Support Vector Machines

Mamoun Awad
University of Texas at Dallas, USA

Latifur Khan
University of Texas at Dallas, USA

INTRODUCTION

The availability of reliable learning systems is of strategic importance, as many tasks cannot be solved by classical programming techniques, because no mathematical model of the problem is available. So, for example, no one knows how to write a computer program that performs handwritten character recognition, though plenty of examples are available. It is, therefore, natural to ask if a computer could be trained to recognize the letter *A* from examples; after all, humans learn to read this way. Given the increasing quantity of data for analysis and the variety and complexity of data analysis problems being encountered in business, industry, and research, demanding the best solution every time is impractical. The ultimate dream, of course, is to have some intelligent agent that can preprocess data, apply the appropriate mathematical, statistical, and artificial intelligence techniques, and then provide a solution and an explanation. In the meantime, we must be content with the pieces of this automatic problem solver. The data miner's purpose is to use the available tools to analyze data and provide a partial solution to a business problem.

The support vector machines (SVMs) have been developed as a robust tool for classification and regression in noisy and complex domains. SVMs can be used to extract valuable information from data sets and construct fast classification algorithms for massive data.

The two key features of support vector machines are the generalization theory, which leads to a principled way to choose a hypothesis, and kernel functions, which introduce nonlinearity in the hypothesis space without explicitly requiring a nonlinear algorithm.

SVMs map data points to a high-dimensional feature space, where a separating hyperplane can be found. This mapping can be carried on by applying the kernel trick, which implicitly transforms the input space into high-dimensional feature space. The separating hyperplane is computed by maximizing the distance of the closest patterns, that is, margin maximization.

SVMs can be defined as "a system for efficiently training linear learning machines in kernel-induced feature spaces, while respecting the insights of generalisation theory and exploiting optimisation theory" (Cristianini & Shawe-Taylor, 2000, p. 93).

Support vector machines have been applied in many real-world problems and in several areas: pattern recognition, regression, multimedia, bio-informatics, artificial intelligence, and so forth.

Many techniques, such as decision trees, neural networks, genetic algorithms, and so on, have been used in these areas; however, what distinguishes SVMs is their solid mathematical foundation, which is based on the statistical learning theory. Instead of minimizing the training error (empirical risk), SVMs minimize the structural risk, which expresses an upper bound on the generalization error, that is, the probability of an erroneous classification on yet to be seen examples. This quality makes SVMs especially suited for many applications with sparse training data.

BACKGROUND

The general problem of machine learning is to search a (usually) very large space of potential hypotheses to determine the one that will best fit the data and any prior knowledge.

"A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P,* if its performance at tasks in *T,* as measured by *P,* improves with experience *E*" (Mitchell, 1997, p. 2).

Machine learning can be categorized into several categories based on the data set and labels of the data set. The data used for learning may be labeled (for example, data might be medical records, where each record reflects the history of a patient and has a label denoting whether that patient had heart disease or not) or unlabeled. If labels are given, then the problem is one of *supervised learning,* in that the true answer is known for a given set of data. If the labels are categorical, then the problem is one of *classification,* for example, predicting the species of a flower given petal and sepal measurements. If the labels are real-valued, then the problem is one of *regression statistics,* for example, predicting property values

from crime, pollution, and so forth. If labels are not given, then the problem is one of *unsupervised learning,* and the aim is to characterize the structure of the data, for example, by identifying groups of examples in the data that are collectively similar to each other and distinct from the other data.

Pattern Recognition

Formally, in pattern recognition, we want to estimate a function $f : R^N \rightarrow \{\pm 1\}$ by using input-output training data, $(x_1, y_1),...,(x_l, y_l) \in R^N \times \{\pm 1\}$, such that f will correctly classify unseen examples $(\mathbf{x},y)$, that is, $f(\mathbf{x})=y$ for examples $(\mathbf{x},y)$ that were generated from the same underlying probability distribution $P(\mathbf{x},\mathrm{y})$ as the training data. Each data point has numerical properties that might be useful to distinguish them and that are represented by $\mathbf{x}$ in $(\mathbf{x},y)$. The y is either +1 or –1 to denote the label or the class to which this data point belongs. For example, in a medical record, $\mathbf{x}$ might be the age, weight, allergy, blood pressure, blood type, disease, and so forth. The y might represent whether the person is susceptible to a heart attack. Notice that some attributes, such as an allergy, might need to be encoded (for example, 1 if the person is allergic to medicine, or 0 if not) in order to be represented as a numerical value.

If we put no restriction on the class of functions that we choose our estimate f from, even a function that does well on the training data, for example, by satisfying $f(x_i) = y_i$ for all $i = 1,...,l$, might not require to generalize well to unseen examples. To see this, note that for each function f and test set $(\bar{x}_1, \bar{y}_1),...,(\bar{x}_{\bar{l}}, \bar{y}_{\bar{l}}) \in R^N \times \{\pm 1\}$ satisfying $\{\bar{x}_1,...,\bar{x}_{\bar{l}}\} \cap \{x_1,...,x_l\} = \{\}$, there exists another function f^* such that $f^*(x_i) = f(x_i)$ for all $i = 1,...,l$, yet $f^*(\bar{x}_i) \neq f(\bar{x}_i)$ for all $i = 1,...,\bar{l}$; that is, both functions, f and f*, return the same prediction for all training examples, yet they disagree on their predictions for all testing examples.

As we are only given the training data, we have no means of selecting which of the two functions (and hence which of the completely different sets of test outputs) is preferable. Hence, only minimizing the training error (or empirical risk),

$$R_{emp}[f] = \frac{1}{l}\sum_{i=1}^{l}\frac{1}{2}|f(x_i) - y_i|, \qquad (1)$$

does not imply a small test error (called risk), averaged over test examples drawn from the underlying distribution $P(x, y)$,

$$R[f] = \int \frac{1}{2}|f(x) - y|\, dP(x, y). \qquad (2)$$

In Equation 1, notice that the error, $f(x_i) - y_i$, is equal to 0 if the data point x_i is correctly classified, because $f(x_i) = y_i$.

Statistical learning theory (Vapnik & Chervonenkis, 1974; Vapnik, 1979), or VC (Vapnik-Chervonenkis) theory, shows that it is imperative to restrict the class of functions that f is chosen from to one that has a capacity suitable for the amount of available training data. VC theory provides

Figure 1A. VC-dimension of H equals the set of all linear dicision surfaces
Figure 1B. Four points cannot be shattered by H