Kernel Methods in Chemoinformatics

Huma Lodhi
Imperial College London, UK

INTRODUCTION

Millions of people are suffering from fatal diseases such as cancer, AIDS, and many other bacterial and viral illnesses. The key issue is now how to design lifesaving and cost-effective drugs so that the diseases can be cured and prevented. It would also enable the provision of medicines in developing countries, where approximately 80% of the world population lives. Drug design is a discipline of extreme importance in chemoinformatics. Structure-activity relationship (SAR) and quantitative SAR (QSAR) are key drug discovery tasks.

During recent years great interest has been shown in kernel methods (KMs) that give state-of-the-art performance. The support vector machine (SVM) (Vapnik, 1995; Cristianini and Shawe-Taylor, 2000) is a well-known example. The building block of these methods is an entity known as the *kernel.* The nondependence of KMs on the dimensionality of the feature space and the flexibility of using any kernel function make them an optimal choice for different tasks, especially modeling SAR relationships and predicting biological activity or toxicity of compounds. KMs have been successfully applied for classification and regression in pharmaceutical data analysis and drug design.

BACKGROUND

Recently, I have seen a swift increase in the interest and development of data mining and learning methods for problems in chemoinformatics. There exist a number of challenges for these techniques for chemometric problems. High dimensionality is one of them. There are datasets with a large number of dimensions and a few data points. Label and feature noise are other important problems. Despite these challenges, learning methods are a common choice for applications in the domain of chemistry. This is due to automated building of predictors that have strong theoretical foundations. Learning techniques, including neural networks, decision trees, inductive logic programming, and kernel methods such as SVMs, kernel principal component analysis, and kernel partial least square, have been applied to chemometric problems with great success. Among these methods, KMs are new to such tasks. They have been applied for applications in chemoinformatics since the late 1990s. KMs and their utility for applications in chemoinformatics are the focus of the research presented in this article. These methods possess special characteristics that make them very attractive for tasks such as the induction of SAR/QSAR. KMs such as SVMs map the data into some higher dimensional feature space and train a linear predictor in this higher dimensional space. The kernel trick offers an effective way to construct such a predictor by providing an efficient method of computing the inner product between mapped instances in the feature space. One does not need to represent the instances explicitly in the feature space. The kernel function computes the inner product by implicitly mapping the instances to the feature space. These methods can handle very high-dimensional noisy data and can avoid overfitting. SVMs suffer from a drawback that is the difficulty of interpretation of the models for nonlinear kernel functions.

MAIN THRUST

I now present basic principles for the construction of SVMs and also explore empirical findings.

Support Vector Machines and Kernels

The support vector machine was proposed in 1992 (Boser, Guyon, & Vapnik, 1992). A detailed analysis of SVMs can be found in Vapnik (1995) and Cristianini and Shawe-Taylor (2000). An SVM works by embedding the input data, $d_i,\ldots,d_n$, into a Hilbert space through a nonlinear mapping, ϕ, and constructing the linear function in this space. Mapping ϕ may not be known explicitly but be accessed via the kernel function described in the later section on kernels. The kernel function returns the inner product between the mapped instances d_i and d_j in a higher dimensional space that is for any mapping $\phi : D \to F,\quad k(d_i,d_j)=\langle \phi(d_i),\phi(d_j)\rangle$. I now briefly describe SVMs for classification, regression, and kernel functions.

Support Vector Classification

The support vector machine for classification (SVC) (Vapnik, 1995) is based on the idea of constructing the maximal margin hyperplane in feature space. This unique hyperplane separates the data into two categories with maximum margin (hard margin). The maximal margin hyperplane fails to generalize well when there is a high level of noise in the data. In order to handle noise, data margin errors are allowed, hence achieving a soft margin instead of a hard margin (no margin errors). The generalization performance is improved by maintaining the right balance between the margin maximization and error. To find a hyperplane, one has to solve a convex quadratic optimization problem. The support vector classifier is constructed by using only the inner products between the mapped instances. The classification function for a new example d is given by $f = \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i c_i k(d_i, d) + b\right)$, where α_i are Lagrange multiplies, $c_i \in \{-1, +1\}$ are categories, and $b \in R$.

Support Vector Regression

Support vector machines for regression (SVR) (Vapnik, 1995) inherit all the main properties that characterize SVC. SVR embeds the input data into a Hilbert space through a nonlinear mapping ϕ and constructs a linear regression function in this space. In order to apply support vector technique to regression, a reasonable loss function is used. Vapnik's ε – insensitive loss function is a popular choice that is defined by $|c - f(d)|_\varepsilon = \max(0, |c - f(d)| - \varepsilon)$, where $c \in R$. This loss function allows errors below some $\varepsilon > 0$ and controls the width of insensitive band. Regression estimation is performed by solving an optimization problem. The corresponding regression function f is given by $f = \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) k(d_i, d) + b$, where $\tilde{\alpha}_i, \alpha_i$ are Lagrange multipliers, and $b \in R$.

Kernels

A kernel function calculates an inner product between mapped instances in a feature space and allows implicit feature extraction. The mathematical foundation of such a function was established during the first decade of the 20th century (Mercer, 1909). A kernel function is a symmetric function $k(d_i, d_j) = k(d_j, d_i)$ for all d_i, d_j and satisfies positive (semi)definiteness, $\sum_{i,j=1}^{n} a_i a_j k(d_i d_j) \geq 0$ for $a_i, a_j \in R$. The $n \times n$ matrix with entries of the form $K_{ij} = k(d_i, d_j)$ is known as the kernel matrix, or the Gram matrix, that is a symmetric positive definite matrix.

Linear, polynomial, and Gaussian Radial Basis Function (RBF) kernels are well-known examples of general purpose kernel functions. Linear kernel function is given by $k_{linear}(d_i, d_j) = k(d_i, d_j) = d_i' d_j$. Given a kernel k, the polynomial construction is given by $k_{poly}(d_i, d_j) = (k(d_i, d_j) + c)^p$. Here, p is a positive integer, and c is a nonnegative constant. Clearly, this incurs a small computational cost to define a new feature space. The feature space corresponding to a degree p polynomial kernel includes all products of at most p input features. Note that for $p = 1$, I get the linear construction. Furthermore, RBF kernel defines a feature space with an infinite number of dimensions. Given a set of instances, the RBF kernel is given by

$$K_{RBF}(d_i, d_j) = \exp\left(\frac{-\|d_i - d_j\|^2}{2\sigma^2}\right).$$

New kernels can be designed by keeping in mind that kernel functions are closed under addition and multiplication. Kernel functions can be defined over general sets (Watkins, 2000; Haussler, 1999). This important fact has allowed successful exploration of novel kernels for discrete spaces such as strings, graphs, and trees (Lodhi, Saunders, Shawe-Taylor, Cristianini, & Watkins, 2002; Kashima, Tsuda, & Inokuchi, 2003; Mahe, Ueda, Akutsu, Perret, & Vert, 2004).

Applications

Given that I have presented basic concepts of SVMs, I now describe the applications of SVMs in chemical domains.

SAR/QSAR analysis plays a crucial role in the design and development of drugs. It is based on the assumption that the chemical structure and activity of compounds are correlated. The aim of mining molecular databases is to select a set of important compounds, hence forming a small collection of useful molecules. The prediction of a new compound with low error probability is an important factor, as the false prediction can be costly