Support vector machines

Shady Nagy

Introduction

- In machine learning, support vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression.

- Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non probabilistic linear classifier.

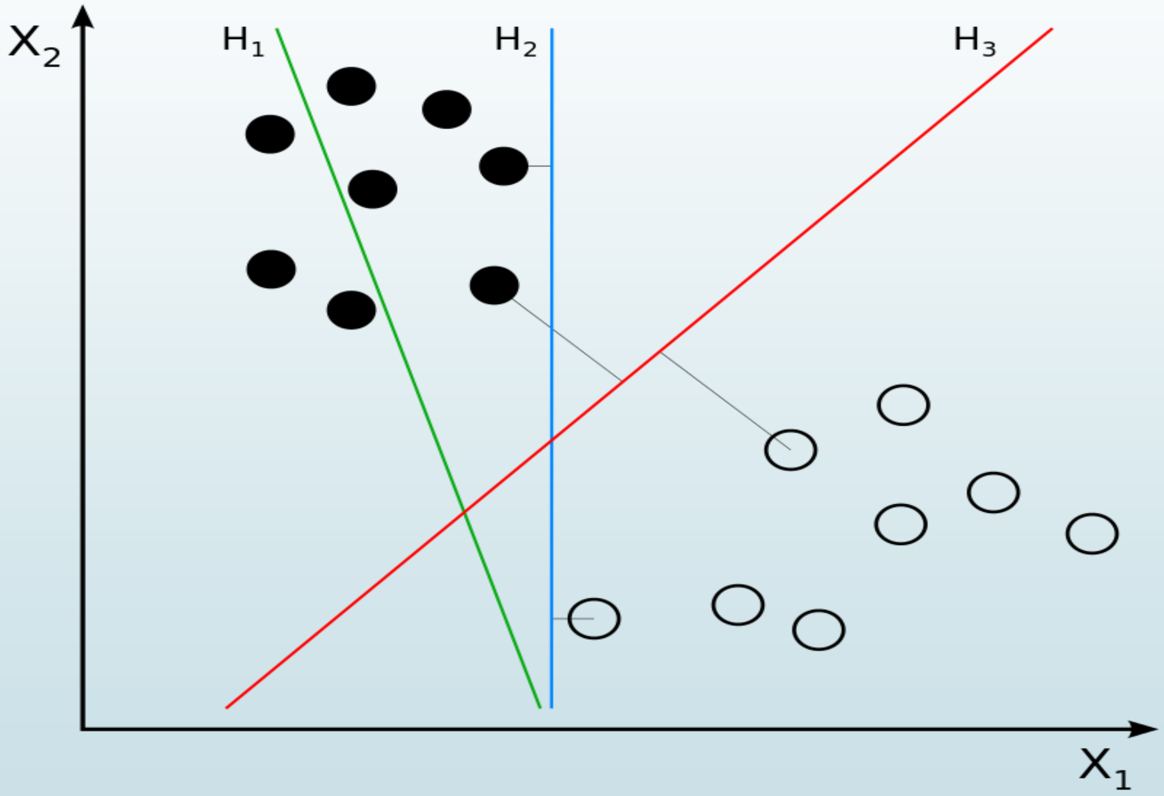
Introduction

- An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick mplicitly mapping their inputs into high-dimensional feature spaces.

History

Motivation

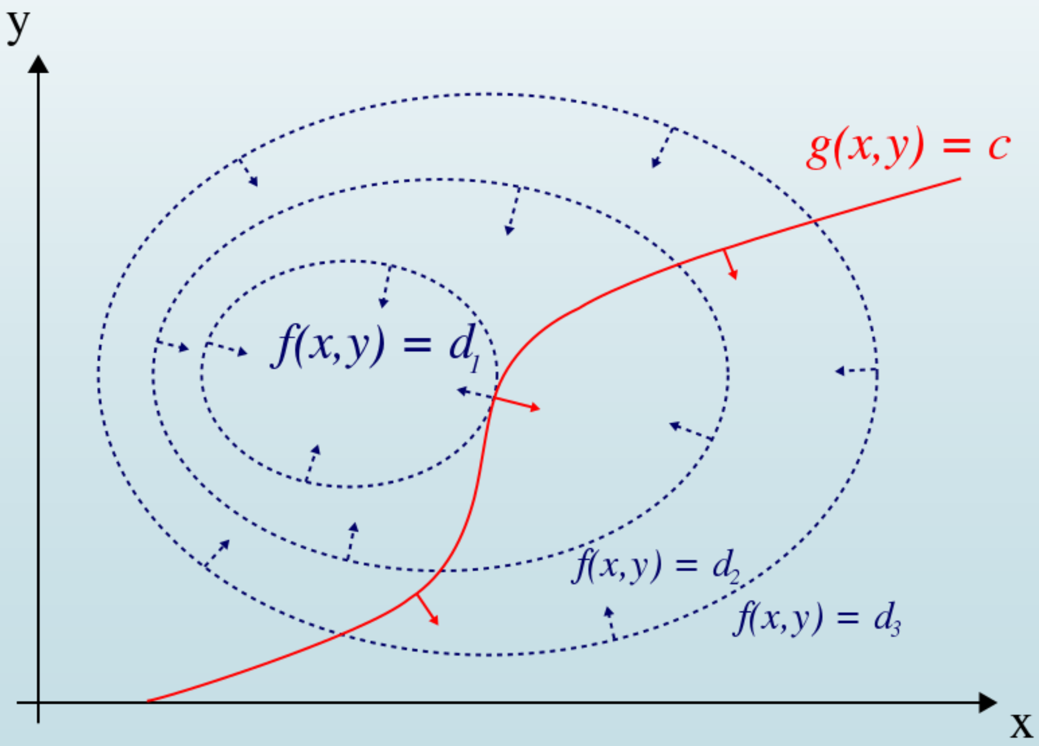


Lagrange Multiplier

- For the case of only one constraint and only two choice variables consider the optimization problem

  maximize $f(x, y)$

  subject to $g(x, y) = 0.$



Lagrange Multiplier

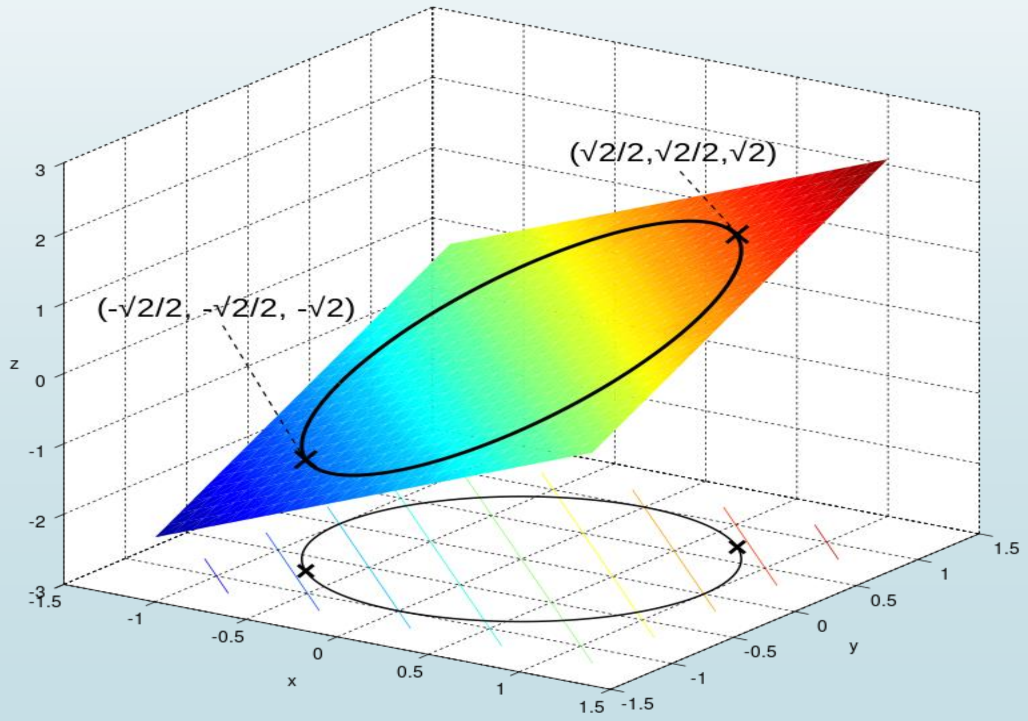
- The gradient of a function is perpendicular to the contour lines, the contour lines of f and g are parallel if and only if the gradients of f and g are parallel. Thus we want points (x, y) where $g(x, y) = 0$ and

$$\nabla_{x,y} f = \lambda \nabla_{x,y} g,$$

$$\nabla_{x,y,\lambda} \mathcal{L}(x, y, \lambda) = 0 \iff \begin{cases} \nabla_{x,y} f(x, y) = \lambda \nabla_{x,y} g(x, y) \\ g(x, y) = 0 \end{cases}$$

Lagrange Multiplier (Example)

- Suppose we wish to maximize $f(x,y) = x + y$
  subject to the constraint $x^2 + y^2 = 1$



Linear SVM

- We are given a training dataset of n points of the form

$$(\overrightarrow{x_1}, y_1), \dots, (\overrightarrow{x_n}, y_n)$$

Where y_i is either 1 or -1, each indicates the class where $\overrightarrow{x_i}$ belongs. Each $\overrightarrow{x_i}$ is a p dimensional real vector. We want to find the "maximum-margin hyperplane" that divides the group of points $\overrightarrow{x_i}$ for which $y_i = 1$ from the group of points for which $y_i = -1$.

which is defined so that the distance between the hyperplane and the nearest point $\overrightarrow{x_i}$ for either group is maximized.

Linear SVM

- Any hyperplane can be written as the set of points $\vec{x}$ satisfying

$$\vec{w} \cdot \vec{x} - b = 0,$$

$$\vec{w} \cdot \vec{x} - b = 1$$

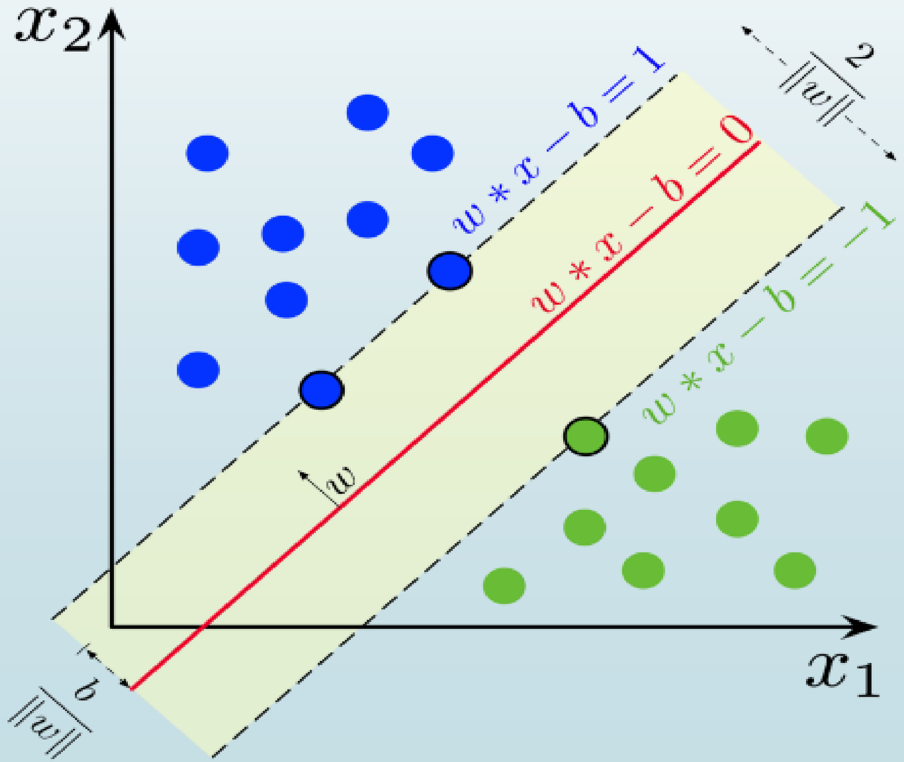
(anything on or above this boundary is of one class, with label 1)

$$\vec{w} \cdot \vec{x} - b = -1$$

(anything on or below this boundary is of the other class, with label −1).

Linear SVM

- Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.



Linear SVM

- In SVMs we are trying to find a decision boundary that maximizes the "margin" or the "width of the road" separating the positives from the negative training data points.

  To find this we minimize:

  $$\frac{1}{2}\|\vec{w}\|^2$$

  subject to the constraints

  $$y_i(\vec{w}\cdot\vec{x}_i+b)\geq 1$$

  The resulting Lagrange multiplier equation we try to optimize is:

$$L=\frac{1}{2}\|w\|^2-\sum_i\alpha_i(y_i(\vec{w}\cdot\vec{x}_i+b)-1)$$

Linear SVM

- Solving the above Lagrangian optimization problem will give us w, b, and alphas, parameters that determines a unique maximal margin (road) solution. On the maximum margin "road", the +ve, and -ve points that stride the "gutter" lines are called support vectors. The decision boundary lies at the middle of the road. The definition of the "road" is dependent only on the support vectors, so changing (adding deleting) non-support vector points will not change the solution. Note, that widest "road" is a 2D concept. If the problem is in 3D we want the widest region bounded by two planes; in even higher dimensions, a subspace bounded by two hyperplanes.

Linear SVM

- A. Equations derived from optimizing the Lagrangian:

1. **Partial of the Lagrangian wrt to b**: From $\frac{\partial L}{\partial b} = 0$

$\sum_i \alpha_i y_i = 0$	Note that $y_i \in \{-1, +1\}$ and $\alpha_i = 0$ for non-support vectors.

2. **Partial of the Lagrangian wrt to w**: From $\frac{\partial L}{\partial w} = 0$

$\sum_i \alpha_i y_i \vec{x_i} = \vec{w}$	For when using a linear kernel. The summation only contains support vectors. Support vectors are training data points with $\alpha_i > 0$
$\sum_i \alpha_i y_i \phi(\vec{x_i}) = \vec{w}$	For when using a decomposable kernel (see definition below).

Linear SVM

B. Equations from the boundaries and constraints:

3. The Decision boundary:

$h(\vec{x}) = \sum_i \alpha_i y_i K(\vec{x_i}, \vec{x}) + b \geq 0$	General form, for any kernel. To classify an unknown $\vec{x}$, we compute the kernel function $K(\vec{x_i}, \vec{x})$ against each of the **support vectors** $\vec{x_i}$. Support vectors are training data points with $\alpha_i > 0$
$h(\vec{x}) = \sum_i [(\alpha_i y_i \vec{x_i}) \cdot \vec{x}] + b \geq 0$ $h(\vec{x}) = \vec{w} \cdot \vec{x} + b \geq 0$	For when using a linear kernel $K(\vec{x_i}, \vec{x}) = \vec{x_i} \cdot \vec{x}$

Linear SVM

- **4. Positive gutter:**

$h(\vec{x})=\sum_i \alpha_i y_i K(\vec{x}_i,\vec{x})+b=1$	General form, for any kernel.
$h(\vec{x})=\sum_i[(\alpha_i y_i \vec{x}_i)\cdot\vec{x}]+b=1$ $h(\vec{x})=\vec{w}\cdot\vec{x}+b=1$	For use when the Kernel is linear.

- **5. Negative gutter:**

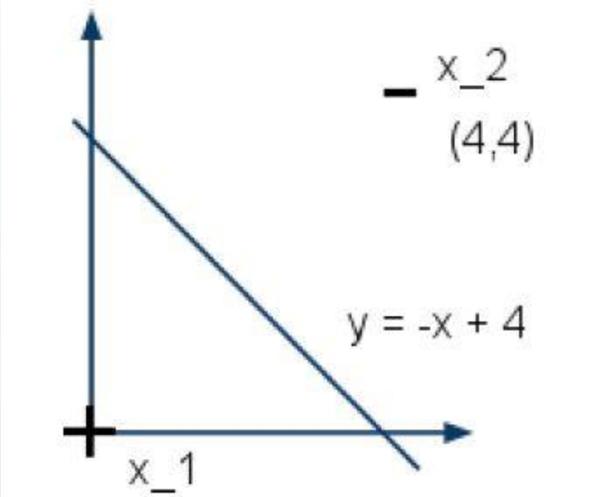
$$h(\vec{x})=\sum_i \alpha_i y_i K(\vec{x}_i,\vec{x})+b=-1 \qquad h(\vec{x})=\vec{w}\cdot\vec{x}+b=-1$$

- **6. The width of the margin (or road):**

$$\text{width of road} \equiv m=\frac{2}{\|\vec{w}\|} \quad \text{where,} \quad \|\vec{w}\|=\sqrt{\sum_i w_i^2}$$

Example on Linear SVM.

- **Method 1 of Solving SVM parameters by inspection:** This is a step-by-step solution to Problem: We are given the following graph with and points on the x-y axis; +ve point at x_1 $(0, 0)$ and a -ve point x_2 at (4, 4).



Can a SVM separate this? i.e. is it linearly separable? Yeah! using the line above.

Example on Linear SVM.

- We can find the decision boundary by graphical inspection.

1. The decision boundary lies on the line: $y = -x + 4$
2. We have a +ve support vector at (0, 0) with line equation $y = -x$
3. We have a -ve support vector at (4, 4) with line equation $y = -x + 8$

Given the equation for the decision boundary, we next massage the algebra to get the decision boundary to conform with the desired form, namely:

$$h(\vec{x}) = w_1 x + w_2 y + b \geq 0$$

1. $y < -x + 4$ (< because +ve is below the line)
2. $x + y - 4 < 0$
3. $-x - y + 4 \geq 0$ (multiplied by -1)
4. $-1x - 1y + 4 \geq 0$ (writing out the coefficients explicitly)

Example on Linear SVM.

- Now we can read the solution from the equation coefficients: $w_1 = -1$, $w_2 = -1$ $b = 4$ Next, using our formula for width of road, we check that these weights gives a road width of:

$$-cx_1 - cx_2 + 4c \geq 0$$

$w_1 = -c \quad w_2 = -c \quad b = 4c$
or $\vec{w} = \begin{bmatrix} -c \\ -c \end{bmatrix}$ and $b = 4c$

Example on Linear SVM.

- Using The Width of the Road Constraint Graphically

we see that the widest width margin should be: $4\sqrt{2}$ The solution weight vector and intercept can be solved by solving for c constrained by the known width-of-the-road. Length of $\vec{w}$ in terms of c:

$$\|\vec{w}\| = \sqrt{(-c)^2 + (-c)^2} = \sqrt{2}c$$

Now plugin all this into the margin width equation and solving for c, we get:

$$\frac{2}{\|\vec{w}\|} = 4\sqrt{2} \quad \text{=>} \quad \frac{2}{\sqrt{2}c} = 4\sqrt{2} \quad \text{=>} \quad \frac{2}{c} = 4 \cdot 2 \quad \text{=>} \quad c = \frac{1}{4}$$

This means the true weight vector and intercept for the SVM solution should be:

$$\vec{w} = \begin{bmatrix} -\frac{1}{4} \\ -\frac{1}{4} \end{bmatrix} \text{ and } b = 4 \cdot \frac{1}{4} = 1$$

Example on Linear SVM.

Next we solve for alphas, using the w vector and equation 1.

$$\sum_i \alpha_i y_i \vec{x_i} = \vec{w}$$

Plugin in the vector values of support vectors and w:

$$\alpha_1(+1)\vec{x_1} + \alpha_2(-1)\vec{x_2} = \alpha_1(+1)\begin{bmatrix}0\\0\end{bmatrix} + \alpha_2(-1)\begin{bmatrix}4\\4\end{bmatrix} = \begin{bmatrix}-\frac{1}{4}\\-\frac{1}{4}\end{bmatrix}$$

We get two identical equations:

$$-\frac{1}{4} = -4\alpha_2 \text{ or } \alpha_2 = \frac{1}{16}$$

now we can solve for the other alpha:

$$(+1)\alpha_1 + (-1)\alpha_2 = 0$$
$$\alpha_1 = \alpha_2 = \frac{1}{16}$$

Common SVM Kernels.

- Decomposable Kernels Idea: Define $\phi(\vec{u})$ that transforms input vectors into a different (usually higher) dimensional space where the data is (more easily) linearly separable.

$$K(\vec{u},\vec{v}) = \phi(\vec{u}) \cdot \phi(\vec{v})$$

Example:

$$\phi(\vec{u}) = \begin{bmatrix} \cos(u_1) \\ \sin(u_2) \end{bmatrix} \quad K(\vec{u},\vec{v}) = \cos(\vec{u_1})\cos(\vec{v_1}) + \sin(\vec{u_2})\sin(\vec{v_2})$$

Common SVM Kernels:

- **Polynomial Kernel**

$$K(\vec{u},\vec{v}) = (\vec{u}\cdot\vec{v}+b)^n \quad n > 1$$

Example: Quadratic Kernel: $K(\vec{u},\vec{v}) = (\vec{u}\cdot\vec{v}+b)^2$

- In 2D resulting decision boundary can look parabolic, linear or hyperbolic depending on which terms in the expansion dominate.
- Here is an expansion of the quadratic kernel, with u = [x, y]

$$K(\vec{u},\vec{v}) = \left(\begin{bmatrix} x \\ y \end{bmatrix}\cdot\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}+b\right)^2$$
$$= (v_1 x + v_2 y + b)^2$$
$$= [(v_1^2)x^2 + (v_2^2)y^2] + [b^2 + (2v_1 b)x + (2v_2 b)y] + [(2v_1 v_2)xy]$$

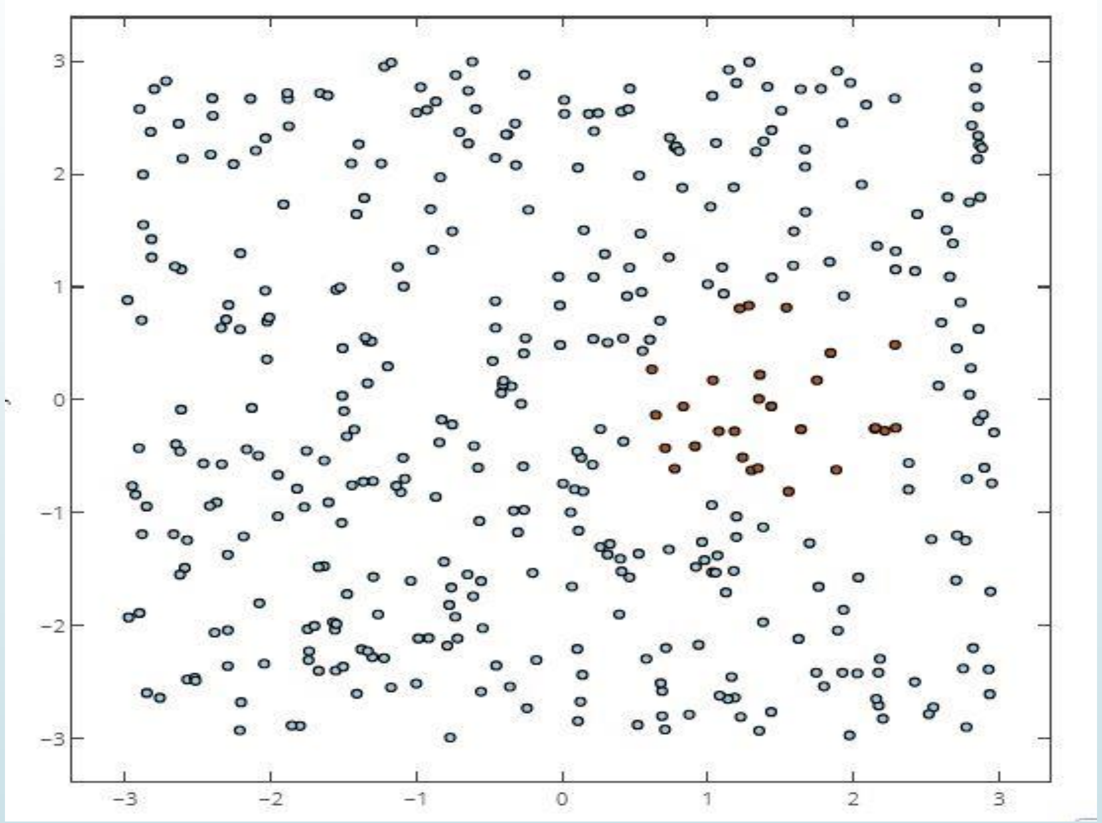
Common SVM Kernels.

- Radial Basis Function (RBF) or Gaussian Kernel
- Will fit almost any data. May exhibit overfitting when used improperly.
- Similar to KNN but with all points having a vote; weight of each vote determined by Gaussia Points farther away get less of a vote than points nearby

$$K(\vec{u},\vec{v}) = \exp\left(-\frac{\|\vec{u}-\vec{v}\|^2}{2\sigma^2}\right)$$

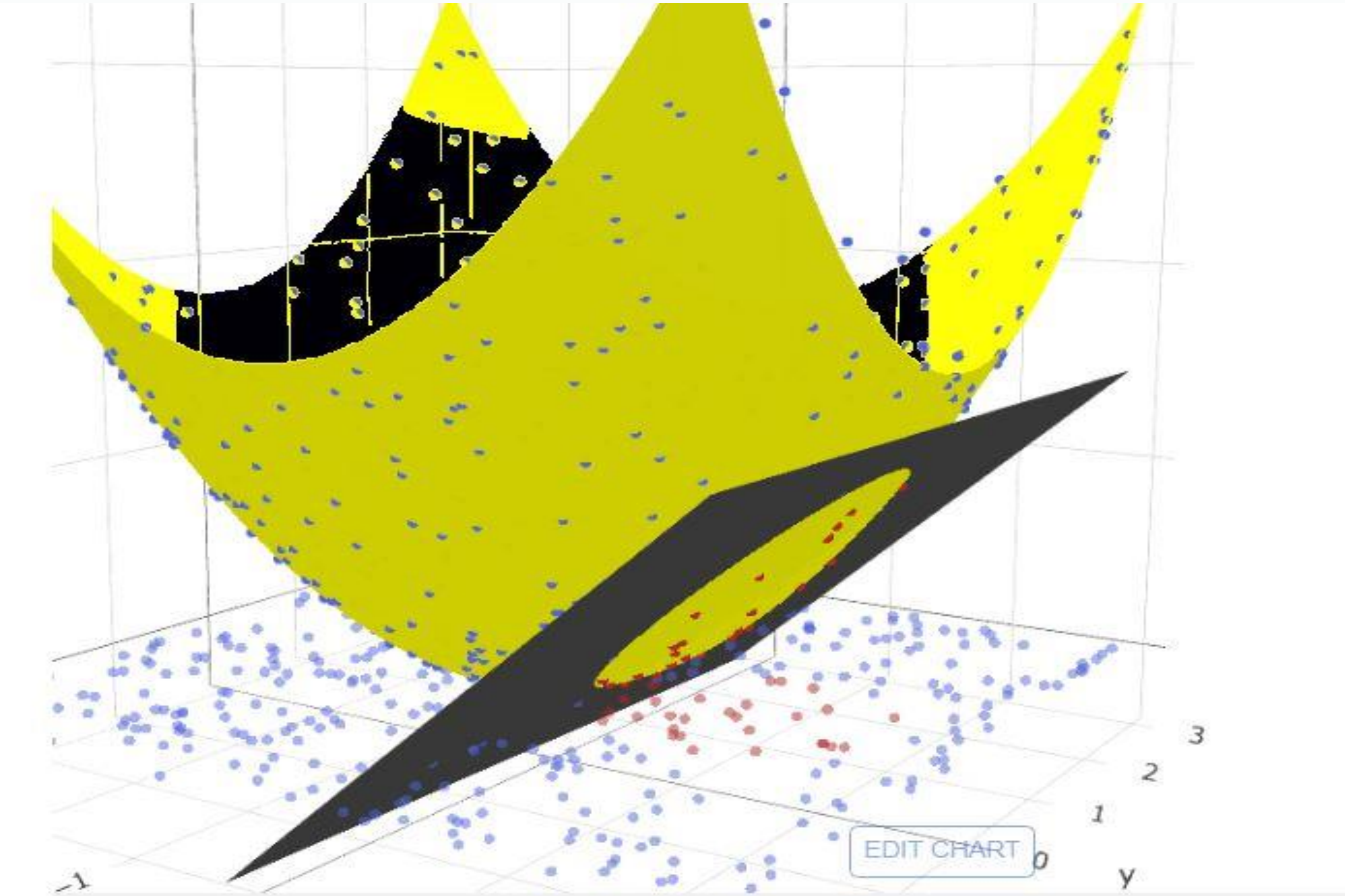
When σ^2 is large you get flatter Gaussians. When σ^2 is small you get sharper Gaussians. (Hence when using a small contour density will appear closer / denser around support vector points).

Common SVM Kernels.



However, if we map the 2-d input data $x = (x, y)$ to 3-d feature space by a function $\Phi(x) = (x, y, x^2 + y^2)$

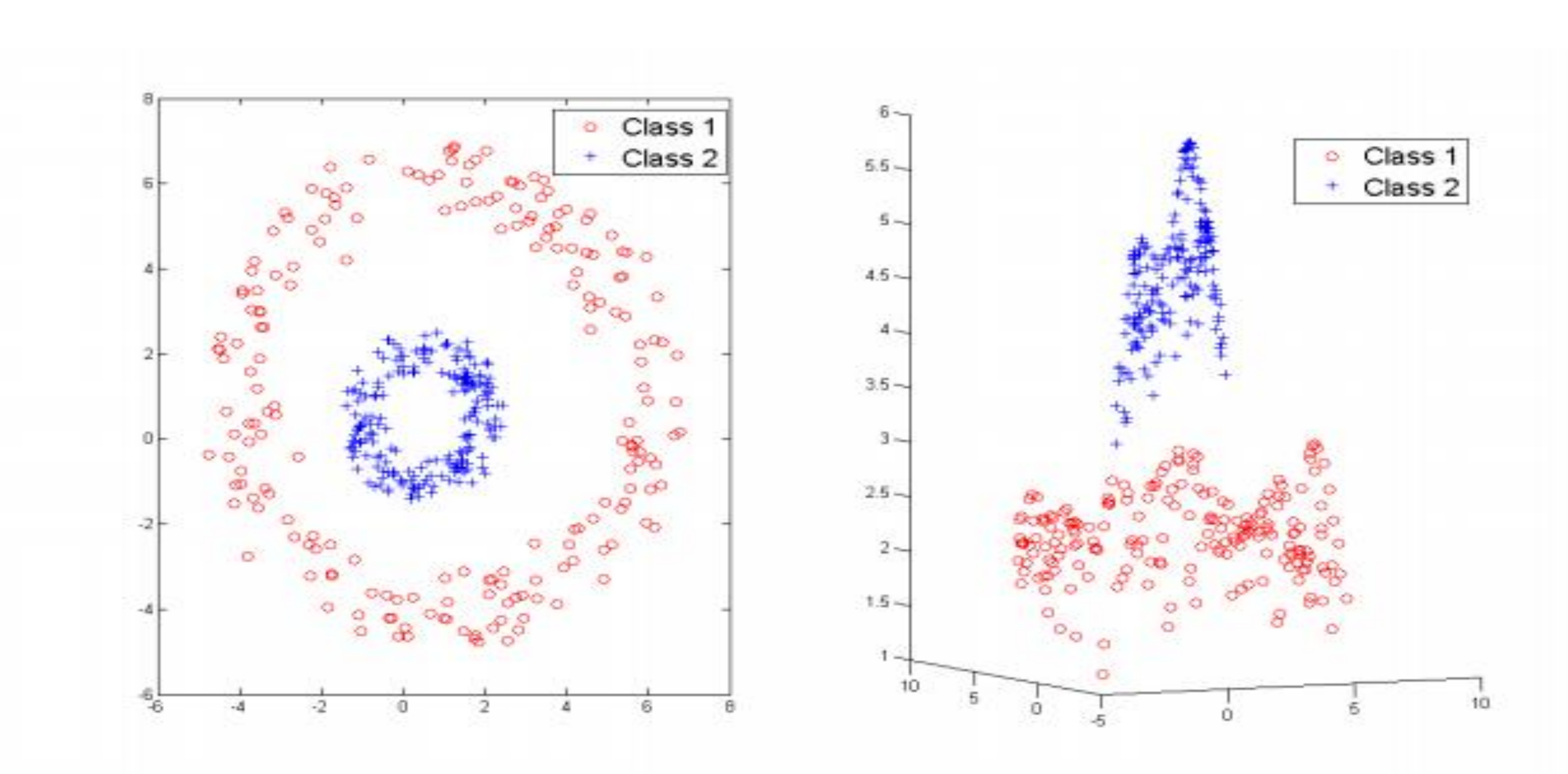
Common SVM Kernels.



However, if we map the 2-d input data $x = (x, y)$ to 3-d feature space by a function $\Phi(x) = (x, y, x^2 + y^2)$

Common SVM Kernels.

- Gaussian Kernel



Thank you!