MACHINE LEARNING

LINEAR MODELS

Dr G.Kalyani

Department of Information Technology

Velagapudi Ramakrishna Siddhartha Engineering College

Topics

- **The Least-Squares Method**

- **Multivariate Linear Regression**

- **Support Vector Machines**
  - **Soft Margin SVM,**
  - **Going beyond Linearity with Kernel Methods.**

Linear Models

- We can use geometric concepts such as lines and planes to build a classification model.

- Models that can be understood in terms of lines and planes, commonly called **Linear Models.**

Characteristics of Linear Models

- **Linear models are parametric, meaning that they have a fixed form with a number of numeric parameters** that need to be learned from data. This is different from tree or rule models, where the structure of the model (e.g., which features to use in the tree, and where) is not fixed in advance.

- **Linear models are stable, which is to say that small variations in the training data have only limited impact on the learned model.** Tree models tend to vary more with the training data, as the choice of a different split at the root of the tree typically means that the rest of the tree is different as well.

- **Linear models are less likely to overfit the training data** than some other models, largely because they have relatively few parameters. The flipside of this is that **they sometimes lead to underfitting**: *e.g., imagine you are learning where the* border runs between two countries from labelled samples, then a linear model is unlikely to give a good approximation.

Linear Models

- Linear models generally have low variance and high bias.
- Linear models are often preferable when you have limited data and want to avoid overfitting.
- High variance–low bias models such as decision trees are preferable if data is abundant.
- It is usually a good idea to start with simple, high-bias models such as linear models and only move on to more elaborate models if the simpler ones appear to be under-fitting.

Linear Models

- Linear models exist for all predictive tasks, including classification, probability estimation and regression.

- Linear regression, in particular, is a well-studied problem that can be solved by the least-squares method

Least-Squares Method

- Recall that the regression problem is to learn a function estimator *f^ :X →R from examples (xi , f (xi))* .

- *The differences* between the actual and estimated function values on the training examples are called Residual.

  $Residual_i = f(xi) - \hat{f}(xi)$.

- *The Least-Squares Method, introduced by Carl* Friedrich Gauss, consists in finding ˆ *f such that* $\sum_{i=1}^{n} \epsilon_i^2$ *is* minimised.

- **In case of a single feature, the regression is called Univariate Regression.**

UNIVARIATE REGRESSION.

Example 7.1 (Univariate linear regression). Suppose we want to investigate the relationship between people's height and weight. We collect n height and weight measurements $(h_i, w_i), 1 \leq i \leq n$. Univariate linear regression assumes a linear equation $w = a + bh$, with parameters a and b chosen such that the sum of squared residuals $\sum_{i=1}^{n}(w_i - (a + bh_i))^2$ is minimised. In order to find the parameters we take partial derivatives of this expression, set the partial derivatives to 0 and solve for a and b:

$$\frac{\partial}{\partial a}\sum_{i=1}^{n}(w_i - (a+bh_i))^2 = -2\sum_{i=1}^{n}(w_i - (a+bh_i)) = 0 \qquad \Rightarrow \hat{a} = \overline{w} - \hat{b}\overline{h}$$

$$\frac{\partial}{\partial b}\sum_{i=1}^{n}(w_i - (a+bh_i))^2 = -2\sum_{i=1}^{n}(w_i - (a+bh_i))h_i = 0$$

$$\Rightarrow \hat{b} = \frac{\sum_{i=1}^{n}(h_i - \overline{h})(w_i - \overline{w})}{\sum_{i=1}^{n}(h_i - \overline{h})^2}$$

Example for UNIVARIATE REGRESSION.

Feature	Target
"h"	"w"
height	Weight
2	4
3	5
5	7
7	10
9	15

Univariate Linear Regression Equation:

$$w = a + bh$$

$$\Rightarrow \hat{b} = \frac{\sum_{i=1}^{n}(h_i - \overline{h})(w_i - \overline{w})}{\sum_{i=1}^{n}(h_i - \overline{h})^2}$$

$$\Rightarrow \hat{a} = \overline{w} - \hat{b}\overline{h}$$

Example for UNIVARIATE REGRESSION.

	Feature	Target				
	"h"	"w"				
	height	Weight	h-mean(h)	w-mean(w)	(h-mean(h))* (w-mean(w))	(h-mean(h))^2
	2	4	-3.2	-4.2	13.44	10.24
	3	5	-2.2	-3.2	7.04	4.84
	5	7	-0.2	-1.2	0.24	0.04
	7	10	1.8	1.8	3.24	3.24
	9	15	3.8	6.8	25.84	14.44
Mean	5.2	8.2		SUM	49.8	32.8

Example for UNIVARIATE REGRESSION.

	Feature	Target
	"h"	"w"
	height	Weight
	2	4
	3	5
	5	7
	7	10
	9	15

Univariate Linear Regression Equation:

$$w = a + bh$$

$$\Rightarrow \hat{b} = \frac{\sum_{i=1}^{n}(h_i - \overline{h})(w_i - \overline{w})}{\sum_{i=1}^{n}(h_i - \overline{h})^2}$$

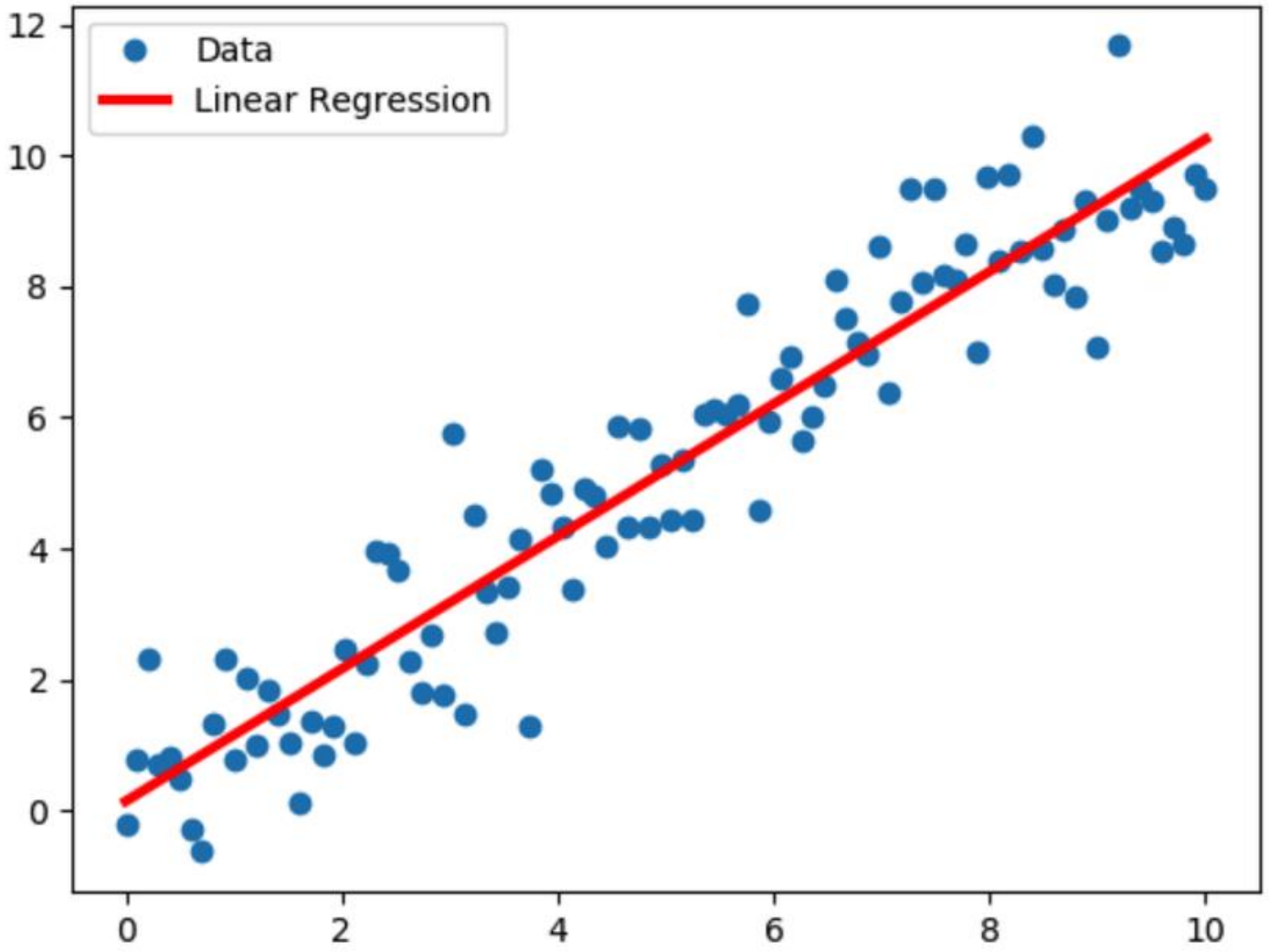
b = 49.8/32.8 = 1.52

$$\Rightarrow \hat{a} = \overline{w} - \hat{b}\overline{h}$$

a = 8.2 – 1.52 * 5.2 = 0.3

Univariate Linear Regression Equation: ***w = 0.3 + 1.52 * h***

Example for UNIVARIATE REGRESSION.

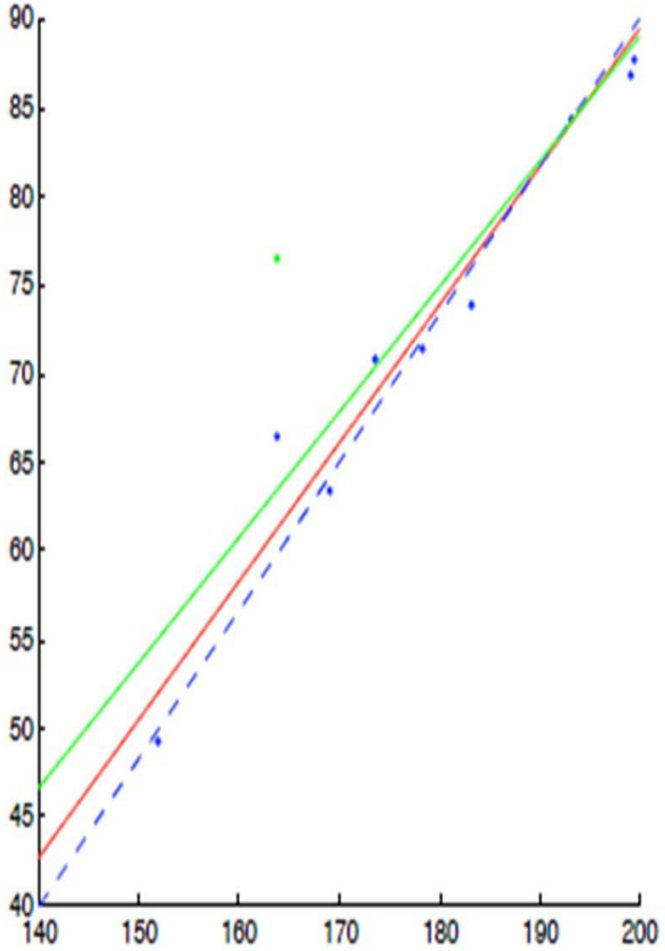


Effect of Outliers on linear regression

- linear regression is susceptible to *outliers:* *points that are far moved from the regression line, often* because of measurement errors.

- Suppose that, as the result of a transcription error, one of the weight values in is increased by 10 kg.

- A considerable effect on the least-squares regression line will be visible.



Topics

- **The Least-Squares Method**
- **Multivariate Linear Regression**
- **Support Vector Machines**
  - **Soft Margin SVM**
  - **Going beyond Linearity with Kernel Methods.**

Multivariate linear regression

This is quite similar to the simple linear regression model we have discussed previously, but with multiple independent variables contributing to the dependent variable and hence multiple coefficients to determine and complex computation due to the added variables. Jumping straight into the equation of multivariate linear regression,

$$Y_i = \alpha + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)} + \ldots + \beta_n x_i^{(n)}$$

Y_i is the estimate of i^{th} component of dependent variable y, where we have n independent variables and x_i^j denotes the i^{th} component of the j^{th} independent variable/feature. Similarly cost function is as follows,

$$E(\alpha, \beta_1, \beta_2, \ldots, \beta_n) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - Y_i)$$

Example for Multivariate linear regression

Suppose we have the following dataset with one response variable *y* and two predictor variables X_1 and X_2:

y	X_1	X_2
140	60	22
155	62	25
159	67	24
179	70	20
192	71	15
200	72	14
212	75	14
215	78	11

Example for Multivariate linear regression

The estimated linear regression equation is: $\hat{y} = b_0 + b_1 * x_1 + b_2 * x_2$

$$b_1 = \frac{[(\sum x_2{}^2)(\sum x_1 y) - (\sum x_1 x_2)(\sum x_2 y)]}{[(\sum x_1{}^2)(\sum x_2{}^2) - (\sum x_1 x_2)^2]}$$

$$b_2 = \frac{[(\sum x_1{}^2)(\sum x_2 y) - (\sum x_1 x_2)(\sum x_1 y)]}{[(\sum x_1{}^2)(\sum x_2^2) - (\sum x_1 x_2)^2]}$$

$$b_0 = \bar{y} - b_1 \bar{x}_1 - b_2 \bar{x}_2$$

Note: All variable are regression sum variables

Example for Multivariate linear regression

Step 1: Calculate X_1^2, X_2^2, X_1y, X_2y and X_1X_2.

	y	X_1	X_2
	140	60	22
	155	62	25
	159	67	24
	179	70	20
	192	71	15
	200	72	14
	212	75	14
	215	78	11
Mean	181.5	69.375	18.125
Sum	1452	555	145

X_1^2	X_2^2	X_1y	X_2y	X_1X_2
3600	484	8400	3080	1320
3844	625	9610	3875	1550
4489	576	10653	3816	1608
4900	400	12530	3580	1400
5041	225	13632	2880	1065
5184	196	14400	2800	1008
5625	196	15900	2968	1050
6084	121	16770	2365	858
Sum 38767	2823	101895	25364	9859

Example for Multivariate linear regression

Step 2: Calculate Regression Sums.

Next, make the following regression sum calculations:

- $\Sigma x_1^2 = \Sigma X_1^2 - (\Sigma X_1)^2 / n = 38{,}767 - (555)^2 / 8 =$ **263.875**
- $\Sigma x_2^2 = \Sigma X_2^2 - (\Sigma X_2)^2 / n = 2{,}823 - (145)^2 / 8 =$ **194.875**
- $\Sigma x_1 y = \Sigma X_1 y - (\Sigma X_1 \Sigma y) / n = 101{,}895 - (555*1{,}452) / 8 =$ **1,162.5**
- $\Sigma x_2 y = \Sigma X_2 y - (\Sigma X_2 \Sigma y) / n = 25{,}364 - (145*1{,}452) / 8 =$ **-953.5**
- $\Sigma x_1 x_2 = \Sigma X_1 X_2 - (\Sigma X_1 \Sigma X_2) / n = 9{,}859 - (555*145) / 8 =$ **-200.375**

Example for Multivariate linear regression

Step 3: Calculate b_0, b_1, and b_2.

The formula to calculate b_1 is: $[(\Sigma x_2^2)(\Sigma x_1 y) - (\Sigma x_1 x_2)(\Sigma x_2 y)] \ / \ [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$

Thus, $\mathbf{b_1} = [(194.875)(1162.5) - (-200.375)(-953.5)] \ / \ [(263.875)(194.875) - (-200.375)^2] = \mathbf{3.148}$

The formula to calculate b_2 is: $[(\Sigma x_1^2)(\Sigma x_2 y) - (\Sigma x_1 x_2)(\Sigma x_1 y)] \ / \ [(\Sigma x_1^2)(\Sigma x_2^2) - (\Sigma x_1 x_2)^2]$

Thus, $\mathbf{b_2} = [(263.875)(-953.5) - (-200.375)(1152.5)] \ / \ [(263.875)(194.875) - (-200.375)^2] = \mathbf{-1.656}$

The formula to calculate b_0 is: $\overline{y} - b_1\overline{X}_1 - b_2\overline{X}_2$

Thus, $\mathbf{b_0} = 181.5 - 3.148(69.375) - (-1.656)(18.125) = \mathbf{-6.867}$

Example for Multivariate linear regression

Step 4: Place b_0, b_1, and b_2 in the estimated linear regression equation.

The estimated linear regression equation is: $\hat{y} = b_0 + b_1 * x_1 + b_2 * x_2$

In our example, it is $\hat{y} = -6.867 + 3.148x_1 - 1.656x_2$

Multivariate linear regression

Now let us talk in terms of matrices as it is easier that way. As discussed before, if we have n independent variables in our training data, our matrix X has $n+1$ rows, where the first row is the 0^{th} term added to each vector of independent variables which has a value of 1 (this is the coefficient of the constant term α). So, X is as follows,

$$X = \begin{bmatrix} X_1 \\ .. \\ X_m \end{bmatrix}$$

X^i contains n entries corresponding to each feature in training data of i^{th} entry. So, matrix X has m rows and $n+1$ columns ($0^{th} column$ is all 1^s and rest for one independent variable each).

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ .. \\ Y_m \end{bmatrix}$$

and coefficient matrix C,

$$C = \begin{bmatrix} \alpha \\ \beta_1 \\ .. \\ \beta_n \end{bmatrix}$$

and our final equation for our hypothesis is,

$$Y = XC$$

Multivariate linear regression

$$C = (X^T X)^{-1} X^T y$$

$(\mathbf{X^T X})^{-1}$ ***acts*** *as a transformation that decorrelates, centres and normalises the features.*

uncorrelated features effectively decomposes a multivariate regression problem into d univariate problems.

Topics

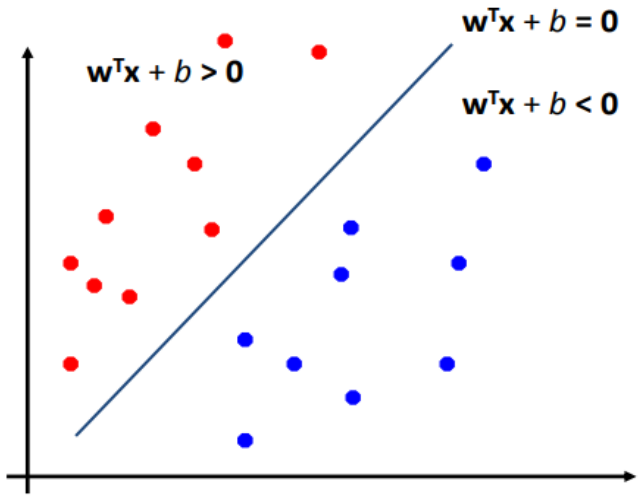
- **The Least-Squares Method**
- **Multivariate Linear Regression**
- **Support Vector Machines**
  - **Soft Margin SVM**
  - **Going Beyond Linearity with Kernel Methods**

Support Vector Machine(SVM)

- Classification: Searches for optimal hyperplane/ decision boundary separating the tuples of one class from another.
- The decision boundary of a Support Vector Machine (SVM) is defined as a linear combination of the support vectors.

$$y = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \ldots$$
$$= w_0 + \sum_{i=1}^{m} w_i x_i$$
$$= w_0 + w^T X$$
$$= b + w^T X$$

X: Input Vector
W:Weight Vector
b: bias



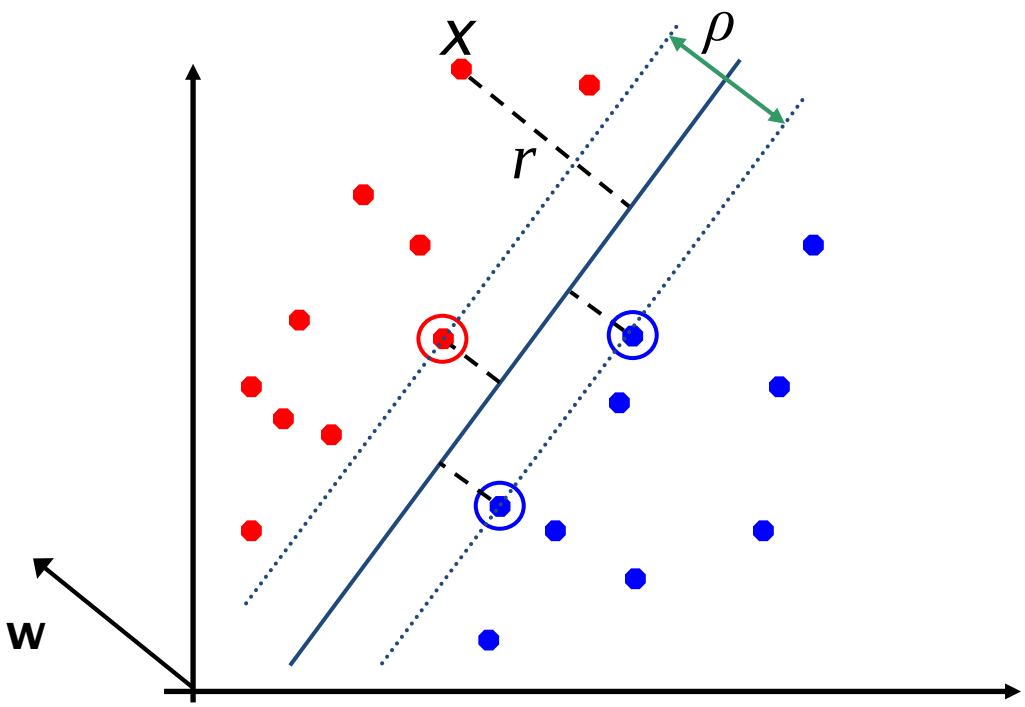
Support Vector Machine(SVM)

- If Feature vector is 2D then the linear equation represents straight line.
- If Feature vector is 3D then the linear equation represents hyper plane.
- W represents orientation & b represents position.
- **$g(x_1)=w^t x_1+b>0$ implies x_1 belongs to C1**
- **$g(x_1)=w^t x_1+b<0$ implies x_1 belongs to C2**

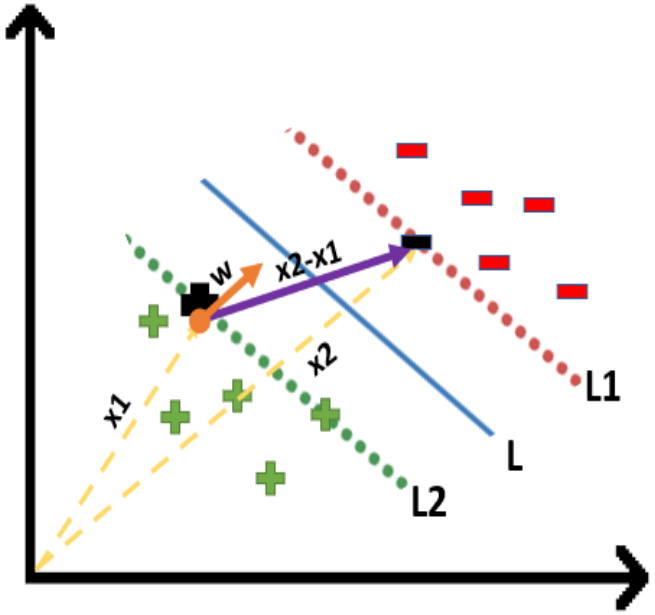
Geometric Margin

- Distance from example to the separator is $r = y\dfrac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors (encircled)***.
- ***Margin*** ρ of the separator is the width of separation between support vectors of classes.

$$r = \frac{2}{\|\mathbf{w}\|}$$



Geometric Margin



$$\Rightarrow (x2 - x1).\frac{\vec{w}}{\|w\|}$$

$$\Rightarrow \frac{x2.\vec{w} - x1.\vec{w}}{\|w\|} \quad ----(1)$$

for positive point $y = 1$

$$\Rightarrow 1 \times (\vec{w}.x1 + b) = 1$$

$$\Rightarrow \vec{w}.x1 = 1 - b \quad -----(2)$$

Similarly for negative point $y = -1$

$$\Rightarrow -1 \times \left(\vec{w}.x2 + b\right) = 1$$

$$\Rightarrow \vec{w}.x2 = -b - 1 \quad -----(3)$$

$$\Rightarrow \frac{(-1 - b) - (1 - b)}{\| w \|}$$

$$\Rightarrow \frac{-1 - b - 1 + b}{\| w \|} = \frac{-2}{\| w \|} = \frac{2}{\| w \|} = d$$

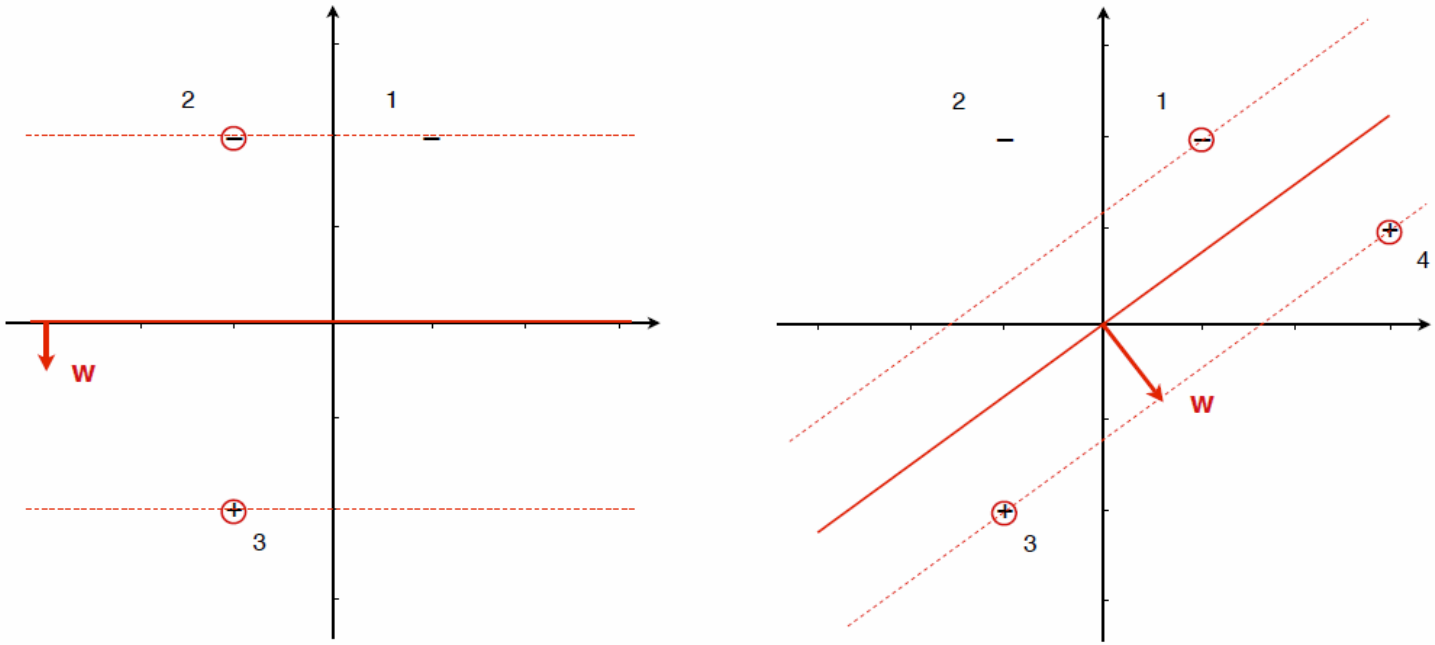
Support Vector Machine(SVM)

Maximising the margin then corresponds to minimising $||\mathbf{w}||$ or, more conveniently, $\frac{1}{2}||\mathbf{w}||^2$, provided of course that none of the training points fall inside the margin.

This leads to a quadratic, constrained optimisation problem:

$$\mathbf{w}^*, t^* = \arg\min_{\mathbf{w},t} \frac{1}{2}||\mathbf{w}||^2 \qquad \text{subject to } y_i(\mathbf{w}\cdot\mathbf{x}_i - t) \geq 1, 1 \leq i \leq n$$

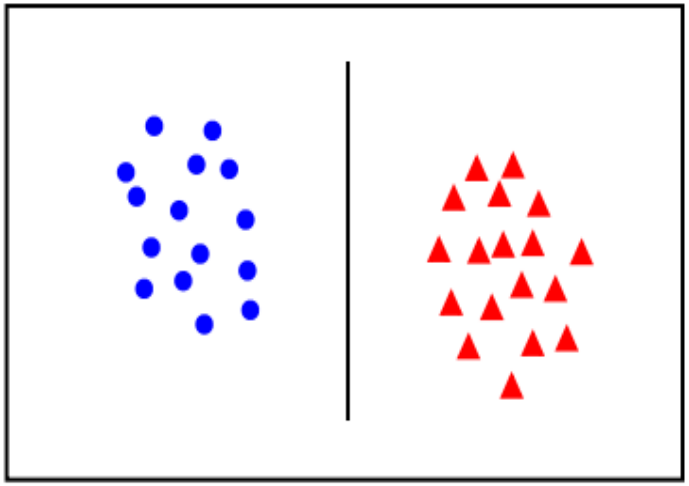
Support Vector Machine(SVM)

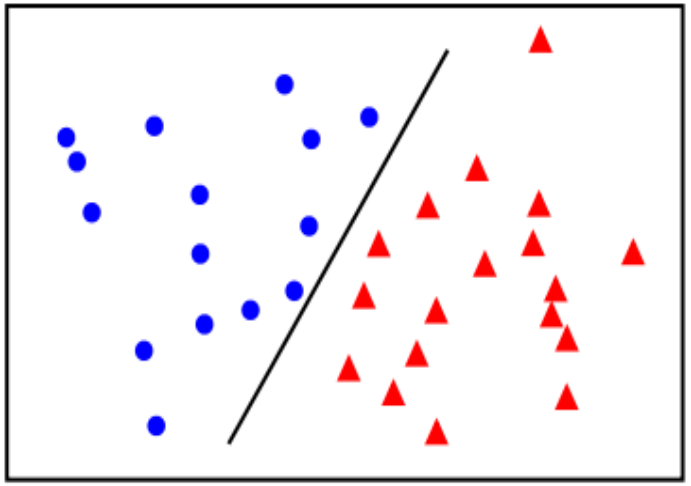


(left) A maximum-margin classifier built from three examples, with $\mathbf{w} = (0, -1/2)$ and margin 2. The circled examples are the support vectors: they receive non-zero Lagrange multipliers and define the decision boundary. **(right)** By adding a second positive the decision boundary is rotated to $\mathbf{w} = (3/5, -4/5)$ and the margin decreases to 1.

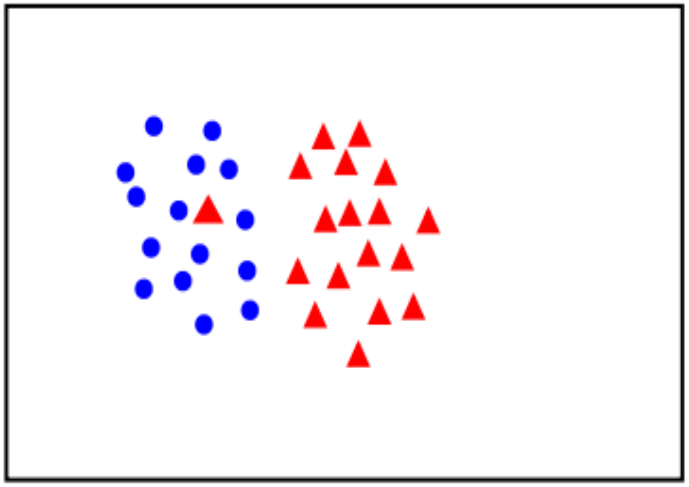
Linearly separable/not

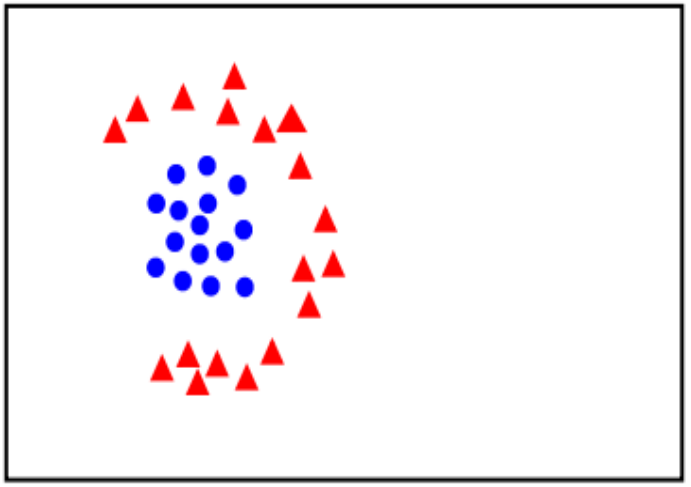
linearly separable





not linearly separable





Soft Margin SVM

- Two solutions to handle not linearly separable data:
  - **Soft Margin SVM:** Still can try to find a line to separate the data, but we tolerate one or few misclassified dots.
  - **Kernel Trick:** Try to find a non-linear decision boundary to separate red and blue dots.

- What Soft Margin does is
  - it tolerates a few data to get misclassified
  - it tries to balance the **trade-off between finding a line that maximizes the margin and minimizes the misclassification.**

- How much tolerance we want to set when finding the decision boundary is an important hyper-parameter for the SVM

“Soft” margin solution

The optimization problem becomes

$$\min_{\mathbf{w}\in\mathbb{R}^d,\xi_i\in\mathbb{R}^+} ||\mathbf{w}||^2+C\sum_i^N \xi_i$$

subject to

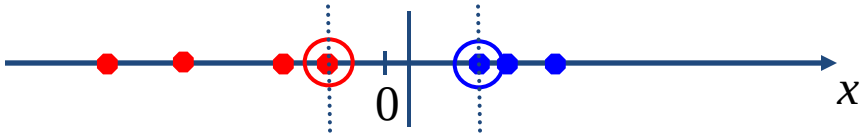
$$y_i\left(\mathbf{w}^\top\mathbf{x}_i+b\right)\geq 1-\xi_i \text{ for } i=1\ldots N$$

- Every constraint can be satisfied if ξ_i is sufficiently large
- C is a regularization parameter:
  - small C allows constraints to be easily ignored $\rightarrow$ large margin
  - large C makes constraints hard to ignore $\rightarrow$ narrow margin
  - $C=\infty$ enforces all constraints: hard margin

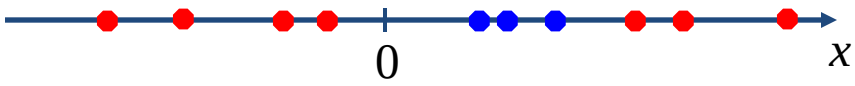
C is used as Trade-off between miss classification and large margin

Kernel Trick for Non-linear SVMs

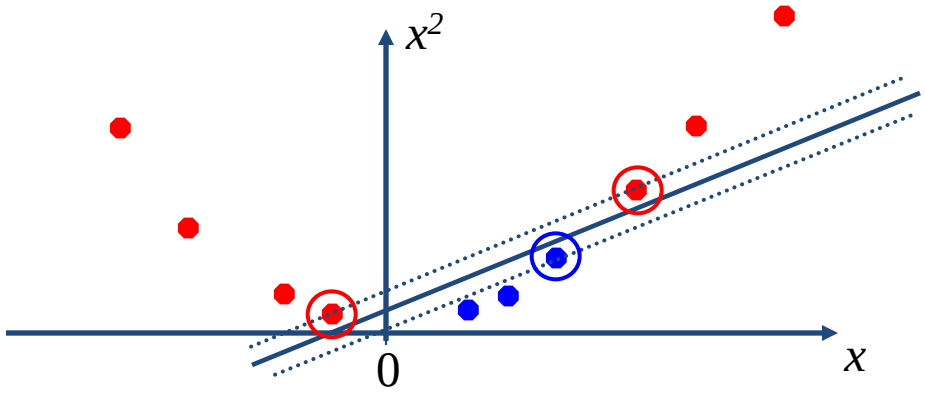
- Datasets that are linearly separable (with some noise) work out great:



- But what are we going to do if the dataset is just too hard?

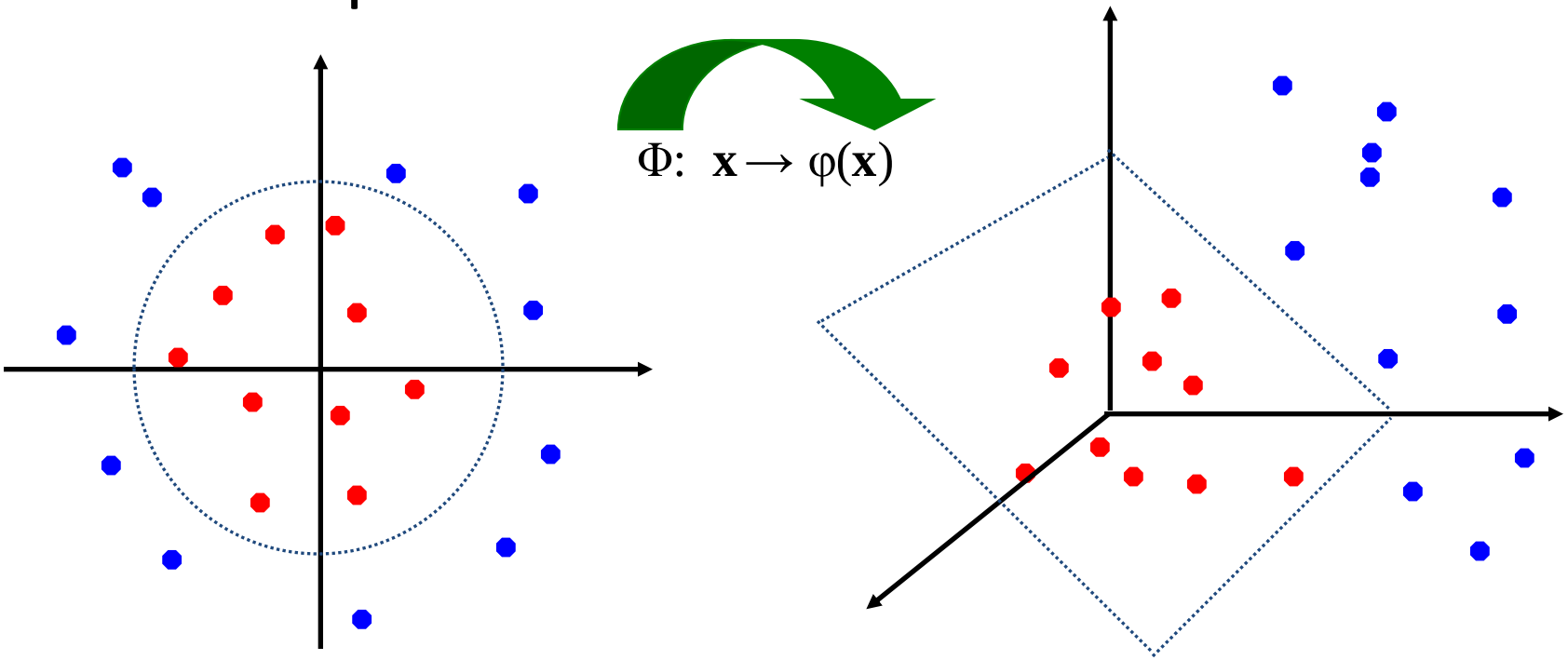


- How about … mapping data to a higher-dimensional space:



Kernel Trick for Non-linear SVMs

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:



Kernel Trick for Non-linear SVMs

- Mathematical definition: K(x, y) = <f(x), f(y)>
  - Here K is the kernel function
  - x, y are n dimensional inputs.
  - f is a map from n-dimension to m-dimension space.
  - < x,y> denotes the dot product.
  - Usually m is much larger than n.

Different Kernel Functions

- Different types of kernels
  - Linear
  - Polynomial
  - Gaussian (RBF)

Linear kernel: $K(x_i, x_j) = x_i . x_j$

Polynomial of power p:

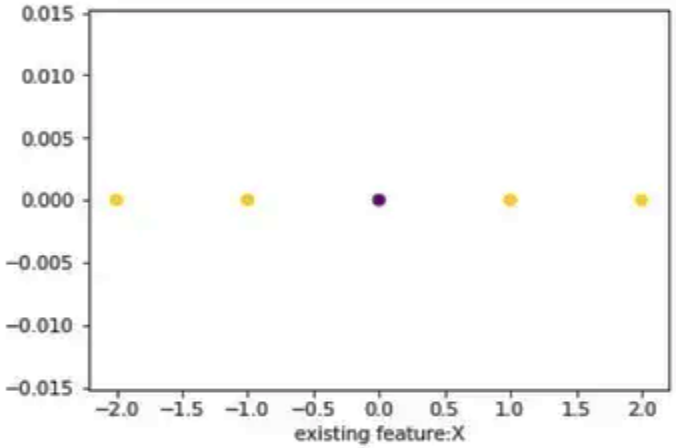
$$K(x_i, x_j) = (1 + x_i . x_j)^p$$

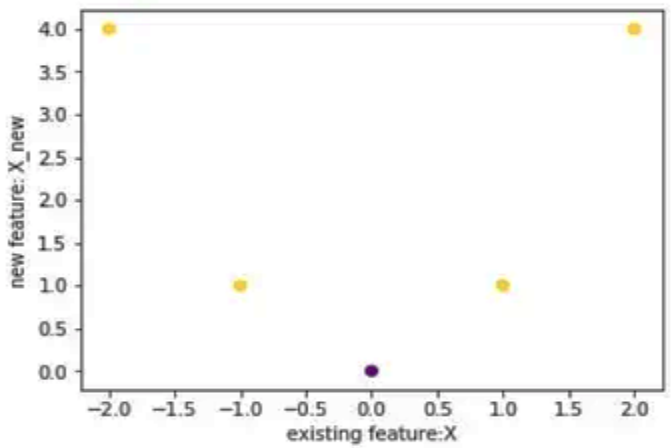
Gaussian (radial-basis function):

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

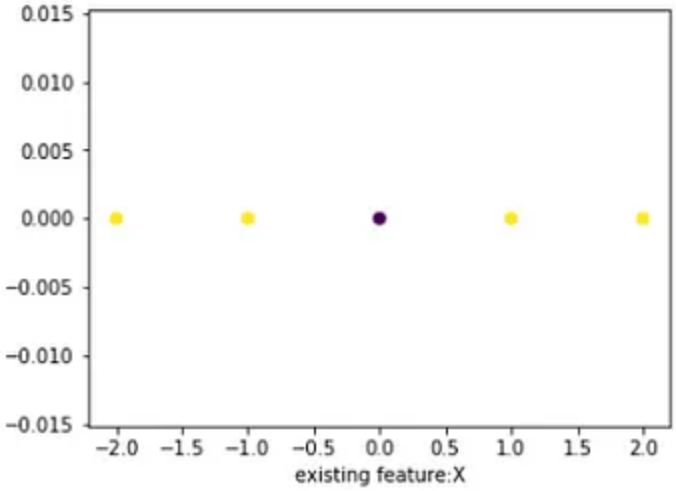
Different Kernel Functions

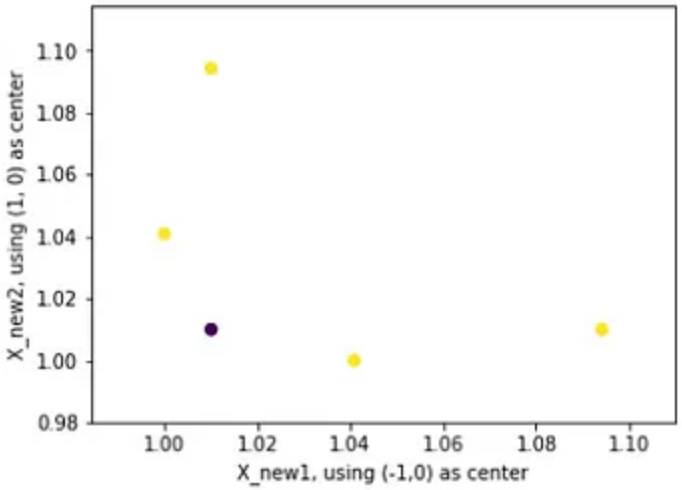
- **Polynomial Kernel**





- **Gaussian (RBF) Kernel**





Example for Linear SVM

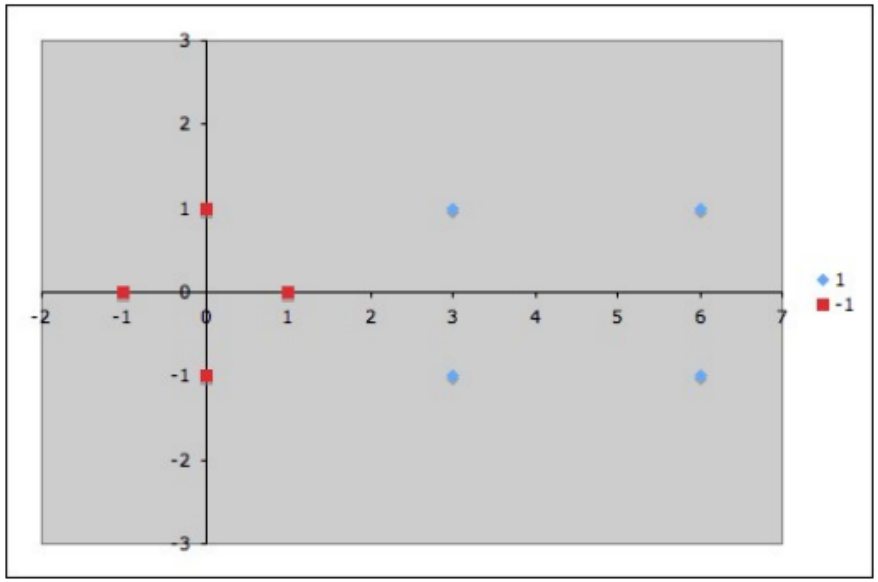
- We would like to discover a simple SVM that accurately discriminates the two classes.

Suppose we are given the following positively labeled data points in $\Re^2$:

$$\left\{\begin{pmatrix}3\\1\end{pmatrix},\begin{pmatrix}3\\-1\end{pmatrix},\begin{pmatrix}6\\1\end{pmatrix},\begin{pmatrix}6\\-1\end{pmatrix}\right\}$$

and the following negatively labeled data points in $\Re^2$ (see Figure 1):

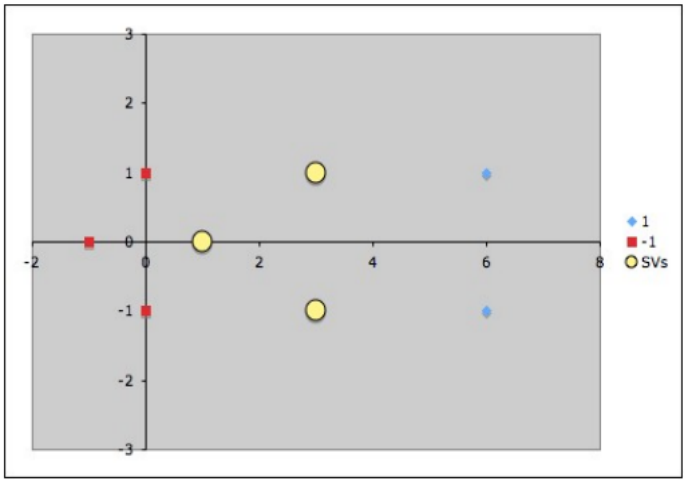
$$\left\{\begin{pmatrix}1\\0\end{pmatrix},\begin{pmatrix}0\\1\end{pmatrix},\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}-1\\0\end{pmatrix}\right\}$$



Example for Linear SVM

The support vectors are:

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \right\}$$



Two +ve support vectors and one –ve support vector. Hence frame 3 equations:

$$\begin{aligned} \alpha_1 \tilde{s_1} \cdot \tilde{s_1} + \alpha_2 \tilde{s_2} \cdot \tilde{s_1} + \alpha_3 \tilde{s_3} \cdot \tilde{s_1} &= -1 \\ \alpha_1 \tilde{s_1} \cdot \tilde{s_2} + \alpha_2 \tilde{s_2} \cdot \tilde{s_2} + \alpha_3 \tilde{s_3} \cdot \tilde{s_2} &= +1 \\ \alpha_1 \tilde{s_1} \cdot \tilde{s_3} + \alpha_2 \tilde{s_2} \cdot \tilde{s_3} + \alpha_3 \tilde{s_3} \cdot \tilde{s_3} &= +1 \end{aligned}$$

Assume each support vector as 3dimensions (since (n+1) coefficients are to be identified):
i.e., S1=(1, 0, 1), s2=(3, 1, 1) and s3= (3, -1, 1)

Example for Linear SVM

$$
\begin{aligned}
2\alpha_1 + 4\alpha_2 + 4\alpha_3 &= -1 \\
4\alpha_1 + 11\alpha_2 + 9\alpha_3 &= +1 \\
4\alpha_1 + 9\alpha_2 + 11\alpha_3 &= +1
\end{aligned}
$$

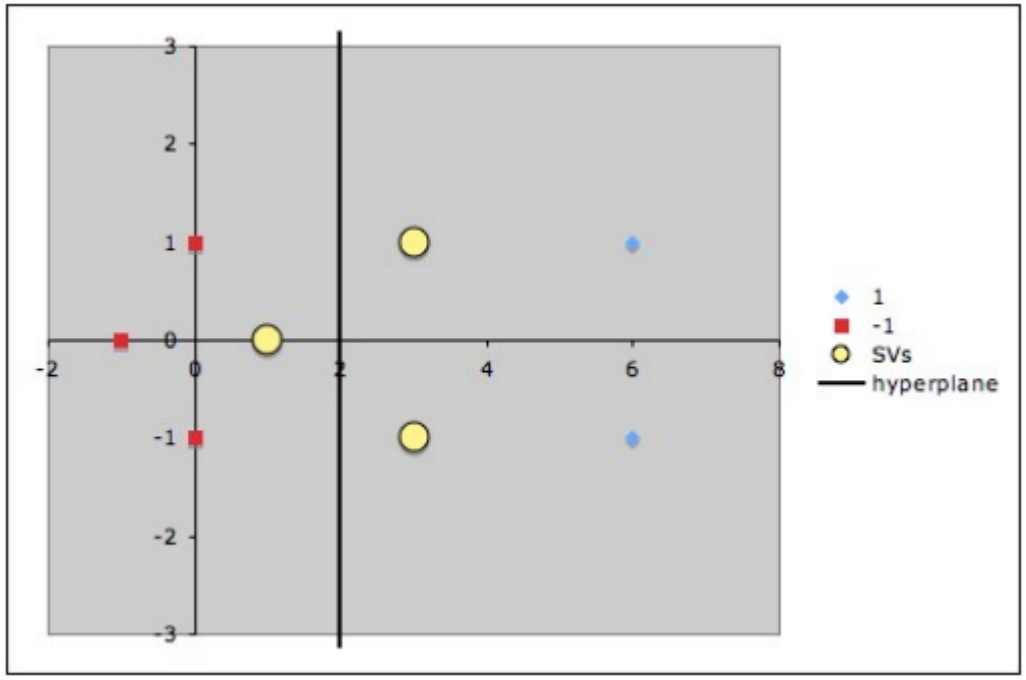
A little algebra reveals that the solution to this system of equations is $\alpha 1 = -3.5$, $\alpha 2 = 0.75$ and $\alpha 3 = 0.75$.

Calculating the weights based on $\alpha 1$, $\alpha 2$ and $\alpha 3$:

$$
\begin{aligned}
\tilde{w} &= \sum_i \alpha_i \tilde{s}_i \\
&= -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}
\end{aligned}
$$

Example for Linear SVM

Finally, remembering that our vectors are augmented with a bias, we can equate the last entry in $\tilde{w}$ as the hyperplane offset b and write the separating hyperplane equation $y = wx + b$ with $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $b = -2$. Plotting the line gives the expected decision surface



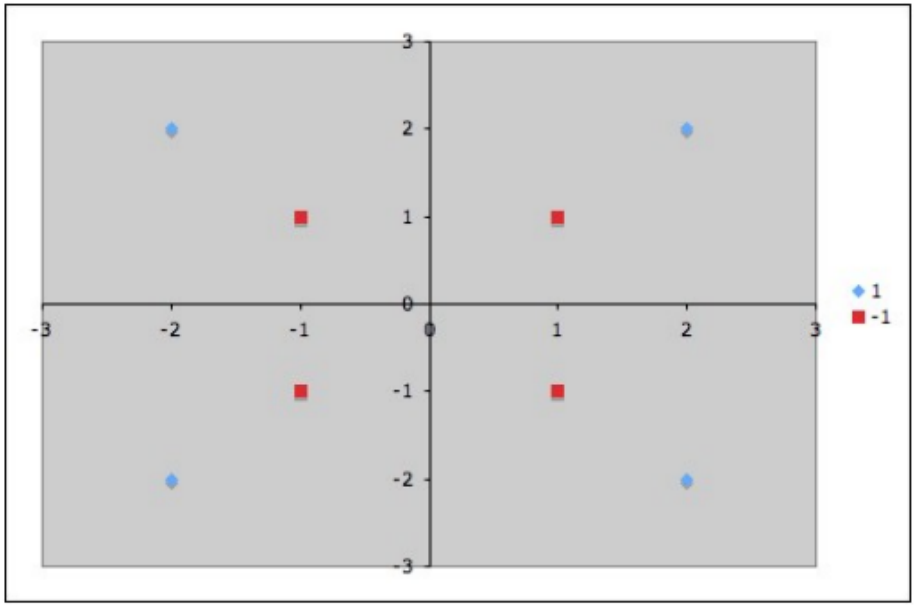
Example for Non-Linear SVM

Now suppose instead that we are given the following positively labeled data points in $\Re^2$:

$$\left\{\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix}\right\}$$

and the following negatively labeled data points in $\Re^2$ (see Figure 5):

$$\left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}\right\}$$



Example for Non-Linear SVM

- we must use a nonlinear SVM (that is, one whose mapping function Φ is a nonlinear mapping from input space into some feature space).

Define

$$\Phi_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} 4 - x_2 + |x_1 - x_2| \\ 4 - x_1 + |x_1 - x_2| \end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} > 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{otherwise} \end{cases}$$

- Given data

$$\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix} \right\}$$

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$$

Example for Non-Linear SVM

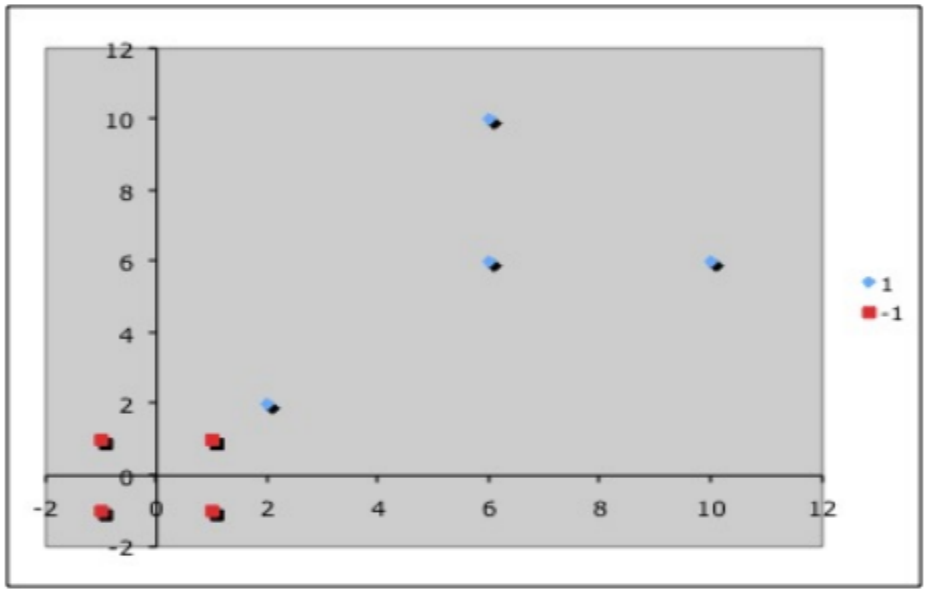
- we can rewrite the data in feature space as

$$\left\{\begin{pmatrix}2\\2\end{pmatrix},\begin{pmatrix}6\\2\end{pmatrix},\begin{pmatrix}6\\6\end{pmatrix},\begin{pmatrix}2\\6\end{pmatrix}\right\}$$

for the positive examples and

$$\left\{\begin{pmatrix}1\\1\end{pmatrix},\begin{pmatrix}1\\-1\end{pmatrix},\begin{pmatrix}-1\\-1\end{pmatrix},\begin{pmatrix}-1\\1\end{pmatrix}\right\}$$

for the negative examples



Example for Non-Linear SVM

Supporting vectors:

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, s_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\}$$

$$\begin{aligned} \alpha_1 \tilde{s_1} \cdot \tilde{s_1} + \alpha_2 \tilde{s_2} \cdot \tilde{s_1} &= -1 \\ \alpha_1 \tilde{s_1} \cdot \tilde{s_2} + \alpha_2 \tilde{s_2} \cdot \tilde{s_2} &= +1 \end{aligned}$$

Now, computing the dot products results in

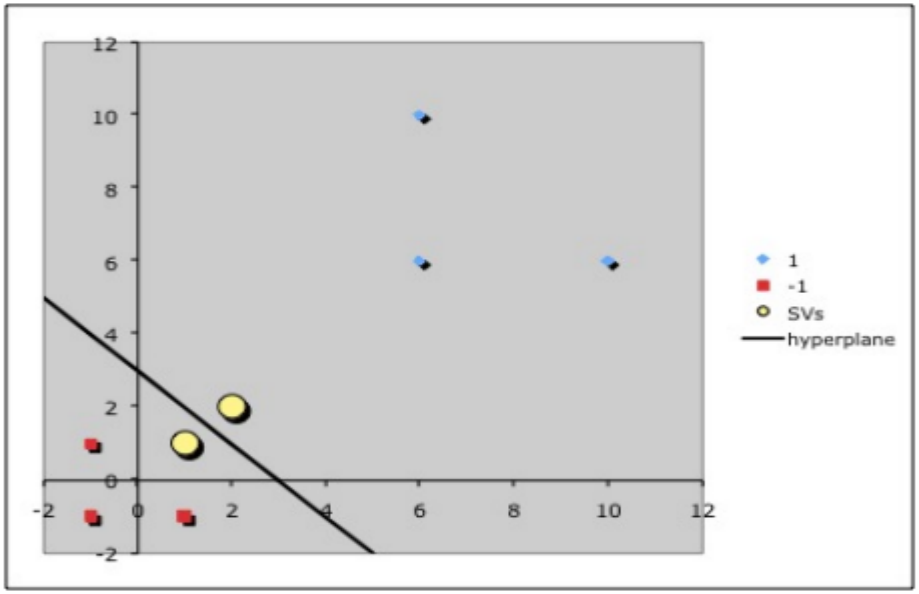
$$\begin{aligned} 3\alpha_1 + 5\alpha_2 &= -1 \\ 5\alpha_1 + 9\alpha_2 &= +1 \end{aligned}$$

Solving the equations gives $\alpha 1 = -7$ and $\alpha 2 = 4$.

Example for Non-Linear SVM

$$\begin{aligned}\tilde{w} &= \sum \alpha_i \tilde{s}_i \\ &= -7\begin{pmatrix}1\\1\\1\end{pmatrix} + 4\begin{pmatrix}2\\2\\1\end{pmatrix} \\ &= \begin{pmatrix}1\\1\\-3\end{pmatrix}\end{aligned}$$

giving us the separating hyperplane equation $y = wx + b$ with $w = \begin{pmatrix}1\\1\end{pmatrix}$ and $b = -3$. Plotting the line gives the expected decision surface .



Pro's and Con's of SVM

- **Pro's:**
  - It is really effective in the higher dimension.
  - Effective when the number of features are more than training examples.
  - Best algorithm when classes are separable
  - The hyperplane is affected by only the support vectors thus outliers have less impact.
  - SVM is suited for extreme case binary classification.
- **Con's:**
  - For larger dataset, it requires a large amount of time to process.
  - Does not perform well in case of overlapped classes.
  - Selecting the appropriate kernel function can be tricky.

Important points to remember

- The SVM's are less effective when the data is noisy and contains overlapping points

- The effectiveness of an SVM depends upon:
  - Selection of Kernel
  - Kernel Parameters
  - Soft Margin Parameter

Topics

- **The least-squares method**
- **Multivariate Linear Regression**
- **Support Vector Machines**
  - **Soft Margin SVM**
  - **Going beyond linearity with kernel methods**