Ensemble of SVM Classifiers for Spam Filtering

Ángela Blanco
Universidad Pontificia de Salamanca, Spain

Manuel Martín-Merino
Universidad Pontificia de Salamanca, Spain

INTRODUCTION

Unsolicited commercial email also known as Spam is becoming a serious problem for Internet users and providers (Fawcett, 2003). Several researchers have applied machine learning techniques in order to improve the detection of spam messages. Naive Bayes models are the most popular (Androutsopoulos, 2000) but other authors have applied Support Vector Machines (SVM) (Drucker, 1999), boosting and decision trees (Carreras, 2001) with remarkable results. SVM has revealed particularly attractive in this application because it is robust against noise and is able to handle a large number of features (Vapnik, 1998).

Errors in anti-spam email filtering are strongly asymmetric. Thus, false positive errors or valid messages that are blocked, are prohibitively expensive. Several authors have proposed new versions of the original SVM algorithm that help to reduce the false positive errors (Kolz, 2001, Valentini, 2004 & Kittler, 1998). In particular, it has been suggested that combining non-optimal classifiers can help to reduce particularly the variance of the predictor (Valentini, 2004 & Kittler, 1998) and consequently the misclassification errors. In order to achieve this goal, different versions of the classifier are usually built by sampling the patterns or the features (Breiman, 1996). However, in our application it is expected that the aggregation of strong classifiers will help to reduce more the false positive errors (Provost, 2001 & Hershop, 2005).

In this paper, we address the problem of reducing the false positive errors by combining classifiers based on multiple dissimilarities. To this aim, a diversity of classifiers is built considering dissimilarities that reflect different features of the data.

The dissimilarities are first embedded into an Euclidean space where a SVM is adjusted for each measure. Next, the classifiers are aggregated using a voting strategy (Kittler, 1998). The method proposed has been applied to the Spam UCI machine learning database (Hastie, 2001) with remarkable results.

THE PROBLEM OF DISSIMILARITIES REVISITED

An important step in the design of a classifier is the choice of the proper dissimilarity that reflects the proximities among the objects. However, the choice of a good dissimilarity for the problem at hand is not an easy task. Each measure reflects different features of the dataset and no dissimilarity outperforms the others in a wide range of problems. In this section, we comment shortly the main differences among several dissimilarities that can be applied to model the proximities among emails. For a deeper description and definitions see for instance (Cox, 2001).

The Euclidean distance evaluates if the features that codify the spam differ significantly among the messages. This measure is sensible to the size of the emails. The cosine dissimilarity reflects the angle between the spam messages. The value is independent of the message length. It differs significantly from the Euclidean distance when the data is not normalized. The correlation measure checks if the features that codify the spam change in the same way in different emails. Correlation based measures tend to group together samples whose features are linearly related. The correlation differs significantly from the cosine if the mean of the vectors that represents the emails are not zero. This measure is distorted by outliers. The Spearman rank correlation avoids this problem by computing a correlation between the ranks of the features. Another kind of correlation measure that helps to overcome the problem of outliers is the kendall-τ index which is related to the Mutual Information probabilistic measure.

When the emails are codified in high dimensional and noisy spaces, the dissimilarities mentioned above are affected by the `curse of dimensionality' (Aggarwal, 2001 & Martín-Merino, 2004). Hence, most of the dissimilarities become almost constant and the differences among dissimilarities are lost (Hinneburg, 2000 & Martín-Merino, 2005). This problem can be avoided selecting a small number of features before the dissimilarities are computed.

COMBINING DISSIMILARITY BASED CLASSIFIERS

In this section, we explain how the SVM can be extended to work directly from a dissimilarity measure. Next, the ensemble of classifiers based on multiple dissimilarities is presented. Finally we comment briefly the related work.

The SVM is a powerful machine learning technique that is able to deal with high dimensional and noisy data (Vapnik, 1998). In spite of this, the original SVM algorithm is not able to work directly from a dissimilarity matrix. To overcome this problem, we follow the approach of (Pekalska, 2001). First, the dissimilarities are embedded into an Euclidean space such that the inter-pattern distances reflect approximately the original dissimilarity matrix. Next, the test points are embedded via a linear algebra operation and finally the SVM is trained and evaluated. We comment briefly the mathematical details.

Let $D \in R^{nxn}$ be the dissimilarity matrix made up of the object proximities for the training set. A configuration in a low dimensional Euclidean space can be found via a metric multidimensional scaling algorithm (MDS) (Cox, 2001) such that the original dissimilarities are approximately preserved. Let $\mathbf{X}= [x_1,\ldots,x_n]^T \in R^{nxp}$ be the matrix of the object coordinates for the training patterns. Define $\mathbf{B} = \mathbf{X}\,\mathbf{X}^T$ as the matrix of inner products which is related to the dissimilarity matrix via the following equation:

$$\mathbf{B} = -1/2\ \mathbf{J}\,\mathbf{D}^{(2)}\,\mathbf{J} \qquad (1)$$

where $\mathbf{J}= \mathbf{I} - 1/n\ \mathbf{1}\,\mathbf{1}^T \in R^{nxn}$ is the centering matrix, $\mathbf{I}$ is the identity matrix and $\mathbf{D}^{(2)} = (\delta_{ij}^2)$ is the matrix of the square dissimilarities for the training patterns. If $\mathbf{B}$ is positive semi-definite, the object coordinates in the low dimensional Euclidean space R^k can be found through a singular value decomposition (Golub, 1996):

$$\mathbf{X}_k = \mathbf{V}_k\,\mathbf{\Lambda}_k^{1/2}, \qquad (2)$$

where $\mathbf{V}_k \in R^{nxk}$ is an orthogonal matrix with columns the first k eigen vectors of $\mathbf{X}\,\mathbf{X}^T$ and $\mathbf{\Lambda}_k = \mathrm{diag}(\lambda_1 \ldots \lambda_k) \in R^{kxk}$ is a diagonal matrix with λ_i the i-th eigenvalue. Several dissimilarities introduced in section 2 generate inner product matrices B non semi-definite positive. Fortunately, the negative values are small in our application and therefore can be neglected without losing relevant information about the data (Pekalska, 2001).

Once the training patterns have been embedded into a low dimensional Euclidean space, the test pattern can be added to this space via a linear projection (Pekalska, 2001). Next we comment briefly the derivation.

Let $\mathbf{X}_k \in R^{nxk}$ be the object configuration for the training patterns in R^k and $\mathbf{X}_n = [x_1,..,\ x_s]^T \in R^{sxk}$ the matrix of the object coordinates sought for the test patterns. Let $\mathbf{D}_n^{(2)} \in R^{sxn}$ be the matrix of the square dissimilarities between the s test patterns and the n training patterns that have been already projected. The matrix $\mathbf{B}_n \in R^{sxn}$ of inner products among the test and training patterns can be found as:

$$\mathbf{B}_n = -\ ½\ (\mathbf{D}_n^{(2)}\,\mathbf{J} - \mathbf{U}\,\mathbf{D}^{(2)}\,\mathbf{J}) \qquad (3),$$

where $\mathbf{J} \in R^{nxn}$ is the centering matrix and $\mathbf{U} = 1/n\ \mathbf{1}^T\,\mathbf{1} \in R^{sxn}$. The derivation of equation is detailed in (Pekalska, 2001). Since the matrix of inner products verifies

$$\mathbf{B}_n = \mathbf{X}_n\,\mathbf{X}_k^T, \qquad (4)$$

then, $\mathbf{X}_n$ can be found as the least mean-square error solution to (4), that is:

$$\mathbf{X}_n = \mathbf{B}_n\,\mathbf{X}_k\,(\mathbf{X}_k^T\,\mathbf{X}_k)^{-1} \qquad (5)$$

Given that $\mathbf{X}_k^T\,\mathbf{X}_k = \mathbf{\Lambda}_k$ and considering that $\mathbf{X}_k = \mathbf{V}_k\,\mathbf{\Lambda}_k^{1/2}$ the coordinates for the test points can be obtained as:

$$\mathbf{X}_n = \mathbf{B}_n\,\mathbf{V}_k\,\mathbf{\Lambda}_k^{-1/2}, \qquad (6)$$

which can be easily evaluated through simple linear algebraic operations.