Kernel trick

Victor Kitov

v.v.kitov@yandex.ru

Mercer kernel definition

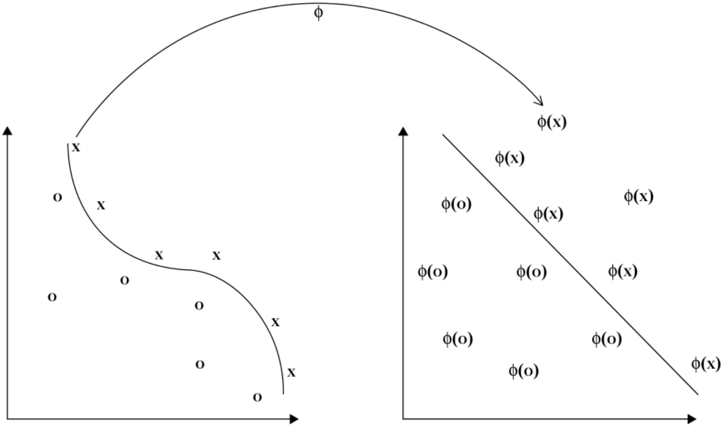
- x is replaced with $\phi(x)$
  - Example: $[x] \to [x,\, x^2,\, x^3]$

Mercer Kernel

Function $K(x, x') : X \times X \to \mathbb{R}$ is a Mercer kernel function if it may be represented as $K(x, x') = \langle \phi(x), \phi(x') \rangle$ for some mapping $\phi : X \to H$, with scalar product defined on H.

- Mercer kernels will be called kernels for short here.
- $\langle x, x' \rangle$ is replaced by $\langle \phi(x), \phi(x') \rangle = K(x, x')$

Illustration



Polynomial kernel

- Example 1: let $D=2$.

$$\begin{aligned} K(x,z) &= (x^T z)^2 = (x_1 z_1 + x_2 z_2)^2 = \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 x_2 z_2 \\ &= \phi^T(x)\phi(z) \end{aligned}$$

for $\phi(x) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2)$

- Example 2: let $D=2$.

$$\begin{aligned} K(x,z) &= (1 + x^T z)^2 = (1 + x_1 z_1 + x_2 z_2)^2 = \\ &= 1 + x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 z_1 + 2x_2 z_2 + 2x_1 z_1 x_2 z_2 \\ &= \phi^T(x)\phi(z) \end{aligned}$$

for $\phi(x) = (1,\ x_1^2,\ x_2^2,\ \sqrt{2}x_1,\ \sqrt{2}x_2,\ \sqrt{2}x_1 x_2)$

- In general for $D \geq 1$ $(x^T z)^M$ yields all polynomials of degree M and $(1 + x^T z)^M$ yields all polynomials of degree less or equal to M.

Kernel properties

Theorem (Mercer): Function $K(x,x')$ is a kernel is and only if

- it is symmetric: $K(x,x')=K(x',x)$
- it is non-negative definite:
    - definition 1: for every function $g: X\to\mathbb{R}$

$$\int_X\int_X K(x,x')g(x)g(x')dxdx'\ge 0$$

    - definition 2 (equivalent): for every finite set $x_1,x_2,...x_M$ Gramm matrix $\{K(x_i,x_j)\}_{i,j=1}^M\succeq 0$ (p.s.d.)

Kernel construction

- Kernel learning - separate field of study.
- Hard to prove non-negative definitness of kernel in general.
- Kernels can be constructed from other kernels, for example from:
    - scalar product $\langle x, x' \rangle$
    - constant $K(x, x') \equiv 1$
    - $x^T A x$ for any $A \succcurlyeq 0$

Constructing kernels from other kernels

If $K_1(x,x')$, $K_2(x,x')$ are arbitrary kernels, $c>0$ is a constant, $q(\cdot)$ is a polynomial with non-negative coefficients, $h(x)$ and $\varphi(x)$ are arbitrary functions $\mathcal{X}\to\mathbb{R}$ and $\mathcal{X}\to\mathbb{R}^M$ respectively, then these are valid kernels:

1. $K(x,x')=cK_1(x,x')$
2. $K(x,x')=K_1(x,x')K_2(x,x')$
3. $K(x,x')=K_1(x,x')+K_2(x,x')$
4. $K(x,x')=K_1(\varphi(x),\varphi(x'))$
5. $K(x,x')=h(x)K_1(x,x')h(x')$
6. $K(x,x')=e^{K_1(x,x')}$

Commonly used kernels

Let x and x' be two objects.

Kernel	**Mathematical form**
linear	$\langle x, x' \rangle$
polynomial	$(\gamma \langle x, x' \rangle + r)^d$
RBF	$\exp(-\gamma \left\Vert x - x' \right\Vert^2)$

Addition

- Algorithms allowing kernelization: K-NN, K-means, K-medoids, nearest medoid, PCA, SVM, etc.
- Kernelization of distance:

Addition

- Algorithms allowing kernelization: K-NN, K-means, K-medoids, nearest medoid, PCA, SVM, etc.
- Kernelization of distance:

$$\begin{aligned}\rho(x,x') &= \langle x-x', x-x'\rangle = \langle x,x\rangle + \langle x',x'\rangle - 2\langle x,x'\rangle \\ &= K(x,x) + K(x',x') - 2K(x,x')\end{aligned}$$

Table of Contents

Linear SVM reminder

- Solution for weights:

$$w = \sum_{i \in \mathcal{SV}} \alpha_i y_i x_i$$

Discriminant function

$$g(x) = \sum_{i \in \mathcal{SV}} \alpha_i y_i \langle x_i, x \rangle + w_0$$

$$w_0 = \frac{1}{n_{\widetilde{\mathcal{SV}}}} \left(\sum_{j \in \widetilde{\mathcal{SV}}} y_j - \sum_{j \in \widetilde{\mathcal{SV}}} \sum_{i \in \mathcal{SV}} \alpha_i y_i \langle x_i, x_j \rangle \right)$$

where $SV = \{i : y_i(x_i^T w + w_0 \leq 1)\}$ are indexes of all support vectors and $\tilde{SV} = \{i : y_i(x_i^T w + w_0 = 1\}$ are boundary support vectors.

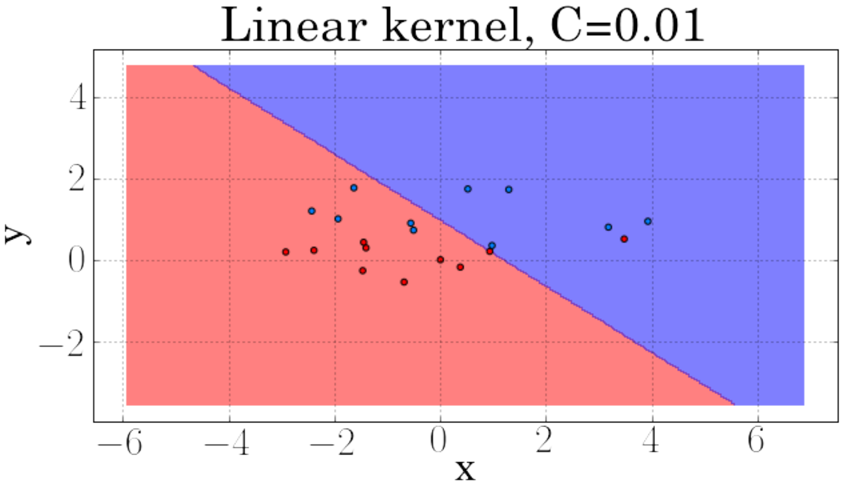
Kernel SVM

Discriminant function

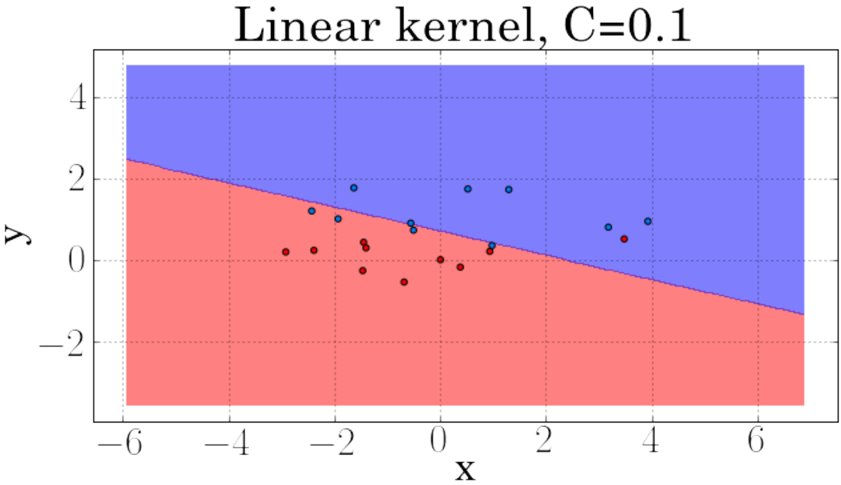
$$g(x) = \sum_{i \in \mathcal{SV}} \alpha_i y_i K(x_i, x) + w_0$$

$$w_0 = \frac{1}{n_{\widetilde{\mathcal{SV}}}} \left(\sum_{j \in \widetilde{\mathcal{SV}}} y_j - \sum_{j \in \widetilde{\mathcal{SV}}} \sum_{i \in \mathcal{SV}} \alpha_i y_i K(x_i, x_j) \right)$$

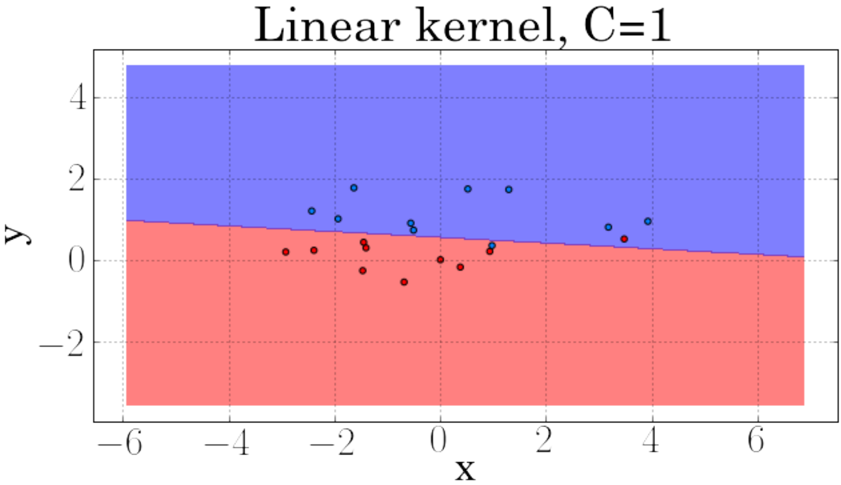
Linear kernel - variable C



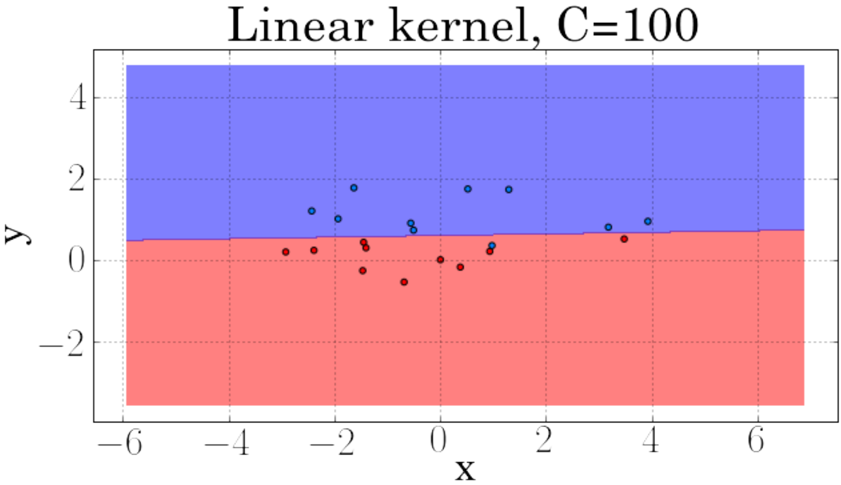
Linear kernel - variable C



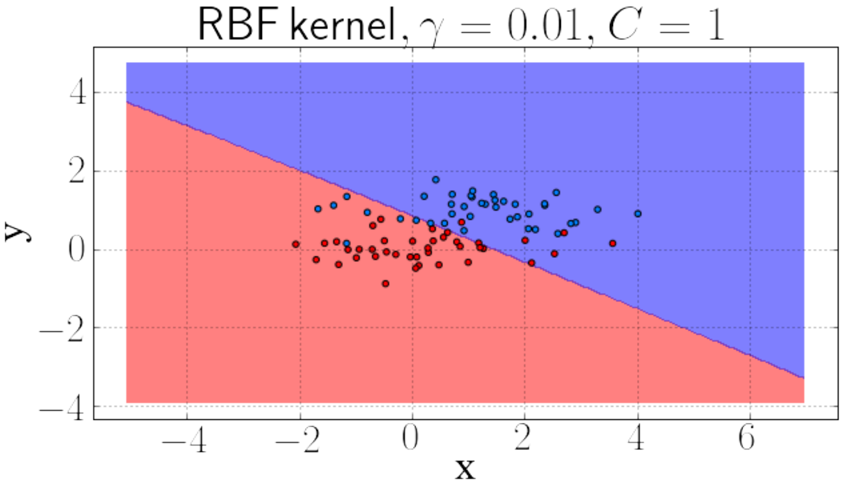
Linear kernel - variable C



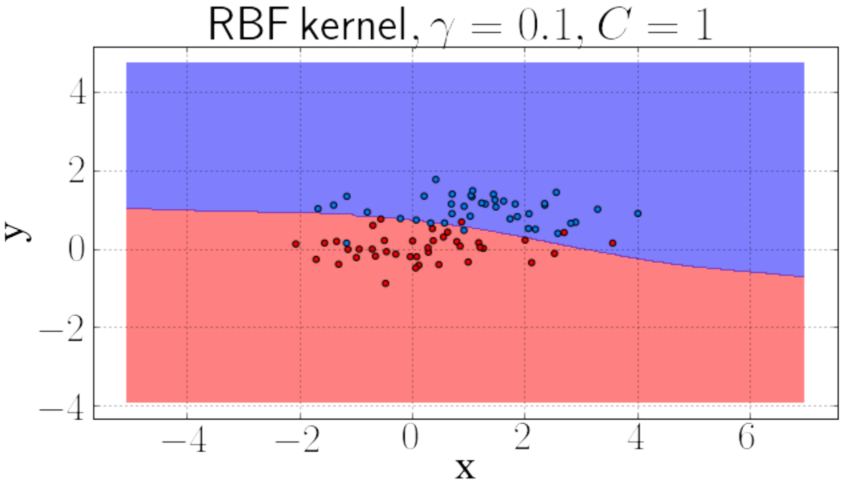
Linear kernel - variable C



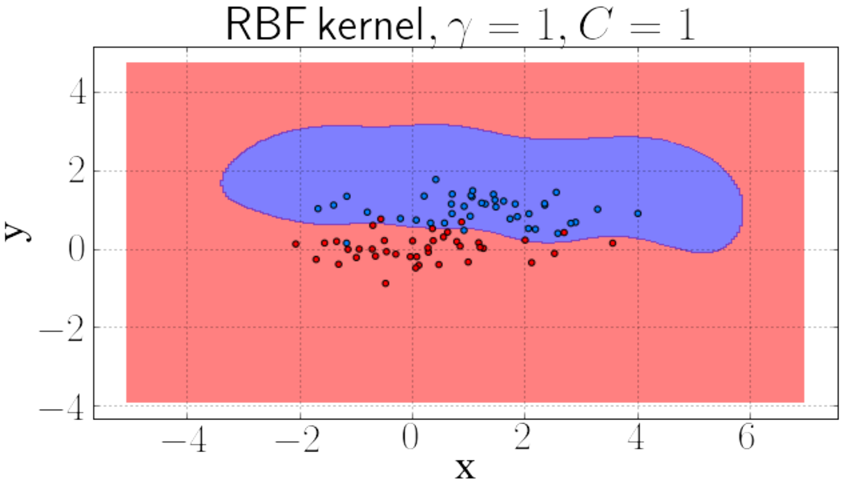
RBF kernel - variable γ



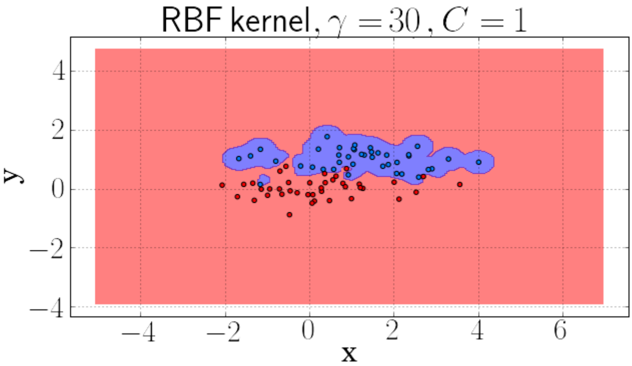
RBF kernel - variable γ



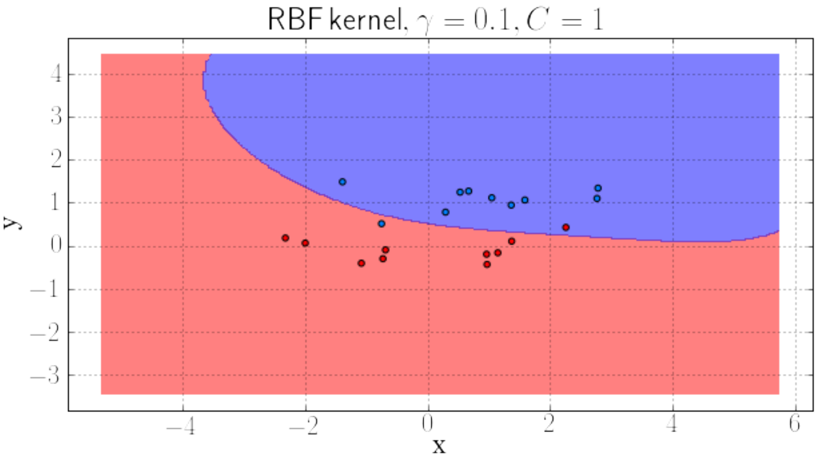
RBF kernel - variable γ



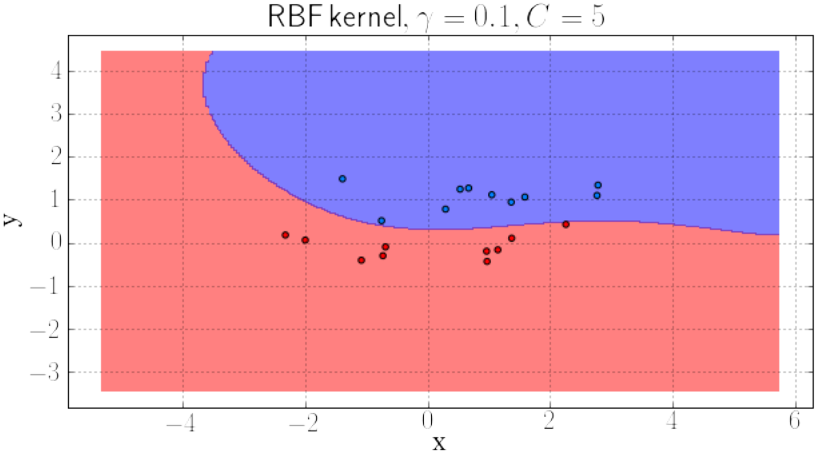
RBF kernel - variable γ



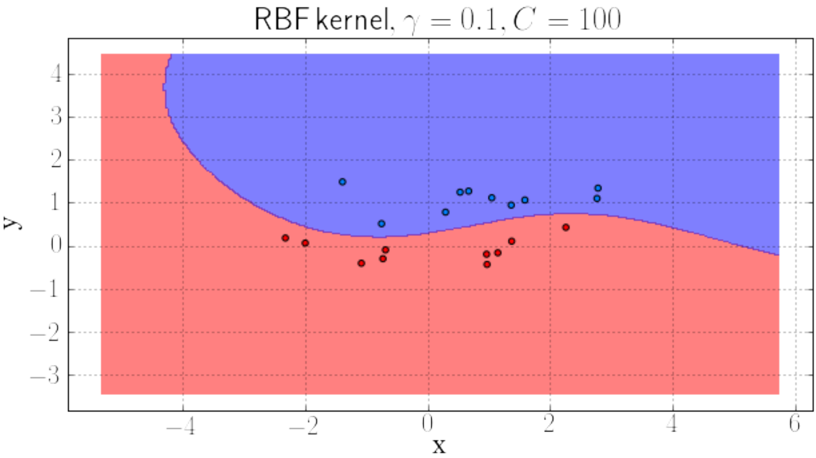
RBF kernel - variable C



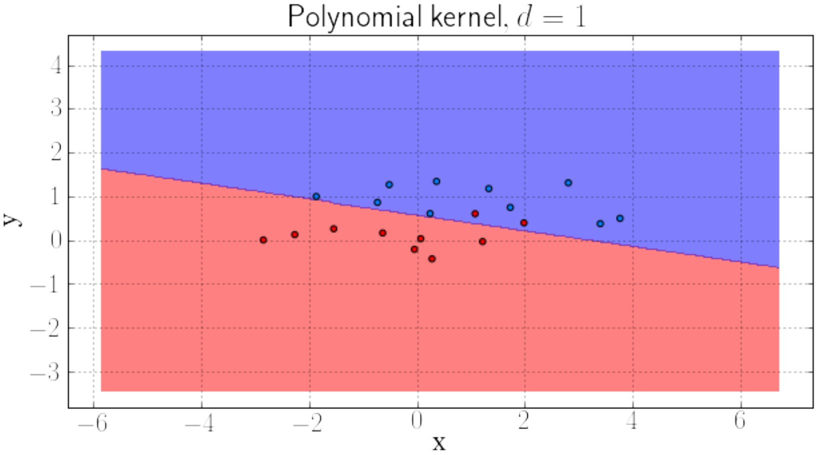
RBF kernel - variable C



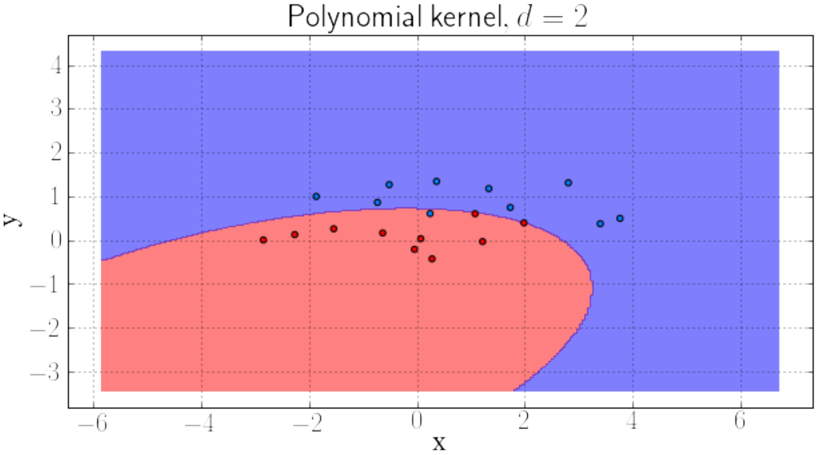
RBF kernel - variable C



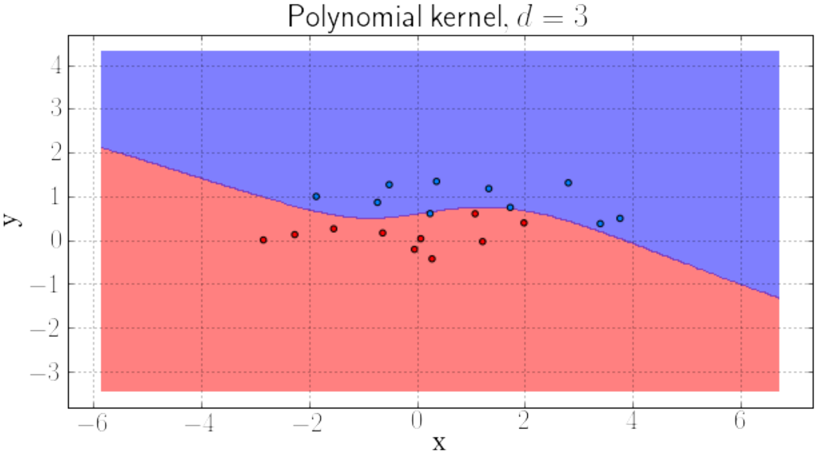
Polynomial kernel - variable d



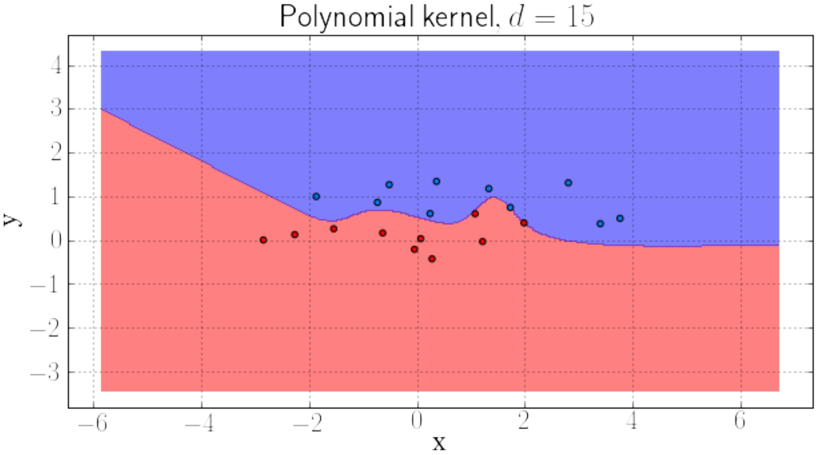
Polynomial kernel - variable d



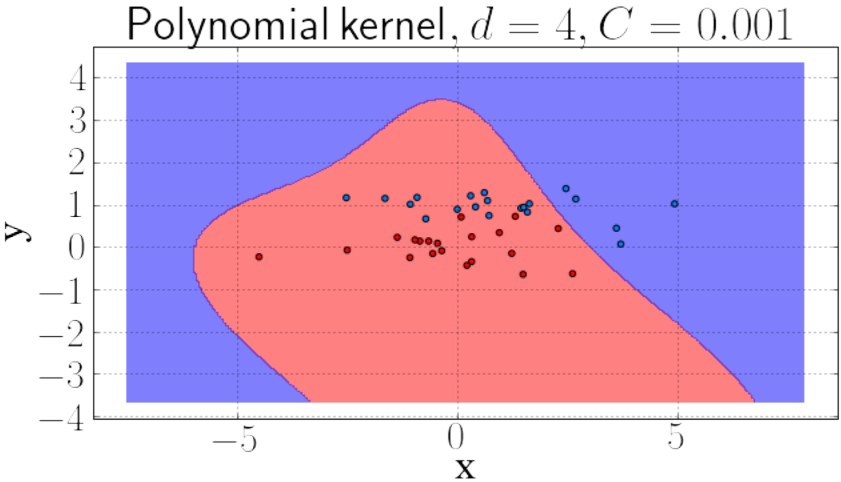
Polynomial kernel - variable d



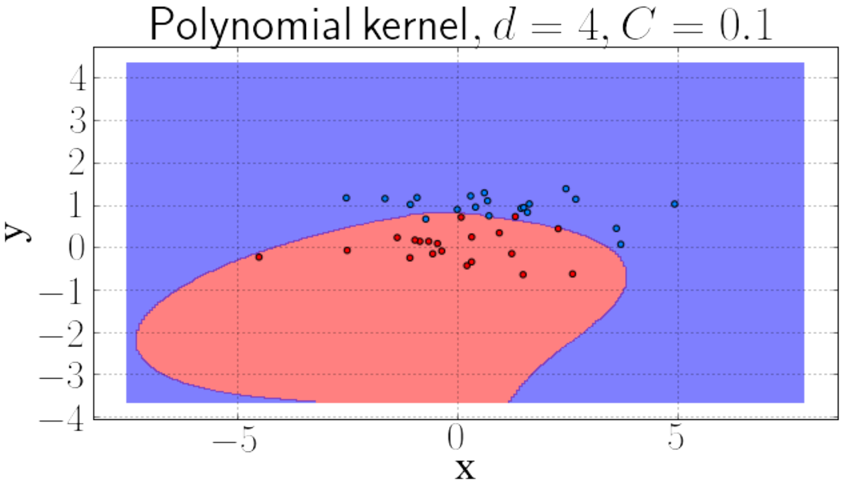
Polynomial kernel - variable d



Polynomial kernel - variable C



Polynomial kernel - variable C



Polynomial kernel - variable C