Multilayer perceptrons, radial basis functions and support vector machines

Jose A. Naranjo
Department of Engineering
Universidad del Pacífico, Lima, Perú
naranjo.sja@alum.up.edu.pe

1 Introduction

The concepts of Multilayer perceptrons, radial basis functions and support vector machines are related in a chronological, historical structure. To understand these concepts we will build upon the most basic required knowledge as separated and encapsulated as possible. The first distinctions are between the fields of knowledge.

1.1 Machine learning

Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. In 1959, Arthur Samuel defined machine learning as a "field of study that gives computers the ability to learn without being explicitly programmed". Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions.

The notion of a machine being able to learn can be further separated the same way a human learns, whether from an initial experience or a past experience. Initial experiences incite learning in a way where the output is unknown but becomes apparent because of similarities and cause-effect results which are heavily related to confirmation bias or unexplored relationships. Past experiences, in contrast, provide a known outcome which can be set as the desired outcome and

thus learn by finding how close the actual outcome is to the desired outcome. This could not be possible if the desired outcome were unknown.

1.2 Supervised learning

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consists of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations. Learning whilst having a desired outcome allows for a machine to be able to classify elements.

1.3 Classification

Classification is the problem of identifying to which of a set of categories (subpopulations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. Classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier.

A classification problem involves being able to separate the feature space, that is, the abstract space in which the data is presented, into partitions. The easiest way to solve this problem is by drawing lines around the groups of information that correspond to a certain class. These lines are represented mathematically through functions, yet some sets of data are not separable by using a single function and some are not separable by using linear functions at all.

1.4 Linear separability

Linear separability is a geometric property of a pair of sets of points. This is most easily visualized in two dimensions by thinking of two sets of points of different classes. These two sets are linearly separable if there exists at least one line in the plane that is able to separate the two different classes of points. For example, figure 1 shows a dataset that is not linearly separable.

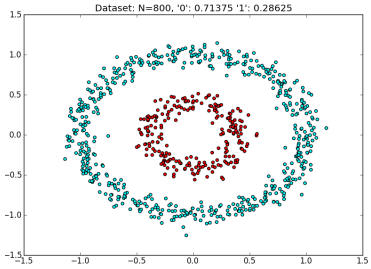

Figure 1: A dataset that is not linearly separable

This idea immediately generalizes to higher-dimensional Euclidean spaces if line is replaced by hyperplane as we will discuss later. Yet, with these concepts handy we can introduce the notion of using systems to classify and separate data.

2 Multilayer perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs.

2.1 Artificial neural network

An artificial neural network (ANN) is a network inspired by biological neural networks (the central nervous systems of animals, in particular the brain) which are used to estimate or approximate functions that can depend on a large number of inputs that are generally unknown. There is no single formal definition of what an artificial neural network is. However, a class of statistical models may commonly be called "neural" if it contains sets of adaptive weights and is capable of approximating non-linear functions of their inputs. The adaptive weights can be thought of as connection strengths between neurons, which are activated during training and prediction.

2.2 Feedforward neural network

A feedforward neural network is an artificial neural network wherein connections between the units do not form a cycle. It was the first and simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes.

An MLP consists of three or more layers (an input and an output layer with one or more hidden layers) of nonlinearly-activating nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron with a nonlinear activation function.

2.3 Activation function

In artificial neural networks, the activation function is usually an abstraction representing the rate of action potential firing in the cell. In its simplest form, this function is binary—that is, either the neuron is firing or not. The function looks like $\phi(v_i) = U(v_i)\phi(v_i) = U(v_i)$, where U is the Heaviside step function (a discontinuous function whose value is zero for negative argument and one for positive argument). There are many types of activation functions, among the most popular ones the identity, binary step, logistic, sigmoid, sinusoid, ramp and gaussian functions.

MLP utilizes backpropagation for training the network, by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result.

2.4 Backpropagation

Backpropagation is a common method of training artificial neural networks used in conjunction with an optimization method such as gradient descent. The method calculates the gradient of a loss function with respect to all the weights in the network. The gradient is fed to the optimization method which in turn uses it to update the weights, in an attempt to minimize the loss function. It requires a known, desired output for each input value in order to calculate the loss function gradient. It is therefore usually considered to be a supervised learning method. It is a generalization of the delta rule to multi-layered feedforward networks and it requires that the activation function used by the artificial neurons be differentiable.

MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable.

2.5 Standard linear perceptron

The perceptron is an algorithm for supervised learning of binary classifiers: functions that can decide whether an input (represented by a vector of numbers) belongs to one class or another. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm

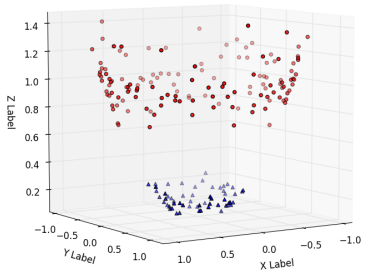

Figure 2: A dataset that is separable in three dimensions

allows for online learning, in that it processes elements in the training set one at a time. The perceptron algorithm was invented in 1957 at the Cornell Aeronautical Laboratory by Frank Rosenblatt; its first implementation, in custom hardware, was one of the first artificial neural networks to be produced. In the modern sense, the perceptron is an algorithm for learning a binary classifier: a function that maps its input x (a real-valued vector) to an output value $f(x)$ (a single binary value).

MLPs were a popular machine learning solution in the 1980s, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software, but have since the 1990s faced strong competition from the much simpler support vector machines which use kernels to solve complex classifications problems. One of the most common and useful types of kernel is an implementation of the radial basis function.

3 Radial basis function

In perceptron-type networks, the activation of hidden units is based on the dot product between the input vector and a weight vector. Radial basis functions are networks where the activation of hidden units is based on the distance between the input vector and a prototype vector. In mathematical terms, a radial basis function (RBF) is a real-valued function whose value depends only on the distance from a point c, called a center, so that $\phi(\mathbf{x}, \mathbf{c}) = \phi(\|\mathbf{x} - \mathbf{c}\|)\phi(\mathbf{x}, \mathbf{c}) = \phi(\|\mathbf{x} - \mathbf{c}\|)$. The norm is usually Euclidean distance, although other distance functions are also possible. Sums of radial basis functions are typically used to approximate given functions. This approximation process can also be interpreted as a simple kind of feed-forward neural network, consisting of a hidden layer of radial kernels and an output later of linear neurons. They can also be used as kernels in support vector classification.

Figure 2 shows the same dataset from figure 1 with an added dimension which allows it to be separable. As we discussed earlier, lines can be used to separate

points in the Eucledian space (two dimensions) but when working in an p-dimensional space, we can use the concept of hyperplane.

3.1 Hyperplane

In a p-dimensional space, a hyperplane is a flat affine subspace of hyperplane dimension $p-1$. For instance, in two dimensions, a hyperplane is a flat one-dimensional subspace—in other words, a line. In three dimensions, a hyperplane is a flat two-dimensional subspace—that is, a plane. In $p > 3$ dimensions, it can be hard to visualize a hyperplane, but the notion of a $(p-1)$-dimensional flat subspace still applies. Figure 2 can be separated by placing a hyperplane between both classes of points.

3.2 Kernel function

A kernel is a function that quantifies the similarity of two observations. For instance, we could simply take the kernel K as:

$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{i,j} x_{i'j}$

This is called a linear kernel because the features are linear. It essentially quantifies the similarity of a pair of observations using Pearson correlation. However, one could replace the sum with the quantity

$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{i,j} x_{i'j})^d$

This is known as a polynomial kernel of degree d. Using this kernel instead of the linear kernel leads to a much more flexible decision boundary. It amounts to fitting a support vector in a higher-dimensional space involving polynomials of degree d rather than in the original feature space which we have seen in Cover's Theorem to be a useful way to make data more separable. Figure 3 shows a support vector machine using a polynomial kernel of $k = 3$ to help classify a problem of non-linearly-separable data.

There are many types of kernels that can be used to fit support vectors more appropiately to the nature of the data. However, this effort requires moving around the dimensions which as we will see later can be circunvented by using some tricks.

4 Support vector machine

A support vector machine (SVM) is a supervised learning model with associated learning algorithms that analyzes data used for classification and regression

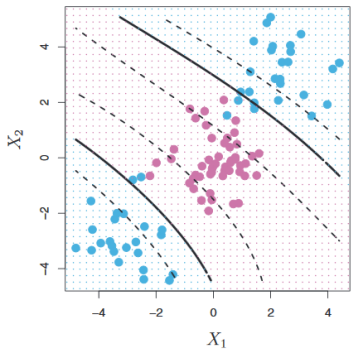

Figure 3: A support vector machine using a polynomial kernel of degree 3

analysis. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. The support vector machine is a generalization of a simple and intuitive classifier called the maximal margin classifier.

4.1 Maximal margin classifier

In general, if our data can be perfectly separated using a hyperplane, then there will in fact exist an infinite number of such hyperplanes. This is because a given separating hyperplane can usually be shifted a tiny bit up or down, or rotated, without coming into contact with any of the observations. In order to construct a classifier based upon a separating hyperplane, we must have a reasonable way to decide which of the infinite possible separating hyperplanes to use. Figure 4 shows two hyperplanes that are equally good at separating a dataset.

A natural choice is the maximal margin hyperplane, which is the separating hyperplane that is farthest from the training observations. That is, we can compute the perpendicular distance from each training observation to a given separating hyperplane; the smallest such distance is the minimal distance from the observations to the hyperplane, and is known as the margin. The maximal margin hyperplane is the separating hyperplane for which the margin is largest—that is, it is the hyperplane that has the farthest minimum distance

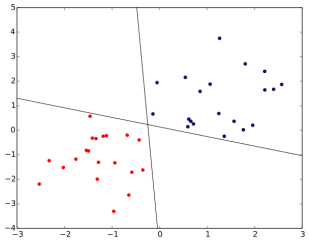

Figure 4: A dataset divided by two equally good hyperplanes

to the training observations. We can then classify a test observation based on which side of the maximal margin hyperplane it lies. This is known as the maximal margin classifier. We hope that a classifier that has a large margin on the training data will also have a large margin on the test data, and hence will classify the test observations correctly.

4.2 Kernel trick

For a dataset with p features (p-dimensional), SVMs find an $p-1$-dimensional hyperplane to separate it (let us say for classification). Thus, SVMs perform very badly with datasets that are not linearly separable. But, quite often, it's possible to transform not-linearly-separable datasets into a higher-dimensional datasets where it becomes linearly separable. Unfortunately, quite often, the number of dimensions you have to add (via transformations) depends on the number of dimensions you already have. For datasets with a lot of features, it becomes next to impossible to try out all the interesting transformations.

Thankfully, the only thing SVMs need to do in the (higher-dimensional) feature space (while training) is computing the pair-wise dot products. For a given pair of vectors (in a lower-dimensional feature space) and a transformation into a higher-dimensional space, there exists a function (the kernel function) which can compute the dot product in the higher-dimensional space without explicitly transforming the vectors into the higher-dimensional space first. This is what's called the kernel trick.

All in all, support vector machines are useful for complex classification problems involving few features yet highly unseparable data points which can be translated into a higher dimension and then applied a function that creates a hyperplane in the shape of whichever kernel function is used in its implementation.

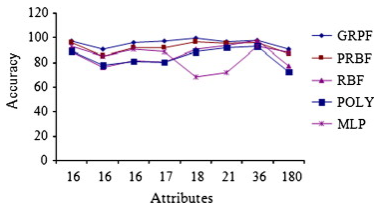

Figure 5: A comparison of SVM using RBF and other kernels and MLP

5 Conclusions

Multilayer perceptrons, radial basis functions and support vector machines are all classification and regression tools that have evolved sequentially from one another yet haven't replaced each other in the process. They are useful for different purposes and yield different results that depend on the nature of the data that is to be classified or regressed.

Figure 5 shows a comparison between support vector machines using different types of kernels, including radial basis function, and multilayer perceptrons. The graph is comparing accuracy of the model's ability to classify correctly and the number of attributes which determine the model's capacity to scale. In terms of scalability and accuracy, SVMs tend to be more accurate than MLPs and they both tend to scale badly, especially considering the RBF kernel and MLP.

E.A. Zanaty explains in his article, "Support Vector Machines (SVMs) versus Multilayer Perception (MLP) in data classification" that "comparing the classification accuracy of the support vector machine to the multilayer neural networks learning algorithms, it is obvious from that support vector machines with the proposed new kernel function (GRPF) accomplishes better accuracy than multilayer networks, especially in high dimension data sets. In MLPs classifiers, the tested data sets need more hidden units and the complexity is controlled by keeping the number of these units small, whereas the SVMs complexity does not depend on the dimension of the data sets. SVMs based on the minimization of the structural risk, whereas MLP classifiers implement empirical risk minimization. So, SVMs are efficient and generate near the best classification as they obtain the optimum separating surface which has good performance on previously unseen data points. However, the main difference is in the complexity of the networks. The MLP network implementing the global approximation strategy usually employs very small number of hidden neurons. On the other side the SVM is based on the local approximation strategy and uses large number of hidden units. The great advantage of SVM approach is the formulation of its learning problem, leading to the quadratic optimization task. It greatly reduces the number of operations in the learning mode. It is well seen for large data sets, where SVM algorithm is usually much quicker."

Specifically, there are advantages and disadvantages for each algorithm and fundamental differences between them:

1. A support vector machine has a regularisation parameter, which makes the user think about avoiding over-fitting. It also uses the kernel trick, so you can build in expert knowledge about the problem via engineering the kernel. Thirdly an SVM is defined by a convex optimization problem (no local minima) for which there are efficient methods. Lastly, it is an approximation to a bound on the test error rate, and there is a substantial body of theory behind it which suggests it should be a good idea. The disadvantages are that the theory only really covers the determination of the parameters for a given value of the regularisation and kernel parameters and choice of kernel. In a way the SVM moves the problem of over-fitting from optimising the parameters to model selection. Sadly kernel models can be quite sensitive to over-fitting the model selection criterion. Support vector machines are also bad at scaling to the number of features in the feature space.

2. Radial basis function networks have advantages of easy design, good generalization, strong tolerance to input noise, and very good online learning ability which is attracting more and more attentions in designing time-variant adaptive control systems. RBF tend to train faster than MLP, and more easy to interpret than MLPs and their hidden layers. However, although the RBF is quick to train, when training is finished and it is being used it is slower than a MLP, so where speed is a factor a MLP may be more appropriate.

3. Multilayer perceptrons are advantageous in contrast to regular perceptrons in that they have the capacity to learn from non-linear models, learn in real time through on-line systems and be used for classification and regression. However, this is also valid for RBF and SVM. Perhaps a valuable asset in MLPs is that they are easy to implement and tend to be faster than SVM and RBF. They are perhaps less precise, but in certain scenarios the trade-off between precision and time cost makes MLPs a suitable algorithm. The main disadvantage of MLP is that they can get stuck in local minima due to the nature of the backpropagation algorithm and that gradient descent tends to be the go-to optimization method. Having many local minima results in false convergence and sometimes divergence depending on the step size of the hill-climb and the way the weights were initalized which is stochastically. MLPs also have many parameters such as the number of hidden layers and neurons, number of iterations and learning rate which make it perhaps a little too modular to be able to test and optimize its parameters properly.