Organizing the Case and Contact Details for Law Firms Using Barrister Groupware with Support Vector Machine

Huian and Adler

Abstract— Barrister Groupware is a kind of all-in-one, easy to use solution that is being designed specifically for law firms. Even in the scenario of reviewing a company’s balance sheet, negotiating terms and conditions with the client or drafting a contract, Barrister Groupware will effectively organize the case and contact details, organize the documents and keep us on top of all our cases and deadlines. A most common and robust machine learning technique, Support Vector Machine (SVM) is employed in this application, which facilitates more enhanced understanding of user search pattern, by this means exponentially improving the common search mechanism achieved through indexing. At the same time as the parameters in SVM are not predefined, real time data allows the application to better understand the judgment pattern, and in that way forecast all possible judgments that the on-going case may have.

I. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) performs classification by constructing an N-dimensional hyperplane that optimally separates the data into two categories. SVM models are closely related to neural networks.

Support Vector Machine (SVM) models are a close cousin to classical multilayer perceptron neural networks. Using a kernel function, SVM’s are an alternative training method for polynomial, radial basis function and multi-layer perceptron classifiers in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a non-convex, unconstrained minimization problem as in standard neural network training.

In the parlance of SVM literature, a predictor variable is called an attribute, and a transformed attribute that is used to define the hyperplane is called a feature. The task of choosing the most suitable representation is known as feature selection. A set of features that describes one case (i.e., a row of predictor values) is called a vector. So the goal of SVM modeling is to find the optimal hyperplane that separates clusters of vector in such a way that cases with one category of the target variable are on one side of the plane and cases with the other category are on the other size of the plane. The vectors near the hyperplane are the support vectors.

A. 2-Dimensional Example

Assume we wish to perform a classification, and our data has a categorical target variable with two categories. Also assume that there are two predictor variables with continuous values. If we plot the data points using the value of one predictor on the X axis and the other on the Y axis we might end up with an image such as shown below. One category of the target variable is represented by rectangles while the other category is represented by ovals.
In this idealized example, the cases with one category are in the lower left corner and the cases with the other category are in the upper right corner; the cases are completely separated. The SVM analysis attempts to find a 1-dimensional hyperplane (i.e. a line) that separates the cases based on their target categories. There are an infinite number of possible lines; two candidate lines are shown above. The question is which line is better, and how do we define the optimal line.

The dashed lines drawn parallel to the separating line mark the distance between the dividing line and the closest vectors to the line. The distance between the dashed lines is called the margin. The vectors (points) that constrain the width of the margin are the support vectors. The following figure illustrates this.

An SVM analysis finds the line (or, in general, hyperplane) that is oriented so that the margin between the support vectors is maximized. In the figure above, the line in the right panel is superior to the line in the left panel.

If all analyses consisted of two-category target variables with two predictor variables, and the cluster of points could be divided by a straight line, life would be easy. Unfortunately, this is not generally the case, so SVM must deal with (a) more than two predictor variables, (b) separating the points with non-linear curves, (c) handling the cases where clusters cannot be completely separated, and (d) handling classifications with more than two categories.

II. A KERNEL FUNCTION

The concept of a kernel mapping function is very powerful. It allows SVM models to perform separations even with very complex boundaries. “Kernel functions provide a way to manipulate data as though it were projected into a higher dimensional space, by operating on it in its original space.”

The kernel trick is a way of mapping observations from a general set \( S \) into an inner product space \( V \) (equipped with its natural norm), without ever having to compute the mapping explicitly, in the hope that the observations will gain meaningful linear structure in \( V \). Linear classifications in \( V \) are equivalent to generic classifications in \( S \). The trick to avoid the explicit mapping is to use learning algorithms that only require dot products between the vectors in \( V \), and choose the mapping such that these high-dimensional dot products can be computed within the original space, by means of a kernel function.

For \( x, y \) on \( S \), certain functions \( K(x,y) \) can be expressed as an inner product (in usually a different space). \( K \) is often referred to as a kernel or a kernel function.

III. RADIAL BASIS FUNCTION (RBF)

The kernel function adopted in this application is the Radial Basis Function. A Radial Basis Function (RBF) neural network has an input layer, a hidden layer and an output layer. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron.

RBF networks are similar to K-Means clustering and PNN/GRNN networks. The main difference is that PNN/GRNN networks have one neuron for each point in the training file, whereas RBF networks have a variable number of neurons that is usually much less than the number of training points. For problems with small to medium size training sets, PNN/GRNN networks are usually more accurate than RBF networks, but PNN/GRNN networks are impractical for large training sets.

A. Working of RBF Networks

Although the implementation is very different, RBF neural networks are conceptually similar to K-Nearest Neighbor (k-NN) models. The basic idea is that a predicted target value of an item is likely to be about the same as other items that have close values of the predictor variables. Consider this figure:
Assume that each case in the training set has two predictor variables, $x$ and $y$. The cases are plotted using their $x,y$ coordinates as shown in the figure. Also assume that the target variable has two categories, positive which is denoted by a square and negative which is denoted by a dash. Now, suppose we are trying to predict the value of a new case represented by the triangle with predictor values $x=6$, $y=5.1$. Should we predict the target as positive or negative?

Notice that the triangle is position almost exactly on top of a dash representing a negative value. But that dash is in a fairly unusual position compared to the other dashes which are clustered below the squares and left of center. So it could be that the underlying negative value is an odd case.

The nearest neighbor classification performed for this example depends on how many neighboring points are considered. If 1-NN is used and only the closest point is considered, then clearly the new point should be classified as negative since it is on top of a known negative point. On the other hand, if 9-NN classification is used and the closest 9 points are considered, then the effect of the surrounding 8 positive points may overbalance the close negative point.

An RBF network positions one or more RBF neurons in the space described by the predictor variables $(x,y$ in this example). This space has as many dimensions as there are predictor variables. The Euclidean distance is computed from the point being evaluated (e.g., the triangle in this figure) to the center of each neuron, and a radial basis function (RBF) (also called a kernel function) is applied to the distance to compute the weight (influence) for each neuron. The radial basis function is so named because the radius distance is the argument to the function.

$$\text{Weight} = \text{RBF}(\text{distance})$$

The further a neuron is from the point being evaluated, the less influence it has.

### B. Training RBF Networks

The following parameters are determined by the training process:

1. The number of neurons in the hidden layer.
2. The coordinates of the center of each hidden-layer RBF function.
3. The radius (spread) of each RBF function in each dimension.
4. The weights applied to the RBF function outputs as they are passed to the summation layer.

Various methods have been used to train RBF networks. One approach first uses K-means clustering to find cluster centers which are then used as the centers for the RBF functions. However, K-means clustering is a computationally intensive procedure, and it often does not generate the optimal number of centers. Another approach is to use a random subset of the training points as the centers.

### IV. OTHER FEATURES IN THE APPLICATION

In addition to the parlance of the technology so far mentioned, the application includes several other modules that eases the workload of barristers by automating tasks.

Staff’s are able to add documents to existing information from external sources including emails, enabling us to view incoming correspondence and so obtain a more complete picture of the status of the matter.

The software supports a wide range of functionality such as Contact Management, Case Management, Instant Messaging service, Phone Messaging, E-mailing and an easy to use, flexible payroll system. Barrister Groupware:

- Provides a tool for analysis of workflow, case status, types of cases handled.
- Sends E-mails.
- Notifies clients and juniors with a text message on the following day’s hearing.
- Provides a chat room that instantly connects everyone in the office.
- Calculates and deducts pay information automatically which are stored for future reference and reporting.

The complete version of the application comes with detailed modules such as General Ledgers, Work in Process Reports, Country Local Rules, Judicial Council Forms and so on, which would be carefully studied and integrated based upon the requirements of the customer.

### REFERENCES