COMP9417 - Week 7 Tutorial notes

Kernel Methods

Primal and Dual Algorithms

- The dual view of a problem is simply just another way to view a problem mathematically
- Instead of pure parameter based learning (i.e minimising a loss function, etc), dual algorithms introduce instance-based learning

In the primal problem, we typically learn parameters:

$$w \in \mathbb{R}^p$$

meaning we learn parameters for each of the p features in our dataset

In the dual problem, we typically learn parameters:

$$\alpha \quad \text{for } i \in [1, n]$$

meaning we learn parameters for each of the n data-points

α_i represents the importance of a data point (x_i, y_i)

The Dual/Kernel Perceptron

Recall the primal perceptron:

converged $\leftarrow 0$

while not converged do

  converged $\leftarrow 1$

  for $x_i \in X, y_i \in y$ do

If we define the number of iterations the perceptron makes as $k \in \mathbb{N}^+$ and assume $\eta = 1$. We can derive an expression for the final weight vector $w^{(k)}$:

if $y_i w \cdot x_i \leq 0$ then

  $w \leftarrow w + \eta y_i x_i$

  Converged $\leftarrow 0$

 end if

 end for

end while

$$w^{(k)} = \sum_{i=1}^{N} \sum_{j=1}^{k} 1\{y_i w^{(j)} x_i \leq 0$$

We can simplify our expression and take out the indicator variable:

$$w^{(k)} = \sum_{i=1}^{N} \sum_{j=1}^{k} 1\{y_i w^{(k)} x_i \leq 0\} y_i x_i$$
$$= \sum_{i=1}^{N} \alpha_i y_i x_i$$

where α_i is the number of times the perceptron makes a mistake on a data point (x_i, y_i).

If we sub in $w^{(k)} = \sum_{i=1}^{N} \alpha_i y_i x_i$. We get the algorithm for the dual perceptron.

Converged $\leftarrow 0$

while not converged do

 Converged $\leftarrow 1$

 for $x_i \in X, y_i \in y$ do

  if $y_i \sum_{j=1}^{N} \alpha_j y_j x_j \cdot x_i \leq 0$ then

   $\alpha_i \leftarrow \alpha_i + 1$

   Converged $\leftarrow 0$

  end if

 end for

end while

$\boxed{\text{Gram Matrix}}$ $- G = X^T X$

$$G = \begin{bmatrix} \langle x_1, x_1 \rangle & \langle x_1, x_2 \rangle & \cdots & \langle x_1, x_n \rangle \\ \langle x_2, x_1 \rangle & \langle x_2, x_2 \rangle & \cdots & \langle x_2, x_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n, x_1 \rangle & \langle x_n, x_2 \rangle & \cdots & \langle x_n, x_n \rangle \end{bmatrix}$$

$$G_{i,j} = \langle x_i, x_j \rangle$$

Transformations

How do we go about solving non-linearly separable datasets with linear classifiers?

Project them to higher dimensional spaces through a transformation $\phi: \mathbb{R}^p \to \mathbb{R}^k$