Notes on Kernel machines

Overview (Definitions)

Firstly, what is a Kernel machine? This section contains notes mostly taken from Wikipedia in order to help break down the concept of kernel machines:

"Kernel machines are a class of algorithms for pattern analysis, whose best-known member is the support-vector machine." Pattern analysis refers to tasks where we want to find and study general types of relations such as classifications, correlations, principal components, rankings, and clusters. We will focus mainly on classification on regression contexts. For many algorithms that solve pattern analysis tasks, the data in raw representation have to be transformed to feature vector representations via user-specified feature maps. Rather than a feature map, kernel methods require a user-specified kernel instead. This is a similarity function over all pairs of data points computed using inner products.

Kernel methods get their name from their use of kernel functions, which enable them to operate in a high-dimensional, but **Implicit feature space** without ever computing coordinates of the data in that space, but rather computing **inner products between the images of all pairs of data in the feature space**. This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the **kernel trick.**

Kernel methods can be thought of as **instance-based learners.** That is, rather than learning some fixed set of parameters corresponding to the features of their inputs, they instead remember the i-th training example $(\boldsymbol{x_i}, \boldsymbol{y_i})$ and learn for it a corresponding weight w_i. Prediction for unlabeled inputs is treated by the application of a **similarity function k, called a kernel,** between the unlabeled input **x'** and each training input $\boldsymbol{x_i}$**.** For example, a kernelized binary classifier typically computes a weighted sum of similarities.

$$\hat{y} = sgn \sum_{i=1}^{n} w_i y_i k(\boldsymbol{x_i}, \boldsymbol{x'})$$

Where $y \in \{-1, +1\}$ is the kernelized binary classifier's predicted label for the unlabeled input **x'** $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is the kernel function that measures similarity between the pair of inputs $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{X}$. The sum ranges over n labelled examples in the classifiers training set. The sign function determines whether the predicted classification comes out positive or negative.

Support vector machines are a supervised learning model which, using associated learning algorithms, analyses data for classification and regression analysis. SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. SVM maps training examples to points in space so as to maximise the width of the gap between the two categories. New examples are then mapped to whichever side they fall on. SVMs can produce non-linear decision boundaries using the **kernel trick.** The kernel trick is used to implicitly map inputs to high-dimensional feature spaces, such that the data points become separable. The SVM was developed at Bell Laboratories by Vladimir Vapnik and his colleagues.

I can now understand intuitively what a Kernel machine is, but to better understand it mathematically, I must first learn about what kernels themselves are. We can then approach learning about support vector machines in more detail, and then the kernel trick and Mercers theorem. From there I can begin the lecture notes and practical skills material.

Good example of the kernel trick on medium: https://towardsdatascience.com/kernel-function-6f1d2be6091#:~:text=In%20machine%20learning%2C%20a%20%E2%80%9Ckernel,2).

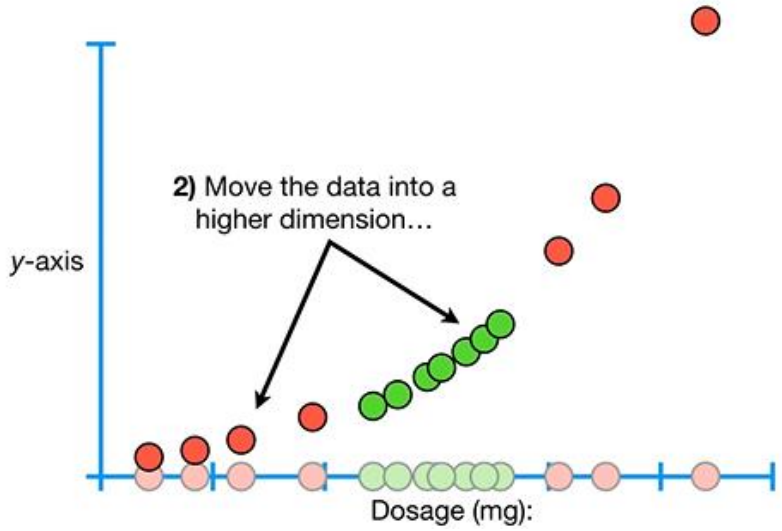
Notes on SVM: (Based on StatQuest video series on Support Vector Machines)

Maximum margin classifiers place the decision threshold at the midpoint between the data points at the edge of the data. This isn't good as the classifier is too sensitive to outliers. Allowing misclassifications allows the classifier to perform better. Lower variance but higher bias.

When we allow misclassifications, the distance between the observations and the threshold is called a **soft margin**. We use cross validation to determine how many misclassifications to allow inside of the soft margin to get the best classification. When we use a soft margin for classification, that is called a **soft margin classifier** or **support vector classifier.**

But support vector classifiers don't work well when the classes in data overlap:

In this case, wherever we put the classifier we will make a lot of misclassifications. This is where Support Vector Machines come in. Instead trying to find a support vector in 1 dimension, we can cast the data into a higher dimension. For example, adding a $Dosage^2$ variable:



SVM uses **kernel functions to systematically find support vector classifiers in higher dimensions**. The example above uses the **Polynomial Kernel** which computes the relationships between each pair of observations in d dimensions. So, when d = n the Polynomial Kernel computes the relationships between each pair of observations in n-Dimensions. Another commonly used kernel function is the Radial Basis Function (RBF) kernel.

For degree d polynomials, the polynomial kernel is defined as

$$K(x, y) = (x \cdot y + c)^d$$

Let's look at practical examples of how the polynomial kernel is used. Let x and y be 1-dimensional vectors, so just single numbers coming from a 1-vairable training dataset. Renaming x and y to a and b for readability and letting $c = 1$ and $d = 2$:

$$(a \times b + 1)^2 = (a \times b + 1)(a \times b + 1)$$
$$= 2ab + a^2b^2 + 1$$
$$= \left(\sqrt{2}a, a^2, 1\right) \cdot (\sqrt{2}b, b^2, 1)$$

I found more generalized mathematical definition of the polynomial kernel here:

https://en.wikipedia.org/wiki/Polynomial_kernel#:~:text=In%20machine%20learning%2C%20the%20polynomial,learning%20of%20non%2Dlinear%20models.

Polynomial Kernel is commonly used in Kernel machines for Natural Language Processing tasks. The Radial basis function kernel is the most popular kernel function used in kernel machine learning algorithms. It is used for SVM classification. It is defined as:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(\frac{\| \boldsymbol{x} - \boldsymbol{x}' \|^2}{2\sigma^2}\right)$$

Since the value of RBF kernel decreases with distance and ranges between zero and one (when **x = x'),** it has a ready interpretation as a similarity measure. More on the RBF here:

https://en.wikipedia.org/wiki/Radial_basis_function_kernel

To find an optimal support vector classifier using a soft-margin (so we're assuming data are not linearly seperable), SVM uses the hinge loss function.