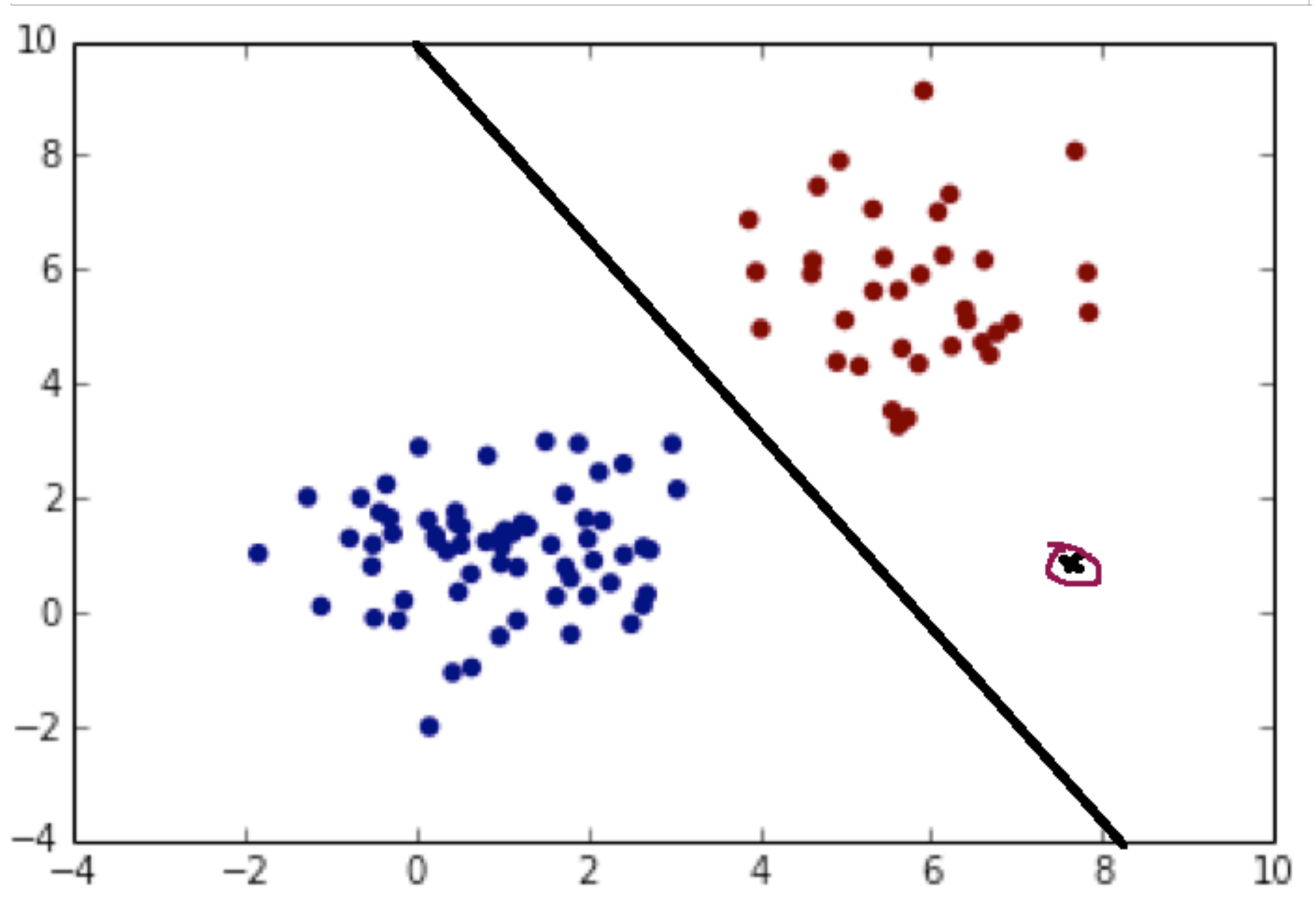
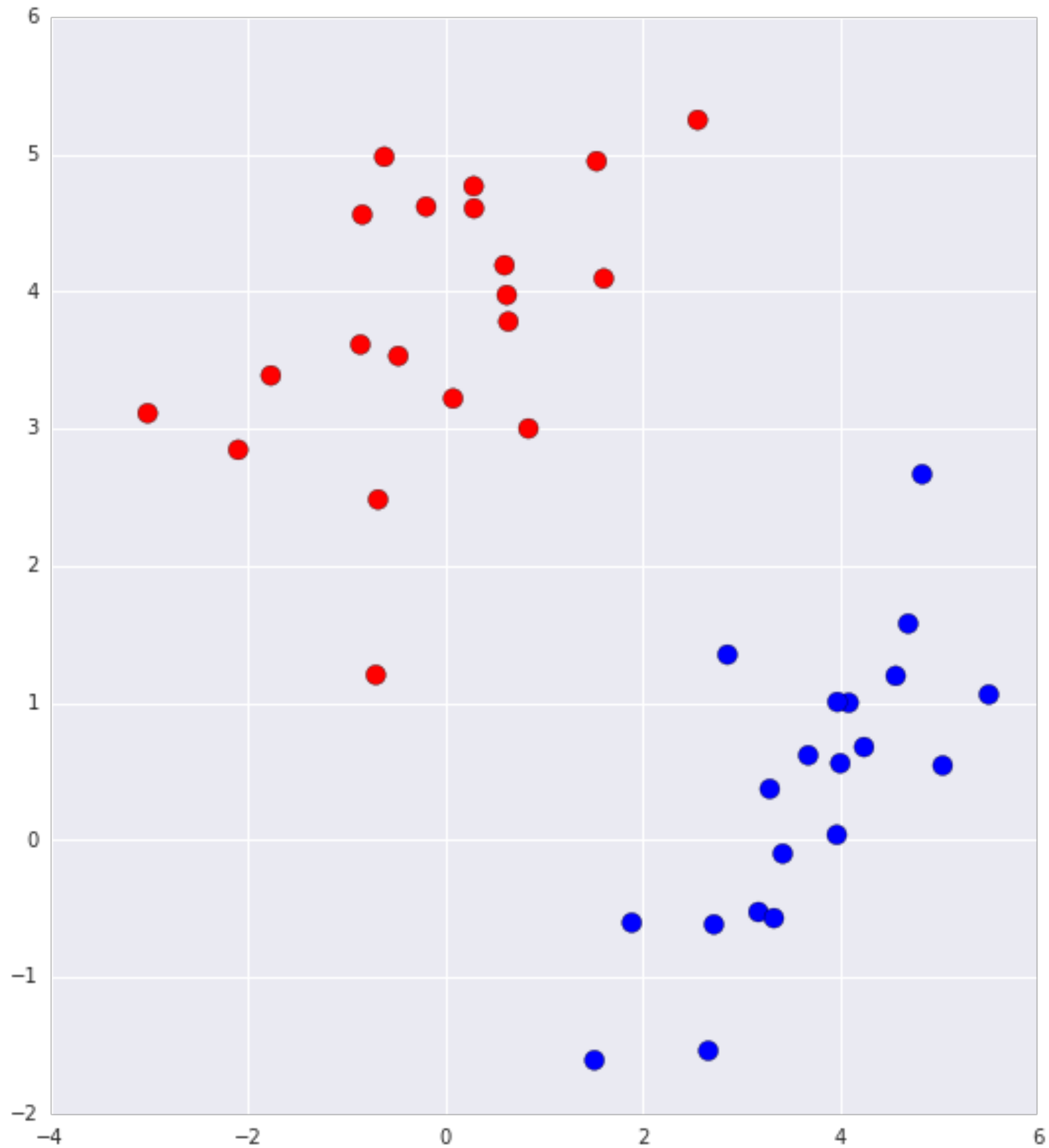
Machine Learning
5006.001/2, spring 2016
Session 7: Support Vector Machines

Instructors: Prof. Stanislav Sobolevsky, Dr. Martin Jankowiak, Dr. Ravi Schroff
Teaching Assistants: Lingjing Wang and Yash Chhajed

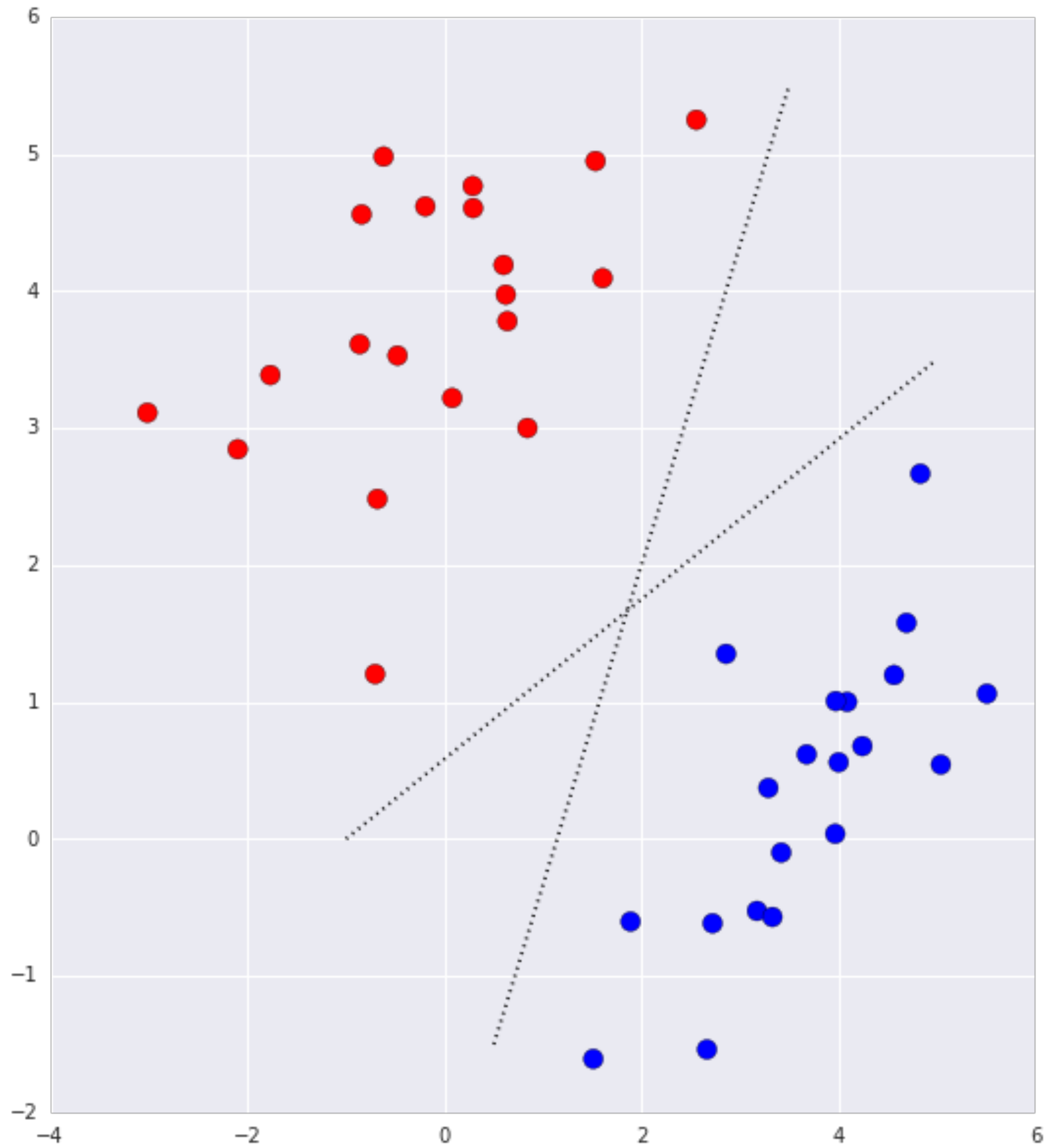
Support Vector Machines



SVM intuition

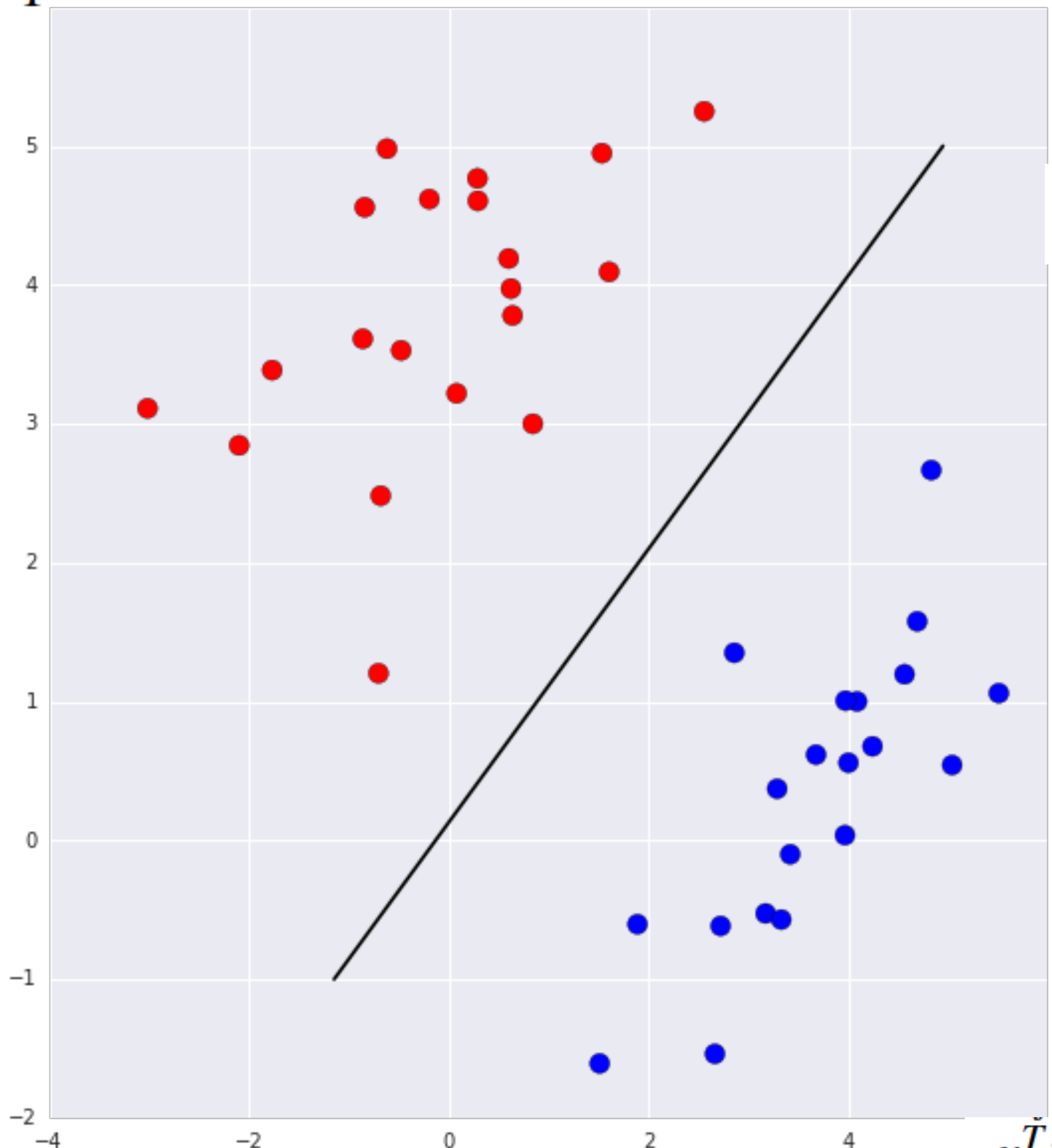


SVM intuition



SVM intuition

$$x_j^T w + b > 0,\ y_j = 1$$



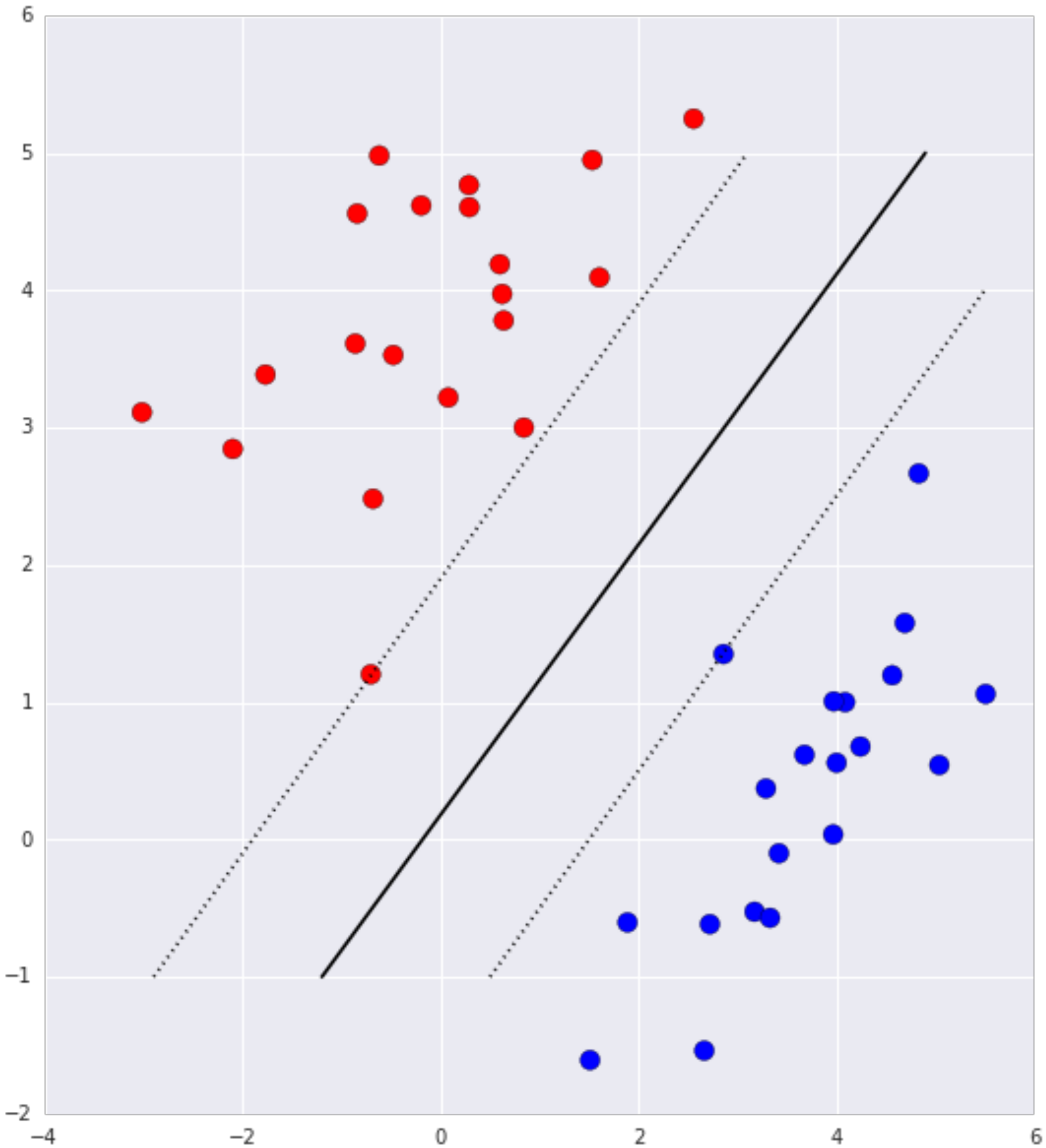
$$x^T w + b = 0$$

$$y_j(x_j^T w + b) > 0$$

$$x_j^T w + b < 0,\ y_j = -1$$

SVM intuition

$$y_j(x_j^T w + b) > 0$$



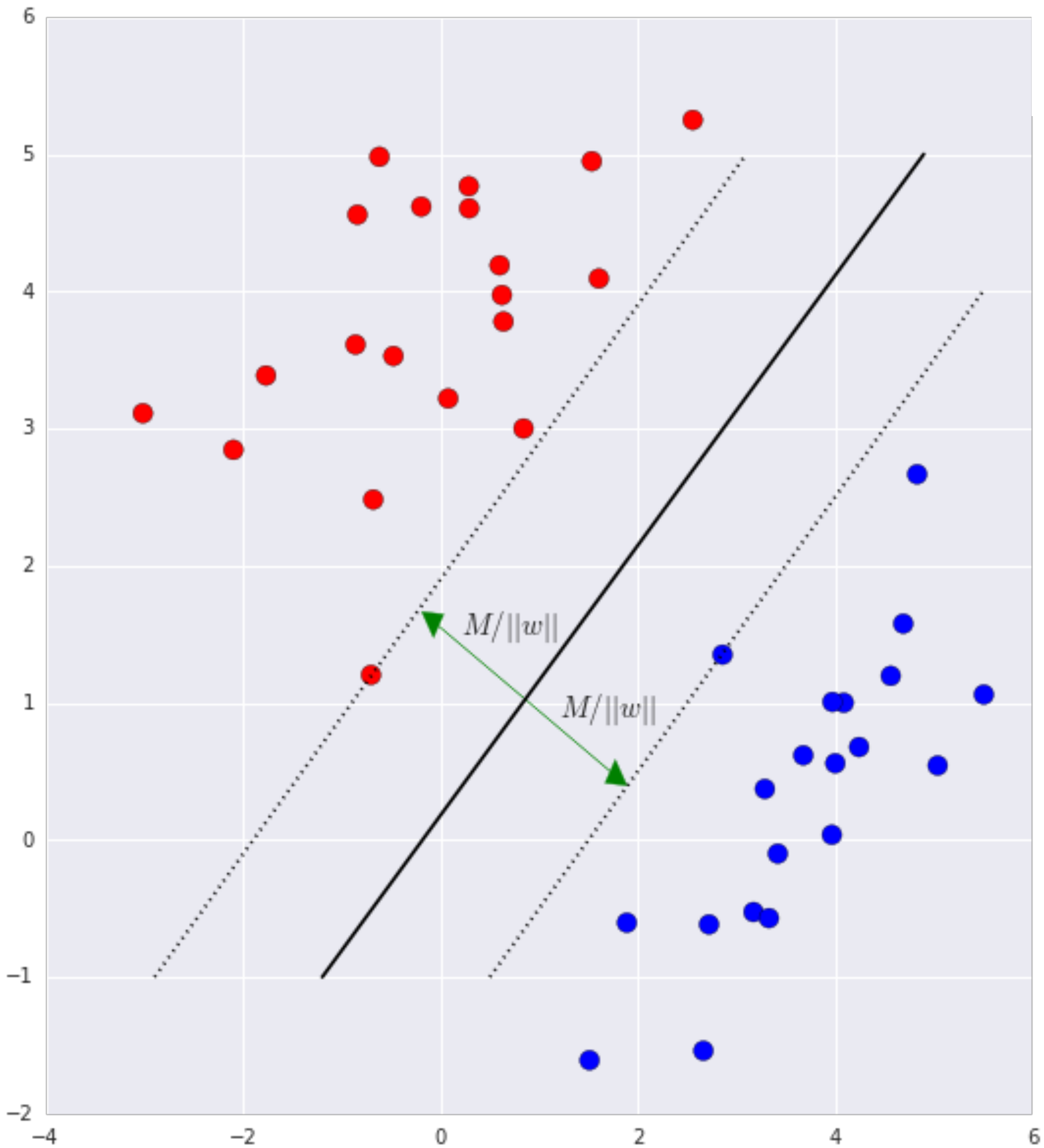
$$M = \min_j y_j(x_j^T + b) > 0$$

$$y_j(x_j^T w + b) \geq M$$

$$y_j = 1 : \; x_j^T w + b + M \geq 0$$

$$y_j = -1 : \; x_j^T w + b + M \leq 0$$

SVM intuition



$$y_j(x_j^T w + b) \geq M$$

$$||w||^2 = w^T w$$

$$||w||^2 = \sum_i (w^i)^2$$

$$M = M(w, b)$$

SVM maths

$$2M/||w|| \to \max$$

$$y_j(x_j^T w + b) \geq M$$

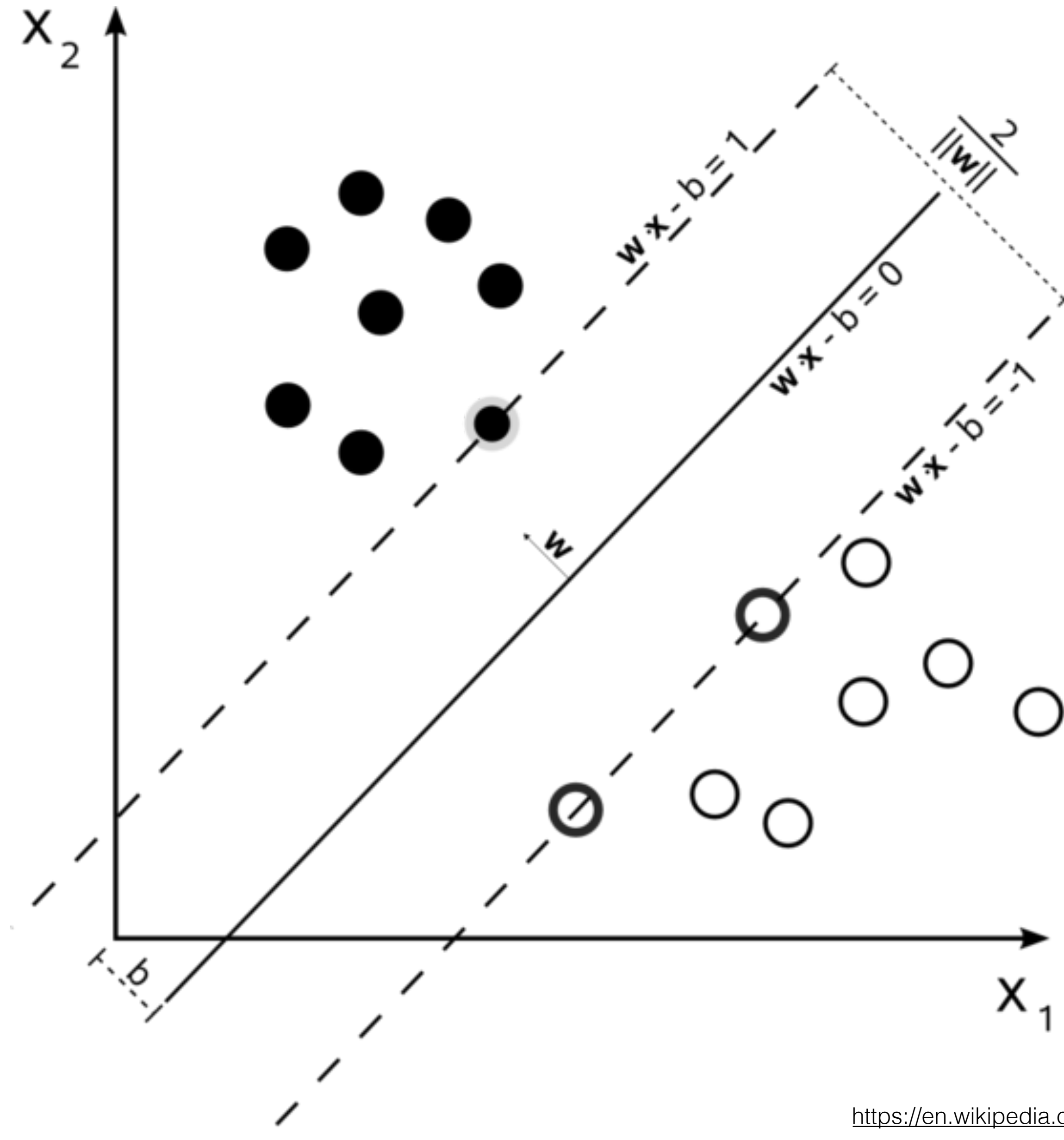
$$w := w/M(w,b), b := b/M(w,b)$$

$$y_j(x_j^T w + b) \geq 1$$

$$w^T w \to \min$$

$$y_j(x_j^T w + b) = 1$$ are called support vectors

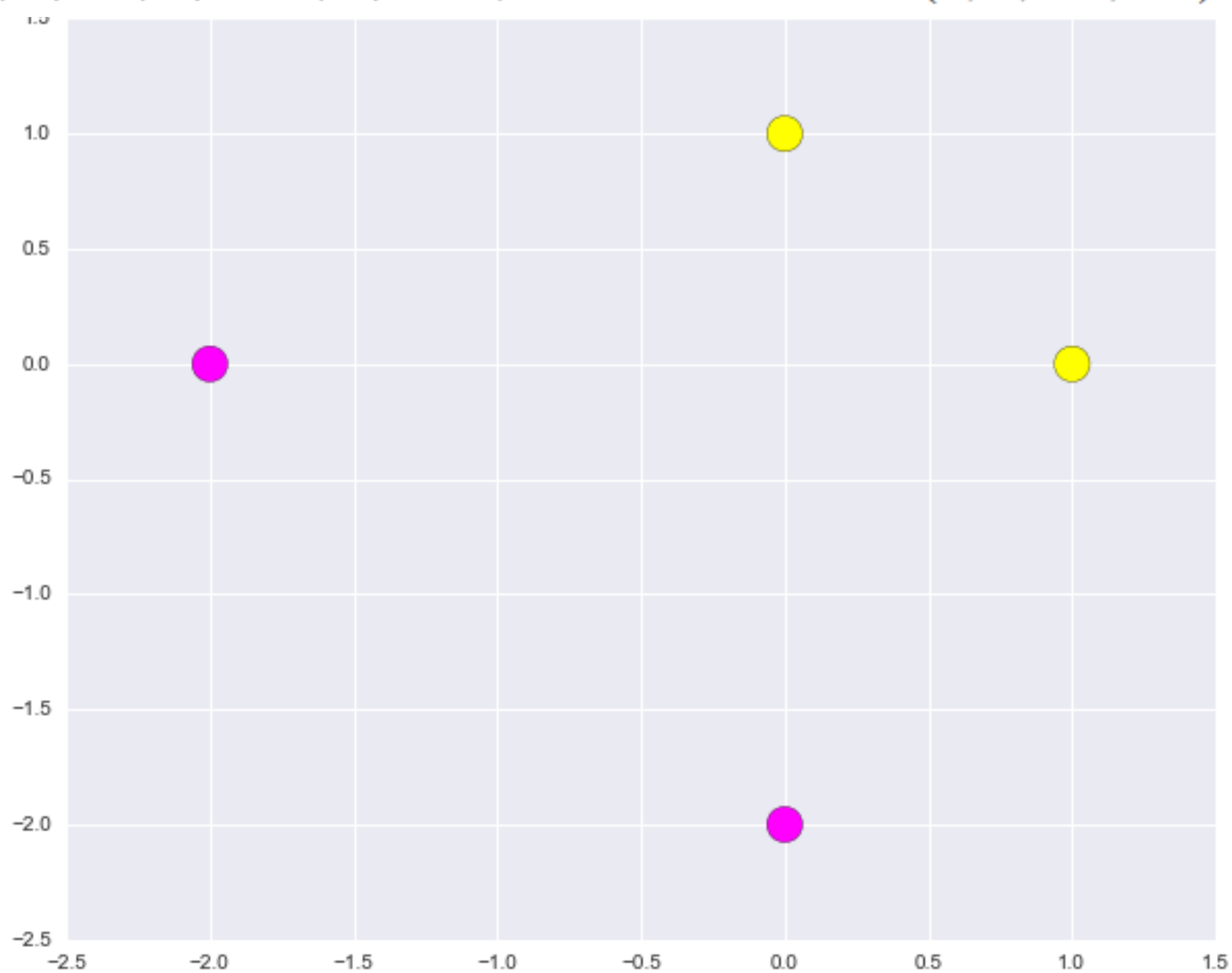
Support Vectors



https://en.wikipedia.org/wiki/Support_vector_machine

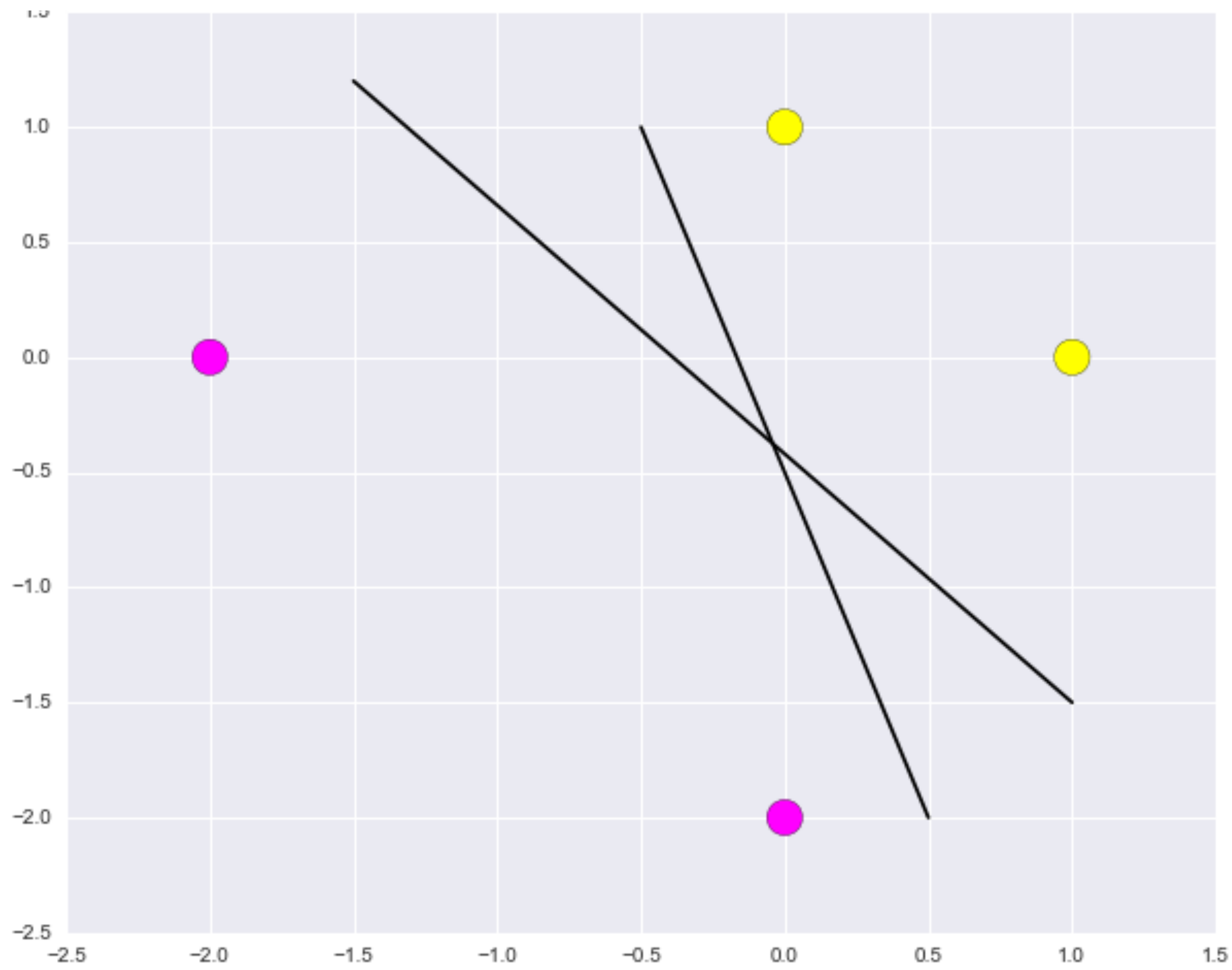
SVM example

$X = [(1,0), (0,1), (-2,0), (0,-2)]$ $Y = (1,1,-1,-1)$



SVM example

$$X = [(1,0),(0,1),(-2,0),(0,-2)] \qquad Y = (1,1,-1,-1)$$



SVM example

$$X = [(1,0),(0,1),(-2,0),(0,-2)] \qquad Y = (1,1,-1,-1)$$

$$w_1 x_1 + w_2 x_2 + b = 0$$

$$\begin{cases} \min\limits_{w,b} \|w\|^2 \\ y_i(x_i^T w + b) \geq 1 \end{cases}$$

$$\begin{cases} \min\limits_{w,b} (w_1^2 + w_2^2) \\ w_1 + b \geq 1 \quad (E1.1) \\ w_2 + b \geq 1 \quad (E1.2) \\ (-1) * (-2w_1 + b) \geq 1 \quad (E1.3) \\ (-1) * (-2w_2 + b) \geq 1 \quad (E1.4) \end{cases}$$

SVM example

$$\begin{cases} \min\limits_{w,b}(w_1^2 + w_2^2) \\ w_1 + b \geq 1 \qquad (E1.1) \\ w_2 + b \geq 1 \qquad (E1.2) \\ (-1) * (-2w_1 + b) \geq 1 \qquad (E1.3) \\ (-1) * (-2w_2 + b) \geq 1 \qquad (E1.4) \end{cases}$$

(E1.1)+(E1.3)$\Rightarrow 3w_1 \geq 2$

(E1.2)+(E1.4)$\Rightarrow 3w_2 \geq 2$

(E1.1),(E1.3)$\Rightarrow 2w_1 - 1 \geq b \geq 1 - w_1$

(E1.2),(E1.3)$\Rightarrow 2w_2 - 1 \geq b \geq 1 - w_2$

$$w_1 = w_2 = 2/3 \qquad b = 1/3$$

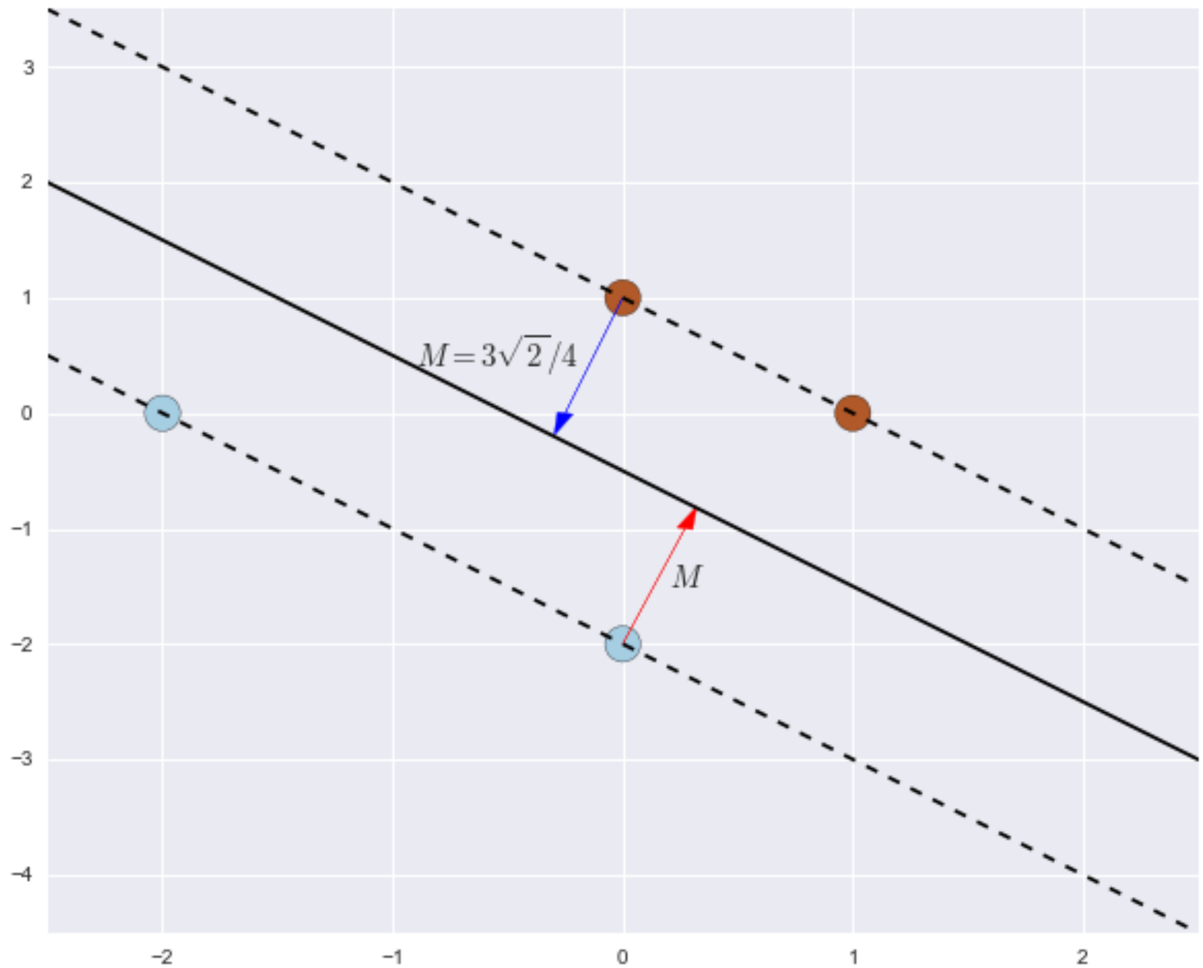
$$2x_1 + 2x_2 + 1 = 0$$

SVM example

$$2x_1/3 + 2x_2/3 + 1/3 = 0$$

$$2x_1 + 2x_2 + 1 = 0$$

$$M = \frac{1}{\|w\|} = \frac{1}{\sqrt{8/9}} = \frac{3\sqrt{2}}{4}$$



Lagrange duality

$$w^T w \to \min,$$
$$\forall j, y_j(x_j^T w + b) \geq 1$$

Lagrangian: $$L(w, b, \alpha) = w^T w - \sum_j \alpha_j \left(y_j(x_j^T w + b) - 1 \right)$$

primal: $$\alpha^* = \alpha^*(w, b) = \mathrm{argmax}_{\alpha, \alpha_j \geq 0} L(w, b, \alpha) \qquad \alpha_j \geq 0$$

$$(w^*, b^*) = \mathrm{argmin}_{w,b} L(w, b, \alpha^*(w, b))$$

CUSP: Machine Learning, Spring 2016. Session

Lagrange duality

dual problem

$$(w^*, b^*) = \mathrm{argmax}_{w,b} L(w, b, \alpha)$$

$$\alpha^* = \mathrm{argmin}_{\alpha, \alpha_i \geq 0} L(w^*(\alpha), b^*(\alpha), \alpha)$$

Solution of the dual problem

$$L(w,b,\alpha) = w^T w - \sum_j \alpha_j \left(y_j (x_j^T w + b) - 1 \right) \qquad \frac{\partial L}{\partial w} = 0 \qquad \frac{\partial L}{\partial b} = 0$$

$$w = \frac{1}{2} \sum_j \alpha_j y_j x_j$$

$$0 = \sum_j \alpha_j y_j$$

$$\alpha^* = \operatorname{argmin}_{\alpha, \alpha_j \geq 0, \sum_j \alpha_j y_j = 0} \left[\frac{1}{8} \sum_{j,k} \alpha_j \alpha_k y_j y_k x_j^T x_k + \sum_j \alpha_j \right]$$

$$b^* = -\frac{\max_{j, y_j = -1} (w^*)^T x_j + \min_{j, y_j = 1} (w^*)^T x_j}{2}$$

Optimization - SMO

$$\alpha^* = \operatorname{argmin}_{\alpha, \alpha_j \geq 0, \sum_j \alpha_j y_j = 0} \left[\frac{1}{8} \sum_{j,k} \alpha_j \alpha_k y_j y_k x_j^T x_k + \sum_j \alpha_j \right]$$

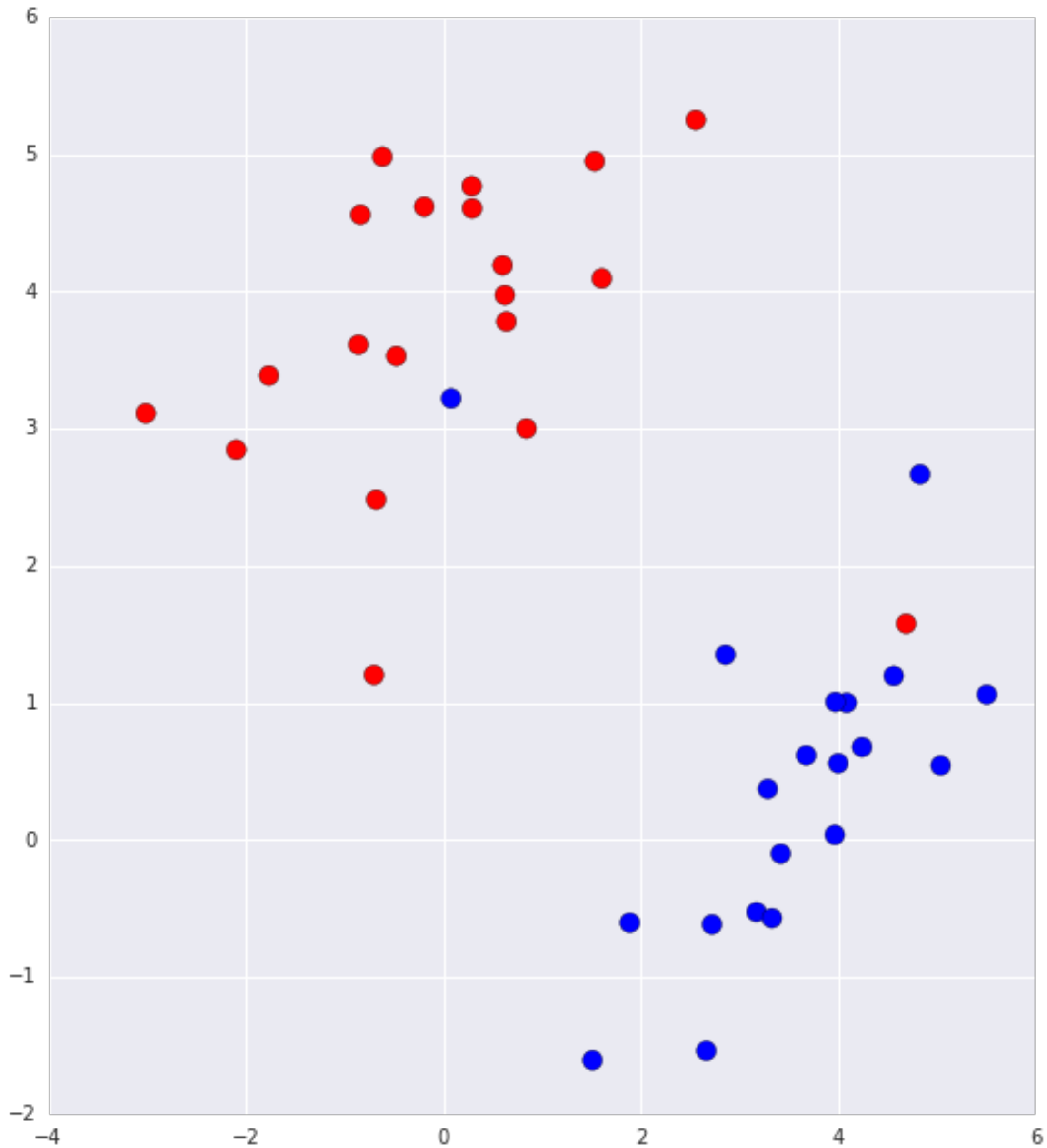
Repeat until convergence:

1. Select a pair of j,k
2. Optimize the expression above by adjusting α_j α_k with respect to constrains $\alpha_j \geq 0, \sum_j \alpha_j y_j = 0$

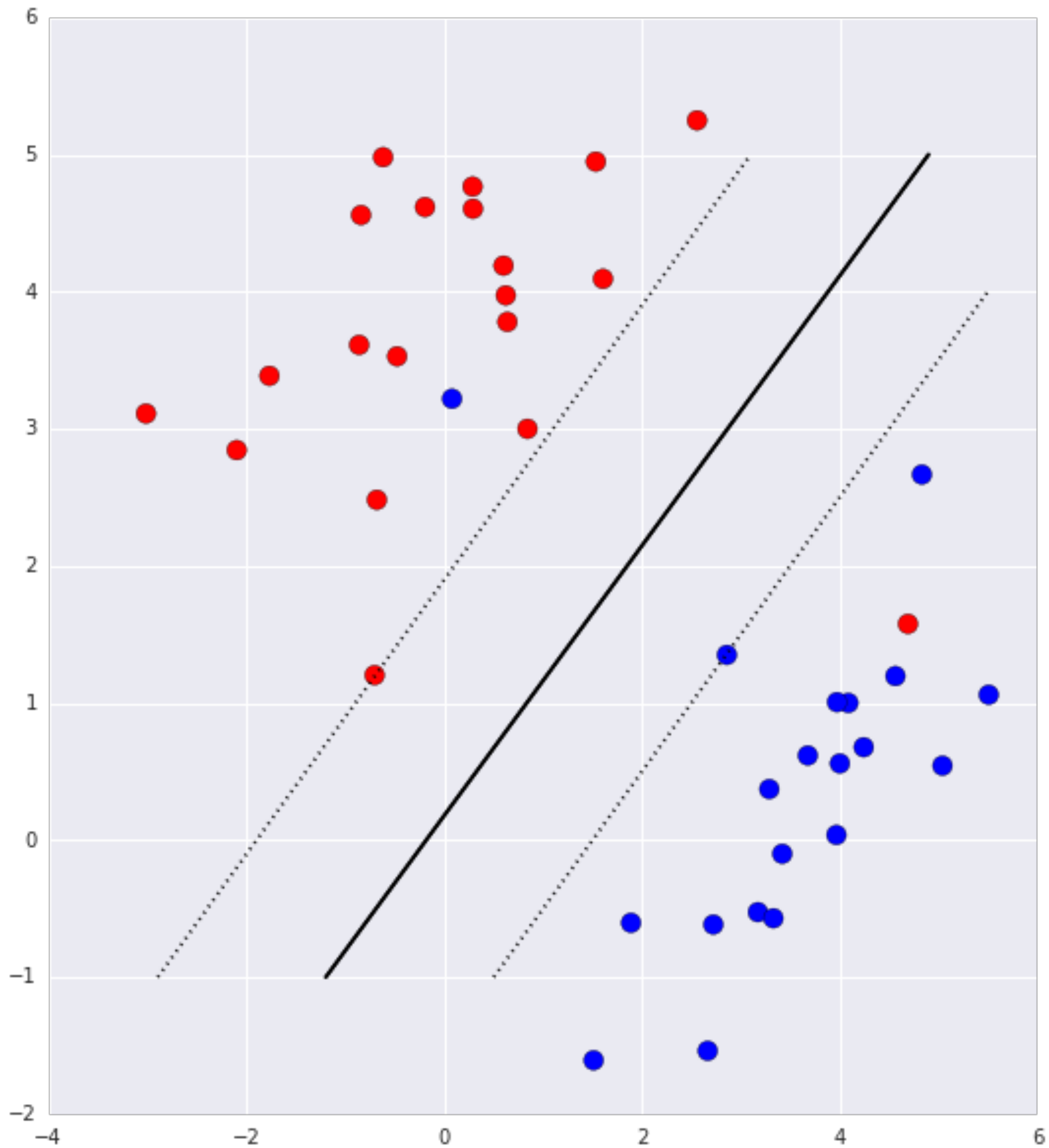
Example 1

Ipython notebook NYU classes - resources - session7
download the NBsession7.ipynb,
download and unzip data.zip in the same folder

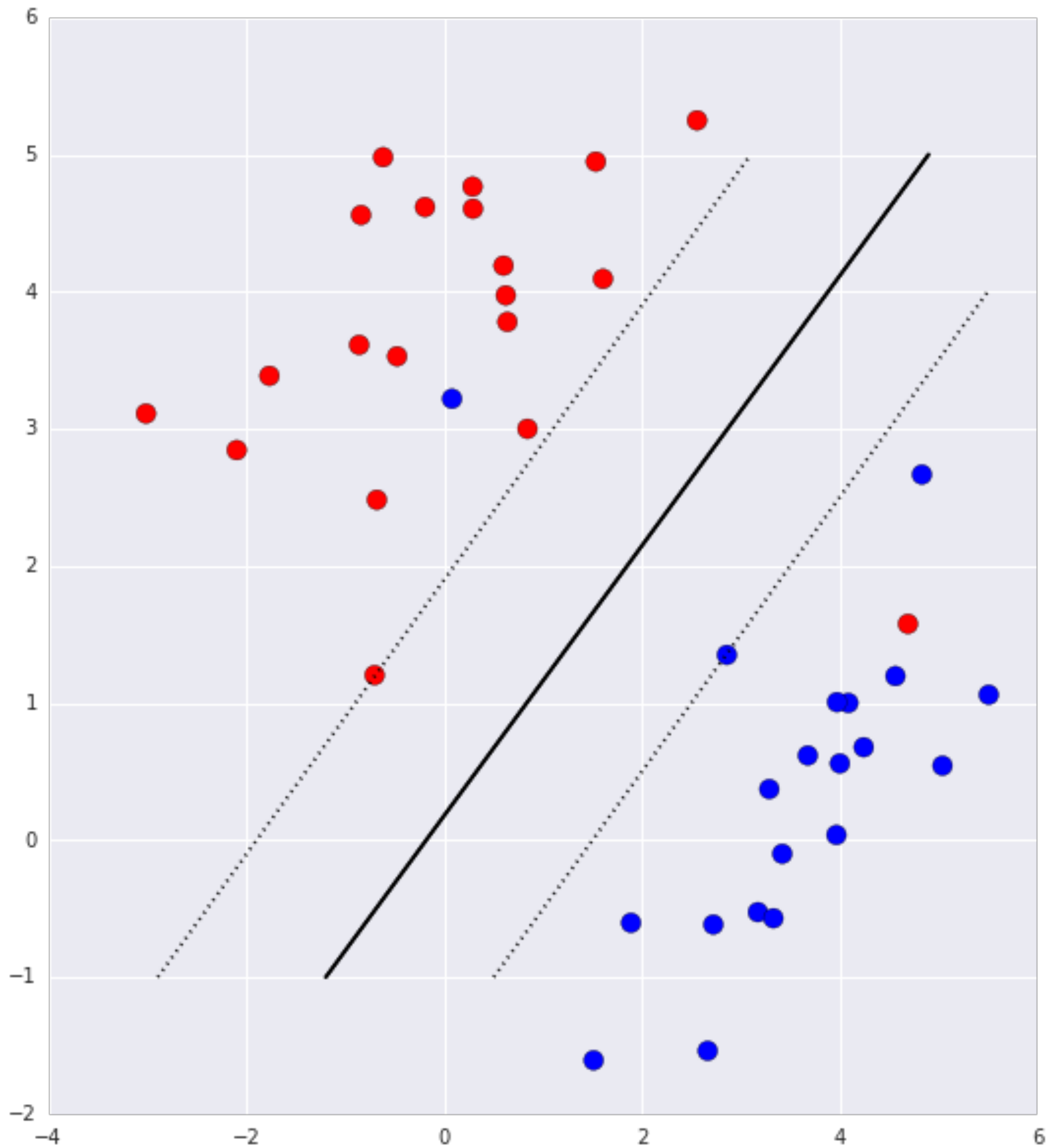
Non-separable case: soft margins



Non-separable case: soft margins

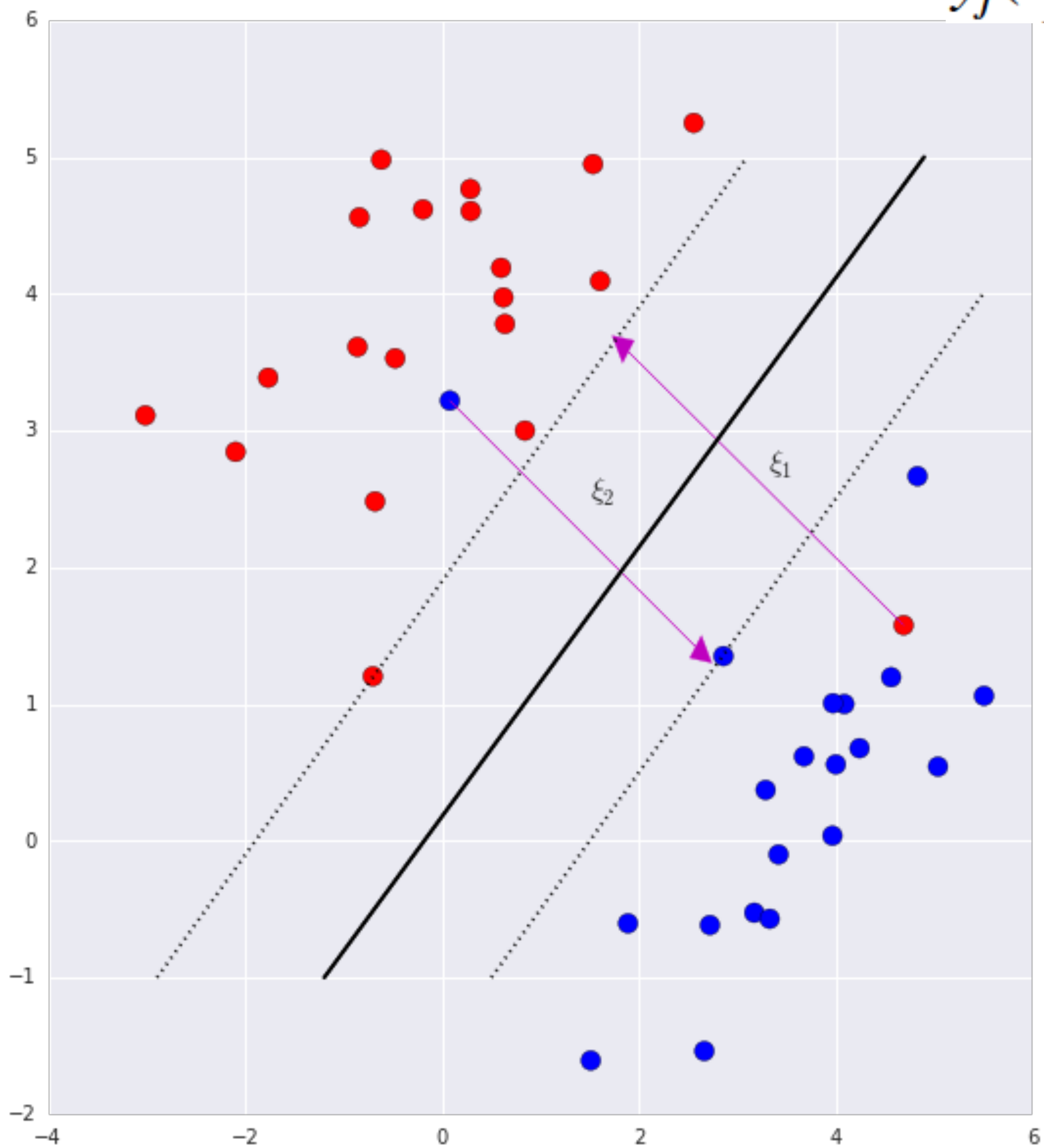


Non-separable case: soft margins

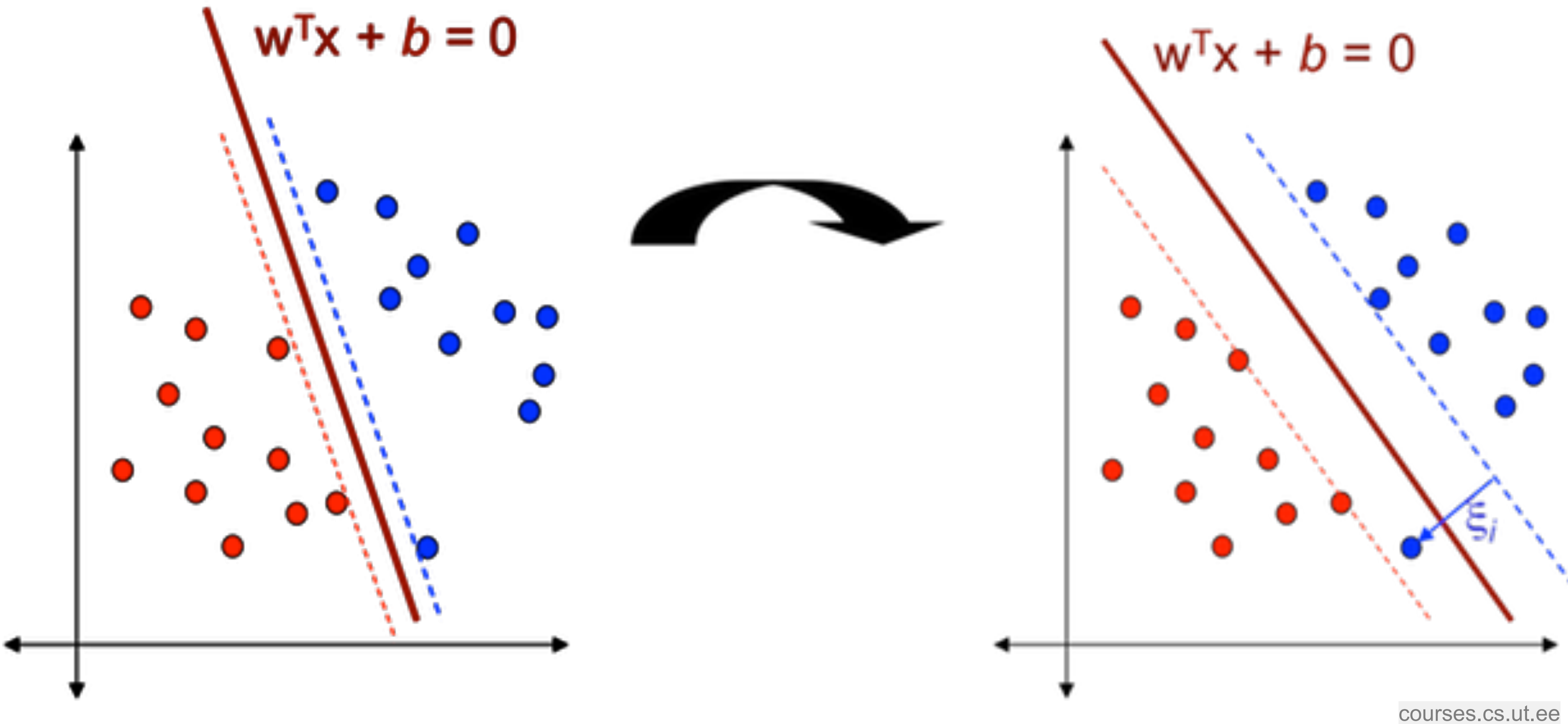


Non-separable case: soft margins

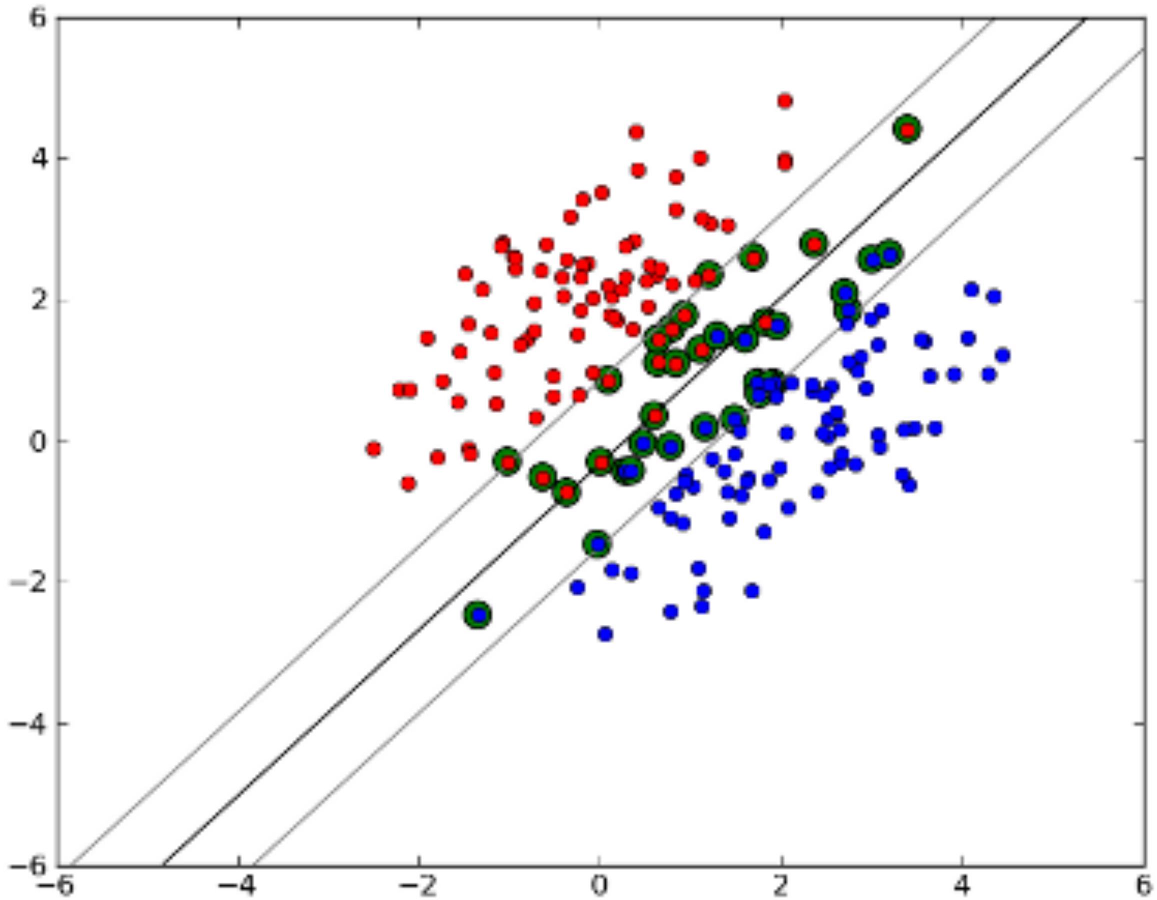
$$y_j(x_j^T w + b) \geq 1 - \xi_j$$



Non-separable case: soft margins



Support Vector Machines



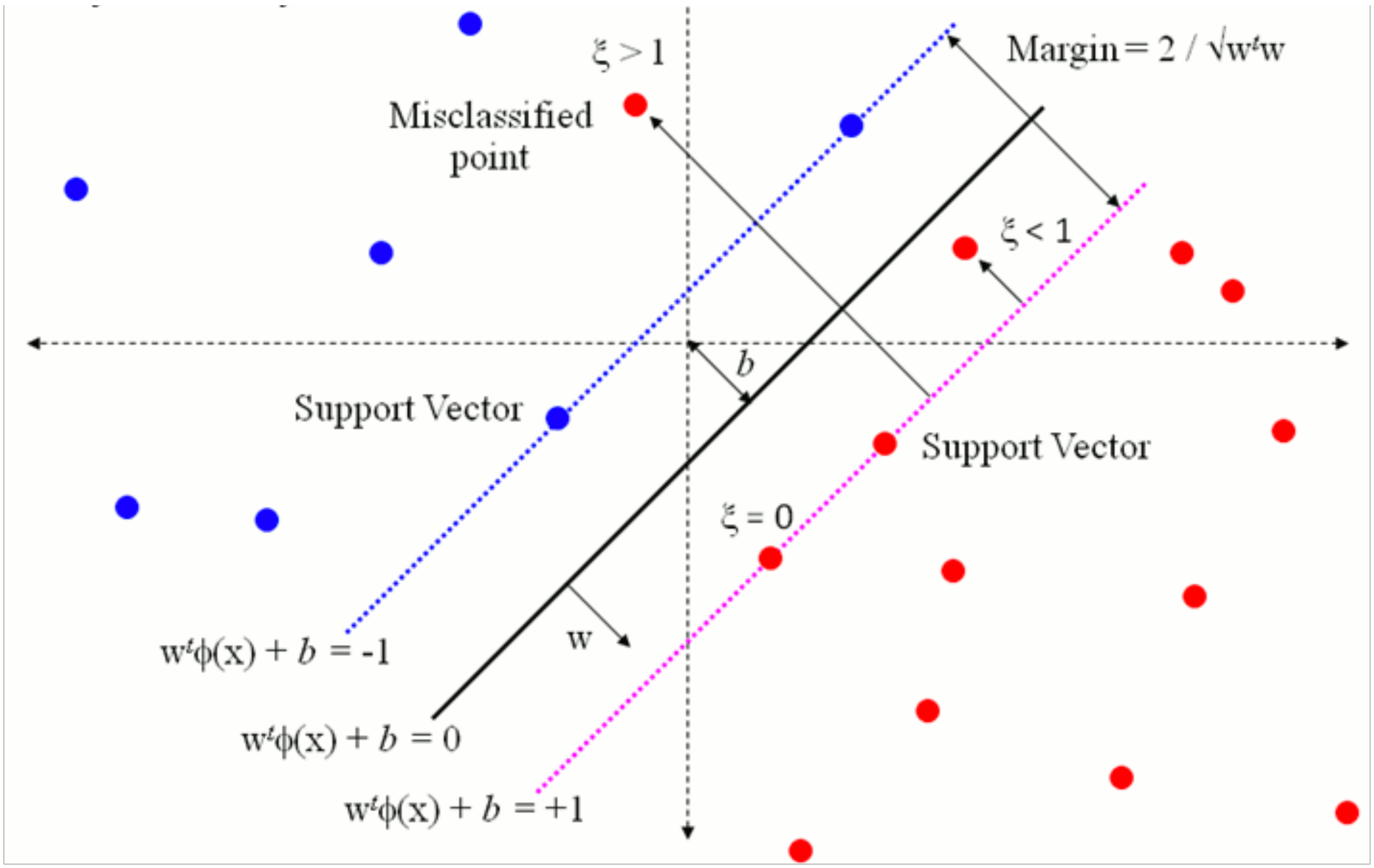
http://www.mblondel.org/journal/2010/09/19/support-vector-machines-in-python/

Non-separable case: soft margins

$$w^T w + \lambda \sum_j \xi_j \to \min,$$

$$\forall j, \xi_j \geq 0,\ y_j(x_j^T w + b) \geq 1 - \xi_j.$$

Non-linearly separable case:



CUSP: Machine Learning, Spring 2016, Session 7

Solving dual problem

$$w = \frac{1}{2}\sum_j \alpha_j y_j x_j$$

$$0 = \sum_j \alpha_j y_j$$

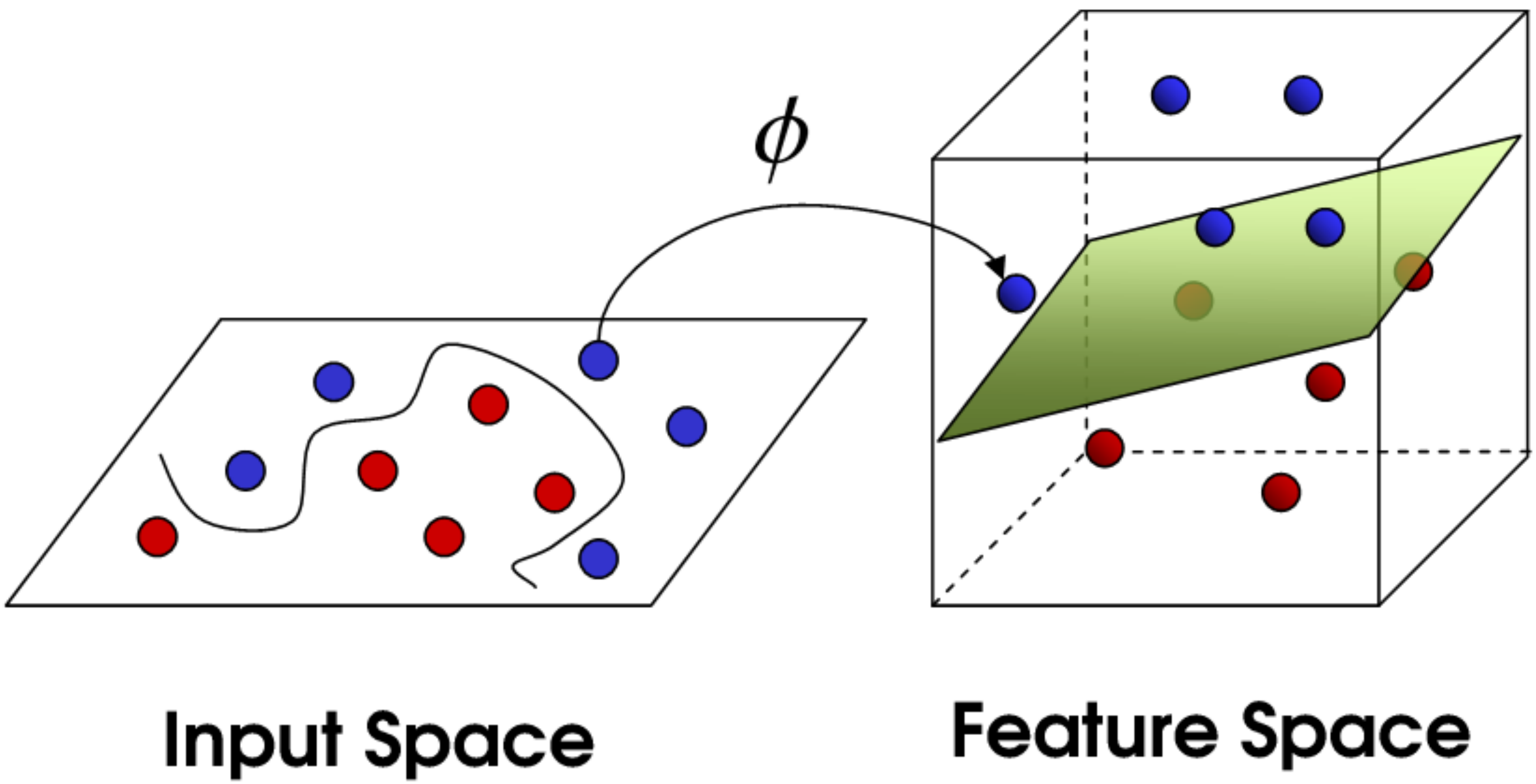
$$\alpha^* = \operatorname{argmin}_{\alpha, \alpha_j \geq 0, \sum_j \alpha_j y_j = 0} \left[\frac{1}{8} \sum_{j,k} \alpha_j \alpha_k y_j y_k x_j^T x_k + \sum_j \alpha_j \right]$$

$$0 \leq \alpha_j \leq C$$

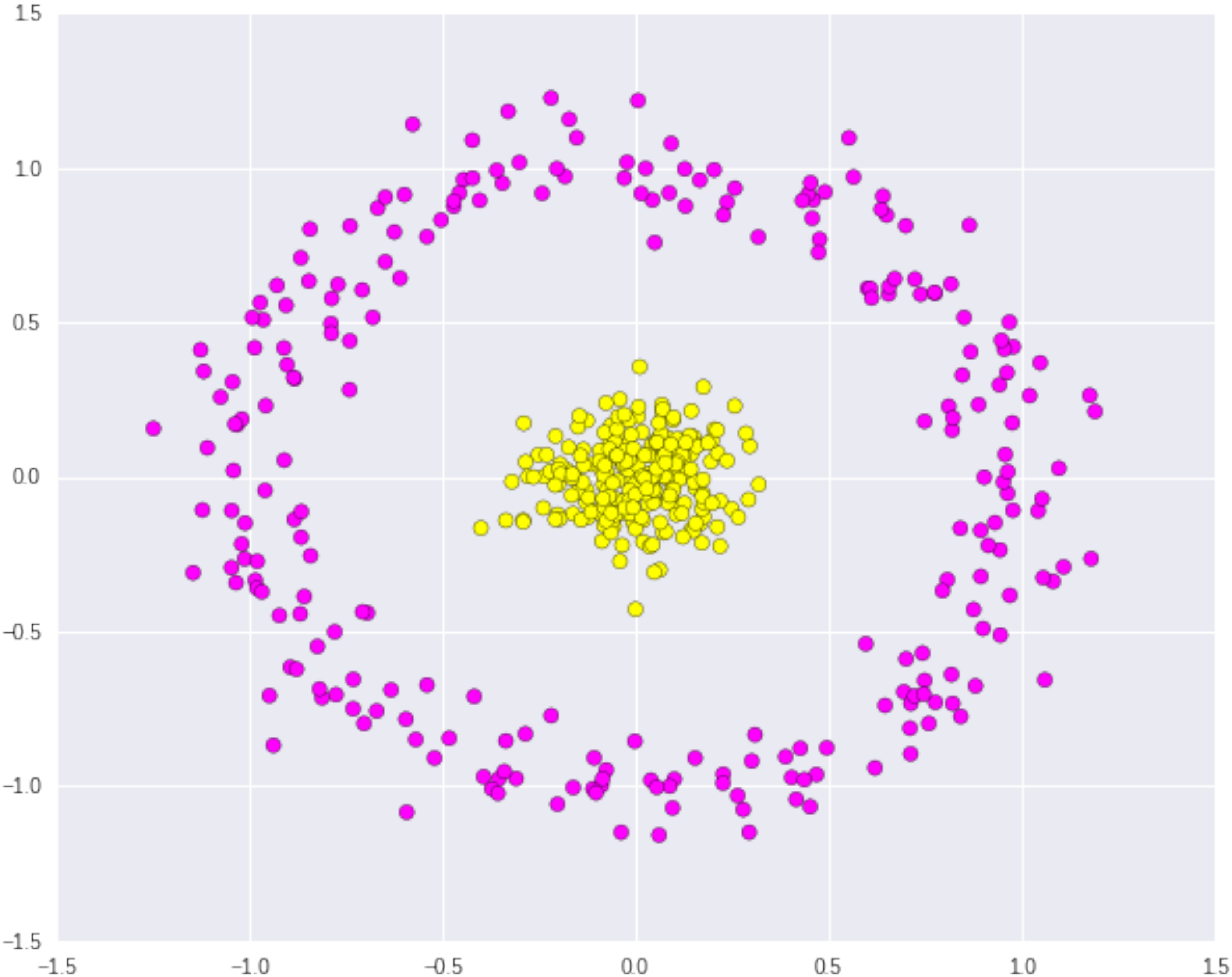
Example 2

Ipython notebook NYU classes - resources - session7
download the NBsession7.ipynb,
download and unzip data.zip in the same folder

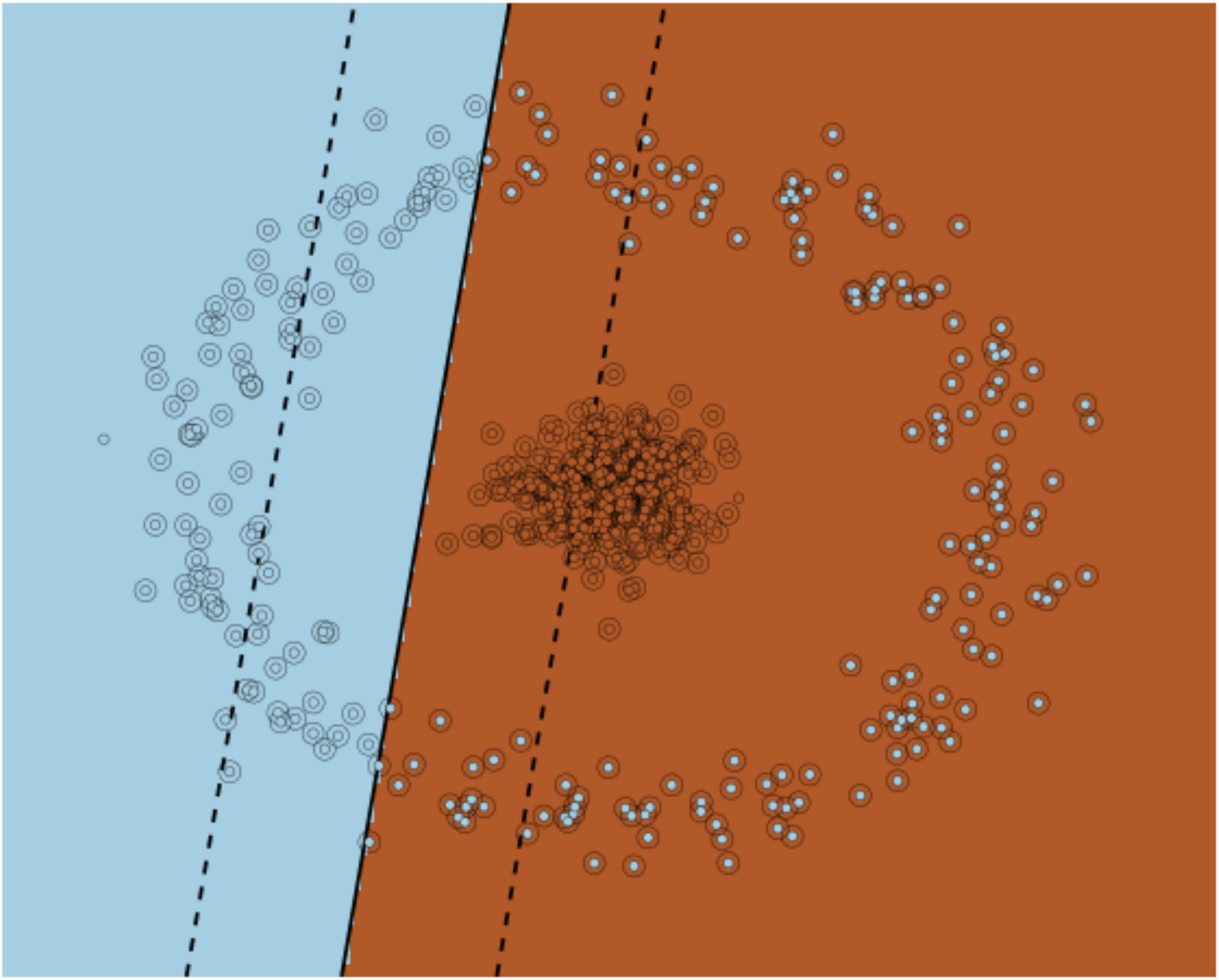
Non-linearly separable case: kernels



Non-linearly separable case: kernels

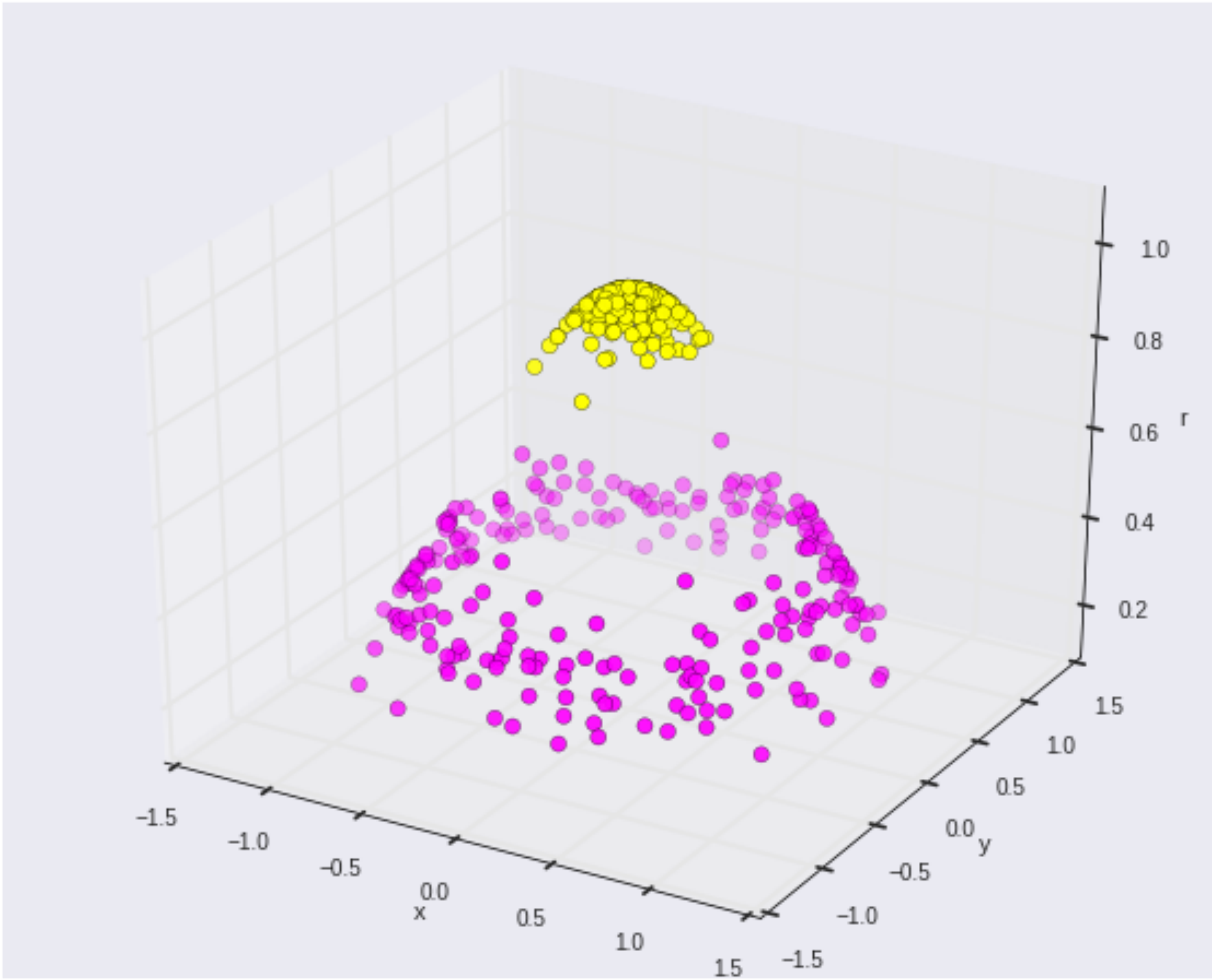


Non-linearly separable case: kernels

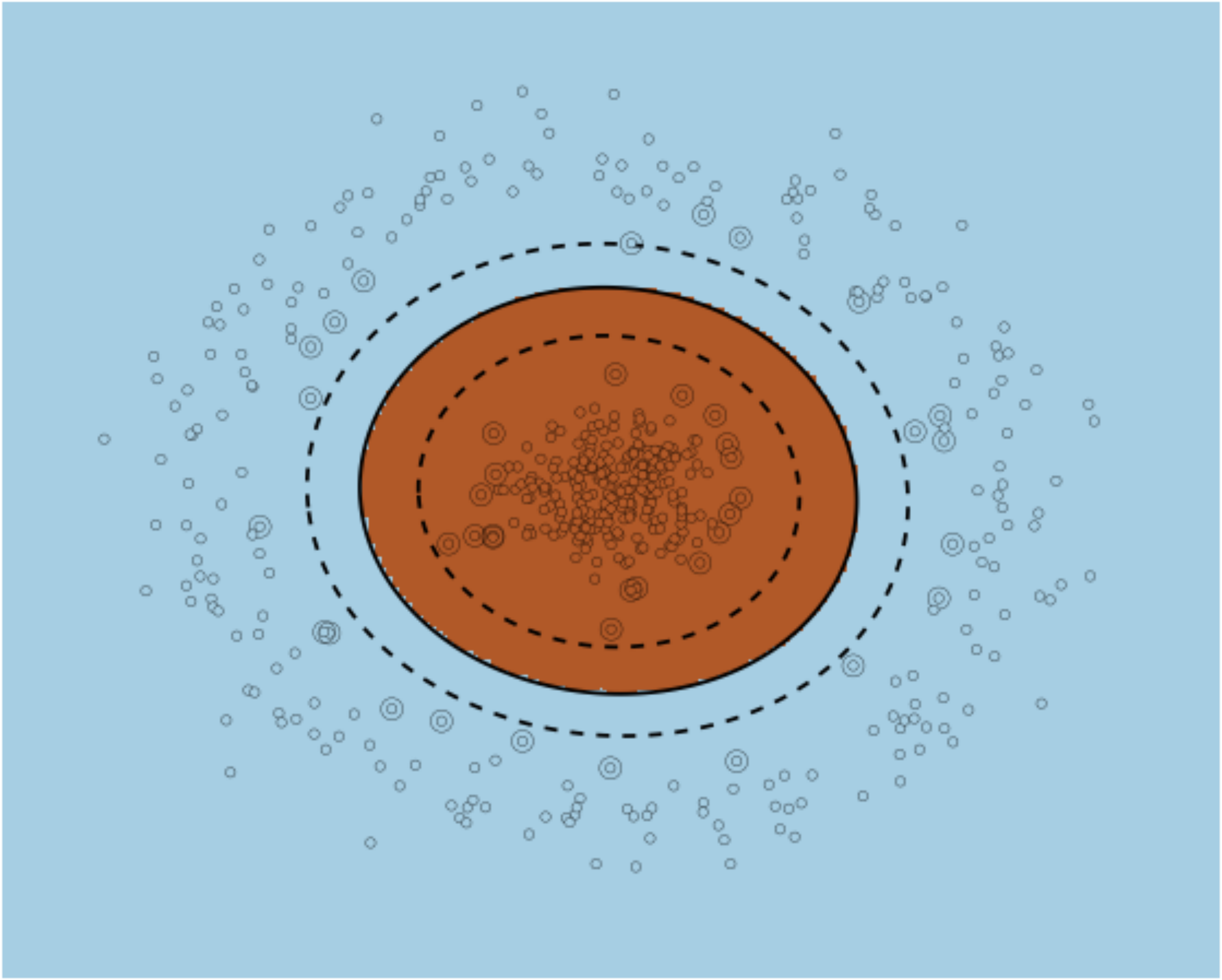


Non-linearly separable case: kernels

$r = \sqrt{(x_1)^2 + (x_2)^2} \quad r := e^{-r^2} = e^{-(x_1)^2-(x_2)^2} \quad \phi : (x, y) \rightarrow (x, y, r)$



Non-linearly separable case: kernels



Kernel trick

$$\alpha^* = \mathrm{argmin}_{\alpha,\alpha_j\geq 0,\sum\limits_j \alpha_j y_j=0}\left[\frac{1}{8}\sum_{j,k}\alpha_j\alpha_k y_j y_k \phi(x_j)^T\phi(x_k)+\sum_j \alpha_j\right]$$

Instead of $\phi(x_j)$

We only need $\phi(x_j)^T\phi(x_k)$

$K(x,z)=\phi(x)^T\phi(z)$ -kernel function

Kernel trick - example

$$x^1, x^2, x^3$$

$$\phi(x) = \{x^1, x^2, x^3, (x^1)^2, (x^2)^2, (x^3)^2, x^1x^2, x^2x^3, x^1x^3\}$$

$$\phi(x) = \{cx^1, cx^2, cx^3, (x^1)^2, (x^2)^2, (x^3)^2, x^1x^2, x^2x^3, x^1x^3\}$$

$$K(x, z) = c^2 \sum_p x^p z^p + \sum_{p,q} x^p z^p x^q z^q = (x^T z + c/2)^2 - c^2/4$$

$$K(x, z) = (x^T z + c/2)^2$$

Common type of kernels

Mercer theorem: For the matrix $K = (K_{j,k},\ j,k = 1..N)$

$K_{j,k} = K(x_j, x_k)$ to be a valid Kernel, i.e. $K_{j,k} = \phi(x_j)^T \phi(x_k)$ it is necessary and sufficient that K is symmetric and positive semi-definite, i.e. for any vector z, $z^T K z > 0$

Common type of kernels

Linear	$\phi : x \to x$	$K(x,z) = x^T z$
Polynomial	$K(x,z) = (x^T z + c)^d$	
Gaussian	$K(x,z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}}$	

Example 3

Ipython notebook NYU classes - resources - session4
download the NBsession7.ipynb,
download and unzip data.zip in the same folder

Generalization - multi-class SVM

Classify with $y = 1, 2, 3, ..., S$

- One vs all
- One vs one

Generalization - support vector regression

$$y \sim x^T w + b$$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|^2 \\ \text{subject to} & \left\{ \begin{array}{lll} y_i - \langle w, x_i \rangle - b & \leq & \varepsilon \\ \langle w, x_i \rangle + b - y_i & \leq & \varepsilon \end{array} \right. \end{array}$$

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \xi_i^*) \\ \text{subject to} & \left\{ \begin{array}{lll} y_i - \langle w, x_i \rangle - b & \leq & \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i & \leq & \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* & \geq & 0 \end{array} \right. \end{array}$$

http://alex.smola.org/papers/2003/SmoSch03b.pdf