **w** and *b* such that the resulting linear plane maximizes the margin.
  - Minimize total «hinge loss»
- Converts problem to this equation (called the dual representation) instead:

$$\underset{\alpha}{\operatorname{argmax}} \boxed{\sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k (\mathbf{x}_j \cdot \mathbf{x}_k)}$$

$$\boxed{\alpha_j \geq 0} \text{ and } \boxed{\sum_j \alpha_j y_j = 0} \quad y_j, y_k \in \{1, -1\}$$

Dot product:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i * b_i$$

Classification

$$\underset{\alpha}{\operatorname{argmax}} \sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k (\mathbf{x}_j \cdot \mathbf{x}_k)$$

$$\alpha_j \geq 0 \text{ and } \sum_j \alpha_j y_j = 0 \quad y_j, y_k \in \{1, -1\}$$

- Once we have found alpha, we can calculate

$$\boxed{\hat{y} = \operatorname{sign}\Big(\sum_j \alpha_j y_j (\mathbf{x} \cdot \mathbf{x}_j) - b\Big)}$$

new
data
point

zero except for the
support vectors

Not linearly separable data

$$\underset{\alpha}{\operatorname{argmax}} \sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k (F(\mathbf{x}_j) \cdot F(\mathbf{x}_k))$$

- F(x) is the new feature space
- The dot product can be computed without computing F
- Kernel function: $K(\mathbf{x}_j, \mathbf{x}_k)$

$$\underset{\alpha}{\operatorname{argmax}} \sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k K(\mathbf{x}_j, \mathbf{x}_k)$$

Kernel function

Classification: $\hat{y} = \operatorname{sign}\left(\sum_j \alpha_j y_j K(\mathbf{x}_j, \mathbf{x}) \right)$

- The kernel is a similarity function

  The higher value the closer the vectors

- For linearly separable the kernel is the dot product itself

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i * b_i$$

Common kernel functions

- Polynomial kernel:

  computes all possible polynomials up to a certain degree of the original features

$$K(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

  - For d=2 (quadratic kernel), and two features we get

$$f_1 = {x_1}^2 \qquad f_2 = {x_2}^2 \qquad f_3 = \sqrt{2} x_1 x_2$$

- Mostly used in NLP problems where using d>2 tends to lead to overfitting

Radial Basis Function

- Radial Basis Function (RBF) kernel = Gaussian Kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2) \Longleftrightarrow K(\mathbf{x}, \mathbf{x}') = e^{(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)}$$

Euclidean distance

Parameter that controls the
width of the Gaussian Kernel

The gamma parameter determines the scale of what it means for points to be close

Kerneled SVC

- In Python the class is called SVC – Support Vector Classifier

- Gamma - a small gamma means a large radius for the Radial Basis Function kernel, which means many points are considered close together (how smooth the decision boundary is)

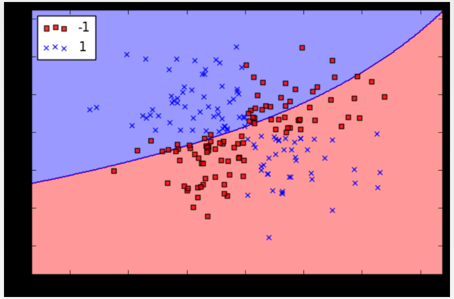
- Two parameters – gamma and C
  - Low gamma - less complex hypothesis.
  - High gamma - more complex hypothesis – creates islands

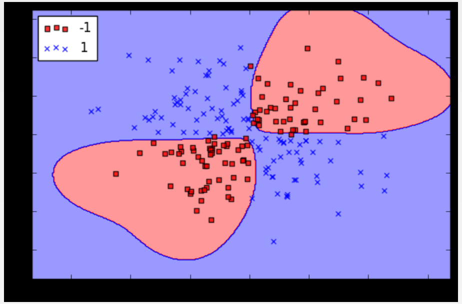
- C controls regularisation.

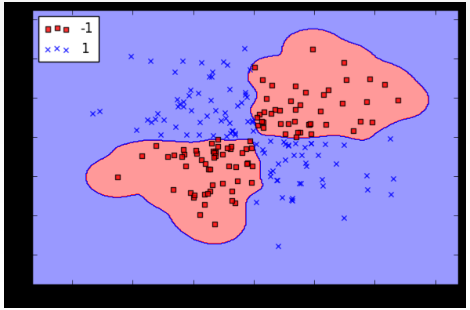
Small C: each misclassified data point have limited influence

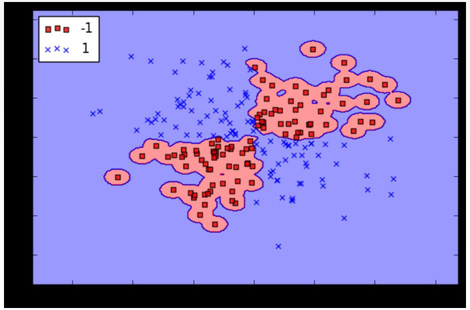
Large C: each misclassified data point contributes to allow «narrower margins» between support vectors.

Gamma variation – 0.01, 1, 10, 100

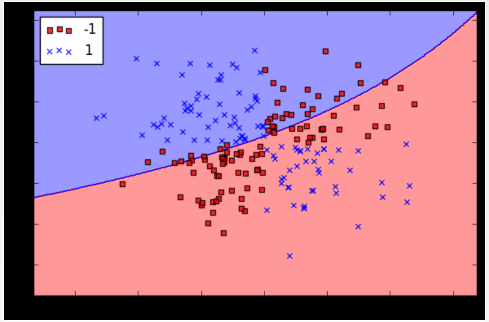


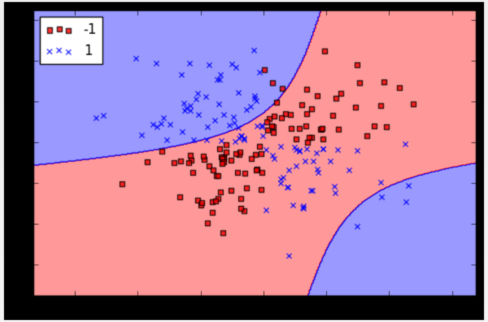


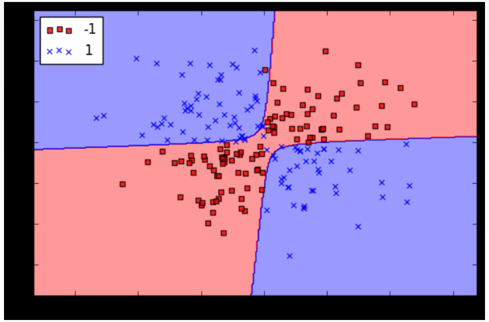


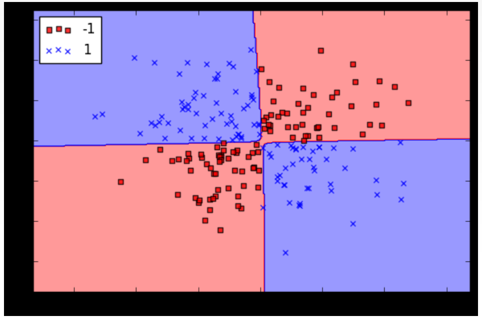


C variation – 1, 10, 1000, 10000









Properties of SVM

- Wanted requirement: all features vary on a similar scale
- For SVM to work, data may need to be preprocessed
    - Common: all features are normalised to values between 0 and 1
- SVM models work regardless of how many features there are (dimensionality of feature space does not matter)
- SVM do not scale very well with the number of samples
- SVM models are hard to inspect and it is difficult to explain why they make a particular prediction

Linear models for multi class classification

One-vs.-rest approach

1. Separate each class from all the rest
2. A binary model is learned for each class
3. To make a prediction, all binary classifiers are run on one point. The classifier with the highest score determines the class

uib.no