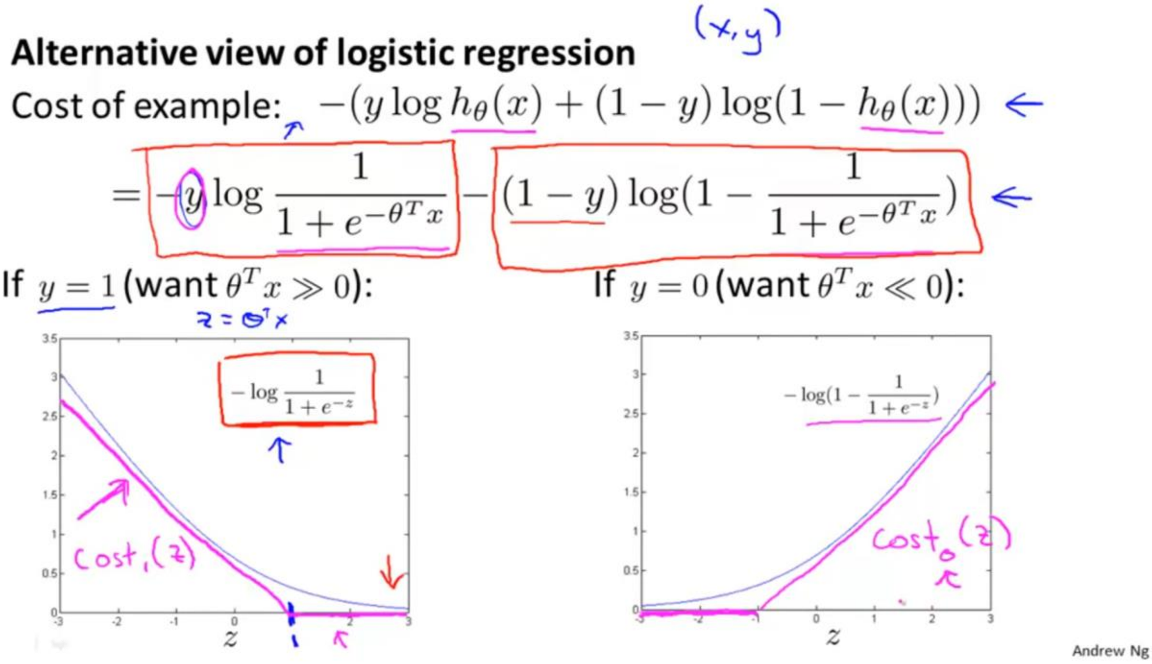
Introduction:

Take a look at the figure below:



We modify the functions $\boxed{-\log \frac{1}{1+e^{-z}}}$ and $-\log(1 - \frac{1}{1+e^{-z}})$ to obtain new cost functions namely $cost_1(z)$ and $cost_0(z)$ respectively. If y=1 we want $theta^T$ *x>>0 so that our cost function is low. When y=0 , we want $theta^T$*x to be <<0 so that our cost function is low.

Support vector machine

Logistic regression:

$$\min_\theta \underbrace{\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\underbrace{\left(-\log h_\theta(x^{(i)})\right)}_{cost_1(\theta^T x^{(i)})} + (1-y^{(i)})\underbrace{\left((-\log(1-h_\theta(x^{(i)})))\right)}_{cost_0(\theta^T x^{(i)})}\right]}_{A} + \underbrace{\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2}_{B} \quad - abc$$

Support vector machine:

$$\min_\theta \cancel{\frac{1}{m}}\, C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)})\, cost_0(\theta^T x^{(i)}) + \frac{1\cancel{\lambda}}{2\cancel{m}}\sum_{j=0}^{n}\theta_j^2$$

$\min_u \left((u-5)^2+1\right) \times 10 \rightarrow u=5$

$\min_u 10(u-5)^2+10 \rightarrow u=5$

$A + \lambda B \leftarrow$

$C A + B \leftarrow$

$C = \frac{1}{\lambda}$

$$\rightarrow \min_\theta C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2 \quad \leftarrow \text{③}$$

Graph 1 corresponds to $cost_1(theta^T * x^{(i)})$ and graph 2 corresponds to $cost_0(theta^T * x^{(i)})$.

To find minimum of equation abc we have to equate it's derivates to 0. If take 1/m common from both terms A and B , it gets removed. We can λ also take common from term B , then the equation becomes :

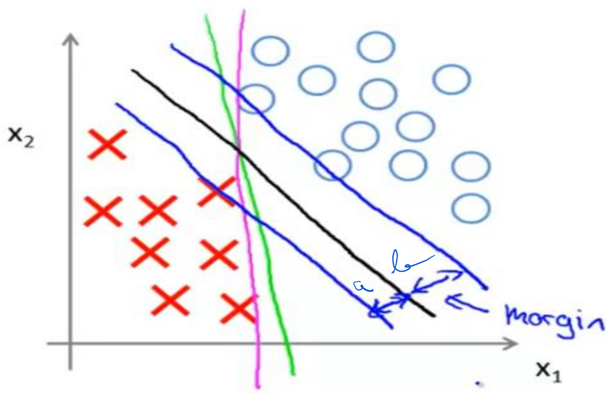
$\frac{\lambda}{m}\left(\frac{1}{\lambda} * a + b\right)$

To find minima, we take derivative of above equation and equate it to zero.

Equation 3 describes the cost function of SVM's.

Decision boundary of SVM's:

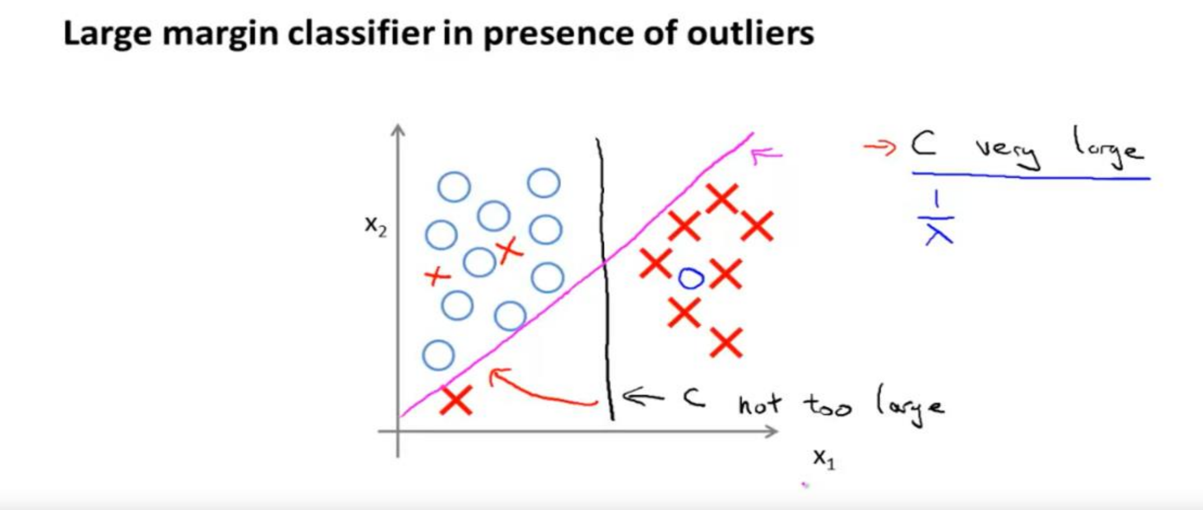
CASE 1:



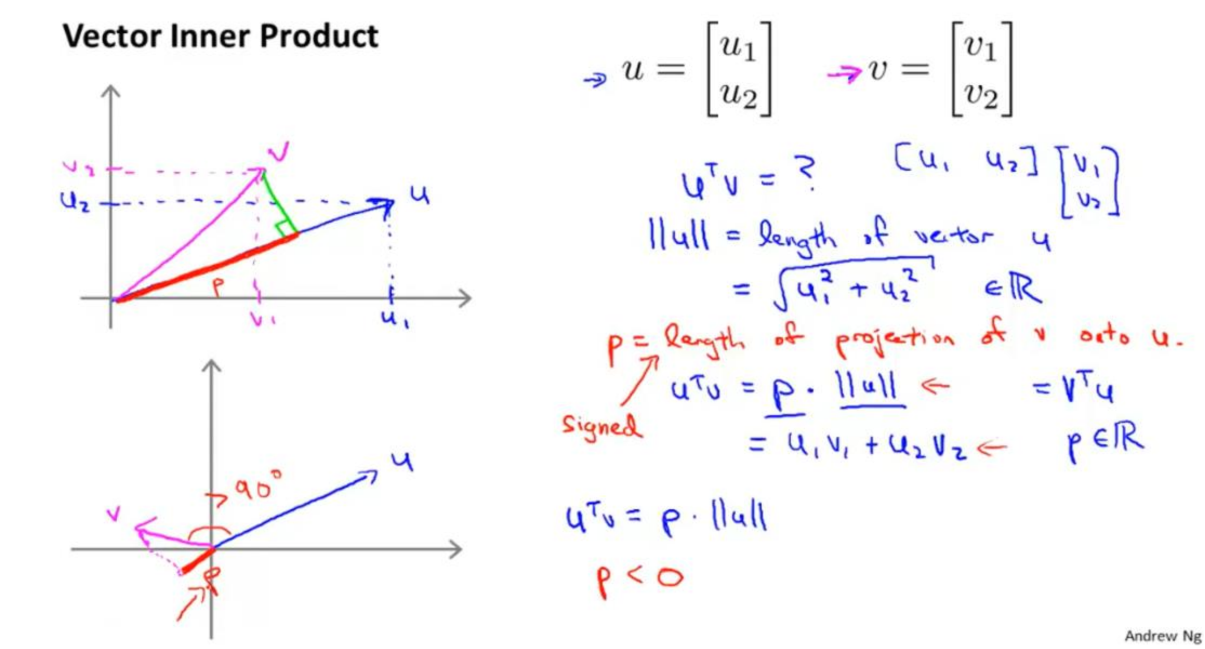
Large margin classifier

For a sufficiently large value of C, in the above case an SVM will draw a decision boundary that has an equal margin of separation from both the values of y, i.e. a=b.

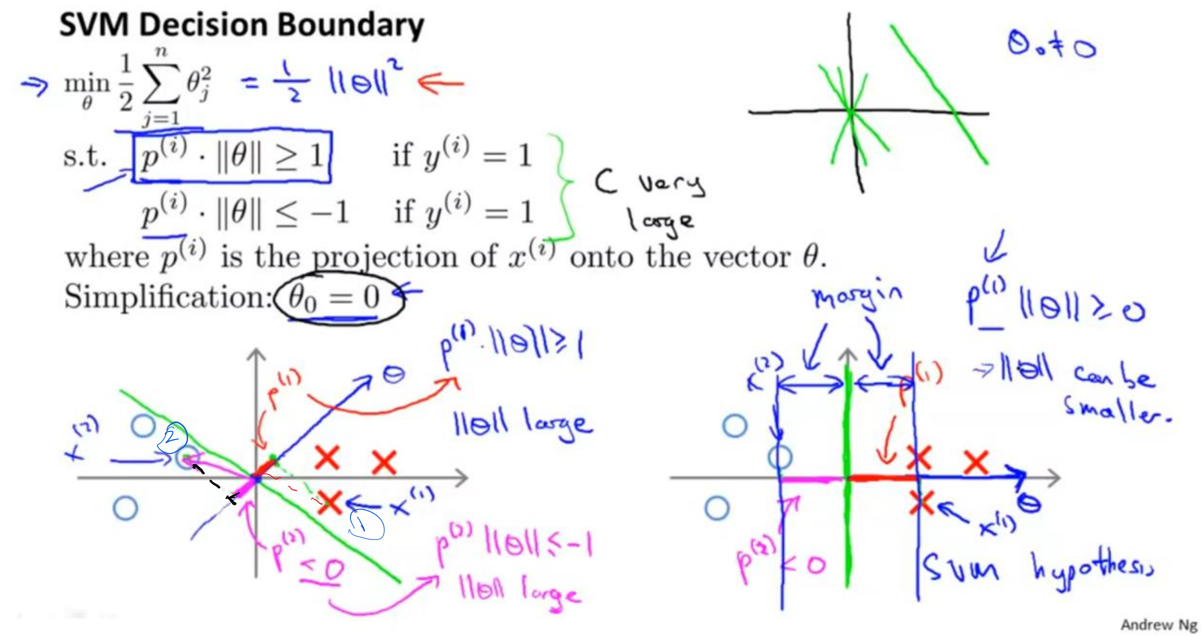
CASE 2:



Mathematics Behind Large Margin Classification:



When we draw a perpendicular line from vector v to u, then P is base of the triangle formed. Note P and u are part of the same line always. P is the projection of v on u. Here 'v' represents x and 'u' represents theta. The green line perpendicular to the theta vector is the decision boundary.



From the above figure, we observe that:

1.The decision boundary is represented by green and since the line representing theta is always perpendicular to the decision boundary, it is represented by the blue line.

2. The point 1 is obtained by multiplying second row of x values (the 1st training set) with respective theta values.

3. The point 2 is obtained by multiplying second row of x values (the 2nd training set) with respective theta values.

4. The thick line in red colour is the projection of x^1(represented by dotted red line) on theta (represented by blue line).

5. The thick line in the pink colour is the projection of x^2(represented by pink line) on theta (represented by blue line).

6. It is clear from the graph on the left side of the figure that projection of x^1 and x^2 is quite small in magnitude. This is due to the fact that the circle and cross pertaining to x^2 and x^1 respectively is quite close to the decision boundary.

For $p^{(1)} \times \|\theta\|$ to be ≥ 1, we need that $\|\theta\|$ is very large, since P is small.

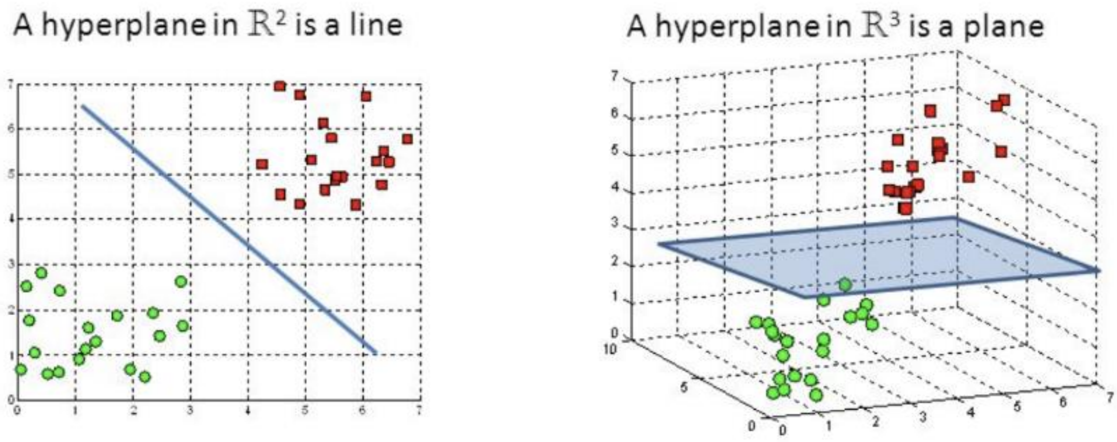
Now if $\|\theta\|$ is large then our cost function will also get large. To avoid this the SVM instead uses a decision boundary shown on graph there on the right side of the above figure. In this graph, since P values for both x^1 and x^2 are sufficiently large, theta vector doesn't have to be so large and the cost function is not affected.

Hence the SVM algorithm prefers decision boundary shown in graph 2 instead of the one shown in graph 1.

Why are SVM's called as large margin classifiers?

To answer this question we firstly try to understand what is a hyperplane:

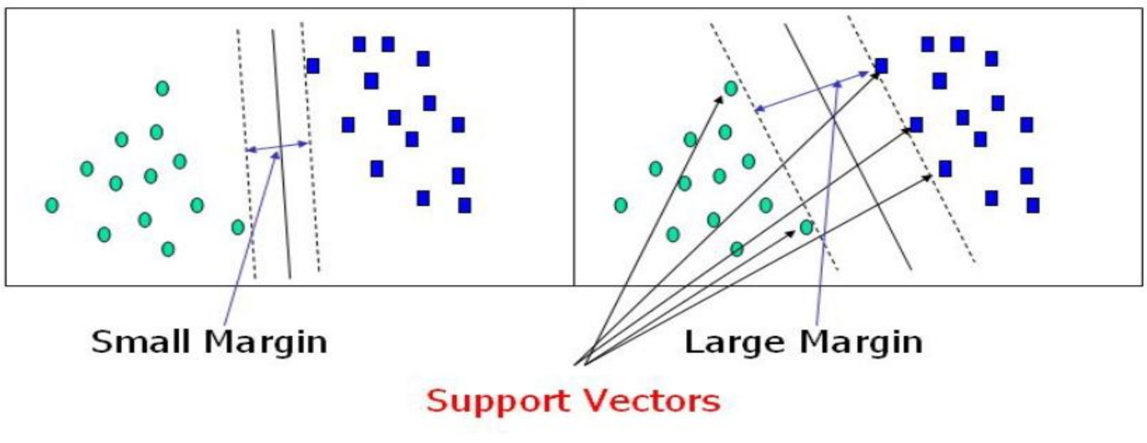
Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.



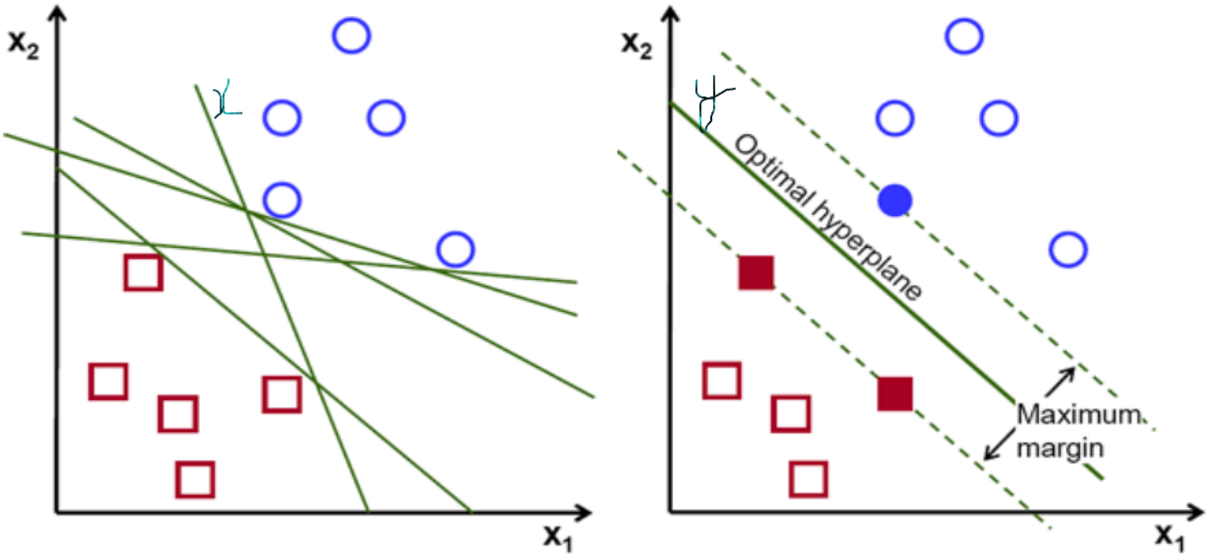
Hyperplanes in 2D and 3D feature space

Hyperplanes and Support Vectors:

Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyperplane. These are the points that help us build our SVM.



To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence. In other words, more margin means more room for error, i.e. if we have an outlier , a large margin helps us accommodate it in the correct class.



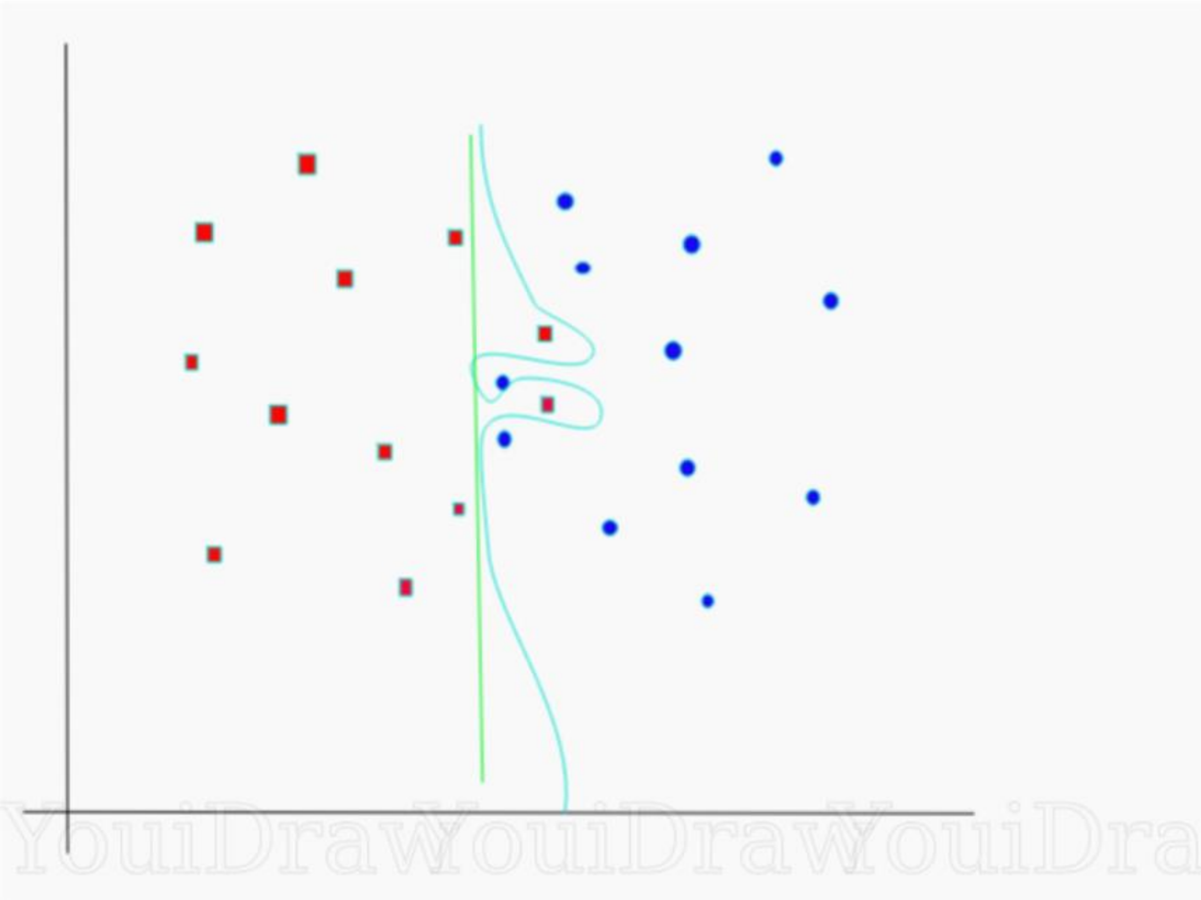
Another thing to be noted is that if value of C is large SVM would predict the training set quite correctly and would lead to overfitting, whereas when C is small, SVM would not be prone to overfitting but can lead to underfitting. If we consider the above figure , then we can say that the SVM will give decision boundary x if C is large and y if C is small.

The parameter sigma (σ)determines the reach of a single training instance. If the value of sigma is low, then every training instance will have a far reach. Conversely, high values of sigma mean that training instances will have a close reach. So, with a high value of sigma, the SVM decision boundary will simply be dependent on just the points that are closest to the decision boundary, effectively ignoring points that are farther away. In comparison, a low value of sigma will result in a decision boundary that will consider points that are further from it. As a result, high values of sigma typically produce highly flexed decision boundaries, and low values of sigma often results in a decision boundary that is more linear.

It defines how far the influence of a single training example reaches. If it has a low value it means that every point has a far reach and conversely high value of sigma means that every point has close reach.

To sum up about sigma: If sigma has a very high value, then the decision boundary is just going to be dependent upon the points that are very close to the line which effectively results in ignoring some of the points that are very far from the decision boundary. This is because the closer points get more weight and it results in a wiggly curve as shown in graph below.

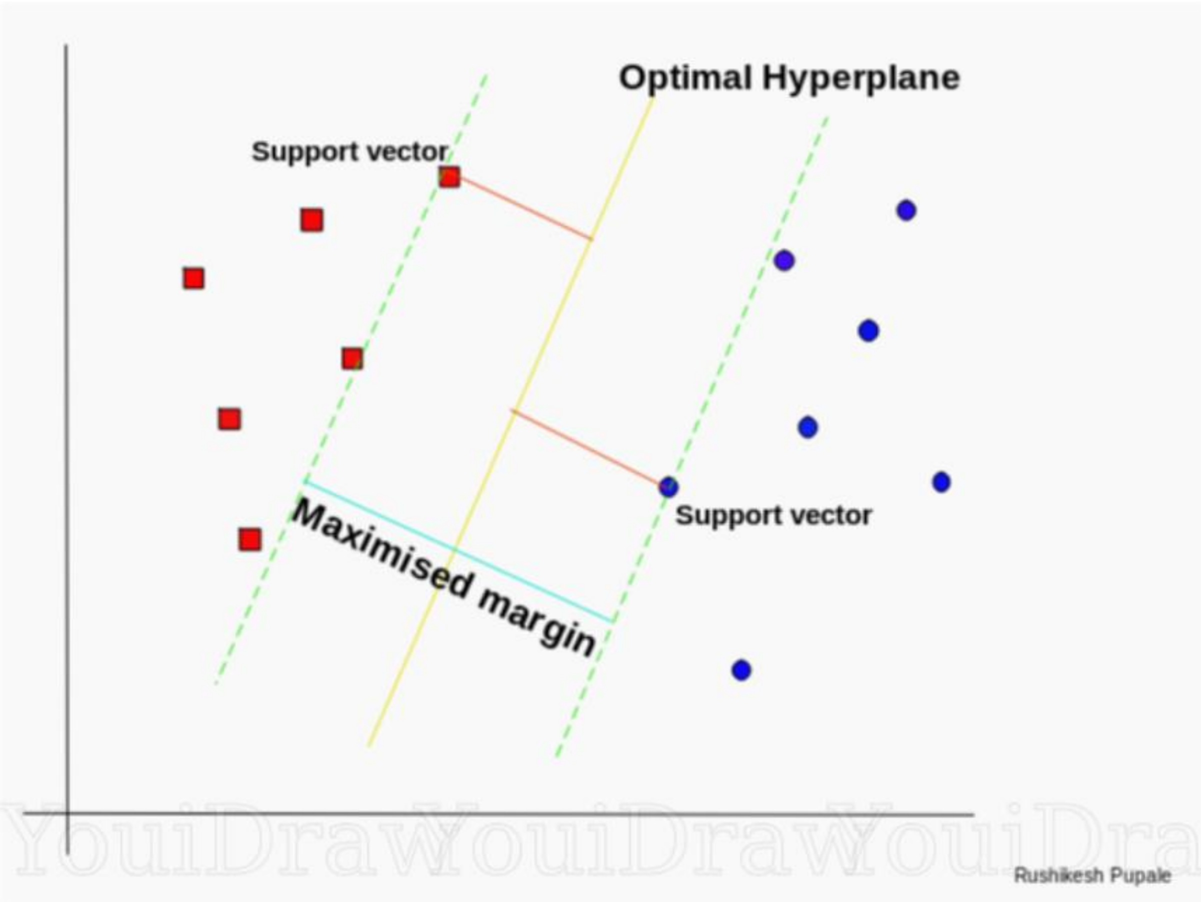
On the other hand, if the sigma value is low even the far away points get considerable weight and we get a more linear curve.



Consider an example as shown in the figure above. There are a number of decision boundaries that we can draw for this dataset. Consider a straight (green colored) decision boundary which is quite simple but it comes at the cost of a few points being misclassified. These misclassified points are called outliers. We can also make something that is considerably more wiggly(sky blue colored decision boundary) but where we get potentially all of the training points correct. Of course the trade off having something that is very intricate, very complicated like this is that chances are it is not going to generalize quite as well to our test set. So something that is simple, more straight maybe actually the better choice if you look at the accuracy. Large value of c means you will get more intricate decision curves trying to fit in all the points. Figuring out how much you want to have a smooth decision boundary vs one that gets things correct is part of artistry of machine learning.

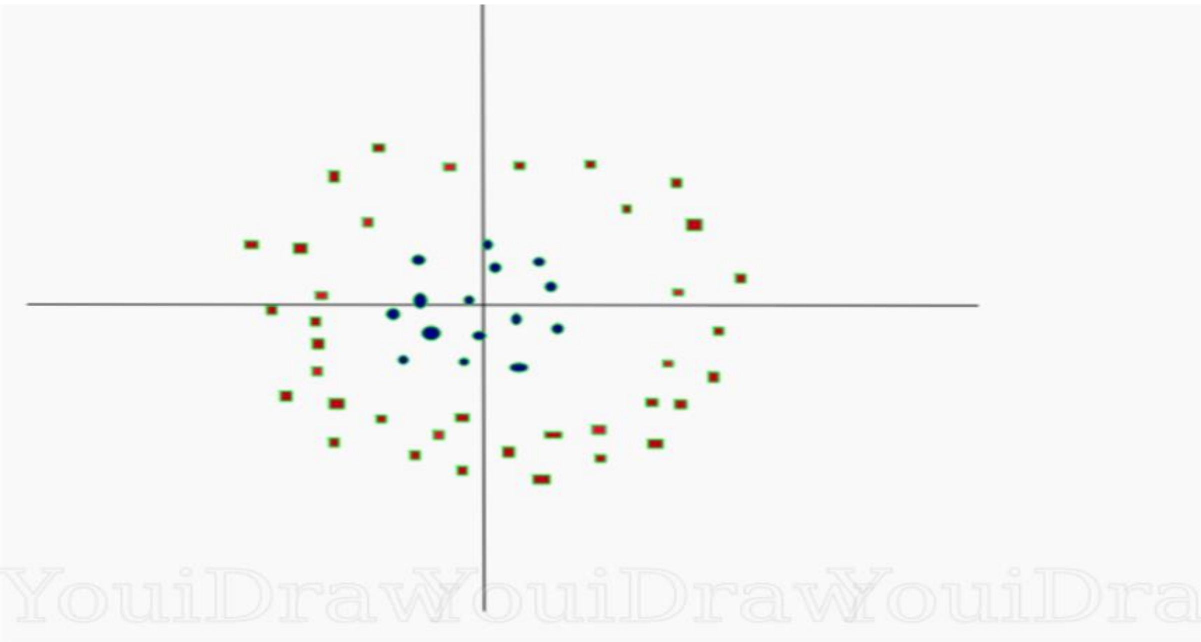
SVM's way to find the best line

According to the SVM algorithm we find the points closest to the line from both the classes. These points are called support vectors. Now, we compute the distance between the line and the support vectors. This distance is called the margin. Our goal is to maximize the margin. The hyperplane for which the margin is maximum is the optimal hyperplane.



Thus SVM tries to make a decision boundary in such a way that the separation between the two classes(that street) is as wide as possible.

Now let's consider a bit complex dataset, which is not linearly separable.



This data is clearly not linearly separable. We cannot draw a straight line that can classify this data. But, this data can be converted to linearly separable data in higher dimension. Lets add one more dimension and call it z-axis. Let the co-ordinates on z-axis be governed by the constraint,

$z = x^2+y^2$

So, basically z co-ordinate is the square of distance of the point from origin. Let's plot the data on z-axis.

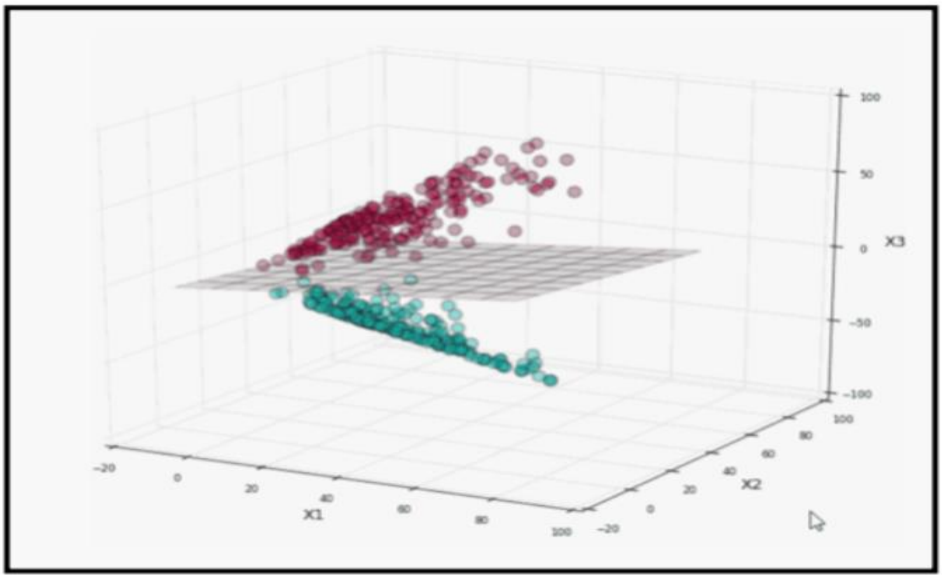
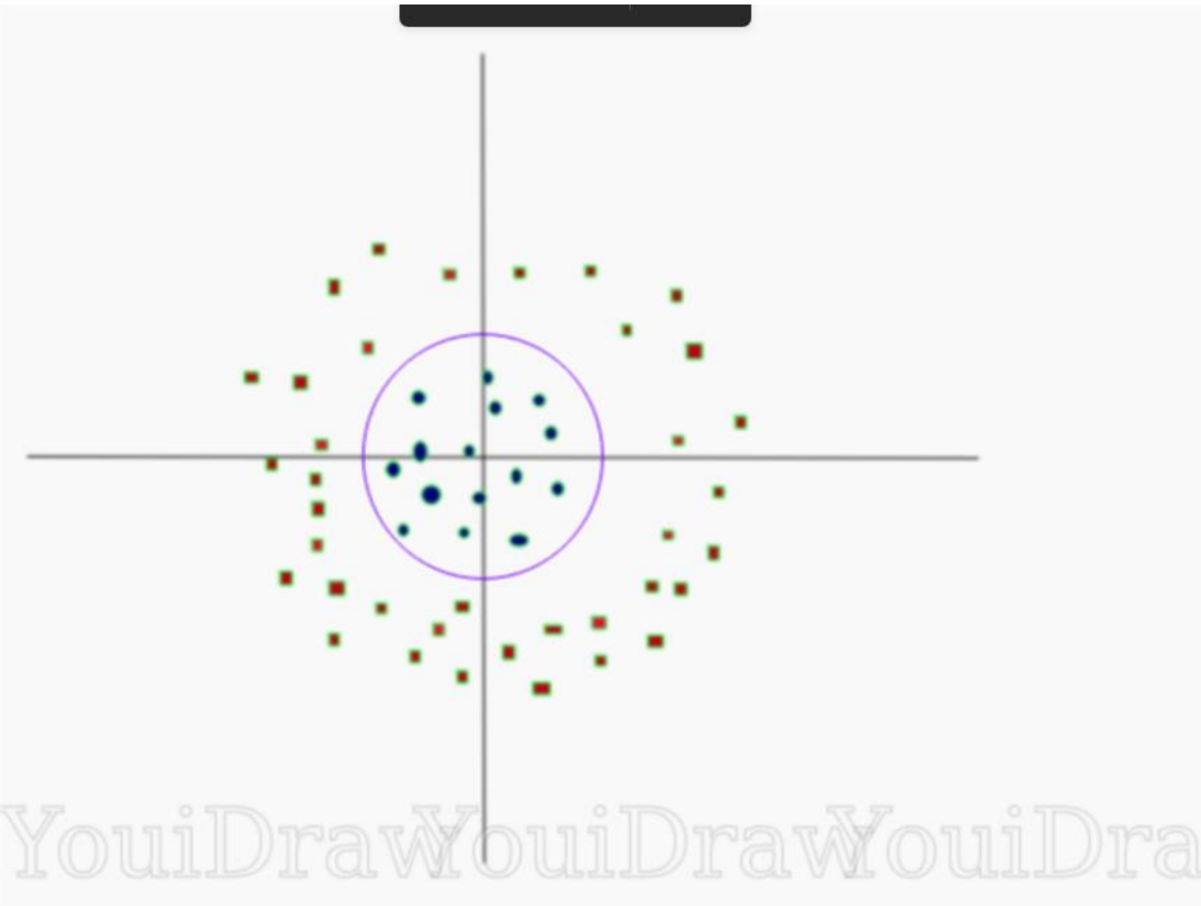


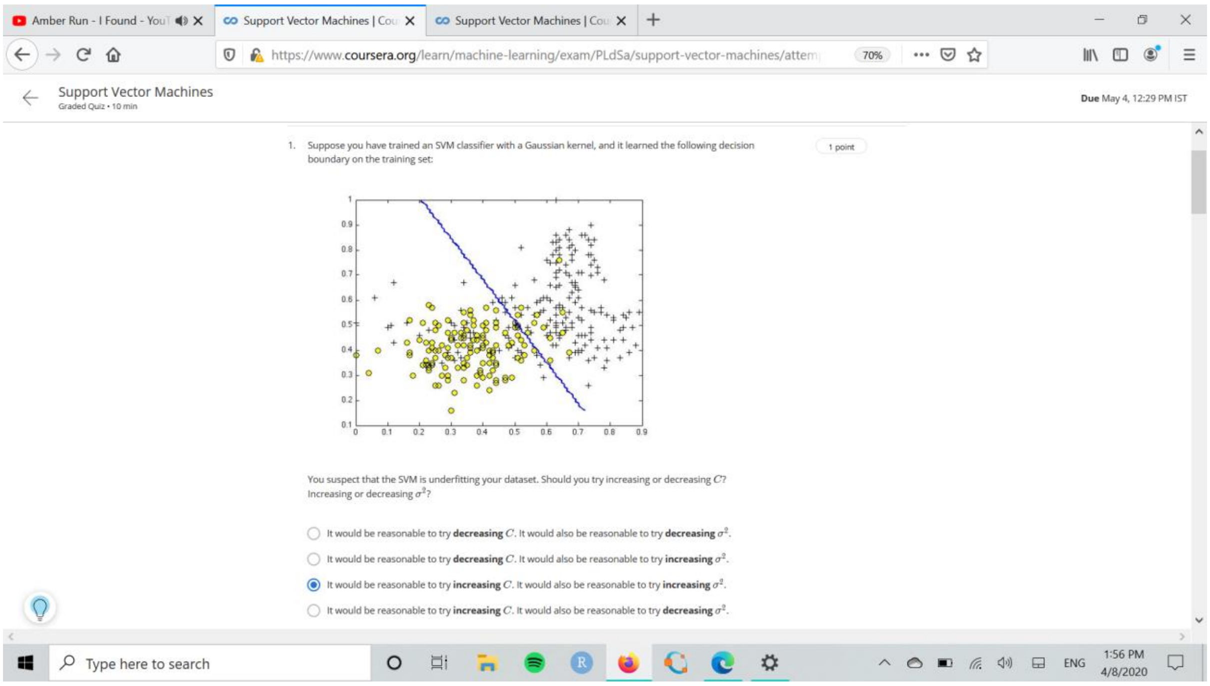
Fig. 5

Now the data is clearly linearly separable. Let the plane separating the data in higher dimension be $z=k$, where k is a constant. Since, $z=x^2+y^2$ we get $x^2 + y^2 = k$; which is an equation of a circle. So, we can project this linear separator in higher dimension back in original dimensions using this transformation.

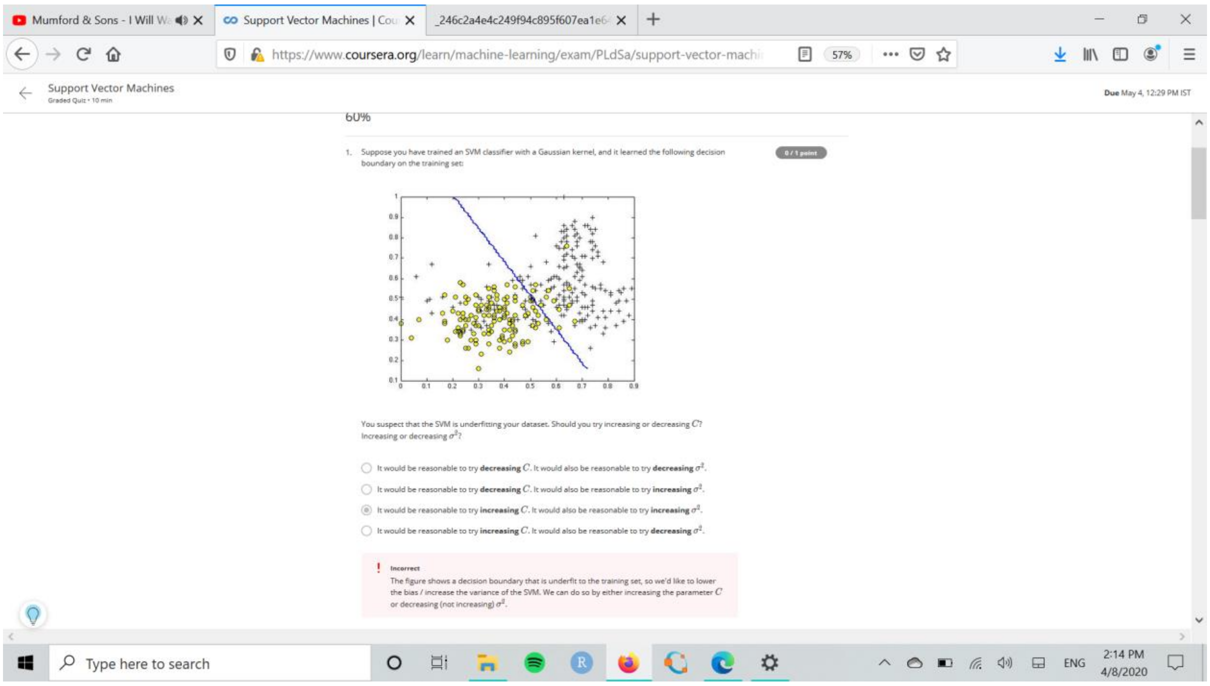


Thus we can classify data by adding an extra dimension to it so that it becomes linearly separable and then projecting the decision boundary back to original dimensions using mathematical transformation. But finding the correct transformation for any given dataset usually isn’t that easy.

QUESTIONS:



Support Vector Machines
Graded Quiz • 10 min
Due May 4, 12:29 PM IST
1. Suppose you have trained an SVM classifier with a Gaussian kernel, and it learned the following decision boundary on the training set:
1 point
You suspect that the SVM is underfitting your dataset. Should you try increasing or decreasing C? Increasing or decreasing σ^2?
It would be reasonable to try **decreasing** C. It would also be reasonable to try **decreasing** σ^2.
It would be reasonable to try **decreasing** C. It would also be reasonable to try **increasing** σ^2.
It would be reasonable to try **increasing** C. It would also be reasonable to try **increasing** σ^2.
It would be reasonable to try **increasing** C. It would also be reasonable to try **decreasing** σ^2.
Type here to search
1:56 PM
4/8/2020



Support Vector Machines
Graded Quiz • 10 min
Due May 4, 12:29 PM IST
60%
1. Suppose you have trained an SVM classifier with a Gaussian kernel, and it learned the following decision boundary on the training set:
0 / 1 point
You suspect that the SVM is underfitting your dataset. Should you try increasing or decreasing C? Increasing or decreasing σ^2?
It would be reasonable to try **decreasing** C. It would also be reasonable to try **decreasing** σ^2.
It would be reasonable to try **decreasing** C. It would also be reasonable to try **increasing** σ^2.
It would be reasonable to try **increasing** C. It would also be reasonable to try **increasing** σ^2.
It would be reasonable to try **increasing** C. It would also be reasonable to try **decreasing** σ^2.
Incorrect
The figure shows a decision boundary that is underfit to the training set, so we'd like to lower the bias / increase the variance of the SVM. We can do so by either increasing the parameter C or decreasing (not increasing) σ^2.
Type here to search
2:14 PM
4/8/2020

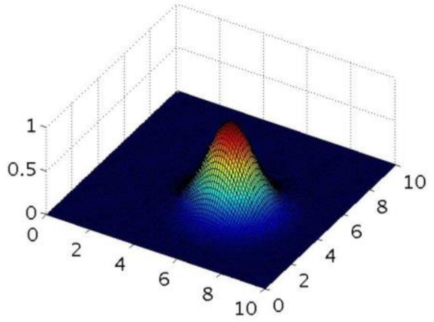
Support Vector Machines
Graded Quiz • 10 min

Due May 4, 12:29 PM IST

2. The formula for the Gaussian kernel is given by $\text{similarity}(x, l^{(1)}) = \exp(-\frac{||x-l^{(1)}||^2}{2\sigma^2})$.

1 point

The figure below shows a plot of $f_1 = \text{similarity}(x, l^{(1)})$ when $\sigma^2 = 1$.



Which of the following is a plot of f_1 when $\sigma^2 = 0.25$?

The xx - Angels (Official Au | Support Vector Machines | Cou | Lecture Slides | Coursera | _246c2a4e4c249f94c895f607ea1e6
https://www.coursera.org/learn/machine-learning/exam/PLdSa/support-vector-machines/ 40%

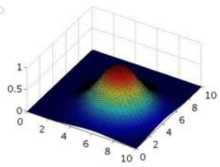


Figure 2

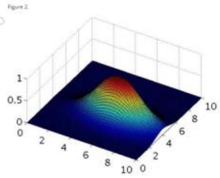
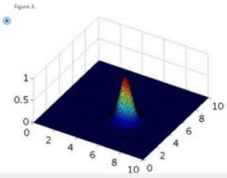


Figure 3

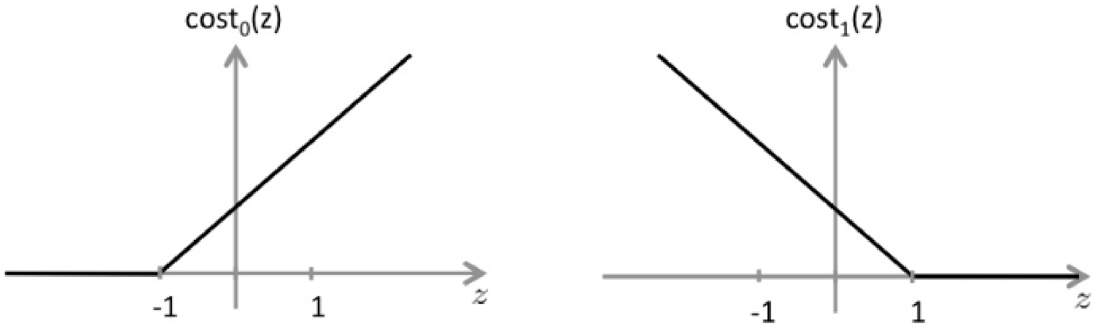


Type here to search
2:02 PM
4/8/2020

3. The SVM solves

$$\min_\theta \; C\sum_{i=1}^{m} y^{(i)}\mathrm{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\mathrm{cost}_0(\theta^T x^{(i)}) + \sum_{j=1}^{n}\theta_j^2$$

where the functions $\mathrm{cost}_0(z)$ and $\mathrm{cost}_1(z)$ look like this:



The first term in the objective is:

$$C\sum_{i=1}^{m} y^{(i)}\mathrm{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\mathrm{cost}_0(\theta^T x^{(i)}).$$

This first term will be zero if two of the following four conditions hold true. Which are the two conditions that would guarantee that this term equals zero?

- [x] For every example with $y^{(i)} = 0$, we have that $\theta^T x^{(i)} \leq -1$.
- [x] For every example with $y^{(i)} = 1$, we have that $\theta^T x^{(i)} \geq 1$.
- [ ] For every example with $y^{(i)} = 1$, we have that $\theta^T x^{(i)} \geq 0$.
- [ ] For every example with $y^{(i)} = 0$, we have that $\theta^T x^{(i)} \leq 0$.

4. Suppose you have a dataset with n = 10 features and m = 5000 examples. (1 point)

   After training your logistic regression classifier with gradient descent, you find that it has underfit the training set and does not achieve the desired performance on the training or cross validation sets.

   Which of the following might be promising steps to take? Check all that apply.

   - [ ] Use an SVM with a linear kernel, without introducing new features.
   - [x] Create / add new polynomial features.
   - [x] Use an SVM with a Gaussian Kernel.
   - [ ] Increase the regularization parameter λ.

5. Which of the following statements are true? Check all that apply. (1 point)

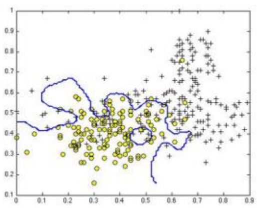
   - [x] Suppose you have 2D input examples (ie, $x^{(i)} \in \mathbb{R}^2$). The decision boundary of the SVM (with the linear kernel) is a straight line.
   - [ ] If the data are linearly separable, an SVM using a linear kernel will return the same parameters θ regardless of the chosen value of C (i.e., the resulting value of θ does not depend on C).
   - [ ] If you are training multi-class SVMs with the one-vs-all method, it is not possible to use a kernel.
   - [x] The maximum value of the Gaussian kernel (i.e., $sim(x, l^{(1)})$) is 1.

Support Vector Machines
Graded Quiz • 10 min
Due May 4, 12:29 PM IST

Support Vector Machines

TOTAL POINTS 5

1. Suppose you have trained an SVM classifier with a Gaussian kernel, and it learned the following decision boundary on the training set: (1 point)



   When you measure the SVM's performance on a cross validation set, it does poorly. Should you try increasing or decreasing C? Increasing or decreasing σ^2?

   - (x) It would be reasonable to try **decreasing** C. It would also be reasonable to try **increasing** σ^2.
   - ( ) It would be reasonable to try **increasing** C. It would also be reasonable to try **decreasing** σ^2.
   - ( ) It would be reasonable to try **increasing** C. It would also be reasonable to try **increasing** σ^2.
   - ( ) It would be reasonable to try **decreasing** C. It would also be reasonable to try **decreasing** σ^2.

Support Vector Machines
Graded Quiz • 10 min

Due May 4, 12:29 PM IST

4. Suppose you have a dataset with n = 10 features and m = 5000 examples. 1 point

After training your logistic regression classifier with gradient descent, you find that it has underfit the training set and does not achieve the desired performance on the training or cross validation sets.

Which of the following might be promising steps to take? Check all that apply.

- [x] Create / add new polynomial features.
- [ ] Increase the regularization parameter λ.
- [x] Use an SVM with a Gaussian Kernel.
- [ ] Use an SVM with a linear kernel, without introducing new features.

5. Which of the following statements are true? Check all that apply. 1 point

- [ ] Suppose you are using SVMs to do multi-class classification and would like to use the one-vs-all approach. If you have K different classes, you will train K - 1 different SVMs.
- [ ] If the data are linearly separable, an SVM using a linear kernel will return the same parameters θ regardless of the chosen value of C (i.e., the resulting value of θ does not depend on C).
- [x] The maximum value of the Gaussian kernel (i.e., $sim(x, l^{(1)})$) is 1.
- [x] It is important to perform feature normalization before using the Gaussian kernel.