ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Dr. Francis Colas
Institute of Robotics and Intelligent Systems
Autonomous Systems Lab

ETH Zürich
CLA E 26 fcolas@mavt.ethz.ch
Tannenstraße 3 www.asl.ethz.ch
8092 Zürich Switzerland

Information Processing in Robotics
Exercise Sheet 6
Topic: Support vector machines

Exercise 1: Implementation of a Support Vector Machine

(a) The input is a set of points with their target value: $(\boldsymbol{x}_n, t_n)_n$

(b) The support vectors are vectors for which the Lagrange multiplier is not 0. They are the points closest to the border between the classes. Additionally to the support vectors, the bias is learned during the training phase. We propose the following signature:

```
float64[] x_vector
int8[] t_vector
---
SupportVector[] support_vectors
float64 bias
```

Where `SupportVector` is a message type:

```
float64[] x
int8 t
float64 a
```

(c)
- $\boldsymbol{P} = (t_i * t_j * k(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j}$,
- $\boldsymbol{q} = -\mathbf{1}$,
- $\boldsymbol{G} = -\boldsymbol{I}$,
- $\boldsymbol{h} = \mathbf{0}$,
- $\boldsymbol{A} = \emptyset$,
- $\boldsymbol{b} = \emptyset$.

(d) See code.

(e) $\text{sign}(\sum_{n \in \mathcal{S}} a_n t_n k(\boldsymbol{x}, \boldsymbol{x_n}) + b)$
See code.

Exercise 2: Experimenting with SVMs and kernel

We want a SVM classifier to discriminate points that are inside a disk centered on $(0, 0)$ with radius 1 from points that are outside this disk.

(a) SVMs allow for the supervised classification of binary data that is linearly separable in some feature space. In this case the separation is a circle (which is not linear) in the data space but using the kernel trick to project the data into a higher dimension space we can probably separate the two classes.

(b) In this case we can use:

$$\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{\phi}(\begin{pmatrix} x \\ y \end{pmatrix}) = \begin{pmatrix} x \\ y \\ x^2 + y^2 \end{pmatrix}.$$

We went from a 2-dimensions space to a 3-dimensions one. However, the data points won't be anywhere in this space, but are restricted to the paraboloid of revolution that we defined.

The associated kernel function is:

$$\begin{aligned} & k(\boldsymbol{x}, \boldsymbol{x}') \\ &= \boldsymbol{\phi}(\boldsymbol{x})^T . \boldsymbol{\phi}(\boldsymbol{x}') \\ &= x.x' + y.y' + (x^2 + y^2)(x'^2 + y'^2). \end{aligned}$$

(c) The boundary is always a hyperplane of the feature space; in this case, as the feature space is 3D, the boundary will be a 2D plane. Furthermore, the projection on paraboloid of revolution keep the center point in the lower part of the space, and the exterior point outside. Therefore the ideal boundary is simply a horizontal plane. The definition of this plane is simply $z = 1$.

To project that boundary in the data space, we have to consider that for the data points, $z = x^2 + y^2$ by the definition of our feature transformation function ϕ. Plugging this expression into the plane equation, we get $x^2 + y^2 = 1$ which is indeed the equation of a circle centered on $(0, 0)$ with radius 1.

(d) Now the boundary is an ellipsis with equation $x^2 + 2y^2 = 4$. We can quickly see that it cannot be expressed as a linear combination of x, y, and $z = x^2 + y^2$, that means that in the current feature space, the data is not linearly separable.

(e) In order to solve that, we need to change kernel. In this case, the simplest method is to distort our paraboloid:

$$\boldsymbol{\phi}'(\boldsymbol{x}) = \begin{pmatrix} x \\ y \\ x^2 + 2y^2 \end{pmatrix}.$$

For quadrics, we can also define a 5D space general enough to handle all cases:

$$\boldsymbol{\phi}''(\boldsymbol{x}) = \left(x, y, x^2, xy, y^2\right)^T.$$

We could also define the surface on which we project our points to be a cone, as quadrics are intersections of a cone and a plane (that solve the aspect ratio issue, but not the position of the center).

(f) See demo.