Tutorial 05 — Answers

(Version 1.1)

1. (a) A maximum margin classifier is one which learns a decision boundary which represents the largest separation between classes of data, and is of equal distance between the two classes.

   This minimises the generalization loss, reducing the probability of data being misclassification.

   (b) A dataset, comprising input vectors $\mathbf{x}$, may not be linearly separable in the input space, but could be separable in some higher dimensional space.

   We can find this hyperplane by transforming the data via some feature vector $F(\mathbf{x})$, to this higher dimensional space.

   However, this transformation (computing new coordinates in the higher dimensional space), can be computationally expensive.

   Instead, kernel function is used to compute the inner product between two (transformation) feature vectors, without explicitly computing the transformation.

   $$K(\mathbf{x}, \mathbf{z}) = F(\mathbf{x}) \cdot F(\mathbf{z})$$

   This is known as *the kernel trick*.

2. In an SVM, the maximum margin objective function can be defined in terms of the *inner-product between input data*.

   $$\arg\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

   As such, we can find linear separator, with maximum margin, in the higher dimensional feature space by solving

   $$\arg\max_{\alpha} \sum_{i} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j F(\mathbf{x}_i) \cdot F(\mathbf{x}_j)$$

   or in a computationally inexpensive way, by using a kernel:

   $$\arg\max_{\alpha} \sum_{i} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j))$$

3. The idea of a soft margin classifier extends what is possible with a support vector classifier.

   (a) When there is noise in the training data, it may be impossible to find a maximum margin classifier that separates the two classes. The idea of a soft margin classifier allows us to stil create an SVM classifier under these conditions, by relaxing the conditions on what counts as a good classifier. This relaxation translates into changing the conditions on the optimization of the margin.

(b) The optimization problem for a regular SVM is stated in terms of minimizing:

$$\frac{1}{2}||w||^2$$

For a soft margin classifier, we adapt this to minimize:

$$\frac{1}{2}||w||^2 + C\sum_i \xi_i$$

The ξ_i are the slack variables, and their sum limits the amount of data which is ignored when drawing the boundary between classes (allowing them to be miss-classified). C limits this total slack.

4. Support vector regression combines two ideas, that of the relationship between a linear classifier and linear regression, and the idea of slack variables.

   For the first idea that is used is that the linear regression for n variables involves finding the best n-dimensional hyperplane that fits the data. The resulting "line" (a line is a 1-dimensional hyperplane) divides the data into two classes — those above and below the line. Where $y = \mathbf{w}.\mathbf{x}$, the two classes are the ones in which $y > \mathbf{w}.\mathbf{x}$ and $y < \mathbf{w}.\mathbf{x}$.

   This duality exists for any classifier — the classifier boundary is the solution to a regression problem — and for any regression problem — the solution to a regression problem is a classifier boundary.

   Support vector regression exploits this idea — the solution to an SVM optimization gives you a surface which is solution to a regression problem — and uses SVM methods to create the regression model.

   Now if this was the only idea, the regression would be trying to fit all the data points. If the data was noisy, this would mean a good chance of overfitting. So support vector regression also uses slack variables. Just as in soft margin classification, the slack variables allow the optimization to ignore some variables, giving a more robust (in terms of generalisation) solution.

Version list

- Version 1.0, January 30th 2020.
- Version 1.1, January 11th 2021.