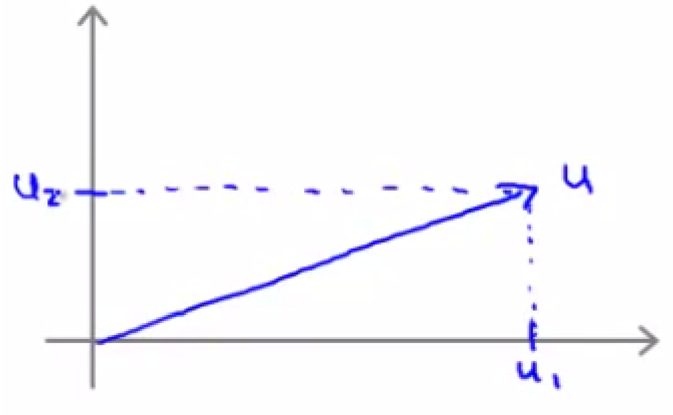
# Mathematics Behind Large Margin Classification

### Vector inner product

Let's say we have two vectors $u$ and $v$, so both two dimensional vectors, what is the inner product ($u^T v$)?

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Plotting $u$ on graph (i.e $u_1$ vs. $u_2$)

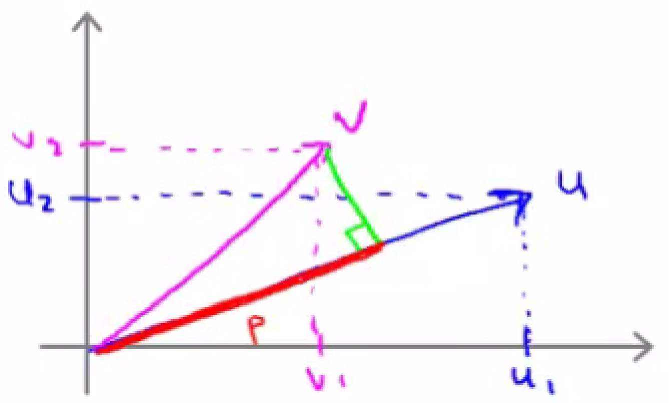


One property which is good to have is the **norm of a vector** (written as $||u||$)

- $||u||$ = The euclidean length of vector $u$
- So $||u|| = \sqrt{u_1^2 + u_2^2} \in \mathbb{R}$
- $||u||$ is the length of the arrow above

We can plot $v$ on the same axis in the same way (i.e $v_1$ vs. $v_2$), so how can we do the inner product between $u$ and $v$?

- For the inner product, take $v$ and orthogonally project down onto $u$.
- Measure the length/magnitude of the projection (the red line $p$).
- $p \in \mathbb{R}$ is the length or is the magnitude projection of the vector $v$ onto the vector $u$.
- So it's possible to show that: $u^T v = p \cdot ||u||$
- Another way to calculate the inner product is: $u^T v = u_1 v_1 + u_2 v_2$
- So therefore $p \cdot ||u|| = u_1 v_1 + u_2 v_2$

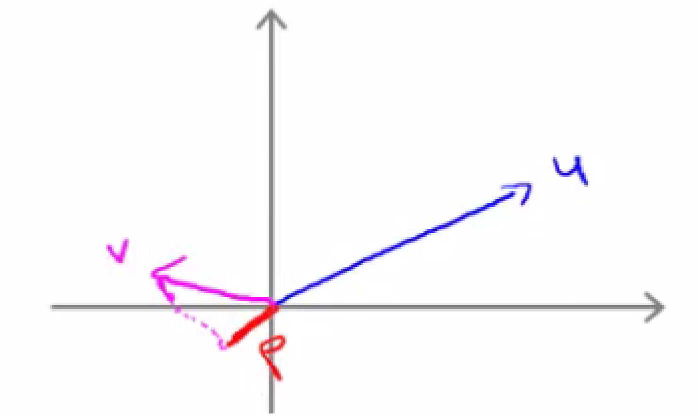


We can reverse this too:

- So we could do $v^T u = v_1 u_1 + v_2 u_2$ (we project $v$ on $u$)

- Do the same process, but with the rows of $u$ and $v$ reversed
- Which would give us the same number

$p$ can be negative if the angle between them is $90$ degrees or more



Then if we project $v$ onto $u$, what we get is a projection it looks like above, so here $p$ is negative

- In this case, we will still have $u^T v = p \cdot ||u||$ where $p < 0$

So that's how vector inner products work. We're going to use these properties of vector inner product to try to understand the support vector machine optimization objective a little better.

## SVM Decision Boundary

$$\begin{aligned} &\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 \\ &\text{s.t.} \quad \theta^T x^{(i)} \geq 1 \qquad \text{if } y^{(i)} = 1 \\ &\qquad\ \ \theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0 \end{aligned}$$

- For the following explanation - two simplification
  - Set $\theta_0 = 0$ (i.e. ignoring the intercept terms)
  - Each example has only 2 features: set $n = 2$ - $(x_1, x_2)$

Given we only have two parameters we can simplify our function to:

- $min_\theta \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2} \theta_1^2 + \theta_2^2$

And, can be re-written as:

- $\frac{1}{2} (\sqrt{\theta_1^2 + \theta_2^2})^2$

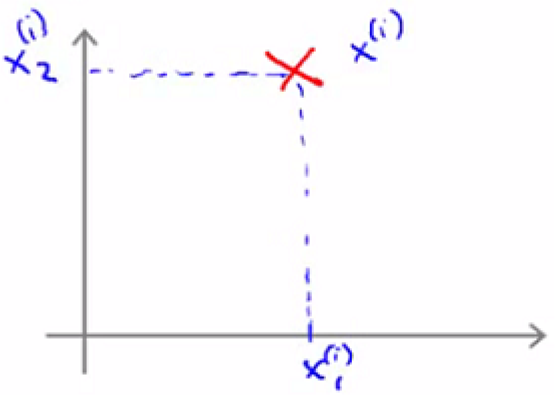
We may notice that:

- $\sqrt{\theta_1^2 + \theta_2^2} = ||\theta||$

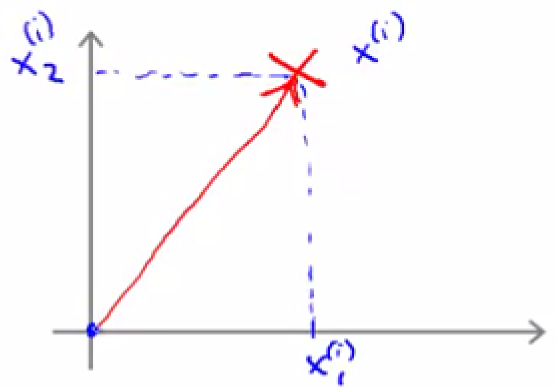
So $||\theta||$ is the norm (euclidean distance) of theta, and finally, this means our optimization function can be re-defined as: $\frac{1}{2} ||\theta||^2$.

- So the SVM is minimizing the squared norm

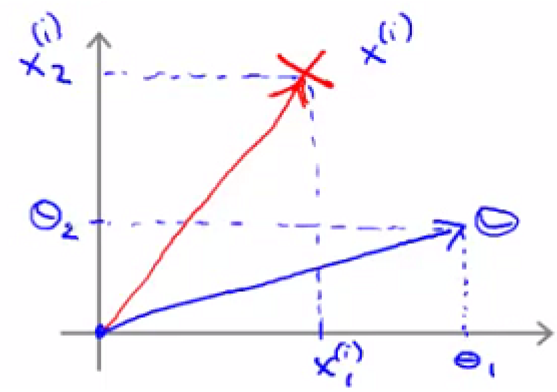
Given this, what are the $(\theta^T x^{(i)})$ parameters doing?, Given $\theta$ and given example $x^{(i)}$ what is this equal to?, Say we have a single positive training example (red cross below):



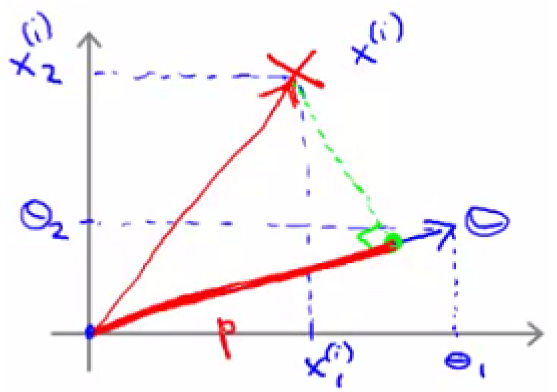
Although we haven't been thinking about examples as vectors it can be described as such



Now, say we have our parameter vector $\theta$ and we plot that on the same axis



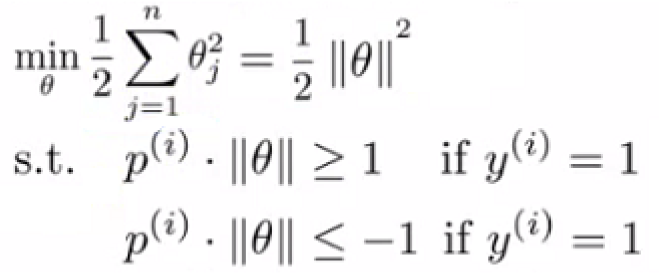
The next question is what is the inner product of these two vectors?



Using our earlier method, the way we compute that is we take our example and project it onto our parameter vector $\theta$, and that give us the length $p$.

- $p$, is in fact $p^{(i)}$, because it's the length of $p$ for example $i$.

    - $(\theta^T x^{(i)}) = p^{(i)} \cdot ||\theta|| = \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$
    - So these are both equally valid ways of computing $\theta^T x^{(i)}$
- The constraints we defined earlier:
    - $(\theta^T x^{(i)}) \geq 1$ if $y = 1$
    - $(\theta^T x^{(i)}) \leq -1$ if $y = 0$
- Can be replaced/substituted with the constraints:
    - $p^{(i)} * ||\theta|| \geq 1$ if $y = 1$
    - $p^{(i)} * ||\theta|| \leq -1$ if $y = 0$
- Writing that into our optimization objective

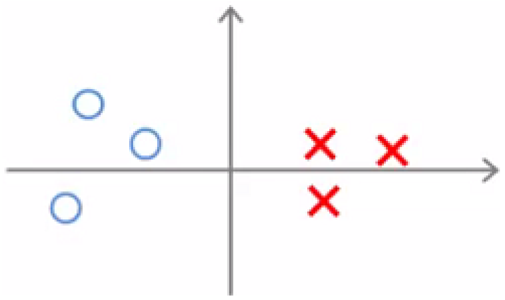


$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2} ||\theta||^2$$

$$\text{s.t.} \quad p^{(i)} \cdot ||\theta|| \geq 1 \quad \text{if } y^{(i)} = 1$$

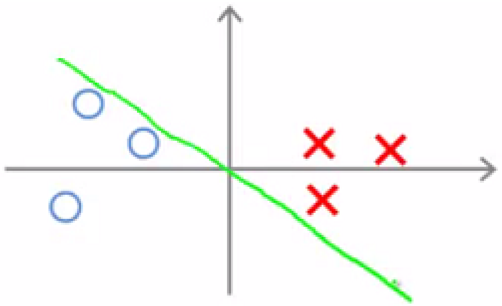
$$p^{(i)} \cdot ||\theta|| \leq -1 \quad \text{if } y^{(i)} = 1$$

So, given we've redefined these functions let us now consider the training example below:

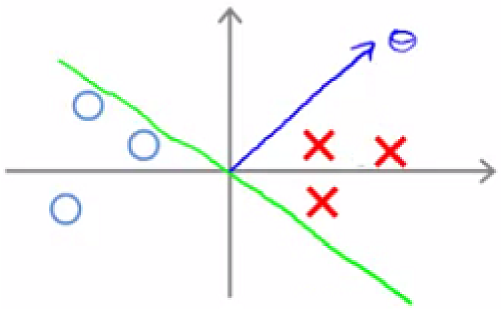


Given this data, what boundary will the SVM choose? Note that we're still assuming $\theta_0 = 0$, which means the boundary has to pass through the origin $(0, 0)$.

Here's one option, let's say the support vector machine were to choose the decision boundary below. This is not a very good choice because it has **very small margins**. This decision boundary comes very close to the training examples.

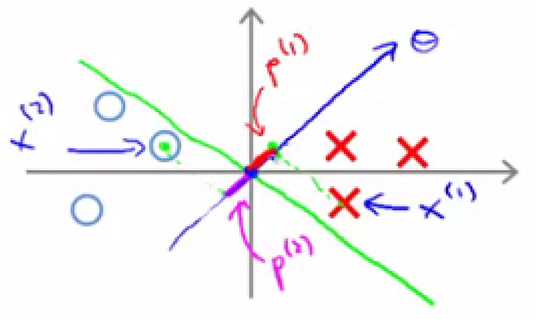


Lets discuss why the SVM would not chose this decision boundary, we can show that $\theta$ is at 90 degrees to the decision boundary. Concretely $\theta$ is always at 90 degrees to the decision boundary



So now lets look at what this implies for the optimization objective, look at first example ($x^{(1)}$)

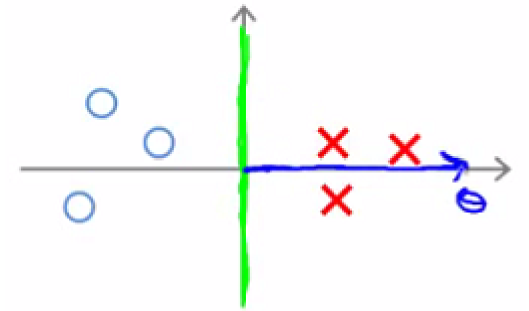
- Project a line from $x^{(1)}$ on to to the $\theta$ vector (so it hits at 90 degrees)
  - The distance between the intersection and the origin is ($p^{(1)}$)
- Similarly, look at second example ($x^{(2)}$)
  - Project a line from $x^{(2)}$ into to the $\theta$ vector
  - This is the magenta line, which will be negative ($p^{(2)}$)
- If we overview these two lines below we see a graphical representation of what's going on:



- We find that both these $p$ values are going to be pretty small
- If we look back at our optimization objective
  - We know we need $p^{(1)} \cdot ||\Theta|| \geq 1$ for positive examples
    - If $p^{(1)}$ is small, means that $||\Theta||$ must be pretty large
  - Similarly, for negative examples we need $p^{(2)} \cdot ||\Theta|| \leq -1$
    - So if $p^{(2)}$ is a small negative number, so $||\Theta||$ must be a large number

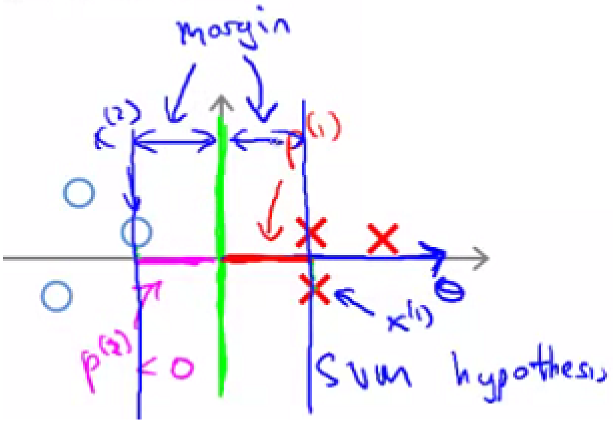
**What we are doing in the optimization objective is we are trying to find a setting of parameters where the norm of theta is small**. So this doesn't seem like a good direction for the parameter vector (because as $p$ values get smaller $||\Theta||$ must get larger to compensate), so we should make $p$ values larger which allows $||\Theta||$ to become smaller.

In contrast, lets look at a different decision boundary:



Now if we look at the projection of the examples to $\theta$ we find that $p^{(1)}$ and $p^{(2)}$ becomes large and $||\Theta||$ can become small.

- This means that by choosing this second decision boundary we can make $||\Theta||$ smaller
  - Which is why the SVM choses this hypothesis as better
  - This is how we generate the large margin effect

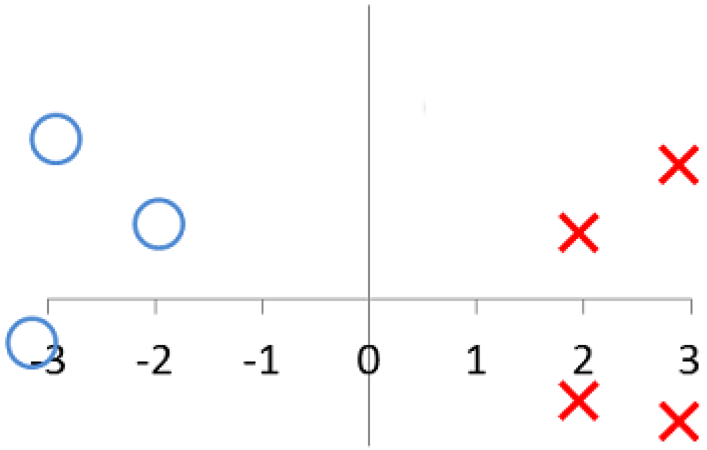


- The magnitude of this margin is a function of the $p$ values
  - So by maximizing these $p$ values we minimize $||\theta||$
- Finally, we did this derivation assuming $\theta_0 = 0$,
  - If this is the case we're entertaining only decision boundaries which pass through $(0, 0)$
  - If we allow $\theta_0$ to be other values then this simply means we can have decision boundaries which cross through the $x$ and $y$ values at points other than $(0, 0)$
  - Can show with basically same logic that this works, and even when $\theta_0$ is non-zero when we have optimization objective described above (when $C$ is very large) that the SVM is looking for a large margin separator between the classes

**Video Question:** The SVM optimization problem we used is:

$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

$$\begin{aligned} \text{s.t.} \|\theta\| \cdot p^{(i)} &\geq 1 \quad \text{if } y^{(i)} = 1 \\ \|\theta\| \cdot p^{(i)} &\leq -1 \quad \text{if } y^{(i)} = 0 \end{aligned}$$



where $p^{(i)}$ is the (signed - positive or negative) projection of $x^{(i)}$ onto $\theta$. Consider the training set above. At the optimal value of $\theta$, what is $||\theta||$?

- 1/4
- [✓ 1/2]
- 1
- 2