Kernel Method

Kernel methods are a class of machine learning algorithms implemented for many different inferential tasks and application areas (Smola and Schuolkopf, 1998; Shawe-Taylor and Cristianini, 2004; Scholkopf and Burges, 1999).

From: Encyclopedia of Bioinformatics and Computational Biology, 2019

Related terms:

Functional Magnetic Resonance Imaging, Filtration, Machine Learning, Classifier, Support Vector Machine, Electroencephalogram

View all Topics

Learn more about Kernel Method

Kernel Machines: Applications

Italo Zoppis, ... Riccardo Dondi, in Encyclopedia of Bioinformatics and Computational Biology, 2019

Introduction

Kernel methods have received considerable attention in many scientific communities, mainly due to their capability of working with linear inference models, allowing at the same time to identify nonlinear relationships among input patterns (Smola and Schuölkopf, 1998; Shawe-Taylor and Cristianini, 2004; Schölkopf and Burges, 1999). Moreover, the possibility offered by these techniques to work homogeneously with structured data has allowed to cope with different problems, even introducing within the inference process specific domain-knowledge.

The discovery of nonlinear relationships between input patterns and the ability to work with heterogeneous data have always been two important and desirable features to be found in machine learning. At this regard, it is necessary to emphasize that kernel methods not only provided these possibilities in a completely natural way, but they were largely founded on a strong and intuitive mathematics.

For a long time, the most popular approach to solve nonlinear problems was to apply multi-layer perceptron: a model, able to approximate any function (Hornik *et al.*, 1989), that can be trained efficiently by using the back-propagation algorithm. The approach followed by the kernel methods is fundamentally different. The strategy is to design procedures, in the feature space, by addressing observations only through inner product operations. Such algorithms are said to employ a kernel, since the pairwise inner products can be computed directly from the original items, using only the so called *kernel function*. In other terms, whether an algorithm expressing operations which exclusively involve inner products in some feature space, can be implicitly done in the input space by using an appropriate kernel function. A nonlinear version of the considered algorithm can be readily obtained by simply replacing the inner products with the kernel function (this is generally referred as *kernel trick*). In this situation, it is not necessary to know the input transformation, instead the kernel trick is enough to ensure the existence of such transformation between the input patterns and the feature space.

In this article, we will show how to implement this technique for simple tasks. Moreover, successful applications in machine learning literature, bioinformatics and pattern recognition will be briefly described.

> Read full chapter

Kernel Machines: Introduction

Italo Zoppis, ... Riccardo Dondi, in Encyclopedia of Bioinformatics and Computational Biology, 2019

Conclusion

Kernel methods have not only enriched the machine learning research by offering the opportunity to dealing with different tasks and different input structures, but have also provided new perspectives for solving typical problems with a methodology supported by strong intuition and well founded mathematical theory.

The discussion outlined in this article is intended to provide the fundamental idea of these techniques: whenever a procedure is adapted to use only inner products between input examples, then in that procedure, inner products can be replaced by a kernel function. By exploiting the so-called kernel trick, we have seen that one can straightforwardly perform nonlinear transformations of the original patterns into a (generally) higher dimensional feature space – which is consequently nonlinearly related to the input space. Many algorithms in the scientific literature have been “kernelized” (i.e., adapted) in the manner described above.

> Read full chapter

Mass Spectrometry Metabolomic Data Handling for Biomarker Discovery

Julien Boccard, Serge Rudaz, in Proteomic and Metabolomic Approaches to Biomarker Discovery, 2013

Kernel Methods

Kernel algorithms were developed to model strong nonlinear relationships between independent and dependent variables. In that perspective, the original data is transformed from the input space into a higher dimensional feature space by a mapping function, □. This transformation must be carefully achieved to offer a reliable linear model in the feature space that corresponds to a nonlinear solution in the original data space. Kernel functions are applied to map the data in the feature space and the kernel matrix summarizes similarity measurements between pairs of observations. Because the model construction is performed on the kernel matrix instead of the initial data table, it takes advantage of the kernel trick to reduce the computational complexity. The most typical kernels include polynomial and radial basis functions.[92] Kernel extensions of well-established bilinear factor models were implemented, including kernel PCA,[93] kernel PLS,[94,95] and kernel O-PLS.[96]

> Read full chapter

Nonlinear Models of Glucose Concentration

Eleni I. Georga, ... Stelios K. Tigas, in Personalized Predictive Modeling in Type 1 Diabetes, 2018

6.3.1 Basic Concepts of Kernel-Based Regression Models

A key feature of kernel methods is the ability to solve a nonlinear regression problem in the input space as a linear one in a new feature space . Kernel methods transform the input space into a high-dimensional Reproducing Kernel Hilbert Space through a mapping . A positive definite kernel function , such that , is utilized and all computations are expressed in terms of the inner product avoiding working directly in the transformed feature space [19–21]. The Representer Theorem ensures that the output of kernel methods lies in the span of the finite set of kernels centered at the input vectors of the training set and it is expressed by a nonparametric function of the form [20]:

(6.9)

where is the coefficient vector of the model. Since the number of adjustable parameters in Eq. (6.9) equals the size of the training dataset, some form of constraint should be imposed on the error function to avoid overfitting.

Gaussian Processes (GP) [22] and Support Vector Regression (SVR) [23] kernel-based methods are extensively applied to nonlinear system identification. In the case of GP, the output for a new point is estimated from a Gaussian distribution with mean and covariance given by

(6.10)

where is the *j*th component of , with denoting the covariance matrix and the target vector , and the vector . The squared exponential kernel is the default one for GP regression. The noise on the observed values is considered and it is further assumed to be Gaussian distributed with zero mean and constant variance for all . The latter contributes to the total variance of the predictive distribution given by (6.10). Note that the kernel function is evaluated for all possible pairs and resulting in a nonsparse model.

SVR obtains a sparse solution by utilizing an □-insensitive loss function in which the error increases linearly with distance beyond the insensitive region. Errors larger than ±□ are treated by introducing the slack variables and for each data point . The optimization problem is defined as

(6.11)

The model's complexity is controlled by the regularization parameter , which determines the tradeoff between the flatness of the SVR function *f* (i.e., small *w*) and the amount up to which deviations larger than □ are tolerated. Solving the optimization problem, it is found that the prediction for a new point can be made using:

(6.12)

where () are the Lagrange multipliers introduced in the constrained optimization process. The corresponding Karush–Kuhn–Tucker conditions imply that for and that all points lying inside the -tube have .

> Read full chapter

Brain Imaging

Eileanoir B. Johnson, Sarah Gregory, in Progress in Molecular Biology and Translational Science, 2019

3.1.3 Smoothing

Smoothing is performed applying an isotropic Gaussian kernel to an image that averages the signal from neighboring voxels. This process not only improves signal-to-noise ratio but also normally distributes the data, which is essential for parametric statistical testing.[19] The kernel size is determined according to the size of the brain area of interest. The Kernel is Full Width Half Maximum (FWHM), with typical sizes between 4 and 8 mm. When deciding on a kernel size, it is important to recognize that there is a compromise between smoothing the data appropriately to establish normally distributed data and reducing noise within the data, but also losing spatial information and thus the ability to localize the signal. The kernel size is typically decided based on the resolution of the data along with the estimated regions of effect.

> Read full chapter

Kernel Methods: Support Vector Machines

Italo Zoppis, ... Riccardo Dondi, in Encyclopedia of Bioinformatics and Computational Biology, 2019

Abstract

Support Vector Machines (SVMs) and Kernel methods have found a natural and effective coexistence since their introduction in the early 90s. In this article, we will describe the main concepts that motivate the importance of this relationship. In fact SVMs use kernels for learning linear predictors in high dimensional feature spaces. First, we will describe intuitively how this mechanism is realized, introducing the main concepts and definitions, i.e., maximum margin hyperplane, kernels and non-linearly separable problems. Then the main mathematical issues for linear and nonlinear SVM-based classification will be detailed. We will also introduce some important extensions of the SVMs ideas, by considering the Soft Margin Classification, SVM multi-class classification, SVM clustering, and SVM regression.

> Read full chapter

Multivariate Analysis: Classification and Discrimination

G. McLachlan, in International Encyclopedia of the Social & Behavioral Sciences, 2001

4 Flexible Discriminant Rules

A common nonparametric approach to discriminant analysis uses the kernel method to estimate the group-conditional densities $f_i(x)$ in forming an estimate of the Bayes' rule. More recently, use has been made of finite mixture models, mainly normal mixtures, to provide flexible rules of discrimination (Hastie and Tibshirani 1996). Mixture models provide an extremely flexible way of modeling a density function (McLachlan and Basford 1988, McLachlan and Peel 2000), and can be fitted in a straightforward manner via the EM algorithm (McLachlan and Krishnan 1997, McLachlan et al. 1999). Among other work on flexible discrimination, there is the FDA (flexible discriminant analysis) approach based on nonparametric regression (see Hastie and Tibshirani 1996). The generic version of FDA based on smoothing splines proceeds by expanding the predictors in a large (adaptively selected) basis set, and then performing a penalized discriminant analysis in the enlarged space using a linear discriminant or a normal mixture model-based rule. The class of nonlinear regression methods that can be used include additive models, the multivariate adaptive regression spline (MARS) model, projection pursuit regression, and neural networks. In machine learning, there has been increasing attention in the case of two groups given to nonlinear rules based on the foundations of support vector machines, as developed by Vapnik (2000). With this approach, the initial feature space is mapped into a higher dimensional space by choosing a nonlinear mapping and then choosing an optimal separate hyperplane in the enlarged feature space.

A rather different approach to the allocation problem as considered up to now, is to portray the rule in terms of a binary tree. The tree provides a hierarchical representation of the feature space. An allocation is effected by proceeding down the appropriate branches of the tree. The classification and regression tree (CART) methodology of Breiman et al. (1984) has contributed significantly to the growing popularity of tree classifiers; see also the C4.5 decision-tree learning algorithm of Quinlan (1993). In the context of tree classifiers in particular, there has been growing interest in the use of boosting, which is one of the most important recent developments in classification methodology. Algorithms like the Adaboost algorithm of

Freund and Schapire (1996) and the bagging algorithm of Breiman (1996) can often improve performance of unstable classifiers like trees or neural nets by sequentially applying them to reweighted versions of the training data and taking a weighted majority vote of the sequence of classifiers so formed. The test error of the weighted classifier usually does not increase as its size increases, and often is observed to decrease even after the training error reaches zero. Friedman et al. (2000) have exhibited the link of boosting to statistical ideas on additive fitting.

A guide to the performance of a given discriminant rule can be obtained by applying it to the training from which it was formed and noting the proportion of these observations misallocated (the apparent rate). However, it is well known that the apparent error rate gives too optimistic a view on how the rule might perform on future cases; see McLachlan (1992, Chap. 10) for an account of error-rate estimation. One way of correcting for this bias is to use cross-validation, but it produces an estimate that often is more highly variable than the apparent error rate. This led Efron (1983) to consider his so-called .632 estimator, which is a weighted sum of the apparent error rate and the bootstrap error rate at an original data point not in the training set *t*; see Efron and Tibshirani (1997) for an improved version known as the .632+ estimator. The bootstrap methodology can be used also to estimate the variability in the errors and in other quantities of interest, such as the estimated parameters in the form of the estimated rule.

An excellent summary of available software and algorithms for discriminant analysis has been given by the Panel on Discriminant Analysis, Classification and Clustering (1989); see also Huberty (1993) and Michie et al. (1994).

In conclusion, discriminant analysis is concerned with the construction of allocation rules for the assignment of unclassified entities to distinct groups in the situation where there are data on entities of known group of origin (training data). In the past, linear rules such as those based on Fisher's linear discriminant function were commonly used in practice. In recent times with the wide availability of high speed computers, there has been increasing attention given to the development of more complex nonlinear rules such as those based on flexible parametric models, neural networks, support vector machines, and classification trees.

See also: Multivariate Analysis: Discrete Variables (Logistic Regression); Multivariate Analysis: Discrete Variables (Loglinear Models); Multivariate Analysis: Discrete Variables (Overview); Multivariate Analysis: Overview

> Read full chapter

Voxel Based Morphometry

J. Ashburner, K.J. Friston, in Encyclopedia of Neuroscience, 2009

Smoothing

The warped gray matter images are now smoothed by convolving with an isotropic Gaussian kernel. This makes the subsequent voxel-by-voxel analysis comparable to a region of interest approach because each voxel in the smoothed images contains the average amount of gray matter from around the voxel (where the region around the voxel is defined by the form of the smoothing kernel; **Figure 5**). This is often referred to as gray matter density, but it should not be confused with cell packing density measured cytoarchitectonically. Critically, smoothing removes finescale structure from the data that is not conserved from subject to subject. This increases the sensitivity of VBM to differences that are expressed at a larger spatial scale. The smoothing conforms to the matched filter theorem, which states that the smoothing should match the scale of the difference in question. Normally, the smoothing kernel is Gaussian with a full width at half maximum (FWHM) of between 4 and 16 mm. By the central limit theorem, smoothing also has the effect of rendering the data more normally distributed, thus increasing the validity of parametric statistical tests.

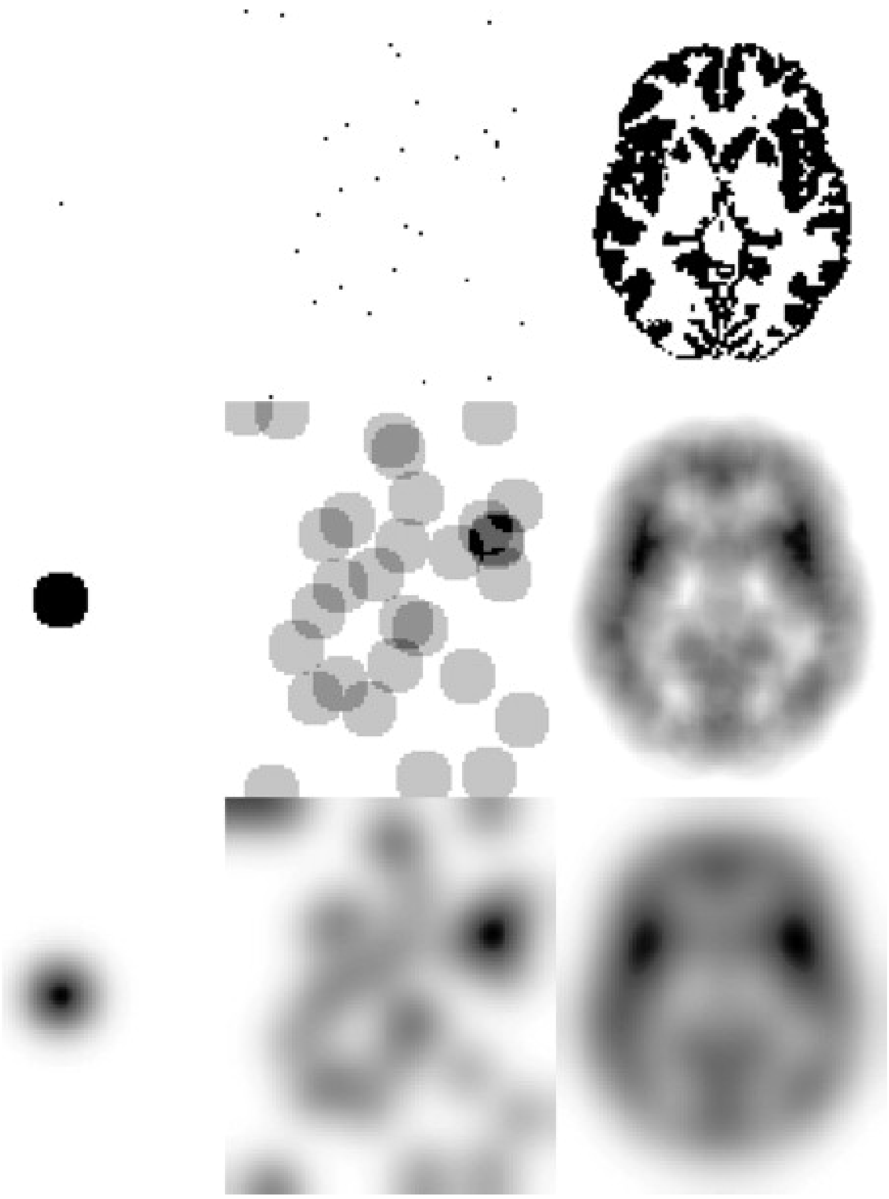

Figure 5. Smoothing effectively converts the images to maps containing a weighted count of the number of gray matter voxels within each region. The top row shows three unsmoothed images. The middle row shows these images after they have been convolved with a circular-shaped kernel. It should be clear that the result is a count of the values within each circular region. The bottom row shows the images after convolving with a Gaussian-shaped kernel. The result is a weighted count of the values around each point, where values in the center are weighted more heavily.

> Read full chapter

Probability Density Estimation

D.W. Scott, in International Encyclopedia of the Social & Behavioral Sciences, 2001

5 Multivariate Densities

Histograms can be constructed in the bivariate case and displayed in perspective, but cannot be used to draw contour plots. Kernel methods are well-suited for this purpose. Given a set of data (x_1, y_1), …, (x_n, y_n), the bivariate normal kernel estimator is given by

A different smoothing parameter can be used in each variable.

For three and four-dimensional data, the kernel estimator can be used to prepare advanced visualization renderings of the data. Scott (1992) provides an extension set of examples.

> Read full chapter

Smoothness Variance Estimate and Its Effects on Probability Values in Statistical Parametric Maps

J-B. POLINE, … K.J. FRISTON, in Quantification of Brain Function Using PET, 1996

III RESULTS

We first validate the approximate expression for the variance of *W* by simulating 1000 3D SPMs (uncorrelated Gaussian noise convolved with a Gaussian kernel, volume size 64 × 64 × 32 voxels), and compare the predicted values for the mean and variance covariance matrix of u to the results of the simulations. Table 1 demonstrates the good agreement between the variance–covariance matrices computed experimentally using simulations (C_{exp}) and those predicted theoretically (C_{theo}) in 3D. The first-order approximations of *W*'s standard deviation (*W* = 4369, SD = 579.8) compared well to the empirically determined values (SD = 607.4).

TABLE 1. Mean and Variance—Covariance of the Vector u

	u_0	u_1	u_2	u_3
Experimental mean (× 10_s)	1.3112	0.0516	0.0689	0.1459
Theoretical mean (× 10_5)	1.3107	0.0515	0.0688	0.1457
Experimental covariance (× 10_7)				
	9.118	0.186	0.241	0.559

u_0				
u_1		0.010	0.005	0.011
u_2			0.018	0.015
u_3				0.095
Theoretical Covariance (× 10^7)				
u_0	8.670	0.170	0.228	0.661
u_0		0.010	0.005	0.010
u_2			0.018	0.014
u_3				0.083

Note. Top: from simulations, below: computed with the first-order approximation.

In Figure 1, we look at the impact of varying W, within the range of two standard deviations, on the p-values, for 3D processes (50000 pixels, $W = 18432$, SD$W = 6766$) corresponding to standard values for the resolution of PET based SPMs. Values of z and cluster size (k) have been chosen so that the probabilities come out around 0.05. Figure 1 shows that an overestimation of W has an opposite effect on the p-values for the two methods: it decreases the estimated p-value for the intensity method but it increases the estimated p-value for the cluster size assessment. Also, overestimating W has a stronger effect that an underestimation. In both cases, these effects are far from being negligible.

FIGURE 1. (a): Variation of cluster size probability for 3D data with the cluster size and with the variation of the smoothness estimate in ±2 SD (SDW), threshold Z = 3, size of the process; 50000 pixels, $W_0 = 18432$, SD$W = 6766$; dashed line, min. p-value, dot–dashed line, maximum p-value.(b): Variation of the z-value probability for 3D data with z and the variation of the smoothness estimate in ±2SDW. In this example, $W_0 = 18432$, SD$W = 6766$; dashed line; $W = W_0 + 2SD_W$; dot-dashed line, $W = W_0 - 2$SD$W$.

We computed the variation of the p-values obtained in an experimental activation study data set when W lies in the range $W \pm 2\mathrm{SD}_\mathrm{W}$. In this case, SD_W was 1706 with $W = 7634$, the size of the process (number of voxels) was 60545. In Table 2 we give a few examples of local maxima or cluster probability with p-values around 0.05 and their variation with the smoothness estimate ±2SDW.

TABLE 2. Range of Variation for P-Values Assessed from the Peak intensity (p_z) or Spatial extent (p_n) of an SPM Taken from an Activation Study

$(x, y, z) = (-38\ -68\ -8)$	$p_z = 0.049$	range: [0.041 0.066].
$(x, y, z) = (22\ -38\ 28)$	$p_z = 0.021$	range: [0.018 0.028].
$(x, y, z) = (-38\ -68\ -8)$	$p_n = 0.059$	range: [0.040 0.070].

> Read full chapter