Machine Learning
Support Vector Machines

Dr. Rizwan Ahmed Khan

Outline

Section Contents

Reference Books

Reference books for this Module:

- Chapter 3: Pattern Recognition, S. Theodoridis et al.,Academic Press, 4^{th} or latest edition.

Reference Books

Reference books for this Module:

- Chapter 3: Pattern Recognition, S. Theodoridis et al.,Academic Press, 4^{th} or latest edition.
- Chapter 6 & 7: Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer Books, latest edition.

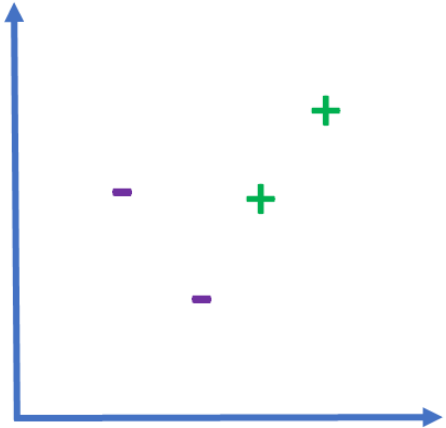
Reference Books

Reference books for this Module:

- Chapter 3: Pattern Recognition, S. Theodoridis et al.,Academic Press, 4^{th} or latest edition.
- Chapter 6 & 7: Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer Books, latest edition.
- Book Support Vector Machines Succinctly, Alexandre Kowalczyk, 2017.

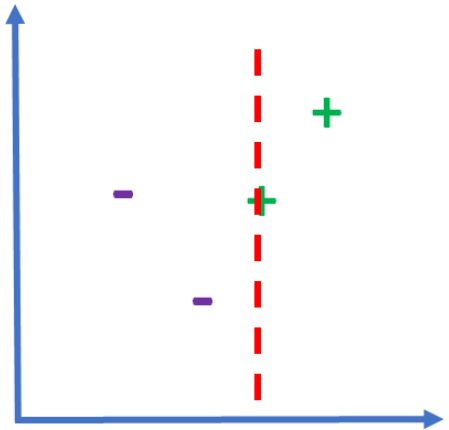
Intuition - Decision Boundary

How would you divide $+ve$ examples from $-ve$ examples?



Intuition - Decision Boundary

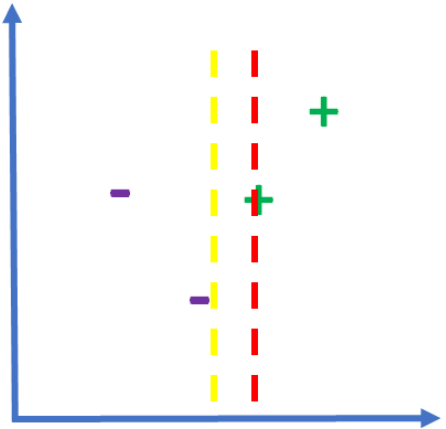
How would you divide $+ve$ examples from $-ve$ examples?



Seems infinite possibilities.

Intuition - Decision Boundary

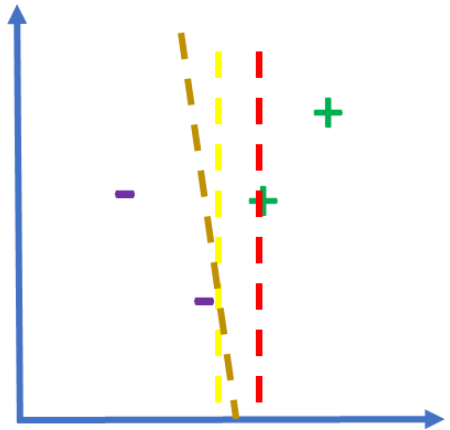
How would you divide $+ve$ examples from $-ve$ examples?



Seems infinite possibilities.

Intuition - Decision Boundary

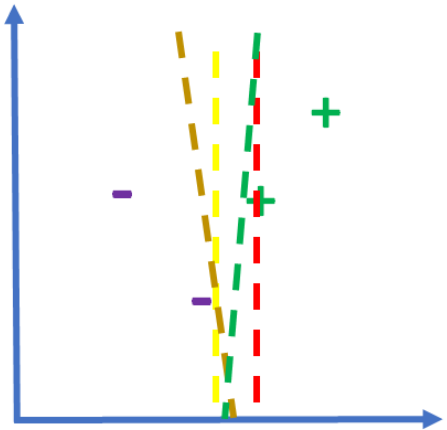
How would you divide $+ve$ examples from $-ve$ examples?



Seems infinite possibilities.

Intuition - Decision Boundary

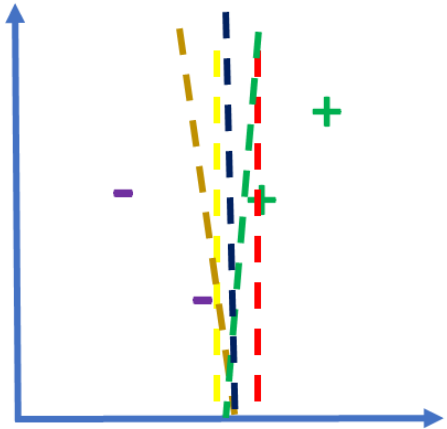
How would you divide $+ve$ examples from $-ve$ examples?



Seems infinite possibilities.

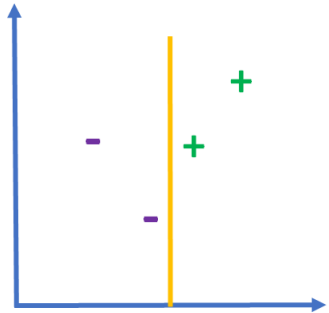
Intuition - Decision Boundary

How would you divide $+ve$ examples from $-ve$ examples?

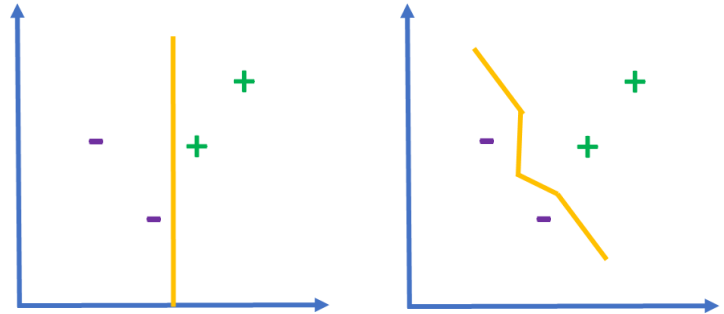


Seems infinite possibilities.

Intuition - Decision Boundary

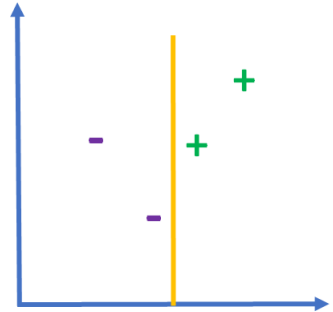


Intuition - Decision Boundary

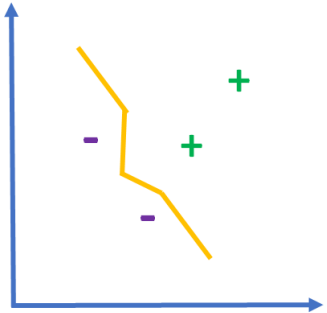


Tree / Perceptron

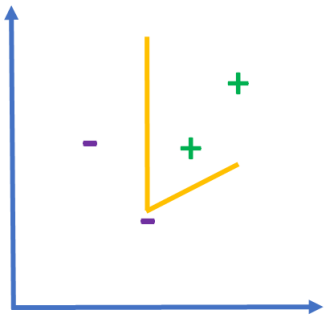
Intuition - Decision Boundary



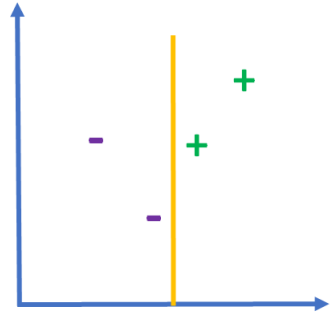
Tree / Perceptron



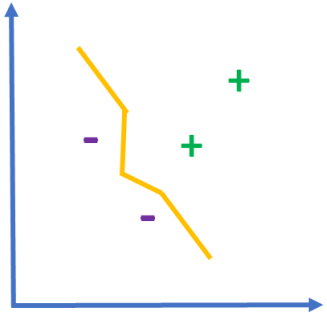
$k-$ Nearest Neighbor



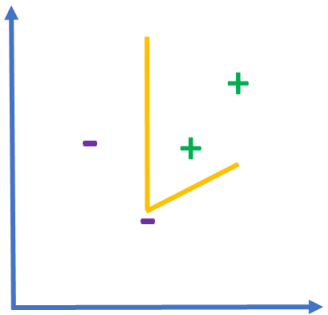
Intuition - Decision Boundary



Tree / Perceptron

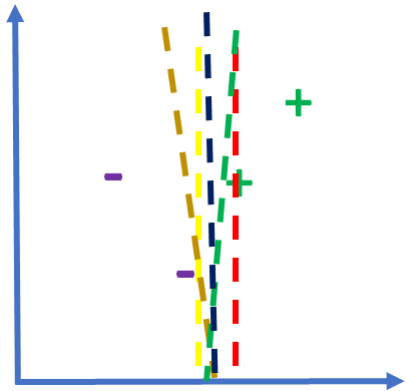


$k-$ Nearest Neighbor



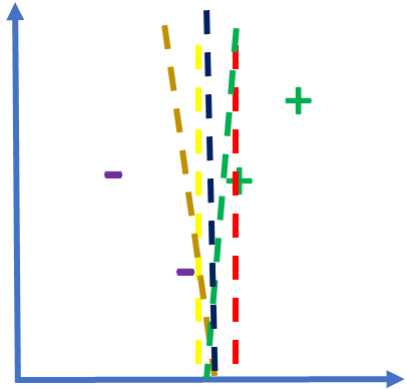
Neural Network

Goal of SVM



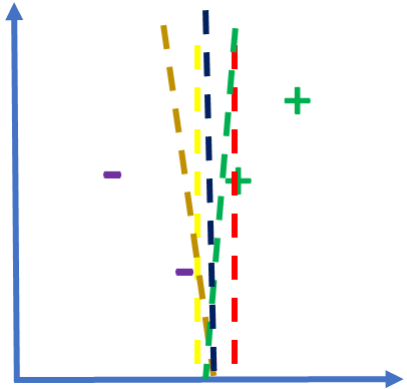
- Goal of SVM is to identify an optimal separating hyperplane which maximizes the margin between different classes of the training data.

Goal of SVM



- Goal of SVM is to identify an optimal separating hyperplane which maximizes the margin between different classes of the training data.
- SVM [a] is completely based on Mathematical Optimization problem.

Goal of SVM



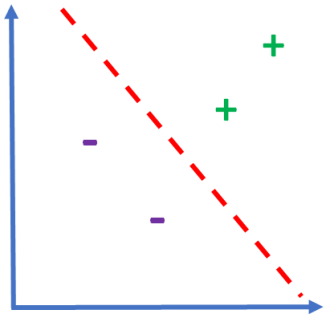
- Goal of SVM is to identify an optimal separating hyperplane which maximizes the margin between different classes of the training data.
- SVM [a] is completely based on Mathematical Optimization problem.
- SVMs are linear classifiers (a line in 2 dimensions, a plane in 3 dimensions, a $n-1$ dimensional hyperplane in n dimensions [b].

[a] Cortes, C., Vapnik, V. Support-vector networks. Machine Learning 20, 273–297 (1995). https://doi.org/10.1007/BF00994018

[b] cs276a SVM Review-Stanford University

Section Contents

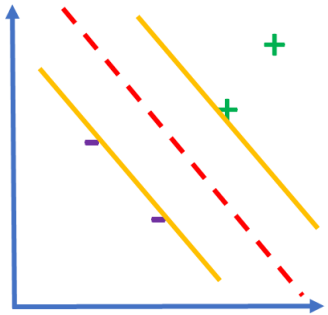
Intuition - Decision Boundary



Widest Street Approach

- Find such a line that maximizes the distance between $+ve$ examples and $-ve$ examples, while deciding decision boundary / surface.

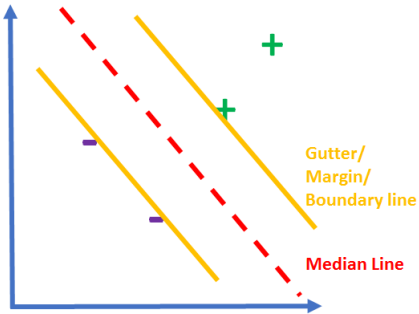
Intuition - Decision Boundary



Widest Street Approach

- Find such a line that maximizes the distance between $+ve$ examples and $-ve$ examples, while deciding decision boundary / surface.
- What would the decision rule?

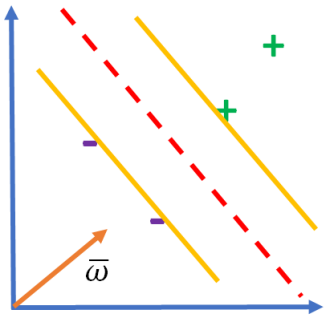
Intuition - Decision Boundary



Widest Street Approach

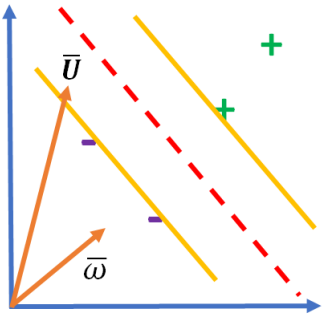
- Find such a line that maximizes the distance between $+ve$ examples and $-ve$ examples, while deciding decision boundary / surface.
- What would the decision rule?

Decision Boundary



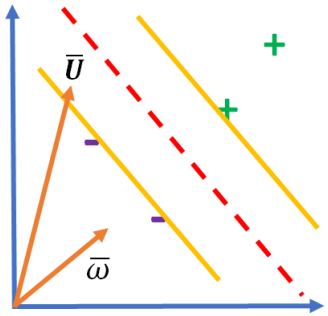
- Consider a vector $\bar{W}$ that is perpendicular to median / or gutter. We don't know anything about its length yet.

Decision Boundary



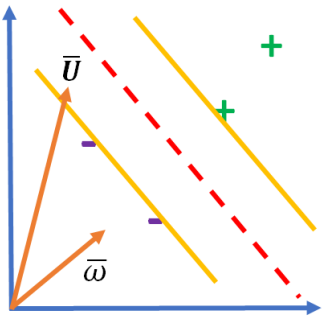
- Consider a vector $\bar{W}$ that is perpendicular to median / or gutter. We don't know anything about its length yet.
- Consider unknown point $\bar{U}$ and a vector points to it.

Decision Boundary



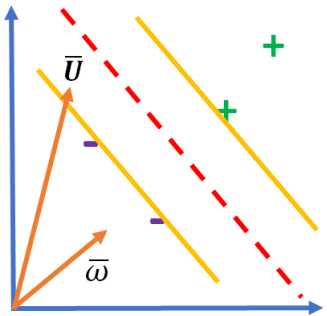
- Consider a vector $\bar{W}$ that is perpendicular to median / or gutter. We don't know anything about its length yet.
- Consider unknown point $\bar{U}$ and a vector points to it.
- We are interested to know whether this unknown is either right side of street or left or we want to know its label.
- What we can do, project that to perpendicular vector. The further we go we can find that its on the right side of the street.

Decision Boundary



- We are interested to know whether this unknown is either right side of street or left or we want to know its label.
- What we can do, project $\bar{U}$ to vector $\bar{W}$ which is perpendicular median line. The further we go, we can find that its on the right side of the street.

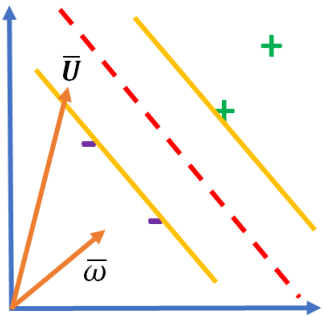
Decision Boundary



- We are interested to know whether this unknown is either right side of street or left or we want to know its label.
- What we can do, project $\bar{U}$ to vector $\bar{W}$ which is perpendicular median line. The further we go, we can find that its on the right side of the street.

$$\bar{W} \cdot \bar{U} \geq C$$

Decision Boundary

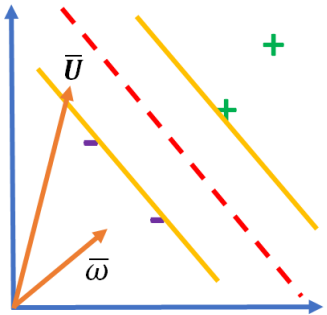


- We are interested to know whether this unknown is either right side of street or left or we want to know its label.
- What we can do, project $\bar{U}$ to vector $\bar{W}$ which is perpendicular median line. The further we go, we can find that its on the right side of the street.

$$\bar{W} \cdot \bar{U} \geq C$$

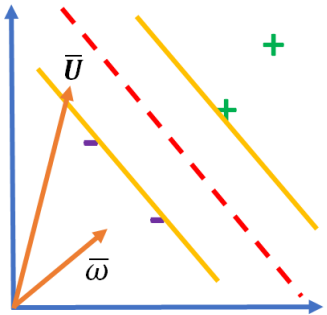
- Dot product is projecting onto $\bar{W}$. The bigger the projection is then it will cross median line and then unknown vector can be labeled as $+ve$ sample.

Decision Boundary



- Then, without loss of generality we can say:

Decision Boundary

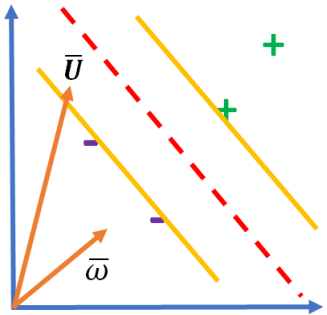


- Then, without loss of generality we can say:

$$\bar{W} \cdot \bar{U} + b \geq 0 \quad \text{THEN } +\text{ve} \tag{1}$$

This is Decision Rule.

Decision Boundary



- Then, without loss of generality we can say:

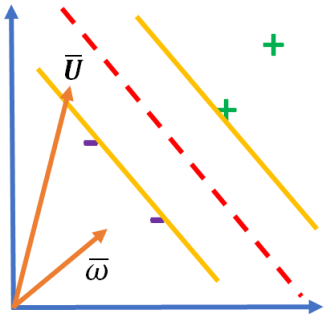
$$\bar{W} \cdot \bar{U} + b \geq 0 \quad \text{THEN } +\text{ve} \tag{1}$$

This is Decision Rule.

- We don't know (yet) what constant b, $C = -b$, to use and what $\bar{W}$ to use either.

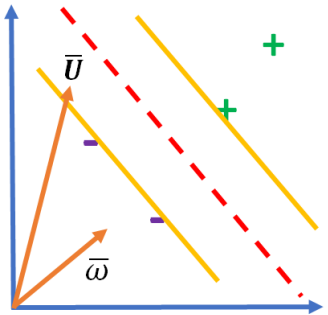
Section Contents

Constraints



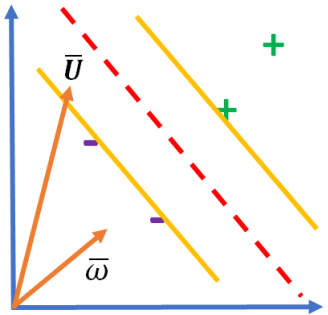
- We just know that $\bar{W}$ needs to be perpendicular to the median line of the street.

Constraints



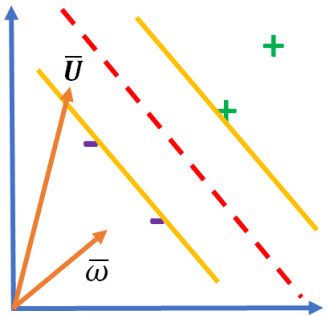
- We just know that $\bar{W}$ needs to be perpendicular to the median line of the street.
- But then there many $\bar{W}_s$ perpendicular to the median line of the street, any length is not fixed yet.
- What should we do?

Constraints



- We just know that $\bar{W}$ needs to be perpendicular to the median line of the street.
- But then there many $\bar{W}_s$ perpendicular to the median line of the street, any length is not fixed yet.
- What should we do?
- So we need to put constraints to find particular $\bar{W}$ and b that maximizes width of the street (separation between $+_s$ and $-_s$).

Constraints



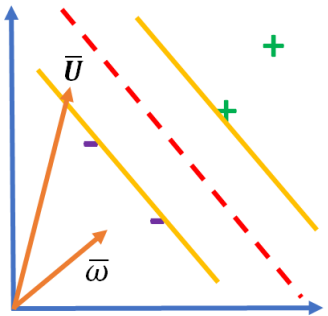
- Putting constraints to calculate $\bar{W}$ and b.

$$\bar{W} \cdot \bar{X_+} + b \geq 1 \quad \{\text{for +ve samples}\} \tag{2}$$

$$\bar{W} \cdot \bar{X_-} + b \leq -1 \quad \{\text{for -ve samples}\} \tag{3}$$

[1] Did you see similarity with "Perceptron" decision rule?

Constraints



$*1$

- Putting constraints to calculate $\bar{W}$ and b.

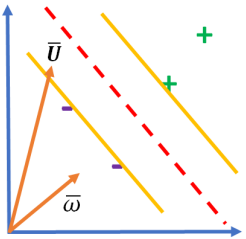
$$\bar{W} \cdot \bar{X_+} + b \geq 1 \quad \{\text{for } +\text{ve samples}\} \tag{2}$$

$$\bar{W} \cdot \bar{X_-} + b \leq -1 \quad \{\text{for -ve samples}\} \tag{3}$$

So imposing separation of -1 to +1 for $-ve$ and $+ve$ samples (maximizing margin).

[1] Did you see similarity with "Perceptron" decision rule?

Constraints



- Equation 2 and 3 can be written / combined as:

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1 \qquad (4)$$

where:
- y_i is $+1$ for $+ve$ samples.
- y_i is -1 for $-ve$ samples.

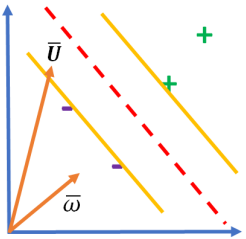
Proof:

1. for $+ve$ samples:

$$1 \times (\bar{W} \cdot \bar{X}_i + b) \geq 1$$
$$\Rightarrow (\bar{W} \cdot \bar{X}_i + b) \geq 1 \qquad (5)$$

Same as Eq. 2.

Constraints



- Equation 2 and 3 can be written / combined as:

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1 \qquad (4)$$

where:
- y_i is $+1$ for $+ve$ samples.
- y_i is -1 for $-ve$ samples.

Proof:

1. for $+ve$ samples:

$$\begin{aligned} 1 \times (\bar{W} \cdot \bar{X}_i + b) &\geq 1 \\ \Rightarrow (\bar{W} \cdot \bar{X}_i + b) &\geq 1 \end{aligned} \qquad (5)$$

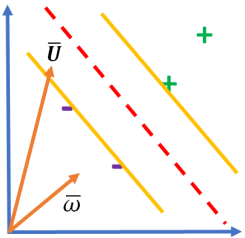
Same as Eq. 2.

2. for $-ve$ samples:

$$\begin{aligned} -1 \times (\bar{W} \cdot \bar{X}_i + b) &\geq 1 \\ \Rightarrow -\bar{W} \cdot \bar{X}_i - b &\geq 1 \\ \Rightarrow \bar{W} \cdot \bar{X}_i + b &\leq -1 \end{aligned} \qquad (6)$$

Same as Eq. 3.

Constraints



Back to equation 4:

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1$$

where:
- y_i is $+1$ for $+ve$ samples.
- y_i is -1 for $-ve$ samples.

Equation 4 can be written as:

$$\begin{aligned} y_i(\bar{W} \cdot \bar{X}_i + b) &\geq 1 \\ y_i(\bar{W} \cdot \bar{X}_i + b) - 1 &\geq 0 \end{aligned} \tag{7}$$

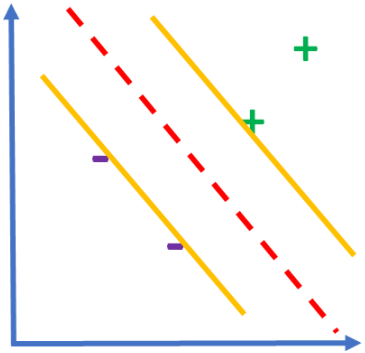
Additional constraint:

$$y_i(\bar{W} \cdot \bar{X}_i + b) - 1 = 0$$

{for samples (X_i) in gutter or at boundary / margin}

(8)

Samples on the boundary / gutter

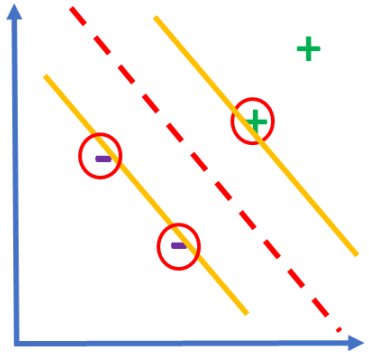


Additional constraint:

$$y_i(\bar{W} \cdot \bar{X}_i + b) - 1 = 0$$

{for samples (X_i) in gutter or at boundary / margin}

Samples on the boundary / gutter



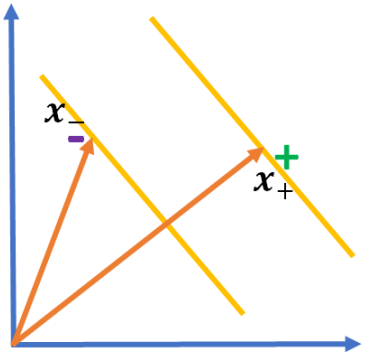
Additional constraint:

$$y_i(\bar{W} \cdot \bar{X}_i + b) - 1 = 0$$

{for samples (X_i) in gutter or at boundary / margin}

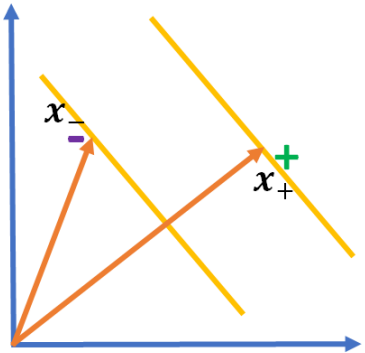
These samples are also called as Support Vectors.

Distance b/w boundary lines / margin / gutters



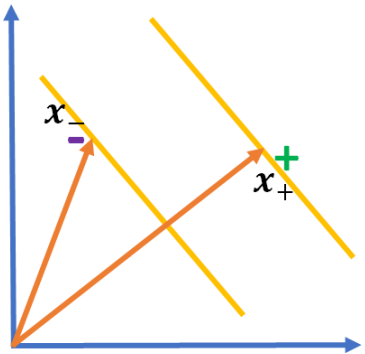
- We are trying to arrange line $\bar{W}$ and b in a such a way that it maximizes width of the street (separation between $+_s$ and $-_s$).

Distance b/w boundary lines / margin / gutters



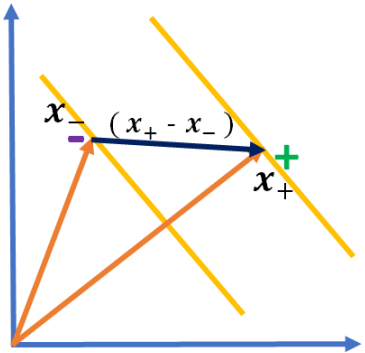
- We are trying to arrange line $\bar{W}$ and b in a such a way that it maximizes width of the street (separation between $+_s$ and $-_s$).
- Boundary lines are parallel to one another, we can pick points on these lines to define width of the street.

Distance b/w boundary lines / margin / gutters



- We are trying to arrange line $\bar{W}$ and b in a such a way that it maximizes width of the street (separation between $+_s$ and $-_s$).
- Boundary lines are parallel to one another, we can pick points on these lines to define width of the street.
- Width of the street is distance b/w the gutters / boundary lines.

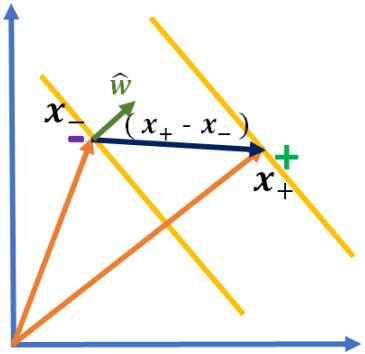
Distance b/w boundary lines / margin / gutters



- We are trying to arrange line $\bar{W}$ and b in a such a way that it maximizes width of the street (separation between $+_s$ and $-_s$).
- Boundary lines are parallel to one another, we can pick points on these lines to define width of the street.
- Width of the street is distance b/w the gutters / boundary lines.
- Difference of two vectors can give us width of the street:

$$(\bar{X_+} - \bar{X_-})$$

Distance b/w boundary lines / margin / gutters

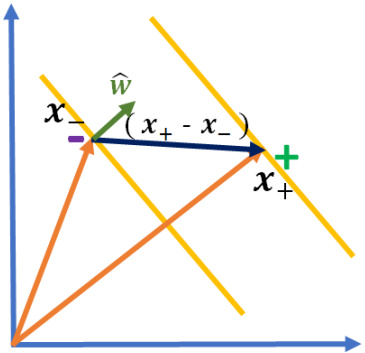


- Width of street:

$$= (\bar{X_+} - \bar{X_-}) \cdot \frac{\bar{W}}{||W||} \tag{9}$$

- Where $\frac{\bar{W}}{||W||}$ is a unit vector ($\hat{W}$) perpendicular / normal to gutter or boundary line.

Distance b/w boundary lines / margin / gutters



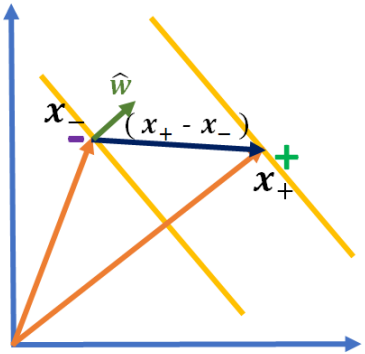
- Width of street:

$$= (\bar{X_+} - \bar{X_-}) \cdot \frac{\bar{W}}{||W||} \tag{9}$$

- Where $\frac{\bar{W}}{||W||}$ is a unit vector ($\hat{W}$) perpendicular / normal to gutter or boundary line.

- In other words, projection of difference vector on to unit vector ($\hat{W}$) will be width of the street (difference in the direction of $\bar{W}$ vector).

Distance b/w boundary lines / margin / gutters



- Width of street:

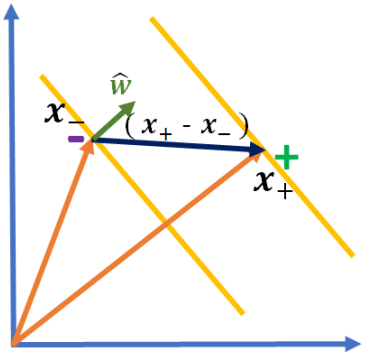
$$= (\bar{X_+} - \bar{X_-}) \cdot \frac{\bar{W}}{||W||} \tag{9}$$

- Where $\frac{\bar{W}}{||W||}$ is a unit vector ($\hat{W}$) perpendicular / normal to gutter or boundary line.

- In other words, projection of difference vector on to unit vector ($\hat{W}$) will be width of the street (difference in the direction of $\bar{W}$ vector).

- It's a dot product, so its scalar, width of the street.

Distance b/w boundary lines / margin / gutters



Width of street:

$$= (\bar{X_+} - \bar{X_-}) \cdot \frac{\bar{W}}{||W||}$$

From Equation 8 we know that, samples in a gutter (enforcing constraint) =

$$y_i(\bar{W} \cdot \bar{X}_i + b) - 1 = 0$$

So,

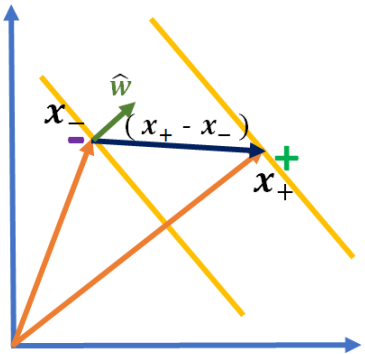
- for $+ve$ sample:

$$\bar{W} \cdot \bar{X}_i = 1 - b \tag{10}$$

- for $-ve$ sample:

$$\begin{aligned} -\bar{W} \cdot \bar{X}_i - b - 1 &= 0 \\ -\bar{W} \cdot \bar{X}_i &= 1 + b \end{aligned} \tag{11}$$

Distance b/w boundary lines / margin / gutters



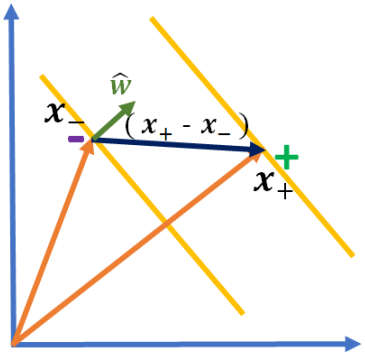
Width of street:

$$(\bar{X_+} - \bar{X_-}) \cdot \frac{\bar{W}}{||W||}$$
$$\bar{W} \cdot \bar{X_+} - \bar{W} \cdot \bar{X_-} \cdot \frac{1}{||W||} \tag{12}$$

Putting back values from Equations 10 and 11 into Equation 12.

$$\text{Width} = \frac{2}{||W||} \tag{13}$$

Distance b/w gutter



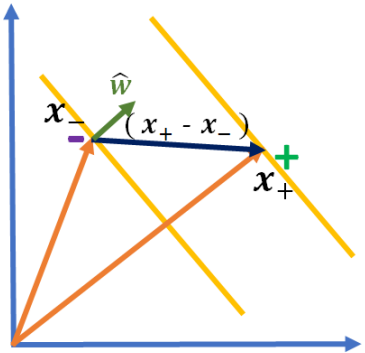
Width of street:

$$\text{Width} = \frac{2}{||W||}$$

- SVM tries to maximize this width, to have maximum possible separation between samples of different classes.

$$\text{Width} = \text{Max}\frac{2}{||W||} \implies \text{Min}||W|| \implies \text{Min}\frac{1}{2}||W||^2$$

Distance b/w gutter



Width of street:

$$\text{Width} = \frac{2}{||W||}$$

- SVM tries to maximize this width, to have maximum possible separation between samples of different classes.

$$\text{Width} = \text{Max}\frac{2}{||W||} \implies \text{Min}||W|| \implies \text{Min}\frac{1}{2}||W||^2$$

$$\text{Width} = \text{Min}\frac{1}{2}||W||^2 \tag{14}$$

Summary

Stages in the development:

1. Decision Rule:
$$\bar{W} \cdot \bar{U} + b \geq 0 \quad \text{THEN +ve}$$
2. Projection of W on U and put a constraint then it should be $\geq +1$ for $+ve$ samples and ≤ -1 for $-ve$ samples.
$$\bar{W} \cdot \bar{X_+} + b \geq 1 \quad \{\text{for +ve samples}\}$$
$$\bar{W} \cdot \bar{X_-} + b \leq -1 \quad \{\text{for -ve samples}\}$$
3. Additional constraint for samples in gutter
$$y_i(\bar{W} \cdot \bar{X}_i + b) - 1 = 0$$
4. Then we discovered we wish to maximize / minimize this expression:
$$\text{Width} = \text{Min}\frac{1}{2}||W||^2$$

What's next?

- We have now transformed the problem into a form that can be efficiently solved.

$$\text{Width} = \text{Min}\frac{1}{2}||W||^2$$

- The above is an optimization problem with a convex quadratic objective and some constraints. Its solution gives us the optimal margin classifier.

What's next?

- We have now transformed the problem into a form that can be efficiently solved.

$$\text{Width} = \text{Min}\frac{1}{2}||W||^2$$

- The above is an optimization problem with a convex quadratic objective and some constraints. Its solution gives us the optimal margin classifier.
- How can we go forward (QUIZ):

What's next?

- We have now transformed the problem into a form that can be efficiently solved.

$$\text{Width} = \text{Min}\frac{1}{2}||W||^2$$

- The above is an optimization problem with a convex quadratic objective and some constraints. Its solution gives us the optimal margin classifier.
- How can we go forward (QUIZ):
  1. Laplace

What's next?

- We have now transformed the problem into a form that can be efficiently solved.

$$\text{Width} = \text{Min}\frac{1}{2}||W||^2$$

- The above is an optimization problem with a convex quadratic objective and some constraints. Its solution gives us the optimal margin classifier.
- How can we go forward (QUIZ):
  1. Laplace
  2. Legendre

What's next?

- We have now transformed the problem into a form that can be efficiently solved.

$$\text{Width} = \text{Min}\frac{1}{2}||W||^2$$

- The above is an optimization problem with a convex quadratic objective and some constraints. Its solution gives us the optimal margin classifier.
- How can we go forward (QUIZ):
  1. Laplace
  2. Legendre
  3. Lagrange

Section Contents

Lagrange Multiplier

- Lagrange multipliers provides a way for finding extremum of a function subject to equality constraints i.e., subject to the condition that one or more equations have to be satisfied exactly by the chosen values of the variables.
- The great advantage of this method is that it allows the optimization to be solved without explicit parameterization in terms of the constraints.
- Method can be summarized as follows: in order to find the stationary points of a function $f(x)$ subjected to the equality constraint $g(x) = 0$, form the Lagrangian function:

$$L(x, \lambda) = f(x) - \lambda g(x) \tag{15}$$

where λ = Lagrange multiplier

2

[2] Refer Section 8 to see disscussion on intuition of Lagrange multiplier

Lagrange Multiplier

- Taking Equation 15 and writing function that we are trying to find extremum.

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i(\text{write down constraints})$$

Lagrange Multiplier

- Taking Equation 15 and writing function that we are trying to find extremum.

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i(\text{write down constraints})$$

- Constraint is given in Equation 8

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right] \tag{16}$$

where α_i = Lagrange multiplier
α_i will be non-zero for vectors connected with samples in gutter, otherwise it will be zero.

Find Extremum

- What needs to be done to find extremum of Equation 16 ?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

Find Extremum

- What needs to be done to find extremum of Equation 16 ?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

- Take derivative and set it equal to zero.

Find Extremum

- What needs to be done to find extremum of Equation 16 ?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

- Take derivative and set it equal to zero.

- **Take derivative w.r.t. "W":**

Find Extremum

- What needs to be done to find extremum of Equation 16 ?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

- Take derivative and set it equal to zero.

- **Take derivative w.r.t. "W":**

$$\frac{\partial L}{\partial W} = \bar{W} - \sum_i \alpha_i y_i \bar{X}_i = 0 \Longrightarrow W = \sum_i \alpha_i y_i \bar{X}_i \tag{17}$$

Where α_i is a scalar, y_i is +1 or -1 and X_i is sample vector. **What this equation signifies?**

Find Extremum

- What needs to be done to find extremum of Equation 16 ?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

- Take derivative and set it equal to zero.

- **Take derivative w.r.t. "W":**

$$\frac{\partial L}{\partial W} = \bar{W} - \sum_i \alpha_i y_i \bar{X}_i = 0 \Longrightarrow W = \sum_i \alpha_i y_i \bar{X}_i \tag{17}$$

Where α_i is a scalar, y_i is +1 or -1 and X_i is sample vector. **What this equation signifies?**

What this equation signifies

Decision vector W is linear sum of **some** samples. Some, in the sense that α_i will be non-zero for few vectors (connected with samples in gutter).

Find Extremum

- Back to Equation 16. Any other variable that may vary?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

Find Extremum

- Back to Equation 16. Any other variable that may vary?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

- **Take derivative w.r.t. "b":**

Find Extremum

- Back to Equation 16. Any other variable that may vary?

$$L = \frac{1}{2}||W||^2 - \sum \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

- **Take derivative w.r.t. "b":**

$$\frac{\partial L}{\partial b} = -\sum_i \alpha_i y_i = 0 \Longrightarrow \sum_i \alpha_i y_i = 0 \tag{18}$$

Find Extremum

Plug back value of W from Equation 17 to Equation 16.

$$W = \sum_i \alpha_i y_i \bar{X}_i$$

$$L = \frac{1}{2}||W||^2 - \sum_i \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

Find Extremum

Plug back value of W from Equation 17 to Equation 16.

$$W = \sum_i \alpha_i y_i \bar{X}_i$$

$$L = \frac{1}{2}||W||^2 - \sum_i \alpha_i \left[y_i(\bar{W} \cdot \bar{X}_i + b) - 1\right]$$

$$\begin{aligned} L = \frac{1}{2}(\sum_i \alpha_i y_i \bar{X}_i) \cdot (\sum_j \alpha_j y_j \bar{X}_j) \\ -(\sum_i \alpha_i y_i \bar{X}_i) \cdot (\sum_j \alpha_j y_j \bar{X}_j) \\ -\sum_i \alpha_i y_i b + \sum_i \alpha_i \end{aligned} \tag{19}$$

Find Extremum

Term shown in red in Equation 19 = 0 , refer Equation 18

Now arrange and re-write Lagrangian Equation 19

Find Extremum

Term shown in red in Equation 19 $= 0$, refer Equation 18

Now arrange and re-write Lagrangian Equation 19

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j \tag{20}$$

Find Extremum

Term shown in red in Equation 19 $= 0$, refer Equation 18

Now arrange and re-write Lagrangian Equation 19

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j \tag{20}$$

What this equation signifies

This optimization / finding extremum of a function depends only on dot products of pairs of samples (dual problem).

Decision Rule

Recall Equation 1, related to decision rule:

$$\bar{W} \cdot \bar{U} + b \geq 0 \quad \text{THEN } +\text{ve}$$

Now, we can update this Equation, with derived value of $\bar{W}$, refer Equation 17:

Decision Rule

Recall Equation 1, related to decision rule:

$$\bar{W} \cdot \bar{U} + b \geq 0 \quad \text{THEN +ve}$$

Now, we can update this Equation, with derived value of $\bar{W}$, refer Equation 17:

$$\sum_i \alpha_i y_i \bar{X}_i \cdot \bar{U} + b \geq 0 \quad \text{THEN +ve} \tag{21}$$

Decision Rule

Recall Equation 1, related to decision rule:

$$\bar{W} \cdot \bar{U} + b \geq 0 \quad \text{THEN +ve}$$

Now, we can update this Equation, with derived value of $\bar{W}$, refer Equation 17:

$$\sum_i \alpha_i y_i \bar{X}_i \cdot \bar{U} + b \geq 0 \quad \text{THEN +ve} \tag{21}$$

What this equation signifies?

Decision rule depends again only on dot product of unknown vector and sample vector.

Summary

Summary

- Recall Equation 13, Width $= \text{Max} \frac{2}{||W||}$, while satisfying constraints, given by: classify training examples correctly $y_i(\bar{W} \cdot \bar{X}_i + b) - 1 \geq 0, \forall i$.

Summary

Summary

- Recall Equation 13, Width $= \text{Max} \frac{2}{||W||}$, while satisfying constraints, given by: classify training examples correctly $y_i(\bar{W} \cdot \bar{X}_i + b) - 1 \geq 0, \forall i$.
- Then, we transformed above equation into equivalent problem (which is easier to solve), Equation 14, Width $= \text{Min} \frac{1}{2}||W||^2$.
- Why its easier?

Summary

Summary

- Recall Equation 13, Width = Max $\frac{2}{||W||}$, while satisfying constraints, given by: classify training examples correctly $y_i(\bar{W} \cdot \bar{X}_i + b) - 1 \geq 0, \forall i$.
- Then, we transformed above equation into equivalent problem (which is easier to solve), Equation 14, Width = Min $\frac{1}{2}||W||^2$.
- Why its easier?
- Actually, this is easier to solve as when we have optimization problem in the form given above while satisfying constraints, is called quadratic programming (QP) problem. QP is well known field and it's solution is easier to find.

Summary

Summary

- Recall Equation 13, Width = Max $\frac{2}{||W||}$, while satisfying constraints, given by: classify training examples correctly $y_i(\bar{W} \cdot \bar{X}_i + b) - 1 \geq 0, \forall i$.
- Then, we transformed above equation into equivalent problem (which is easier to solve), Equation 14, Width = Min $\frac{1}{2}||W||^2$.
- Why its easier?
- Actually, this is easier to solve as when we have optimization problem in the form given above while satisfying constraints, is called quadratic programming (QP) problem. QP is well known field and it's solution is easier to find.
- Optimization problems of this form have convex function and thus unique solution is always guaranteed.

Summary

Summary

- Recall Equation 13, Width = Max $\frac{2}{||W||}$, while satisfying constraints, given by: classify training examples correctly $y_i(\bar{W} \cdot \bar{X}_i + b) - 1 \geq 0, \forall i$.
- Then, we transformed above equation into equivalent problem (which is easier to solve), Equation 14, Width = Min $\frac{1}{2}||W||^2$.
- Why its easier?
- Actually, this is easier to solve as when we have optimization problem in the form given above while satisfying constraints, is called quadratic programming (QP) problem. QP is well known field and it's solution is easier to find.
- Optimization problems of this form have convex function and thus unique solution is always guaranteed.
- Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$

Summary

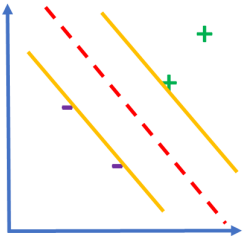
Summary

- Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$
  While, $W = \sum_i \alpha_i y_i \bar{X}_i$ and
  $\sum_i \alpha_i y_i = 0$

Summary

Summary

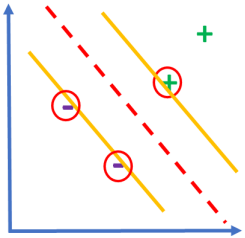
- Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$
  While, $W = \sum_i \alpha_i y_i \bar{X}_i$ and
  $\sum_i \alpha_i y_i = 0$



It turns out most of α_i are zeros, which implies that only few vectors (with non-zero α_i) matters in finding solution / decision boundary while most of vectors do not. Thus, building a **machine** with few **support vectors** (with non-zero α_i).

Summary

- Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$
  While, $W = \sum_i \alpha_i y_i \bar{X}_i$ and
  $\sum_i \alpha_i y_i = 0$



It turns out most of α_i are zeros, which implies that only few vectors (with non-zero α_i) matters in finding solution / decision boundary while most of vectors do not. Thus, building a **machine** with few **support vectors** (with non-zero α_i).

Summary

Summary

- Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$
  While, $W = \sum_i \alpha_i y_i \bar{X}_i$ and
  $\sum_i \alpha_i y_i = 0$
- This optimization / finding extremum of a function depends only on dot products of pairs of samples , $(\bar{X}_i^T \cdot \bar{X}_j)$. Note[a].
- Analyzing : $\bar{X}_i^T \cdot \bar{X}_j$, What it actually means?

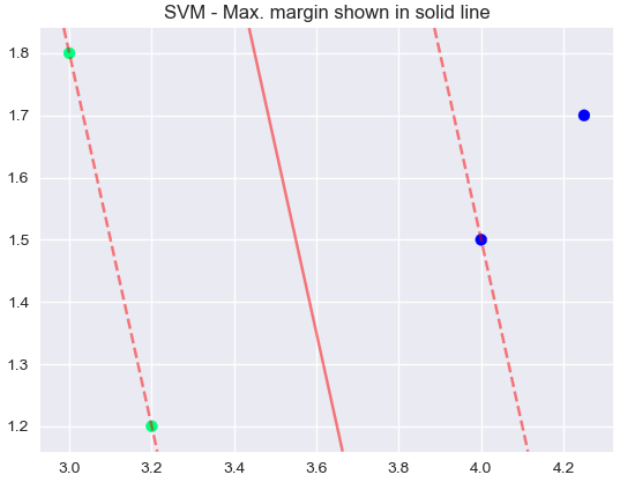
Summary

Summary

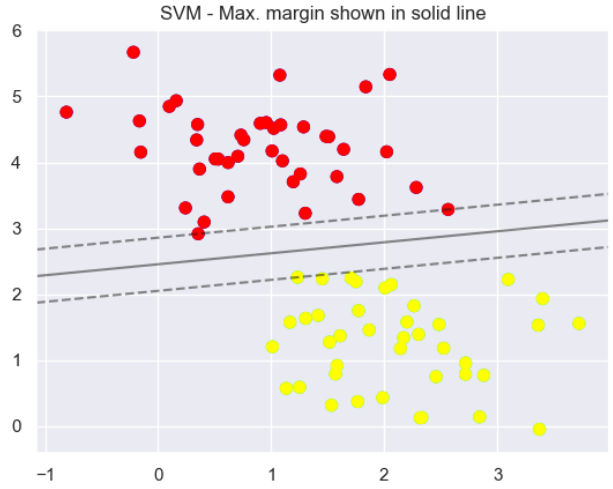
- Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$
  While, $W = \sum_i \alpha_i y_i \bar{X}_i$ and
  $\sum_i \alpha_i y_i = 0$
- This optimization / finding extremum of a function depends only on dot products of pairs of samples , $(\bar{X}_i^T \cdot \bar{X}_j)$. Note[a].
- Analyzing : $\bar{X}_i^T \cdot \bar{X}_j$, What it actually means?
  1. Its a dot product, projection of one on another.
  2. It is a **measure of similarity** between two non-zero vectors. If vectors are orthogonal value will be zero, and if vector in opposite direction value will be $-ve$.

[a] Transpose is used to make matrix dimensions compatible

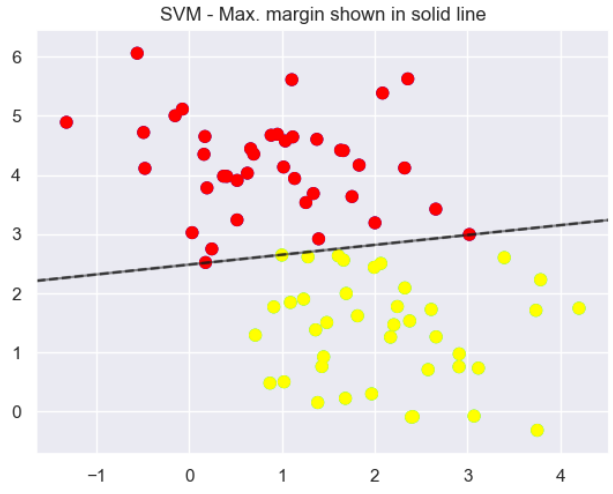
Ocular proof



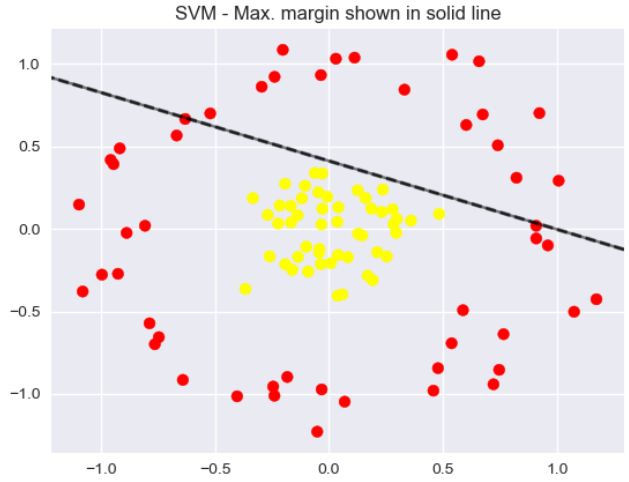
Ocular proof



Ocular proof



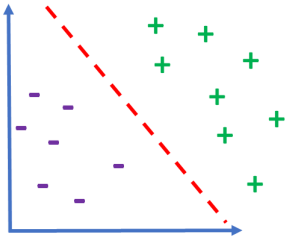
Ocular proof



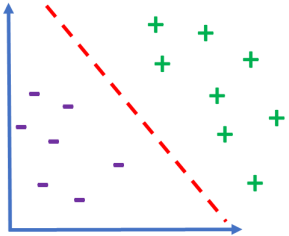
What to do here? Data is not linearly separable!

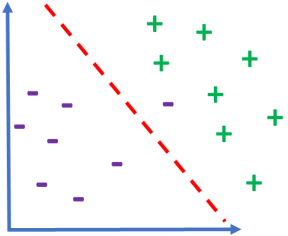
Section Contents

Non-linearly separable

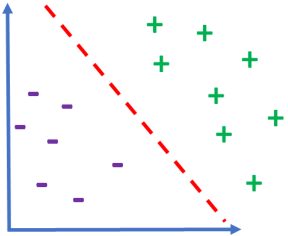


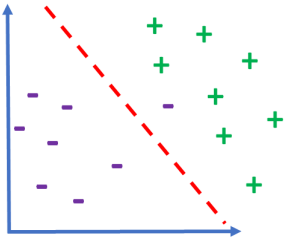
Non-linearly separable





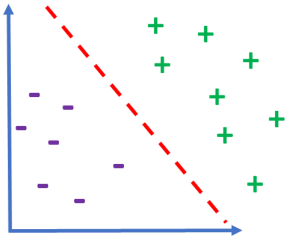
Non-linearly separable

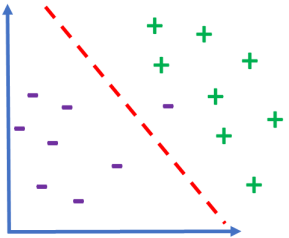




- What to do, since SVM find linear classification boundary? SVMs are linear classifiers (a line in 2 dimensions, a plane in 3 dimensions, a $n-1$ dimensional hyperplane in n dimensions.
- Some probabilities:

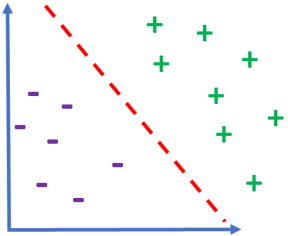
Non-linearly separable

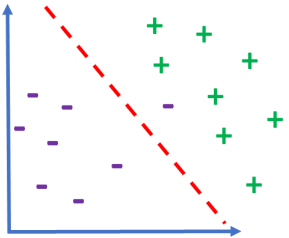




- What to do, since SVM find linear classification boundary? SVMs are linear classifiers (a line in 2 dimensions, a plane in 3 dimensions, a $n-1$ dimensional hyperplane in n dimensions.
- Some probabilities:
  1. Probably its a outlier!

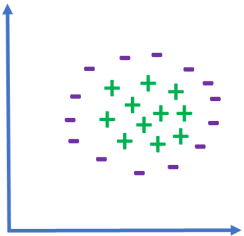
Non-linearly separable



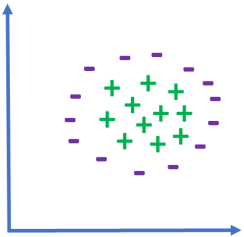


- What to do, since SVM find linear classification boundary? SVMs are linear classifiers (a line in 2 dimensions, a plane in 3 dimensions, a $n-1$ dimensional hyperplane in n dimensions.
- Some probabilities:
  1. Probably its a outlier!
  2. or find linear decision boundary while minimizing some cost function that penalizes for training error.

Non-linearly separable

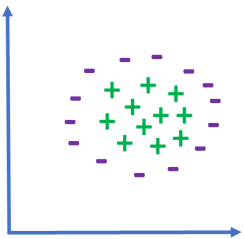


Non-linearly separable



- In previous example, one example seemed outlier and solution was proposed, but in this case it is impossible to come up with linear classifier.

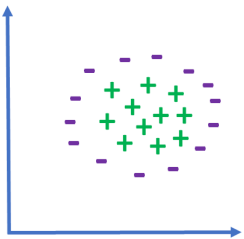
Non-linearly separable



- In previous example, one example seemed outlier and solution was proposed, but in this case it is impossible to come up with linear classifier.

- As it seems impossible to use SVM / linear classifier, should we just remove SVM from machine learning toolkit?

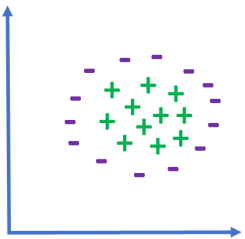
Non-linearly separable



- In previous example, one example seemed outlier and solution was proposed, but in this case it is impossible to come up with linear classifier.

- As it seems impossible to use SVM / linear classifier, should we just remove SVM from machine learning toolkit?
- No, there is a little trick that can be done to transform data (change the data point without changing the data point) in a such away that it become linearly separable.

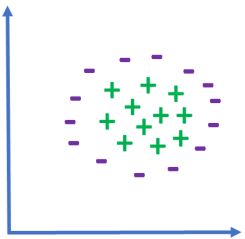
Non-linearly separable



- In previous example, one example seemed outlier and solution was proposed, but in this case it is impossible to come up with linear classifier.

- As it seems impossible to use SVM / linear classifier, should we just remove SVM from machine learning toolkit?
- No, there is a little trick that can be done to transform data (change the data point without changing the data point) in a such away that it become linearly separable.
- Define function Φ that will take data point and change its dimension (in our example there are two dimension).

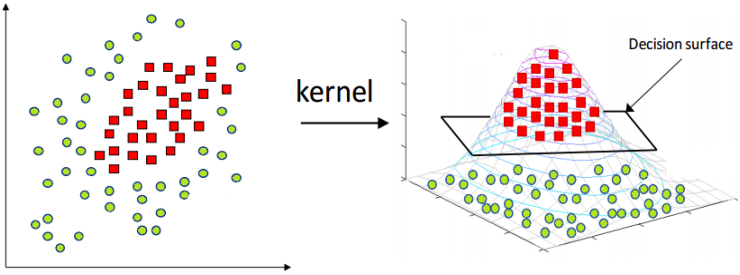
Non-linearly separable



- In previous example, one example seemed outlier and solution was proposed, but in this case it is impossible to come up with linear classifier.

- As it seems impossible to use SVM / linear classifier, should we just remove SVM from machine learning toolkit?
- No, there is a little trick that can be done to transform data (change the data point without changing the data point) in a such away that it become linearly separable.
- Define function Φ that will take data point and change its dimension (in our example there are two dimension).
- For example, we can transform data from our example in three dimensions using:
  $\Phi(x) =< x_1^2, x_2^2, \sqrt{2}x_1x_2 >$
  where: x_1 and x_2 are dimension of same vector

Intuition

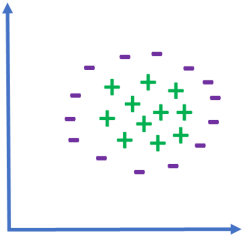


Video showing XOR data from 2D to 3D. [3]

- There is a little trick that can be done to transform data (change the data point without changing the data point) in a such away that it become linearly separable.
- Define function Φ that will take data point and change its dimension.

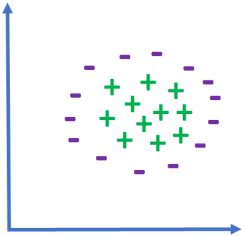
[3] https://youtu.be/5KIYu3zKvqo

Non-linearly separable



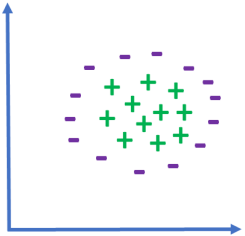
- $\Phi(x) =< x_1^2, x_2^2, \sqrt{2}x_1x_2 >$
- There isn't any new information!

Non-linearly separable



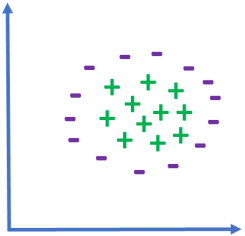
- $\Phi(x) =< x_1^2, x_2^2, \sqrt{2}x_1x_2 >$
- There isn't any new information!
- Reminder: Quadratic programming problem form:
  $L = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i\alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$

Non-linearly separable



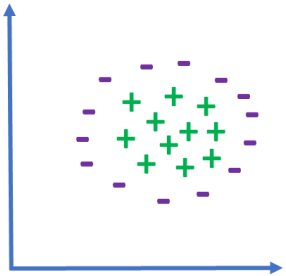
- $\Phi(x) =< x_1^2, x_2^2, \sqrt{2}x_1x_2 >$
- There isn't any new information!
- Reminder: Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$
- Reminder: This optimization / finding extremum of a function depends only on dot products of pairs of samples, given by $\bar{X_i}^T \cdot \bar{X_j}$.

Non-linearly separable



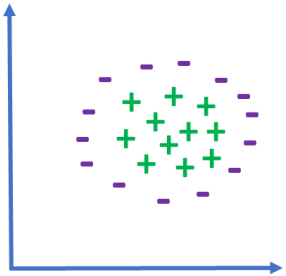
- $\Phi(x) =< x_1^2, x_2^2, \sqrt{2}x_1x_2 >$
- There isn't any new information!
- Reminder: Quadratic programming problem form: $L = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$
- Reminder: This optimization / finding extremum of a function depends only on dot products of pairs of samples, given by $\bar{X_i}^T \cdot \bar{X_j}$.
- What will be $\Phi(\bar{X_i})^T \cdot \Phi(\bar{X_j})$ for a given $\Phi(x)$

Non-linearly separable



- **QUIZ:** What will be $\Phi(\bar{X}_i)^T \cdot \Phi(\bar{X}_j)$ for a given $\Phi(x)$?

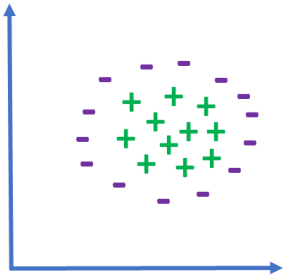
Non-linearly separable



- **QUIZ:** What will be $\Phi(\bar{X}_i)^T \cdot \Phi(\bar{X}_j)$ for a given $\Phi(x)$?

- Consider X_i as x and X_j as y, for the ease of notations: $\Phi(x)^T \Phi(y) =$

Non-linearly separable

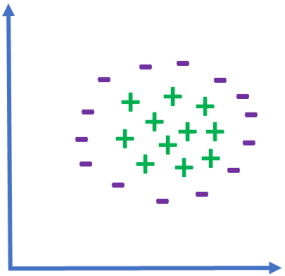


- **QUIZ:** What will be $\Phi(\bar{X}_i)^T \cdot \Phi(\bar{X}_j)$ for a given $\Phi(x)$?

- Consider X_i as x and X_j as y, for the ease of notations: $\Phi(x)^T\Phi(y) =$

$$< x_1^2, x_2^2, \sqrt{2}x_1x_2 >^T \cdot < y_1^2, y_2^2, \sqrt{2}y_1y_2 >$$
$$x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \qquad (22)$$
$$(x_1y_1 + x_2y_2)^2$$

Non-linearly separable



- **QUIZ:** What will be $\Phi(\bar{X}_i)^T \cdot \Phi(\bar{X}_j)$ for a given $\Phi(x)$?

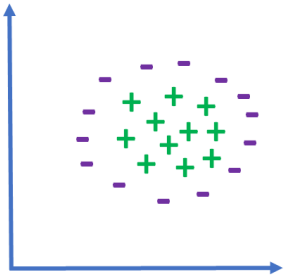
- Consider X_i as x and X_j as y, for the ease of notations: $\Phi(x)^T \Phi(y) =$

$$< x_1^2, x_2^2, \sqrt{2}x_1x_2 >^T \cdot < y_1^2, y_2^2, \sqrt{2}y_1y_2 >$$
$$x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \quad (22)$$
$$(x_1y_1 + x_2y_2)^2$$

- or this is equal to:

$$(x^Ty)^2 \quad (23)$$

Non-linearly separable



- **QUIZ:** What will be $\Phi(\bar{X}_i)^T \cdot \Phi(\bar{X}_j)$ for a given $\Phi(x)$?

- Consider X_i as x and X_j as y, for the ease of notations: $\Phi(x)^T \Phi(y) =$

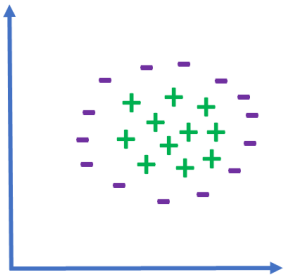
$$< x_1^2, x_2^2, \sqrt{2}x_1x_2 >^T \cdot < y_1^2, y_2^2, \sqrt{2}y_1y_2 >$$
$$x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2 \qquad (22)$$
$$(x_1y_1 + x_2y_2)^2$$

- or this is equal to:

$$(x^Ty)^2 \qquad (23)$$

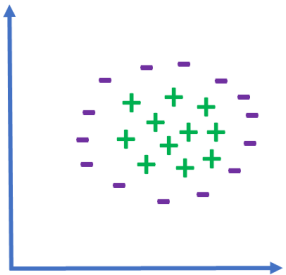
- So, dot product becomes square of last dot product.

Kernel Trick



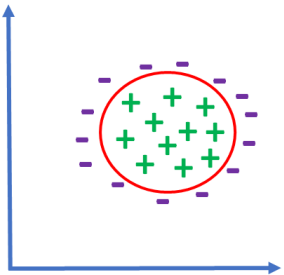
- Refer Equation 23, its particular form of equation of circle in matrix form.

Kernel Trick



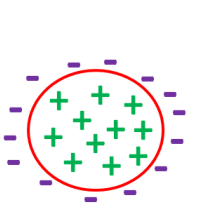
- Refer Equation 23, its particular form of equation of circle in matrix form.
- Somehow, the notion of similarity $(\bar{X_i}^T \cdot \bar{X_j})$ is transformed to notion of circle where some points remain inside the circle while others don't.

Kernel Trick



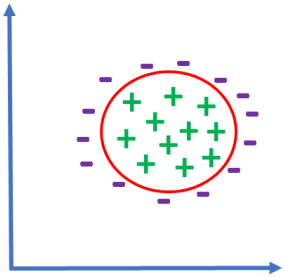
- Refer Equation 23, its particular form of equation of circle in matrix form.
- Somehow, the notion of similarity $(\bar{X}_i^T \cdot \bar{X}_j)$ is transformed to notion of circle where some points remain inside the circle while others don't.
- So, data got transformed from 2D to 3D (without any additional information) in such a way that now it can be separated by hyperplane.

Kernel Trick



- Refer Equation 23, its particular form of equation of circle in matrix form.
- Somehow, the notion of similarity $(\bar{X}_i^T \cdot \bar{X}_j)$ is transformed to notion of circle where some points remain inside the circle while others don't.
- So, data got transformed from 2D to 3D (without any additional information) in such a way that now it can be separated by hyperplane.
- This little trick of projecting data into higher dimension space to make it separable is called Kernel trick.

Kernel Trick



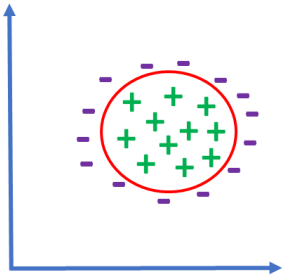
- Coming back to this equation:
  Quadratic programming problem form:

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$$

and Equation 23:

$$(x^T y)^2$$

Kernel Trick



- Coming back to this equation:
  Quadratic programming problem form:

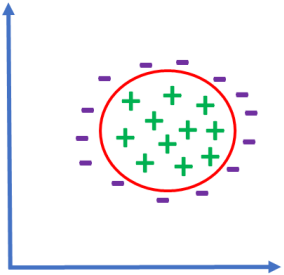
$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$$

  and Equation 23:

$$(x^T y)^2$$

- This signifies that data points don't need to be transformed separately, but rather its just a dot product squared! **So we actually never used** Φ.

Kernel Trick



- Coming back to this equation:
  Quadratic programming problem form:

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$$

  and Equation 23:

$$(x^T y)^2$$

- This signifies that data points don't need to be transformed separately, but rather its just a dot product squared! **So we actually never used Φ.**
- **That's the beauty of mathematics and SVM.**

Kernel Trick

- $$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \bar{X}_i \cdot \bar{X}_j$$

- This equation can now be written as:

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(\bar{X}_i \cdot \bar{X}_j) \tag{24}$$

  Where K is Kernel function, that takes X_i and X_j and returns scalar, the inner product (generalization of the dot product) between two points in a suitable feature space.

- Kernel functions allow to inject domain knowledge into classifier.

Kernel Trick

- Gaussian Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}} \tag{25}$$

- Polynomial

$$K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i)^T \mathbf{x}_j + c)^p \tag{26}$$

- Neural-net inspired!

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\kappa \mathbf{x}_i \mathbf{x}_j - \delta)^p \tag{27}$$

- Radial Basis

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{(\gamma ||\mathbf{x}_i - \mathbf{x}_j||^2)} \tag{28}$$

Section Contents

Python for SVM

```
from __future__ import division, print_function
import numpy as np
from sklearn import datasets, svm
#from sklearn.cross_validation import train_test_split
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt

from sklearn.tree import DecisionTreeClassifier
from sklearn. ensemble import RandomForestClassifier, BaggingClassifier,
    AdaBoostClassifier, VotingClassifier

iris = datasets.load_iris()
X = iris.data[:,:2]  # First two features, can take last two using [:,2:]
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
    random_state=42)
```

Python for SVM

```
def evaluate_on_test_data(model=None):
    predictions = model.predict(X_test)
    correct_classifications = 0
    for i in range(len(y_test)):
        if predictions[i] == y_test[i]:
            correct_classifications += 1
    accuracy = 100*correct_classifications/len(y_test) #Accuracy as a
    percentage
    return accuracy

kernels = ('linear','poly','rbf')
accuracies = []
for index, kernel in enumerate(kernels):
    model = svm.SVC(kernel=kernel)
    model.fit(X_train, y_train)
    acc = evaluate_on_test_data(model)
    accuracies.append(acc)
    print("{} % Test accuracy obtained with kernel = {}".format(acc, kernel))
```

Python for SVM

```
#Train SVMs with different kernels
svc = svm.SVC(kernel='linear').fit(X_train, y_train)
rbf_svc = svm.SVC(kernel='rbf', gamma=0.7).fit(X_train, y_train)
poly_svc = svm.SVC(kernel='poly', degree=9).fit(X_train, y_train)



#Create a mesh to plot in
h = .02  # step size in the mesh
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))

#Define title for the plots
titles = ['SVM with linear kernel',
          'SVM with RBF kernel',
          'SVM with polynomial (degree 9) kernel']
```

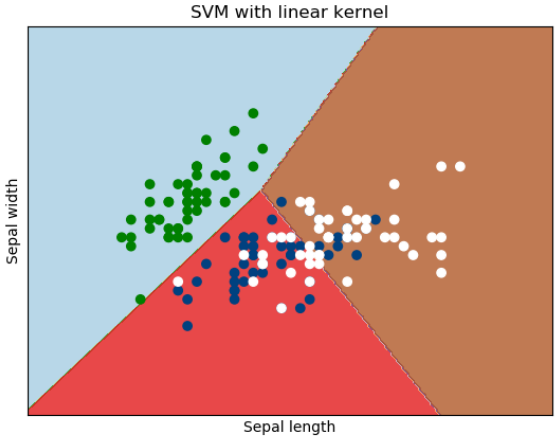
Python for SVM

```
for i, clf in enumerate((svc, rbf_svc, poly_svc)):
    # Plot the decision boundary. For that, we will assign a color to each
    # point in the mesh [x_min, m_max]x[y_min, y_max].
    plt.figure(i)

    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    # Put the result into a color plot
    Z = Z.reshape(xx.shape)
    plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=0.8)

    # Plot also the training points
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.ocean)
    plt.xlabel('Sepal length')
    plt.ylabel('Sepal width')
    plt.xlim(xx.min(), xx.max())
    plt.ylim(yy.min(), yy.max())
    plt.xticks(())
    plt.yticks(())
    plt.title(titles[i])
plt.show()
```

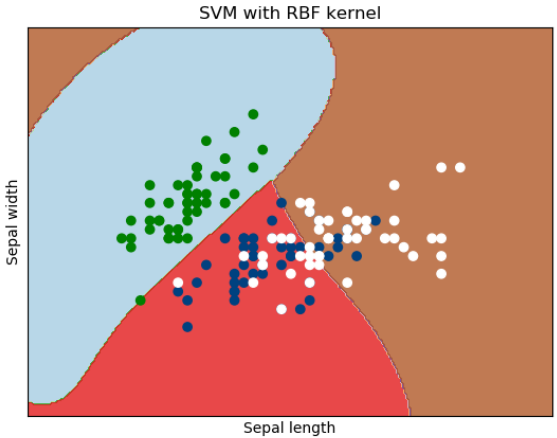
SVM Visulaization



SVM with Linear Kernel (No transformation)

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}^T \mathbf{x}_j + c$$

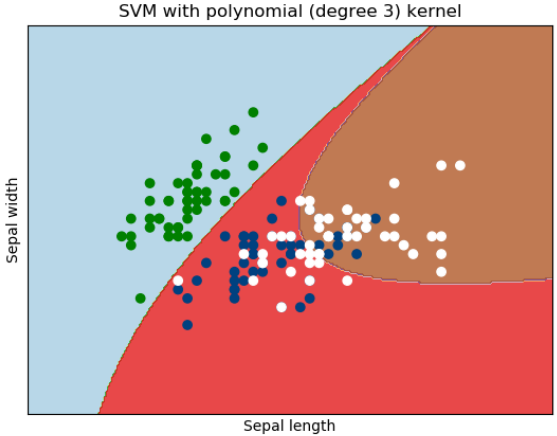
SVM Visulaization



SVM with RBF

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

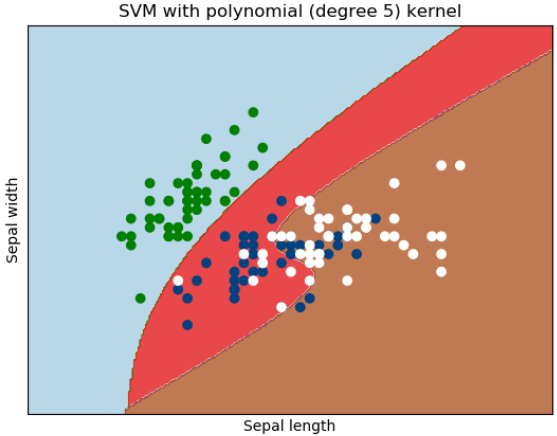
SVM Visulaization



SVM with Polynomial Kernel (Degree 3)

$$K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i)^T \mathbf{x}_j + c)^p$$

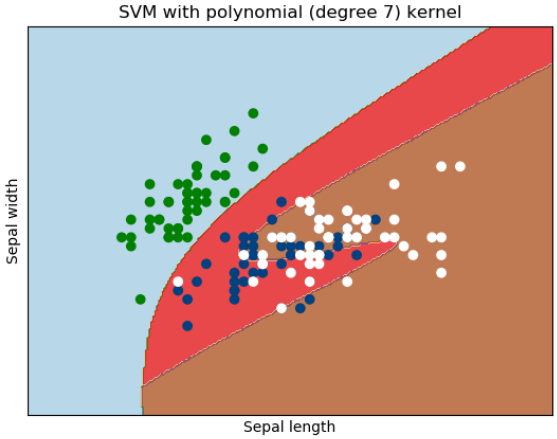
SVM Visulaization



SVM with Polynomial Kernel (Degree 5)

$$K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i)^T \mathbf{x}_j + c)^p$$

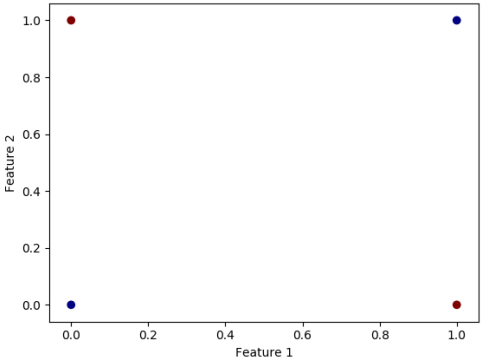
SVM Visulaization



SVM with Polynomial Kernel (Degree 7)

$$K(\mathbf{x}_i, \mathbf{x}_j) = ((\mathbf{x}_i)^T \mathbf{x}_j + c)^p$$

XOR problem

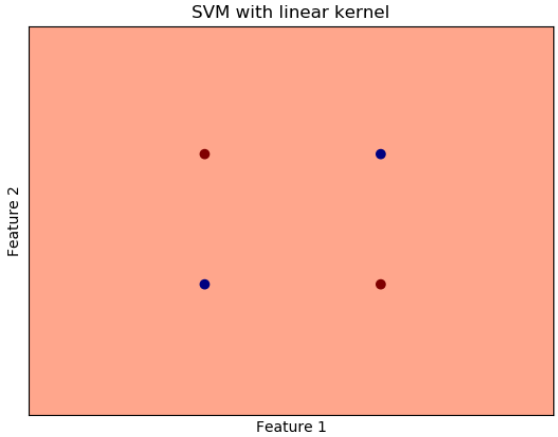


- XOR data.
- Problem is non-linearly separable in the given feature space.

Video [4]

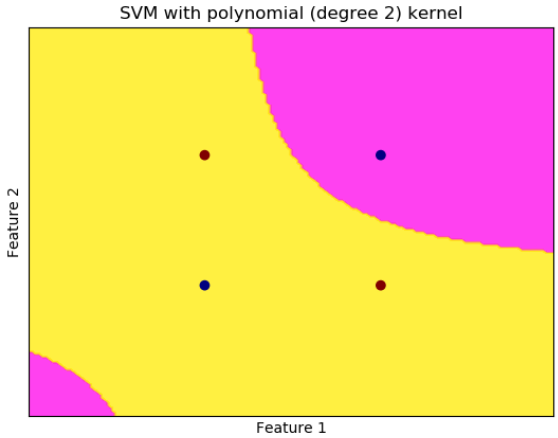
[4] https://youtu.be/5KIYu3zKvqo

XOR problem



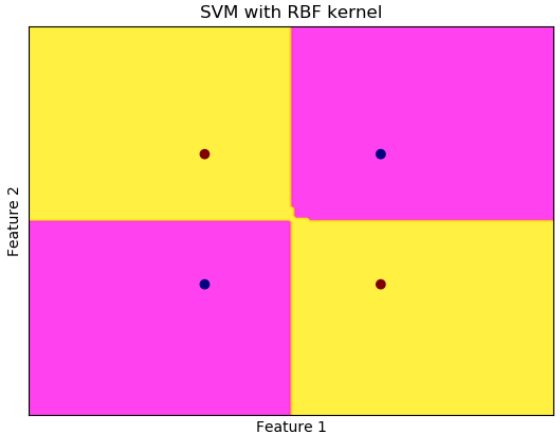
SVM with Linear Kernel (No transformation)

XOR problem



SVM with Polynomial Kernel

XOR problem



SVM with RBF Kernel

Conclusion

- Powerful theoretical grounds.
- Global, unique solution (convex optimization function).
- Performance depends on choice of kernel and parameters.
- Training is memory-intensive.
- Complexity dependent on the number of support vectors.

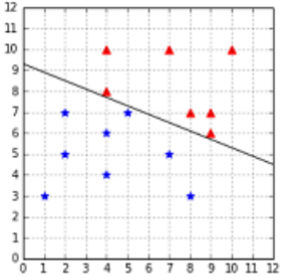
Video [5]

[5] https://youtu.be/FxLIsbnp_5c

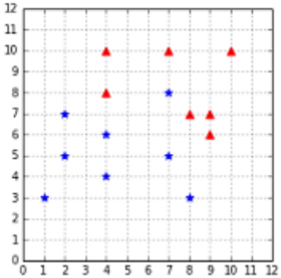
Section Contents

Need for soft Margin SVM

- Real-life data is often noisy, thus linear separability is an issue.
- Even when the data is linearly separable, the outlier can be closer to the other examples than most of the examples of its class, thus reducing the margin, or it can be among the other examples and break linear separability.



Outlier reducing the margin



Outlier breaks linear separability *6

- In this case, there is no solution to the optimization problem solved earlier.

[6] Image taken from SVM Succinctly

Slack variable

- In 1995, Vapnik and Cortes introduced a modified version of the original SVM that allows the classifier to make some mistakes.
- The goal is now not to make zero classification mistakes, but to make as few mistakes as possible.
- Constraints of the optimization problem was modified, so the constraint (refer Eq 4)

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1$$

becomes

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1 - \xi_i \tag{29}$$

where: ξ= slack variable and $\forall_i \geq 0$.

Slack variable

- In 1995, Vapnik and Cortes introduced a modified version of the original SVM that allows the classifier to make some mistakes.
- The goal is now not to make zero classification mistakes, but to make as few mistakes as possible.
- Constraints of the optimization problem was modified, so the constraint (refer Eq 4)

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1$$

becomes

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1 - \xi_i \tag{29}$$

where: ξ= slack variable and $\forall_i \geq 0$.

- ξ is subtracted from 1, in order to make it possible to satisfy constraint.

Slack variable

- The problem is that we could choose a huge value of ξ for every example, and all the constraints will be satisfied.
- To avoid this, we need to modify the objective function (refer Eq. 14 for objective function of hard margin SVM) to penalize the choice of a big ξ:

$$\underset{W,b,\xi}{\operatorname{argmin}} \frac{1}{2}||W||^2 + C\sum_{i=1}^{n}\xi_i \tag{30}$$

subject to

$$y_i(\bar{W} \cdot \bar{X}_i + b) \geq 1 - \xi_i$$

and $\xi_i \geq 0 \, \forall_i$

Slack variable

- Consider:

$$\underset{W,b,\xi}{\operatorname{argmin}} \frac{1}{2}||W||^2 + C\sum_{i=1}^{n}\xi_i$$

  The slack variable ξ_i allows the input xi to be closer to the hyperplane (or even be on the wrong side), but there is a penalty in the objective function for such "slack".

- How to select hyper-parameter C?

Slack variable

- Consider:

$$\underset{W,b,\xi}{\operatorname{argmin}} \frac{1}{2}||W||^2 + C\sum_{i=1}^{n}\xi_i$$

The slack variable ξ_i allows the input xi to be closer to the hyperplane (or even be on the wrong side), but there is a penalty in the objective function for such "slack".

- How to select hyper-parameter C?

Value of C (hyper-parameter tuning)

1. If C is very large (penalty is higher), the SVM becomes very strict and tries to classify all data points correctly.

Slack variable

- Consider:

$$\underset{W,b,\xi}{\operatorname{argmin}} \frac{1}{2}||W||^2 + C\sum_{i=1}^{n}\xi_i$$

The slack variable ξ_i allows the input xi to be closer to the hyperplane (or even be on the wrong side), but there is a penalty in the objective function for such "slack".

- How to select hyper-parameter C?

Value of C (hyper-parameter tuning)

1. If C is very large (penalty is higher), the SVM becomes very strict and tries to classify all data points correctly.
2. If C is very small, the SVM becomes very loose and may "sacrifice" some points to obtain a simpler solution.

Slack variable

- Consider:

$$\underset{W,b,\xi}{\operatorname{argmin}} \frac{1}{2}||W||^2 + C\sum_{i=1}^{n}\xi_i$$

The slack variable ξ_i allows the input xi to be closer to the hyperplane (or even be on the wrong side), but there is a penalty in the objective function for such "slack".

- How to select hyper-parameter C?

Value of C (hyper-parameter tuning)

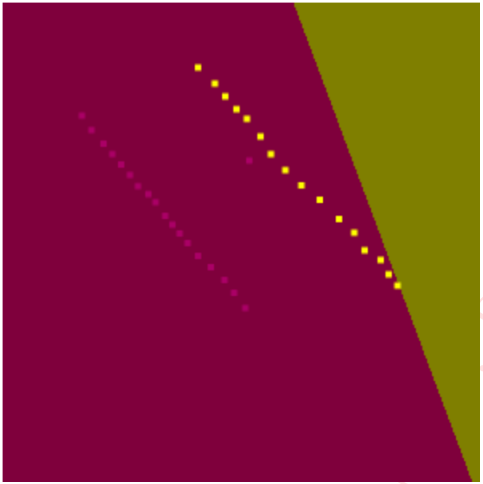
1. If C is very large (penalty is higher), the SVM becomes very strict and tries to classify all data points correctly.
2. If C is very small, the SVM becomes very loose and may "sacrifice" some points to obtain a simpler solution.
3. Usually telescopic / grid search is applied to find best C for the given dataset.

Ocular Proof

This image corresponds to large value of C.

[7] Demo images from LIBSVM website: `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

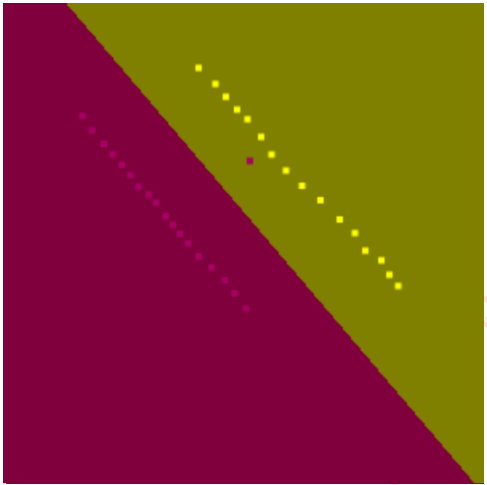
Ocular Proof



This image corresponds to small value of C, over-simplification of solution.

[7] Demo images from LIBSVM website: `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Ocular Proof



∗[7]

This image corresponds to appropriate value of C given dataset (maximizing margin, sacrificing few (one data point) to obtain wider margin / better generalization).

[7] Demo images from LIBSVM website: `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

Section Contents

Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

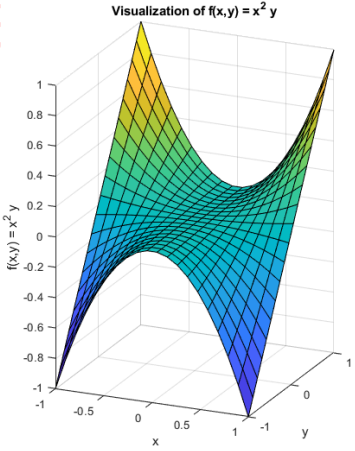
Visualization

Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

Visualization



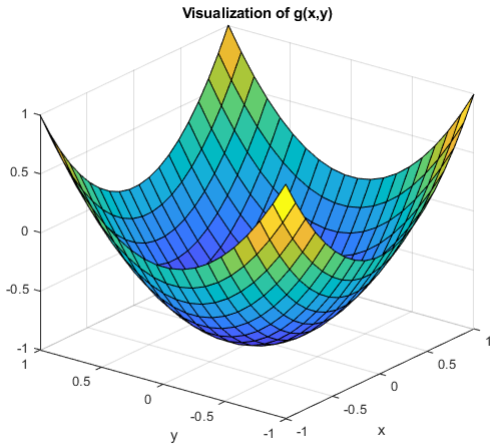
The function is in 3-D. To analytically examine the problem , we can use contour plot.

Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

Visualization

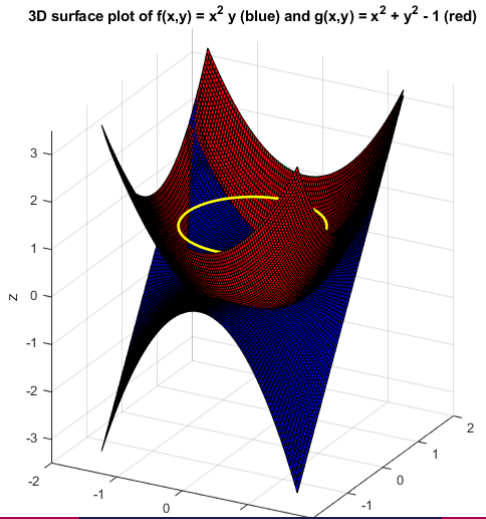


Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

Visualization

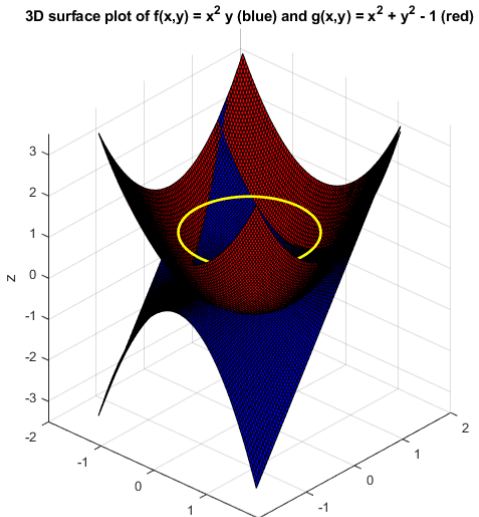


Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

Visualization

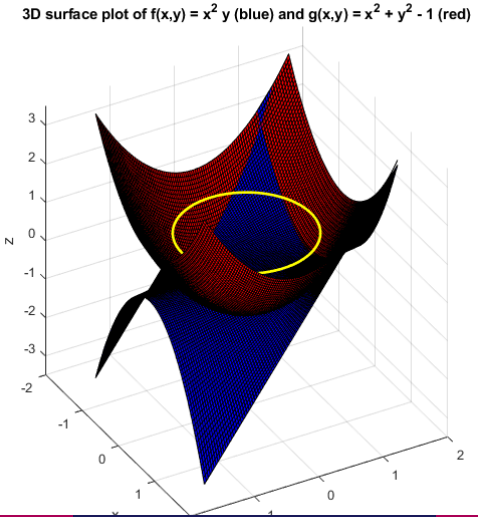


Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

Visualization



Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

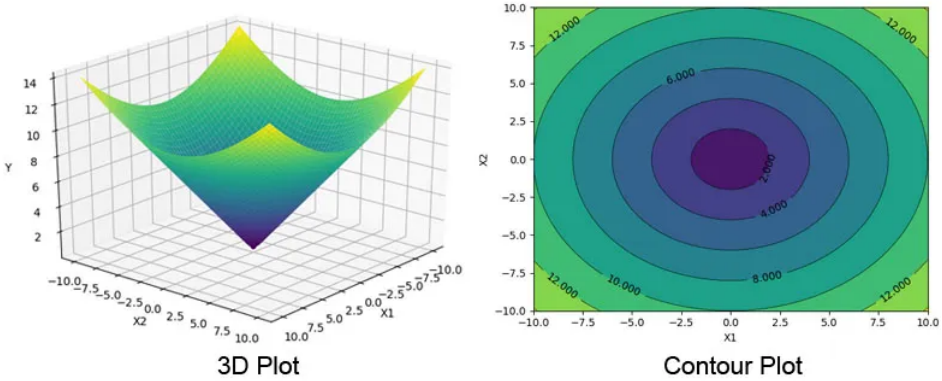
- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint $g(x, y)$ that $x^2 + y^2 = 1$ (unit circle)

Visualization

Contour plot

A contour plot is a graphical technique for representing a 3-D surface by plotting constant z slices, called contours, on a 2-D format. That is, lines are drawn for all possible pairs of (x, y) that produce same/constant output (z).

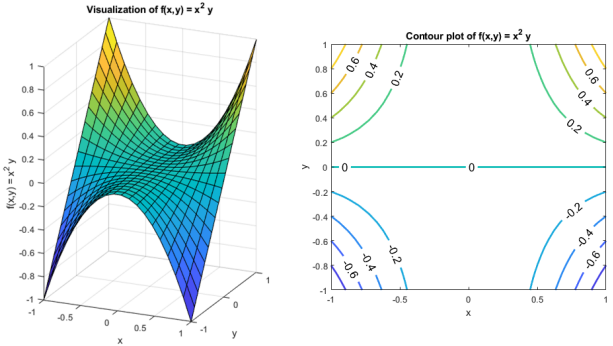
Contour Plot



Contour plot

A contour plot is a graphical technique for representing a 3-D surface by plotting constant z slices, called contours, on a 2-D format. That is, lines are drawn for all possible pairs of (x, y) that produce same/constant output (z).

Contour Plot



3D plot of function, along with corresponding contour plot of the problem in hand.

Contour plot

A contour plot is a graphical technique for representing a 3-D surface by plotting constant z slices, called contours, on a 2-D format. That is, lines are drawn for all possible pairs of (x, y) that produce same/constant output (z).

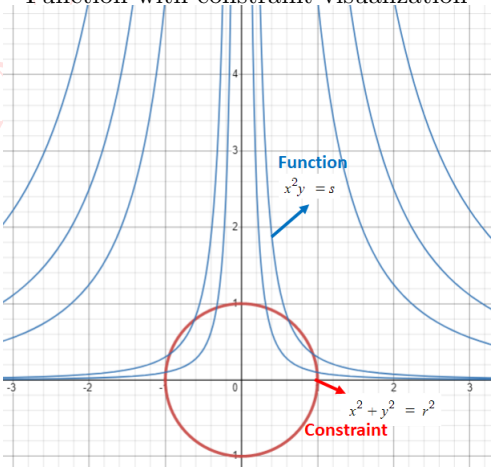
Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint that $x^2 + y^2 = 1$ (unit circle)

Visualization

Function with constraint visualization



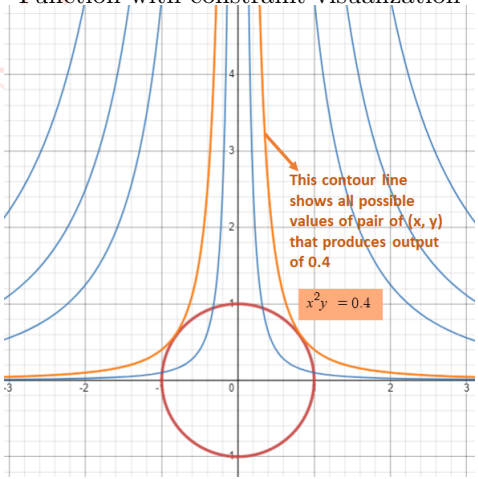
Optimizing function with constraint

- Maximize $f(x, y) = x^2 y$ on the set $x^2 + y^2 = 1$.

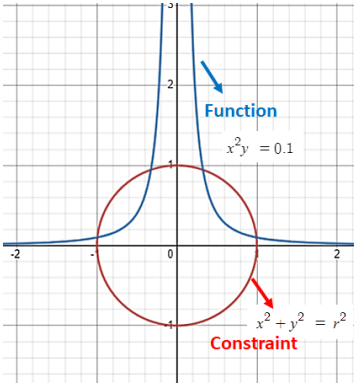
- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint that $x^2 + y^2 = 1$ (unit circle)

Visualization

Function with constraint visualization

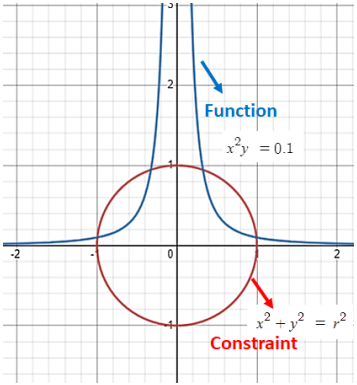


Optimizing function with constraint

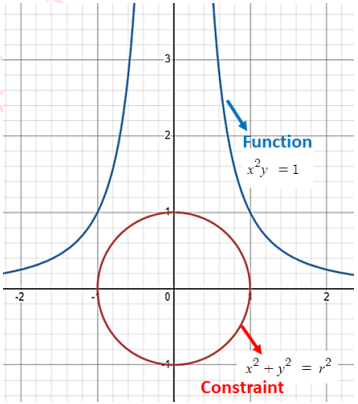


Function intersects with the constraint. This means that this pair of (x, y) satisfies constraint (four possible pair of (x, y) values), but visually we can observe that they are maximum values.

Optimizing function with constraint



Function intersects with the constraint. This means that this pair of (x, y) satisfies constraint (four possible pair of (x, y) values), but visually we can observe that they are maximum values.



Function never intersects with the constraint. This means that this pair of (x, y) is off the constraint. It also shows that, as we are max. x^2y subject to constraint, we can never go as high as these values of (x, y).

Optimizing function with constraint

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint that $x^2 + y^2 = 1$ (unit circle)

Optimizing function with constraint

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint that $x^2 + y^2 = 1$ (unit circle)

Objective

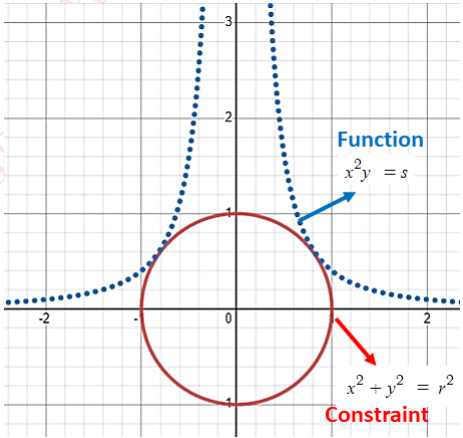
To maximize function $f(x, y) = x^2 y$ while satisfying constraint $x^2 + y^2 = 1$, is to find maximum value of pair of (x, y) or value of constant s to the point that afterwards its off the constraint.
This will only happen when the two functions ($f(x, y)$ & $g(x, y)$) are **tangent**.

Optimizing function with constraint

- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint that $x^2 + y^2 = 1$ (unit circle)

Objective

To maximize function $f(x, y) = x^2 y$ while satisfying constraint $x^2 + y^2 = 1$, is to find maximum value of pair of (x, y) or value of constant s to the point that afterwards its off the constraint.
This will only happen when the two functions ($f(x, y)$ & $g(x, y)$) are **tangent**.

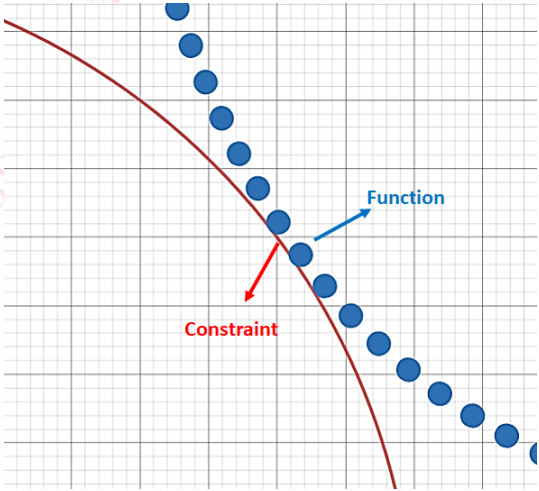


Optimizing function with constraint

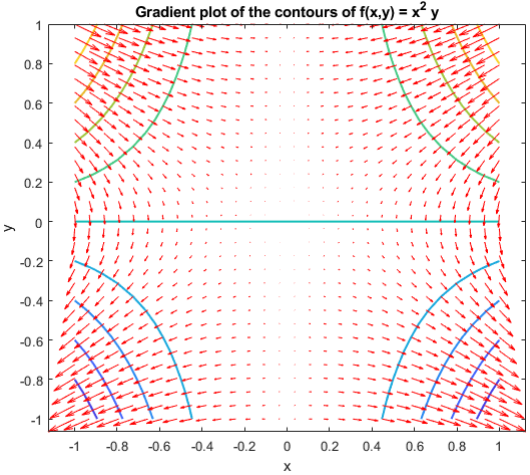
- Here we are trying to
  - Optimize multi-variable function $f(x, y) = x^2 y$
  - with the constraint that $x^2 + y^2 = 1$ (unit circle)

Objective

To maximize function $f(x, y) = x^2 y$ while satisfying constraint $x^2 + y^2 = 1$, is to find maximum value of pair of (x, y) or value of constant s to the point that afterwards its off the constraint.
This will only happen when the two functions ($f(x, y)$ & $g(x, y)$) are **tangent**.

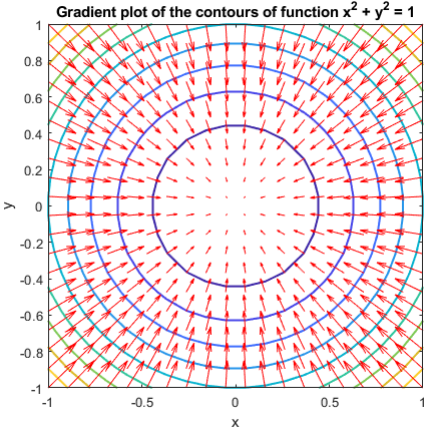


Optimizing function with constraint



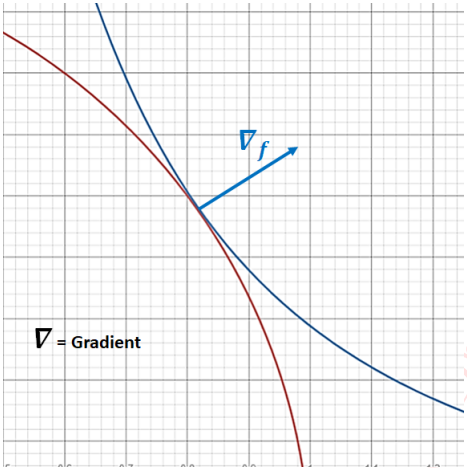
The gradient of f or g evaluated at a point (x_0, y_0) always gives a vector perpendicular to the contour line passing through that point (as there is no change in value along contour line).

Optimizing function with constraint



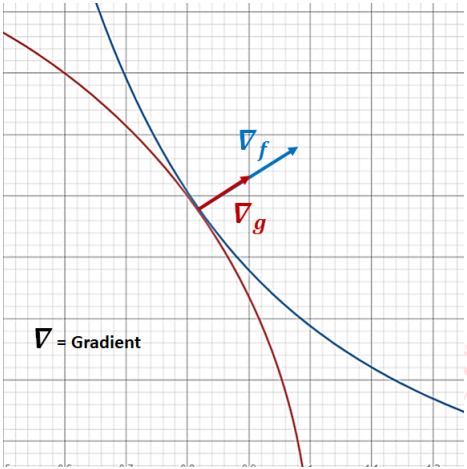
The gradient of f or g evaluated at a point (x_0, y_0) always gives a vector perpendicular to the contour line passing through that point (as there is no change in value along contour line).

Optimizing function with constraint



- The gradient of f evaluated at a point (x_0, y_0) always gives a vector perpendicular to the contour line passing through that point (as there is no change in value along contour line).
- When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.

Optimizing function with constraint

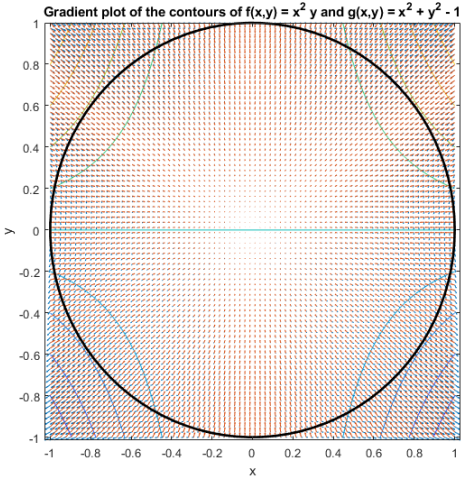


- The fact that contour lines are tangent tells us nothing about the magnitude of each of these gradient vectors, but that's okay. When two vectors point in the same direction, it means we can multiply one by some constant to get the other.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

λ = Lagrange multiplier
$f(x,y)$ = Function
$g(x,y)$ = Constraint

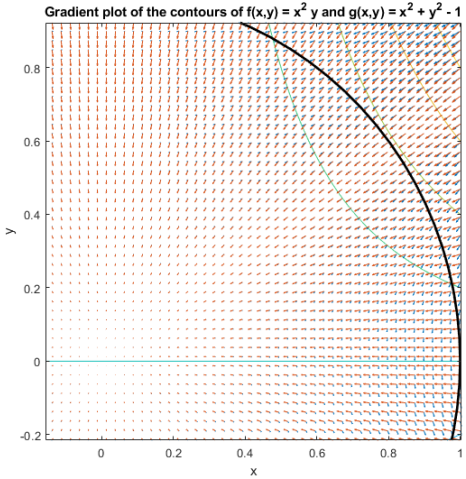
Ocular Proof



- Ocular Proof: When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

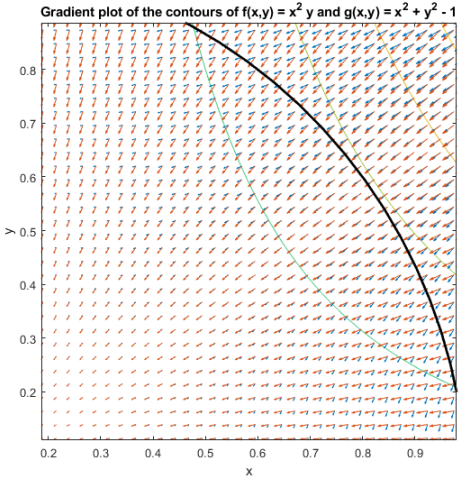
Ocular Proof



- Ocular Proof: When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

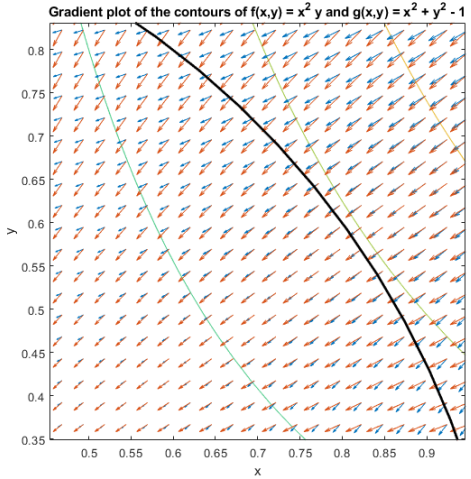
Ocular Proof



- Ocular Proof: When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

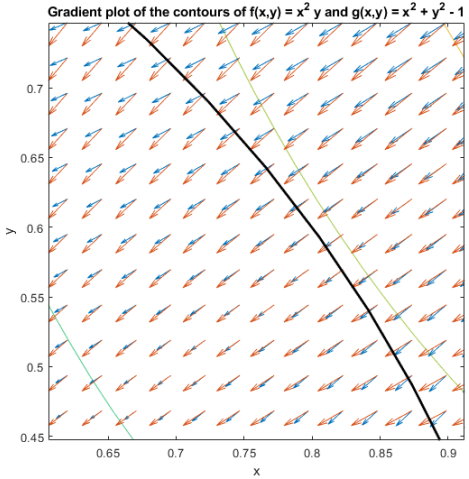
Ocular Proof



- Ocular Proof: When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

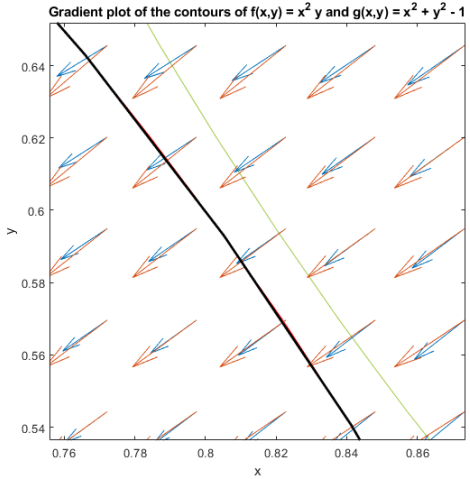
Ocular Proof



- Ocular Proof: When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

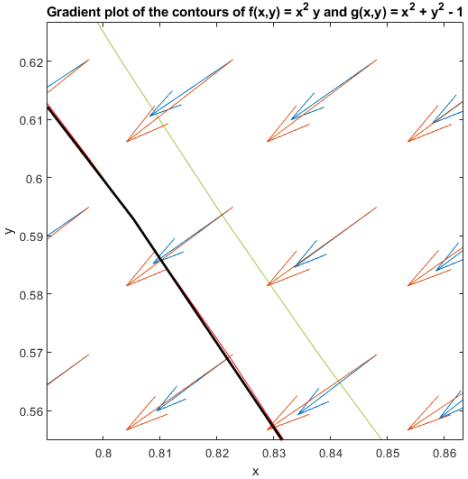
Ocular Proof



- Ocular Proof: When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

Ocular Proof



- Ocular Proof: When the contour lines of two functions f and g are tangent, their gradient vectors are parallel.
- Since this tangency means their gradient vectors align:

$$\bigtriangledown f(x,y) = \lambda \bigtriangledown g(x,y)$$

Optimizing function with constraint: Example Solution

$$\begin{aligned} L &= f(x,y) - \lambda g(x,y) \\ &= x^2 y - \lambda[x^2 + y^2] \end{aligned}$$

To find max. take derivative. First partial derivative w.r.t "x":

$$\begin{aligned} \frac{\partial L}{\partial x} &= 2xy - \lambda 2x \\ y &= \lambda \end{aligned} \tag{31}$$

Partial derivative w.r.t "y":

$$\begin{aligned} \frac{\partial L}{\partial y} &= x^2 - \lambda 2y \\ x^2 &= \lambda 2y \\ x^2 &= 2\lambda^2 (\text{as } y = \lambda) \\ x &= \pm\sqrt{2}\lambda \end{aligned} \tag{32}$$

Optimizing function with constraint: Example Solution

Putting back values of "x" and "y" found from equations 31 and 32 in the constraint equation:

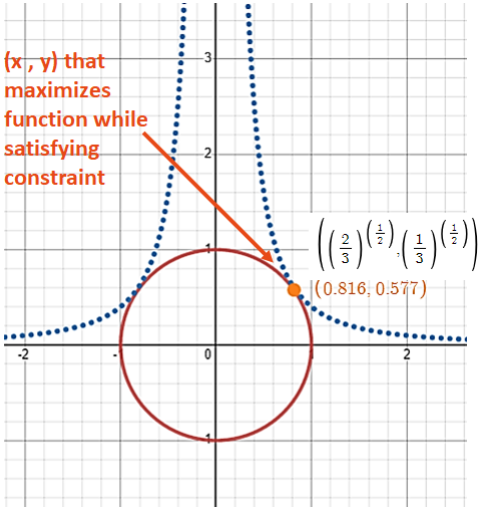
$$\begin{aligned} x^2 + y^2 &= 1 \\ \left[\sqrt{2}\lambda\right]^2 + \lambda^2 &= 1 \\ 3\lambda^2 &= 1 \\ \lambda &= \pm\sqrt{\frac{1}{3}} \end{aligned} \tag{33}$$

Put back value of λ in equations 31 and 32 to find value of x and y:

$$y = \pm\sqrt{\frac{1}{3}} \tag{34}$$

$$x = \pm\sqrt{2}\sqrt{\frac{1}{3}} = \pm\sqrt{\frac{2}{3}} \tag{35}$$

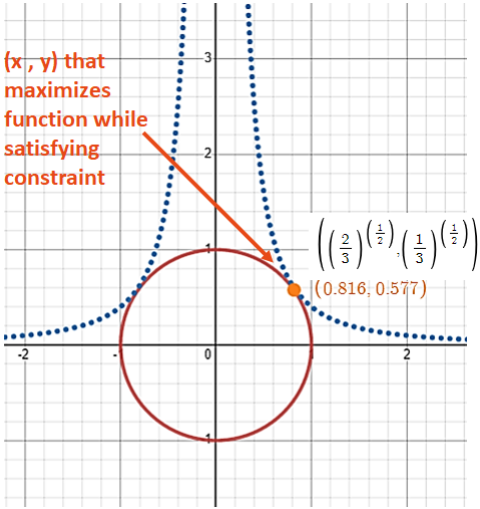
Optimizing function with constraint: Example Solution



- From equations 34 and 35 , we know values of x and y. They make four possible pairs of (x, y):

1. $(\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}})$
2. $(-\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}})$
3. $(\sqrt{\frac{2}{3}}, -\sqrt{\frac{1}{3}})$
4. $(-\sqrt{\frac{2}{3}}, -\sqrt{\frac{1}{3}})$

Optimizing function with constraint: Example Solution



- From equations 34 and 35 , we know values of x and y. They make four possible pairs of (x, y):

1. $(\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}})$
2. $(-\sqrt{\frac{2}{3}}, \sqrt{\frac{1}{3}})$
3. $(\sqrt{\frac{2}{3}}, -\sqrt{\frac{1}{3}})$
4. $(-\sqrt{\frac{2}{3}}, -\sqrt{\frac{1}{3}})$

- Last two point make function (x^2y) negative (will not max. func.) and first two points gives same output and that is the maximum value function can achieve while satisfying constraint (refer image on the left).