19T3: COMP9417 Machine Learning and Data Mining

Lectures: Kernel Methods
Topic: Questions from lecture topics

Introduction

Some questions and exercises from the course lectures covering "Kernel Methods", focusing on Vapnik's Support Vector Machine (SVM) for classification tasks.

Question 1 Review the development of the Perceptron training algorithm on slides 13-18 of the lecture " Kernel Methods ". Now compare this to the algorithm for Perceptron training *in dual form* introduced on slides 24-25 of the lecture "Kernel Methods". Basically, the two algorithms differ only around lines 6–7. Provide an explanation of how the dual version of the algorithm relates to the original.

Question 2 The Support Vector Machine is a essentially an approach to learning linear classifiers, but uses a alternative objective function to methods we looked at before, namely *maximising the margin.* Learning algorithms for this problem typically use quadratic optimization solvers, but it is possible to derive the solution manually for a small number of support vectors.

Here is a toy data set of three examples shown as the matrix $\mathbf{X}$, of which the first two are classified as positive and the third as negative, shown as the vector $\mathbf{y}$. Start by constructing the *Gram matrix* for this data, incorporating the class labels, i.e., form the matrix $\mathbf{X}'\mathbf{X}'^{\mathrm{T}}$. Then solve to find the support vectors, their Lagrange multipliers α, then determine the weight vector $\mathbf{w}$, threshold t and the margin m.

$$\mathbf{X} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 0 & 1 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} +1 \\ +1 \\ -1 \end{pmatrix} \qquad \mathbf{X}' = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 0 & -1 \end{pmatrix}$$

Background To find a maximum margin classifier requires finding a solution for $\mathbf{w}$, t and margin m. For this we can use the following steps (refer to slides 50-56 from the "Kernel Methods" lecture):

1. set up Gram matrix for labelled data
2. set up expression to be minimised
3. take partial derivatives
4. set to zero and solve for each multiplier
5. solve for $\mathbf{w}$
6. solve for t
7. solve for m

Question 3 You are given 4 examples from a text classification problem where the feature values are simple word counts. Furthermore, all the examples are support vectors, with coefficients $\alpha_i = 1$. Here are the support vectors:

goal	referee	anti	campaign	ban	national	Class
2	0	1	0	4	1	+1
0	3	0	1	0	2	+1
1	0	3	2	0	2	-1
0	2	2	4	0	2	-1

Use the classification rule

$$\hat{y} = \sum_{\mathbf{x}_i \text{ is a support vector}} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}$$

to classify the example $\mathbf{x} = (1, 1, 0, 0, 3, 3)$. What is the class $\hat{y}$ of this example ? Now try this one and determine its class: $\mathbf{x} = (0, 1, 2, 3, 1, 1)$. (It's probably easier to write a short script to do this.)

Question 4 You are told that the "kernel trick" means that a non-linear mapping can be realised from the original data representation to a new, implicit feature space simply by defining a kernel function on dot products of pairs of instances from the original data. To see why this is so, you take two instances $\mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\mathbf{y} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}$, and take their dot product $\mathbf{x} \cdot \mathbf{y}$ and obtain the answer 7. Clearly, raising this dot product to the power of two will give $(\mathbf{x} \cdot \mathbf{y})^2 = 49$.

Now expand out this expression to show that this is the same answer you would have obtained if you had simply done a set of feature transformations on the original data.

Question 5 Generate at least one example from each of the two classes in the original feature space (left of diagram below) and apply the feature transformation (at top of diagram below) to show that the transformation gives rise to a linear separating hyperplane in the new feature space (right of diagram below).

$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := \left(x_1^2, x_2^2, \sqrt{2}x_1x_2\right)$$