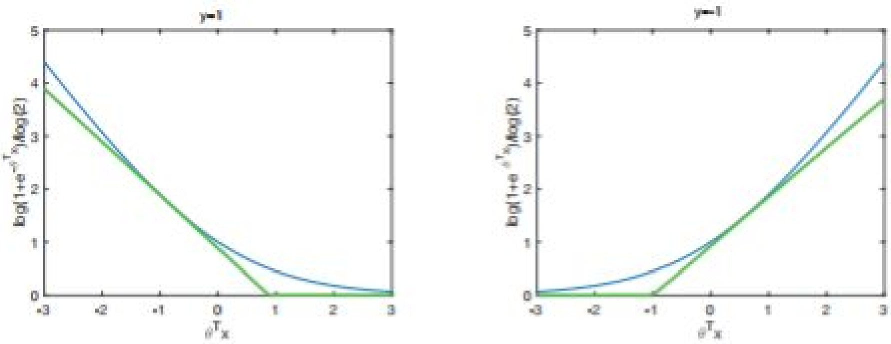
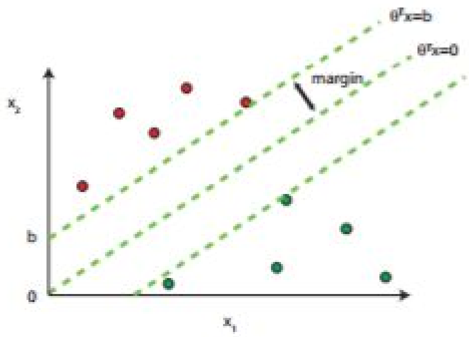
SVMs

- **SVMs**
    - **Choice of cost function**
        - Use the hinge loss function
            - $max(0, 1 - y\theta^T x)$
            - 
        - **Not differentiable (non-smooth)**
        - **Assigns zero penalty to all values of θ which ensure $\theta^T x \geq 1$ when y = 1, and $\theta^T x \leq -1$ when y = -1**
        - So long as $y\theta^T x > 0$, we can scale up θ sufficiently
            - I.e. we can always force $y\theta^T x > 1$ which will allow for 0 cost outputs
        - **To get sensible behavior we have to penalise large values of θ**
            - **By adding this penalty:** $\theta^T\theta = \sum_{j=1}^{n} \theta_j^2$
        - **So our final cost function:**
            - $$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \max(0, 1 - y^{(i)}\theta^T x^{(i)}) + \lambda\theta^T\theta$$
            - **Where λ > 0 is some weighting parameter that we set**

- **Maximising the margin**
    - We have freedom in choosing the line to separate two classes
    - Idea: **choose the line which maximises the margin**
        - The margin is determined by the points which support (the support vectors) the upper and lower boundaries
        - **The cost function we chose, only penalises points for which $y\theta^T x < 1$**
    - $\theta^T x = 0$ **is the decision boundary, and** $\theta^T x = b$ **is a parallel line shifted up by b, which passes through the point** x^i **for which** $b = \theta^T x^{(i)}$
        - $|b| = y^{(i)}\theta^T x^{(i)}$ since $y^{(i)} = 1 \; or \; -1$
        - **The margin =** $\frac{|b|}{\theta^T\theta} = \frac{y^{(i)}\theta^T x^{(i)}}{\theta^T\theta}$

- 

    - **Maximising $\frac{y^{(i)}\theta^T x^{(i)}}{\theta^T\theta}$ is the same as minimising** $-\frac{y^{(i)}\theta^T x^{(i)}}{\theta^T\theta}$
- Trade off exists between classification accuracy and maximised margin
    - $$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\max(0, 1 - y^{(i)}\theta^T x^{(i)}) + \lambda\theta^T\theta$$
    - **Decreasing λ increases accuracy but decreases margin**
    - **Increasing λ decreases accuracy but increases margin**

- **Gradient Descent for SVMs**
    - As we said before, the hinge loss function is not differentiable due to max() (non continuous function)
        - $$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\max(0, 1 - y^{(i)}\theta^T x^{(i)}) + \lambda\theta^T\theta$$
    - **We can use a subgradient though**
        - Subgradient of max(0,1-z) is -1 when z$\leq$1 and 0 when z>1
        - Derivative of 1-$y\theta^T x$ with respect to θ_j is yx_j
        - Putting these together, subgradient of max(0,1-$y\theta^T x$) is
            - $$\begin{cases} -yx_j & \text{when } y\theta^T x \leq 1 \\ 0 & \text{when } y\theta^T x > 1 \end{cases}$$

For $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \max(0, 1 - y^{(i)}\theta^T x^{(i)}) + \lambda\theta^T\theta$, subgradient with respect to θ_j is:

- $2\lambda\theta_j - \frac{1}{m}\sum_{i=1}^{m} y^{(i)}x_j^{(i)}\mathbb{1}(y^{(i)}\theta^T x^{(i)} \leq 1)$ where $\mathbb{1}(y^{(i)}\theta^T x^{(i)} \leq 1) = 1$ when $y^{(i)}\theta^T x^{(i)} \leq 1$ and zero otherwise.

So subgradient descent algorithm for SVMs is:

- Start with some θ
- Repeat:

  for j=0 to n

  $\{tempj := \theta_j - \alpha(2\lambda\theta_j - \frac{1}{m}\sum_{i=1}^{m} y^{(i)}x_j^{(i)}\mathbb{1}(y^{(i)}\theta^T x^{(i)} \leq 1))$

  for j=0 to n $\{\theta_j := tempj\}$

$J(\theta)$ is convex, has a single minimum. Iteration moves downhill until it reaches the minimum

- **Nonlinear decision boundary**
  - Add extra (polynomial) features