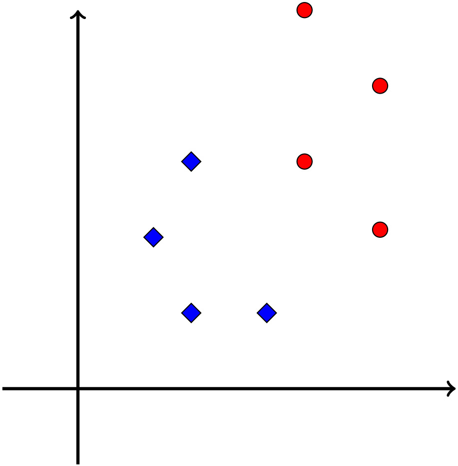
Support Vector Machines (SVM) and Sequential Minimal Optimization (SMO)

Felix Brockherde

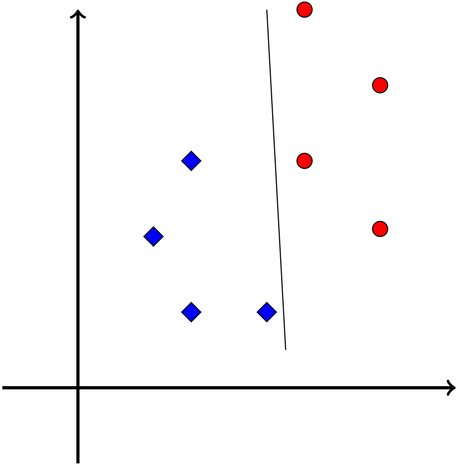
Technische Universität Berlin and Max Planck Institute of Microstructure Physics

25 Jun 2014

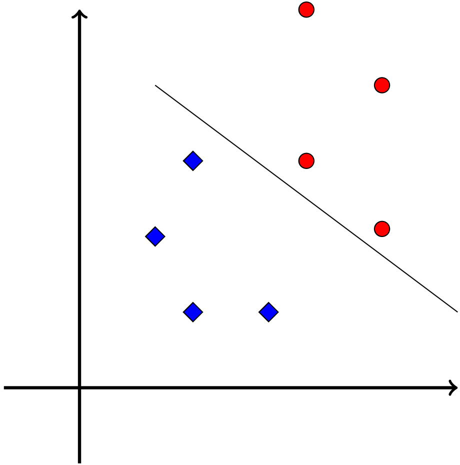
Support Vector Machines (SVM)



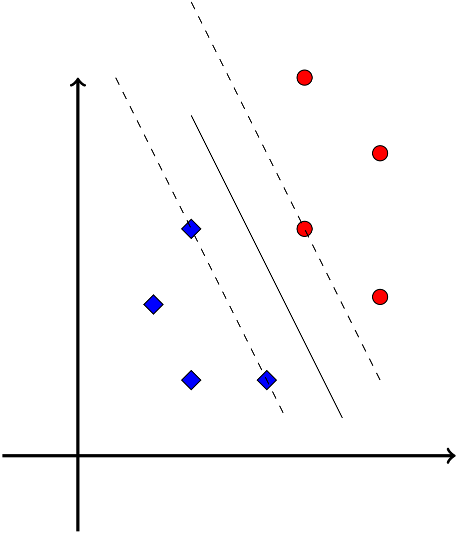
Support Vector Machines (SVM)

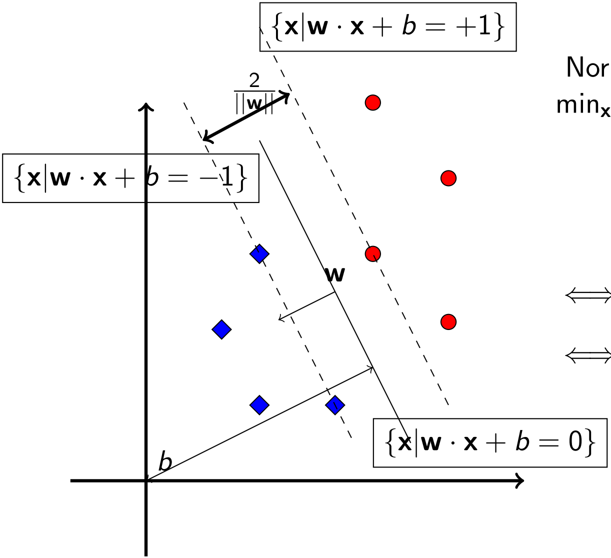


Support Vector Machines (SVM)



Support Vector Machines (SVM)

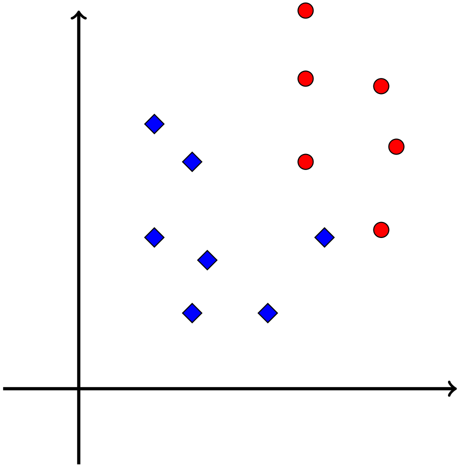




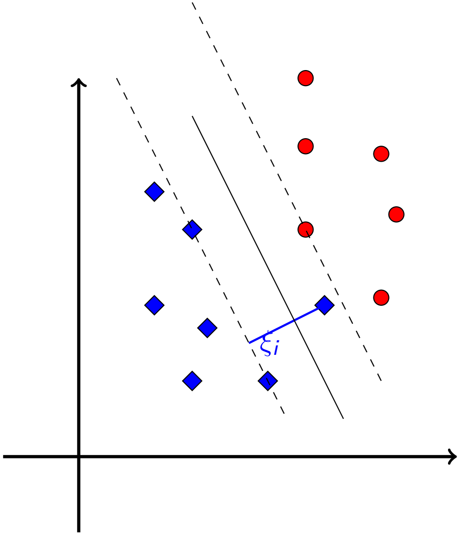
Normalize $\mathbf{w}$ so that
$\min_{\mathbf{x}_i} \mathbf{w} \cdot \mathbf{x}_i + b = 1.$

$$
\begin{aligned}
& \mathbf{w} \cdot \mathbf{x}_1 + b = +1 \\
& \mathbf{w} \cdot \mathbf{x}_2 + b = -1 \\
\iff & \mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2 \\
\iff & \frac{\mathbf{w}}{||\mathbf{w}||} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = \frac{2}{||\mathbf{w}||}
\end{aligned}
$$

Slack variables



Slack variables



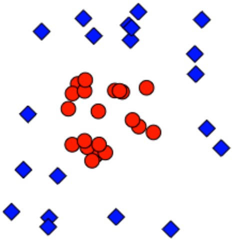
Introduce slack variables ξ_i:

$$\min_{\mathbf{w},b,\xi_i} \quad ||\mathbf{w}||^2 + C\sum_{i=1}^{N} \xi_i$$

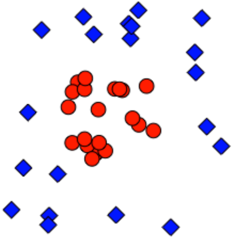
$$\text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x_i} + b) \geq 1 - \xi_i$$

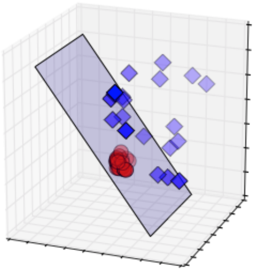
$$\xi_i \geq 0$$

Non-linear hyperplanes



Non-linear hyperplanes





Map into a higher dimensional feature space:

$$\begin{aligned} \Phi : \quad \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (x_1, x_2) &\mapsto (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{aligned}$$

Dual SVM

Primal

$$\min_{\mathbf{w},b,\xi_i} \quad ||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w}\cdot\Phi(\mathbf{x_i})+b) \geq 1-\xi_i \;\text{ and }\; \xi_i \geq 0 \quad \text{for } i = 1\ldots N$$

Dual

$$\max_{\alpha} \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\Phi(\mathbf{x_i})\cdot\Phi(\mathbf{x_j})\right)$$

$$\text{subject to} \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \;\text{ and }\; C \geq \alpha_i \geq 0 \quad \text{for } i = 1\ldots N$$

Data points x_i only appear in scalar products $(\Phi(\mathbf{x_i})\cdot\Phi(\mathbf{x_j}))$.

The Kernel Trick

Replace scalar products with kernel function (**?**):

$$k(x, y) = \Phi(x) \cdot \Phi(y)$$

- Compute kernel matrix $K_{ij} = k(x_i, x_j)$, i.e. never use Φ directly
- Underlying mapping Φ can be unknown
- Kernels can be adopted to specific task, e.g. using prior knowledge (kernels for graphs, trees, strings, ...)

Common kernels

Gaussian Kernel: $k(x, y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right)$

Linear Kernel: $k(x, y) = x \cdot y$

Polynomial Kernel: $k(x, y) = (x \cdot y + c)^d$

The Support Vectors in SVM

$$\max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \left(\Phi(\mathbf{x_i}) \cdot \Phi(\mathbf{x_j}) \right)$$

$$\text{subject to} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \;\text{ and }\; C \geq \alpha_i \geq 0 \quad \text{for } i = 1 \dots N$$

KKT conditions

$$y_i \left[\mathbf{w}\Phi(\mathbf{x_i})) + b\right] > 1 \Longrightarrow \qquad a_i = 0 \longrightarrow \qquad x_i \text{ irrelevant}$$

$$y_i \left[\mathbf{w}\Phi(\mathbf{x_i})) + b\right] = 1 \Longrightarrow \qquad \text{on/in margin} \longrightarrow \qquad x_i \text{ Support Vector}$$

Old model $f(x) = w \cdot \Phi(x_i) + b$ becomes via $w = \sum_{i=1}^{N} \alpha_i y_i \Phi(x_i)$:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b \quad \longrightarrow \quad f(x) = \sum_{x_i \in SV} \alpha_i y_i k(x_i, x) + b$$

Quadratic Programming (QP)

Reminder: The SVM optimization problem

$$\max_{\alpha} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\text{subject to} \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \ \text{ and } \ C \geq \alpha_i \geq 0 \quad \text{for } i = 1 \ldots N$$

Quadratic Programming

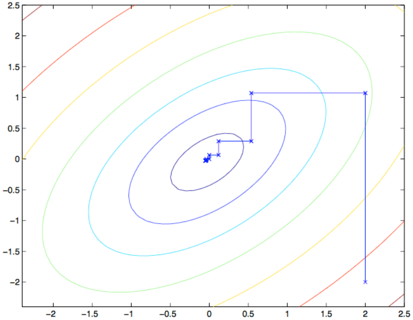
$$\begin{aligned} \min_{x} \quad & \frac{1}{2} x^{\intercal} P x + q^{\intercal} x \\ \text{s.t.} \quad & Gx \preceq h \\ & Ax = b \end{aligned}$$

$\Longrightarrow$ We can solve the SVM problem with a QP solver.

Coordinate Descent

Imagine a multivariate function $W(x_1, x_2, \ldots, x_N)$ and the problem

$$\operatorname{argmax}_{\mathbf{x}} W(x_1, x_2, \ldots, x_N)$$



Coordinate Descent Algorithm

while not converged **do**

  for $i = 1 \ldots N$ **do**

    $x_i \leftarrow \operatorname{argmax}_{x_i} W(x_1, \ldots, x_i, \ldots, x_N)$

  end for

end while

Sequential Minimal Optimization (SMO) Idea

We have to solve the SVM optimization problem subject to the condition

$$\sum_{i=1}^{N} \alpha_i y_i = 0.$$

I.e. we can *not* optimize one single variable individually.

Solution: Optimize two variables α_i and α_j with $y_i \neq y_j$ while keeping the other α fixed.

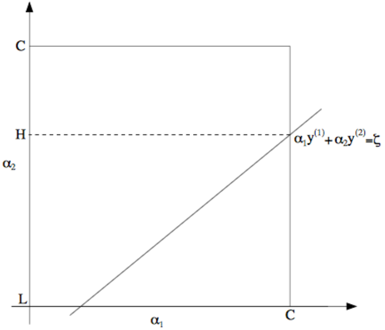
From KKT conditions, we get (optimizing α_1 and α_2 WLOG)

$$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i$$

where the right handside is fixed $\zeta = -\sum_{i=3}^{N} \alpha_i y_i$.

Box constraints

- From the second contraint $0 \leq \alpha_i \leq C$, we know that α_1 and α_2 have to lie inside the pictured box:
- We can see, that there exists a upper and lower bound for α_2: $L \leq \alpha_2 \leq H$.



- If we write $\alpha_1 = (\zeta - \alpha_2 y_2) y_1$, we have to solve a quadratic function in one variable (α_2).
- The new α_2 then has to be clipped to the box constraints L and H (!)
- Find α_1 via $\alpha_1 = (\zeta - \alpha_2 y_2) y_1$. Then pick two new α and repeat.