Overview

- Support Vector Machines (SVMs)
- Maximising Margin
- Gradient Descent for SVMs
Logistic Regression: Choice of Cost Function

- Hypothesis: \( h_{\theta}(x) = \text{sign}(-\theta^T x) \)
- Parameters: \( \theta \)
- Cost Function: \( J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \log(1 + e^{-y^{(i)}\theta^T x^{(i)}}) \)
- Goal: Select \( \theta \) that minimise \( J(\theta) \)

Loss function: \( \log(1 + e^{-y\theta^T x})/\log(2) \)

gives a small penalty when \( \theta^T x \gg 0 \) and \( y = 1 \), and when \( \theta^T x \ll 0 \) and \( y = -1 \).
SVM: Choice of Cost Function

In an SVM use the “hinge” loss function \( \max(0, 1 - y \theta^T x) \):

Main differences from logistic loss function:

- hinge-loss is not differentiable (“non-smooth”)
- hinge loss assigns zero penalty to all values of \( \theta \) which ensure \( \theta^T x \geq 1 \) when \( y = 1 \), and \( \theta^T x \leq -1 \) when \( y = -1 \)
SVM: Choice of Cost Function

In an SVM use the “hinge” loss function $\max(0, 1 - y\theta^T x)$:

- So long as $y\theta^T x > 0$ then by scaling up $\theta$ sufficiently, e.g. to $10\theta$ or $100\theta$, then we can always force $y\theta^T x > 1$ i.e. $\max(0, 1 - y\theta^T x) = 0$
- To get sensible behaviour we have to penalise large values of $\theta$. We do this by adding penalty $\theta^T \theta = \sum_{j=1}^{\text{n}} \theta_j^2$
- Final SVM cost function is:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y^{(i)}\theta^T x^{(i)}) + \lambda\theta^T \theta$$

where $\lambda > 0$ is a weighting parameter that we have to choose.
Maximising Margin

- We have some freedom in the choice of line used to separate two classes
- Idea: select separating line that maximises the margin
Maximising Margin

- Margin is determined by points that touch (or “support”) the upper and lower boundaries, and not affected by other points.

- Hinge loss function $\max(0, 1 - y\theta^T x)$ assigns zero penalty to points where $y\theta^T x \geq 1$, i.e. value of cost function $J(\theta)$ is determined only by those points for which $y\theta^T x < 1$. 
Maximising Margin

- $\theta^T x = 0$ is the decision boundary. $\theta^T x = b$ is a parallel line shifted up by $b$, and line passing through point $x^{(i)}$ has $b = \theta^T x^{(i)}$
- $|b| = y^{(i)} \theta^T x^{(i)}$ since $y^{(i)} = 1$ or $y^{(i)} = -1$
- Margin\(^1\) equals $\frac{|b|}{\theta^T \theta} = \frac{y^{(i)} \theta^T x^{(i)}}{\theta^T \theta}$. Maximising $\frac{y^{(i)} \theta^T x^{(i)}}{\theta^T \theta}$ is the same as minimising $-\frac{y^{(i)} \theta^T x^{(i)}}{\theta^T \theta}$.
- Hinge loss $\max(0, 1 - y \theta^T x)$ assigns zero penalty to points where $y \theta^T x \geq 1$ i.e. value of cost function $J(\theta)$ is determined only by those points for which $y \theta^T x < 1$

\(^1\)https://en.wikipedia.org/wiki/Distance_from_a_point_to_a_line
Maximising Margin

- There is a trade-off between maximising the margin and classification accuracy.
- Recall cost function $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y^{(i)} \theta^T x^{(i)}) + \lambda \theta^T \theta$
- Decreasing $\lambda$ tends to reduce misclassification but leads to smaller margin.
- Increasing $\lambda$ tends to increase misclassification but leads to larger margin.
SVM Summary

- Hypothesis: $h_\theta(x) = \text{sign}(-\theta^T x)$
- Parameters: $\theta$
- Cost Function: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y^{(i)} \theta^T x^{(i)}) + \lambda \theta^T \theta$
- Goal: Select $\theta$ that minimise $J(\theta)$
Gradient Descent for SVMs

As before, can find $\theta$ using:

- Start with some $\theta$
- Repeat:
  
  Update vector $\theta$ to new value which makes $J(\theta)$ smaller

We can’t use gradient descent directly since

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \max(0, 1 - y^{(i)} \theta^T x^{(i)}) + \lambda \theta^T \theta$$

is not differentiable due to $\max()$. But can still use a subgradient approach.
Gradient Descent for SVMs

- 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 $\theta_j$ is $yx_j$.
- Putting these together, subgradient of $\max(0, 1 - y\theta^T x)$ is

$$
\begin{cases} 
  -yx_j & \text{when } y\theta^T x \leq 1 \\
  0 & \text{when } y\theta^T x > 1
\end{cases}
$$
Gradient Descent for SVMs

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 $\theta_j$ 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 $\theta$
- 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.
As with linear and logistic regression we can add extra “features” e.g. use $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2$, so predict $y = 1$ when $\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 > 0$